МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКА НАЦІОНАЛЬНА АКАДЕМІЯ МІСЬКОГО ГОСПОДАРСТВА

О. В. Грицунов

ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

НАВЧАЛЬНИЙ ПОСІБНИК

Рекомендовано Міністерством освіти і науки України

Харків
ХНАМГ
2010

УДК 004.65(075)
ББК 32.98я73-6
Г85

Рецензенти:

Руденко Олег Григорович, завідувач кафедри електронних обчислювальних машин Харківського національного університету радіоелектроніки, доктор технічних наук, професор.

Дехтярук Леонід Васильович, професор кафедри фізики Харківського державного технічного університету будівництва та архітектури, доктор фізико-математичних наук.

Рекомендовано Міністерством освіти і науки України
як навчальний посібник для студентів вищих навчальних закладів
(лист № 1.4/18-Г-785 від 25.03.2010 р.)

Грицунов О. В.

Г85 Інформаційні системи та технології: навч. посіб. для студентів за напрямом підготовки «Транспортні технології» / О. В. Грицунов; Харк. нац. акад. міськ. госп-ва. – Х.: ХНАМГ, 2010. – 222 с.

ISBN 978-966-695-195-6

Навчальний посібник містить теоретичні відомості з основ розробки і використання інформаційних систем та технологій: принципів побудови сучасних баз даних та систем управління базами даних, організації діалогу користувача з інформаційною системою, методів збору, обробки, пересилки і використання транспортної та комерційної інформації.

УДК 004.65(075)
ББК 32.98я73-6

© О. В. Грицунов, 2010
© ХНАМГ, 2010

ISBN 978-966-695-195-6

ЗМІСТ

ВСТУП 7

Розділ 1. ІНФОРМАЦІЙНЕ СУСПІЛЬСТВО ТА ІНФОРМАЦІЙНІ СИСТЕМИ 10
1.1. Роль та місце інформаційних технологій у сучасному суспільстві… 10
1.2. Мета та завдання дисципліни 13
1.3. Терміни та визначення 14
1.4. Організація процесів керування …..... 18
1.4.1. Структура управління організацією 19
1.4.2. Персонал організації 22
1.4.3. Інші елементи організації 22
1.5. Реалізація і застосування інформаційних систем та технологій 23
1.5.1. Етапи розвитку інформаційних систем та технологій 25
1.5.2. Процеси в інформаційній системі 26
1.5.3. Приклади інформаційних систем 27
Питання й тести для самоконтролю 29

Розділ 2. БАЗИ ДАНИХ ТА БАНКИ ДАНИХ 30
2.1. Загальні принципи зберігання інформації 30
2.2. Організація банків даних 33
2.2.1. Переваги банків даних 33
2.2.2. Користувачі банків даних 34
2.2.3. Недоліки банків даних 36
2.3. Структура типового банку даних 36
2.3.1. Інформаційне забезпечення 37
2.3.2. Програмне забезпечення 38
2.3.3. Лінгвістичне забезпечення 39
2.3.4. Технічне забезпечення 43
2.3.5. Організаційно-методичне забезпечення 46
2.3.6. Адміністративне забезпечення 46
Питання й тести для самоконтролю 50

Розділ 3. КЛАСИФІКАЦІЯ ІНФОРМАЦІЙНИХ СИСТЕМ 51
3.1. Загальні принципи класифікації систем обробки інформації 51
3.1.1. Типи інформаційних систем для розв’язання частково структурованих задач 52
3.1.2. Альтернативні класифікації інформаційних систем 54
3.2. Класифікація баз та банків даних 56
3.2.1. Класифікація баз даних 56
3.2.2. Класифікація систем управління базами даних 64
3.2.3. Класифікація банків даних 66
Питання й тести для самоконтролю 69

Розділ 4. ЗБЕРІГАННЯ І ОБРОБКА ІНФОРМАЦІЇ В НАЙПРОСТІШИХ БАНКАХ ДАНИХ 70
4.1. Загальний опис системи програмування Microsoft Excel 70
4.1.1. Електронні таблиці 71
4.1.2. Основи роботи з системою Microsoft Excel 72
4.2. Створення банку даних на основі системи Microsoft Excel 74
4.2.1. Створення і заповнення списків 75
4.2.2. Сортування списків 77
4.2.3. Фільтрація даних 79
4.2.4. Формули і функції 83
4.3. Робота з банком даних на основі системи Microsoft Excel 86
4.3.1. Підведення підсумків 86
4.3.2. Створення звідних таблиць 89
4.3.3. Побудова діаграм 94
Питання й тести для самоконтролю 98

Розділ 5. ВИКОРИСТАННЯ МОВ ПРОГРАМУВАННЯ В ІНФОРМАЦІЙНИХ СИСТЕМАХ 99
5.1. Загальний опис мови VBA 100
5.1.1. Об'єкти мови 101
5.1.2. Властивості об'єктів 101
5.1.3. Події 102
5.2. Процедури й функції мови VBA 102
5.2.1. Створення користувальницьких функцій 105
5.3. Змінні в мові VBA 107
5.3.1. Типи змінних 108
5.3.2. Область видимості змінних 109
5.4. Керуючі конструкції мови VBA 110
5.4.1. Розгалуження 111
5.4.2. Цикли 116
5.5. Введення й вивід інформації в мові VBA 119
Питання й тести для самоконтролю 121

Розділ 6. СТВОРЕННЯ БАНКІВ ДАНИХ ЗА ДОПОМОГОЮ СУЧАСНОЇ РЕЛЯЦІЙНОЇ СУБД 122
6.1. Створення бази даних в MS Access 123
6.2. Створення таблиць 123
6.2.1. Опис характеристик таблиці 125
6.2.2. Задавання ключа 128
6.2.3. Характеристики полів 129
6.2.4. Збереження характеристик таблиці 131
6.2.5. Приклад створення таблиці 131
6.2.6. Коректування структури таблиці 132
6.2.7. Альтернативні способи задавання структури таблиць 133
6.3. Зв’язування таблиць 135

6.4. Задавання обмежень цілісності 137
6.4.1. Тип даних 138
6.4.2. Розмір поля 138
6.4.3. Умови на значення 139
6.4.4. Маска введення 140
6.4.5. Індексоване поле 141
6.5. Введення даних у базу 142
6.5.1. Введення даних у режимі «Таблиця» 142
6.5.2. Використання масок при введенні 143
6.6. Обмін інформацією між банками даних 144
6.6.1. Експорт таблиць з MS Access 146
6.6.2. Імпорт таблиць в MS Access 146
Питання й тести для самоконтролю..... 147

Розділ 7. РЕАЛІЗАЦІЯ ЗАПИТІВ В СУБД 149
7.1. Створення запитів в MS Access 150
7.1.1. Додавання в запит таблиць 151
7.1.2. Видалення із запиту таблиць 153
7.1.3. Додавання в запит полів 153
7.1.4. Задавання умов відбору 156
7.1.5. Керування виводом повторюваних рядків 156
7.1.6. Попередній перегляд результатів виконання запиту 157
7.1.7. Збереження запиту 158
7.2. Різновиди запитів 158
7.2.1. Простий запит 159
7.2.2. Складний запит 161
7.2.3. Запит до зв'язаних таблиць 162
7.2.4. Запит з обчислюваними полями 164
7.2.5. Коригувальні запити 166
Питання й тести для самоконтролю..... 170

Розділ 8. СКЛАДАННЯ ЕКРАННИХ ФОРМ ТА ЗВІТІВ 171
8.1. Створення нескладних форм за допомогою Майстра 173
8.2. Доробка форм у режимі Конструктора 177
8.2.1. Зміна включених у форму елементів керування 179
8.2.2. Включення нових елементів у форму 180
8.2.3. Зміна типу елемента керування 181
8.2.4. Створення форм, що складаються з декількох сторінок 182
8.2.5. Послідовність обходу полів 183
8.2.6. Властивості форми 184
8.3. Можливі причини помилок 185
8.4. Створення звіту за допомогою Майстра 186
8.5. Коректування та збереження звіту 194
Питання й тести для самоконтролю..... 195

Розділ 9. ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ 196
9.1. Поняття інформаційної технології 196
9.1.1. Визначення інформаційної технології 196
9.1.2. Нові інформаційні технології 197
9.1.3. Інструментарій інформаційних технологій 199
9.1.4. Співвідношення інформаційних систем та інформаційних технологій 200
9.1.5. Складові інформаційної технології 201
9.2. Етапи розвитку інформаційних технологій 202
9.2.1. Етапи розвитку задач та процесів обробки інформації 202
9.2.2. Етапи розв’язання проблем на шляху інформатизації суспільства 202
9.2.3. Етапи розвитку переваг, принесених комп’ютерними технологіями 203
9.2.4. Етапи вдосконалення інструментарію інформаційної технології 204
9.3. Види інформаційних технологій 205
9.3.1. Інформаційна технологія обробки даних 205
9.3.2. Інформаційна технологія керування 207
9.4. Інфологічні моделі 210
9.4.1. Основні поняття й визначення 210
9.4.2. Опис зв’язків 211
Питання й тести для самоконтролю 213

БІБЛІОГРАФІЧНИЙ СПИСОК 214

ПРЕДМЕТНИЙ ПОКАЖЧИК 215

ДОДАТКИ 218

Додаток А. Типові завдання для виконання РГР 219

ВСТУП

В останні десятиліття інформацію розглядають як один з основних ресурсів розвитку суспільства, а інформаційні системи та технології – як засіб підвищення продуктивності праці та ефективності роботи персоналу.

Переробка інформації – найважливіша функція, без якої немислима цілеспрямована діяльність будь-якої соціально-економічної, організаційно-виробничої системи (підприємства, організації, території і т.п.). Систему, що реалізує функції збору, зберігання, обробки і передачі інформації, називають **інформаційною системою** (ІС). Найважливішими функціями цих системам є прогнозування, планування, облік, аналіз, контроль і регулювання. Технології виконання таких функцій називаються інформаційними технологіями.

Найбільш широко інформаційні системи та технології використовуються у виробничій, управлінській і фінансовій діяльності, хоча почалися зсуви у свідомості людей, зайнятих в інших сферах, щодо необхідності впровадження й активного застосування таких систем і технологій.

Робота інформаційної системи пов'язана з обміном інформацією між її компонентами, а також системи з навколишнім середовищем. У процесі роботи персонал одержує відомості про стан системи в кожний момент часу, про досягнення (або не досягненні) заданої мети для того, щоб впливати на систему й забезпечити виконання управлінських рішень.

Інформаційна система – це сукупність внутрішніх і зовнішніх потоків прямого і зворотного інформаційного зв'язку певних об'єктів, засобів, фахівців, що беруть участь у процесі обробки інформації та виробленні управлінських рішень. Практично всі інформаційні системи в наш час організуються на основі засобів автоматики й обчислювальної техніки (комп'ютерів).

Автоматизована інформаційна система – сукупність інформації, економіко-математичних методів і моделей, технічних, програмних, технологічних засобів і фахівців, призначених для автоматизованої обробки інформації та прийняття управлінських рішень.

У навчальному посібнику «Інформаційні системи та технології» розглядаються питання розробки й ефективного застосування сучасних інформаційних систем і технологій для розв’язання різних класів економічних і управлінських задач у сьогоднішніх складних умовах функціонування підприємств. До них відносяться:

– економічний аналіз;

– обґрунтування стратегічних рішень;

– інвестиційне проектування;

– планування, реорганізація та моніторинг бізнес-процесів;

– адаптація корпоративних інформаційних систем;

– організація доступу до корпоративної пам'яті (сховища даних).

Метою навчального посібника є ознайомлення студентів, що навчаються за фахом «Транспортні технології», з проблематикою й областями використання сучасних програмних засобів в економічних та управлінських інформаційних системах, висвітлення теоретичних і організаційно-методичних питань побудови і функціонування систем, заснованих на реляційних базах даних, прищеплювання навичок практичних робіт із проектування банків даних. У результаті вивчення навчального посібника студенти одержать знання по архітектурі й класифікації інформаційних систем, областям їхнього застосування, методам зберігання та обробки даних, а також навчаться вибирати адекватні проблемній області інструментальні засоби розробки інформаційних систем і методи проектуванні структури баз даних, розробляти додатки в обраному інструментальному середовищі.

Навчальний посібник «Інформаційні системи та технології» може бути корисним також для студентів інших спеціальностей, які освоюють методи прийняття економічних та управлінських рішень, заснованих на науково обґрунтованому аналізі ситуації та доступі до систем збору, зберігання й обробки інформації.

Навчальний посібник може бути корисним для фахівців, що розробляють і використовують системи керування даними, системи підтримки прийн-

яття управлінських рішень, корпоративні інформаційні системи, а також для викладачів та аспірантів вищих навчальних закладів, що займаються дослідженнями в області застосування інформаційних технологій у керуванні різними об'єктами й процесами.

В основу методології викладу предмета покладено матеріали декількох відомих публікацій. Це, зокрема, навчальний посібник С.М.Діго за курсом «Бази даних» [1]. З урахуванням його змісту писалися розділи 2, 3, 6 – 8. Розділи 4 і 5 створювалися на базі підручника В.Кузьміна «Microsoft Office Excel 2003. Навчальний курс» [2]. Для розділів 1, 3 і 9 задіяні матеріали курсової роботи «Інформаційні системи і технології» студента гр. АСОІ-1-99 Нарінського державного університету Е.Іманалієва (керівник А.Токтакунов, 2003 р.) [3]. Використовувалися також методичні розробки викладачів кафедри ПМІТ Харківської національної академії міського господарства (ХНАМГ) проф. К.О.Метешкіна, доц. О.Б.Костенко та ас. Т.С.Сенчук. Для більш поглибленого вивчення дисципліни студенти можуть використати підручники та монографії [4] – [8].

Як програмну базу для підготовки матеріалу використано пакет програм Microsoft Office XP, зокрема, такі його додатки, як MS Excel і MS Access. Застосовувалися оригінальні (нерусифіковані) версії цих продуктів. Це зумовлено наступними міркуваннями. По-перше, русифіковані версії описано в переважній більшості численних російськомовних публікацій, і студент при бажанні може без зусиль ними скористатися. По-друге, за глибоким переконанням автора, знання хоча б основ англійської мови є настільки ж необхідним атрибутом повноцінного сучасного фахівця, як і знання основ застосування обчислювальної техніки. По-третє, ряд інших програмних продуктів надходить на український ринок у нерусифікованому вигляді, і англомовні меню не повинні бути проблемою для випускника вищого навчального закладу. Все це, разом узяте, обумовило вибір версії програмних продуктів для прикладів та ілюстрацій у цьому навчальному посібнику.

Розділ 1. ІНФОРМАЦІЙНЕ СУСПІЛЬСТВО ТА ІНФОРМАЦІЙНІ СИСТЕМИ

Згідно із прогнозами багатьох політологів, на сучасному етапі розвитку цивілізації відбувається поступовий перехід від індустріального до постіндустріального або інформаційного суспільства. Одне з його визначень таке: **інформаційне суспільство** – теоретична концепція постіндустріального суспільства; історична фаза можливого розвитку цивілізації, у якій основними продуктами виробництва є інформація й знання. Головні риси такого суспільства:

– збільшення ролі інформації та знань;

– зростання числа людей, зайнятих інформаційними комунікаціями та виробництвом інформаційних продуктів і послуг;

– створення глобального інформаційного простору, що забезпечує ефективну інформаційну взаємодію людей, їхній доступ до світових інформаційних ресурсів і задоволення їхніх потреб в інформаційних продуктах і послугах.

1.1. Роль та місце інформаційних технологій у сучасному суспільстві

Термін «інформаційне суспільство» і проекти створення такого суспільства вперше з'явилися в останній третині XX століття в США та країнах Західної Європи. Відтоді поняття інформаційного суспільства міцно зайняло своє місце не тільки в лексиконі фахівців з інформатики, але й у мові політичних діячів, економістів, вчених інших спеціальностей. Як правило, це поняття асоціюється з розвитком інформаційних технологій і засобів телекомунікацій, що дозволяють на базі громадянського суспільства (або, принаймні, декларованих його принципів) здійснити новий еволюційний стрибок, щоб ввести цивілізацію в XXI століття вже у вигляді інформаційного суспільства або, принаймні, його початкового етапу.

Згідно з вищевказаним, 27 березня 2006 року генеральна Асамблея ООН прийняла резолюцію під номером A/RES/60/252, що проголошує 17 травня Міжнародним днем інформаційного суспільства.

З огляду на глибину й розмах технологічних і соціальних наслідків комп'ютеризації і цифрового зв'язку в різних сферах громадського життя, процеси інформатизації нерідко називають комп'ютерною або інформаційною революцією. Більше того, західна суспільно-політична думка висунула різні варіанти концепції інформаційного суспільства, що мають своєю метою пояснити явища, породжені новим етапом науково-технічного прогресу. Про значимість і зростаючу популярність цієї концепції на Заході свідчить наростаючий потік публікацій на цю тему. В наш час у громадській думці концепція інформаційного суспільства поступово висувається на те ж саме місце, яке в 1970-х роках займала теорія постіндустріального суспільства.

Необхідно відзначити, що ряд західних політологів й економістів схиляються до того, щоб провести різку грань, що відокремлює концепцію інформаційного суспільства від постіндустріалізму. Однак, хоча концепція інформаційного суспільства покликана замінити теорію постіндустріального суспільства 1970-х років, її прихильники повторюють і далі розвивають ряд найважливіших положень технократизму, постіндустріалізму та традиційної футурології.

Характерно, що ряд провідних дослідників, що сформулювали теорію постіндустріального суспільства, у наш час виступають як прихильники концепції інформаційного суспільства. Концепція інформаційного суспільства стає своєрідним новим етапом розвитку теорії постіндустріального суспільства. Як констатував автор теорії постіндустріального суспільства Д. Белл, «революція в організації й обробці інформації й знання, у якій центральну роль грає комп'ютер, розвивається в контексті того, що я назвав постіндустріальним суспільством» [3].

Ключовим елементом інформаційного суспільства є системи зберігання, обробки й передачі цифрової інформації (у тому числі телекомунікації).

Не випадковим є той факт, що інформаційне суспільство затвердилося спочатку в тих країнах (Японія, США, Західна Європа), у яких в 1960...70-х роках сформувалися розвинені телекомунікаційні мережі.

Говорячи про інформаційне суспільство, його варто розуміти не в буквальному сенсі, а розглядати як орієнтир, тенденцію змін у сучасній цивілізації. У цілому ця модель орієнтована на майбутнє, але в розвинених капіталістичних країнах уже зараз можна назвати цілий ряд викликаних інформаційною технологією змін, які підтверджують концепцію інформаційного суспільства:

– структурні зміни в економіці, особливо в сфері розподілу робочої сили;

– зросле усвідомлення важливості інформації;

– зростаюче розуміння необхідності комп'ютерної грамотності;

– широке поширення інформаційних технологій;

– підтримка урядом розвитку комп'ютерної мікро- та наноелектронної технології й телекомунікацій.

Якісно новим моментом є можливість керування більшими комплексами організацій і виробничих систем, що вимагає координації діяльності сотень тисяч і навіть мільйонів людей. Іде бурхливий розвиток нових наукових напрямків, таких як інформаційна теорія, кібернетика, теорія прийняття рішень, теорія ігор і т.д., тобто напрямків, пов'язаних саме з проблемами організаційних множин.

Говорячи про зміни й зрушення, що сприяють переходу сучасного суспільства в якісно нову стадію, прихильники розглянутої концепції (інформаційного суспільства) опираються на об'єктивні процеси розвитку наукомістких, енерго- і працезберігаючих галузей економіки, процеси роботизації виробництва, комп'ютеризації й інформатизації найважливіших сфер суспільного й політичного життя. Дійсно, у цей час від новітніх наукомістких й енергозберігаючих компонентів техніки залежить вирішення таких життєве важливих питань, як економічне зростання, зайнятість, підвищення життєвого

рівня і т.д. Вони торкаються основних принципів функціонування й життєдіяльності сучасного суспільства, порушуючи кардинальні питання щодо соціальних і політичних змін, які несе із собою впровадження інформаційної технології. Це впливає на перспективу суспільно-історичного розвитку людства, на долю людини, його місце й роль у цьому процесі.

Інформатизація й комп'ютеризація вимагають від людей нових навичок, нових знань і нового мислення, покликаних полегшити адаптацію людини до умов і реалій комп'ютеризованого суспільства й забезпечити йому гідне місце в цьому суспільстві. Тому не можна не погодитися з тим, що інформатизація впливає на образ і якість життя всіх членів суспільства, як на індивідуальному, так і на організаційному рівні, на робочому місці та у побуті. Добре це чи погано, але вона є силою, що не просто трансформує життя цілих співтовариств, але сприяє перебудові самого процесу відносин між людьми.

1.2. Мета та завдання дисципліни

Як видно з попереднього підрозділу, інформаційні технології вже не можна розглядати як щось приналежне винятково світу техніки, тому що вони настільки глибоко проникли в життя людей, уплелися в саму тканину її повсякденності, що вичленувати їх із загального світоглядного й культурологічного контексту вже не бачиться можливим. Єдиний шлях – прийняти світ таким, який він є, і підкоритись його законам і правилам. А правила ці граничнo прості: або людина вивчає, щонайменше, основи інформатики і рухається далі разом з усім суспільством, або вона їх ігнорує, і в результаті неминуче виявляється «на узбіччі». Навчальна дисципліна «Інформаційні системи та технології» побудована так, щоб дати студентові, безпосередньо не зв'язаному по роду своєї спеціальності із цифровою обробкою інформації, той мінімум знань, що дозволить йому надалі успішно адаптуватися до сучасного інформатизованого світу.

Мета курсу «Інформаційні системи та технології» – формування у студентів стійких знань в області побудови сучасних інформаційних систем і технологій, а також вироблення навичок створення баз даних (БД) з використанням сучасних систем управління базами даних (СУБД) і банків даних.

Дана дисципліна відіграє важливу роль у виробленні у студентів професійних знань, умінь і навичок. Вона відноситься до блоку фундаментальних навчальних дисциплін й опирається на знання студентів, отримані при вивченні інформатики й вищої математики. Відомості, набуті студентами, що освоїли дану дисципліну, повинні бути розвинені при вивченні інших предметів. Вони стануть основою для самостійного освоєння нових, створюваних інформаційних систем та інноваційних технологій. Вивчаючи запропонований матеріал, студенти повинні його розглядати крізь призму всеосяжних, планетарних явищ сучасності – глобалізації й інтеграції різних сфер людської діяльності. Глибокі знання в області інформаційних систем і технологій забезпечать майбутнім бакалаврам і фахівцям високу конкурентноздатність на ринку праці й ефективне виконання своїх професійних обов'язків.

Курс «Інформаційні системи та технології» припускає вивчення основних дій з популярними СУБД Excel й Access, а також організацію їхньої взаємодії. Знання однієї з мов програмування високого рівня (Visual Basic for Applications, VBA) дасть майбутньому фахівцеві можливість вирішення нескладних розрахункових, логічних і графічних завдань. Дисципліна носить практичну спрямованість й у процесі її вивчення передбачається виконання розрахунково-графічної роботи, що виробляє у студента самостійність і навички в побудові окремих елементів інформаційних систем.

1.3. Терміни та визначення

В основі навчальної дисципліни «Інформаційні системи та технології» лежать терміни й визначення, з одного боку, що мають широке тлумачення, а з іншого боку – нормовані державними стандартами. До таких стандартів ві-

дносяться, зокрема, ДСТУ 2481-94 «Системи обробки інформації. Інтелектуальні інформаційні технології. Терміни й визначення», ДСТУ 2941-94 «Системи обробки інформації. Розробка систем. Терміни й визначення» і багато інших.

У наш час є безліч різних тлумачень терміна «інформація». Тому наведемо кілька визначень даному поняттю, як ключовому в назві даної дисципліни.

Інформація (від латинського informatio – роз'яснення, виклад), спочатку – відомості, передані людьми усним, письмовим або іншим способом (за допомогою умовних сигналів, технічних засобів і т.д.). Із середини XX століття – загальнонаукове поняття, що включає обмін відомостями між людьми, людиною й автоматом, автоматом та автоматом, обмін сигналами у тваринному й рослинному світі, передачу ознак від клітки до клітки, від організму до організму; одне з основних понять кібернетики.

Інформація, одна з основних універсальних властивостей предметів, явищ, процесів об'єктивної дійсності, людини й створених нею керуючих ЕОМ, що полягає в здатності сприймати внутрішній стан і впливи навколишнього середовища й зберігати певний час результати, перетворювати отримані відомості й передавати результати обробки (перетворення) іншим предметам, явищам, процесам, машинам, людям.

Інформація (для процесу обробки даних), будь-які знання про предмети, факти, поняття і т.п. проблемного середовища, якими обмінюються користувачі системи обробки даних.

Як видно, інформація одержала настільки широке поширення в житті цивілізації, що для неї недостатньо одного визначення. Аналогічна ситуація має місце із другим ключовим поняттям – «система». В інформатиці цей термін широко розповсюджений і має безліч змістових значень. Найчастіше він використається стосовно набору технічних засобів і програм. Системою може називатися апаратна частина комп'ютера або безліч програм, призначених

для вирішення конкретних прикладних завдань, доповнених процедурами ведення документації й керування розрахунками.

Система (від грецького systema – ціле, складене із частин; з'єднання), безліч елементів, що перебувають у відносинах і зв'язках один з одним, що утворять певну цілісність, єдність. Виділяють матеріальні й абстрактні системи. Перші розділяються на системи неорганічної природи (фізичні, геологічні, хімічні, технічні й ін.) і живі системи (біологічні системи – клітки, тканини, організми, популяції, види, екосистеми); особливий клас систем – соціальні системи (від найпростіших соціальних об'єднань до соціальної структури суспільства). Абстрактні системи – поняття, гіпотези, теорії, наукові знання про систему, лінгвістичні (мовні), формалізовані, логічні системи та ін. У сучасній науці дослідження систем різного роду проводиться в рамках системного підходу, загальної теорії системи, різних спеціальних теорій систем, у кібернетиці, системотехніці, системному аналізі і т.д.

У державних стандартах два розглянутих терміни поєднуються в єдине поняття – «інформаційна система». Запишемо його визначення.

Інформаційна система, система, що аналізує пам'ять і маніпулює інформацією про проблемну область.

Додавання до поняття «система» слова «інформаційна» відбиває мету її створення й функціонування. Інформаційні системи забезпечують збір, зберігання, обробку, пошук, видачу інформації, необхідної в процесі прийняття рішень завдань із будь-якої області. Вони допомагають аналізувати проблеми й створювати нові продукти.

Сучасне розуміння інформаційної системи припускає використання в якості основного технічного засобу переробки інформації персонального комп'ютера. У великих організаціях поряд з персональним комп'ютером до складу технічної бази інформаційної системи може входити мейнфрейм (mainframe) або супер-ЕОМ. Крім того, технічне втілення інформаційної системи саме по собі нічого не значить, якщо не враховано роль людини, для

якої призначено вироблену інформацію і без якої неможливі її одержання й обробка.

Необхідно розуміти різницю між комп'ютерами й інформаційними системами. Комп'ютери, оснащені спеціалізованими програмними засобами, є технічною базою й інструментом для інформаційних систем. Інформаційна система немислима без персоналу, взаємодіючого з комп'ютерами й телекомунікацій.

Крім вищенаведених основних, розглянемо також інші терміни, що мають відношення до дисципліни «Інформаційні системи та технології». До їхнього числа відносяться наступні.

Технологія (від грецького techne – мистецтво, майстерність, уміння), сукупність методів, способів і прийомів одержання, обробки або переробки сировини й напівфабрикатів з метою отримання готової продукції; наукова дисципліна, що вивчає механічні, фізичні, хімічні й інші зв'язки й закономірності, що діють у технологічних процесах. Іноді технологією називають також самі заходи щодо видобутку, обробки, транспортування, зберігання, контролю, що є частиною загального виробничого процесу.

Інтелектуальна інформаційна технологія, прийоми, способи й методи виконання функцій збору, зберігання, обробки, передачі й використання знань.

Дані, інформація, оформлена у формалізованому вигляді, зручному для пересилання, інтерпретації або обробки за участю людини або автоматичних засобів.

База даних, сукупність взаємозалежних даних, організованих згідно зі схемою бази даних так, щоб з ними міг працювати користувач.

Система управління базами даних, сукупність програмних та мовних засобів, які забезпечують керування базами даних.

Знання, сукупність фактів, закономірностей, відносин і евристичних правил, що відображає рівень поінформованості про проблеми деякої предметної області.

База знань, упорядкована сукупність правил, фактів, механізмів виводу й програмних засобів, що описує деяку предметну область і призначена для роботи з накопиченими в ній знаннями.

Система управління базою знань, сукупність програмних та апаратних засобів для організації й ведення бази знань.

1.4. Організація процесів керування

Для кваліфікованого створення й використання інформаційної системи необхідно чітко розуміти її призначення та функції. Як правило, у будь-якій організації (установі, фірмі) інформаційні системи та технології призначаються для вирішення конкретного кола завдань. Існують деякі загальні правила, яких варто дотримуватися, щоб впровадження інформаційних технологій виправдало покладені на нього очікування. До них, зокрема, відносяться такі:

– структура інформаційної системи, її функціональне призначення повинні відповідати цілям, що поставлено перед організацією. Наприклад, у комерційній фірмі – це ефективний бізнес; у державному підприємстві – вирішення соціальних й економічних завдань;

– інформаційна система повинна контролюватися людьми, ними розумітися й використатися відповідно до основних соціальних й етичних принципів;

– щоб результати виявилися адекватними, повинно бути організоване виробництво достовірної, надійної, своєчасної й систематизованої інформації.

Перед створенням і використанням інформаційної системи необхідно зрозуміти структуру, функції й політику організації, мету **керування** й прийняття рішень, можливості комп'ютерної технології. Інформаційна система є частиною організації, а ключові елементи будь-якої організації – структура й органи керування, стандартні процедури, персонал, субкультура.

Побудова інформаційної системи повинна починатися з аналізу структури керування організацією.

1.4.1. Структура управління організацією

Координація роботи всіх підрозділів організації здійснюється через органи управління різного рівня. Під **управлінням** розуміють забезпечення поставленої мети за умови реалізації таких функцій: організаційної, планової, облікової, аналізу, контрольної, стимулювання. Розглянемо зміст управлінських функцій.

Організаційна функція полягає в розробці організаційної структури та комплексу нормативних документів: штатного розкладу установи, відділу, лабораторії, групи і т.п. із вказівкою підпорядкованості, відповідальності, сфери компетенції, прав, обов'язків і т.д. Найчастіше це викладається в положенні щодо відділу, лабораторії або посадових інструкціях.

Планування (планова функція) полягає в розробці й реалізації планів по виконанню поставлених завдань. Наприклад, бізнес-план для всієї фірми, план виробництва, план маркетингових досліджень, фінансовий план, план проведення науково-дослідної роботи і т.д. на різні строки (рік, квартал, місяць, день).

Облікова функція полягає в розробці або використанні вже готових форм і методів обліку показників діяльності фірми: бухгалтерський облік, фінансовий облік, управлінський облік і т.п. У загальному випадку облік можна визначити як одержання, реєстрацію, нагромадження, обробку й надання інформації про реальні господарські процеси.

Аналіз або аналітична функція пов'язується з вивченням підсумків виконання планів і замовлень, визначенням факторів, що впливають, виявленням резервів, вивченням тенденцій розвитку і т.д. Виконується аналіз різними фахівцями залежно від складності й рівня об'єкта або процесу. Аналіз результатів господарської діяльності фірми за рік і більше проводять фахівці, а

на рівні цеху, відділу – менеджер цього рівня (начальник або його заступник) спільно з фахівцем-економістом.

Контрольна функція найчастіше здійснюється менеджером: контроль виконання планів, витрати матеріальних ресурсів, використання фінансових засобів і т.п.

Стимулювання або мотиваційна функція припускає розробку й застосування різних методів стимулювання праці підлеглих працівників:

– фінансові стимули – зарплата, премія, акції, підвищення в посаді і т.п.;

– психологічні стимули – подяки, грамоти, звання, ступені, дошки пошани і т.п.

В останні роки в сфері керування все активніше стали застосовуватися поняття «ухвалення рішення» і пов'язані із цим поняттям системи, методи, засоби підтримки прийняття рішень.

Ухвалення рішення – акт цілеспрямованого впливу на об'єкт керування, заснований на аналізі ситуації, визначенні мети, розробці програми досягнення цієї мети.

Структура керування будь-якої організації традиційно ділиться на три рівні: поточний, функціональний і стратегічний.

Рівні керування (вид управлінської діяльності) визначаються складністю розв'язуваних завдань. Чим складніше завдання, тим більш високий рівень керування потрібен для його вирішення. При цьому варто розуміти, що простих завдань, що вимагають негайного вирішення, виникає значно більша кількість, тому й рівень керування для них потрібен інший – більш низький, де приймаються рішення негайно. При керуванні необхідно також враховувати динаміку реалізації прийнятих рішень, що дозволяє розглядати керування під кутом часового фактора.

Рівні керування співвідносяться з такими факторами, як ступінь зростання влади, відповідальність, складність розв'язуваних завдань, а також динаміка прийняття рішень щодо реалізації завдань.

Поточний (нижній) рівень керування забезпечує вирішення багаторазово повторюваних завдань і швидке реагування на зміни вхідної поточної інформації. На цьому рівні досить великий як обсяг виконуваних робіт, так і динаміка прийняття управлінських рішень. На даному рівні великий обсяг займають облікові завдання, до яких, наприклад, відносяться:

– облік кількості проданої продукції;

– облік витрат часу, сировини й матеріалів при виконанні окремих виробничо-технологічних етапів;

– облік виробленої продукції;

– бухгалтерський облік і т.д.

Функціональний (тактичний) рівень керування забезпечує вирішення завдань, що вимагають попереднього аналізу інформації, підготовленої на першому рівні. На цьому рівні великого значення набуває така функція керування, як аналіз. Обсяг розв'язуваних завдань зменшується, але зростає їхня складність. При цьому не завжди вдається виробити потрібне рішення достатньо швидко, потрібен додатковий час на аналіз, осмислення, збір відсутніх відомостей і т.п. Керування пов'язане з деякою затримкою від моменту надходження інформації до прийняття рішень і їхньої реалізації, а також від моменту реалізації рішень до одержання реакції на них.

Стратегічний рівень забезпечує вироблення управлінських рішень, спрямованих на досягнення довгострокових стратегічних цілей організації. Оскільки результати прийнятих рішень проявляються через тривалий час, особливе значення на цьому рівні має така функція керування, як стратегічне планування. Інші функції керування на даному рівні в цей час розроблені недостатньо повно. Часто стратегічний рівень керування називають стратегічним або довгостроковим плануванням. Правомірність прийнятого на цьому рівні рішення може бути підтверджена через досить тривалий час. Можуть пройти місяці або роки. Відповідальність за прийняття управлінських рішень тут надзвичайно велика й визначається не тільки результатами аналізу з ви-

користанням математичного та спеціального апарату, але й професійною інтуїцією менеджерів.

1.4.2. Персонал організації

Персонал організації – співробітники різного ступеня кваліфікації й рівнів керування – від секретарів, що виконують найпростіші типові дії, до фахівців і менеджерів, що приймають стратегічні рішення.

На верхньому, стратегічному, рівні керування, як правило, перебувають менеджери вищої ланки керівництва організації (фірми) і їхні заступники. Основне їхнє завдання – стратегічне планування діяльності фірми на ринку й координація внутрішньофірмової тактики керування.

На середньому, функціональному, рівні розташовуються менеджери середньої ланки й фахівці (начальники служб, відділів, цехів, начальник зміни, ділянки, наукові співробітники і т.п.). Основне завдання тут – тактичне керування установою при вирішенні основних функцій у заданій сфері діяльності.

На нижньому, поточному, рівні – виконавці й менеджери нижчої ланки (бригадири, інженери, відповідальні виконавці, майстри, нормувальники, техніки, лаборанти і т.п.). Основне їхнє завдання – своєчасне реагування на зміну ситуації.

На всіх рівнях керування працюють як менеджери, що здійснюють тільки загальні функції, так і менеджери-фахівці, які реалізують функції керування в сфері своєї компетенції.

1.4.3. Інші елементи організації

Стандартні процедури в організації – це певні правила виконання завдань у різних ситуаціях. Вони охоплюють усі сторони функціонування організації, починаючи від технологічних процесів по складанню документів на

вироблену продукцію й кінчаючи розбором скарг споживачів.

Субкультура будь-якої організації – сукупність понять, принципів, типів поводження. Особливу роль грає найважливіша її складова – інформаційна культура фахівця. Це також повинне знаходити висвітлення в інформаційній системі.

Існує взаємозалежність між стратегією, правилами, процедурами організації й апаратними, програмними, телекомунікаційної частинами інформаційної системи. Тому дуже важлива на етапі впровадження й проектування інформаційних систем активна участь менеджерів, що визначають коло передбачуваних для вирішення проблем, завдань і функцій у своїй предметній області.

Варто відмітити також, що інформаційні системи самі по собі доходу не приносять, але можуть сприяти його одержанню. Вони можуть виявитися дорогими і, якщо їхня структура й стратегія використання не були ретельно продумані, навіть марними. Впровадження інформаційних систем пов'язане з необхідністю автоматизації функцій працівників, а тому сприяє їхньому вивільненню. Можуть також настати більші організаційні зміни в структурі фірми, які, якщо не враховано людський фактор і не обрана правильна соціальна й психологічна політика, часто проходять досить конфліктно.

1.5. Реалізація і застосування інформаційних систем та технологій

Як правило, інформаційні системи та технології використовуються з метою прийняття оптимальних рішень. У теорії систем виділяють як самостійно функціонуючі інформаційні системи, такі як інформаційно-розрахункові системи різного призначення, системи керування вантажо- і пасажироперевізками, геоінформаційні системи, інтелектуальні системи – експертні, що радять, та інші, так і функціонуючі в складі **автоматизованих систем управління** (АСУ). В автоматизованих системах управління вони відіграють роль інформаційної підсистеми, що здійснює збір, зберігання й пере-

дачу інформації споживачам.

У наш час все більша увага приділяється розробці інтелектуальних інформаційних систем, в основі яких лежать моделі, що реалізують інтелект людини. Застосування на практиці таких систем забезпечує створення інтелектуальних інформаційних технологій, які отримали широкий розвиток, як у рамках науки (теорії прийняття рішень, розпізнавання образів, передачі інформації та ін.), так і у процесі створення конкретних інтелектуальних програмних засобів.

Найбільш широко інформаційні системи та технології використовуються у виробничій, управлінській і фінансовій діяльності, хоча почалися рухи у свідомості людей, зайнятих і в інших сферах, щодо необхідності їхнього впровадження й активного застосування. Це визначає кут зору, під яким у цей час розглядаються основні області їхнього застосування. Головна увага приділяється розгляду інформаційних систем та технологій з позицій використання їхніх можливостей для підвищення ефективності праці працівників інформаційної сфери виробництва і підтримки прийняття рішень в організаціях (фірмах).

Як видно з попереднього підрозділу, впровадження інформаційних систем та технологій в організації спрямовано на досягнення таких основних цілей:

– одержання більш раціональних варіантів вирішення управлінських завдань за рахунок впровадження математичних методів та інтелектуальних систем;

– звільнення працівників від рутинної роботи за рахунок її автоматизації;

– забезпечення достовірності інформації;

– заміна паперових носіїв даних на електронні носії, що призводить до більш раціональної організації обробки інформації на комп'ютері й зниженню обсягів документів на папері;

– удосконалювання структури потоків інформації і системи документо-

обігу в установі;

– зменшення витрат на виробництво товарів і послуг;

– надання споживачам унікальних послуг;

– відшукання нових ринкових ніш;

– прив'язка до виробника споживачів і постачальників за рахунок надання їм різноманітних знижок і послуг.

1.5.1. Етапи розвитку інформаційних систем та технологій

Перші інформаційні системи з'явилися у 1950-х роках. В ті роки вони були орієнтовані на обробку рахунків і розрахунку зарплати, а реалізовувалися на електромеханічних бухгалтерських рахункових машинах. Це приводило до деякого скорочення витрат і часу на підготовку паперових документів.

В 1960-і роки відбувається зміна відношення до інформаційних систем і технологій. Інформація, отримана з їхньою допомогою, стала застосовуватися для періодичної звітності по багатьом параметрам. Для цього організаціям було потрібне комп'ютерне устаткування широкого призначення, здатне обслуговувати безліч функцій, а не тільки обробляти рахунки й нараховувати зарплату, як було раніше. Найбільш знаменитою відповіддю на потреби практики в даний період з'явилося створення ряду програмно-сумісних ЕОМ фірми International Business Machines IBM/360, а згодом – їх радянських «клонів» за назвою ЄС ЕОМ.

В 1970-х – на початку 1980-х років інформаційні системи починають широко використовуватися як засіб управлінського контролю, що підтримує й прискорює процес прийняття рішень, а також оптимізує саму суть цих рішень.

До кінця 1980-х років концепція використання інформаційних систем знову змінюється. Вони стають стратегічним джерелом інформації й використовуються на всіх рівнях організації будь-якого профілю. Інформаційні сис-

теми цього періоду, надаючи вчасно інформацію, необхідну рядовому співробітникові й керівникові, допомагають організації досягти успіху в діловій діяльності, створювати нові товари й послуги, знаходити нові ринки збуту, забезпечувати гідних собі партнерів, організовувати випуск продукції за низькою ціною й багато чого іншого.

Починаючи з 1990-х років, здійснюється новий перегляд головної області застосування інформаційних систем і технологій. Тепер вони розглядаються як основний інструмент глобалізації суспільства. Відбувається розвиток цифрових телекомунікаційних систем, що поєднують інформаційно-обчислювальні ресурси розвинених країн у єдину мережу. Різко зростає роль Інтернету в житті суспільства. Виникають нові технології різних сфер суспільної діяльності (наприклад, дистанційна освіта). Даний етап триває й у наш час.

1.5.2. Процеси в інформаційній системі

Процеси, що забезпечують роботу інформаційної системи будь-якого призначення, складаються з таких основних етапів:

– введення інформації із зовнішніх або внутрішніх джерел;

– обробка вхідної інформації й структуризація її в зручному виді;

– вивід інформації для передачі споживачам або в іншу систему;

– повторне введення або корекція незадовільно зібраної вхідної інформації.

Інформаційна система визначається наступними властивостями:

– будь-яка інформаційна система може бути піддана аналізу, побудована й керована на основі загальних принципів побудови систем;

– інформаційна система є динамічною й такою, що розвивається;

– при побудові інформаційної системи необхідно використати системний підхід;

– вихідною продукцією інформаційної системи є інформація, на основі

якої приймаються рішення;

– інформаційну систему варто сприймати як змішану (людино-комп'ютерну) систему обробки інформації.

У наш час переважна більшість інформаційних систем реалізується за допомогою обчислювальної техніки. Однак у загальному випадку інформаційну систему можна побудувати й у нецифровому варіанті.

Щоб розібратися в роботі інформаційної системи, необхідно зрозуміти суть проблем, які вона вирішує, а також організаційні процеси, у які вона включена. Так, наприклад, при визначенні можливості комп'ютерної інформаційної системи для підтримки прийняття рішень варто враховувати:

– структурованість розв'язуваних управлінських завдань;

– рівень ієрархії керування організацією, на якому рішення повинне бути прийняте;

– приналежність розв'язуваного завдання до тієї або іншої функціональної сфери бізнесу;

– вид використовуваної інформаційної технології.

Технологія роботи в комп'ютерній інформаційній системі в загальному випадку доступна для розуміння фахівцем некомп'ютерної області й може бути успішно використана для контролю процесів професійної діяльності й керування цими процесами.

1.5.3. Приклади інформаційних систем

Інформаційна система по відшуканню ринкових ніш. При придбанні товарів у деяких фірмах інформаційна система реєструє дані про покупця, що дозволяє:

– визначати групи покупців, їхній склад і запити, а потім орієнтуватися у своїй стратегії на найбільш численну групу;

– посилати потенційним покупцям різні пропозиції, рекламу, нагадування;

– надавати постійним покупцям товари й послуги в кредит, із знижкою, з відстрочкою платежів.

Інформаційні системи, що прискорюють потоки товарів. Припустимо, фірма спеціалізується на поставках продуктів у певну установу, наприклад, у кафе. Як відомо, мати більші запаси продуктів на складах фірми дуже невигідно, а не мати їх неможливо. Для того, щоб знайти оптимальне вирішення цієї проблеми, фірма встановлює термінали в установі, що обслуговується, і підключає їх до інформаційної системи. Замовник прямо з терміналу вводить свої побажання по надаваному йому каталогу. Ці дані надходять в інформаційну систему по обліку замовлень.

Менеджери, роблячи вибірки по замовленнях, що надійшли, приймають поточні управлінські рішення по доставці замовникові потрібного товару за короткий проміжок часу. Таким чином, заощаджуються величезні гроші на зберігання товарів, прискорюється і спрощується потік товарів, відслідковуються потреби покупців.

Інформаційні системи по зниженню витрат виробництва. Ці інформаційні системи, відслідковуючи всі фази виробничого процесу, сприяють поліпшенню керування й контролю, більш раціональному плануванню й використанню персоналу і, як наслідок, зниженню собівартості продукції й послуг.

Інформаційні системи автоматизації технології. Суть цієї технології (т. зв. «менеджмент поступок») полягає в тому, що, якщо дохід фірми залишається в рамках рентабельності, споживачеві робляться різні знижки залежно від кількості й тривалості контрактів. У цьому випадку споживач стає зацікавленим у взаємодії з фірмою, а фірма тим самим залучає додаткове число клієнтів. Якщо ж клієнт не бажає взаємодіяти з даною фірмою й переходить на обслуговування до іншої, то його витрати можуть зрости через втрату надаваних йому раніше знижок.

Інформаційні системи із продажу авіаквитків. Дозволяють проаналізувати архівні дані за багато років, оцінити перспективи наповнення салону,

призначити розумну ціну на кожне місце, знизити кількість непроданих квитків та ін. Відомо, наприклад, що в США кожне місце на літак резервується за три місяці до польоту 1,5 рази, тобто два місця резервуються за трьома пасажирами.

Інформаційні системи банків. Забезпечують всі види оплат по рахунках клієнтів. Вони навмисно робляться несумісними з інформаційними системами інших банків. Таким чином, клієнт попадає в коло послуг банку, з якого йому важко вийти. В обмін банк пропонує йому різні знижки й безкоштовні послуги.

Питання й тести для самоконтролю

1. Для чого необхідно вивчати навчальну дисципліну «Інформаційні системи та технології»?

2. Яке місце займає дисципліна «Інформаційні системи та технології» серед інших дисциплін, що забезпечують повноту знань у рамках Вашої спеціальності?

3. Дайте визначення основним термінам, які становлять термінологічний набір дисципліни «Інформаційні системи та технології».

4. Поясніть терміни «база даних», «база знань» і «система управління базою даних», «система управління базою знань». Співвіднесіть між собою ці поняття.

5. Що таке інтелектуальна інформаційна технологія і чим вона відрізняється від виробничої технології?

6. Опишіть основні процеси й завдання керування в типовій організації.

7. Як, по вашому, вплине впровадження інформаційної системи на роботу персоналу установи (фірми)?

8. Перелічить етапи розвитку інформаційних технологій.

9. Яку роль грають інформаційні системи (підсистеми) в автоматизованих системах управління?

10. Наведіть приклади інформаційних систем та інформаційних технологій.

Розділ 2. БАЗИ ДАНИХ ТА БАНКИ ДАНИХ

Як видно з попереднього розділу, інформація є таким же продуктом суспільного виробництва, як сировина або готові товари. Подібно матеріальним цінностям, вона повинна зберігатися в спеціально відведених для цього місцях у період між її одержанням і переробкою, а також після переробки, аж до споживання. За аналогією з сировинними базами та складами виробленої продукції, сховища інформації називаються базами даних і банками даних. При цьому базу даних можна ототожнити зі складським приміщенням, а банк даних – з організацією, що займається реалізацією складських послуг (товарно-сировинна база, елеватор, нарешті, власне банк, якщо інформацію ототожнити з фінансовими ресурсами). Інтуїтивне співвідношення між цими поняттями очевидне. Однак, щоб сформулювати його на науковому рівні, необхідно спочатку ознайомитися із загальними принципами зберігання даних.

2.1. Загальні принципи зберігання інформації

Існують два підходи до організації масивів даних на носіях інформації (наприклад, магнітних дисках): файлова організація та організація у вигляді баз даних.

Файлова організація припускає спеціалізацію й зберігання даних в **файлах**, орієнтованих, як правило, на одну прикладну задачу, і забезпечується самим прикладним програмістом. Така організація дозволяє досягти високої швидкості обробки даних, але має суттєві недоліки:

– значну надмірність даних. Вузька спеціалізація програм і файлів спричиняє необхідність зберігання однакових елементів даних у різних системах (дублювання інформації);

– неможливість виявлення суперечливих даних. Причина очевидна – для виконання тих самих дій з різними файлами потрібні різні програми, а керування здійснюється різними особами (групами осіб);

– низький рівень задоволення інформаційних запитів користувачів. Розроблені для спеціалізованих прикладних програм файли не можна використати для задоволення запитів користувачів, що перекривають дві й більше предметні області. Більше того, файлова організація даних не забезпечує виконання багатьох інформаційних запитів навіть у тих випадках, коли всі необхідні елементи знаходяться в наявних файлах.

Тому виникає необхідність відокремити дані від їхнього опису, визначити таку організацію зберігання даних з урахуванням існуючих зв'язків між ними, що дозволила б використати ці дані для багатьох додатків. Реалізація такого підходу привела до створення баз даних і систем керування базами даних, що мають метою забезпечувати ефективність процесів ведення й доступу до даних.

База даних – це сукупність взаємозалежних даних, що зберігаються спільно в зовнішній пам'яті обчислювального комплексу й використовуються, як правило, більш ніж однією програмою або користувачем.

До БД пред'являються певні вимоги:

– інтегрованість даних, тобто зберігання даних у єдиному сховищі;

– незалежність даних. Розрізняють логічну (зміна схеми БД не вимагає коректування прикладних програм) і фізичну (зміна методу організації даних не впливає ні на прикладні програми, ні на схему БД) незалежність даних;

– адекватність БД предметній області, тобто можливість відображення будь-яких фактів, що характеризують предметну область, повнота й несуперечливість даних, актуальність інформації (відповідність її стану відображуваної реальної системи на даний момент часу);

– цілісність (задоволення пропонованим логічним вимогам) даних;

– можливість взаємодії користувачів різних категорій та у різних режимах, забезпечення високої ефективності доступу для різних додатків;

– мінімальна надмірність (дублювання) збережених даних, що забезпечує необхідну продуктивність БД;

– здатність БД до розширення;

– можливість пошуку по декількох ключах;

– «приязність» інтерфейсів і малий час на освоєння системи, особливо для кінцевих користувачів;

– забезпечення захисту даних від несанкціонованого доступу або випадкового знищення збережених даних. Припускає введення заходів щодо ідентифікації користувачів і контролю їхніх дій з погляду наданих повноважень, а також відновлення БД при апаратних збоях обчислювального комплексу;

– прийнятні економічні й фізичні характеристики функціонування БД (вартість обробки, час реакції системи на запити, необхідні машинні ресурси та ін.).

По частоті відновлення й поповнення даних розрізняють БД реального часу, мінливі, статичні.

БД реального часу допускають поповнення, зміну й видалення даних, що утримуються в них, безпосередньо по запитах користувачів у реальному масштабі часу. Бази даних цього типу використовуються в першу чергу для організації інформаційної взаємодії окремих **підсистем** програмного забезпечення і є найбільш складними в реалізації.

Для мінливих БД характерним є те, що функція модифікації даних виконується людиною (адміністратором БД) за заявками користувачів через певні (досить тривалі) проміжки часу.

Статичні БД на відновлення даних не розраховані й припускають, як правило, зберігання інформації нормативно-довідкового характеру.

Всі функції по організації, обслуговуванню й доступу до БД виконуються за допомогою спеціального програмного забезпечення, що носить назву СУБД.

Система управління базами даних – програмна система, що забезпечує використання й ведення БД. Основне призначення СУБД – надання користувачам БД засобів маніпулювання даними в абстрактних термінах, не пов'язаних зі способом їхнього зберігання в обчислювальній системі.

СУБД гарантує несуперечність, цілісність, таємність і мінімальну надмірність даних у БД. Ефективність СУБД визначається швидкістю доступу до даних, раціональним використанням пам'яті обчислювального комплексу, простотою розробки прикладних програм, призначених для доступу до даних з бази.

Як правило, взаємодіючі між собою БД і СУБД, у сукупності називаються **банком даних** (БНД).

2.2. Організація банків даних

Описані вище загальні принципи зберігання інформації можуть здатися зайво абстрактними й далекими від вирішення практичних завдань. Це наслідок універсальності даних понять, орієнтованих на охват усього розмаїття додатків інформаційних систем та технологій. Розглянемо тепер організацію БД і СУБД більш конкретно. Зазначимо відразу, що термінологія в даній області сильно розрізняється залежно від місця видання літератури і навіть від конкретних БД. За кордоном замість терміна «банк даних» частіше використовується словосполучення «система бази даних» (database system). Більше того, іноді під базою даних розуміється практично будь-яка сукупність даних, що може бути оброблена за допомогою ЕОМ. Іноді це виправдано, тому що, наприклад, права власності й інші права не можуть залежати від того, за допомогою якого програмного засобу створено файли, і який у них спосіб організації. Але, з іншого боку, таке широке тлумачення терміна «база даних» може призвести до нівелювання особливостей банків даних як особливої інформаційної технології.

2.2.1. Переваги банків даних

Особливості організації даних у вигляді БНД визначають їхні основні переваги перед файловою організацією. Наявність єдиного відображення пе-

вної частини реального світу дозволяє забезпечити несуперечність і цілісність інформації, можливість звертатися до неї не тільки при вирішенні заздалегідь визначених завдань, але й з нерегламентованими запитами. Інтегроване зберігання скорочує надмірність збережених даних, що призводить до скорочення витрат не тільки на створення й зберігання даних, але й на підтримку їх в актуальному стані. Використання БНД при правильній їхній організації повинне істотно змінити діяльність установи, де вони впроваджуються: привести до забезпечення більшої доступності даних для всіх категорій співробітників, скороченню документообігу, можливості одержання різноманітних за формою й змістом документів, перерозподілу функцій між співробітниками, зміні характеру виконуваних функцій та, як наслідок, поліпшенню всієї системи керування підприємством. Централізоване керування даними також дає цілий ряд переваг.

Використання СУБД забезпечує високу якість виконання функцій керування даними й полегшує процес створення інформаційних систем. Виділення спеціальної групи співробітників, що виконують функції з проектування й розвитку БНД (адміністраторів БНД), і звільнення від цих функцій всіх інших користувачів не тільки приводить до зниження вимог до інших учасників процесу створення й функціонування БНД, але й підвищує якість розробок, тому що питаннями організації даних займається невелике число професіоналів у цій галузі. Перевагою банків даних є також те, що вони забезпечують можливість більш повної реалізації принципу незалежності прикладних програм від даних, ніж це можливо при організації файлового зберігання.

2.2.2. Користувачі банків даних

У процесі створення й експлуатації БНД з ним взаємодіють **користувачі** різних категорій. Бази даних створюються для задоволення потреб кінцевих користувачів. Найчастіше – це фахівці конкретних предметних областей, що використовують БД для виконання своїх професійних обов'язків.

Останнім часом БД все частіше використовуються і для задоволення невиробничих інформаційних потреб. **Кінцеві користувачі** – найбільш численна група користувачів.

Категорія «кінцеві користувачі» неоднорідна: кінцеві користувачі розрізняються широтою інформаційних потреб, кваліфікацією, режимами взаємодії із БНД та ін. Це можуть бути випадкові користувачі, що звертаються до бази даних час від часу, а можуть бути й регулярні користувачі. Кінцеві користувачі можуть відрізнятися один від одного і ступенем володіння обчислювальною технікою. Від кінцевих користувачів не можна вимагати якихось спеціальних знань в області обчислювальної техніки і мовних засобів. При створенні БНД важлива не тільки побудова класифікаційної схеми, але й розподіл реальних кінцевих користувачів по групах, тому що від характеристики користувачів будуть залежати прийняті проектні рішення.

У зв'язку з тим, що використання БНД впливає на всі аспекти діяльності організації, особливу роль грають керівники організації. Саме вони повинні забезпечити проведення єдиної інформаційної політики та організацію взаємодії різних підрозділів через загальну базу даних. Вони повинні створювати підрозділи, відповідальні за організацію та функціонування БНД, визначати функціональні обов'язки співробітників, які істотно зміняться із впровадженням БНД. Крім того, керівники організації виступають як кінцеві користувачі з найбільш високим пріоритетом.

Окремі користувачі в процесі роботи з базою даних можуть змінювати зміст БД. Інші тільки використовують збережену в БД інформацію. Користувачі можуть взаємодіяти з БД як безпосередньо (термінальні користувачі), так і через посередників. Поняттям «кінцеві користувачі» визначається не тільки окрема особа або група осіб, але й обчислювальні процеси (завдання), а іноді й цілі системи, що взаємодіють із БНД.

Залежно від особливостей створюваного банку даних, коло осіб, що мають доступ до нього, може істотно розрізнятися. Однак серед них завжди присутні адміністратори БНД – особи, відповідальні за створення БНД і його

надійне функціонування, за дотримання регламенту доступу до збереженої інформації, за розвиток БНД.

Наявність у складі СУБД засобів, орієнтованих на різні категорії користувачів, уможливлює роботу з базою даних не тільки професіоналів в області обробки даних, але практично будь-яких співробітників установи, причому це використання може бути як для їхніх професійних цілей, так і для задоволення особистих потреб в інформації.

2.2.3. Недоліки банків даних

Недоліки БНД є побічним результатом їхніх переваг («за все треба платити»). Створення інтегрованої системи, природно, складніше, ніж створення безлічі локальних систем. Як наслідок, пред'являються високі вимоги до кваліфікації розробників БНД.

У результаті інтеграції можлива деяка втрата ефективності окремих додатків (але загальна ефективність всієї системи буде вище). Для керування даними потрібне спеціалізоване програмне забезпечення, що, залежно від класу системи, може бути порівняно дорогим, та пред'являти підвищені вимоги до технічних засобів.

Експлуатація розподілених корпоративних БНД – процес складний і дорогий. Але, незважаючи на деякі недоліки, властиві такій формі організації даних, переваги БНД значно перевершують їхні недоліки. Крім того, є дуже широке коло СУБД різних класів, а також технологій їхнього використання. Правильний вибір системи дозволяє звести негативні наслідки до мінімуму.

2.3. Структура типового банку даних

Будь-який банк даних є складною програмно-технічною системою, що включає різні взаємозалежні компоненти. Як правило, у складі БНД виділяють такі **забезпечення**:

– інформаційне забезпечення (базу даних);

– програмне забезпечення;

– лінгвістичне забезпечення;

– технічне забезпечення;

– організаційно-методичне забезпечення;

– адміністративне забезпечення.

Розглянемо кожну з цих складових докладніше.

2.3.1. Інформаційне забезпечення

База даних є ядром БНД. У технічній документації деяких СУБД, а також у деяких літературних джерелах до складу БД включаються не тільки власне збережені дані про предметну область, але й описи БД. Більш правильно описи баз даних вважати самостійними компонентами БНД, навіть якщо вони й зберігаються разом із самими даними. Описи баз даних відносяться до метаінформації, тобто інформації про інформацію. Опис баз даних часто називають схемою. Крім того, у БНД можуть бути присутні описи окремих частин бази даних з точки зору конкретних користувачів. Такий опис називається підсхемою. Крім опису баз даних, до складу метаінформації, збереженої в БНД, може включатися інформація про предметну область, необхідна для проектування автоматизованої інформаційної системи, про користувачів БНД, про проектні рішення й деяка інша інформація.

Централізоване сховище метаінформації називається словником даних. У літературі використовуються також терміни «словник-довідник», «енциклопедія», «репозиторій». У деяких джерелах виявляються розходження між цими термінами, в інших вони використовуються як синоніми. Для нас ці розходження несуттєві.

Роль словникової системи особливо зростає при використанні засобів автоматизованого проектування інформаційних систем. Для більшості із цих

засобів «репозиторий» є ядром всієї системи. Крім того, роль «репозиторію» особливо значима в розподілених системах.

До банку даних не відносяться немашинні документи, що слугують джерелами інформації, що вводять у БД: файли вхідної й вихідної інформації, архівні файли, вихідні документи. Однак багато СУБД включають у свій склад мовні засоби для опису цих компонентів. У цьому випадку описи, використовувані в процесі функціонування БНД, будуть входити в його склад.

Як ми вже відзначали вище, термінологія, використовувана в різних системах і різних літературних джерелах, істотно розрізняється. Так, у ранніх версіях багатьох СУБД взагалі не використовувався механізм баз даних (тобто створювалися фактично окремі таблиці, кожна з яких запам'ятовувалася в окремому файлі бази даних). У багатьох літературних джерелах кожен такий файл стали називати базою даних, що неправильно. У деяких системах, наприклад, Microsoft Access, під БД розуміють сукупність різних об'єктів: таблиць, запитів, форм, звітів, макросів і модулів, тобто поняття бази даних розширено і містить у собі практично всі інформаційні компоненти, створені для конкретного додатка. В інших системах, зокрема, в Paradox, для позначення подібної сукупності взаємозалежних об'єктів використовується поняття «сімейство», що, мабуть, термінологічно більш правильно. При роботі з конкретною системою треба, насамперед, уточнити термінологію, використовувану в ній.

2.3.2. Програмне забезпечення

Програмні засоби БНД – складний програмно-алгоритмічний комплекс, що забезпечує взаємодію всіх частин інформаційної системи при її функціонуванні.

Основу програмного забезпечення БНД становлять програмні компоненти СУБД. Серед них можна виділити ядро СУБД, що забезпечує створення БД, організацію введення, обробки та зберігання даних, тобто саме те, що

називається «керуванням даними», а також інші компоненти, що забезпечують настроювання системи, засоби тестування, утиліти, що виконують допоміжні функції, такі як відновлення бази даних, збір статистики про функціонування БНД тощо. Важливим компонентом СУБД є транслятори або компілятори для використовуваних нею мовних засобів.

До складу більшості СУБД включено програмні компоненти, що дозволяють автоматизувати проектування систем обробки інформації (генератори звітів, меню та ін.). Взагалі, ці функції не є безпосередньо функціями керування даними, але більшість сучасних програмних засобів, які продовжують називатися СУБД, виходять за названі рамки й фактично є потужними комплексними інструментальними засобами, що дозволяють автоматизувати процес створення інформаційних систем.

Для задоволення конкретних потреб користувачів пишуться відповідні програми, які є прикладним програмним забезпеченням БНД. Поняття «програма» тут трактується широко. Це можуть бути, наприклад, **екранні форми**, створені з використанням візуальних засобів, запити, написані на будь-якій мові запитів і т.п. Залежно від використовуваних технологій створення й функціонування систем можуть з'являтися ті або інші додаткові компоненти. Так, при роботі в архітектурі клієнт-сервер програмні засоби будуть підрозділятися на відповідні компоненти: клієнтська частина, що забезпечує інтерфейс користувача із системою, серверна частина, що реалізує обробку запиту на сервері, і зв'язуюча частина, що забезпечує взаємодію елементів у мережі. Програмні засоби, використовувані при створенні й експлуатації БНД, залежать також від масштабу БНД і вимог до нього.

2.3.3. Лінгвістичне забезпечення

Лінгвістичні засоби СУБД є найважливішим компонентом банків даних, тому що, в остаточному підсумку, вони забезпечують інтерфейс користувачів різних категорій з банком даних. Набір використовуваних мовних

засобів широкий і різноманітний. Мовні засоби, використовувані в БНД, можна класифікувати за різними ознаками.

Мовні засоби більшості сучасних СУБД відносяться до мов четвертого покоління (до першого покоління мов відносять машинні мови, до другого – символічні мови асемблера, до третього – **алгоритмічні мови** типу PL/I, COBOL і т.п., які в 1960-і роки називалися мовами високого рівня, але рівень яких набагато нижче, ніж у мов четвертого покоління. Є ще мови п'ятого покоління, до якого відносять мови систем штучного інтелекту.

Мови четвертого покоління створювалися за принципом: «люди коштують дорожче, ніж машини». При їхньому проектуванні використовуються такі принципи:

– мінімуму роботи (мова повинна забезпечити мінімум зусиль, щоб «змусити» машину працювати);

– мінімуму майстерності (робота повинна бути так проста, як тільки це можливо, вона не повинна бути інструментом обраних і зрозумілою лише присвяченим);

– природності мови, скасування «стороннього» синтаксису й мнемоніки (мова не повинна вимагати від користувачів значних зусиль у вивченні синтаксису або містити багато мнемонічних або інших позначень, які швидко забуваються);

– мінімуму часу (мова повинна дозволяти без істотної затримки реалізовувати виникаючі потреби в доступі до інформації і її обробці);

– мінімуму помилок (технологія повинна бути спроектована таким чином, щоб мінімізувати помилки людини, а якщо вони виникли, то, по можливості, «виловити» їх автоматично);

– мінімуму підтримки (механізм мов четвертого покоління повинен дозволяти легко вносити зміни в наявні додатки);

– максимуму результату (мови четвертого покоління надають користувачам потужний інструмент для вирішення різноманітних завдань).

Можна виділити дві концепції розвитку мовних засобів: концепцію поділу й концепцію інтеграції. При використанні концепції поділу розрізняють мови опису даних (МОД) і мови маніпулювання даними (ММД). Призначення кожного із цих підкласів зрозуміле з їхньої назви. Іноді в особливу групу виділяють мови запитів (МЗ). Спочатку під мовами запитів розуміли мови високого рівня, орієнтовані на кінцевого користувача, тобто призначені для формування запитів до БД (у такому трактуванні їх можна вважати однією з різновидів ММД). Однак зараз МЗ розуміється ширше: багато МЗ містять у собі також можливості опису даних і коректування БД.

Мови маніпулювання даними розділяються на дві більші групи: процедурні й непроцедурні. При користуванні процедурними мовами треба вказати, які дії та над якими об'єктами необхідно виконати, щоб одержати результат. У непроцедурних мовах вказується, що треба одержати у відповіді, а не як цього досягти.

Процедурні мови можуть розрізнятися основними інформаційними одиницями, якими вони маніпулюють. Це можуть бути мови, орієнтовані на обробку в один прийом одного запису даних, і мови, орієнтовані на дії одночасно над безліччю записів. Типовою мовою другого типу є широко використовувана мова структурованих запитів SQL (Structured Query Language).

Мовні засоби призначаються для користувачів різних категорій: кінцевих користувачів, системних аналітиків, професійних програмістів. Підвищення рівня мовних засобів, їх приязності призводить до того, що все більше число функцій виконується користувачами-непрограмістами самостійно, без посередників.

З точки зору функціональних можливостей звичайно виділяють такі категорії мов:

– мови, що підтримують тільки можливості запитів. Вони забезпечують вивід необхідних даних на екран або друк в потрібному форматі. У наш час використовуються рідко;

– комплексні мови запитів/відновлень. Більш розвинені мови, які дозволяють формулювати складні запити, що відносяться до декількох взаємозалежних записів, а також обновляти дані так само легко, як і формулювати запити. Використовуючи їх, користувачі можуть створювати свої власні файли;

– генератори звітів. Вони дозволяють вибирати потрібні дані з файлів або баз даних і форматувати їх у вигляді необхідних форм документів;

– графічні мови. Використання графічних засобів у наш час постійно розширюється. Ці мови дозволяють виводити дані у вигляді різних графіків і діаграм, а також використовувати інші наочні можливості. Так само як генератори звітів, графічні мови дозволяють здійснювати відбір інформації з файлів або баз даних за різними критеріями, а також виконувати арифметичні й логічні маніпуляції з даними;

– інструментальні засоби підтримки рішень. Мови цього типу призначені для створення систем прийняття рішень. Це можуть бути системи за принципом «що буде, якщо...?», системи, що виконують тимчасовий або трендовий аналіз та інші. Можливе використання як універсальних, так і проблемно-орієнтованих засобів;

– генератори додатків. Мовні засоби, призначені для генерації додатків, забезпечують можливість опису непроцедурним шляхом необхідної обробки інформації й подальшої автоматичної генерації програм;

– машинно-орієнтовані мови специфікацій. Фактично є генераторами додатків (точніше, подальшим їхнім розвитком). На відміну від генераторів додатків, мови специфікацій більш універсальні й дозволяють специфікувати додатки різних типів;

– мови дуже високого рівня. У більшості випадків додатки будуються за допомогою непроцедурних мов. Однак деякі мови є процедурними, але програмування на них значно коротше, ніж, наприклад, на мовах типу COBOL;

– пакети прикладних програм (ППП). Ця категорія програмних засобів відома давно й «четверте покоління» відноситься до таких ППП, які допускають легку модифікацію самого пакета, дозволяють користувачам генерувати власні звіти, запити до бази даних, і т.д.

Багато мов четвертого покоління є універсальними мовами. Інші – спроектовані для специфічних додатків. Прикладами таких мов є мови для керування фінансами, управління роботою верстатів із програмним керуванням і т.д. За формою запису розрізняють аналітичні, табличні й графічні мовні засоби. Класифікація мовних засобів за формою запису відноситься як до мов опису даних, так і до мов маніпулювання даними.

Крім згаданих мовних засобів, БНД містять у собі генератори екранних форм, звітів і додатків, а також мови розгалуженої ієрархічної системи типу «меню», що дозволяє користувачеві вибрати ті дії, які він бажає виконати. Часто СУБД забезпечують автоматичне перетворення «текстів» з однієї мови на іншу. Так, наприклад, чимало СУБД, таких як Microsoft Access, FoxPro та інші, використовують мови запитів табличного типу не тільки для безпосередньої реалізації запитів, але й як засіб для більш простого опису запиту і подальшого автоматичного перетворення його на мову SQL.

2.3.4. Технічне забезпечення

Як технічні засоби для банків даних найчастіше використовуються універсальні ЕОМ, периферійні засоби для введення інформації в базу даних і для відображення інформації, що виводиться. Іноді використовуються додаткові технічні засоби для зберігання більших обсягів даних на зовнішніх носіях. Якщо банк даних реалізується в мережі – необхідні відповідні технічні засоби для забезпечення її роботи. Склад і тип технічних засобів, на яких реалізуються БНД, залежить від багатьох факторів. Основними з них є: технічні характеристики устаткування, використовувані технології обробки даних,

масштаб системи, обмеження на час реакції системи, складність обробки, вартісні характеристики та ін.

Спочатку БНД реалізовувалися, в основному, на великих ЕОМ, а для доступу до БД використовувалися термінали. У зв'язку зі значним і постійним поліпшенням характеристик персональних ЕОМ, з'явилася можливість реалізовувати банки даних і на машинах цього класу. Але спочатку характеристики персональних ЕОМ були недостатніми, щоб повною мірою реалізувати ідеологію банків даних. Як наслідок, стала спостерігатися деяка роздробленість інформаційних систем, що, у свою чергу, привело до бурхливого розвитку мережних технологій і використання відповідних їм технічних засобів.

Існують і спеціалізовані технічні засоби, призначені для створення й експлуатації банків даних (машини баз даних), але вони не знайшли широкого розповсюдження. В останні роки деякі фірми (Oracle, Sun) активно розвивають ідею використання так званих «мережних комп'ютерів». Ці комп'ютери – фактично дешеві робочі станції без дискових накопичувачів (пізніше з'явилися «мережні комп'ютери» із власними накопичувачами), які працюють у мережі й використовують програмні засоби і дані, що перебувають на сервері. Використання «мережних комп'ютерів» припускає обов'язкове застосування потужних ЕОМ як серверів, висуває високі вимоги до організації зберігання даних, до якості каналів зв'язку. При цьому стає багато в чому стандартизованою технологія обробки даних (особливо в частині розподілу функцій між клієнтом і сервером).

Використання «мережних комп'ютерів» обумовлено не стільки спробою заощадити кошти за рахунок використання більш дешевих комп'ютерів, скільки бажанням упорядкувати використання програмних засобів, спростити систему обробки інформації в цілому, полегшити і здешевити підтримку системи. Недоліками такого підходу є:

– більша залежність від «центральної» системи, втрата самостійності кінцевими користувачами;

– вразливість системи;

– неможливість або неефективність забезпечення потреби всіх користувачів. Хоча потреби користувачів і дублюються, але ступінь дублювання може бути різною. Крім того, не виключається наявність суттєво персональних даних, зберігання яких на віддаленому сервері призводить до непродуктивних витрат;

– високі вимоги до серверної частини системи.

Іншим новим явищем є використання кишенькових персональних комп'ютерів (КПК) як комунікаційних пристроїв для доступу до корпоративних даних у глобальних мережах. Характеристики кишенькових комп'ютерів істотно поліпшуються. Для них створюється відповідне програмне забезпечення, що дозволяє використати їх для мобільних користувачів, що працюють у загальній системі (тиражування й синхронізація даних). Ці комп'ютери є більш легкими, що важливо для мобільних користувачів, а також коштують дешевше, ніж звичайні персональні комп'ютери. Провідні виробники СУБД пристосовують свої великомасштабні серверні системи для доступу до них за допомогою КПК.

Тип використовуваних ЕОМ залежить також від масштабу створюваної системи. У наш час у переважній більшості випадків БНД реалізуються в мережному середовищі з використанням безлічі різнотипних ЕОМ, причому їхній склад постійно змінюється в процесі експлуатації банку даних.

Технічні засоби БНД не обмежуються тільки ЕОМ. Сюди входить весь комплекс технічних засобів зберігання, відображення й передачі інформації. Особливу роль для забезпечення ефективного й надійного функціонування банку даних відіграють засоби зберігання інформації. Пам'ять у БНД звичайно організується у вигляді багаторівневої системи. Необхідно звертати увагу не тільки на вибір запам'ятовуючих пристроїв для організації зберігання даних, призначених для поточного доступу до них, але й для зберігання архівних даних. У банках даних, як і у всіх інших інформаційних системах, виконуються дії по введенню, зберіганню, обробці й виводу інформації. При ви-

конанні кожного з цих дій можуть використовуватися різні технології та, як наслідок, різні технічні й програмні засоби для їхньої підтримки.

2.3.5. Організаційно-методичне забезпечення

Під організаційно-методичними засобами банку даних звичайно розуміють різні інструкції, методичні й регламентуючі матеріали, призначені для користувачів різних категорій, що взаємодіють з банком даних. Це можуть бути інструкції кінцевим користувачам по роботі з базою даних, документи, що визначають права доступу й регламент роботи; сюди ж відносяться й методики проектування баз даних.

2.3.6. Адміністративне забезпечення

Функціонування БНД неможливе без участі фахівців, що забезпечують його створення, функціонування й розвиток. Така група фахівців називається адміністраторами банку даних. Залежно від складності й обсягу банку даних, від особливостей використовуваної СУБД, служба адміністрації банку даних може розрізнятися як по складу й кваліфікації фахівців, так і по кількості працюючих у цій службі.

Адміністратори виконують роботи зі створення й забезпечення функціонування БНД протягом всіх етапів життєвого циклу інформаційної системи. У складі групи адміністраторів банку даних можна виділити різні підгрупи, залежно від виконуваних ними функцій. Чисельність групи адміністраторів і виконувані ними функції в значній мірі залежать від масштабу банку даних, специфіки збереженої в ньому інформації, типу БНД, особливостей використовуваних програмних засобів і деяких інших факторів.

У складі адміністраторів БНД повинні бути системні аналітики, проектувальники структур даних і зовнішнього (стосовно банку даних) інформаційного забезпечення, проектувальники технологічних процесів обробки да-

них, системні та прикладні програмісти, оператори, фахівці з технічного обслуговування. Якщо мова йде про комерційний банк даних – важливу роль грають фахівці з маркетингу.

Адміністратори банку даних виконують велике коло різноманітних функцій, у тому числі:

– аналіз предметної області: опис предметної області, виявлення обмежень цілісності, визначення статусу інформації, потреб користувачів, статусу користувачів, відповідності «дані-користувач», об'ємно-часових характеристик обробки даних;

– проектування структури бази даних: визначення складу й структури інформаційних одиниць, що становлять базу даних, задавання зв'язків між ними, вибір методів упорядкування даних і методів доступу до інформації, опис структури БД на МОД;

– задавання обмежень цілісності при описі структури бази даних і процедур обробки БД: задавання обмежень цілісності, властивих предметної області, визначення обмежень цілісності, викликаних структурою бази даних, розробку процедур забезпечення цілісності БД при введенні й коректуванні даних, забезпечення обмежень цілісності при паралельній роботі користувачів у багатокористувацькому режимі;

– первісне завантаження й ведення бази даних: розробку технології первісного завантаження й ведення (зміни, додавання, видалення записів) БД, проектування форм введення, створення програмних модулів, підготовку вихідних даних, введення й контроль введення;

– захист даних від несанкціонованого доступу: забезпечення парольного входу в систему, реєстрацію користувачів, призначення й зміну паролів, вибір або створення програмно-технологічних засобів захисту даних, шифрування інформації з метою захисту даних від несанкціонованого використання, фіксацію спроб несанкціонованого доступу до інформації;

– захист даних від руйнувань (одним зі способів захисту від втрати даних є резервування, що використовується як при фізичному псуванні файлу, так і у випадку, якщо в БД внесені небажані необоротні зміни);

– забезпечення відновлення БД: розробку програмно-технологічних засобів відновлення БД, організацію ведення системних журналів;

– аналіз звертань користувачів до БД: збір статистики звертань користувачів до БД, її зберігання й аналіз (хто з користувачів, до якої інформації, як часто звертався, які виконував дії), час виконання запитів, аналіз причин безуспішних, у тому числі аварійних, звертань до БД;

– аналіз ефективності функціонування БНД і розвиток інформаційної системи: аналіз показників функціонування системи (час обробки, обсяг пам'яті, вартісні показники), реорганізацію й реструктуризацію баз даних, зміну складу баз даних, розвиток програмних і технічних засобів;

– роботу з користувачами: збір інформації про зміни в предметній області, про оцінку користувачами роботи БНД, визначення регламенту роботи користувачів із БНД, навчання й консультування користувачів;

– підготовку й підтримку системних програмних засобів: збір та аналіз інформації про СУБД, придбання програмних засобів, їхню установку, перевірку працездатності, підтримку системних бібліотек, розвиток програмних засобів;

– організаційно-методичну роботу: вибір або створення методики проектування БД, визначення цілей і напрямків розвитку системи, планування етапів розвитку БНД, розробка й випуск організаційно-методичних матеріалів тощо.

У процесі своєї діяльності адміністратори БНД взаємодіють з іншими категоріями користувачів банку даних, а також з «зовнішніми» фахівцями, що не є користувачами БНД. Насамперед, якщо банк даних створюється для інформаційного обслуговування якого-небудь підприємства або організації, необхідні контакти з адміністрацією цієї організації. Як вказувалося вище, впровадження БНД призводить до значних змін не тільки системи обробки

даних, але й всієї системи керування організацією. Природно, що такі великі проекти не можуть бути виконані без активної участі й підтримки керівників організації. Керівництво організації повинне бути ознайомлене з можливостями, надаваними БНД, проінформоване про їхні переваги й недоліки, а також проблеми, викликані створенням і функціонуванням БНД.

Тому, що база даних є динамічним інформаційним відображенням предметної області, бажано, щоб адміністратор БНД, у свою чергу, був вчасно поінформований про перспективи розвитку об'єкта, для якого створюється інформаційна система. Керівництвом організації й адміністратором БНД повинні бути узгоджені цілі, основні напрямки і строки створення БНД і його розвитку, черговість підключення користувачів.

Дуже тісний зв'язок адміністрації БНД на всіх етапах життєвого циклу БНД спостерігається з кінцевими користувачами. Ця взаємодія починається на початкових стадіях проектування системи, коли вивчаються потреби користувачів, уточнюються особливості предметної області. Цей зв'язок постійно підтримується як протягом процесу проектування, так і функціонування системи.

Слід зазначити, що останнім часом спостерігається активний перерозподіл функцій між кінцевими користувачами й адміністраторами банку даних. Це, насамперед, пов'язане з розвитком мовних і програмних засобів, орієнтованих на кінцевих користувачів. Сюди відносяться прості й одночасно потужні мови запитів, а також засоби автоматизації проектування. Якщо банк даних функціонує у складі якої-небудь інформаційної системи, адміністратори БНД повинні працювати в контакті з фахівцями з обробки даних у цій системі.

Адміністратори БНД взаємодіють і з зовнішніми стосовно них групами фахівців, насамперед, постачальниками СУБД і ППП, адміністраторами інших БНД. Банки даних часто створюються спеціалізованими проектними колективами на основі договору на розробку інформаційної системи в цілому або БНД як самостійного об'єкта проектування. У цьому випадку служба ад-

міністрації БНД повинна створюватися як в організації-розробнику, так і в організації-замовнику.

На ефективність роботи БНД впливають безліч зовнішніх і внутрішніх факторів. Зростання складності й масштабів БНД, висока ціна неправильних або запізнілих рішень з адміністрування БД, високі вимоги до кваліфікації фахівців роблять актуальним завдання використання розвинених засобів автоматизованого (або навіть автоматичного) адміністрування БНД.

Питання й тести для самоконтролю

1. Порівняйте переваги й недоліки файлової організації даних і організації у вигляді баз даних.
2. Чим розрізняються поняття «база даних» і «банк даних»?
3. До якої з категорій користувачів БНД Ви б віднесли себе на даний момент і чому?
4. Чи існують у банків даних недоліки і як, по Вашому, їх можна усунути?
5. Яку роль у банку даних грає база даних?
6. Навіщо необхідне програмне забезпечення БНД?
7. Що Ви знаєте про мови, використовувані у банках даних?
8. Які ЕОМ використовуються в наш час для експлуатації банків даних?
9. Чому, на Ваш погляд, в інформаційних системах необхідно застосовувати мережні технології?
10. Які функції адміністраторів банку даних?

Розділ 3. КЛАСИФІКАЦІЯ ІНФОРМАЦІЙНИХ СИСТЕМ

З попереднього розділу стає зрозумілим, що інформаційні системи (бази даних, а тим більше, банки даних) є складними системами зі своєю внутрішньою структурою, частково прихованою від кінцевого користувача, частково – від адміністратора. Ступінь «приватності» інформації визначається рівнем, на якому ця інформація розглядається.

При створенні або при класифікації інформаційних систем неминуче виникають проблеми, пов'язані з формальним – математичним й алгоритмічним описом розв'язуваних задач. Від ступеня формалізації багато в чому залежать ефективність роботи всієї системи, а також рівень автоматизації, обумовлений ступенем участі людини при ухваленні рішення на основі одержуваної інформації.

Чим точніше математичний опис завдання, тим вище можливості комп'ютерної обробки даних і тим менше ступінь участі людини в процесі його вирішення. Це й визначає ступінь автоматизації завдання.

3.1. Загальні принципи класифікації систем обробки інформації

Розрізняють три типи задач, для яких створюються інформаційні системи:

– структуровані (такі, що формалізуються);

– неструктуровані (такі, що не формалізуються);

– частково структуровані.

Структурована (така, що формалізуються) задача – це задача, для якої відомі всі її елементи та взаємозв'язки між ними.

Неструктурована (така, що не формалізуються) задача – для якої неможливо виділити елементи та установити між ними зв'язки.

У структурованій задачі вдається виразити її зміст у формі математичної моделі, що має точний алгоритм рішення. Подібні задачі звичайно дово-

диться вирішувати багаторазово, і вони носять рутинний характер. Метою використання інформаційної системи для рішення структурованих задач є повна автоматизація їхнього рішення, тобто зведення ролі людини до нуля.

На практиці більшість задач є частково структурованими. Саме для рішення таких задач найчастіше доводиться створювати й використовувати інформаційні системи. Тому зупинимося на даному питанні більш докладно.

3.1.1. Типи інформаційних систем для розв'язання частково структурованих задач

Інформаційні системи, використовувані для вирішення частково структурованих задач, підрозділяються на два види:

– ІС, що створюють управлінські звіти й орієнтовані головним чином на обробку даних (пошук, сортування, фільтрацію і т.п.). Використовуючи відомості, що вміщено в цих звітах, управлінський персонал сам приймає рішення;

– ІС, що розробляють можливі альтернативи рішення. Ухвалення рішення людиною при цьому зводиться до вибору однієї з запропонованих системою альтернатив.

Інформаційні системи, що створюють управлінські звіти, забезпечують інформаційну підтримку користувача, тобто надають доступ до інформації в базі даних та її часткову обробку. Процедури маніпулювання даними в інформаційній системі повинні забезпечувати такі можливості:

– складання комбінацій даних, одержуваних з різних джерел;

– швидке додавання або виключення того або іншого джерела даних й автоматичне перемикання джерел при пошуку даних;

– керування даними з використанням можливостей систем керування базами даних;

– автоматичне відстеження потоку інформації для наповнення баз даних.

Інформаційні системи, що розробляють альтернативи рішень, можуть бути модельними й експертними.

Модельні інформаційні системи надають користувачеві математичні, статистичні, фінансові й інші моделі, використання яких полегшує виробіток й оцінку альтернатив рішення. Користувач може одержати відсутню йому для ухвалення рішення інформацію шляхом установлення діалогу з моделлю в процесі її дослідження.

Основними функціями модельної інформаційної системи є:

– можливість роботи в середовищі типових математичних моделей, включаючи вирішення основних завдань моделювання типу «як зробити, щоб?», «що буде, якщо?», аналіз чутливості та інше;

– досить швидка й адекватна інтерпретація результатів моделювання;

– своєчасна підготовка й коректування вхідних параметрів й обмежень моделі;

– можливість графічного відображення динаміки моделі;

– при необхідності – пояснення користувачеві можливих кроків формування й роботи моделі.

Експертні інформаційні системи забезпечують вироблення й оцінку можливих альтернатив користувачем за рахунок створення експертних систем, пов'язаних з обробкою знань. Експертна підтримка прийнятих користувачем рішень реалізується на двох рівнях.

Робота першого рівня експертної підтримки виходить із концепції «типових управлінських рішень», відповідно до якої часто виникаючі в процесі керування проблемні ситуації можна звести до деяких однорідних класів управлінських рішень, тобто деякого типового набору альтернатив. Для реалізації експертної підтримки на цьому рівні створюється інформаційний фонд зберігання й аналізу типових альтернатив.

Якщо деяка проблемна ситуація не асоціюється з наявними класами типових альтернатив, у роботу повинен вступати другий рівень експертної підтримки управлінських рішень. Цей рівень генерує альтернативи на базі

наявних в інформаційному фонді даних, правил перетворення й процедур оцінки синтезованих альтернатив.

3.1.2. Альтернативні класифікації інформаційних систем

Поряд з вищенаведеною класифікацією інформаційних систем по ступеню структурованості розв'язуваних завдань, можливі інші критерії для класифікації ІС. Розглянемо ті з них, які найбільш часто застосовуються на практиці.

Класифікація по ступеню автоматизації. Залежно від ступеня автоматизації інформаційних процесів у системі керування установою, інформаційні системи підрозділяються на ручні, автоматичні й автоматизовані.

Ручні ІС характеризуються відсутністю сучасних технічних засобів переробки інформації й виконанням всіх дій людиною. Наприклад, про діяльність менеджера у фірмі, де відсутні комп'ютери, можна говорити, що він працює з ручною ІС.

Автоматичні ІС виконують всі операції по переробці інформації без участі людини.

Автоматизовані ІС допускають участь у процесі обробки інформації як людини, так і технічних засобів, причому головна роль приділяється комп'ютеру. У сучасному тлумаченні в термін «інформаційна система» вкладається обов'язково поняття системи, що автоматизується.

Внаслідок широкого використання автоматизованих ІС в організації процесів керування, вони мають різні модифікації і можуть бути далі класифіковані, наприклад, по характеру використання інформації і по сфері застосування.

Класифікація по характеру використання інформації. З цього погляду ІС діляться на інформаційно-пошукові й інформаційно-вирішальні.

Інформаційно-пошукові системи роблять введення, систематизацію, зберігання, видачу інформації за запитом користувача без складних перетво-

рень даних. Наприклад, інформаційно-пошукова системи в бібліотеках, касах продажу залізничних й авіаквитків.

Інформаційно-вирішальні системи здійснюють всі процеси переробки інформації за певним алгоритмом. Серед них можна провести класифікацію по ступеню впливу виробленої результатної інформації на процес прийняття рішень і виділити два класи: керуючі й дорадчі.

Керуючі ІС виробляють інформацію, на підставі якої людина приймає рішення. Для цих систем характерні типи завдань розрахункового характеру й обробка більших обсягів даних. Прикладом можуть служити система короткострокового планування випуску продукції або система бухгалтерського обліку.

Дорадчі ІС виробляють інформацію, що приймається людиною до відома і не перетворюється негайно в серію конкретних дій. Ці системи мають більш високий ступінь інтелекту, тому що для них характерна обробка знань, а не даних.

Класифікація по сфері застосування. Узагальнюючи всю безліч областей практичного використання інформаційних систем, можна виділити такі основні сфери.

ІС організаційного керування призначені для автоматизації функцій управлінського персоналу. З огляду на найбільш широке застосування й розмаїтість цього класу систем, часто будь-які інформаційні системи розуміють саме в даному сенсі. До цього класу відносяться інформаційні системи керування як промисловими фірмами, так і непромисловими об'єктами: готелями, банками, торговельними фірмами та ін.

Основними функціями подібних систем є: контроль і регулювання, облік та аналіз, короткострокове й перспективне планування, бухгалтерський облік, керування збутом і постачанням та інші економічні й організаційні завдання.

ІС керування технологічними процесами слугують для автоматизації функцій виробничого персоналу. Вони широко використовуються при орга-

нізації підтримки технологічного процесу в металургійній і машинобудівній промисловості.

ІС автоматизованого проектування призначені для автоматизації функцій проектувальників, інженерів-проектувальників, конструкторів, архітекторів, дизайнерів при створенні нової техніки або технології. Основними функціями подібних систем є: інженерні розрахунки, створення графічної документації (креслень, схем, планів), створення проектної документації, моделювання проектованих об'єктів.

Інтегровані (корпоративні) ІС використовуються для автоматизації всіх функцій підприємства й охоплюють весь цикл робіт від проектування до збуту продукції. Створення таких систем досить складне, оскільки вимагає системного підходу з позицій головної мети, наприклад одержання прибутку, завоювання ринку збуту і т.д. Такий підхід може призвести до істотних змін у самій структурі організації, на що може зважитися не кожен керівник.

3.2. Класифікація баз та банків даних

Як випливає з попереднього розділу, банки даних – це складні системи, і їхня класифікація може бути зроблена як для всього банку даних у цілому, так і для кожного його компонента окремо. Класифікація для кожного з компонентів БНД може бути проведена по безлічі різних ознак (рис. 3.1).

3.2.1. Класифікація баз даних

Центральним компонентом банку даних є база даних, і більшість класифікаційних ознак відносяться саме до неї. За формою подання інформації розрізняють візуальні й аудіосистеми, а також системи мультимедіа. Ця класифікація показує, у якому вигляді інформація зберігається в БД і видається з баз даних користувачам: у вигляді зображення, звуку або є можливість використання різних форм відображення інформації. Поняття «зображення» тут

використовується в широкому сенсі: це може бути символьний текст, нерухоме графічне зображення (рисунки, креслення і т.п.), фотографії, географічні карти, рухомі зображення. Класифікація способів подання інформації являє собою самостійну проблему і тут не розглядається.

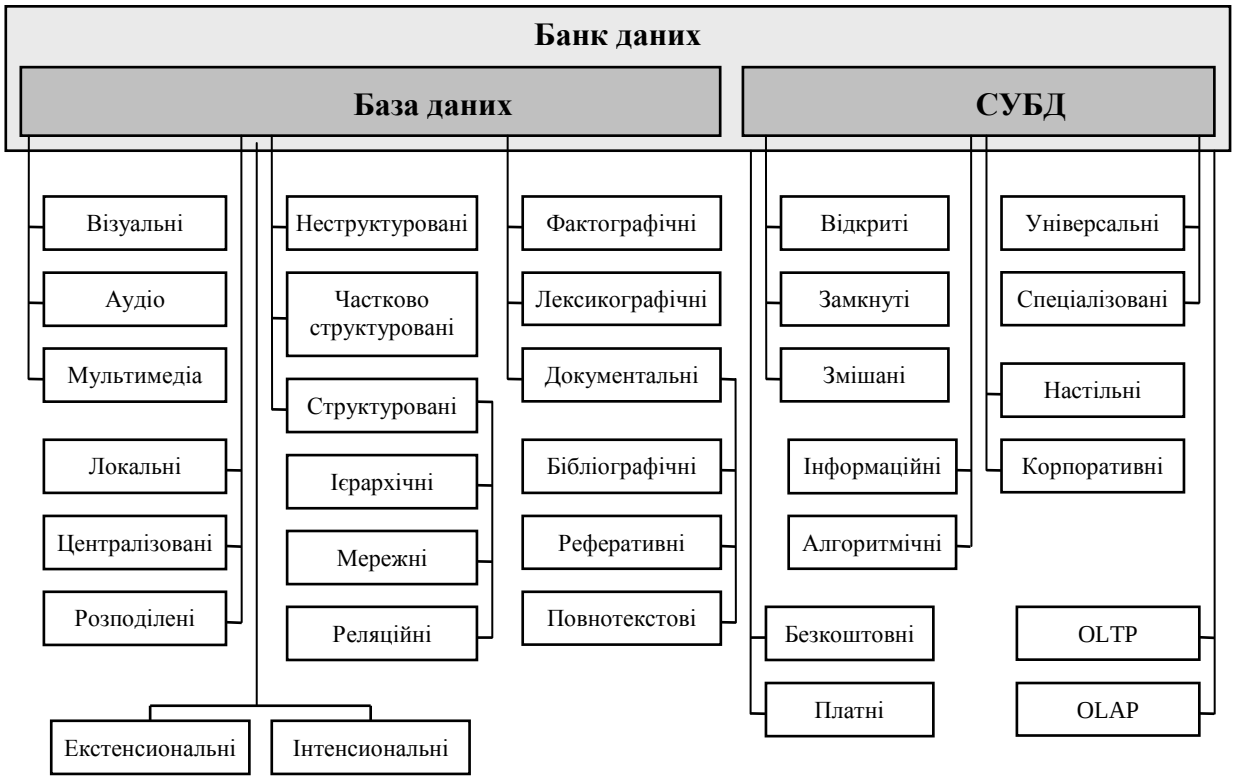


Рисунок 3.1 – Приклад класифікації банків даних

По *характеру організації даних* БД можуть бути розділені на неструктуровані, частково структуровані й структуровані. Ця класифікаційна ознака відноситься до інформації, поданої в символьному вигляді. До неструктурованих БД можуть бути віднесені бази, організовані у вигляді семантичних мереж. Частково структурованими можна вважати бази даних у вигляді звичайного тексту або гіпертекстові системи. Структуровані БД вимагають попереднього проектування й опису структури БД. Тільки після цього бази даних такого типу можуть бути заповнені даними.

Структуровані БД у свою чергу по *типу використовуваної* ***моделі*** діляться на ієрархічні, мережні, реляційні, змішані й мультимодельні. Класифікація по типу моделі поширюється не тільки на бази даних, але й на СУБД.

У структурованих БД звичайно розрізняють кілька рівнів інформаційних одиниць, що входять одна в іншу. Число цих рівнів може бути різним навіть для систем, що відносяться до того самого класу. Більшість структурованих систем підтримують рівень поля, запису й файлу. Ці інформаційні одиниці можуть називатися в різних системах по різному, але суть залишається однією й тією ж, а саме: полю відповідає найменша семантична одиниця інформації; сукупність полів або інших, більш складних інформаційних одиниць, якщо вони припустимі в конкретній СУБД, утворюють запис, а кілька однотипних записів являють собою файл бази даних. Останнім часом більшість СУБД у явному вигляді підтримують і рівень бази даних, як сукупності взаємозалежних файлів БД.

В **ієрархічній** (рис. 3.2) і **мережній** (рис. 3.3) моделях між інформаційними одиницями (ІО), тобто записами різних файлів, можуть задаватися зв'язки. Як видно з наведених схем, графічне подання ієрархічної моделі являє собою граф типу «дерево». У такій моделі є одна вершина – корінь дерева, що є входом у структуру. Кожна вершина, відмінна від кореня, може мати тільки одну вихідну вершину та, у загальному випадку, скільки завгодно породжених вершин.

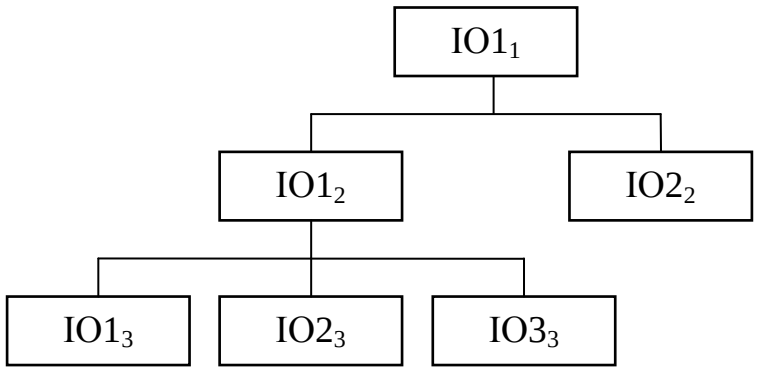


Рисунок 3.2 – Ієрархічна модель бази даних

Графічне зображення мережної моделі – це граф типу «мережа». Входом у таку структуру може бути будь-яка вершина. Кожна вершина може мати як кілька породжених, так і декілька вихідних вершин. Між парою вершин може бути задано кілька зв'язків (Зв). Переважна більшість СУБД підтримує прості мережні структури. Напрямок і характер зв'язку в мережних моделях не є очевидними, як у випадку ієрархічної моделі, тому при зображенні структури БД напрямок зв'язків повинен бути зазначений.

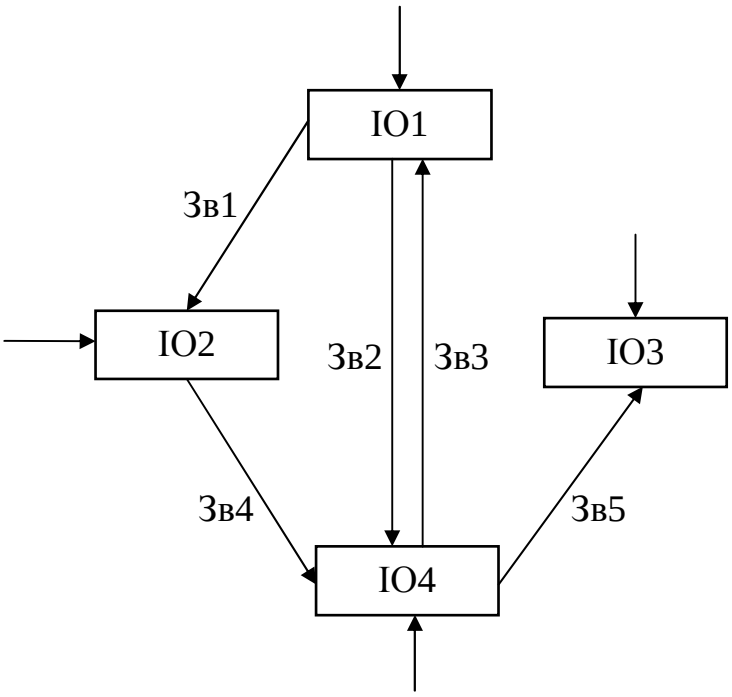


Рисунок 3.3 – Мережна модель бази даних

Крім мережних моделей з рівноправними файлами існують мережні моделі **з різнотипними файлами**. У таких моделях розрізняють головні (Гол) і залежні (Зал) файли (рис. 3.4). Вхід у структуру можливий тільки через головні файли. Зв'язуватися між собою можуть тільки записи різних типів.

Зв'язки в ієрархічних і мережних моделях описуються при проектуванні БД. Найчастіше ці зв'язки при зберіганні даних у БД передаються за допомогою адресних покажчиків. Ієрархічні й мережні моделі БД не накладають

обмеженнь на тип внутрішньозаписної структури. У принципі, вона може бути як простою лінійною (тобто складатися тільки з простих полів, що розташовані у записі послідовно один за одним), так і складною ієрархічною, що включає в себе різні складні одиниці інформації (вектори, що повторюються, групи і т.п.). Конкретні ж СУБД накладають обмеження на припустимі в них інформаційні одиниці, характер зв'язків між ними, порядок їхнього розташування в записі, а також часто мають і різні кількісні обмеження.

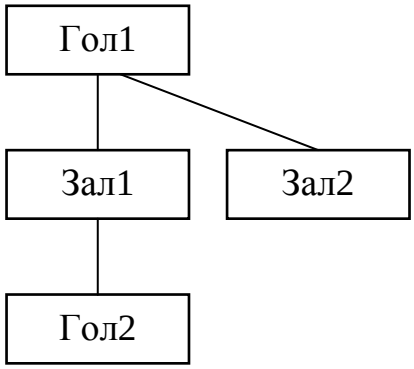


Рисунок 3.4 – Мережна модель бази даних з різнотипними файлами

Особливе місце серед структурованих систем займають системи, побудовані на використанні **інвертованих файлів**. Особливість організації даних у них полягає в тому, що власне збережені дані й інформація про зв'язки між ними логічно й фізично відділені одне від одного. Основні дані в цих системах зберігаються у файлах, записи яких можуть мати складну структуру. Вся керуюча інформація зосереджена в так званому асоціаторі. Логічний зв'язок між файлами встановлюється за допомогою компонента асоціатора, що називається мережею зв'язку. Реально, зв'язки встановлюються не безпосередньо з елементами зв'язку, а через перетворювач адреси. Відділення асоціативної інформації від власне збережених даних дозволяє змінювати зв'язки, не змінюючи при цьому самих файлів.

У **реляційних** моделях (на відміну від ієрархічних і мережних) зв'язки між файлами БД визначаються динамічно в момент виконання запиту. Ці

зв'язки визначаються по рівності значень відповідних полів (полів зв'язку), що знаходяться в кожному зі зв'язаних файлів. Іншою відмінною рисою реляційних моделей є обмеження на внутрішньозаписну структуру: записи мають лінійну структуру і можуть містити лише адресні (А) або прості (П) поля (рис. 3.5). Ці відмінні риси відіграють вирішальну роль при проектуванні структури БД.

Рисунок 3.5 – Реляційна модель бази даних

Вісімдесяті роки минулого століття були часом інтенсивного розвитку реляційних систем. В 1992 році (перша комерційна реляційна СУБД була випущена фірмою Oracle в 1979 році) рівень продажів реляційних СУБД уперше перевищив рівень продажів нереляційних СУБД. Однак до 90 % даних на промислових підприємствах зберігалися до цього моменту в нереляційних базах даних на потужних ЕОМ.

По *типу збереженої інформації* БД діляться на документальні, фактографічні й лексикографічні. Серед документальних баз розрізняють бібліографічні, реферативні й повнотекстові. До лексикографічних баз даних відносяться різні словники (класифікатори, багатомовні словники, словники основ слів і т.п.). У системах фактографічного типу в БД зберігається інформація про об'єкти, що цікавлять користувача предметної області у вигляді «фактів» (наприклад, біографічні дані про співробітників, дані про випуск продукції виробниками і т.п.); у відповідь на запит користувача видається необхідна інформація про об'єкт, що його цікавить, або повідомлення про те, що необхідна інформація відсутня в БД.

У документальних БД одиницею зберігання є який-небудь документ (наприклад, текст закону або статті) і користувачеві у відповідь на його запит видається або посилання на документ, або сам документ, у якому він може знайти інформацію, що його цікавить. БД документального типу можуть бути організовані по-різному: без зберігання й зі зберіганням самого вихідного документа на машинних носіях. До систем першого типу можна віднести бібліографічні та реферативні БД, а також БД-покажчики, що «відсилають» до джерела інформації. Системи, у яких передбачене зберігання повного тексту документа, називаються повнотекстовими.

У системах документального типу метою пошуку може бути не тільки якась інформація, що зберігається в документах, але й самі документи. Так, можливі запити типу «скільки документів було створено за певний період часу» і т.п. Часто в критерій пошуку як ознаки включаються «дата прийняття документа», «ким прийнятий» й інші «вихідні дані» документів.

Специфічним різновидом баз даних є бази даних форм документів. Вони мають деякі риси документальних систем (шукається документ, а не інформація про конкретний об'єкт, форма документа має назву, по якому звичайно й здійснюється її пошук), так і специфічні особливості (документ шукається не з метою одержання з нього інформації, а з метою використання його в якості «шаблона»).

В останні роки активно розвивається об’єктно-орієнтований підхід до створення інформаційних систем. Об'єктні бази даних організовані як об'єкти й посилання до об'єктів. Об'єкт – це дані й правила, які встановлюють алгоритми роботи із цими даними. Об'єкт включає метод, що є частиною визначення об'єкта й запам'ятовується разом з об'єктом. В об'єктних базах даних дані запам'ятовуються як об'єкти, класифіковані по типах класів й організовані в ієрархічне сімейство класів. Клас – колекція об'єктів з однаковими властивостями. Об'єкти належать класу. Класи організовані в ієрархії.

За *характером організації зберігання даних і звертання до них* розрізняють локальні (персональні), загальні (інтегровані, централізовані) і розпо-

ділені бази даних. Персональна база даних – це база даних, призначена для локального використання одним користувачем. Локальні БД можуть створюватися кожним користувачем самостійно, або бути одержані з загальної БД.

Інтегровані і розподілені БД припускають можливість одночасного звернення декількох користувачів до однієї й тієї ж інформації (багатокористувацький, паралельний режим доступу). Це привносить специфічні проблеми при їхньому проектуванні й у процесі експлуатації БНД. Розподілені БД крім цього мають характерні риси, пов'язані з тим, що фізично різні частини БД можуть бути розташовані на різних ЕОМ, а логічно, з погляду користувача вони повинні являти собою єдине ціле.

Технології, які на перший погляд начебто б перебувають на різних кінцях спектра (локальна й розподілена обробка), насправді дуже близькі й розрізняються практично тим, як підтримується зв'язок між окремими частинами БД. У випадку локальних систем підтримка цього зв'язку не є централізованою, а у випадку розподілених БНД – повинна організовуватися СУБД. Технологією, що дозволяє поєднати ідеї локальної роботи й централізованої підтримки єдиної БД, є технологія тиражування, при якій засоби СУБД дозволяють тиражувати окремі частини загальної БД, локально використовувати їх, а потім «узгоджувати» окремі фрагменти БД у рамках єдиної бази даних.

Концепції централізованої й розподіленої обробки даних також не так суттєво розрізняються між собою, як здається на перший погляд. Так звані клієнт-серверні системи з «тонким клієнтом» дуже близькі до централізованих баз даних. Банк даних є складною людино-машинною системою, і розподілятися по вузлах мережі можуть не тільки БД, але й інші компоненти БНД. Сама БД при цьому може бути й не розподіленою (наприклад, при забезпеченні багатокористувацького доступу до централізованого БНД у мережі). Тому розрізняються два поняття: розподілені БД і розподілені БНД. При цьому під розподіленим БНД розуміється банк даних, у якому розподілена хоча б одна з його компонентів.

БД класифікуються за *обсягом*. Особливе місце тут займають так звані дуже великі бази даних. Це викликано тим, що для великих баз даних по іншому оцінюються питання забезпечення ефективності зберігання інформації й забезпечення її обробки.

3.2.2. Класифікація систем управління базами даних

Розглянемо тепер ряд класифікаційних ознак, що відносяться до СУБД. За *мовами спілкування* СУБД діляться на відкриті, замкнуті й змішані. Відкриті системи – це системи, у яких для звертання до баз даних використовуються універсальні мови програмування. Замкнуті системи мають власні мови спілкування з користувачами БНД. По числу рівнів в архітектурі розрізняють однорівневі, дворівневі, трьохрівневі системи. В принципі, можливе виділення й більшого числа рівнів. Під архітектурним рівнем СУБД розуміють функціональний компонент, механізми якого слугують для підтримки деякого рівня абстракції даних (логічний і фізичний рівень, а також «погляд» користувача – зовнішній рівень).

За *виконуваними функціями* СУБД діляться на інформаційні та алгоритмічні. Інформаційні СУБД дозволяють організувати зберігання інформації й доступ до неї. Для виконання більш складної обробки необхідно писати спеціальні програми. Алгоритмічні СУБД виконують досить складну обробку, наприклад, автоматично дозволяють одержувати агреговані показники, що не зберігаються безпосередньо в базі даних, можуть змінювати алгоритми обробки і т.д.

По *сфері можливого застосування* розрізняють універсальні й спеціалізовані, звичайно проблемно-орієнтовані СУБД. Системи керування базами даних підтримують різні типи даних. Набір типів даних, припустимих у різних СУБД, різний. Крім того, ряд СУБД дозволяє розроблювачеві додавати нові типи даних і нові дії над цими даними. Такі системи називаються розширюваними системами баз даних (РСБД). Подальшим розвитком концепції

РСБД є системи об’єктно-орієнтованих баз даних, що володіють досить потужними виразними можливостями, щоб безпосередньо моделювати складні об'єкти.

По *«потужності»* СУБД діляться на «настільні» і «корпоративні». Характерними рисами настільних СУБД є порівняно невисокі вимоги до технічних засобів, орієнтація на кінцевого користувача, низька вартість. Корпоративні СУБД забезпечують роботу в розподіленому середовищі, високу продуктивність, підтримку колективної роботи при проектуванні систем, мають розвинені засоби адміністрування і більш широкі можливості підтримки цілісності. У зв'язку з перерахованими вище рисами корпоративних СУБД очевидно, що ці системи складні, дорогі, вимагають значних обчислювальних ресурсів.

Системи обох класів інтенсивно розвиваються, причому деякі тенденції розвитку властиві кожному із цих класів. Насамперед, це використання високорівневих засобів розробки додатків (що раніше було властиве, в основному, настільним системам), ріст продуктивності й функціональних можливостей, робота в локальних і глобальних мережах та ін. Найбільш відомими з корпоративних СУБД є Oracle, Informix, Sybase, Microsoft SQL Server, Progress і деякі інші. Спостерігається зв'язок між класом СУБД і використовуваною операційною системою. Системи під UNIX позиціонуються як корпоративні розподілені системи. Зараз у цей сектор «пробивається» система Microsoft Windows 7.

По *орієнтації на переважну категорію користувачів* можна виділити СУБД для розроблювачів і для кінцевих користувачів. Системи, що відносяться до першого класу, повинні мати якісні компілятори й дозволяти створювати «відчужувані» програмні продукти, мати розвинені засоби налагодження, включати засоби документування проекту і мати інші можливості, що дозволяють створювати ефективні складні системи. Основними вимогами, до систем, орієнтованих на кінцевого користувача, є: зручність інтерфей-

су, високий рівень мовних засобів, наявність інтелектуальних модулів підказок, підвищений захист від ненавмисних помилок і т.п.

3.2.3. Класифікація банків даних

За *умовами надання послуг* розрізняють безкоштовні та платні банки даних. Платні БНД, у свою чергу, діляться на безприбуткові й комерційні. Безприбуткові банки даних функціонують на принципі самоокупності і не ставлять своєю метою одержання прибутку. Це звичайно БНД соціально значимої інформації, що має широке коло користувачів, або наукової, бібліотечної інформації. Основною метою створення комерційних банків даних є одержання прибутку від інформаційної діяльності.

Банки даних розрізняються за *характером переважної обробки інформації*. В одних в основному реалізується велика кількість достатньо простих запитів [такі системи одержали назву OLTP (On-Line Transaction Processing) – системи онлайнової обробки транзакцій]. В інших, навпроти, потрібна складна аналітична обробка даних [для такого класу систем став використовуватися термін OLAP (On-line Analytical Processing)].

Термін OLAP є порівняно новим і у різних літературних джерелах трактується іноді по-різному. Цей термін часто ототожнюють з підтримкою прийняття рішень [DSS (Decision Support Systems) – системи підтримки ухвалення рішення]. Як синонім для останнього терміна використовують Data Warehousing – сховища (склади) даних, розуміючи під цим набір організаційних рішень, програмних і апаратних засобів для забезпечення аналітиків інформацією на основі даних із систем обробки транзакцій нижнього рівня та інших джерел.

«Склади даних» дозволяють обробляти дані, накопичені за тривалі періоди часу. Ці дані є різнорідними (і не обов'язково структурованими). Для «складів даних» властивий багатовимірний характер запитів. Величезні обся-

ги даних, складність структури як даних, так і запитів вимагає використання спеціальних методів доступу до інформації.

В інших джерелах поняття системи підтримки прийняття рішень (СППР) вважається більш широким. Сховища даних і засоби оперативної аналітичної обробки можуть бути одними з можливих компонентів архітектури СППР.

Іноді розрізняють «OLAP у вузькому сенсі» – системи, які забезпечують тільки вибірку даних у різних розрізах, і «OLAP у широкому сенсі», або просто OLAP, що включають в себе:

– підтримку декількох користувачів, що редагують БД;

– функції моделювання, у тому числі обчислювальні механізми одержання похідних результатів, а також агрегування й об'єднання даних;

– прогнозування, виявлення тенденцій і статистичний аналіз.

Природно, що кожний з цих типів БНД вимагає специфічної організації даних, а також спеціальних програмних засобів, що забезпечують ефективне виконання поставлених завдань.

Для забезпечення швидкої обробки даних при їхньому аналізі використовуються різноманітні прийоми. Одним з них є організація даних у вигляді так званих багатовимірних БД (ББД). Інформація в ББД зберігається не у вигляді індексованих записів у таблицях, а у формі логічно впорядкованих масивів. Єдиної загальновизнаної багатовимірної моделі зберігання даних не існує. У ББД відсутній стандартизований метод доступу до даних, і вони можуть відповідати вимогам специфічної аналітичної обробки даних.

Сховища даних можуть бути розбиті на два типи: корпоративні сховища даних (enterprise data warehouses) і кіоски даних (data marts).

Корпоративні сховища даних містять інформацію, що відноситься до всієї корпорації і зібрана з безлічі оперативних джерел для консолідованого аналізу. Звичайно такі сховища охоплюють цілий ряд аспектів діяльності корпорації і використовуються для прийняття як тактичних, так і стратегічних рішень.

Кіоски даних містять підмножини корпоративних даних і будуються для відділів або підрозділів всередині організації. Кіоски даних часто будуються силами самого відділу й охоплюють конкретний аспект, що цікавить співробітників даного відділу. Кіоск даних може одержувати дані з корпоративного сховища (залежний кіоск), або, що більше поширено, дані можуть надходити безпосередньо з оперативних джерел (незалежний кіоск). Кіоски й сховища даних будуються за подібними принципами і використовують практично ті самі технології.

По *ступеню доступності* БНД діляться на загальнодоступні й з обмеженим колом користувачів.

По *охопленню* БД можуть класифікуватися, у свою чергу, у різних «розрізах»:

– всесвітній;

– державний;

– міський;

– відомчий;

– проблемний (тематичний).

Відомча й проблемна ознаки класифікації можуть відноситися не тільки до інформації, що зберігається в БД, але й до кола користувачів, що обслуговуються нею.

По *характеру взаємодії з користувачами* (хто ініціює дії) БНД бувають:

– активні;

– пасивні.

У пасивних БНД провідна роль належить користувачеві. В активних система може самостійно змінювати поводження. Останнім часом активні бази даних стали всі частіше використовуватися для різних цілей.

За *формою власності* БНД діляться на:

– державні;

– недержавні;

– приватні;

– групові;

– особисті.

У літературі зустрічаються й інші аспекти класифікації банків даних, але перераховані вище є найбільш значимими.

Питання й тести для самоконтролю

1. Навіщо необхідна класифікація інформаційних систем?
2. Опишіть загальні принципи класифікації систем обробки інформації.
3. Перелічіть типи інформаційних систем для вирішення частково структурованих задач.
4. У чому полягає різниця між модельними й експертними інформаційними системами?
5. Опишіть класифікацію інформаційних систем по ступеню їхньої автоматизації.
6. Як класифікуються інформаційні системи по характеру використання інформації?
7. Класифікуйте інформаційні системи по сфері їхнього застосування.
8. Які Ви знаєте критерії класифікації баз даних?
9. По яких ознаках можуть бути класифіковані СУБД?
10. Перелічить відомі Вам критерії класифікації банків даних.

Розділ 4. ЗБЕРІГАННЯ І ОБРОБКА ІНФОРМАЦІЇ В НАЙПРОСТІШИХ БАНКАХ ДАНИХ

Після ознайомлення з теоретичними основами розробки і класифікації баз і банків даних, перейдемо до питань практичної реалізації БНД за допомогою сучасних програмних засобів, спеціально розроблених, або призначених, в числі іншого, для цих цілей. Почнемо з найбільш простого варіанту СУБД, відомого як система Microsoft (MS) Excel.

MS Excel (також іноді звана Microsoft Office Excel) – програма для роботи з електронними таблицями, створена корпорацією Microsoft для операційних систем Windows і Mac OS. Вона надає можливості економіко-статистичних розрахунків, графічні інструменти і мову макропрограмування VBA (Visual Basic for Applications). MS Excel входить до складу Microsoft Office і на сьогоднішній день є одним з найбільш популярних застосувань в світі.

4.1. Загальний опис системи програмування Microsoft Excel

MS Excel – додаток, що дозволяє за допомогою електронних таблиць аналізувати дані і виконувати обчислення або працювати із списками. Він спрощує одержання і аналіз ділової інформації, що зберігається на персональному комп'ютері, в локальній мережі і на сайтах Інтернету.

Комп'ютери полегшили не тільки відображення даних, але і їх обробку. Програми, використовувані для цієї мети, отримали назву табличних процесорів або електронних таблиць. Електронна таблиця, як і таблиця, розграфлена на папері, розділена на стовпці і рядки, в осередки яких записана різного роду інформація: тексти, числа, формули і т.д.

В наш час на ринку існує велика кількість програм, що забезпечують зберігання і обробку табличних даних: MS Excel, Lotus 1-2-3, Quattro Pro та інші. Електронні таблиці розрізняються, в основному, набором виконуваних

функцій і зручністю інтерфейсу. Більше 90 % користувачів персональних комп'ютерів працюють з MS Excel.

MS Excel застосовується при вирішенні планово-економічних, фінансових, техніко-економічних і інженерних завдань, при виконанні бухгалтерського і банківського обліку, для статистичної обробки інформації, для аналізу даних і прогнозування проектів, при заповненні податкових декларацій.

4.1.1. Електронні таблиці

Електронні таблиці MS Excel дозволяють обробляти статистичну інформацію і візуалізовувати дані у вигляді графіків і діаграм. Їх можна використовувати і в повсякденному житті для обліку і аналізу витрат грошових коштів: при щоденній покупці продуктів і господарських товарів, при оплаті рахунків і т.д.

Двовимірні (прямокутні) таблиці широко використовуються для впорядкованого зберігання даних і наочного подання чисел або текстової інформації в багатьох галузях професійної діяльності. У таблиці може бути відображена як початкова (первинна) інформація, так і результати виконання арифметичних, логічних або інших дій над початковими даними. До появи комп'ютерів таблиці створювалися на папері у вигляді розграфлених аркушів або картотеки.

Електронна таблиця має вигляд прямокутної матриці, розділеної на стовпці та рядки. В ній можуть зберігатися різні дані: тексти, числа, дати, результати виконання арифметичних, логічних або інших дій над вихідною інформацією.

Одна з головних переваг електронних таблиць – автоматичний перерахунок даних по заздалегідь заданих формулах і оновлення діаграм при зміні значень, що зберігаються в осередках.

Можливість швидкого перерахунку даних під час надходження нової інформації, розрахунок декількох варіантів початкових даних, полегшує моделювання різних ситуацій і вибір оптимального варіанту.

Електронні таблиці можна легко вставити в документ, складений в текстовому процесорі MS Word або іншому додатку **MS Office**. Тісна інтеграція додатків, що входять в пакет, дозволяє підвищити ефективність роботи користувача, створювати професійно оформлені документи і використовувати можливості локальних мереж і глобальної інформаційної мережі Інтернет для колективної роботи і публікації даних.

4.1.2. Основи роботи з системою Microsoft Excel

Як і у інших компонент пакету Microsoft Office, інтерфейс програми дуже схожий з інтерфейсом текстового процесора MS Word. Після запуску програми вікно MS Excel містить п'ять областей (рис. 4.1):

– рядок меню;

– панелі інструментів;

– рядок формул;

– вікно книги;

– рядок стану.

Основною відмінністю системи MS Excel від системи MS Word є наявність замість вікна документа так званого вікна книги (електронної таблиці).

Книгу MS Excel розбито на кілька аркушів (**таблиць MS Excel**). Аркуші можна видаляти або додавати нові. Як і будь-яка таблиця, аркуш MS Excel складається з рядків та стовпців, перехрестя яких утворюють комірки.

У нижній частині вікна книги знаходиться декілька кнопок, за допомогою яких можна переходити від одного аркуша до іншого. Якщо видно не всі ярлички аркушів, то для проглядання змісту книги можна використовувати чотири кнопки, розташовані в нижньому лівому кутку вікна.

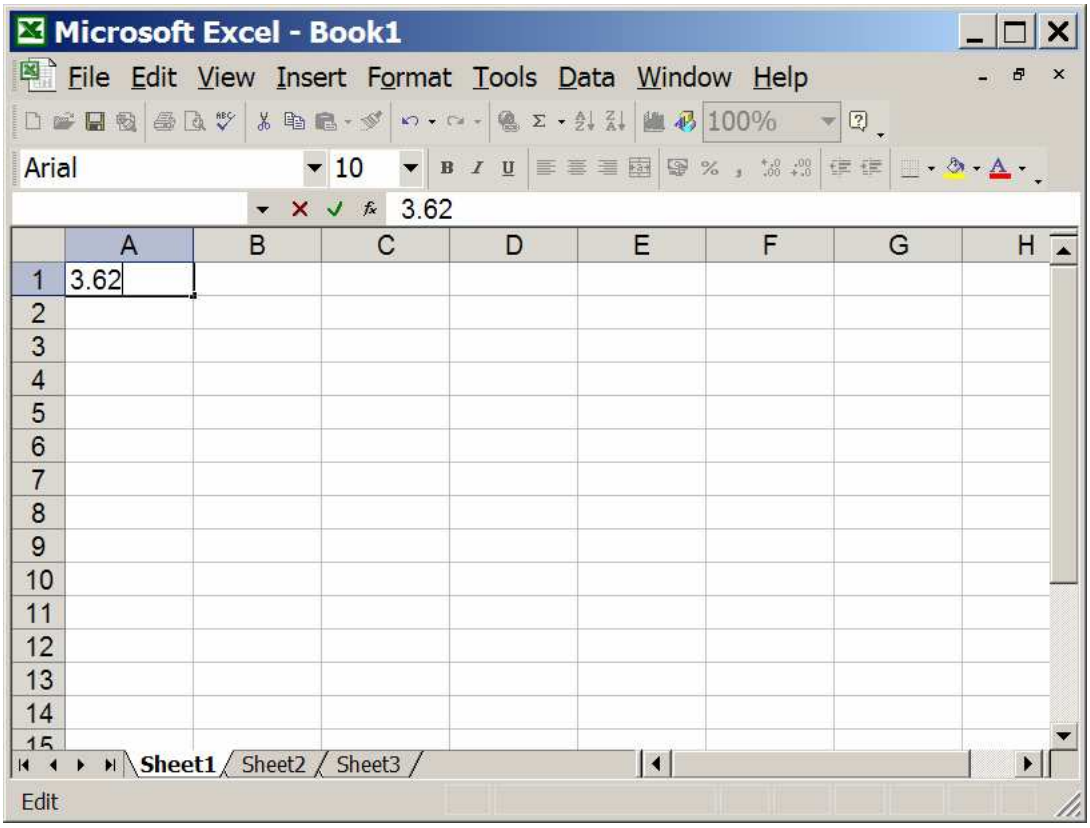


Рисунок 4.1 – Вікно Microsoft Excel

Комірки Excel є основними «будівельними одиницями» робочого аркуша. Кожна комірка має свої строго визначені координати, або адресу комірки, де можна зберігати та відображати інформацію. Наприклад, комірка, що знаходиться на перехресті стовпця А і рядка 1 має адресу А1 і т.д.

Часто межі даних таблиці виходять за межі екрану монітора. В цьому випадку для проглядання вмісту аркуша треба використовувати смуги прокрутки, розташовані уздовж правої і нижньої сторін вікна книги.

У верхній частині робочої області MS Excel розташований рядок меню. Нижче знаходяться панелі інструментів з кнопками, що виконують найбільш часто використовувані функції. Робота з цією частиною робочої області MS Excel аналогічна роботі в MS Word. Нижче за панелі інструментів розташований рядок формул.

Вміст активного (виділеною на даний момент) осередку MS Excel завжди з'являється в рядку формул. В процесі введення або редагування даних в осередку, в рядку формул з'являються три кнопки:

– кнопка відміни (червоний хрестик);

– кнопка введення (зелена пташка);

– кнопка зміни формули (букви «*fx*»).

Інформацію можна вводити як безпосередньо в осередок, так і в рядок формул.

У самому низу вікна робочої області знаходиться рядок стану, що показує режим роботи програми.

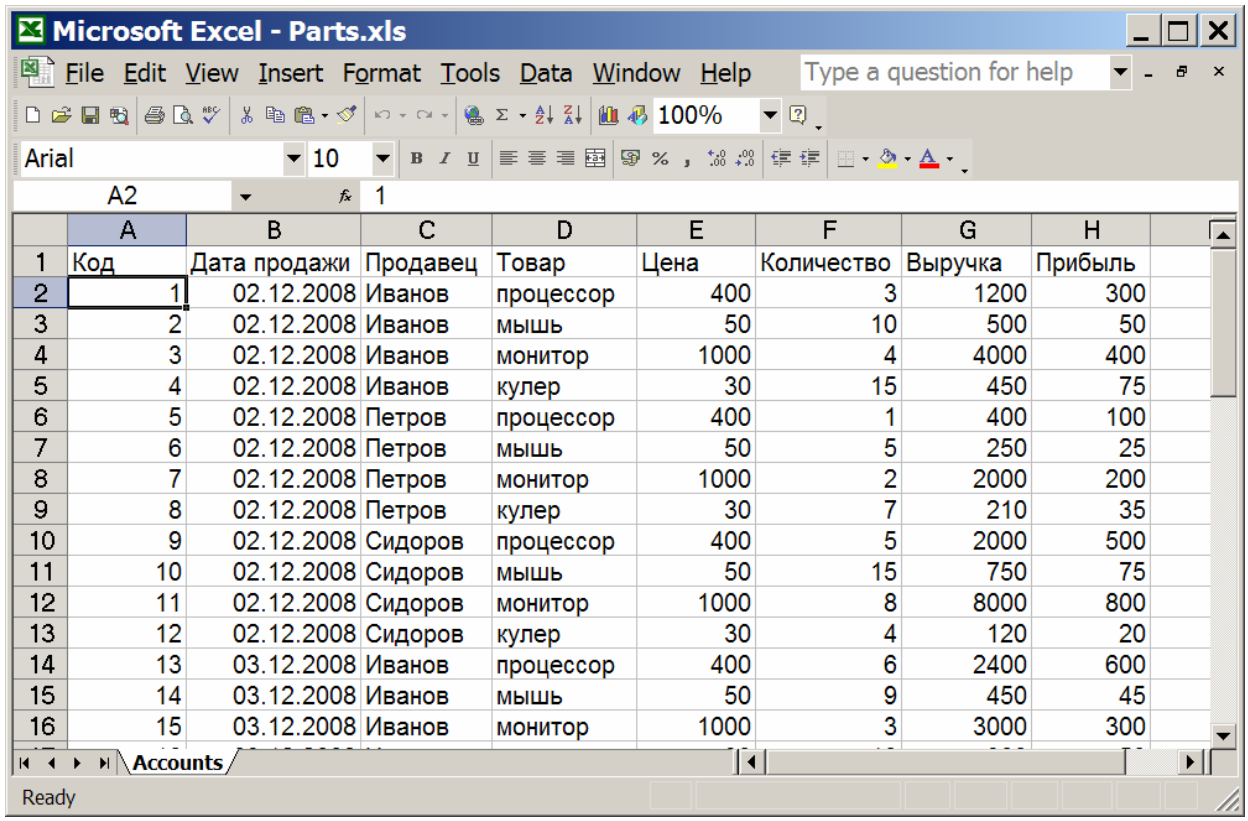
Обмежений об'єм навчального посібника не дозволяє тут детально описати прийоми роботи з системою MS Excel. Для детальнішого ознайомлення рекомендується використовувати спеціальну літературу по пакету Microsoft Office (наприклад [2]).

4.2. Створення банку даних на основі системи Microsoft Excel

Великі масиви інформації зазвичай зберігаються у вигляді баз даних і обробляються спеціальними програмами. У MS Excel аналогом простої бази даних є список. Його можна розглядати як групу рядків, що містять зв'язані дані. Відмінна особливість списку полягає в тому, що кожен його стовпець містить однотипні дані. Якщо провести аналогію між списком і базою даних, то можна сказати, що стовпці списку є полями бази даних, а його рядки – записами.

Класичним прикладом списку може служити список продажів (бухгалтерська книга) магазина, в якому вказані: дата продажу, прізвище продавця, найменування товару, ціна, кількість, виручка і прибуток (рис. 4.2).

Пропоновані MS Excel засоби для роботи зі списками значно полегшують виконання стандартних дій з сортування даних і пошуку потрібної інформації. Отримані при цьому результати можна використовувати в розрахунках. Наприклад, спочатку можна задати пошук певних значень в списку, а потім провести з цими значеннями обчислення або створити на їх основі діаграму.



	A	B	C	D	E	F	G	H
1	Код	Дата продажи	Продавец	Товар	Цена	Количество	Выручка	Прибыль
2	1	02.12.2008	Иванов	процессор	400	3	1200	300
3	2	02.12.2008	Иванов	мышь	50	10	500	50
4	3	02.12.2008	Иванов	монитор	1000	4	4000	400
5	4	02.12.2008	Иванов	кулер	30	15	450	75
6	5	02.12.2008	Петров	процессор	400	1	400	100
7	6	02.12.2008	Петров	мышь	50	5	250	25
8	7	02.12.2008	Петров	монитор	1000	2	2000	200
9	8	02.12.2008	Петров	кулер	30	7	210	35
10	9	02.12.2008	Сидоров	процессор	400	5	2000	500
11	10	02.12.2008	Сидоров	мышь	50	15	750	75
12	11	02.12.2008	Сидоров	монитор	1000	8	8000	800
13	12	02.12.2008	Сидоров	кулер	30	4	120	20
14	13	03.12.2008	Иванов	процессор	400	6	2400	600
15	14	03.12.2008	Иванов	мышь	50	9	450	45
16	15	03.12.2008	Иванов	монитор	1000	3	3000	300

Рисунок 4.2 – Список продажів

Таблиці-списки обробляються подібно до звичайних таблиць, тобто редагування даних в них здійснюється за допомогою загальновідомих команд. Існують і спеціальні засоби для роботи зі списками, які надають користувачам додаткові можливості. Так, над списком, що розглядається як приклад, можна провести наступні дії: включити в нього нові записи або видалити старі (при поверненні товару), знайти в ньому певну інформацію по заданих критеріях.

4.2.1. Створення і заповнення списків

Перш ніж приступати до створення списку, необхідно ретельно продумати його структуру і визначити, які дані повинні бути в нього включені. Починати потрібно з формування заголовків стовпців. Символи використовуваного в заголовках шрифту повинні бути іншого кольору або мати характерне зображення, наприклад напівжирне або курсивне. Якщо заголовки оформлені

по-іншому, програма автоматично визначає, що створюється список, і надає допомогу в його заповненні.

Порядок введення записів в список великої ролі не грає, тому що для їх впорядкування досить виконати декілька клацань мишею. Всі записи повинні бути оформлені однаково, але не так, як заголовки стовпців. При введенні даних не слід додавати порожні рядки, рядки, що складаються з символів дефіса або символів підкреслення, оскільки це може утруднити в подальшому сортування і фільтрацію даних.

MS Excel може розпізнавати списки автоматично. Якщо необхідно, припустимо, відсортувати записи списку по певному критерію, досить помістити покажчик осередку в осередок усередині списку, і після активізації функції сортування MS Excel автоматично виділить весь список.

При автоматичному виділенні області списку MS Excel порівнює вміст першою і другою його рядків. Якщо дані в цих рядках розрізняються за типом або по оформленню, програма сприймає перший рядок як заголовки стовпців і не обробляє його. Але якщо при виклику команди Form (Форма) або Subtotals (Підсумки) з меню Data (Дані) MS Excel не виявить відмінностей між записами першого і другого рядків, на екрані з'явиться повідомлення про неможливість знайти назви для створення заголовків стовпців.

У такому разі програма видасть запит про те, чи слід використовувати як заголовки стовпців дані першого рядка виділеного діапазону осередків. Можна створити заголовки стовпців, клацнувши на кнопці ОК, або не робити цього.

При формуванні списку, який містить значення, що повторюються, можна застосовувати функцію автозаповнення. Для її включення на вкладці Edit (Правка) діалогового вікна Options (Параметри) необхідно встановити прапорець Enable AutoComplete for cell values (Автозаповнення значень осередків).

Коли функція автозаповнення активна, при введенні тексту в елемент таблиці програма MS Excel перевіряє вміст інших осередків даного стовпця.

Якщо в них знаходиться текст, перші символи якого співпадають з символами, введеними користувачем, введення даних в осередок продовжується автоматично. Щоб підтвердити необхідність введення запропонованого програмою варіанту тексту слід натиснути клавішу Enter. Якщо цей варіант не підходить – треба проігнорувати пропозицію MS Excel і продовжити введення в осередок необхідних даних.

Проте іноді функція AutoComplete лише заважає введенню даних в таблицю, дратуючи користувача. У таких випадках її доцільно відключити, як вказано вище.

4.2.2. Сортування списків

Необхідність сортування записів в списках може виникнути при виконанні різноманітних завдань, наприклад, при пошуку в довгому прайс-листі дешевших товарів або в процесі угрупування записів про продажі, що відносяться до одного менеджера. Сортувати можна як числові, так і текстові дані. Текстові дані сортують в алфавітному порядку (від А до Я або від Я до А), а числові – в порядку убування, або в порядку зростання.

Можливе сортування як рядків, так і стовпців таблиці. На практиці найчастіше використовується сортування рядків, яке ми і розглянемо тут.

За умовчанням рядки виділеного списку сортуються по вмісту одного із стовпців, але існує можливість відсортувати список по вмісту двох і навіть трьох стовпців одночасно. При використанні функцій списку MS Excel виділить список автоматично. Проте користувач може перед викликом команди сортування заздалегідь самостійно виділити діапазон осередків, що підлягають сортуванню.

Критерії сортування встановлюються в діалоговому вікні Sort (Сортування діапазону), показаному на рис. 4.3, яке викликається пунктами меню Data – Sort (Дані – Сортування).

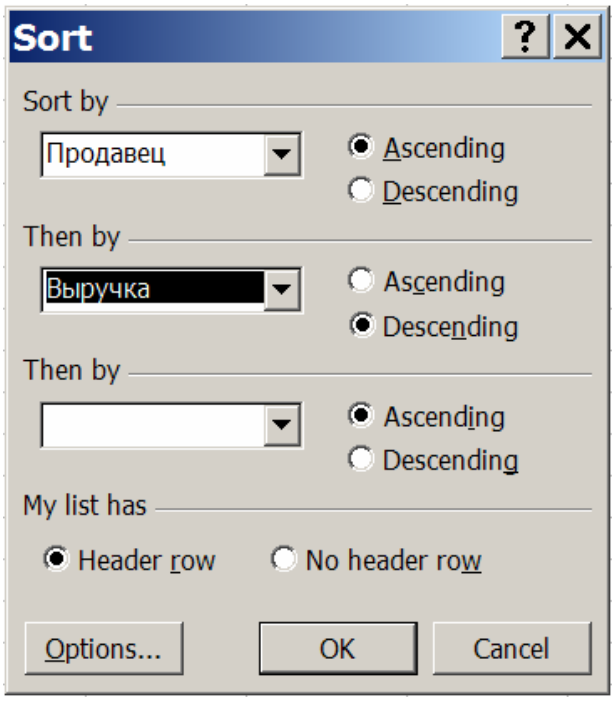


Рисунок 4.3 – Вікно сортування

Якщо перший рядок виділеного діапазону містить заголовки стовпців, то, для того, щоб виключити його з числа рядків, що підлягають сортуванню, слід встановити в області My list has (Мій список має) перемикач Header row (Рядок заголовка). Навпаки, перемикач No header row (Немає рядка заголовка) цієї області дозволяє виконати сортування з урахуванням першого запису виділеного діапазону осередків, тобто програмі дається вказівка, що перший рядок – це не заголовок, а дані.

Стовпець, вміст осередків якого підлягає сортуванню, вибирається в полі списку області Sort by (Сортувати по). Тут же слід задати порядок сортування – Ascending (Зростання) або Descending (Убування). При цьому MS Excel автоматично визначить тип даних – текст або числові значення. Якщо стовпець, дані якого сортуються, містить і текст, і числові значення, числові значення повинні передувати тексту.

На рис. 4.4 показаний список продажів, відсортований по прізвищах продавців (в алфавітному порядку) і виручці (в убуваючому порядку, як було задано на рис. 4.3). Заголовки при сортуванні стовпців не враховувалися.

Parts.xls

	A	B	C	D	E	F	G	H
1	Код	Дата продажи	Продавец	Товар	Цена	Количество	Выручка	Прибыль
2	3	02.12.2008	Иванов	монитор	1000	4	4000	400
3	41	02.12.2008	Иванов	процессор	400	10	4000	1000
4	15	03.12.2008	Иванов	монитор	1000	3	3000	300
5	13	03.12.2008	Иванов	процессор	400	6	2400	600
6	1	02.12.2008	Иванов	процессор	400	3	1200	300
7	31	04.12.2008	Иванов	монитор	1000	1	1000	100
8	50	03.12.2008	Иванов	монитор	1000	1	1000	100
9	32	04.12.2008	Иванов	кулер	30	20	600	100
10	2	02.12.2008	Иванов	мышь	50	10	500	50
11	4	02.12.2008	Иванов	кулер	30	15	450	75
12	14	03.12.2008	Иванов	мышь	50	9	450	45
13	29	04.12.2008	Иванов	процессор	400	1	400	100
14	16	03.12.2008	Иванов	кулер	30	10	300	50
15	47	03.12.2008	Иванов	мышь	50	6	300	60
16	37	02.12.2008	Иванов	мышь	50	5	250	50
17	30	04.12.2008	Иванов	мышь	50	2	100	10
18	44	03.12.2008	Иванов	кулер	30	4	60	20
19	46	02.12.2008	Петров	монитор	1000	4	4000	400
20	27	04.12.2008	Петров	монитор	1000	3	3000	300
21	38	03.12.2008	Петров	монитор	1000	3	3000	300
22	7	02.12.2008	Петров	монитор	1000	2	2000	200
23	19	03.12.2008	Петров	монитор	1000	1	1000	100
24	17	03.12.2008	Петров	процессор	400	2	800	200
25	25	04.12.2008	Петров	процессор	400	2	800	200
26	5	02.12.2008	Петров	процессор	400	1	400	100
27	26	04.12.2008	Петров	мышь	50	7	350	35
28	6	02.12.2008	Петров	мышь	50	5	250	25

Accounts

Рисунок 4.4 – Відсортований список продажів

Сортування можна проводити і за допомогою кнопок Sort Ascending (Сортувати за збільшенням) і Sort Descending (Сортувати по убуванню) стандартної панелі інструментів. Якщо список складений відповідно до вимог MS Excel, то для його сортування по певному стовпцю досить встановити в цьому стовпці покажчик осередку і натиснути потрібну кнопку. Якщо перед використанням кнопки виділити деякий діапазон осередків, при сортуванні враховуватимуться дані крайнього лівого його стовпця.

4.2.3. Фільтрація даних

Процеси введення записів в список і видалення з нього, а також процес пошуку інформації можна значно спростити за рахунок використання фільт-

рів, інтегрованих в MS Excel. Перевага застосування фільтрів в порівнянні з використанням форми даних полягає в тому, що результати запиту можна скопіювати в окрему область таблиці і використовувати в обчисленнях. Записи, які не відповідають заданому критерію, при фільтрації не відображаються, проте загальний порядок їх розташування залишається тим самим.

Можлива фільтрація з використанням як функції автофільтру, так і функції розширеного фільтру. Як правило, функція автофільтру може задовольнити користувача більш, ніж в 90 % випадків. Враховуючи це, а також її граничну простоту в порівнянні з функцією розширеного фільтру, обмежимося тут розглядом тільки функції автофільтру. При необхідності про функцію розширеного фільтру можна прочитати в [2].

Функція автофільтру дозволяє за допомогою визначення ряду критеріїв проводити відбір записів безпосередньо в робочому аркуші. Викликається вона за допомогою команди Data – Filter – AutoFHter (Дані – Фільтр – Автофільтр). Проте фільтри встановлюються тільки в тому випадку, якщо покажчик осередку знаходиться усередині списку.

Як критерій автофільтру можна використовувати вміст будь-якого осередку, окрім тих, в яких зберігаються заголовки стовпців. Щоб задати як критерій вміст якого-небудь стовпця, клацніть на кнопці зі стрілкою в рядку його заголовка і виберіть із списку потрібний елемент (рис. 4.5).

Якщо, наприклад, із списку стовпця Продавець вибрати елемент Петров, будуть відображені записи, що відносяться до продавців, які мають таке прізвище, а в рядку стану з'явиться повідомлення про те, скільки записів, що відповідають заданому критерію, знайдено (рис. 4.6).

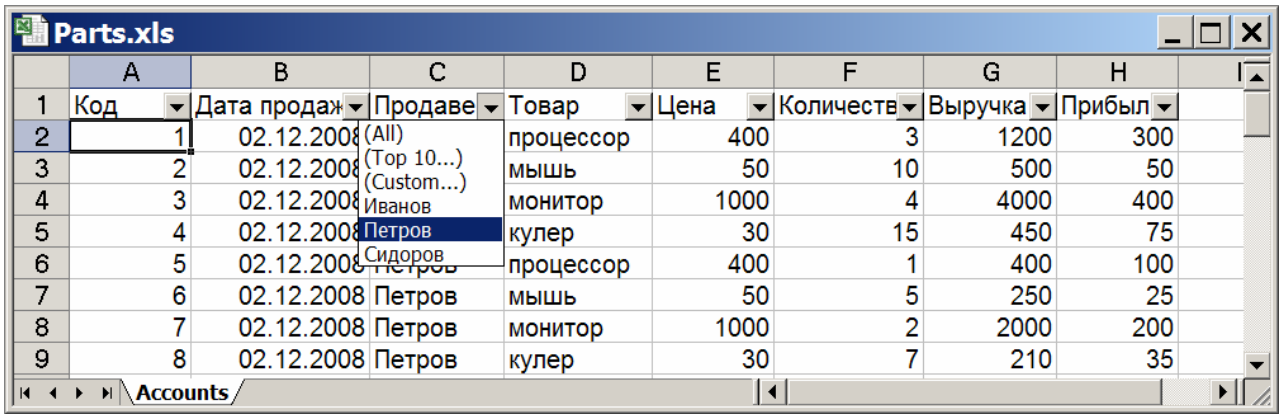


Рисунок 4.5 – Виклик функції автофільтру

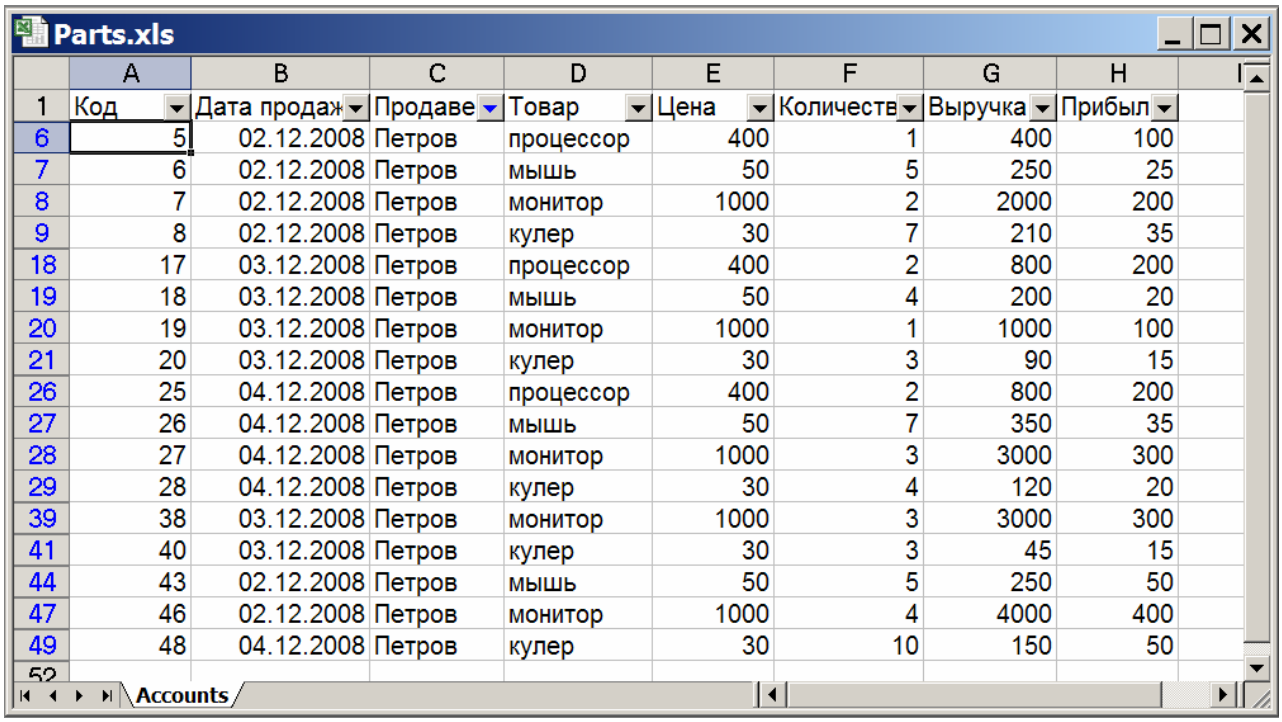


Рисунок 4.6 – Відфільтрований список продажів

Відбір записів можна продовжити, задаючи критерії в інших стовпцях. Новий критерій зв'язується з попереднім за допомогою логічного оператора And. У Excel 2003 і подальших версіях з'явилася можливість за допомогою функції автофільтру виконувати сортування записів. Для цього в списку критеріїв потрібно вибрати елемент Sort Ascending (За збільшенням) або Sort Descending (По убуванню).

Елемент (All) (Все) в списку автофільтру дозволяє відмінити критерій відбору записів для даного стовпця. Відмінити всі критерії фільтрації для списку можна за допомогою команди Data – Filter – Show All (Дані – Фільтр – Відобразити все), особливо корисної у тому випадку, коли задано декілька критеріїв фільтрації.

Встановивши елемент (Top 10...) (Перші 10...) у фільтрі стовпця, що містить числові значення, можна створити новий список, що складається з 10 найбільших або найменших чисел попереднього списку (значення 10 встановлюється за умовчанням). Після вибору елементу (Top 10...) відкриється діалогове вікно Top 10 AutoFilter з трьома полями введення. Вибравши в крайньому лівому полі елемент Top (Найбільші) або Bottom (Найменші), можна вказати системі, які значення попереднього списку повинні бути відображені в новому списку – максимальні або мінімальні. Центральне поле призначене для вказівки кількості пунктів, яку повинен містити список. Елементи Items і Percent, що знаходяться в правому полі діалогового вікна Top 10 AutoFilter дозволяють встановити числове або процентне обмеження на кількість елементів списку, що виводяться на екран.

При використанні функції автофільтру можна встановлювати і призначені для користувача критерії. Відкрийте поле списку стовпця, дані якого ви хочете відфільтрувати, і виберіть елемент (Custom...) (Призначений для користувача...). Коли з'явиться діалогове вікно Custom AutoFilter, задайте комплексний критерій, об'єднавши його компоненти за допомогою логічного оператора And або Or. Наприклад, на рис. 4.7 показане вікно Custom AutoFilter, в якому для стовпця «Ціна» вказаний критерій відбору рядків «більше або рівне 50» і «менше або рівне 400».

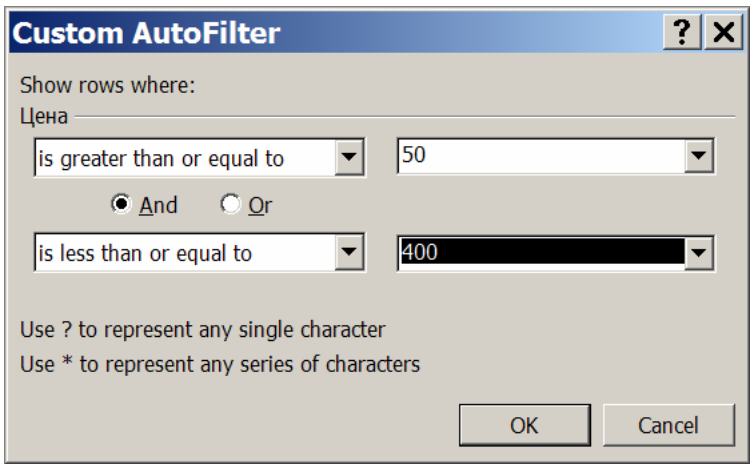


Рисунок 4.7 – Задавання призначеного для користувача критерію відбору

Отриманий в результаті фільтрації список можна вставити в іншу позицію робочого аркуша для подальшої обробки, використовувати при створенні діаграми або роздрукувати. Докладніше ці дії описані в наступному підрозділі.

4.2.4. Формули і функції

Формула – це вираз, що починається зі знаку рівності і визначає, які розрахунки треба виконати на робочому аркуші. Формула в комірці може включати наступні елементи: числові та текстові значення, посилання на інші комірки, знаки математичних і логічних дій, а також звернення до функцій.

При обчисленні формули використовується порядок дій, прийнятий в математиці. Для його зміни можна застосовувати круглі дужки. Види дій, які допускається використовувати у формулах, приведені в табл. 4.1.

Результатом обчислення формули, що включає арифметичні дії, є числове значення. Формула з операторами порівняння дає в результаті логічне значення True (Істина) або False (Неправда). Наприклад, якщо набрати в якому-небудь з вільних осередків вираз

=(A2+B2)*C2

після натиснення клавіші Enter або переходу в інший осередок замість цієї формули з'явиться число, що дорівнює сумі значень в осередках A2 і B2, помноженій на значення в осередку C2. Якщо необхідно повторити одну й ту ж формулу для ряду послідовних осередків – не обов'язково її багато разів набирати або копіювати. Можна навести покажчик миші на правий нижній кут осередку, в якому вже набрана одна формула (щоб цей покажчик прийняв форму тонкого хрестика), потім при натиснутій лівій кнопці миші «розтягнути» формулу на потрібні осередки. Адреси початкових даних при цьому автоматично коректуватимуться.

Таблиця 4.1 – Арифметичні і логічні дії

Код	Опис
+	Додавання
-	Віднімання
*	Множення
/	Ділення
%	Узяття відсотка
^	Піднесення до ступеня
=	Рівно
>	Більше
<	Менше
<=	Менше або рівно
>=	Більше або рівно
<>	Не рівно
&	Об'єднання рядків
:	Роздільник меж діапазону (наприклад, A2:C10)
;	Об'єднання діапазонів або несуміжних осередків (наприклад, A2;B3)
Пробіл	Перетин діапазонів (наприклад, вираз A2:E5 C1:E7 означає діапазон C2:E5)

Знаки додавання і віднімання, як і знак рівності, слугують для ідентифікації формули, тобто MS Excel розглядає дані, що починаються з будь-якого з цих знаків, як формулу. Після введення формули, яка починається знаком складання або віднімання, і натиснення клавіші Enter програма автоматично вставляє перед формулою знак рівності. Навики складання і викори-

стання формул студент може отримати самостійно (це дуже просто і цікаво) або звернувшись до літератури за системою MS Excel.

MS Excel надає також в розпорядження користувачів безліч спеціальних функцій, які можна застосовувати в обчисленнях. **Функція** – це символьний вираз деякої формули, що виконує визначені математичні, логічні чи інші дії. Вихідні дані передаються в неї за допомогою аргументів. Звернення до функції виконується шляхом задавання її імені, після якого йдуть круглі дужки. Якщо функція має аргументи, вони перераховуються в дужках і відділяються один від одного крапкою з комою. Наприклад, вираз

=SUM(A2:A4;A6:A8)

розмістить в осередку, в якому він набраний, суму чисел з осередків A2, A3, A4, A6, A7, A8.

Як аргумент можна задати числове або текстове значення, адресу осередку (абсолютну або відносну), адресу або ім'я діапазону. Викликаються функції з формул, причому в одній формулі може бути декілька функцій, об'єднаних різними знаками арифметичних дій. Якщо як аргументи функції використовуються інші функції, то такі функції називаються вкладеними.

Виклик функції у формулі, тобто введення її імені, дужок і аргументів, можна виконати з клавіатури – або безпосередньо в осередку, або в рядку формул. При цьому адреси осередків, вміст яких повинен використовуватися як аргументи функції, зручно визначати, виділяючи осередки за допомогою миші.

Аргументами функції можуть бути адреси як суміжних, так і несуміжних діапазонів осередків. Адреса суміжного діапазону складається з адрес першого і останнього осередків, розділених двокрапкою, а адреса несуміжного діапазону включає адреси несуміжних осередків, розділені крапкою з комою. Обидва символи розділення вставляються у формулу автоматично, при вказівці діапазону осередків за допомогою миші.

У разі некоректного запису функції (припустимо, користувач забув вказати обов'язковий аргумент, кількість закриваючих дужок не відповідає кількості тих, що відкривають і т. д.) з'являється повідомлення про помилку.

При роботі з функціями рекомендується використовувати рядок формул і **Майстер функцій**. В цьому випадку аргументи встановлюються в діалоговому вікні, що повністю виключає появу помилок, вірогідність виникнення яких при введенні функції з клавіатури досить висока. Детальніше про використання Майстра функцій можна почитати, наприклад, в [2].

4.3. Робота з банком даних на основі системи Microsoft Excel

У попередньому підрозділі були розглянуті базові дії, що дозволяють організувати на основі системи MS Excel простий банк даних. У даному підрозділі вивчимо складніші можливості цієї системи, за допомогою яких можна істотно розширити наочність і доступність інформації, що зберігається в створеному БНД. До таких можливостей слід віднести:

– підведення підсумків;

– створення звідних таблиць;

– побудова діаграм.

Розглянемо перераховані дії докладніше.

4.3.1. Підведення підсумків

Система MS Excel може автоматично обчислювати проміжні і загальні підсумки. Ця дія виконується тільки з тими даними, які організовані у вигляді списку, і лише над стовпцями, що містять числа. Перед визначенням проміжних підсумків потрібно відсортувати список так, щоб згрупувати значення в стовпцях, по яких потрібно розрахувати підсумкові значення.

Як простий приклад підрахуємо виручку і прибуток, принесені магазину кожним з продавців окремо за весь час торгівлі. Спочатку відсортуємо

список по продавцях, як було описано в п. 4.2.2. Потім викличемо команду Data – Subtotals (Дані – Підсумки). В результаті проведених дій буде відкрито діалогове вікно Subtotal (Проміжний підсумок), показане на рис. 4.8.

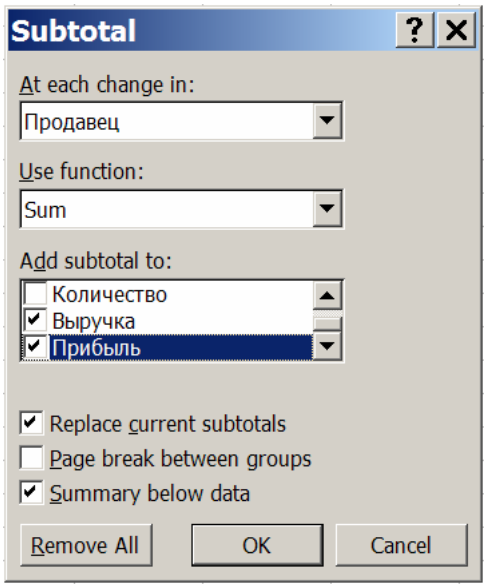


Рисунок 4.8 – Вікно проміжних підсумків

У списку At each change in (При кожній зміні в) вибирається заголовок стовпця, для якого необхідно обчислювати проміжні підсумки після кожної зміни даних на робочому аркуші. У списку Use function (Використовувати функцію) встановлюється функція, яка використовуватиметься при обчисленні підсумкових значень. Опис всіх доступних у вказаному вікні функцій наведений в табл. 4.2.

У списку Add subtotal to (Додати підсумки) необхідно встановити прапорці для тих стовпців, значення яких використовуватимуться при обчисленні підсумків (у нашому випадку це «Виручка» і «Прибуток»). За умовчанням рядки, що містять підсумкові значення, розміщуються під рядками з початковими даними. Якщо їх потрібно розмістити над початковими даними, слід відключити прапорець Summary below data (Підсумки під даними). При виводі на друк кожну групу підсумкових даних можна розташувати на окремій сторінці. Для цього потрібно встановити прапорець Page break between groups

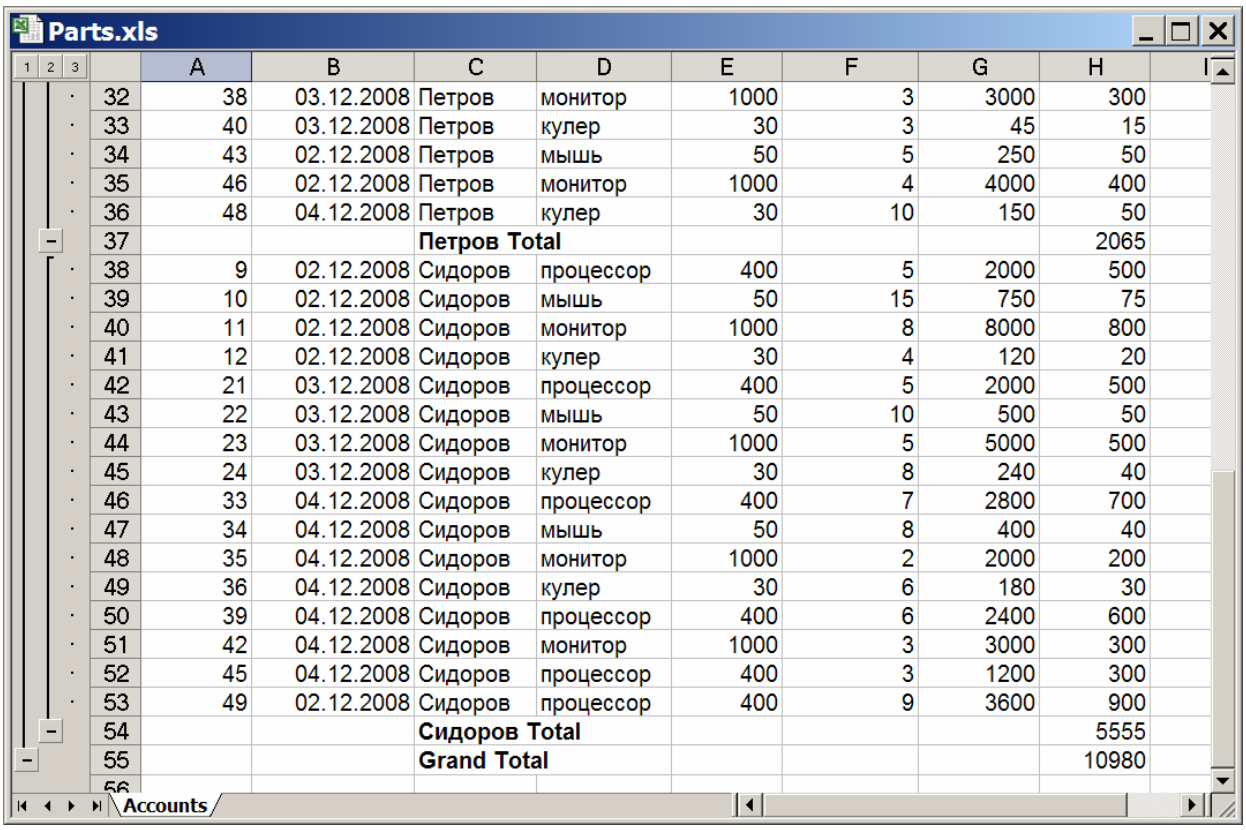
(Розрив сторінки між групами). Прапорець Replace current subtotals (Замінити поточні підсумки) управляє видаленням раніше створених підсумкових значень. Для видалення всіх рядків з такими даними призначена кнопка Remove All (Прибрати все).

Таблиця 4.2 – Функції, використовувані при обчисленні підсумків

Функція	Опис
Sum (Сума)	Підсумовує всі значення і видає загальний підсумок
Count (Кількість)	Визначає кількість елементів ряду
Average (Середнє)	Обчислює середнє арифметичне для ряду
Max (Максимум)	Визначає найбільше значення в ряду
Min(Мінімум)	Визначає найменше значення в ряду
Product (Добуток)	Обчислює добуток всіх значень ряду
Count Nums (Кількість чисел)	Встановлює у виділеній області кількість осередків, що містять числові значення
StdDev (Стандартне відхилення)	Визначає значення стандартного відхилення для вибірки даних, генеральної вибірки, що є підмножиною
StdDevp (Стандартне відхилення для всього ряду)	Визначає значення стандартного відхилення для генеральної вибірки
Var (Дисперсія)	Визначає значення дисперсії для вибірки даних, генеральної вибірки, що є підмножиною
Varp (Дисперсія для всього ряду)	Визначає значення дисперсії для генеральної вибірки

Після того, як задано розрахунок підсумків по стовпцях, що містять суми виручки і прибутку, таблиця доповнюється рядками, що містять підсумкові значення по кожному продавцеві. У останньому зі вставлених рядків знаходиться загальний підсумок Grand Total (рис. 4.9).

Істотно, що цей загальний підсумок обчислюється не на основі проміжних підсумкових значень, а з урахуванням всіх початкових даних стовпця. Наприклад, якщо обчислювати середнє значення, при розрахунку загального підсумку використовуватимуться всі дані стовпця (програма не обчислює «середнє середнього»).



Parts.xls

	A	B	C	D	E	F	G	H
32	38	03.12.2008	Петров	монитор	1000	3	3000	300
33	40	03.12.2008	Петров	кулер	30	3	45	15
34	43	02.12.2008	Петров	мышь	50	5	250	50
35	46	02.12.2008	Петров	монитор	1000	4	4000	400
36	48	04.12.2008	Петров	кулер	30	10	150	50
37			**Петров Total**					2065
38	9	02.12.2008	Сидоров	процессор	400	5	2000	500
39	10	02.12.2008	Сидоров	мышь	50	15	750	75
40	11	02.12.2008	Сидоров	монитор	1000	8	8000	800
41	12	02.12.2008	Сидоров	кулер	30	4	120	20
42	21	03.12.2008	Сидоров	процессор	400	5	2000	500
43	22	03.12.2008	Сидоров	мышь	50	10	500	50
44	23	03.12.2008	Сидоров	монитор	1000	5	5000	500
45	24	03.12.2008	Сидоров	кулер	30	8	240	40
46	33	04.12.2008	Сидоров	процессор	400	7	2800	700
47	34	04.12.2008	Сидоров	мышь	50	8	400	40
48	35	04.12.2008	Сидоров	монитор	1000	2	2000	200
49	36	04.12.2008	Сидоров	кулер	30	6	180	30
50	39	04.12.2008	Сидоров	процессор	400	6	2400	600
51	42	04.12.2008	Сидоров	монитор	1000	3	3000	300
52	45	04.12.2008	Сидоров	процессор	400	3	1200	300
53	49	02.12.2008	Сидоров	процессор	400	9	3600	900
54			**Сидоров Total**					5555
55			**Grand Total**					10980

Рисунок 4.9 – Аркуш з підсумками

При створенні підсумків таблиця структурується. У цьому неважко переконатися, поглянувши на рис. 4.9. Створення рівнів структури приводить до підвищення наочності таблиці. Щоб відобразити на екрані тільки підсумкові дані, слід виконати клацання на кнопці другого рівня структури, внаслідок чого дані третього рівня (початкові значення) будуть приховані. Для того, щоб відновити відображення початкових значень, необхідно виконати клацання на кнопці третього рівня. Клацаючи на кнопках «+» і «–», можна приховувати надмірну, неактуальну інформацію і формувати потрібний звіт з певним ступенем деталізації.

4.3.2. Створення звідних таблиць

Звідна таблиця є інтерактивним засобом, що дозволяє одержати дані з деякого джерела (списку, таблиці, бази даних) і виконати їх аналіз та перегляд різними способами. За допомогою звідної таблиці можна переглянути

необхідну для аналізу частину наявної інформації, відфільтрувати дані і розташувати результати на різних робочих аркушах, відсортувати дані, автоматично згенерувати підсумки. Для більш наочного відображення результатів разом зі звідною таблицею можна створити діаграму.

Звідну таблицю можна створити на основі області таблиці, цілої таблиці або декількох таблиць. Початкові дані для звідних таблиць повинні бути організовані у вигляді списку, кожен рядок якого містить одні і ті ж категорії інформації, наведені в певному порядку. У першому рядку списку повинні знаходитися заголовки стовпців. Вони використовуються для створення полів звідної таблиці і ідентифікації окремих елементів даних, тобто унікальних значень полів, узятих зі списку. Не дивлячись на те, що поля створюються на основі стовпців, дані, що містяться в них, можна розміщувати як в стовпцях, так і в рядках звідної таблиці.

Слід уникати ситуацій, коли робочий аркуш містить декілька списків, оскільки в цьому випадку можна отримати несподіваний результат при виконанні фільтрації і сортування або при обчисленні загальних підсумків. Впевніться, що список відокремлено від інших даних робочого аркуша порожніми рядками і стовпцями, але намагайтесь не застосовувати їх безпосередньо в списку. Якщо необхідно відокремити назви рядків або стовпців даних, використовуйте межі, а не порожні рядки або стовпці, і не заповнюйте рядок символами підкреслення.

Створення і обробка звідних таблиць здійснюються за допомогою спеціального **Майстра звідних таблиць**, який запускається командою Data, – PivotTable and PivotChart Report (Дані – Звідна таблиця). Параметри звідної таблиці задаються в трьох діалогових вікнах. Продемонструємо роботу Майстра, склавши звідну таблицю, що відображає підсумки роботи продавців.

Після виклику Майстра з'являється діалогове вікно PivotTable and PivotChart Wizard – Step 1 of 3, в якому вказуються джерело даних для звідної таблиці і тип звіту – звідна таблиця або діаграма (рис. 4.10).

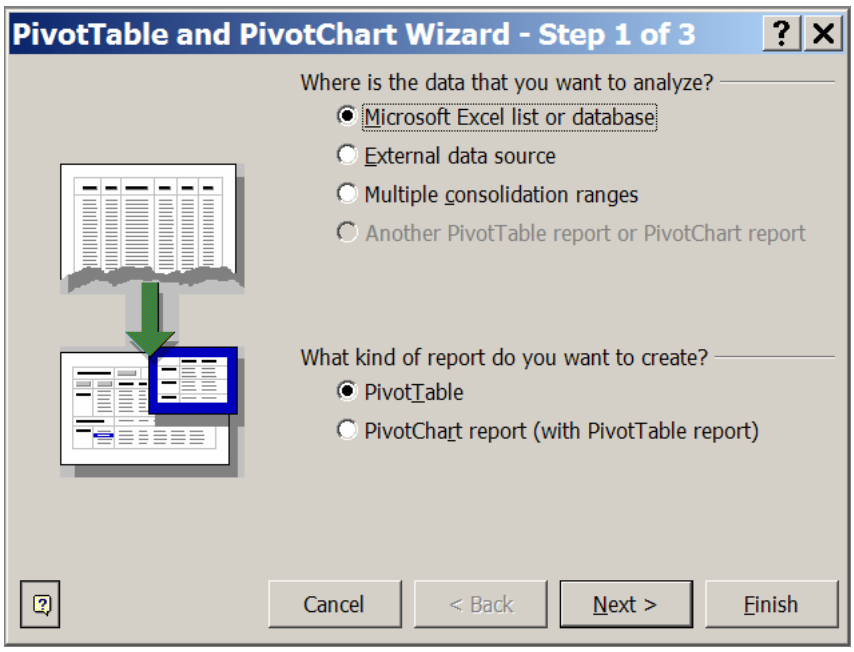


Рисунок 4.10 – Перше вікно Майстра звідних таблиць

В області Where is the data that you want to analyze? (Де дані, які Ви хочете аналізувати?) цього вікна за умовчанням активізований перемикач Microsoft Excel list or database (Список або база даних Microsoft Excel), за допомогою якого задається пошук джерела даних на робочому аркуші. Якщо джерело знаходиться у файлах зовнішньої бази даних, необхідно вибрати перемикач External data source (Зовнішнє джерело даних). Для того, щоб об'єднати декілька однакових по структурі списків, виберіть перемикач Multiple consolidation ranges (Декілька діапазонів консолідації). Якщо ви хочете як початкові дані використовувати інформацію, що зберігається в іншій звідній таблиці, потрібно вибрати перемикач Another PivotTable report or PivotChart report (Інша звідна таблиця або діаграма). Існуюча звідна таблиця може слугувати джерелом даних для нової звідної таблиці.

В області What kind of report do you want to create? (Вид створюваного звіту) за умовчанням активізований перемикач PivotTable (Звідна таблиця). Залиште його незмінним.

Після натиснення кнопки Next здійснюється перехід до наступного діалогового вікна Майстра (рис. 4.11). У цьому вікні визначається діапазон осе-

редків, дані з яких будуть включені в звідну таблицю. Якщо перед запуском Майстра покажчик осередку знаходився всередині таблиці, то програма автоматично вставить в поле Range адресу таблиці. Якщо ж дані для звідної таблиці знаходяться в іншій книзі, слід, натиснувши кнопку Browse, завантажити її з діалогового вікна вибору файлу. Вкажіть діапазон осередків, на основі якого повинна бути складена звідна таблиця, і, натиснувши кнопку Next, перейдіть в наступне вікно Майстра.

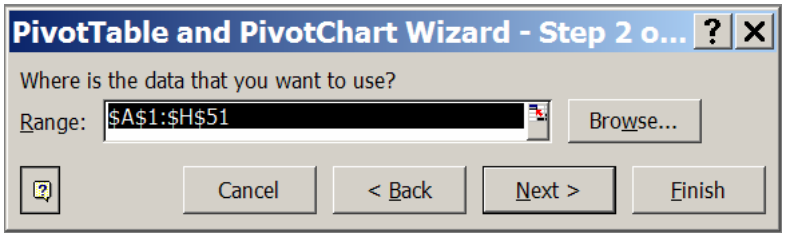


Рисунок 4.11 – Друге вікно Майстра звідних таблиць

Третє вікно Майстра звідних таблиць (рис. 4.12) призначене для вибору розташування звідної таблиці – на новому аркуші (перемикач New worksheet) або на поточному аркуші (перемикач Existing worksheet). У разі розміщення таблиці на поточному робочому аркуші необхідно вказати позицію її лівого верхнього кута. На практиці доцільніше розміщувати таблицю на новому аркуші.

Після натиснення в третьому вікні Майстра кнопки Finish будуть відображені панель інструментів PivotTable і вікно PivotTable Field List (Список полів звідної таблиці), а на робочому аркуші з'являться області, в які можна перетягувати поля (рис. 4.13). Всього таких областей чотири:

– Drop Column Fields Here (Перетягніть сюди поля стовпців);

– Drop Row Fields Here (Перетягніть сюди поля рядків);

– Drop Page Fields Млості (Перетягніть сюди поля сторінок);

– Drop Data Items Here (Перетягніть сюди елементи даних).

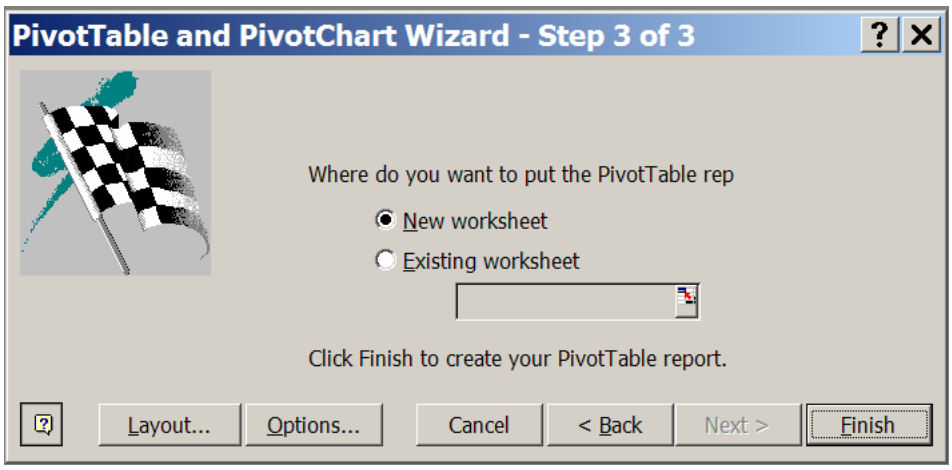


Рисунок 4.12 – Третє вікно Майстра звідних таблиць

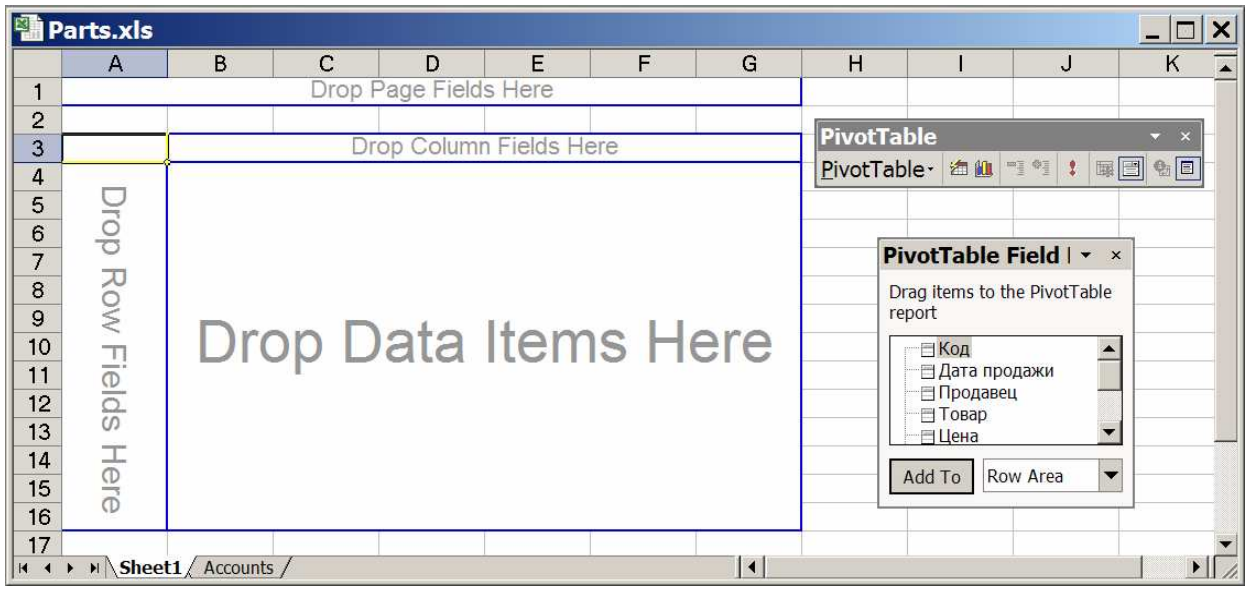


Рисунок 4.13 – Області перетягання і панель інструментів

В області стовпців і рядків розміщують поля, по яких проводиться порівняння або аналіз. У область елементів даних перетягують поля, призначені для розрахунків підсумкових показників. Область сторінок не обов'язкова для заповнення, її використовують при необхідності створити декілька звітів, заданих в попередніх областях.

Які поля куди слід перетягувати – залежить від типу звіту. Так, для нашої таблиці можна створити звіт, в якому відображатимуться дані про прибуток, принесений магазину кожним з продавців в кожен з днів. З вікна

PivotTable Field List перетягуємо в полі рядків поле Продавець, в полі стовпців – поле Дата продажу, а в область даних – поле Прибуток. В результаті одержуємо звідну таблицю, зображену на рис. 4.14.

4.3.3. Побудова діаграм

Діаграмою називається малюнок, що зображає в графічному вигляді деяку сукупність числових даних. Виключну корисність діаграм зумовлено тим, що людина сприймає інформацію в графічній формі значно легше і швидше, ніж в числовій.

Parts.xls

	A	B	C	D	E
1					
2					
3	Sum of Прибыль	Дата продажи			
4	Продавец	02.12.2008	03.12.2008	04.12.2008	Grand Total
5	Иванов	1875	1175	310	3360
6	Петров	810	650	605	2065
7	Сидоров	2295	1090	2170	5555
8	Grand Total	4980	2915	3085	10980
9					

Sheet1 Accounts

Рисунок 4.14 – Готова звідна таблиця

Залежно від виду графічного зображення даних, діаграми розрізняються за типом. MS Excel надає користувачам можливість працювати з діаграмами різноманітних типів. Діаграма завжди пов'язана з даними, на основі яких вона створена, і автоматично оновлюється при їх зміні. Для створення діаграм вельми зручно використовувати **Майстер діаграм**. Розглянемо роботу з ним на простому прикладі. Намалюємо графік синусоїди на її періоді.

Перш ніж викликати Майстер діаграм, на робочому аркуші необхідно виділити осередки, дані яких будуть зображені на діаграмі. Виділений фрагмент повинен містити осередки з назвами рядків і стовпців, які згодом будуть використані в легенді діаграми, а також на самій діаграмі як мітки осей. При

необхідності дані слід відсортувати. Стовпець значень синусоїди побудуємо за допомогою відповідної функції, як було описано раніше.

Після вибору пункту меню Insert – Chart (Вставити – Діаграма), на екрані з'явиться перше діалогове вікно Майстра діаграм, в якому можна вибрати тип діаграми (рис. 4.15). Це вікно містить дві вкладки: Standard Types (Стандартні типи) і Custom Types (Призначені для користувача типи). У списку Chart type (Тип діаграми) першої вкладки слід вибрати тип діаграми, а в полі Chart sub-type (Підтип діаграми) – її підтип. Діаграми одного типу можна відформатувати по-різному. Рекомендується вибирати такий формат, який найбільшою мірою відповідає меті побудови діаграми. Надалі формат діаграми можна буде змінити. Для нашого прикладу виберемо тип Line (Лінія), підтип Line.

Для того, щоб перейти до наступного етапу створення діаграми, натисніть кнопку Next. На екрані з'явиться друге діалогове вікно Майстра із зображенням діаграми вибраного типу (рис. 4.16). На вкладці Data Range (Діапазон даних) цього вікна вкажіть адреси осередків, значення яких будуть ви-

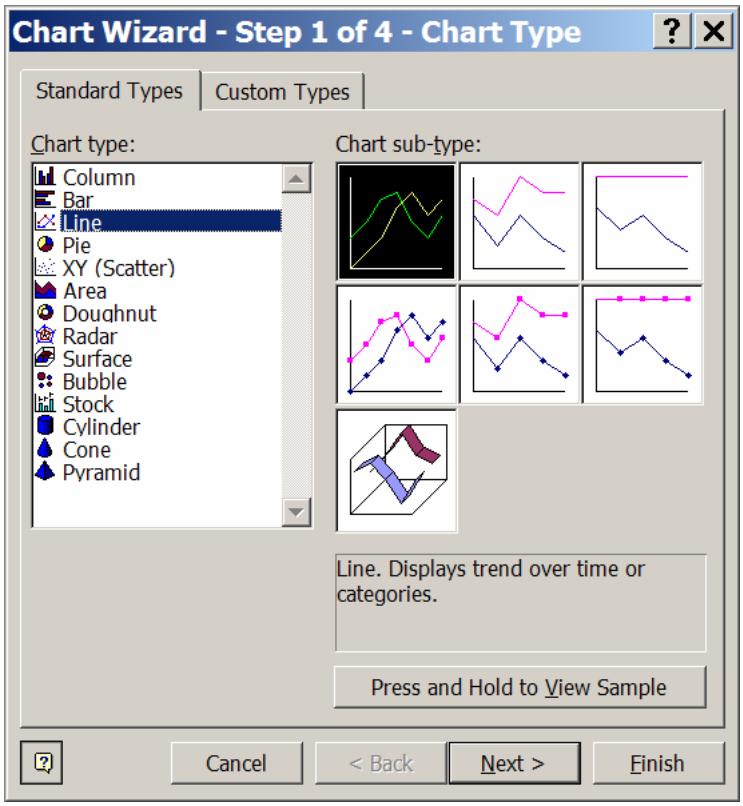


Рисунок 4.15 – Перше вікно Майстра діаграм

користані при побудові діаграми. Якщо Майстер діаграм викликано після виділення потрібних осередків, то в цьому діалоговому вікні вже відображаються адреси цих осередків. При необхідності пропоновані адреси можна змінити.

Перемикачі Rows (Рядки) і Columns (Стовпці) області Series in (Ряди в) дозволяють визначити варіант побудови рядів даних – по рядках або по стовпцях. Якщо програма автоматично визначає, що використовуватиметься як ряд даних, відповідний перемикач активізується вже при відкритті вікна.

Натиснувши кнопку Next (Далі), переходимо в третє діалогове вікно Майстра, де можна задати параметри форматування діаграми. Це вікно містить декілька вкладок, в кожній з яких визначається вигляд одного з елементів діаграми. Докладніше про зміст цих вкладок можна прочитати в [2].

Натиснувши кнопку Next ще раз, відкриваємо останнє вікно Майстра діаграм, з тим щоб задати в ньому місцеположення діаграми: на окремому аркуші (перемикач As new sheet) або на аркуші з таблицею (перемикач As object in). У останньому випадку потрібний аркуш вибирається з поля списку (рис. 4.17).

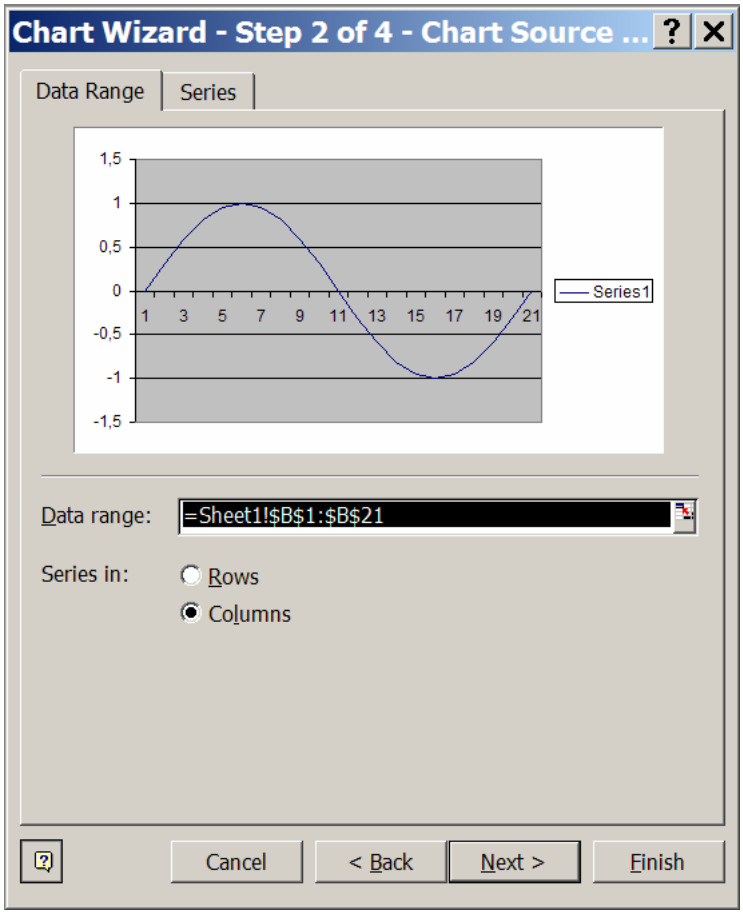


Рисунок 4.16 – Друге вікно Майстра діаграм

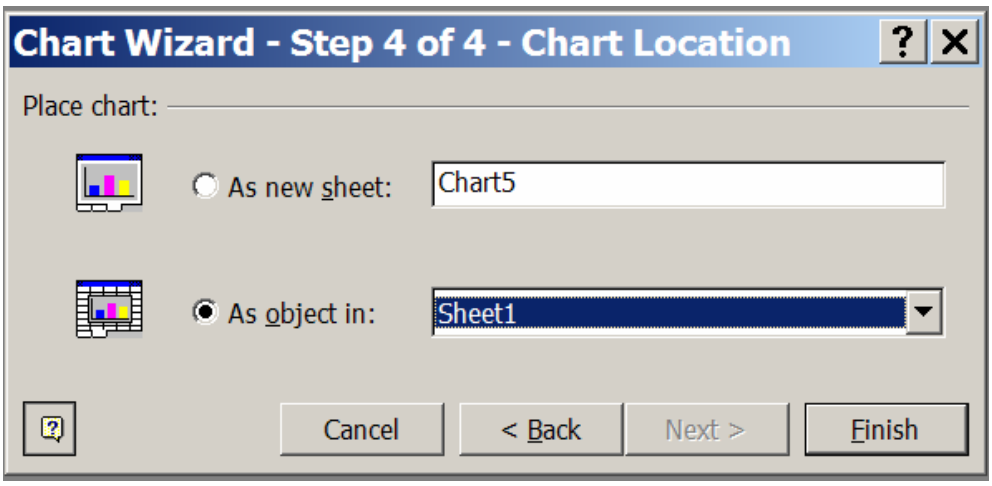


Рисунок 4.17 – Четверте вікно Майстра діаграм

Закрийте діалогове вікно, натиснувши кнопку Finish (Готово), після чого створена діаграма з'явиться на робочому аркуші (рис. 4.18). Діаграму, як і інші об'єкти, можна розмістити в будь-якому місці аркуша шляхом переміщення її за допомогою миші. Крім того, існує можливість методом переміщення маркерів змінювати розміри області діаграми.

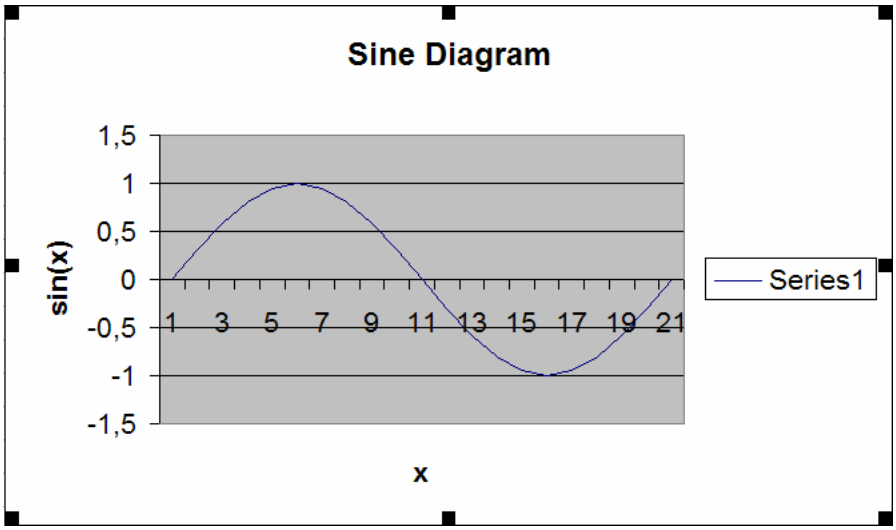


Рисунок 4.18 – Готова діаграма

Питання й тести для самоконтролю

1. Навіщо потрібна система програмування Microsoft Excel?
2. Що таке електронна таблиця?
3. Дайте загальний опис системи програмування Microsoft Excel.
4. Як створити список за допомогою MS Excel?
5. Розкажіть про сортування списків.
6. Що таке фільтрація даних і як вона здійснюється?
7. Що Ви знаєте про формули і функції MS Excel?
8. Як підвести підсумки в списку?
9. Розкажіть про створення звідних таблиць.
10. Що таке діаграми і як вони будуються?

Розділ 5. ВИКОРИСТАННЯ МОВ ПРОГРАМУВАННЯ В ІНФОРМАЦІЙНИХ СИСТЕМАХ

Незважаючи на досить широкі можливості системи MS Excel по створенню нескладних банків даних, у ряді випадків виникає ситуація, коли користувачеві не вистачає якої-небудь нескладної функції для повного задоволення його потреб, а використати більш розвинені СУБД немає необхідності або бажання. У таких випадках може допомогти розширення можливостей системи MS Excel шляхом написання власних функцій користувача, що дозволяють виконувати якісь нестандартні, але необхідні користувачеві дії з обробки даних.

Такі засоби надає вбудований у систему MS Excel інтерпретатор мови високого рівня Visual Basic for Applications (скорочено **VBA**). Цюя мову використано по таких причинах:

– VBA є єдиним середовищем розробки додатків у всіх програмах Microsoft Office. Подібний вибір обумовлено винятковими властивостями мови Basic, що бере свій початок ще в 1960-х роках і безупинно розвивається з тих пір. Однією з таких властивостей є унікальне сполучення простоти базових конструкцій мови й широких можливостей по складанню алгоритмів обробки даних для різних цілей. У цьому сенсі мова VBA поступається, хіба що, Фортрану;

– за допомогою VBA можна записувати послідовності повторюваних команд, характерних для якого-небудь додатка, і призначати правила (умови), відповідно до яких ці послідовності (процедури VBA) будуть викликатися в додатку. Наприклад, після натискання деякої кнопки, робочий аркуш MS Excel може бути відформатований з урахуванням вимог користувача, записаних на VBA і збережених у робочій книзі або шаблоні;

– застосувавши засіб VBA, можна забезпечити нестандартний діалог з користувачем, створюючи діалогові форми й обробляючи реакцію користувача на подію в додатку;

– використання єдиної мови програмування сприяє більш тісній взаємодії Excel із продуктами Microsoft Office. За допомогою VBA можна розробляти додатки, що одночасно застосовують компоненти декількох програм.

Розглянемо загалом найбільш корисні можливості мови VBA з погляду розроблювача СУБД. Зрозуміло, в одному розділі неможливо описати всі засоби мови програмування. Тому для більш глибокого ознайомлення з можливостями VBA варто звернутися до спеціальної літератури (наприклад, [2]).

5.1. Загальний опис мови VBA

Visual Basic for Applications – це об’єктно-орієнтована мова програмування високого рівня, вбудована в усі програми пакета MS Office. Включення в зазначений пакет сучасної повнофункціональної мови програмування, інструментів для написання на ній програм, їх редагування й налагодження, а також великих бібліотек стандартних програм дозволяє створювати програмні продукти, здатні з достатньою повнотою автоматизувати вирішення конкретних завдань користувача.

VBA є об’єктно-орієнтованим середовищем, що містить великий набір об'єктів, кожен з яких має безліч властивостей і методів. Сам об'єкт і всі інструменти, необхідні для його створення, відображення або знищення, відносяться до певного класу. Ім'я класу вказує на тип об'єктів, які він містить. Класи можна розглядати як основу для створення інших об'єктів цього ж типу. Тому всі об'єкти одного класу будуть діяти однаково.

Властивості й методи є членами класу. За допомогою властивостей описується, як виглядає об'єкт, зокрема дається інформація про прийоми форматування тексту, колір і розмір шрифту. Методи є процедурами, або, інакше кажучи, множиною операторів, що здійснюють певні завдання, які можуть бути виконані для об'єкта (процедури створення й видалення об'єкта, процедури подій, що визначають принцип взаємодії об'єкта з користувачем, і т.д.).

5.1.1. Об'єкти мови

При об’єктно-орієнтованому програмуванні практично всі компоненти середовища розробки є **об'єктами**. В MS Excel VBA-об'єктом вважається будь-який елемент додатка – комірка, аркуш, робоча книга, діаграма. Фактично об'єктом є й сам додаток MS Excel.

Об'єкти можуть включати області комірок, рамки комірок, вікна, сценарії, стилі, і цей список можна продовжити. Кожен клас об'єктів має свою множину властивостей, функцій і подій. Коли об'єкт додається на робочий аркуш, створюється екземпляр класу об'єктів, і об'єкту привласнюються значення властивостей, які визначені в цьому класі.

За замовчуванням екземпляри того самого класу мають ім'я, що складається з імені батьківського класу й порядкового номера. Властивості кожного екземпляра того самого класу ініціюються тими самими значеннями, за винятком властивості Name. Тому, що VBA привласнює значення всім елементам, які цього потребують, користувачеві можна починати працювати в даному середовищі, володіючи навіть мінімумом знань.

5.1.2. Властивості об'єктів

Властивість є атрибутом об'єкта, що описує, як об'єкт виглядає (його колір, розмір і місце розташування) і як він діє (чи є видимим, чи посилається на інший об'єкт). Коли створюється об'єкт, MS Excel виконує процедуру створення екземпляра цього об'єкта. Дана процедура, збережена в класі об'єктів, привласнює значення всім властивостям, що дозволяє відразу ж працювати з об'єктом.

Якщо користувач захоче змінити кілька властивостей, то він, як правило, виявить, що здебільшого вони задані саме так, як йому потрібно, тому вносити зміни звичайно не доводиться. Для того, щоб у макросі VBA все-таки привласнити властивості нове значення, необхідно лише створити опе-

ратор присвоювання, у якому ліворуч від знака рівності будуть зазначені ім'я й властивість об'єкта (розділені крапкою), а праворуч – нове значення.

Властивість аркуша Name відображається на його ярличку. Причому дана властивість входить також у безліч інших класів.

Метод – це дія, що може бути виконана над об'єктом. Методи реалізуються за допомогою виконання процедури, що є членом класу об'єктів. Синтаксис виклику методу такий: ім'я об'єкта й ім'я методу, розділені крапкою.

5.1.3. Події

Щоразу, коли користувач взаємодіє з певним об'єктом на робочому аркуші, відбувається **подія**. Кожен клас об'єктів має власну групу подій, на які об'єкти даного класу реагують. Так, робоча книга має подію NewSheet, що виконується щоразу, коли користувач додає в неї новий аркуш; подія SelectionChange відбувається при виборі користувачем іншої комірки або області комірок. При події створення процедури MS Excel автоматично додає в її код перший і останній рядки.

При бажанні можна самостійно написати програму для обробки події, на яку повинні реагувати об'єкти. Для цього необхідно знати основні правила взаємодії об'єктів, властивостей і методів. Наприклад, коли Ви створюєте аркуш у робочій книзі, функцією-членом у класі Worksheet створюється об'єкт Worksheet. Значення властивостей нового об'єкта привласнюються за замовчуванням,

5.2. Процедури й функції мови VBA

В VBA підтримується така програмна структура (рис. 5.1). На вищому рівні ієрархії знаходиться додаток, далі йдуть проекти, пов'язані з фактичними документами цього додатка. На третьому рівні перебувають модулі (мо-

дулі додатка, модулі користувача, модулі класу, модулі форм і модулі посилань), а на нижчому – їхні процедури та функції.

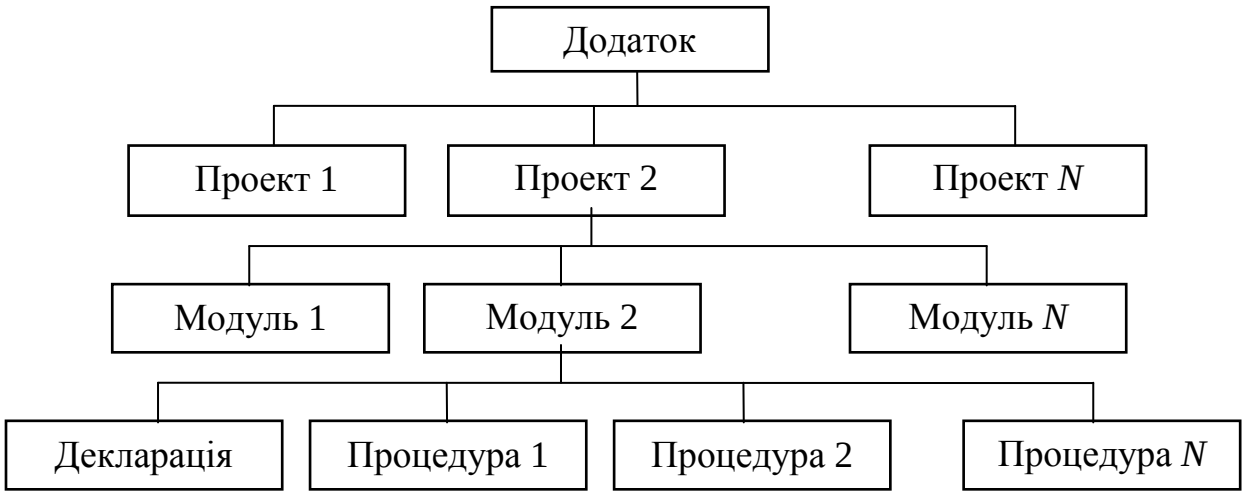


Рисунок 5.1 – Структура програми в VBA

Модуль – це частина програми, оформлена у вигляді, що допускає її незалежну трансляцію. Модуль складається із двох розділів: розділу оголошень (Declaration) і розділу процедур і функцій. У першому розділі описуються глобальні змінні, типи, визначені користувачем, і зліченні типи, у другому – процедури та функції.

Процедурою називається фрагмент коду (мінімальна семантично закінчена програмна конструкція), розміщений між операторами Sub і End Sub. Спрощений синтаксис процедури такий:

[Private|Public] Sub <*ім'я_процедури*> (<*аргумент_1*>, ... , <*аргумент N*>)

<*оператор*>

...

<*оператор*>

End Sub

Синтаксис функцій трохи відрізняється від синтаксису процедур, тому що для функції необхідно вказати тип значення, що повертається нею:

[Private|Public] Function <*ім'я_функції*> (<*аргумент_1*>, ... , <*аргумент N*>) As <*тип_значення_що_повертається*>

<*оператор*>

...

<*оператор*>

<*ім'я_функції*> = <*обчислене_значення*>

End Function

Ключове слово Private обмежує область видимості процедури або функції, для якої воно задане, модулем, у якому вона описана. У результаті її можуть викликати тільки процедури або функції цього ж модуля. Ключове слово Public, навпаки, оголошує процедуру або функцію доступною для всіх модулів проекту. За замовчуванням будь-яка процедура або функція загальнодоступна, тобто має статус Public.

Виконання програмного коду процедури або функції відбувається при передачі їй керування. Передача керування (виклик) може здійснюватися різними способами. У загальному випадку підпрограма викликається із програмного коду за допомогою спеціального оператора Call, у якому, крім цього ключового слова, вказуються ім'я процедури й фактичні параметри виклику, список яких вміщується в круглі дужки.

Дуже важливу роль у програмуванні на VBA грає такий різновид процедури, як процедура без параметрів. Процедури цього типу можуть виступати в ролі процедур обробки подій. Процедури обробки подій є основою механізму зв'язку подій з програмним кодом для їхньої обробки.

Функції в VBA, як правило, містять списки параметрів. Вони не можуть використовуватися для обробки подій. Викликати ж їх можна як за допомогою оператора Call, так і безпосередньо в процесі розрахунку значення

виразу VBA (з коду інших процедур або функцій) або формули робочого аркуша MS Excel.

Звичайно програма мовою VBA складається з декількох процедур і функцій. Опис цих програмних одиниць зберігається в модулях. Модулі, у свою чергу, як складові частини входять у проекти. Проекти зберігаються у файлах робочих книг MS Excel, але працювати з ними можна лише при наявності окремого додатка – редактора Visual Basic. Якщо потрібно викликати процедуру або функцію, опис якої перебуває в іншому модулі, її ім'я при виклику вказується в такий спосіб: *ім'я_модуля.им'я_процедури*. Але якщо опис процедури або функції розміщено в іншому проекті, при виклику треба використати інший синтаксис: *ім'я_проекту.ім'я_модуля.ім'я_процедури*.

5.2.1. Створення користувальницьких функцій

За допомогою мови VBA можна створювати власні функції, доступ до яких здійснюється за допомогою вікна Майстра функцій. Спробуємо написати нескладну функцію для розрахунку ціни товару з урахуванням податку на додану вартість (ПДВ).

Спочатку створюється робоча книга, у комірці A1 якої записується ціна товару без ПДВ. За допомогою команди Visual Basic Editor з меню Tools/Macro, відкривається вікно Microsoft Visual Basic та активізується команда Module з меню Insert або з контекстного меню, викликаного щигликом правої кнопки миші на вільному полі у вікні Project, як показано на рис. 5.2. Коли в області коду з'явиться вікно редактора VBA, вводиться функція, текст якої наведений на рис. 5.3.

Функція починається з оператора Function. Після ключового слова потрібне ім'я функції. У дужках вказується аргумент, що повинен бути переданий функції при виклику. Потім йдуть виконувані оператори (обчислення ПДВ і роздрібної ціни). Закінчується функція оператором End Function. При наборі варто уважно дивитися на екран, тому що редактор VBA часто нама-

гається «допомогти» у наборі, автоматично додаючи потрібні, на його думку, елементи в текст програми. Іноді така «послужливість», навпаки, утрудняє роботу. Викликати створену функцію можна, ввівши її ім'я Retail_Price з клавіатури.

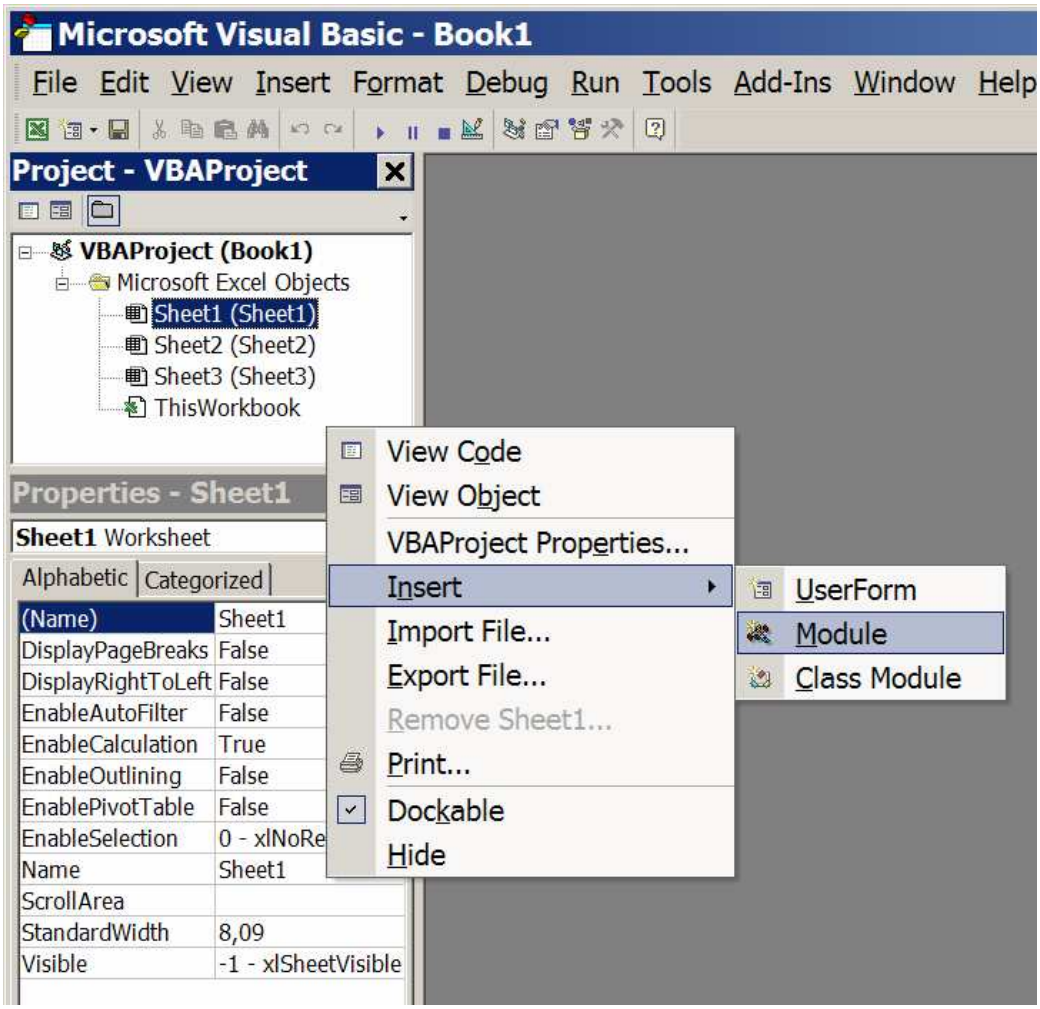


Рисунок 5.2 – Вставка нового модуля VBA

Додану користувачем функцію можна використати як звичайну функцію MS Excel. Для цього шляхом виклику команди Function меню Insert відкрийте вікно Insert Function (Майстер функцій), у списку «Select a category», відмітьте елемент User Defined, і ім'я створеної вами функції Retail_Price буде висвітлено в списку наявних функцій «Select a function» (рис. 5.4).

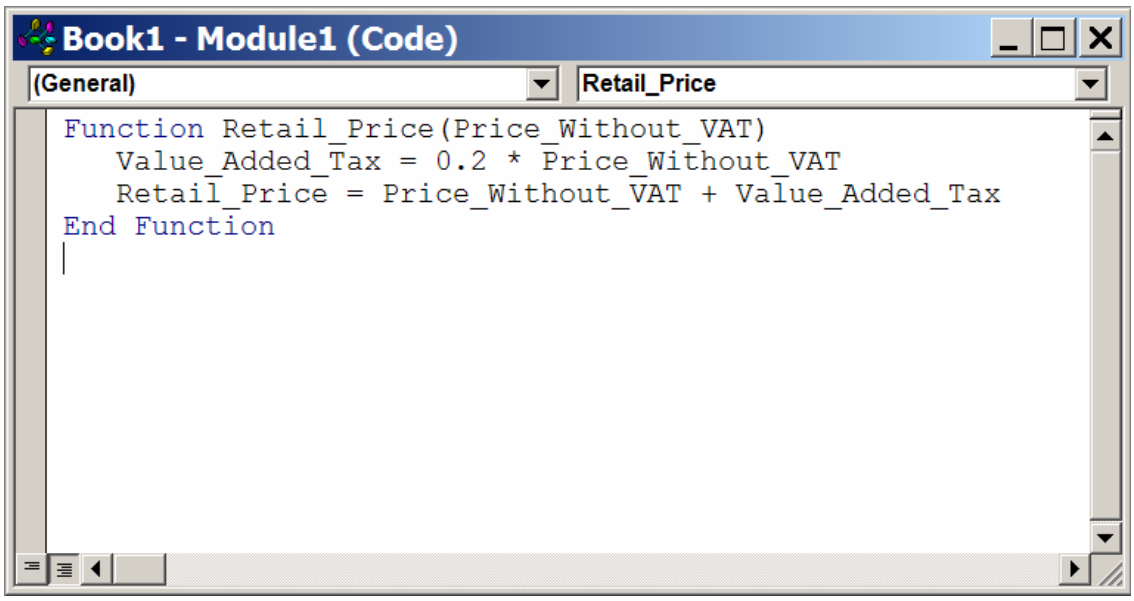


Рисунок 5.3 – Вікно редактора VBA

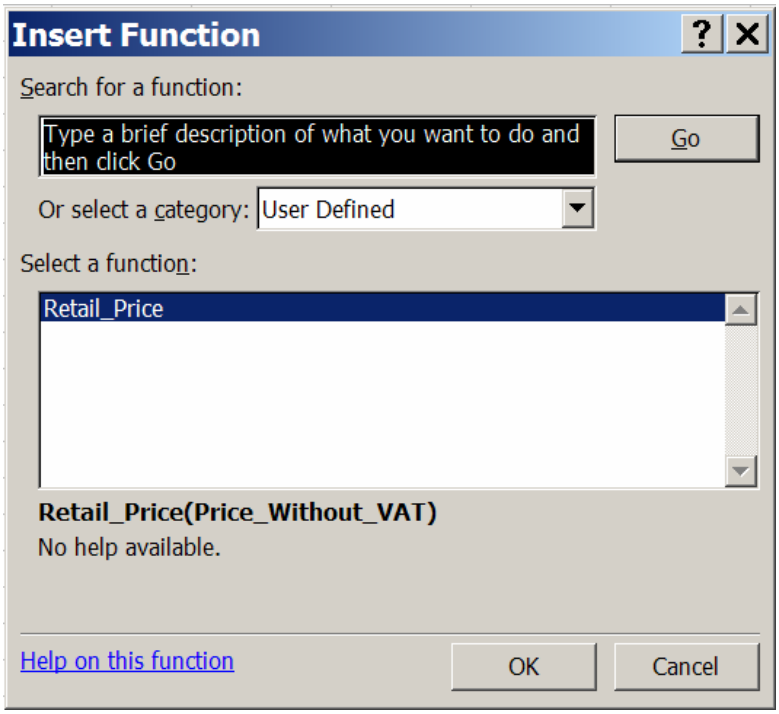


Рисунок 5.4 – Вибір функції користувача

5.3. Змінні в мові VBA

Описані в процедурах VBA дії виконуються над змінними або об'єктами. **Змінна** – це поіменована область пам'яті, що використовується для збері-

гання даних протягом роботи процедури. Змінні VBA аналогічні змінними інших мов програмування. Використанню змінної звичайно передує її оголошення за допомогою оператора Dim, у якому вказуються ім'я змінної й тип даних, для зберігання значень якого вона призначена. Хоча такий оператор не є обов'язковим (VBA здатний визначати тип даних змінної за останнім символом її імені: !, #, $, %, &, @), проте, хороший стиль програмування вимагає попереднього опису всіх використовуваних змінних.

Імена змінних можуть мати довжину до 255 символів (припустимі й символи кирилиці, хоча їхнє використання не можна вважати грамотним підходом) і не повинні містити пробілів, розділових знаків і перерахованих вище спеціальних символів (за винятком символу @). Вони не можуть збігатися з ключовими словами VBA або іменами стандартних об'єктів.

Приклади опису змінних:

```
Dim x. y As Integer
Dim Holiday As Date
```

5.3.1. Типи змінних

У процедурах і функціях VBA можуть використовуватися змінні таких **типів**:

– Boolean (логічні змінні, що приймають одне з двох значень: True або False);

– Byte (ціле число від 0 до 255);

– Currency (змінні для грошових обчислень з фіксованою кількістю розрядів після десяткової коми; дозволяють уникнути нагромадження похибок при округленні);

– Date (змінні для зберігання дати й часу);

– Double (змінні, що використаються для зберігання значень із плаваючою точкою подвійної точності);

– Integer (цілі числа від –32 768 до +32 767);

– Long (цілі числа від –2 147 483 648 до +2 147 483 647);

– Object (змінні для зберігання посилань на об'єкти);

– Single (змінні для зберігання значень одинарної точності з плаваючою точкою);

– String (змінні для зберігання рядків символів довжиною від 0 до 64 Кбайт);

– Variant (змінні, тип яких визначається типом останнього привласненого їм значення),

VBA надає можливість користувачеві поряд зі стандартними типами створювати власні типи даних (структури) і зберігати дані в масивах (групах змінних із загальним ім'ям, що розрізняють по комбінації індексів). Типи даних описуються не тільки для змінних, але й для формальних параметрів процедур і функцій, а також для самих функцій (тип значення, що повертається функцією, дивись підрозділ 5.2).

5.3.2. Область видимості змінних

Область видимості змінної – це область програми, де ім'я змінної вважається доступним (видимим) і, таким чином, можливий доступ до її значення.

Існують три рівні видимості змінної:

– процедура (областю видимості є тільки та процедура, у якій змінна оголошена);

– модуль (область видимості – всі процедури модуля, у якому змінна оголошена);

– додаток (область видимості – всі процедури всіх модулів активного додатка).

Для досягнення видимості в межах процедури оператор Dim оголошує змінну в будь-якому місці процедури, але завжди передує операторам, що

використовують її. Час життя такої змінної дорівнює часу виконання процедури, тобто при вході в цю процедуру під змінну виділяється пам'ять і відбувається її ініціалізація. У процесі виконання цієї процедури значення змінної може змінюватися. Після виходу із процедури виділена пам'ять звільняється, отже, втрачається значення змінної.

Якщо необхідно зберегти значення змінної між входженнями до процедури, використовується оператор Static. Дія його аналогічна дії оператора Dim, але він оголошує статичну змінну. Різниця в тім, що при виході з процедури пам'ять статичної змінної не звільняється, а стає тимчасово недоступною, тому значення змінної зберігається.

Для досягнення видимості за межами процедур необхідно, щоб оператор Dim оголошував змінну в розділі описів Declaration модуля (тобто поза його процедурами).

Таким чином, область змінних VBA визначається місцем їхнього опису. Якщо змінна описана всередині процедури або функції, її область видимості обмежується цією процедурою або функцією. Будь-яка спроба використати ім'я даної змінної поза процедурою або функцією, що містить її опис, приведе або до помилки, або до створення нової змінної – з тим же ім'ям, але з іншою областю видимості та, природно, значенням.

Якщо змінну описано на рівні модуля (тобто відповідний їй оператор Dim розміщено безпосередньо в модулі, перед описом процедур і функцій), її можна використовувати в будь-якій процедурі або функції, описаної в цьому модулі. Але якщо передбачається, що змінна буде застосовуватися в процедурах і функціях, описаних у різних модулях проекту, то її варто описати в одному із цих модулів, скориставшись замість оператора Dim співпадаючим з ним за синтаксисом оператором Public.

5.4. Керуючі конструкції мови VBA

Керуючі конструкції мови програмування – це інструкції й групи ін-

струкцій, застосування яких дозволяє змінювати в міру необхідності послідовність виконання інших інструкцій програми. У сучасних мовах програмування керуючі конструкції підрозділяються на розгалуження й цикли. **Розгалуженням** називається керуюча конструкція, що дозволяє пропускати при виконанні ті або інші групи інструкцій залежно від значення умови (If...Else, If...Then...Else, If...Then...ElseIf...Else та Select Case).

Цикл – це керуюча конструкція, що надає можливість багаторазово виконувати групи інструкцій, до настання якої-небудь події.

5.4.1. Розгалуження

Конструкція If...Then. Конструкція If...Then дає VBA вказівку прийняти найпростіше з рішень: якщо умова If істинна, потрібно виконати наступний за ним оператор (або оператори); якщо ж умова неправдива, потрібно перейти до рядка, розташованого безпосередньо за умовною конструкцією. Ця конструкція може розташовуватися як в одній, так і у декількох рядках. Однорядкова конструкція If...Then виглядає так:

If <*умова*> Then <*оператор(и)*>

Якщо умова виконується, VBA обробляє оператор або оператори, які знаходяться за умовою; у противному випадку ці оператори пропускаються.

Конструкція If...Then, що займає кілька рядків (відома також як блок If), виглядає в такий спосіб:

```
If <умова> Then
    <оператор>
    ...
    <оператор>
End If
```

Якщо умова істинна, VBA, знов-таки, виконує оператор або оператори, а в противному випадку передає керування рядку, розташованої під рядком End If.

Конструкція If...Then...Else. Конструкція If ...Then – ідеальний інструмент для виконання однієї дії, що залежить від умови. Але якщо доводиться вибирати один із двох можливих напрямків, у яких будуть виконуватися дії, варто скористатися конструкцією If...Then...Else. За допомогою цієї конструкції вибирається один напрямок діяльності, якщо умова істинна, і інший, якщо вона неправдива.

Синтаксис розглянутої конструкції такий:

```
If <умова> Then
        <оператор>
        ...
        <оператор>
Else
        <оператор>
        ...
        <оператор>
End If
```

Якщо умова істинно, VBA виконує першу групу операторів, якщо умова неправдива – здійснюється перехід до рядка Else, а потім виконується друга група операторів.

Конструкція If...Then...Elself...Else. Остання з числа розглянутих нами конструкцій If – це конструкція If...Then...ElseIf...Else, яку зручно застосовувати для вибору одного з декількох напрямків дій. Ви можете використати скільки завгодно рядків із ключовим словом Elself – їхня кількість буде залежати від складності умови, що перевіряється.

Конструкції If...Then...Elself...Else можуть бути однорядковими і блоковими. Практично завжди блокову конструкцію If...Then...Elself...Else простіше створити, прочитати й налагодити.

Конструкція If...Then..,ElseIf...Else має такий синтаксис:

If <*умова_1*> Then
    <*оператор*>
    ...
    <*оператор*>
Elself <*умова_2*> Then
    <*оператор*>
    ...
    <*оператор*>
Elself <*умова_3*> Then
    <*оператор*>
    ...
    <*оператор*>
...
Else
    <*оператор*>
    ...
    <*оператор*>
End If

Якщо <*умова_1*> істинна, VBA виконує перший блок операторів, а потім переходить до рядка, розташованого після оператора End If. Якщо <*умова_1*> неправдива, здійснюється перехід до першого оператора Elself і перевіряється <*умова_2*>. Якщо вона істинна, VBA виконує другий блок операторів, а потім переходить до рядка, розташованого за оператором End If. У тому

випадку, коли *<умова_2>* неправдива, керування переходить до наступного оператора Elself (якщо такий є) і перевіряється чергова умова і т.д.

Якщо ж у всіх операторах Elself умови неправдиві, VBA переходить до ключового слова Else (при його наявності) і виконує оператори, що розміщено за ним.

Потім оператор End If завершує умовну конструкцію, а керування переходить до рядка, розташованого після оператора End If.

У блоковій конструкції можна використати будь-яке число операторів Elself, забезпечуючи кожен наступний новою умовою. Але якщо застосовувані конструкції If містять велику кількість операторів Elself (скажімо, більше 5), доцільно замість них скористатися конструкцією Select Case, про яку мова йтиме нижче.

Оператор Else є необов'язковим, хоча завжди рекомендується вказувати VBA інший напрямок діяльності, якщо жодна з умов, заданих в операторах If і Elself, не виконується.

Конструкція Select Case. Замість декількох операторів Elself можна застосувати конструкцію Select Case, що дозволяє закодувати ту ж процедуру прийняття рішення у вигляді більш стислої та ефективної програми.

Конструкцію Select Case рекомендується використати в тому випадку, якщо рішення, яке потрібно прийняти в програмі, залежить від однієї змінної або від виразу, що має не менш трьох-чотирьох значень.

Конструкцію Select Case легше читати, чим складні оператори If...Then, головним чином тому, що вона більш компактна. Крім того, у неї легше вносити зміни: коли потрібно змінити одне або кілька використовуваних значень (доводиться переглядати меншу кількість операторів). Синтаксис оператора Select Case такий:

Select Case *<вираз_що_перевіряється>*
    Case *<вираз_1>*
        <оператор>

```
        ...
        <оператор>
    Case <вираз_2>
        <оператор>
        ...
        <оператор>
    ...
    Case Else
        <оператор>
        ...
        <оператор>
End Select
```

На перший погляд, такий синтаксис може здатися складним, але це не так. Оператор Select Case починає конструкцію, а оператор End Select завершує її. Вираз, що перевіряється, визначає, який з операторів Case буде виконано, а <*вираз*_1>, <*вираз*_2> і т.д. – це вирази, з якими VBA порівнює вираз, що перевіряється. Припустімо, необхідно перевірити, кнопку з яким номером вибрав користувач у діалоговому вікні або у формі. У цьому випадку перевіряється вираз, що буде пов'язаний з подією вибору кнопки. Якщо це перша кнопка, значення виразу, що перевіряється, співпаде з «Виразом 1». У такому випадку будуть виконані оператори, що перебувають у рядках, що розташовані за оператором Case <*вираз_1*>. Якщо це друга кнопка, значення виразу, що перевіряється, співпаде з «Виразом 2», отже, будуть виконані оператори, розташовані в рядках, що розміщені за оператором Case <*вираз*_2> і т.д.

Оператор Case Else подібний до ключового слова Else в операторі If. Він є необов'язковим оператором, що, будучи зазначеним, виконується в тих випадках, коли жоден з запропонованих виразів не підходить.

5.4.2. Цикли

При створенні процедур і функцій часто виникають ситуації, коли ті самі дії потрібно повторювати по кілька разів. Для повторного виконання оператора або групи операторів призначені спеціальні керуючі конструкції VBA – цикли. Існує три типи циклів:

– з керуючою умовою;

– з лічильником;

– цикли за структурою даних.

У циклах з керуючою умовою оператор або група операторів повторюється доти, поки таку умову не буде виконано. За назвою оператора, що становить основу цієї керуючої конструкції, такі цикли називаються циклами Do. Залежно від типу керуючої умови та часу її перевірки (щодо виконання повторюваних операторів) розрізняють чотири види циклів Do. Розглянемо синтаксис всіх чотирьох видів циклу Do. Цикл Do першого типу:

```
Do While <умова>
        <оператор>
        ...
        <оператор>
Loop
```

Цикл Do другого типу:

```
Do
        <оператор>
        ...
        <оператор>
Loop While <умова>
```

Цикл Do третього типу:

```
Do Until <умова>
        <оператор>
        ...
        <оператор>
Loop
```

Цикл Do четвертого типу:

```
Do
        <оператор>
        ...
        <оператор>
Loop Until <умова>
```

У циклі Do першого типу умова перевіряється до початку обробки повторюваних операторів. Якщо умова виявляється виконаною (при обчисленні логічного виразу одержано значення True), оператори, записані між ключовими словами Do і Loop (тіло циклу), обробляються однократно, потім керування знову передається ключовому слову Do і процес повторюється. Якщо ж керуюча умова не виконується, то повторювана група операторів пропускається й виконується оператор, що знаходиться безпосередньо за ключовим словом Loop.

У циклі Do другого типу умова перевіряється після виконання тіла циклу. Цикл повторюється, доки керуюча умова виконується.

У циклі Do третього типу умова перевіряється до початку виконання повторюваних операторів, але тіло циклу виконується тільки в тому випадку, коли керуючу умову не виконано (при обчисленні логічного виразу виходить значення False).

У циклі Do четвертого типу умова перевіряється після виконання тіла циклу. Цикл повинен повторюватися доти, поки не буде виконано керуючу умову.

Циклами з керуючим лічильником користуються в тих випадках, коли заздалегідь відомо кількість повторень тіла циклу і перевіряти керуючу умову немає необхідності. Цикли такого типу підрозділяються на дві групи: цикли з лічильником і цикли за структурою даних.

У циклах з лічильником використовується спеціальна змінна – лічильник, значення якої при кожнім повторенні тіла циклу збільшується або зменшується на задану величину – крок циклу. Цикл завершується після того, як значення лічильника досягне (або перевищить) кінцеве значення лічильника циклу. Якщо в процесі виконання циклу значення лічильника убувають (негативний крок циклу), цикл завершується, коли значення лічильника стане рівним або менше кінцевого значення. Синтаксис циклу з лічильником наступний:

For <*лічильник*> = <*початкове_значення*> To <*кінцеве_значення*>
Step <*крок_циклу*>
    <*оператор*>
    ...
    <*оператор*>
Next <*лічильник*>

У циклах за структурою даних тіло циклу по черзі повторюється для всіх однорідних об'єктів, що становлять масив або сімейство. У цьому випадку в ролі лічильника виступає об'єктна змінна. Синтаксис циклу даного типу:

For Each <*елемент*> In <*структура_даних*>
    <*оператор*>
    ...

<оператор>

Next <*елемент*>

5.5. Введення й вивід інформації в мові VBA

Як і у будь-якій мові програмування, в VBA є засоби для обміну інформацією із зовнішнім середовищем – **оператори вводу-виводу**. Докладний опис їх істотно збільшив би обсяг даного навчального посібника. Тому нижче розглянуто лише дві функції, найбільш корисні й широко використовувані при обміні інформацією між програмою й типовими пристроями вводу-виводу (дисплеєм і клавіатурою). Це функція виводу на екран MsgBox і функція введення з клавіатури InputBox.

Функція **MsgBox** має такий вигляд:

MsgBox (*prompt* [, *buttons*] [, *title*] [, *helpfile*, *context*])

Функція виводить повідомлення в діалоговому вікні, очікує натискання клавіші користувачем і повертає ціле число, що відповідає натиснутій клавіші. Всі аргументи, крім *prompt* є необов'язковими. Функція MsgBox повертає значення, що може бути оброблене в програмі.

Зміст операндів такий:

– *prompt*: підказка або інформація для виводу (обов'язковий параметр);

– *buttons*: чисельний вираз, що є сумою значень, що визначають такі характеристики вікна: число й кількість кнопок, стиль значка, кнопку за замовчуванням, модальність вікна. За замовчуванням значення *buttons* дорівнює нулю;

– *title*: заголовок вікна MsgBox;

– *helpfile*: посилання на файл допомоги, організований програмістом;

– *context*: покажчик на розділ у файлі допомоги.

Функція повертає такі значення:

– 1: було натиснуто клавішу OK;

– 2: було натиснуто клавішу Cancel;

– 3: було натиснуто клавішу Abort;

– 4: було натиснуто клавішу Retry;

– 5: було натиснуто клавішу Ignore;

– 6: було натиснуто клавішу Yes;

– 7: було натиснуто клавішу No.

Функція **InputBox** має такий вигляд:

InputBox (*prompt* [, *title*] [, *default*] [, *xpos*] [, *ypos*] [, *helpfile*, *context*])

Функція відображає підказку в діалоговому вікні, очікує введення тексту або натискання кнопки й повертає рядкове значення, що містить введений текст. Всі аргументи, крім *prompt* є необов'язковими.

Зміст операндів такий:

– *prompt*: підказка (обов'язковий параметр);

– *title*: заголовок вікна InputBox;

– *default*: значення, що повертає функція у випадку, якщо інформацію користувачем не введено;

– *xpos*: відстань (в одиницях 1/20 пікселя) лівого краю діалогового вікна від лівої границі екрана. Якщо аргумент відсутній – вікно центрується по горизонталі;

– *ypos*: відстань (в одиницях 1/20 пікселя) верхнього краю діалогового вікна від верхньої границі екрана. Якщо аргумент відсутній – вікно центрується по вертикалі;

– *helpfile*: посилання на файл допомоги, організований програмістом;

– *context*: покажчик на розділ у файлі допомоги.

Питання й тести для самоконтролю

1. Навіщо потрібні мови програмування при розробці інформаційних систем?
2. Опишіть загальні властивості мови VBA.
3. Які Ви знаєте об'єкти мови VBA?
4. Зобразіть і поясніть структуру програми мовою VBA.
5. Розкажіть про процедури мови VBA.
6. Що Ви знаєте про функції мови VBA?
7. Які в мові VBA існують типи змінних і що таке область видимості?
8. Розкажіть про оператори розгалуження в мові VBA.
9. Які Ви знаєте оператори циклу в VBA?
10. Опишіть найпростіші функції введення-виводу мови VBA.

Розділ 6. СТВОРЕННЯ БАНКІВ ДАНИХ ЗА ДОПОМОГОЮ СУЧАСНОЇ РЕЛЯЦІЙНОЇ СУБД

Після ознайомлення з матеріалом розділу 3, уважний читач міг припустити, що, оскільки реляційна модель бази даних є найбільш гнучкою й універсальною, то саме вона повинна полягати в основу більшості сучасних систем управління базами даних. Це дійсно так. Практично всі скільки-небудь складні бази даних у наш час організуються з використанням реляційних моделей. Через важливість даного питання, варто звернути особливу увагу на теоретичну підготовку в зазначеній області.

На сьогоднішній день є безліч різноманітних книг по роботі в середовищі конкретних СУБД: від книг, орієнтованих на починаючих офісних працівників, до книг для професійних програмістів. У процесі навчання можна використати будь-які з них як додаткову літературу. Але особливо хотілося б звернути увагу на активне використання різноманітної допомоги, включеної в самі СУБД: це й контекстна допомога, і довідкова система, і приклади конкретних проектних рішень, включені в поставку СУБД. Прийоми, використовувані при побудові баз даних, є універсальними і мало залежать від предметної області.

Як приклад, розглянемо СУБД **MS Access**, що входить до складу відомої програмної системи MS Office. Перевагами її є простота використання (завдяки типовому для фірми Microsoft розвиненому візуальному інтерфейсу), велика колекція файлів допомоги, а також невимогливість до фізичних ресурсів ЕОМ, що дозволяє встановити дану СУБД на домашньому комп'ютері середнього рівня.

Відповідно до описаної раніше класифікації, система MS Access відноситься до класу так званих «настільних» СУБД, які мають високорозвинені мовні засоби, призначені для полегшення роботи з ними користувачів різної кваліфікації, у тому числі й користувачів, що не є фахівцями в області інформаційних технологій.

Зазначимо, що в СУБД MS Access, як взагалі у кожної скільки-небудь розвинутої програмної системи, є своя специфічна термінологія. Наприклад, базою даних у ній називається сукупність таблиць, форм, звітів, запитів, модулів, макросів. Вся ця сукупність запам'ятовується в одному файлі бази даних.

6.1. Створення бази даних в MS Access

Для створення нової бази даних необхідно після запуску системи MS Access вибрати позиції меню File – New і у вертикальному вікні, що з'явиться в правій частині екрана, «New File» вибрати позицію «Blank Database». У вікні, що з'явилося, «File New Database» (рис. 6.1) треба задати ім'я створюваного файлу БД і визначити місце, де він буде зберігатися, після чого натиснути кнопку «Create». У нашому прикладі для файлу бази даних задано ім'я «Parts» і він буде зберігається в папці «My Documents». Після натискання кнопки «Create» файл БД Parts.mdb буде створено на диску (.mdb – розширення імені файлу БД системи MS Access), а на екрані з'явиться нове вікно – вікно **об'єктів** (рис. 6.2). Таким чином, на відміну від інших програмних систем, що входять до складу MS Office, система MS Access створює файл на диску не наприкінці, а на початку роботи.

6.2. Створення таблиць

Створення бази даних починається зі створення **таблиць MS Access**, у яких зберігається інформація про предметну область. База даних звичайно включає кілька взаємозалежних таблиць. Для створення нової таблиці у вікні об'єктів треба вибрати закладку «Tables» (рис. 6.2) і натиснути бажаний пункт меню, у результаті чого з'явиться відповідне вибору вікно для створення нової таблиці.

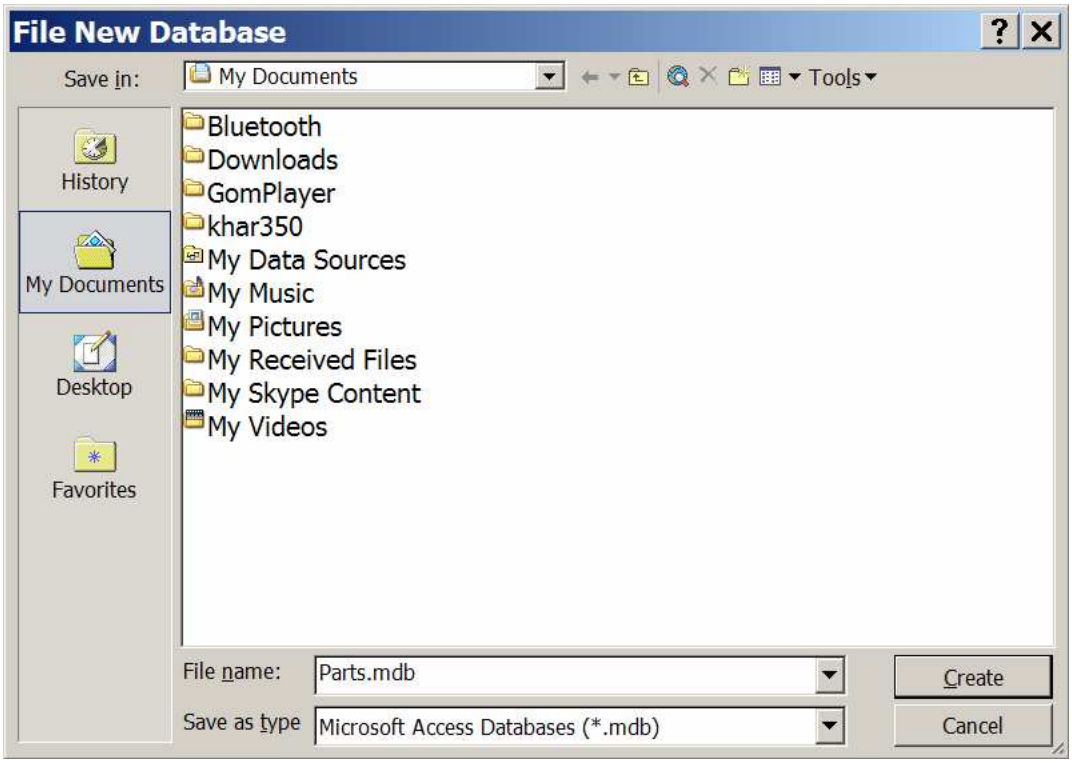


Рисунок 6.1 – Вікно нової бази даних

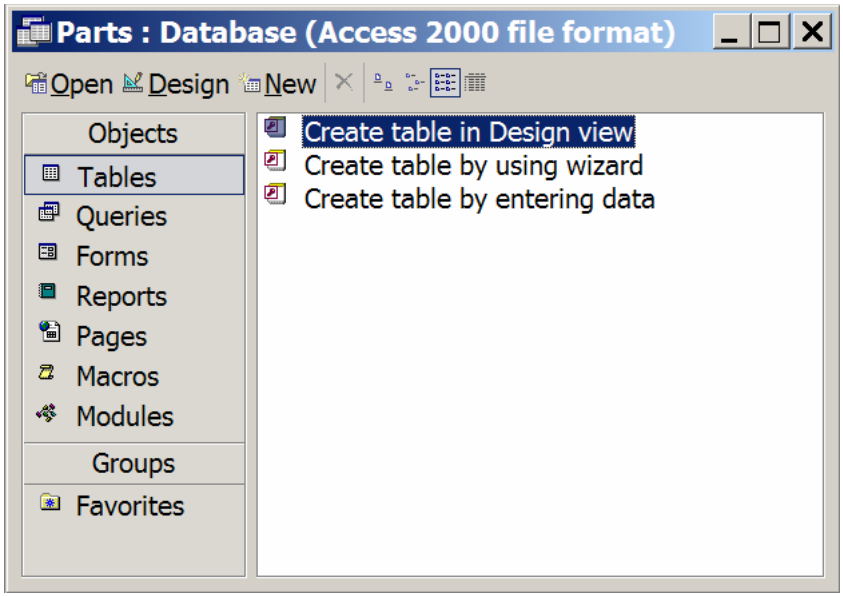


Рисунок 6.2 – Вікно об'єктів

Створити таблицю можна в різних режимах:

– Конструктора таблиць;

– Майстра таблиць;

– введення даних.

Крім того, можливий імпорт таблиць і зв'язок з таблицями. Почнемо розгляд можливостей створення таблиць з режиму **Конструктора таблиць** (Design view), як найбільш гнучкого й часто використовуваного. Для цього у вікні об'єктів виберіть режим створення нового об'єкта – «Create table in Design view». Після чого з'явиться вікно для опису структури таблиці й інших її характеристик (рис. 6.3).

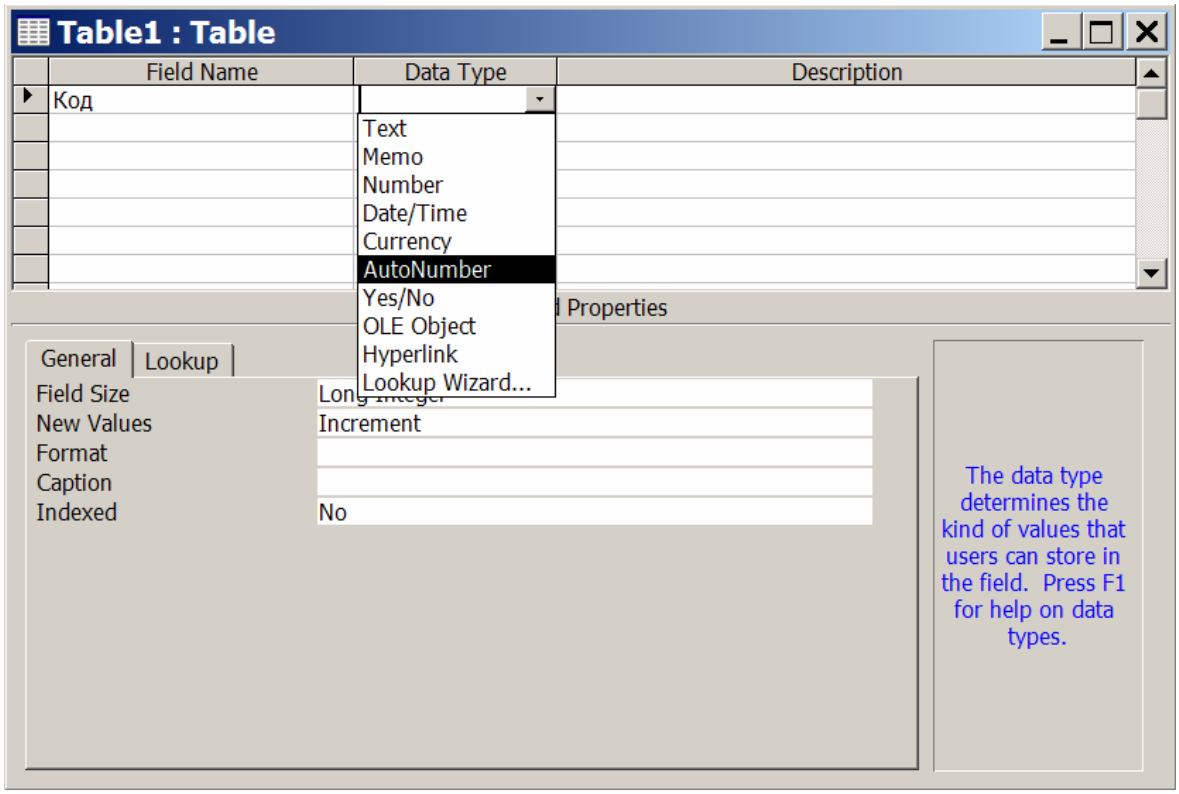


Рисунок 6.3 – Вікно нової таблиці в режимі Конструктора

6.2.1. Опис характеристик таблиці

У табличній формі треба послідовно описати всі **поля** створюваної таблиці. Спочатку задається ім'я поля (Field Name). MS Access допускає задавання довгих імен із пробілами російською мовою. Діють наступні обмеження на імена полів:

– ім'я не повинне містити більше 64 символів;

– ім'я може включати будь-яку комбінацію букв, цифр, пробілів і спеціальних символів за винятком крапки, знака оклику, апострофа і квадратних дужок;

– ім'я не повинне починатися із символу пробілу;

– ім'я не повинне включати керуючі символи (з кодами ASCII від 0 до 31 у десятковій системі).

Хоча пробіли усередині імен полів і є припустимими, вони можуть при деяких обставинах викликати конфлікти у роботі з іншими системами. Тому їх не рекомендується використовувати (у даному навчальному посібнику довгі імена із пробілами даються винятково з метою досягнення більшої наочності матеріалу, що викладається).

Взагалі до задавання довгих імен російською мовою треба ставитися з обережністю, особливо, якщо є ймовірність, що створюваний додаток буде надалі використовуватися в розподілених системах. При задаванні імен не допускайте їхнього збігу із зарезервованими словами. Наприклад, не слід давати полю ім'я Count, Name і т.п. Ім'я поля повинно бути унікальним у межах таблиці. І хоча система не забороняє використання однакових імен полів у різних таблицях, уникайте використання однакових імен для позначення різних за змістом атрибутів.

Ім'я повинно бути зрозумілим не тільки в контексті даної конкретної таблиці. Так, наприклад, якщо в таблиці «Accounts» є поле «Код», і таке ж поле є в таблиці «Sellers», то в першому випадку це буде код рахунків, а в другому – код продавців. Багато систем (і MS Access у тому числі) автоматично зв'язують таблиці по полях, які мають однакові ім'я, тип і довжину. Якщо імена даних не продумано, можуть або виникнути неправильні зв'язки, або процес задавання зв'язків буде трохи складнішим, ніж при правильному завданні імен.

Після задавання імені треба вибрати тип даних. Якщо клацнути мишкою по вільному осередку графи «Data Type», висвітиться список припустимих типів даних (див. рис. 6.3), з якого й варто вибрати підходящий для опи-

суваного поля тип. Ім'я й тип даних повинні задаватися обов'язково. Графа «Description» може не заповнюватися. Ця графа використовується з метою документування проекту. **Можливі типи даних** в MS Access та їхня коротка характеристика наведені в таблиці 6.1.

Таблиця 6.1 – Можливі типи даних

Тип даних	Вміст поля	Розмір
Текстовий (Text)	Текст або числа, що не вимагають проведення розрахунків, наприклад, номери телефонів, коди і т.п.	Максимальне число символів – 255
Memo (Memo)	Довгий текст або комбінація тексту й чисел	До 65535 символів
Числовий (Number)	Числові дані, використовувані для проведення розрахунків	1, 2, 4 або 8 байт
Дата/час (Date/Time)	Дати й час, що відносяться до років з 100 по 9999 включно	8 байт
Грошовий (Currency)	Спеціальний формат для кодування числових даних. Точність – до 15 знаків у цілій і до 4 знаків у дробовій частині	8 байт
Лічильник (AutoNumber)	Унікальні послідовно зростаючі (на 1) або випадкові числа, що автоматично вводяться при додаванні кожного нового запису в таблицю	4 байт
Логічний (Yes/No)	Поля, які можуть містити одне із двох можливих значень (True/False, Так/Ні)	1 біт
Об'єкт OLE (OLE Object)	Об'єкт, зв'язаний або впроваджений у таблицю MS Access	До 1 Гбайт (обмежується обсягом диска)
Гіперпосилання (Hyperlink)	Рядок, що складається з букв і цифр, і вміщує адресу гіперпосилання. Адреса гіперпосилання може складатися максимум із трьох частин: 1. Текст – текст, виведений у поле або в елементі керування; 2. Адреса – шлях до файлу (у форматі шляху UNC) або сторінки (адреса URL); 3. Додаткова адреса – зсув усередині файлу або сторінки	Кожна із трьох частин у типі Гіперпосилання може містити до 2048 символів

Вибір типу даних є важливим кроком при проектуванні БД. Ухвалене рішення впливає на виконуваний при введенні контроль правильності даних, на припустимі дії над даними та особливості їхнього виконання, необхідний обсяг пам'яті, швидкість роботи, сумісність різних частин БД. Припустимо, що ми створюємо таблицю «Accounts», що містить відомості про продажі комплектуючих для комп'ютерів. Склад і тип полів створюваної таблиці показані на рис. 6.4.

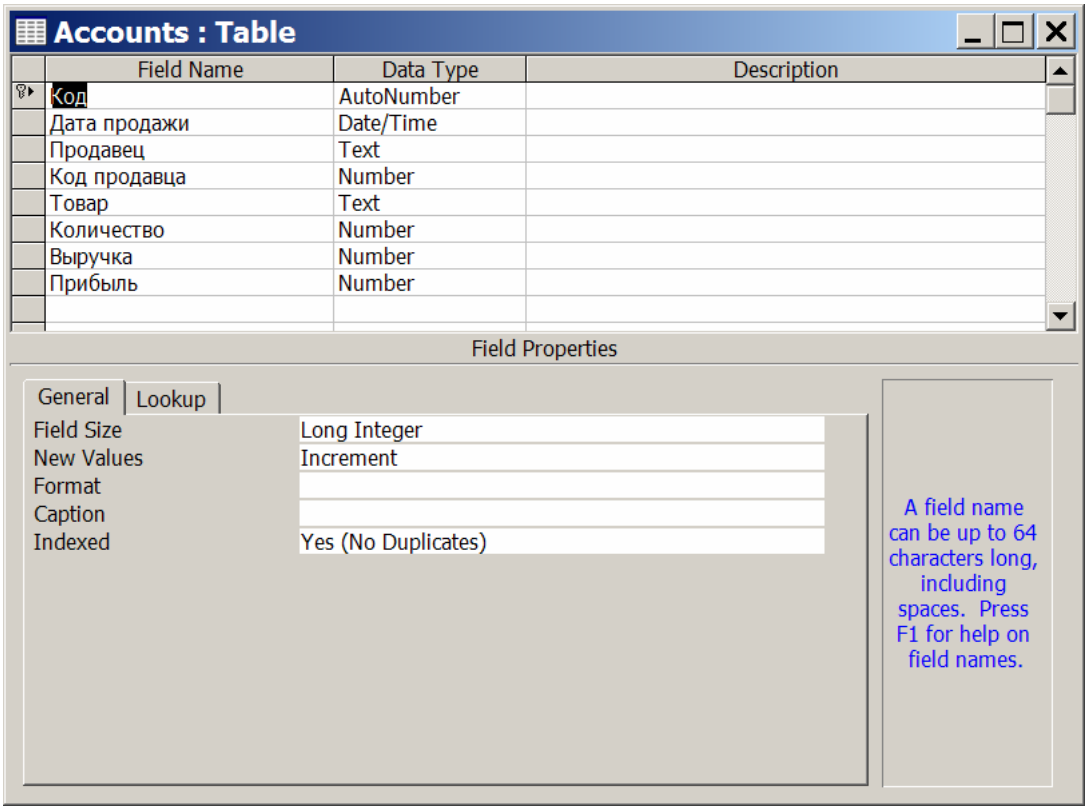


Рисунок 6.4 – Таблиця облікових записів про продажі

6.2.2. Задавання ключа

Кожна реляційна таблиця по визначенню має ключ запису. MS Access дозволяє задавати ключ при опису таблиці, але також дозволяє й відмовитися від цієї можливості. По ключу система автоматично виконує індексування, а також перевіряє унікальність значень ключа при введенні нових записів або їхньому коректуванні.

Якщо Ви бажаєте як ключ запису вибрати код, що задається автоматично системою, (тобто поле типу «лічильник»), то можна це поле спочатку не описувати, а підтвердити необхідність його створення при завершенні опису таблиці. MS Access створить це поле автоматично.

Якщо ви визначаєте ключ самостійно, то це можна зробити декількома шляхами: позиціюватися за допомогою миші на відповідне поле й натиснути кнопку «Primary Key» у головному меню (із зображенням ключа); або вибрати позицію меню Edit – Primary Key; або скористатися правою кнопкою миші для виклику контекстного меню, попередньо також позиціювавшись на те поле, що визначається як ключове.

Для зручності подальшої роботи з таблицею «Accounts» у ній задано ключове поле «Код» як показано на рис. 6.4.

Ключ може бути також складовим. Щоб визначити складовий ключ, треба виділити за допомогою миші відповідну сукупність полів, а потім виконати ті ж дії, що й при задаванні простого ключа.

6.2.3. Характеристики полів

У нижній частині екрана опису таблиці (рис. 6.4) відображається список властивостей обраного поля. Перелік цих властивостей змінний і залежить від обраного типу поля. Деякі з властивостей полів очевидні без додаткових пояснень. Деякі будуть зрозумілі з прикладів, інші властивості можуть бути з'ясовані зі спеціальної літератури. Властивість «Indexed» визначає, чи треба створювати індекс по цьому полю. Індекс прискорює виконання запитів, у яких використовуються індексовані поля, полегшує сортування та групування записів.

Властивість «Indexed» може мати три значення:

– «No» (значення за замовчуванням, індекс не створюється);

– «Yes, Duplicates OK» (в індексі допускаються повторювані значення);

– «Yes, No Duplicates» (в індексі не допускаються повторювані значення).

Наприклад, у зв'язку з тим, що пошук та упорядкування інформації часто здійснюється по полю «Продавець», бажано по ньому зробити індексацію. Тому що серед співробітників можливі однофамільці, повинні бути дозволені збіги значень індексованого поля. Не допускається створення індексів для полів Memo, гіперпосилань і об'єктів OLE.

Для полів типу «Лічильник» існує властивість «New Values». Вона визначає спосіб збільшення значення поля лічильника при додаванні в таблицю нових записів. Властивість «New Values» може мати такі значення:

– «Increment» – значення поля лічильника збільшується на 1 у кожному новому записі;

– «Random» – поле лічильника в новому записі отримує випадкове значення типу Long Integer.

Слід зазначити, що багато СУБД для полів такого типу дозволяють використовувати довільний крок збільшення.

Властивість «Allow Zero Length» визначає, чи допускається введення в дане поле порожніх рядків (рядків, що не містять символів). Властивість «Allow Zero Length» може мати такі значення:

– «Yes» (порожні рядки є припустимими значеннями);

– «No» (порожні рядки не є припустимими значеннями);

При задаванні значення «Yes» для властивості «Allow Zero Length» MS Access розрізняє неіснуючі дані (зберігаються у вигляді порожніх рядків) і дані, які існують, але не відомі [зберігаються у вигляді порожніх (Null) значень].

Часто для якогось поля існує обмежений перелік можливих варіантів даних (наприклад, для посади співробітника це можуть бути: продавець, продавець-консультант, старший продавець, старший менеджер, завідувач відділом, директор магазина і т.п.). У такому випадку доцільно було б замінити введення цих значень вибором їх зі списку. У сучасних версіях MS Access

для цієї мети слугує так званий «Майстер підстановок» («Lookup Wizard»). У даному навчальному посібнику ця можливість не розглядається, тому що вона докладно описана в літературі [1].

6.2.4. Збереження характеристик таблиці

Після того, як опис характеристик таблиці завершено, його треба зберегти. Це можна зробити різними шляхами: вибрати меню «File – Save» або «File – Close» (після чого на питання «Do you want to save changes to the design of table 'XXX'?» відповісти «Yes») або клацнути кнопку «View» головного меню й вибрати «Datasheet View» і на повідомлення «You must first save the table. Do you want to save the table now?» відповісти «Yes» (цей спосіб треба використовувати тоді, коли Ви хочете відразу після опису структури таблиці вводити дані в цю таблицю). У віконці, що з'явиться після зазначених дій, слід ввести ім'я створеної таблиці.

6.2.5. Приклад створення таблиці

Як перший приклад розглянемо створення таблиці облікових записів про продажі комп'ютерних комплектуючих «Accounts», зображеної на рис. 6.4. Тут, крім розглянутого вище ключового поля «Код», присутні поля «Дата продажу» з типом даних «дата/час»; «Продавець» і «Товар» з текстовим типом; «Код продавця», «Кількість», «Виторг» та «Прибуток» (всі – числового типу).

Другий приклад – таблиця облікових записів про продавців «Sellers», зображена на рис. 6.5. Ключове поле тут має ім'я «Код продавця». Інші поля називаються: «Прізвище» і «Посада» з текстовим типом; «Дата народження» з типом даних «дата/час»; «Стать» з логічним типом; «Оклад» (числового типу).

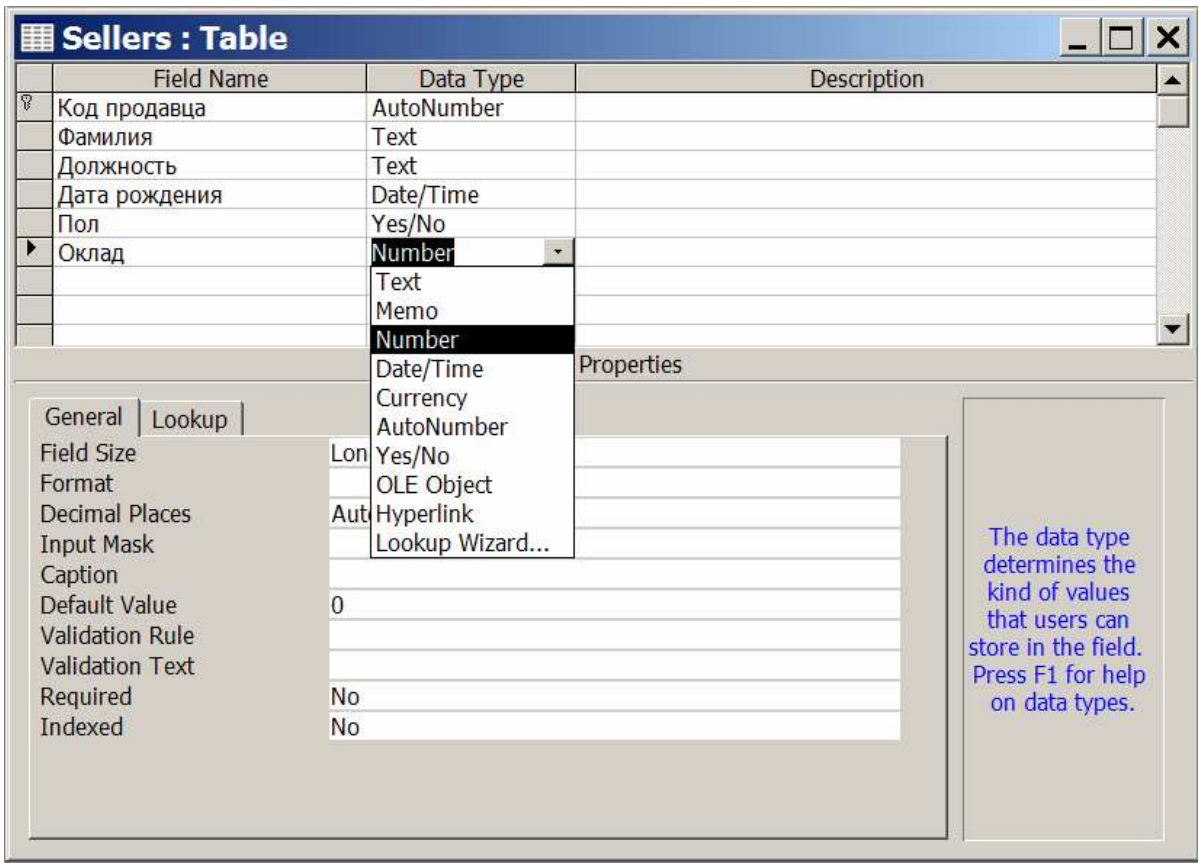


Рисунок 6.5 – Таблиця облікових записів про продавців

При створенні таблиць варто пам'ятати, що в реляційних базах даних зв'язування таблиць відбувається за значеннями відповідних полів зв'язку. Ці поля повинні відповідати один одному за типом і довжиною. Якщо в основній таблиці ключове поле має тип «Лічильник», то в підлеглій таблиці відповідне поле зв'язку повинно мати тип «Числове» і розмір поля – «Довге ціле».

6.2.6. Коректування структури таблиці

У випадку помилки при описі структури таблиці або, якщо по яких-небудь інших причинах Ви хочете змінити її, це можна зробити досить просто. Якщо Ви вже вийшли із процесу створення таблиці, але ще продовжуєте працювати з нею, то можна перейти назад у режим «Design view», скориставшись кнопкою «View». Якщо потрібна таблиця взагалі закрита, то її можна

відкрити в режимі «Design view» й у такий спосіб повернутися у вікно опису таблиці.

Для додавання поля в таблицю виберіть рядок, над яким потрібно додати нове поле, і натисніть кнопку «Insert Rows» на панелі інструментів, або просто натисніть клавішу «Ins».

При роботі з MS Access (як, втім, і з іншими Windows-орієнтованими системами) рекомендується активно користуватися правою кнопкою миші для виклику контекстного меню. Часто це буває найпростішим шляхом вибору потрібної дії. Для додавання поля в кінець таблиці виберіть перший порожній рядок і вводите в нього опис чергового поля. Якщо таблиця вже містить дані, то до зміни типів даних і розмірів полів рекомендується зробити її копію, тому що несумісність існуючих даних з новим значенням властивості «Data Type» може призвести до втрати даних.

6.2.7. Альтернативні способи задавання структури таблиць

Якщо Ви створюєте таблицю, структура якої має багато спільного зі структурою раніше створеної таблиці, то можна скопіювати структуру існуючої таблиці (для цього треба позиціюватися на відповідній таблиці, вибрати позицію меню «Edit – Copy», потім – «Edit – Paste», після чого у вікні, що з'явилося (рис. 6.6) ввести ім'я знову створюваної таблиці, а як параметр вставки вибрати «Structure Only»). Структура створеної в такий спосіб таблиці може бути згодом скоректована звичайним способом. У наведеному прикладі в базі даних торгового закладу на основі таблиці «Accounts» будується таблиця «Sales».

Крім того, створити таблицю можна з використанням **«Майстра таблиць»** («Wizard»). У лівій частині вікна «Table Wizard» висвічується перелік зразків таблиць, з яких Ви можете вибрати підходящу по змісту таблицю (рис. 6.7).

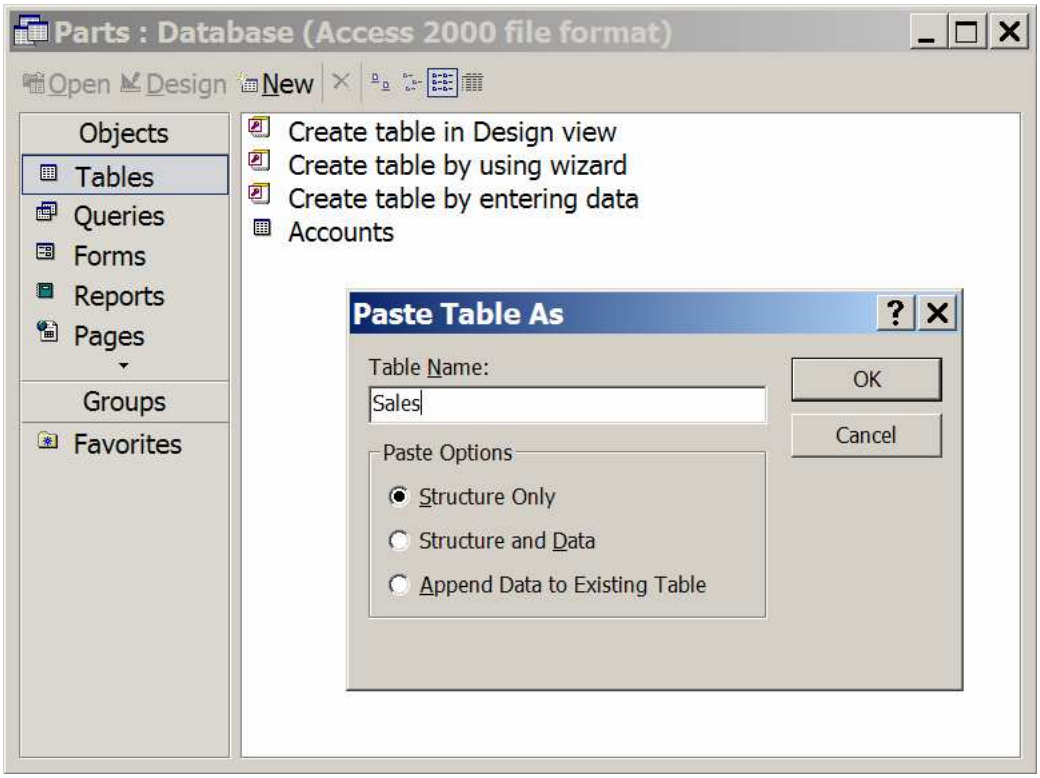


Рисунок 6.6 – Створення таблиці копіюванням структури

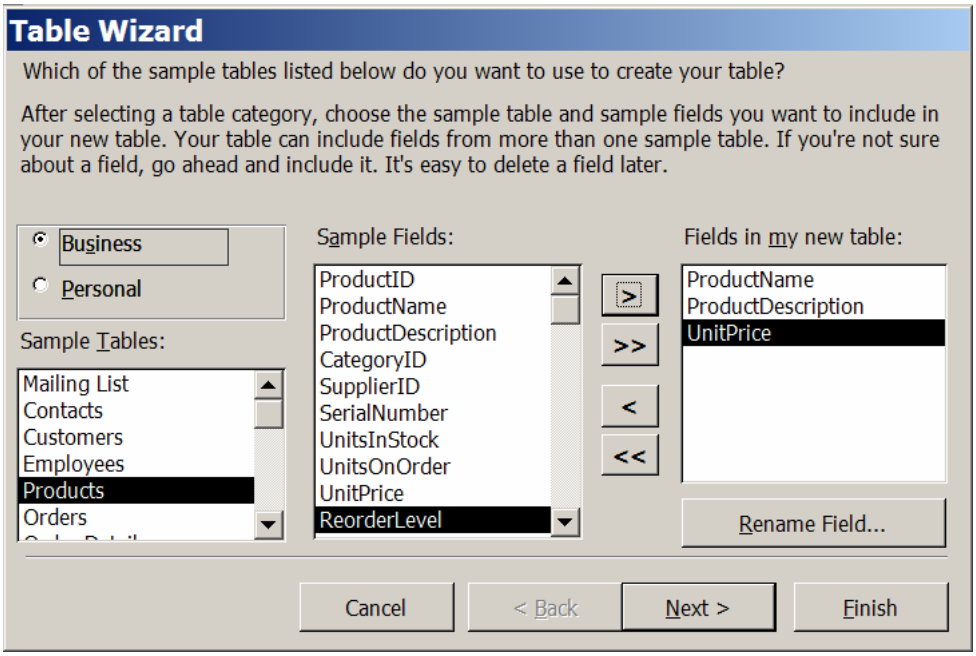


Рисунок 6.7 – Створення таблиці за допомогою Майстра таблиць

З обраної таблиці Ви можете перенести всі або деякі поля в знову створювану таблицю, можете змінити ім'я поля. Щоб ввести які-небудь інші зміни в структуру створюваної таблиці, варто завершити формування таблиці за допомогою Майстра, після чого відкоригувати структуру у звичайному порядку (у режимі Конструктора). Як ми бачимо, використання цієї можливості не звільняє від розуміння основ проектування БД, тому варто уважно оцінити, наскільки пропоноване як зразок рішення відповідає вашим потребам, і, при необхідності, змінити пропоновану структуру БД.

Створити таблицю можна й шляхом імпорту її з інших систем. Крім того, у вигляді таблиці можна зберегти результат запиту. В MS Access є ще можливість створювати таблицю в режимі введення даних («Entering Data»), але ця можливість не завжди виправдана, оскільки в цьому випадку тип даних кожного поля вибирається автоматично, виходячи з введеної в це поле інформації, і можливі важко виявлювані помилки.

6.3. Зв'язування таблиць

Після створення таблиць можна, при необхідності, визначити їхню зв'язаність. Для цього треба вибрати позицію головного меню «Tools – Relationships» (або натиснути відповідну кнопку на панелі інструментів). Далі, у вікно, що відкрилося («Relationships») треба додати ті таблиці, між якими буде визначатися зв'язок.

Таблиці, між якими визначається зв'язок, найчастіше зв'язані відношенням 1:*M* (див. підрозділ 9.4). Для встановлення зв'язку треба позиціюватися на поле зв'язку (звичайно це первинний ключ) в основній таблиці (тій, котра знаходиться на стороні «1»), не відпускаючи ліву клавішу миші, перетягнути значок, що з'явився, на відповідне поле в «залежному» файлі й відпустити клавішу миші. Після цього на екрані з'явиться вікно «Edit Relationships» (рис. 6.8).

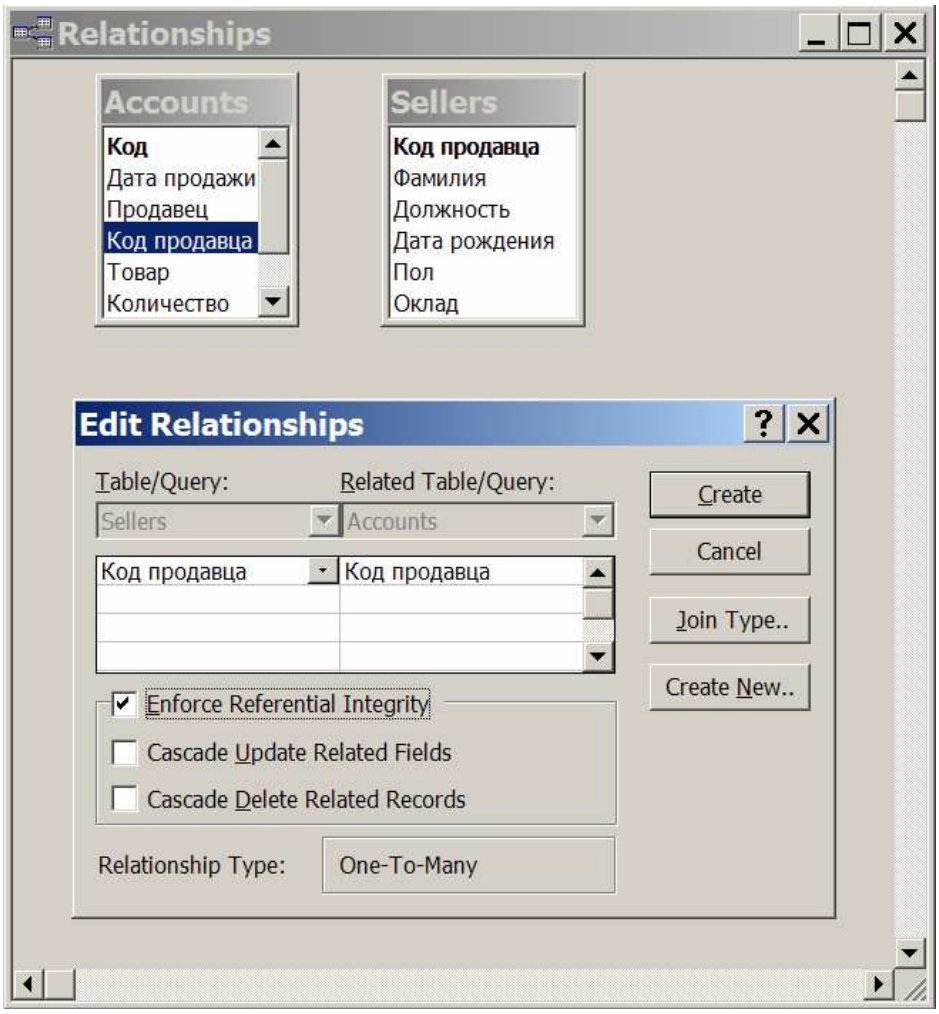


Рисунок 6.8 – Зв'язування таблиць по полю «Код продавця»

Далі варто визначити, чи треба задавати обмеження цілісності зв'язку, і якщо так, то вибрати режими коректувань (відновлення й видалення). Якщо ви задаєте обмеження цілісності, то поле зв'язку основного запису повинно бути проіндексоване. У розглянутому нами як приклад випадку при зв'язуванні таблиць «Accounts» і «Sellers» обмеження цілісності варто задати, щоб у таблиці «Accounts» не з'являлися коди продавців, які відсутні в таблиці «Sellers».

Задавати «Каскадне відновлення зв'язаних полів» («Cascade Update Related Fields») у цьому випадку не має сенсу. Каскадне відновлення означає, що при зміні первинного ключа в основній таблиці, відповідні поля у зв'язаній таблиці автоматично змінюються. У таблиці «Sellers» поле «Код продавця» має тип «Лічильник». Це означає, що змінювати значення цього поля не

можна. Задавати каскадне видалення в цьому випадку теж небезпечно, тому що у випадку ліквідації таблиці «Accounts» виявляться вилученими всі записи співробітників, що працювали в цьому магазині.

Забезпечити посилальну цілісність можна й іншим способом – використовуючи поле підстановки: якщо значення будуть переноситися зі зв'язаної таблиці, то в підлеглій не може з'явитися значення, відсутнє в основній таблиці. Існують поняття внутрішнього, лівого й правого з'єднання.

У вікні «Relationships», що з'являється при встановленні зв'язку між двома таблицями, є кнопка «Join Type» («Тип об'єднання»), натиснувши на яку, користувач попадає у вікно «Join Properties» («Параметри об'єднання»). У ньому він може вибрати один із трьох варіантів:

– об'єднання тільки тих записів, у яких значення зв'язаних полів обох таблиць збігаються;

– об'єднання всіх записів першої таблиці й тільки тих записів із другої таблиці, у яких значення зв'язаних полів обох таблиць збігаються;

– об'єднання всіх записів другої таблиці й тільки тих записів з першої таблиці, у яких значення зв'язаних полів обох таблиць збігаються.

Перша з перерахованих альтернатив означає внутрішнє, друга – ліве, третя – праве з'єднання.

6.4. Задавання обмежень цілісності

Забезпечення цілісності бази даних є одним з найважливіших завдань при створенні банку даних, тому що забезпечення адекватності бази даних відображуваної предметної області є одною з основних вимог, встановлених до БНД. При викладі питань створення й зв'язування таблиць ми вже торкалися деяких аспектів забезпечення цілісності БД. Розглянемо інші можливості задавання **обмежень цілісності**. В MS Access багато обмежень цілісності можуть задаватися при створенні таблиці.

6.4.1. Тип даних

Тип даних визначає припустимі символи, які можуть бути використані при заповненні відповідного поля. Для деяких типів даних, наприклад, поля типу «дата», здійснюється й більш складна перевірка. Якщо допущено помилку в типі даних або неправильно введена дата, користувач повинен обов'язково виправити помилку, тому що СУБД не дає інших можливостей продовжити роботу.

Багато які з властивостей полів також дозволяють забезпечувати контроль цілісності. Такі властивості полів, як:

– розмір поля;

– формат поля;

– маска введення;

– значення за замовчуванням;

– умови на значення;

– повідомлення про помилку;

– обов'язкове поле;

– порожні рядки;

– індексоване поле,

в тій або іншій мірі пов'язані з обмеженнями цілісності.

Пояснимо використання деяких з перерахованих вище властивостей з метою забезпечення контролю цілісності на окремих прикладах.

6.4.2. Розмір поля

У поле не можна ввести більше символів, чим це зафіксовано у властивості «розмір поля» або визначено типом поля.

6.4.3. Умови на значення

Однією з найбільш гнучких можливостей визначення обмежень цілісності є задавання «Умови на значення». Умови вводяться як вираз. Вирази можуть бути простими або складними. Використовуючи їх можна задавати й діапазони. Наприклад, умова >#1.89#, задана як «Умова на значення» для поля «Дата народження», буде означати, що допустимо введення дат тільки після 1989 року [значення дати необхідно обмежити з обох боків символами номера (#)]. Таке обмеження цілісності може бути використано, наприклад, у випадку, якщо організація, для якої ведеться БД, не має права приймати на роботу співробітників молодших певного віку. При задаванні даного обмеження цілісності введення значення в поле буде обов'язковим (навіть якщо у властивості поля «Умова на значення» зафіксовано – «ні»). Умови на значення можуть задаватися для полів або записів.

Вирази, що визначають умови на значення, не повинні містити функції, обумовлені користувачем, статистичні функції або функції підмножини, функції CurrentUser або Eval, а також посилання на форми, запити й таблиці. Крім того, вираз, зазначений як умова для поля, не повинен містити посилання на інші поля. Вираз, зазначений як умова на значення для запису, може містити посилання на поля тієї ж таблиці. Умови на значення для записів задаються у вікні властивостей таблиці, що відкривається командою «Properties» меню «View» у режимі конструктора таблиці.

Якщо користувач задає значення властивості «Умова на значення», але не визначає властивість «Повідомлення про помилку», то при порушенні умови на значення MS Access виводить стандартне повідомлення про помилку. Якщо значення властивості «Повідомлення про помилку» задано (рис. 6.9), то в повідомленні про помилку виводиться текст, зазначений як значення цієї властивості.

В MS Access немає спеціального способу перевірки правильності задавання перелічимих типів. Як було показано вище, цієї мети можна досягти,

використовуючи «Майстер підстановки». Крім того, перевірку можна виконати й шляхом задавання відповідного виразу для властивості «Умови на значення». Наприклад, для поля «Посада» у БД співробітників магазина можна задати умова «продавець» Or (або) «продавець-консультант» Or (або) «старший продавець» і т.д.

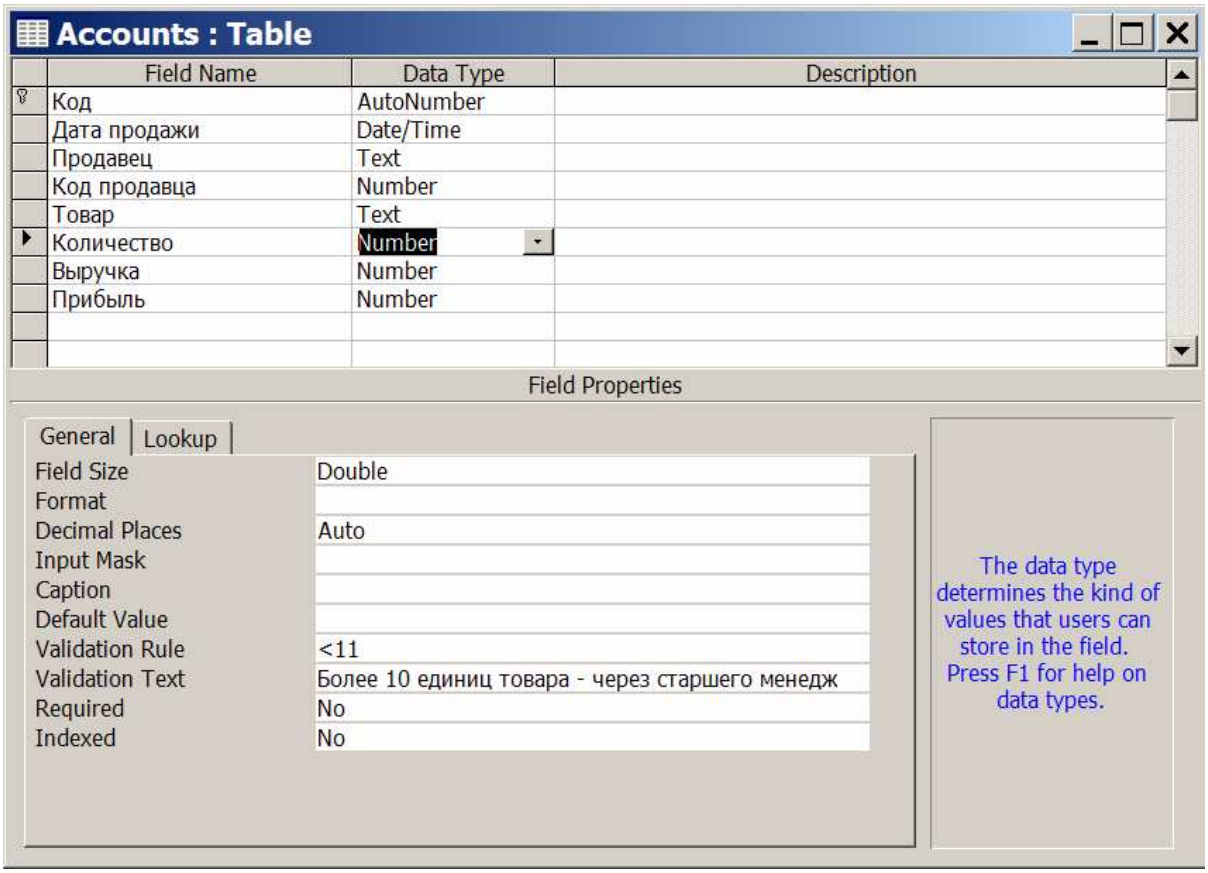


Рисунок 6.9 – Задавання умови на значення

Зазначимо ще, що MS Access автоматично (тобто без спеціальних на те вказівок) накладає умови на значення, обумовлені типом даного поля, наприклад, не допускається введення тексту в числові поля.

6.4.4. Маска введення

Припустимо, у таблицю вводяться імена співробітників. Для відповідного поля можна задати маску введення, що дозволить використати тільки

букви при введенні, забезпечить перетворення першого символу у верхній регістр, всіх інших – у нижній, і допустить використання не менше двох букв у кожнім слові (вважається, що імен і прізвищ, що складаються з однієї букви, немає).

В MS Access така маска введення буде виглядати в такий спосіб: *>L<L*. Символ «*L*» у масці означає, що в дану позицію повинна бути введена буква, символ «?» означає, що в дану позицію може бути введена буква. Символ «>» перетворить всі символи, розташовані правіше цього знака, у верхній регістр, символ «<»перетворить всі символи, розташовані правіше цього знака, у нижній регістр.

Всі символи, які можуть бути використані в масках, і їхнє призначення можна знайти в «Довідковій системі» (Help) MS Access. Використання подібних масок введення не тільки забезпечує контроль використання припустимих символів, але й полегшує процес введення даних.

6.4.5. Індексоване поле

Індексоване поле можна використати для контролю на унікальність. В MS Access, як і у багатьох інших системах, при визначенні для індексованого поля значення властивості «унікальний індекс», у це поле не допускається введення повторюваних значень. У тих СУБД, які підтримують концепцію ключа (у тому числі й в MS Access), після того як у таблиці визначається ключ, по цьому полю виконується індексування й забороняється введення повторюваних або порожніх значень ключа.

Як відзначалося вище, при задаванні зв'язку між таблицями в MS Access можна встановити прапорець «Підтримка цілісності даних», і в цьому випадку система буде автоматично підтримувати обмеження цілісності зв'язку. При видаленні основного запису, пов'язаного з декількома підлеглими, можуть бути обрані різні стратегії відновлення:

– заборонити видаляти основний запис, якщо є підлеглі;

– видалити разом з основним записом і всі підлеглі (каскадне видалення).

В MS Access, наприклад, при підтримці цілісності зв'язку автоматично приймається перша стратегія. Щоб скасувати її, для даного зв'язку треба встановити прапорець «Каскадне видалення зв'язаних записів». Для задавання складних умов можна використати макроси або модулі.

6.5. Введення даних у базу

Після того як завершено проектування структури бази даних, БД описана, можна приступати до введення даних. Це можна зробити як відразу по закінченні опису структури таблиці, так і потім.

6.5.1. Введення даних у режимі «Таблиця»

Як відзначалося вище, щоб відразу після опису структури таблиці вводити дані в цю таблицю, треба натиснути кнопку «View» і вибрати «Properties». Після збереження опису таблиці, вона висвічується на екрані в табличному вигляді (перший рядок цієї таблиці містить імена полів таблиці, друга – порожня, у яку й вводяться дані). Для того щоб потрапити в режим «Таблиця» для введення даних у вже існуючій таблиці, треба у вікні бази даних на вкладці «Tables» позиціюватися на рядку, що відповідає назві необхідної таблиці, і натиснути кнопку «Open».

Кожна таблиця містить порожній запис, який йде за останнім існуючим записом і призначений для введення нових даних (цей запис відзначений ліворуч символом «зірочка»). Позиціюватися на цей запис можна різними способами, наприклад, натиснувши відповідну кнопку в інструментальному меню або просто мишею. Після чого слід ввести необхідні дані з клавіатури.

В MS Access для раціоналізації процесу введення даних у БД можна використати властивість поля «Значення за замовчуванням». Властивість

«Значення за замовчуванням» дозволяє вказати значення, що буде автоматично вводиться в поле при створенні нового запису. Як значення за замовчуванням найчастіше вибирається те значення, що найчастіше зустрічається в записах БД. Наприклад, для значення поля «Посада» у таблиці, що містить відомості про співробітників магазина, це буде «продавець».

Звичайно як значення за замовчуванням вказується постійне значення, однак, можна використати й вирази. Наприклад: для введення поточної дати можна ввести вираз =Date(), що використає функцію «Date()», що виводить поточну дату. Якщо функція використовується у виразі за замовчуванням, то значення відповідного поля може бути згодом змінене вручну.

Вирази, які використовуються як значення за замовчуванням, не повинні містити посилання на елементи керування й інші поля, а також функції, визначені користувачем. Вирази можуть записуватися безпосередньо або будуватися за допомогою «Будівника виразів» («Expression Builder»). У кожному разі треба з обережністю ставитися до використання значень за замовчуванням, щоб у процесі введення відповідні поля випадково не залишилися незміненими.

6.5.2. Використання масок при введенні

Про використання масок введення вже говорилося в підрозділі «Задавання обмежень цілісності». Розглянемо деякі інші приклади. Можна використати маски для введення конфіденційної інформації (якщо використати маску типу «пароль», то замість символів, введених у поле, на екрані будуть зображуватися зірочки). Якщо, наприклад, в інституті прийняте позначення студентських груп, що включає дві великі букви, дефіс і три цифри, то для цього поля можна використати таку маску введення: >*LL*\-000. При цьому не треба буде перемикатися при введенні у верхній регістр, у якості двох перших символів можна буде ввести тільки букви, а останніх трьох – тільки ци-

фри. Знак «-» вводитися і зберігатися в записах БД не буде, він присутній тільки в масці при введенні й виводі даних.

Для прискорення введення даних у поточне поле таблиці можуть бути використані певні комбінації клавіш:

– Ctrl-; вводить поточну дату;

– Ctrl-: вводить поточний час;

– Ctrl-Alt-пробіл вводить значення поля, встановлене за замовчуванням;

– Ctrl-' або Ctrl-" вводить значення аналогічного поля з попереднього запису.

Будь-який запис автоматично зберігається у файлі бази даних на диску при переході до іншого запису.

6.6. Обмін інформацією між банками даних

Як видно з вищевикладеного, два додатки Microsoft Office (MS Excel і MS Access) дозволяють, хоч і з незрівнянно різними можливостями, але створювати банки даних того або іншого призначення. Було б дивним, якби фірма Microsoft не передбачила засобів для обміну інформацією між цими додатками.

Такі засоби дійсно існують. Обмін інформацією між зазначеними системами може здійснюватися в обох напрямках. При цьому пересилання даних з MS Access в MS Excel може знадобитися, зокрема, у таких випадках:

– відділ або робоча група застосовують для обробки даних і MS Access, і MS Excel. Дані зберігаються в базах даних MS Access, а аналіз даних і розсилання результатів аналізу проводяться за допомогою MS Excel. Звичайно співробітники експортують дані в MS Excel в міру необхідності, але потрібно зробити цей процес більш ефективним;

– співробітники звикли працювати з MS Access, але керівник воліє переглядати звіти в MS Excel. Співробітникам доводиться періодично копіюва-

ти дані в MS Excel, і потрібно автоматизувати цей процес, щоб заощаджувати час.

Зворотне пересилання даних (з MS Excel в MS Access) необхідне, наприклад, якщо:

– співробітники довго працювали з MS Excel, однак вирішили через якийсь час використати для цього MS Access. Їм потрібно перемістити дані з одного або декількох аркушів MS Excel в одну або кілька нових баз даних MS Access;

– відділ чи робоча група працюють з MS Access, але іноді співробітники одержують дані у форматі MS Excel, які необхідно ввести в бази даних MS Access. При одержанні цих аркушів MS Excel їх потрібно імпортувати в базу даних;

– співробітник обробляє дані за допомогою MS Access, але одержує від своїх колег щотижневі звіти у вигляді книг MS Excel. Має сенс автоматизувати процес імпорту, щоб щотижня в певний час ці дані були легко імпортовані в бази даних.

Існує кілька способів обміну даними між додатками MS Access та MS Excel.

Щоб дані з MS Access перемістити в MS Excel, можна:

– скопіювати дані таблиці MS Access і вставити їх в аркуш MS Excel;

– підключити базу даних MS Access до аркуша MS Excel;

– експортувати таблицю даних MS Access в аркуш MS Excel.

Дані з MS Excel можна перемістити в MS Access такими способами:

– скопіювати дані аркуша MS Excel і вставити їх у таблицю MS Access;

– імпортувати аркуш MS Excel у таблицю MS Access;

– зв'язати аркуш MS Excel з таблицею MS Access.

Необхідно мати на увазі, що в MS Excel слово «імпорт» має трохи інше значення, ніж в MS Access. В MS Excel «імпорт» означає встановлення постійного підключення до даних, які можуть бути оновлені. В MS Access «ім-

порт» – це однократне перенесення даних без постійного підключення до даних.

Крім того, варто пам'ятати, що книгу MS Excel неможливо зберегти у вигляді бази даних MS Access. Ні в MS Excel, ні в MS Access немає можливості створити базу даних MS Access на основі даних MS Excel.

З перерахованих вище можливостей обміну інформацією між MS Access і MS Excel як приклад розглянемо два: експорт таблиць з MS Access та імпорт таблиць в MS Access.

6.6.1. Експорт таблиць з MS Access

За допомогою майстра експорту MS Access можна експортувати в аркуш MS Excel об'єкт MS Access, наприклад, таблицю, запит, форму або виділені записи. При виконанні експорту можна зберегти параметри цієї дії для наступного використання, а також установити розклад автоматичного виконання експорту через задані проміжки часу.

Для **експорту таблиці** необхідно виконати наступні дії. У меню «File» вибрати позицію «Export» або правою кнопкою миші натиснути на назві об'єкта (таблиці) і з контекстного меню, що з'явилося, вибрати ту ж позицію.

Далі необхідно вибрати ім'я файлу («File name»), тип файлу («Save as type») (рекомендується «Microsoft Excel 97-2002») і натиснути кнопку «Export All». Експорт виконано. Залишається відкрити створений файл за допомогою MS Excel.

6.6.2. Імпорт таблиць в MS Access

Щоб у базі даних MS Access зберегти дані з MS Excel, а потім продовжити працювати з ними в додатку MS Access, дані можна імпортувати. При імпорті даних MS Access зберігає їх у новій або існуючій таблиці, не змінюючи даних з MS Excel. За один імпорт можна імпортувати тільки один аркуш

даних. Щоб імпортувати дані з декількох аркушів, необхідно повторити імпорт для кожного аркуша.

Для імпорту таблиці необхідно зробити наступне. У вікні об'єктів (рис. 6.2) вибрати позиції «New», потім «Import Table» або натиснути правою кнопкою миші на вільному місці вікна й вибрати в контекстному меню, що з'явилося, позицію «Import».

Далі необхідно вибрати ім'я файлу («File name»), тип файлу («Files of type») (у нашому випадку – «Microsoft Excel») і натиснути кнопку «Import». Після цього вибираються необхідні робочі аркуші (Worksheets). Слідом за цим, MS Access пропонує використати перший рядок (заголовок) таблиці MS Excel для задавання імен полів у таблиці MS Access. Як правило, має сенс погодитися, залишивши пташку у відповідному віконці.

Наступний етап – визначити, чи хоче користувач створити нову таблицю MS Access з аркуша MS Excel, або додати дані в існуючу таблицю. Звичайно вибирають нову таблицю («In a New Table»). Далі визначається тип полів (як правило – за замовчуванням) і необхідність додавання ключа запису (тільки якщо в таблиці MS Excel він був відсутній). Заключний етап – вибір імені таблиці MS Access. Після натискання кнопки «Finish» експорт виконано. Залишається відкрити створену (модифіковану) таблицю MS Access.

Питання й тести для самоконтролю

1. Опишіть СУБД MS Access з погляду класифікації інформаційних систем.
2. Як створити нову базу даних в MS Access?
3. Які Ви знаєте способи створення таблиць в MS Access?
4. Перелічить можливі типи даних у записах таблиці MS Access.
5. Що таке ключ запису й навіщо він потрібний?
6. Чим розрізняються режим Конструктора таблиць і режим Майстра таблиць?

7. Як виконується зв'язування таблиць у СУБД MS Access?
8. Як забезпечується цілісність бази даних?
9. Що таке маски введення і для чого вони слугують?
10. Розкажіть про зв'язування банків даних MS Excel і MS Access.

Розділ 7. РЕАЛІЗАЦІЯ ЗАПИТІВ В СУБД

Після того, як банк даних створено, основними функціями для роботи з ним стають пошук і вибірка необхідної частини інформації, додавання й коректування даних. Дві останні функції принципово не відрізняються від розглянутих у попередньому розділі. Навпаки, доступ до підмножин інформації бази даних має ряд особливостей, які буде розглянуто в даному й наступному розділах.

На сьогоднішній день основним механізмом вибірки інформації з БД є апарат запитів.

Запити (queries) – це об'єкт бази даних, що слугує для добування даних з таблиць і надання їх користувачеві в зручному вигляді. Особливість запитів полягає в тому, що вони отримують дані з базових таблиць і створюють на їх основі тимчасову таблицю. Застосування запитів дозволяє уникнути дублювання даних у таблицях і забезпечує максимальну гнучкість при пошуку та відображенні даних у базі даних.

У наш час найчастіше використовуються запити двох типів: QBE-запити й SQL-запити.

QBE-запити (Query By Example) – запити, що будуються за допомогою конструктора запитів, що є деяким графічним інструментом для створення запитів за зразком.

SQL-запити – запити, що будуються за допомогою уніфікованого набору інструкцій SQL (Structured Query Language – структурована мова запитів).

Всі запити діляться на дві групи: запити-вибірки і запити-дії.

Запити-вибірки здійснюють вибірку даних з таблиць відповідно до заданих умов. До цієї групи запитів відносяться, зокрема:

– запит до зв'язаних таблиць. Дозволяє робити вибірку даних зі зв'язаних таблиць;

– перехресний запит. Відображає підсумкові дані з групуванням їх по горизонталі й вертикалі, виводячи результати їхньої обробки у вигляді таблиць;

– запит з параметром. Дозволяє користувачеві задати критерій відбору, ввівши потрібний параметр при виклику запиту;

– запит з обчислюваним полем. Дає можливість розрахувати дані на основі інших полів з того ж рядка запиту;

– запит з критерієм пошуку. Дозволяє робити відбір записів відповідно до заданого критерію пошуку;

– запит з підсумками. Робить математичні обчислення й видає результат.

Запити-дії дозволяють модифікувати дані в таблицях: видаляти, обновляти, додавати записи. До цієї групи запитів відносяться такі:

– запити на створення таблиці. Створюють таблиці на підставі даних, що утримуються в результуючій множині запиту;

– запити на додавання записів. Дозволяють додавати в таблицю записи, створювані запитом;

– запити на відновлення. Змінюють значення існуючих полів відповідно до заданого критерію;

– запити на видалення. Видаляють записи з однієї або декількох таблиць одночасно.

У розглянутій нами як приклад СУБД MS Access QBE-запити можна створювати двома способами: за допомогою **Майстра запитів** і за допомогою Конструктора. У даному навчальному посібнику розглядається тільки Конструктор запитів, як найбільш зручний для практичних застосувань.

7.1. Створення запитів в MS Access

Після опису таблиць і заповнення їх даними, до бази даних можна формулювати різноманітні запити. Для задавання запиту в MS Access варто пе-

рейти до закладки «Queries» у вікні бази даних. Для створення нового запиту слід натиснути кнопку «New», у результаті чого з'явиться вікно «New Query» (рис. 7.1). Існує кілька способів створення запитів. Найчастіше використовується можливість створення запиту за допомогою **«Конструктора запитів»** («Design View»).

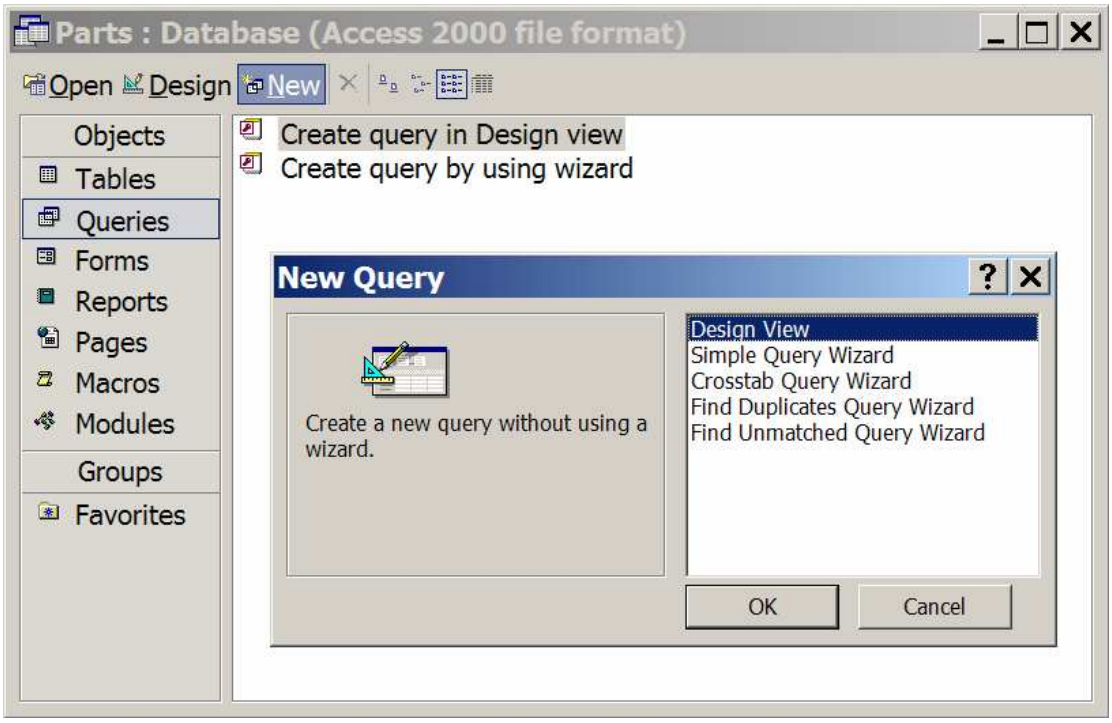


Рисунок 7.1 – Вікно нової бази даних

7.1.1. Додавання в запит таблиць

Початковим етапом при створенні запиту є визначення таблиць, що містять вихідну інформацію. Допускається також створення запиту на підставі інших запитів або одночасно як таблиць, так і запитів. Використання попередньо створених запитів при створенні нового запиту може допомогти зробити складний запит, що містить велику кількість взаємозалежних таблиць і різноманітних умов відбору, більш простим для його формулювання. У деяких випадках без розбивки запиту на кілька послідовно виконуваних кроків не можна обійтися.

Якщо позиціюватися на рядок «Design View» у вікні нового запиту, з'явитися вікно «Show Table» (рис. 7.2), що дозволяє вибрати таблиці або запити, що є джерелом даних для створюваного запиту. Для того, щоб вказати, на чому буде базуватися створюваний запит (таблиці, запиті або тому й іншому одночасно), треба просто вибрати відповідну закладку. Установивши в списку доступних таблиць і запитів, що з'явився, покажчик на ім'я таблиці або запиту, що додається, треба або виконати подвійний щиглик мишею, або натиснути клавішу «Введення».

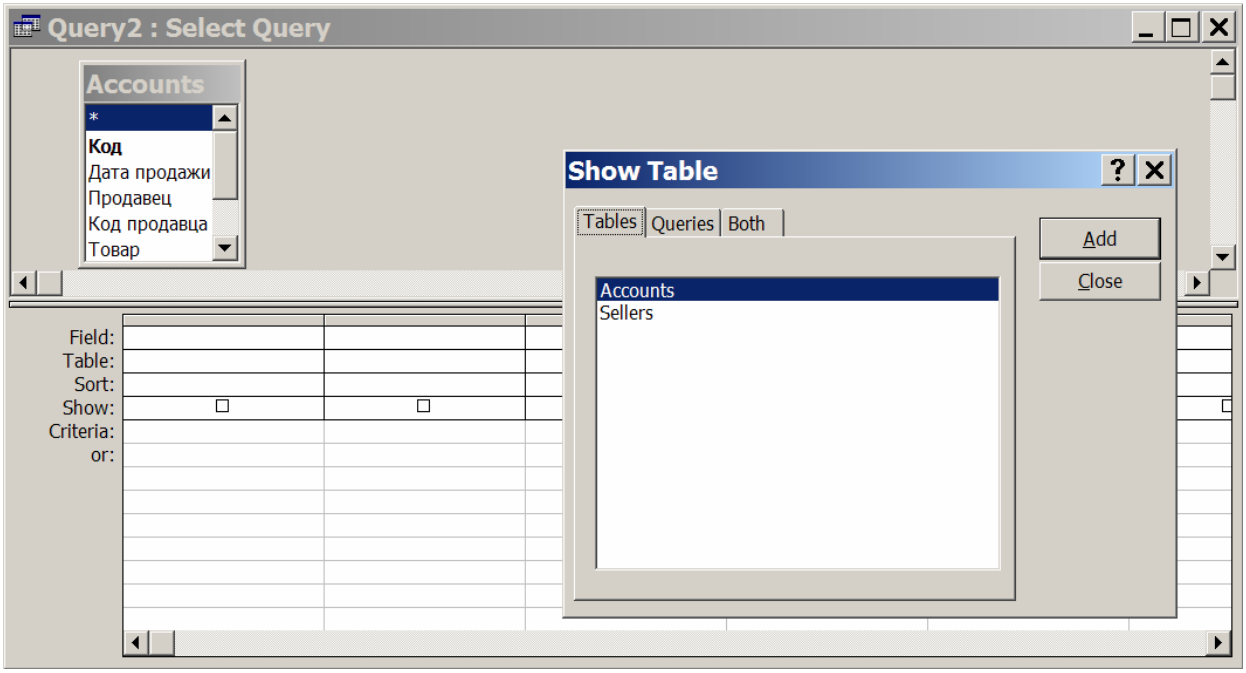


Рисунок 7.2 – Вибір джерела запиту

Допускається додавання в запит відразу декількох таблиць або запитів. Для цього треба, утримуючи натиснутою клавішу Ctrl, вибрати імена таблиць або запитів, що додаються, і натиснути кнопку «Add». У верхній частині вікна запиту виводиться список полів доданої таблиці або запиту (рис. 7.2).

Можливе використання ще кількох способів включення в запит таблиць, на яких базується запит. По-перше, для додавання таблиці можна в режимі Конструктора запиту натиснути кнопку «Show Table» на панелі інструментів або вибрати в меню «Query» команду «Show Table». При цьому від-

кривається вікно діалогу «Show Table». Користувач має також можливість додати в запит таблицю або запит, вибравши їхні імена у вікні бази даних і перемістивши їх за допомогою миші у верхню частину вікна запиту. Крім того, можна позиціюватися на вільне місце у верхній частині вікна запиту, натиснути праву кнопку миші та у спадаючому меню, що з'явилось, вибрати позицію «Show Table». Для додавання в запит таблиці з іншої бази даних або іншого додатка варто спочатку приєднати цю таблицю до активної бази даних. Це приєднання виконується шляхом використання команди «Link Tables» (меню «File»).

7.1.2. Видалення із запиту таблиць

Якщо Ви помилково включили якусь таблицю в запит або по яких-небудь інших причинах Вам треба видалити раніше включену таблицю із запиту, це легко можна зробити. Існує кілька способів видалення таблиці із запиту:

– вибрати ім'я таблиці або запиту у відповідному списку і натиснути клавішу Del або вибрати в меню «Query» команду «Remove Table»;

– подвійним щигликом миші виділити потрібну таблицю у відповідній зоні екрана, а потім натиснути клавішу Del. Імена полів вилученої таблиці або запиту видаляються із бланка запиту.

Зазначимо, що видалення з деякого запиту таблиці або запиту, на яких він базується, не призводить до видалення відповідних об'єктів з бази даних.

7.1.3. Додавання в запит полів

Після того, як вихідні таблиці й запити визначені, треба вибрати поля, використовувані в створюваному запиті. Існує кілька способів переносу поля в бланк запиту:

– перетаскування за допомогою миші;

– подвійний щиглик мишею на імені відповідного поля в списку полів;

– вибір поля в списку полів, що розкривається, який з'являється, якщо натиснути на знак стрілки в рядку «Field» бланка запиту.

Можна переносити в бланк запиту не по одному полю, а відразу необхідну сукупність полів. Виділення полів, що підлягають переносу, здійснюється стандартним для MS Windows способом (натиснути клавішу Shift для виділення полів, розташованих один за одним, і, утримуючи її, вибрати перше й останнє поля набору. Або вибирати поля при натиснутій клавіші Ctrl. Це зручніше робити для полів, що розташовуються в довільному порядку, не підряд один за одним).

Не всі поля, які використовуються при формуванні запиту, повинні обов'язково відбиватися у відповіді. Так, наприклад, поле може бути необхідно для задавання умови відбору, але потреба в його появі у відповіді відсутня. Поля, виведені у відповідь, вказуються в рядку Конструктора запиту «Вивід на екран» («Show»). Відповідні стовпчики цього рядка відзначаються знаком входження поля у відповідь («пташкою»).

Можна перенести в бланк запиту одночасно всі поля. Для цього треба встановити покажчик на заголовок списку полів і двічі клацнути кнопкою миші або встановити покажчик на символ зірочки й натиснути кнопку миші.

Є різниця, як поля були введені в запит. При використанні символу зірочки в запит автоматично включаються всі поля, додані в базову таблицю або запит після створення даного запиту. Всі вилучені поля будуть автоматично видалятися із запиту. З одного боку, це добре, з іншого боку – може трапитися, що користувач у результаті видачі того самого запиту буде одержувати різні відповіді, і, можливо, не ті, які він очікує.

Наприклад, якщо в таблиці «Accounts» спочатку фіксувалися тільки основні дані по торгівлі, а потім було введено багато інших полів, то зовсім не обов'язково, що користувач захоче бачити всі ці дані, у відповідь на свій запит. Якщо ж поле, включене в запит явним способом, було згодом вилучене з таблиці, то запит може виконуватися не зовсім коректно. Тому що поля,

включені в запит шляхом використання зірочки, у явному вигляді в бланку запиту не висвічуються, ті поля, які використовуються в умові відбору, треба додатково включити в бланк запиту. Щоб ці поля двічі не виводилися у відповідь, треба в цих полів зняти прапорець «Show» (рис. 7.3).

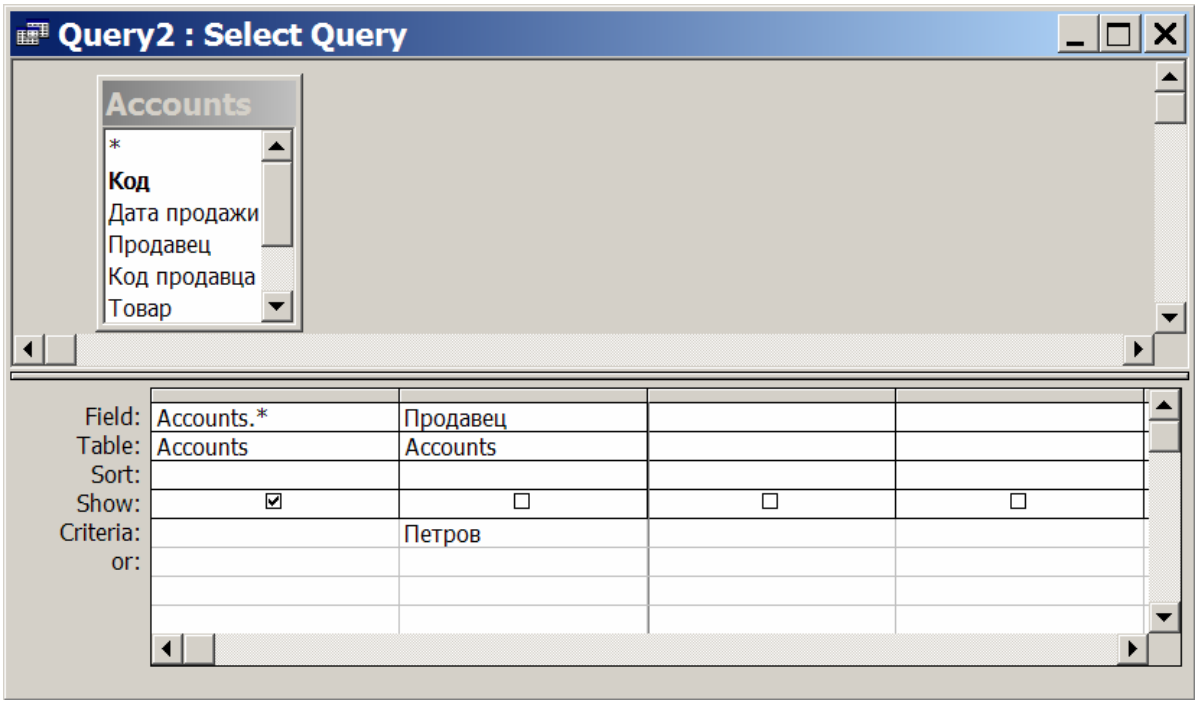


Рисунок 7.3 – Використання в запиті зірочки

Зображений на рис. 7.3 запит реалізує вивід всіх даних, що містяться в таблиці «Accounts», для продавця «Петров», причому поле «Продавець» виводиться у відповіді тільки один раз, тому що відсутня пташка в колонці «Продавець».

Узагальнюючи вищесказане, можна зробити наступний висновок: поля мають включатися в бланк запиту в тому випадку, якщо вони потрібні у відповіді, або якщо вони використовуються для задавання умов відбору. В останньому випадку вони можуть включатися, а можуть і не включатися у відповідь.

7.1.4. Задавання умов відбору

Очевидно, що при створенні запитів найважливішим моментом є задавання умов відбору. У попередньому прикладі ми вже використовували умову відбору для одержання інформації про одного конкретного продавця. Мова QBE, реалізована у СУБД MS Access, відноситься до класу табличних двовимірних мов. Умову відбору необхідно задавати в таблиці бланка запиту в тій графі, до якої відноситься ця умова. На рис. 7.3 така умова задана в графі «Продавець». Розрізняють кілька різновидів запитів:

– запит на вибірку (Select);

– перехресний запит (CrossTab);

– створення таблиці (Make-Table);

– відновлення (Update);

– додавання (Append);

– видалення (Delete).

Більш докладно створення запитів різних видів буде розглянуто нижче в цьому розділі.

7.1.5. Керування виводом повторюваних рядків

Якщо у відповіді не виводяться всі поля вихідної таблиці, може трапитися, що рядки в ньому будуть повторюватися. Наприклад, якщо вивести тільки список товарів з таблиці «Accounts», то найменування тих самих товарів можуть зустрічатися кілька разів. Для того, щоб керувати виводом повторюваних рядків, можна позиціюватися на довільне місце поза бланком запиту і списку полів, натиснути праву кнопку миші, і в контекстному меню, що з'явилося (рис. 7.4) вибрати рядок «Properties» (або вибрати відповідну кнопку на панелі інструментів).

Серед властивостей запиту (рис. 7.5) є два: «Unique Values» («Унікальні значення») та «Unique Records» («Унікальні записи»), зміст яких зрозумілий з

назви. Якщо ви хочете, щоб у відповідь видавався список товарів без повторів, задайте для властивості «Unique Values» значення «Yes».

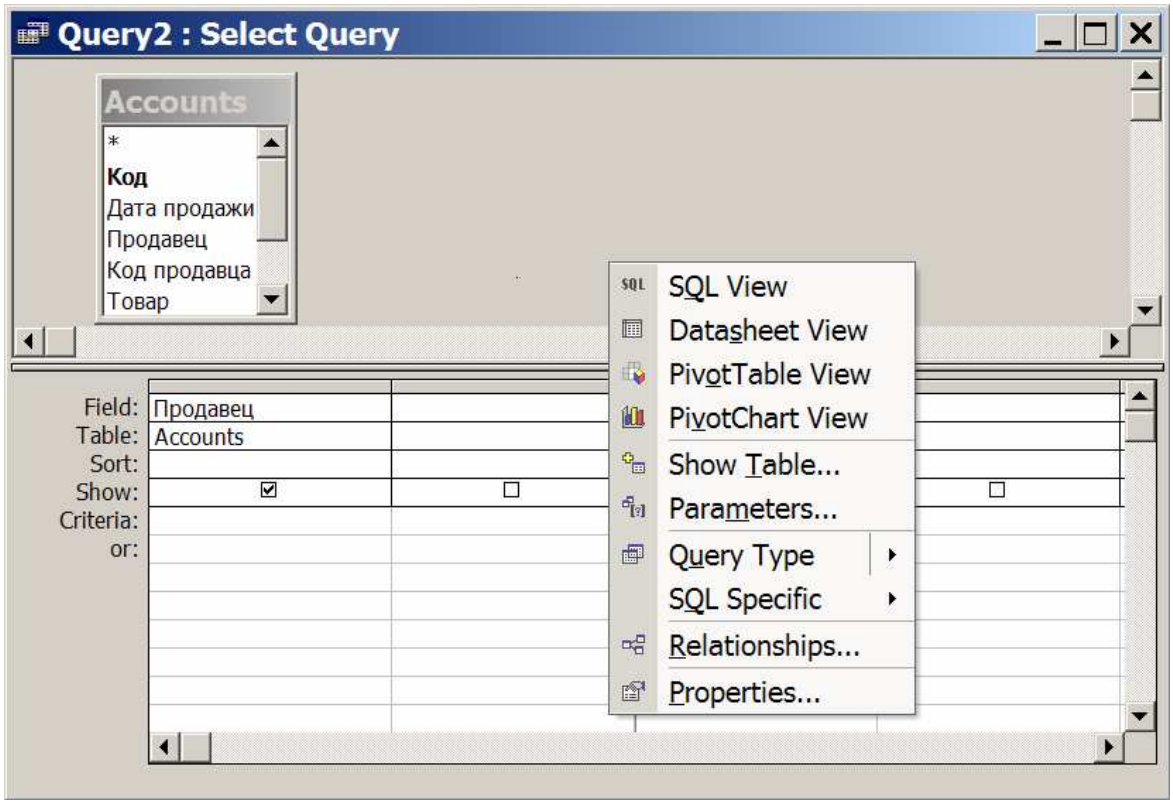


Рисунок 7.4 – Контекстне меню поля в запиті

7.1.6. Попередній перегляд результатів виконання запиту

Щоб подивитися, що буде виходити у відповідь на сформований запит, і переконатися, що результат відповідає бажаному, можна клацнути мишею на кнопці «Run» («!») на панелі інструментів, або вибрати відповідну можливість з меню «Query – Run», або клацнути на стрілку на кнопці «View» і вибрати зі списку, що з'явився, пункт «Datasheet View». Для того, щоб знову повернутися до побудови або коректуванню запиту, треба вибрати режим «Design View» (рис. 7.6).

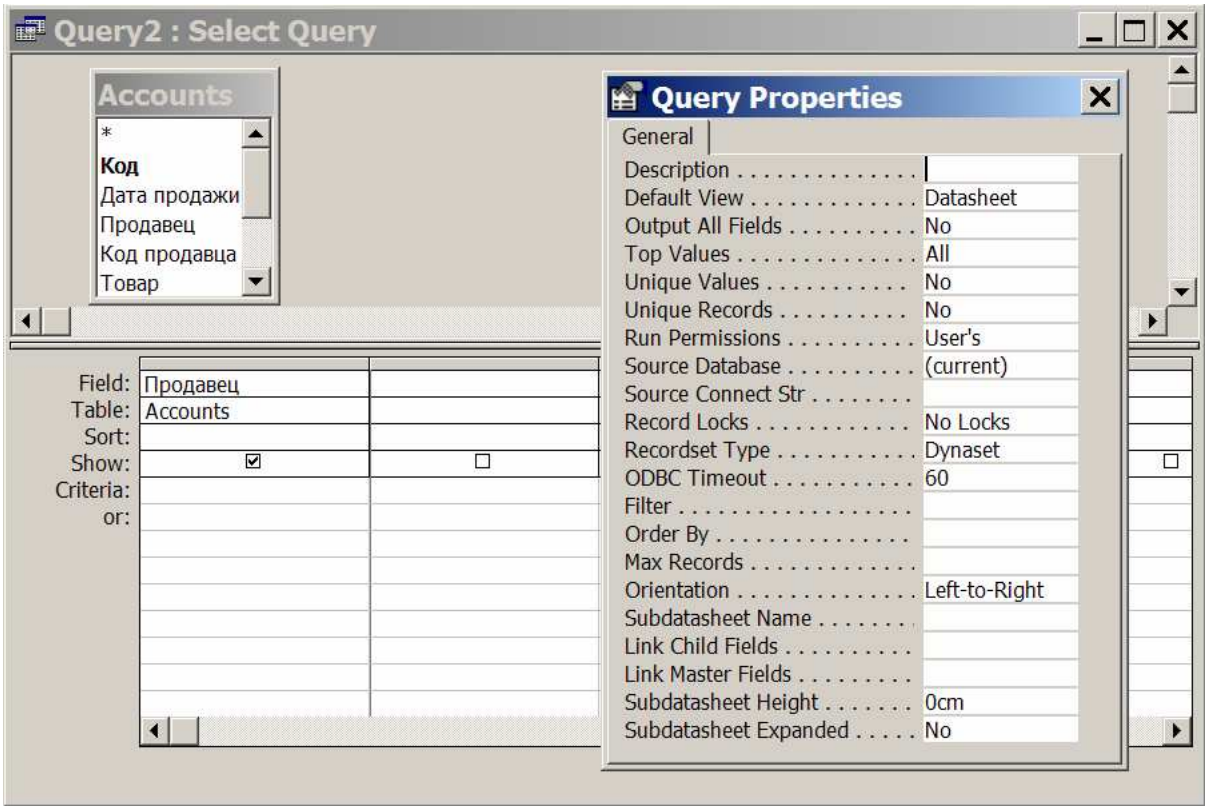


Рисунок 7.5 – Властивості запиту

7.1.7. Збереження запиту

Сформований запит можна зберегти для наступного використання. Це можна зробити декількома способами, наприклад, вибравши позиції головного меню «File – Close», відповівши «Yes» на питання про збереження файлу і задавши після цього ім'я запиту. Збережений запит можна згодом «відкривати», що означає його виконання. Збережений запит може бути скоректований, якщо відкрити його в режимі Конструктора.

7.2. Різновиди запитів

Як ми вже відзначали, існують кілька різновидів запитів. Найпоширенішим є запит на вибірку інформації з бази даних. Розглянемо часто використовувані види запитів докладніше.

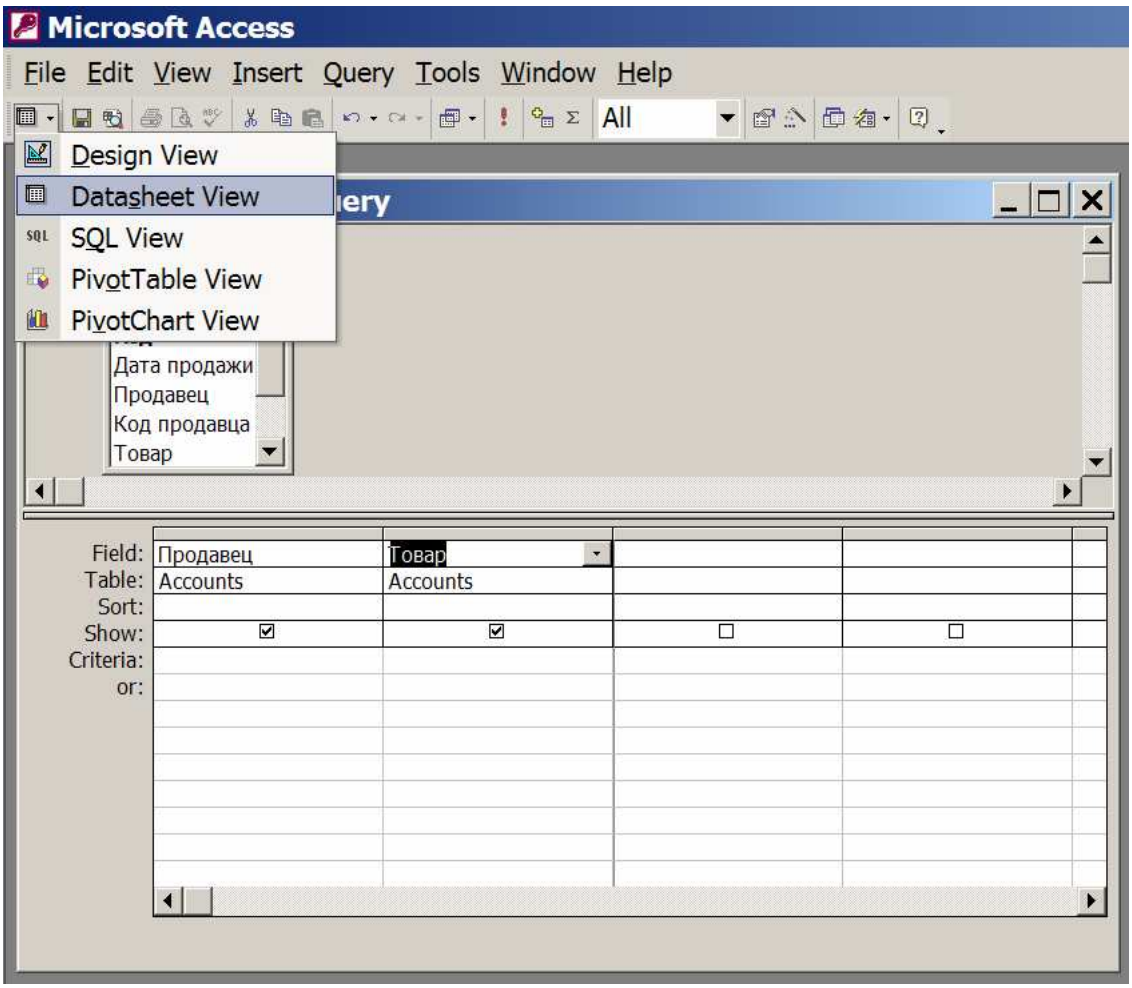


Рисунок 7.6 – Зміна режимів перегляду

7.2.1. Простий запит

Простий запит – це запит з найпростішими умовами, що включають тільки один аргумент пошуку. При створенні простого запиту умова відбору записується у відповідний стовпець бланка запиту. Наприклад, якщо треба відібрати інформацію про конкретного продавця, у стовпець «Продавець» у рядку «Criteria» необхідно записати прізвище даного продавця (рис. 7.3).

Як відомо, у більшості СУБД при введенні у вираз значень того або іншого типу використовуються відповідні даному типу дані обмежники. В MS Access при задаванні запиту обмежники можна не ставити. Залежно від типу поля, що вводиться у вираз, який визначає умову відбору, обмежники додаються системою автоматично:

– прямі лапки (") навколо строкових значень;

– символи # навколо дат.

У стовпці можна записувати не тільки значення атрибута, але й знак порівняння; за замовчуванням приймається знак «дорівнює». Якщо Вам, наприклад, треба визначити список всіх продавців, що продали товари ціною більше 1000 грн, то запит буде виглядати так, як зображено на рис. 7.7.

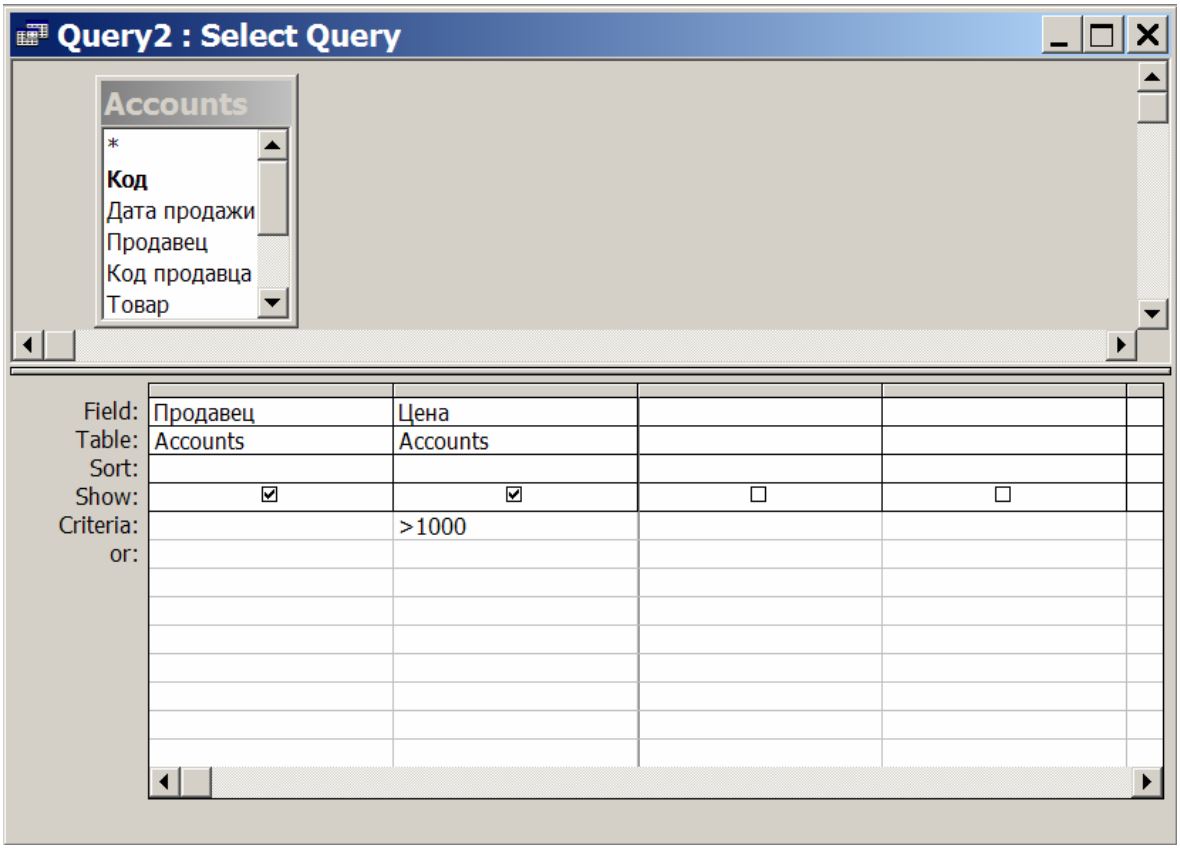


Рисунок 7.7 – Використання умов порівняння

В умовах відбору можна задавати й діапазон значень. У цьому випадку запит буде виглядати подібно тому, як зображено на рис. 7.8. Цю ж умову відбору в графі «Виторг» можна було б задати й у такий спосіб: >=100 And <=300. В MS Access можна задавати й запити з відкритими двосторонніми діапазонами. Наприклад, для видачі списку продажів, у яких отриманий прибуток менше 50 грн або більше 500 грн (тобто дрібних і великих угод), умову відбору треба задати в такий спосіб: <50 Or >500.

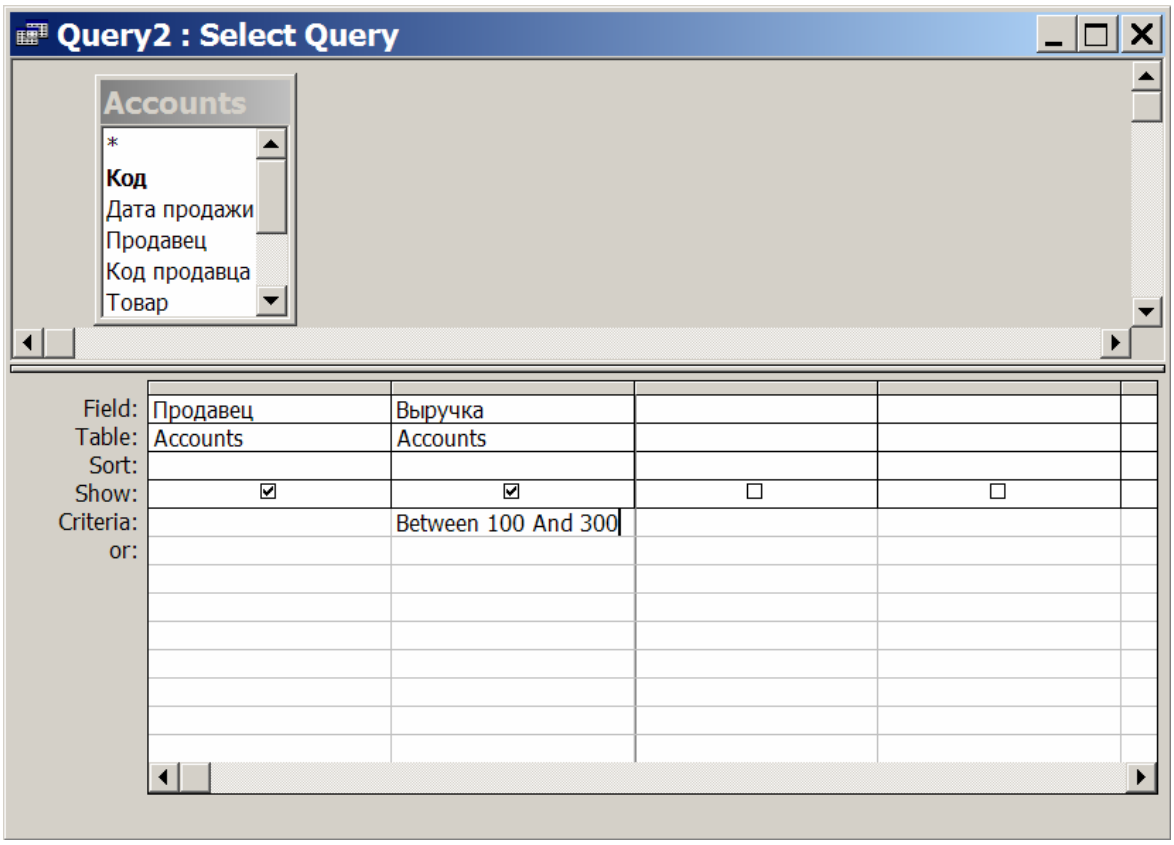


Рисунок 7.8 – Задавання діапазону значень

7.2.2. Складний запит

Якщо в умовах відбору використовується кілька полів, то вони можуть з'єднуватися оператором «And» або «Or». Якщо аргументи пошуку записані в одному рядку – вважається, що вони з'єднані оператором «And». Якщо аргументи пошуку записані в різних рядках – вважається, що вони з'єднані оператором «Or».

На рис. 7.9 і рис. 7.10 зображені приклади таких запитів. Перший з них видає список всіх продажів, у яких Іванов приніс магазину не менш, ніж 300 грн. прибутку (запит «And»; аргументи запиту розташовані на одному рядку), другий (запит «Or»; аргументи запиту розташовані на різних рядках) – всіх продажів, у яких ціна товару була менше 100 грн., або кількість проданих товарів не перевищувала трьох.

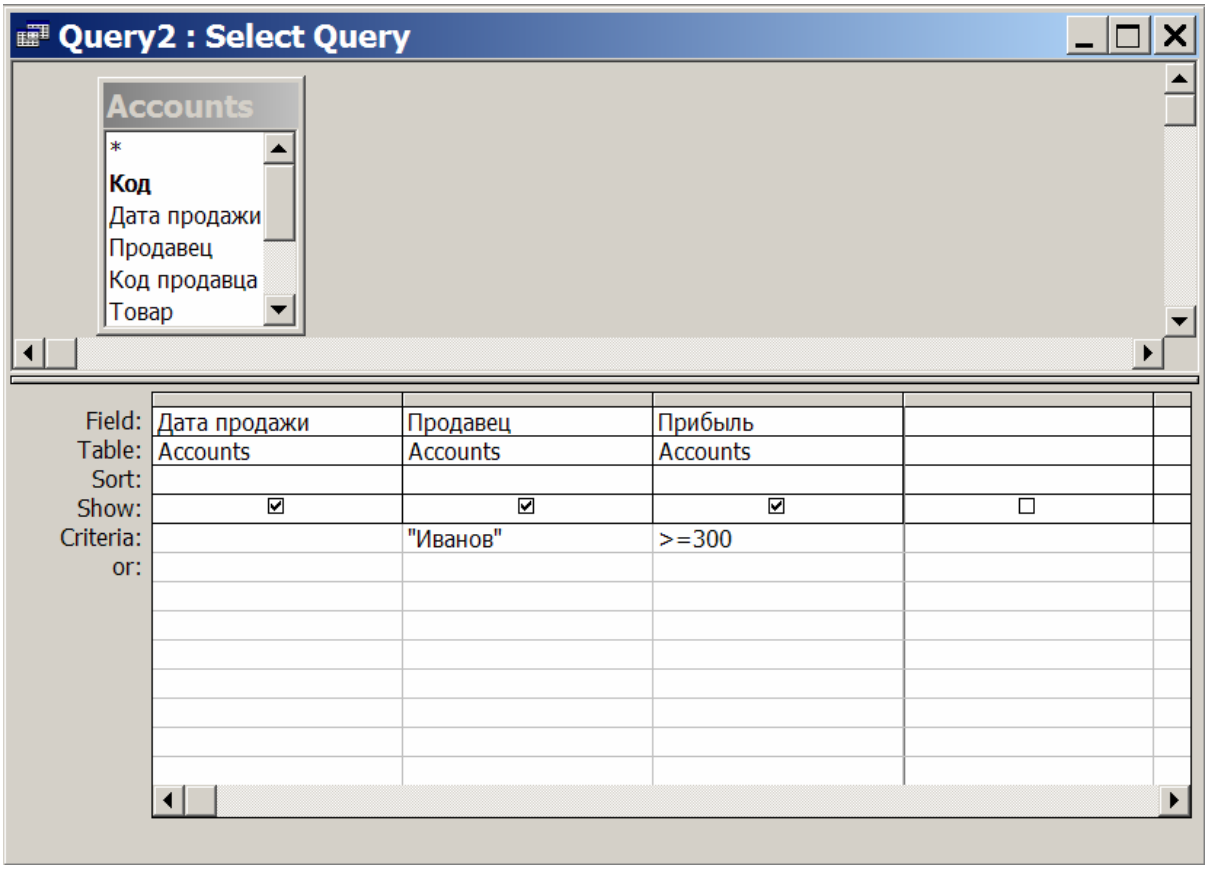


Рисунок 7.9 – Складний запит (оператор And)

7.2.3. Запит до зв'язаних таблиць

Якщо при створенні бази даних попередньо була визначена схема даних, то при додаванні таблиць у запит вони виявляться зв'язаними. Навіть якщо зв'язки між таблицями не були створені користувачем попередньо, однаково при додаванні в запит двох таблиць, що містять поля з однаковим або сумісним типом даних, а також, якщо одне з полів зв'язку є ключовим, зв'язки можуть бути створені автоматично.

Автоматичне об'єднання (з'єднання) можна дозволити або заборонити. Для цього треба виконати таку послідовність кроків:

– у меню «Tools» вибрати команду «Options»;

– перейти до вкладки «Tables – Queries»;

– установити або зняти прапорець «Enable AutoJoin».

Параметр «Enable AutoJoin» відноситься тільки до нових запитів.

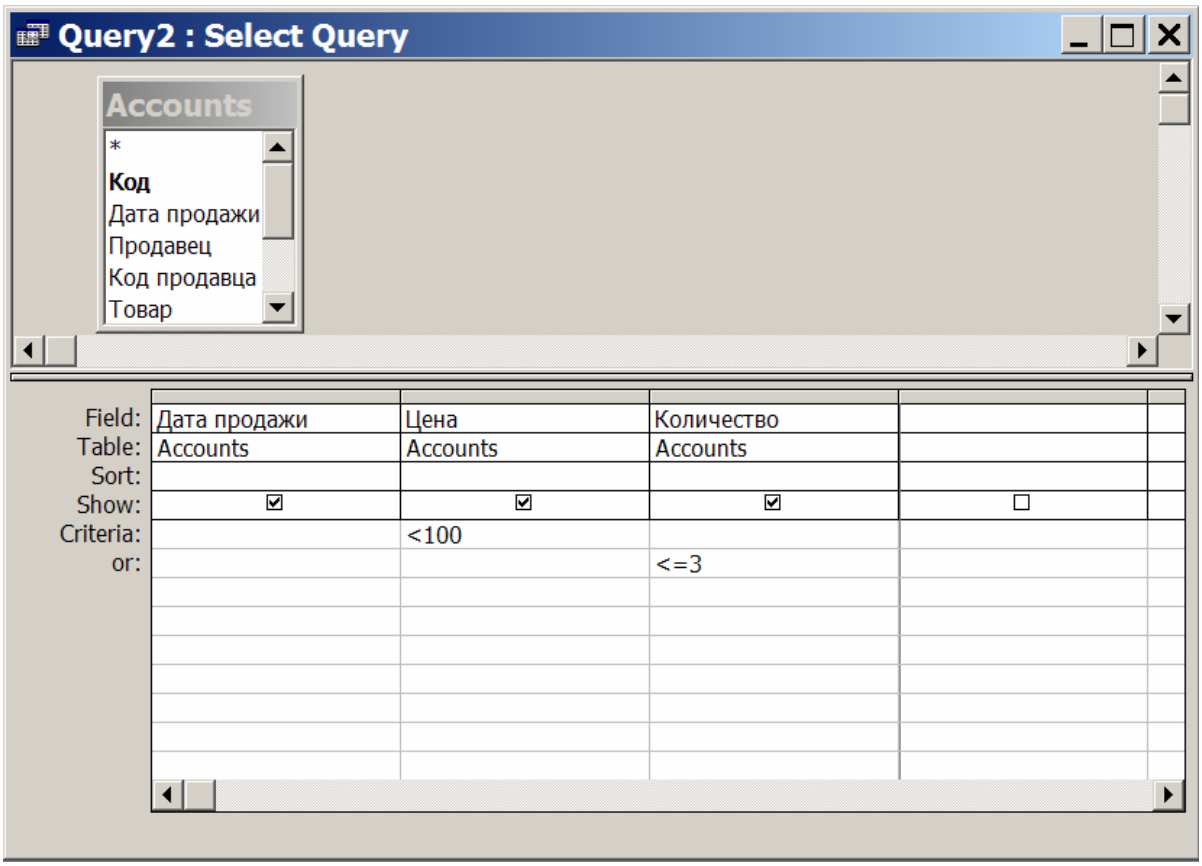


Рисунок 7.10 – Складний запит (оператор Or)

Якщо зв'язки не були визначені попередньо і не створені автоматично, треба задати з'єднання таблиць вручну (так само, як це робилося при задаванні схеми). Якщо зв'язок не заданий (і не відмінений параметр «Enable AutoJoin»), буде здійснюватися зв'язок кожного запису однієї таблиці з кожним записом другої таблиці.

Треба уважно ставитися до формування запитів до зв'язаних таблиць. Наприклад, не можна однозначно відповісти, що буде одержано в результаті запиту, зображеного на рис. 7.11. Для цього необхідна додаткова інформація. Зокрема, треба знати, якими є параметри об'єднання (якщо Ви уважні, то по виду лінії зможете визначити вид зв'язку) і які значення мають властивості «Unique Values» й «Unique Records» (цього на схемі не видно). Якщо задано звичайне («внутрішнє») з'єднання таблиць і для властивості «Unique Values» задане значення «Yes», то у відповідь на запит, що містить у бланку запиту поле «Продавець» і більше нічого, буде одержаний порожній список.

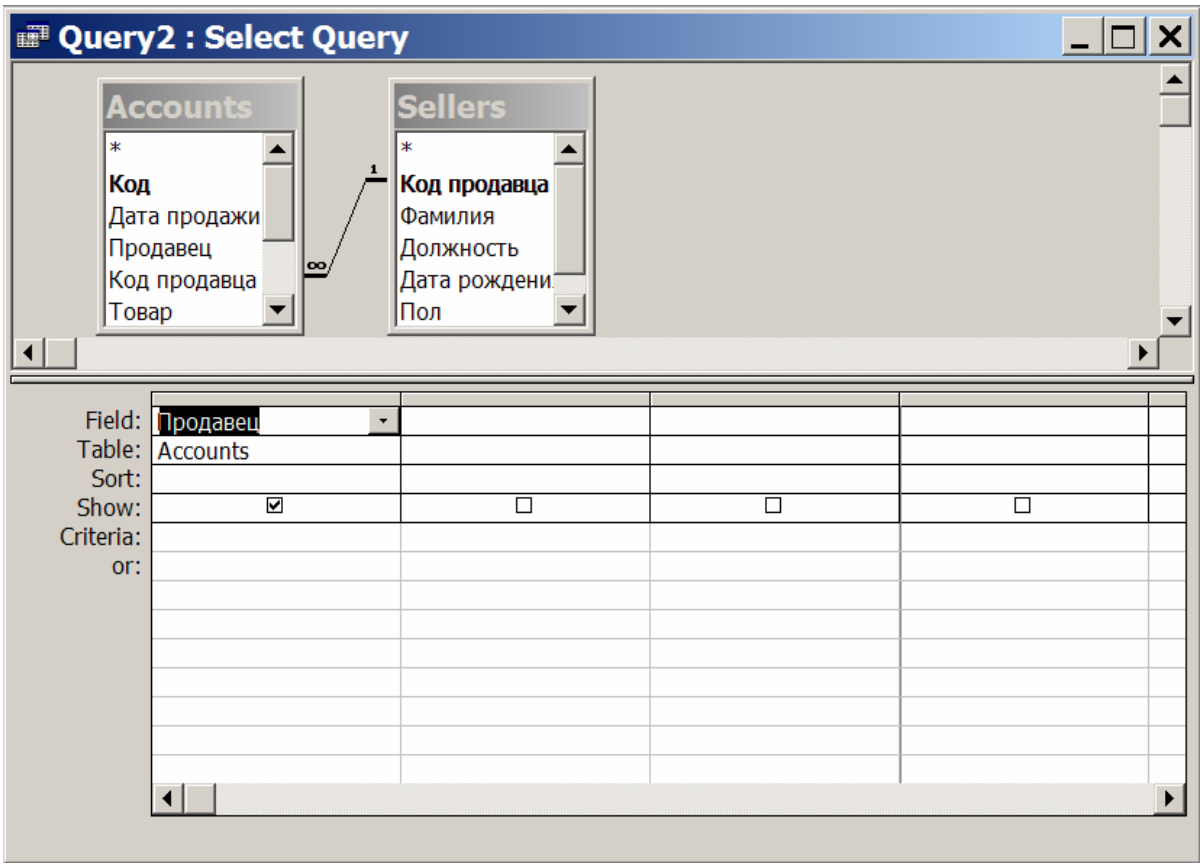


Рисунок 7.11 – Запит до зв'язаних таблиць

Таким чином, щоб уникнути помилок, рекомендується:

– при задаванні запиту видаляти з нього всі таблиці, поля яких не беруть участі у формуванні запиту;

– при проектуванні структури бази даних ретельно продумувати імена, які даються полям різних таблиць;

– перевіряти зв'язки, які система задає автоматично.

7.2.4. Запит з обчислюваними полями

Існує ряд обчислень, які можна виконати в запиті. Наприклад, перемножити значення двох полів або обчислити дату, що відстоїть на три місяці від поточної дати. Вирази, обумовлені користувачем, дають можливість виконувати дії із числами, датами й текстовими значеннями в кожному записі з використанням даних з одного або декількох полів таблиці.

Дозволені дії залежать від типу полів, що беруть участь у виразі. Так, для текстових полів можливо використання оператора «+», що у цьому випадку сприймається як конкатенація (з'єднання) рядків. Припустимо, поряд з полем «Продавець» у таблиці є поле «Касир». Якщо Ви хочете вивести обидва поля в одному стовпці, можна використати вираз: [Продавець] + " " + [Касир] (посередині вставляється пробіл, щоб два прізвища не зливалися).

Однак треба мати на увазі, що, якщо хоча б одне із цих двох полів буде не заповнене (причому байдуже, буде це порожнє поле, або там будуть введені пробіли), весь результуючий рядок виявиться «порожнім». У зв'язку з цим рекомендується замість знака «+», використовувати знак конкатенації «&». У цьому випадку вираз буде мати вигляд: [Accounts]![Продавець] & " " & [Accounts]![Касир] .

Для розрахунків з використанням формул, обумовлених користувачем, потрібно створити нове обчислюване поле безпосередньо в бланку запиту. Поле, що обчислюється, створюється шляхом введення необхідного виразу в порожню колонку в рядок «Field» у бланку запиту. Вираз може вводитися вручну, а можна для цих цілей скористатися Будівником виражень.

На рис. 7.12 використовується додавання числових величин. Результати обчислень не обов'язково повинні відображатися у відповіді. Їх можна використати в умовах відбору для визначення записів, які вибираються в запиті, або для визначення записів, над якими виконуються які-небудь дії. Наприклад, на рис. 7.12 зображено запит: «Видати список продажів, що принесли виторг понад 1000 грн із урахуванням акційної знижки на ціну товару 10 %». Саме обчислюване поле тільки використовується в умовах відбору, але у відповідь не виводиться.

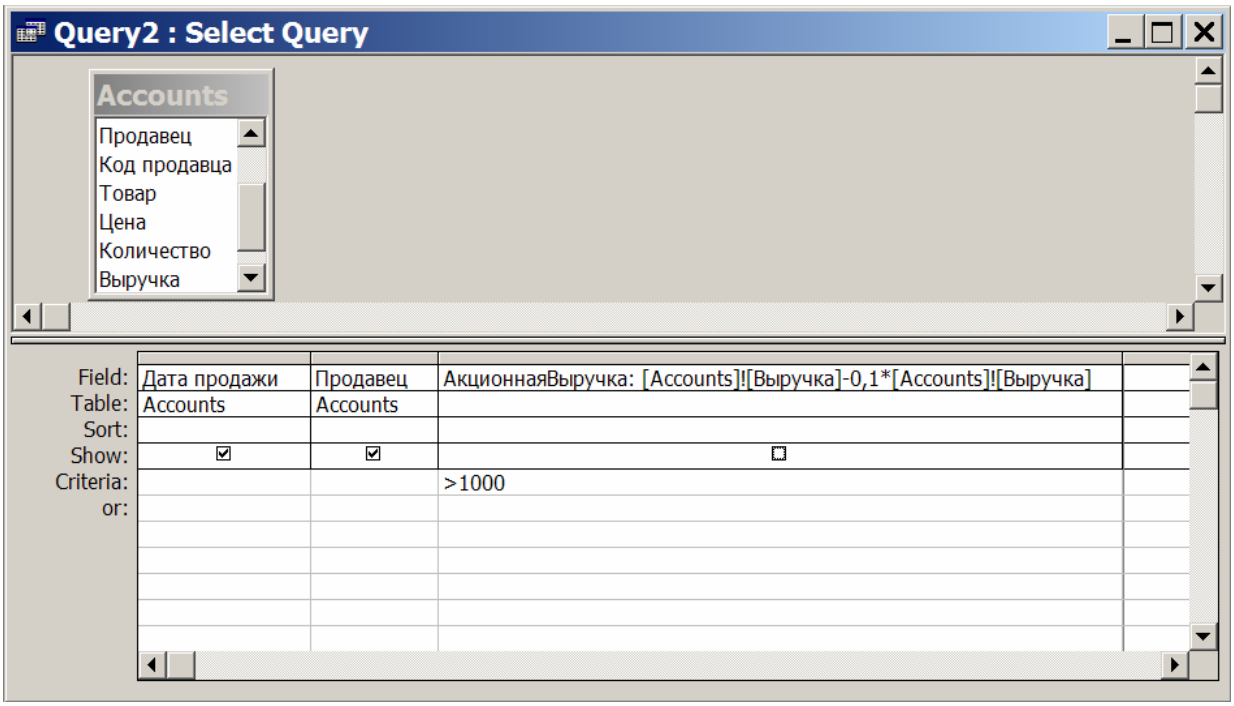


Рисунок 7.12 – Запит з обчислюваним полем для умови відбору

Варто також звернути увагу на можливі дії з датами. Над полями з цим типом можна робити такі дії: від дати можна відняти іншу дату; при цьому виходить число, що показує, на скільки днів відстоїть одна дата від іншої (якщо Ви хочете одержати інтервал часу в інших одиницях виміру, варто скористатися функцією DateDiff). Також від дати можна відняти або додати до неї число; при цьому виходить дата, що відстоїть від даної на задане число днів.

7.2.5. Коригувальні запити

Коригувальні запити (запит на відновлення (Update), видалення (Delete), додавання (Append)) можуть змінювати як всі записи таблиці, так і певну їх підмножину – це буде залежати від умови відбору.

Для створення коригувального запиту треба в режимі Конструктора запиту вибрати відповідну позицію в меню «Query» (або натиснути стрілку поруч із кнопкою «Тип запиту» на панелі інструментів), як показано на рис. 7.13. Щоб переглянути обновлювані записи перед виконанням запиту можна

натиснути кнопку «View» на панелі інструментів. Виведений список буде містити старі значення полів відібраних у запиті записів.

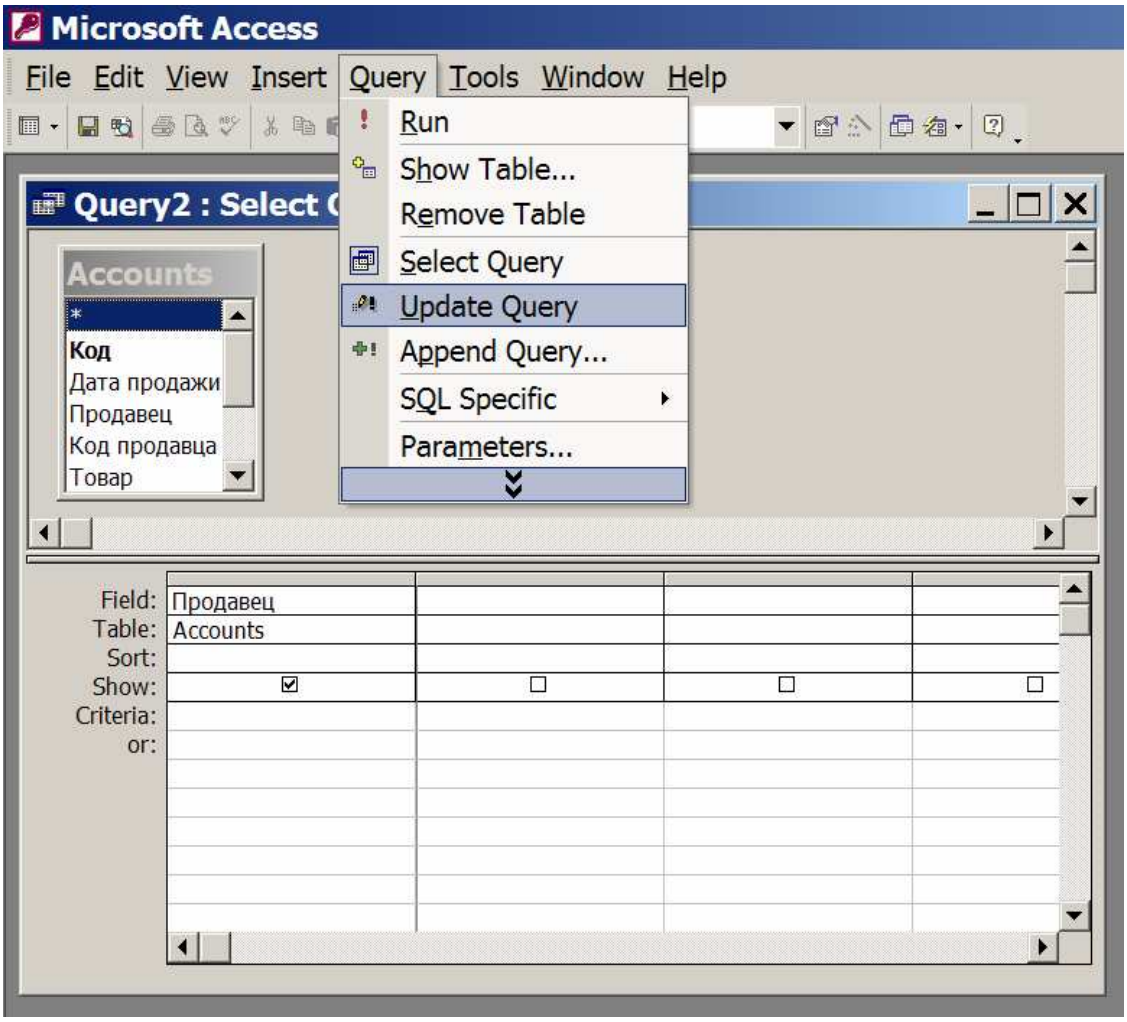


Рисунок 7.13 – Створення коригувального запиту

Треба бути уважним перед виконанням коригувальних запитів (для цього в списку запитів перед їхнім ім'ям стоїть знак оклику), тому що кожен їхній запуск на виконання змінює вміст таблиць, і скасувати результат виконання неможливо. Перед виконанням коригувального запиту рекомендується зберегти копію змінюваних таблиць.

Розглянемо спочатку запити на відновлення. Наприклад, запит, зображений на рис. 7.14, збільшить ціну всіх товарів (тому що умова відбору не задана) на 30 %.

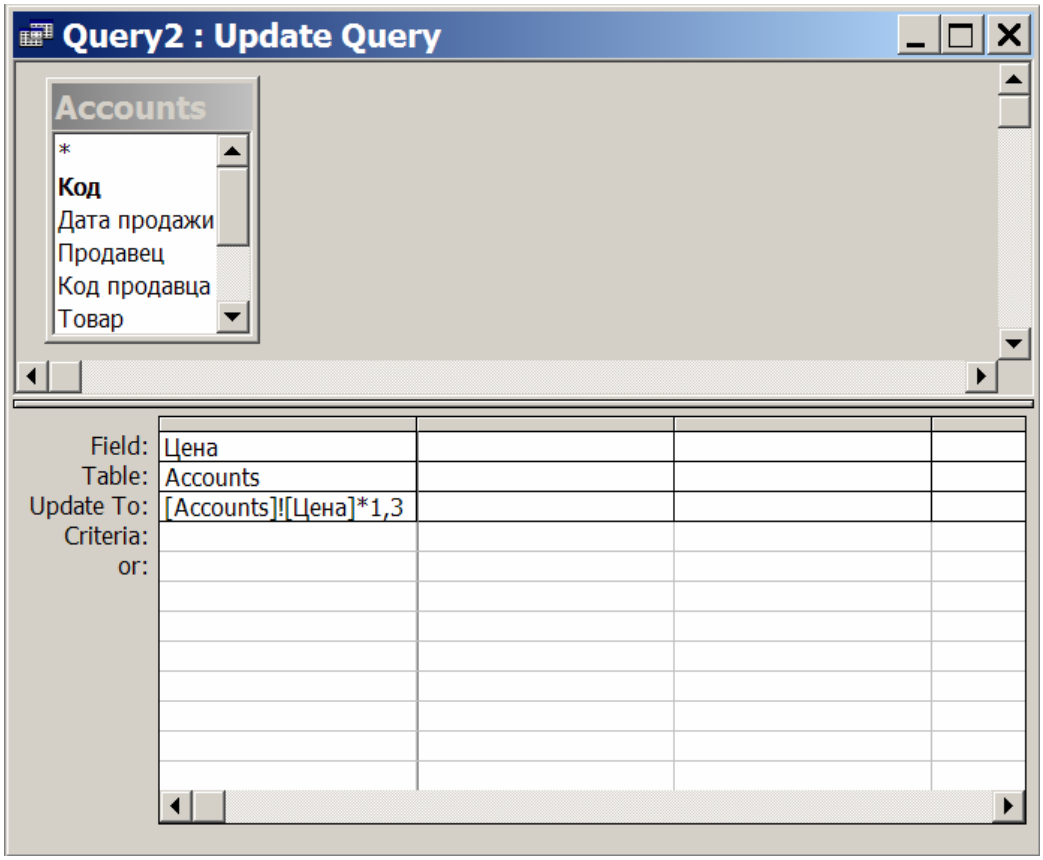


Рисунок 7.14 – Збільшення ціни всіх товарів

Запит, зображений на рис. 7.15, змінить ціну одного конкретного товару (див. умову відбору даного запиту) і встановить для неї значення, зазначене в запиті.

При виконанні коригувальних запитів система здійснює контроль обмежень цілісності. Так, наприклад, якщо при описі таблиці було задано обмеження на максимально припустиме значення поля «Кількість», то при виконанні запиту, у випадку порушення обмеження може бути видане попереджуюче повідомлення. Якщо Ви виберете варіант «Yes», то для записів, що порушують обмеження на значення, коректування взагалі виконане не буде. Іноді такий вибір може бути неправильним, тому що, наприклад, в одних товарів ціна буде підвищена, а в інших – ні. Причому потім розібратися, у чого вона була підвищена, а в чого – ні, буде практично неможливо. У розглянутій ситуації варто відмовитися від виконання такого запиту, скорегувати при необхідності обмеження на значення і тільки після цього виконати запит.

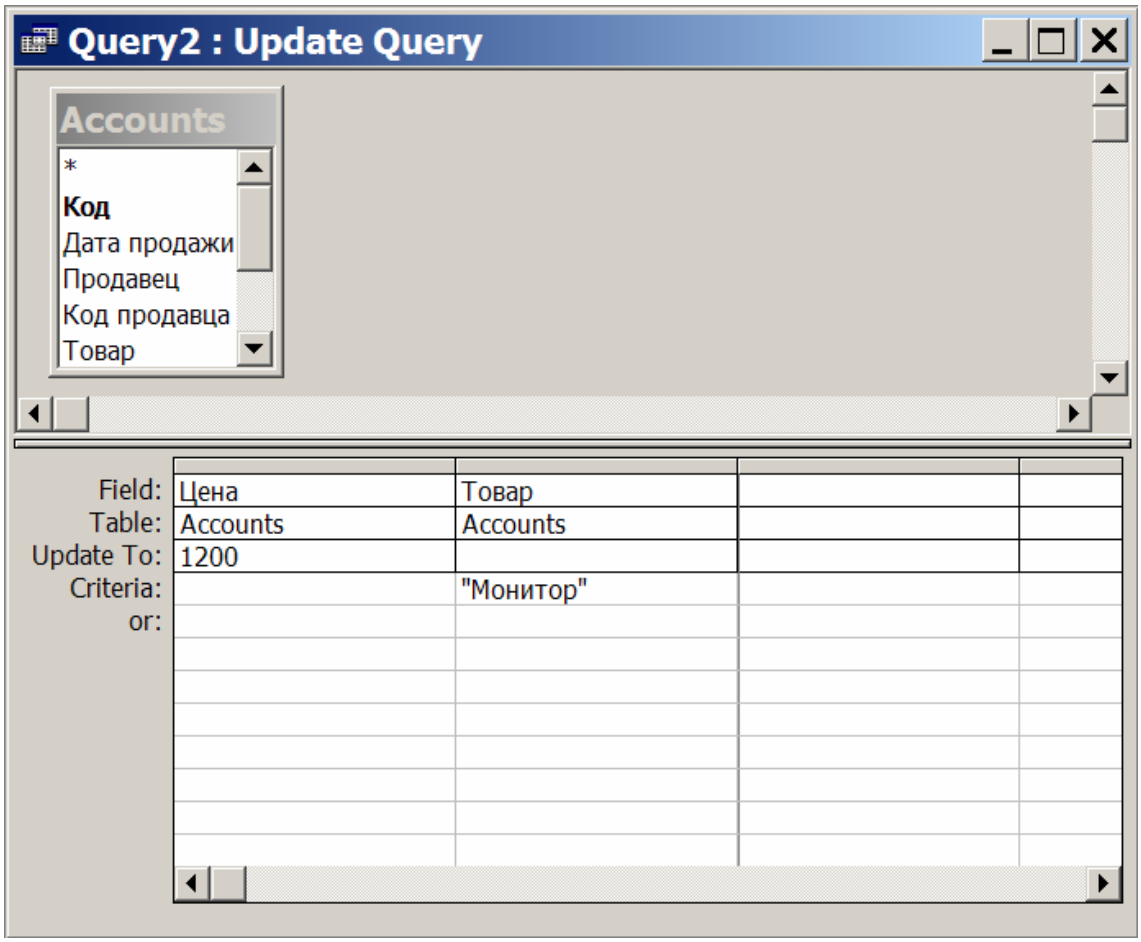


Рисунок 7.15 – Установка ціни конкретного товару

Розглянемо тепер запити на видалення й додавання. При виборі запиту на видалення в бланку запиту в рядку «Criteria», так само, як і у запитах на вибірку, задається умова. Записи, що задовольняють умові, буде вилучено з бази даних. Для того, щоб бути впевненим, що запит задано вірно, і, як наслідок, будуть вилучені саме ті записи, які необхідно, рекомендується спочатку задати запит на вибірку, подивитися отриманий результат, а потім змінити тип запиту, вибравши запит на видалення.

Якщо в «запиті на видалення» не задано ніяких умов відбору, то з таблиці будуть вилучені всі записи.

Запит на додавання додає групу записів з однієї або декількох таблиць у кінець однієї або декількох таблиць. Для задавання запиту такого типу треба спочатку створити запит, що містить таблицю, записи з якої необхідно додати в іншу таблицю. Потім у режимі Конструктора запиту треба натиснути

стрілку поруч із кнопкою «Query Type» на панелі інструментів і вибрати команду «Append Query» (або вибрати відповідну позицію в меню «Query»). На екрані з'явиться діалогове вікно «Append». У поле «Table Name» треба ввести ім'я таблиці, у яку необхідно додати записи.

Таблиця, у яку здійснюється додавання, може бути як у тій же базі даних, так і в іншій, причому це не обов'язково повинна бути база даних MS Access. Зі списку полів у бланк запиту треба перемістити поля, які необхідно додати, а також ті, які будуть використані при визначенні умови відбору. Якщо всі поля в обох таблицях мають однакові імена, то можна просто перемістити за допомогою миші в бланк запиту символ «зірочка». Однак, при цьому, навіть якщо структури обох таблиць повністю збігаються, можуть виникнути проблеми із ключами (якщо ключове поле має тип «лічильник», то для автоматичного додавання значень лічильника не треба при створенні запиту переміщати поле лічильника в бланк запиту). Якщо в обох таблицях виділені поля мають однакові імена, відповідні імена автоматично вводяться в рядок «Append to». Якщо імена полів двох таблиць відмінні одне від одного, в рядок «Append to» треба ввести імена полів, що додаються у таблицю.

Питання й тести для самоконтролю

1. Що таке запит до бази даних?
2. Які Ви знаєте типи запитів?
3. Як створити новий запит у СУБД MS Access?
4. Як додати в запит таблицю?
5. Що таке умови відбору і які вони бувають?
6. Як створити простий запит у СУБД MS Access?
7. Опишіть алгоритм створення складного запиту.
8. У чому полягають особливості запиту до зв'язаних таблиць?
9. Продемонструйте можливості запитів з полями, що обчислюються.
10. Розкажіть про коригувальні запити.

Розділ 8. СКЛАДАННЯ ЕКРАННИХ ФОРМ ТА ЗВІТІВ

Екранні форми можна розглядати як деякі маски, через які користувач розглядає поля запису бази даних. Маска приховує від користувача непотрібні йому в цей момент поля. Користувач може створити екранні форми, у яких поля розміщено по всьому полю екрана в зручному йому порядку. На його вимогу у форму можуть бути інтегровані й інші елементи (написи, командні кнопки, селекторні кнопки, контрольні індикатори, списки, ілюстрації і т.п.). Форми можна розфарбувати будь-якими доступними фарбами, використати для оформлення растри і графічні елементи (лінії й прямокутники).

Екранні форми призначені для того, щоб полегшити користувачеві введення даних у базу і як додаткову інформацію можуть містити тексти допомоги, що відповідають на питання, як ввести дані в деяке поле, а також зразки заповнення полів. Форми повинні бути доступними для огляду й наочними, в іншому випадку втрачається їхній сенс.

Форми – один з основних засобів для роботи з базами даних у СУБД MS Access. Вони використовуються для введення нових записів (рядків таблиць), перегляду й редагування вже наявних даних, задавання параметрів запитів і виводу відповідей на них та ін. Форми виглядають як прямокутні вікна з розміщеними в них елементами керування. Існує можливість створення форм динамічно при виконанні програми, однак природним режимом їхнього створення є режим візуального конструювання.

Звіт – це форматоване зображення даних, що виводиться на екран, друк або файл на диску. Звіти дозволяють видобути з бази потрібні відомості й відобразити їх у вигляді, зручному для сприйняття, а також надають широкі можливості для узагальнення й аналізу даних. При друку таблиць і запитів інформація видається практично в тому вигляді, у якому зберігається. Часто виникає необхідність перегрупувати дані у вигляді звітів, які мають традиційний вигляд і легко читаються. Докладний звіт включає всю інформацію з

таблиці або запиту, але містить заголовки й розбитий на сторінки з додаванням верхніх і нижніх колонтитулів.

Звіти, як і форми, є головними елементами інтерфейсу баз даних. Вони дозволяють виводити на друк інформацію про вміст бази в зручному для користувача вигляді. Більша частина звітів створюється на етапі проектування офісного додатка.

MS Access відображає у звіті дані з запиту або таблиці, додаючи до них текстові елементи, які спрощують його сприйняття. До числа таких елементів відносяться:

– заголовок. Цей розділ друкується тільки у верхній частині першої сторінки звіту. Використовується для виводу даних, таких як текст заголовка звіту, дата або констатуюча частина тексту документа, які варто надрукувати один раз на початку звіту;

– верхній колонтитул. Використовується для виводу даних, таких як заголовки стовпців, дати або номери сторінок, що друкуються зверху на кожній сторінці звіту;

– область даних, розташована між верхнім і нижнім колонтитулами сторінки. Містить основний текст звіту. У цьому розділі з'являються дані, що роздруковують для кожного з тих записів у таблиці або запиті, на яких засновано звіт;

– нижній колонтитул. Цей розділ з'являється в нижній частині кожної сторінки. Використовується для виводу даних, таких як підсумкові значення, дати або номери сторінки, що друкуються знизу на кожній сторінці звіту;

– примітка. Використовується для виводу даних, таких як текст висновку, загальні підсумкові значення або підпис, які варто надрукувати один раз наприкінці звіту.

Різноманіття можливостей формування різних типів форм і звітів у сучасних розвинених СУБД, гнучке настроювання їхнього вигляду, багатий набір елементів керування роблять практично неможливим докладний розгляд цих складових банку даних у даному навчальному посібнику. Тому в цьому

розділі розглядаються тільки найбільш уживані можливості СУБД MS Access з розробки екранних форм і звітів.

8.1. Створення нескладних форм за допомогою Майстра

Форму можна створити декількома різними способами. Для того, щоб створити нову екранну форму, треба вибрати вкладку «Forms» у вікні бази даних і натиснути кнопку «New». Після цього з'явитися вікно «New Form» (рис. 8.1).

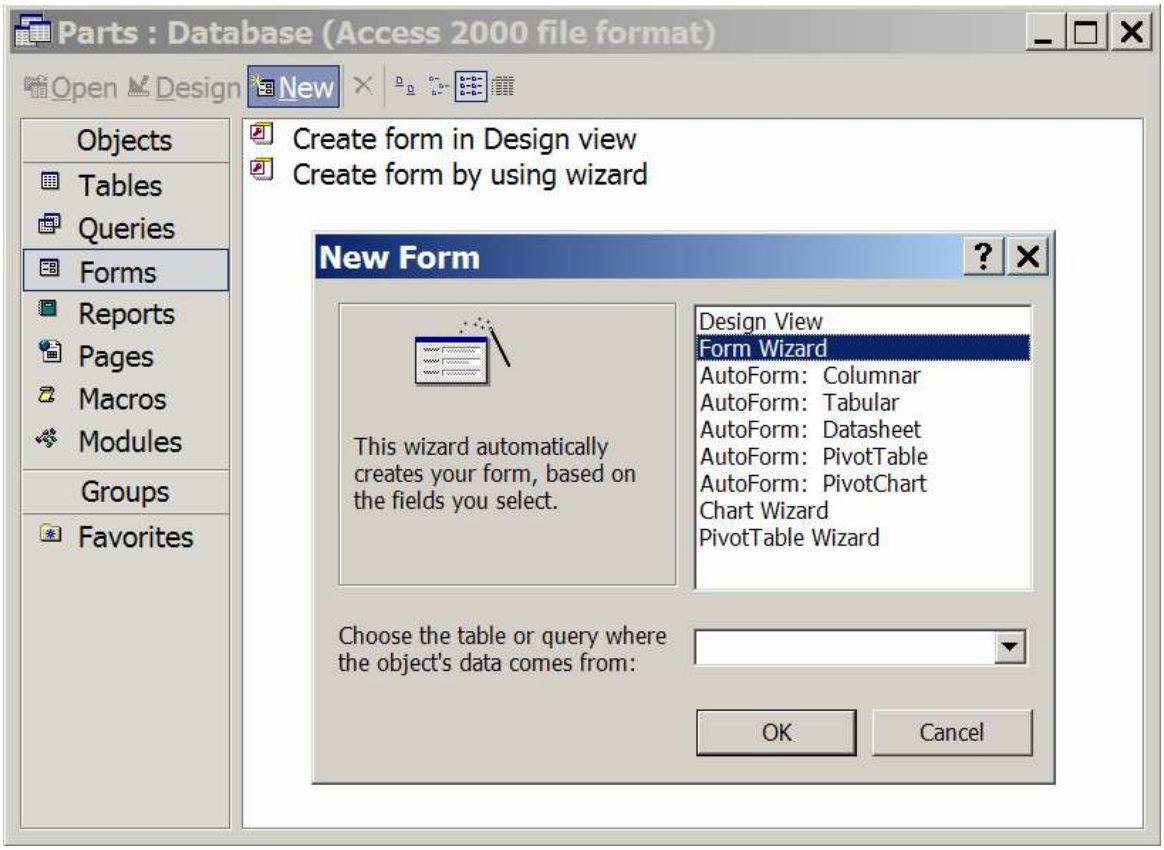


Рисунок 8.1 – Початкове вікно для створення форми

При створенні форми, пов'язаної з таблицями бази даних, краще спочатку скористатися можливостями **Майстра форм**, а потім модифікувати одержану в такий спосіб форму. Це прискорить і спростить процес створення форм. У вікні «New Form» крім вибору способу створення та виду форми можна визначити й джерело даних для створюваної форми (рис. 8.1). З вікна

«New Form» можна вибрати тільки одну таблицю як джерело даних для форми.

Як приклад, створимо форму для таблиці «Accounts». Вибір джерела можна здійснити й на наступному кроці (рис. 8.2) у віконці «Tables/Queries». Якщо джерелом повинні бути кілька таблиць, то можна скористатися декількома різними способами, які докладно розглянуті в літературі [1].

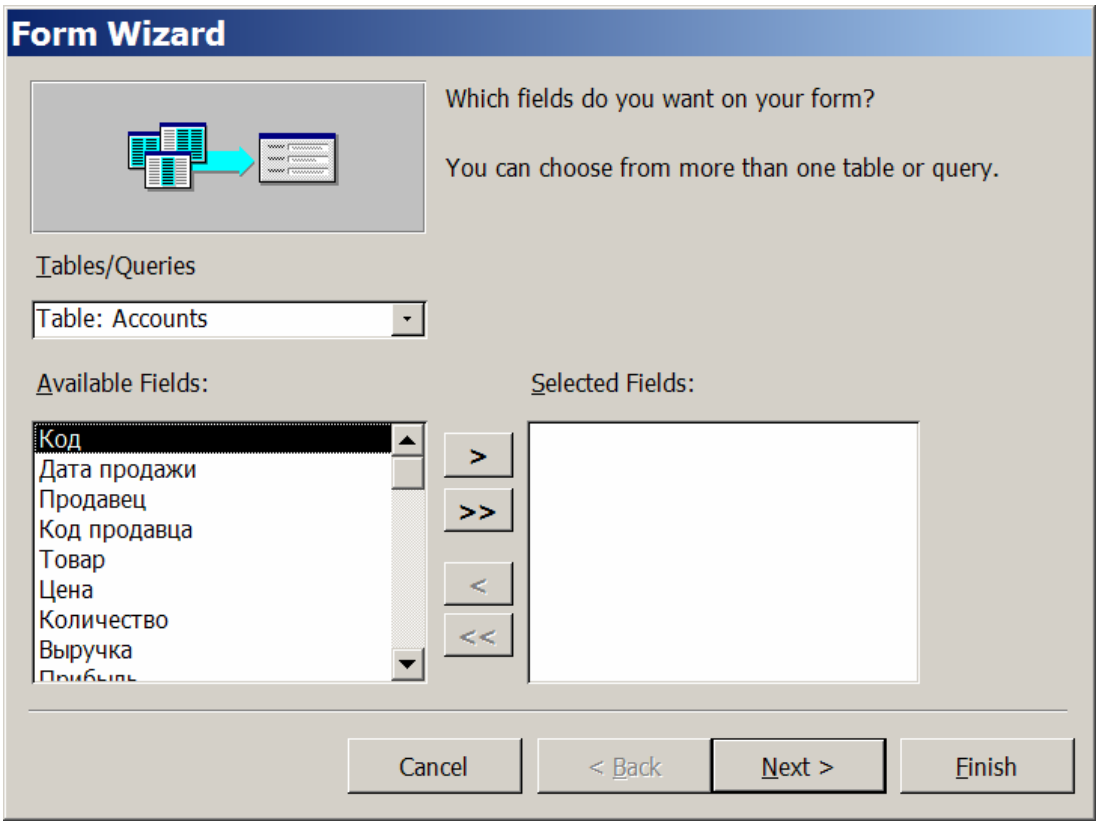


Рисунок 8.2 – Вибір полів для включення у форму

Наступним кроком при створенні форм є вибір полів, які будуть включені в екранну форму (рис. 8.2). Поля можуть переноситися у форму по одному або всі відразу. У першому випадку треба позиціюватися на потрібне поле й натиснути кнопку з одинарною стрілкою, спрямованої вправо. Щоб перенести всі поля, треба скористатися кнопкою з подвійною стрілкою. Коли поля помилково включені у форму, на етапі визначення складу полів їх можна виключити, скориставшись кнопками зі стрілками, спрямованими вліво.

При використанні Майстра для створенні форми, вибір таблиці та хоча б одного поля обов'язкові.

Джерелом даних для форми можуть бути не тільки таблиця або таблиці, але й запити. Нагадаємо, що в запитах можна використовувати знак «зірочка». При його використанні всі поля вихідної таблиці виводяться у відповідь. Якщо вже після створення такого запиту в таблицю будуть додані нові поля, то вони потраплять у відповідь. Якщо ж на основі такого запиту створена екранна форма, то в неї ввійдуть ті поля, які на момент створення екранної форми розташовувалися в таблиці.

Поля, які вставлено в таблицю після створення екранної форми, у запит, що є джерелом, будуть попадати, але в екранній формі відбиватися не будуть. Тому, якщо необхідно, щоб всі поля вихідної таблиці були відображені у формі, то створену форму доведеться коректувати вручну. Припустимо, що ми хочемо створити форму на основі таблиці «Accounts» і включити в цю форму всі поля вихідної таблиці (рис. 8.2).

Після визначення складу полів, включених у форму, варто вибрати зовнішній вигляд форми (рис. 8.3). Зручніше спочатку вибрати вигляд «Columnar», а потім розмістити елементи форми на екрані так, як це зручно для наступної роботи з формою. Назва виду форми «Columnar» трохи умовна. Дійсно, MS Access при такому виборі створює екран «анкетної» форми: один під іншим розташовуються пари «назва поля/зміст поля». Але коли полів багато, і всі вони не вміщаються в один стовпець на екрані, то система автоматично поміщає їх у два, три і т.д. стовпці.

Наступним кроком є вибір стилю форми (рис. 8.4). Стиль екранної форми дозволяє вибрати кольори й «фактуру» форми, тобто дозволяє вирішити чисто оформлювальні проблеми.

Створення форми завершується задаванням її імені (рис. 8.5). За замовчуванням дається ім'я, що збігається з ім'ям джерела даних.

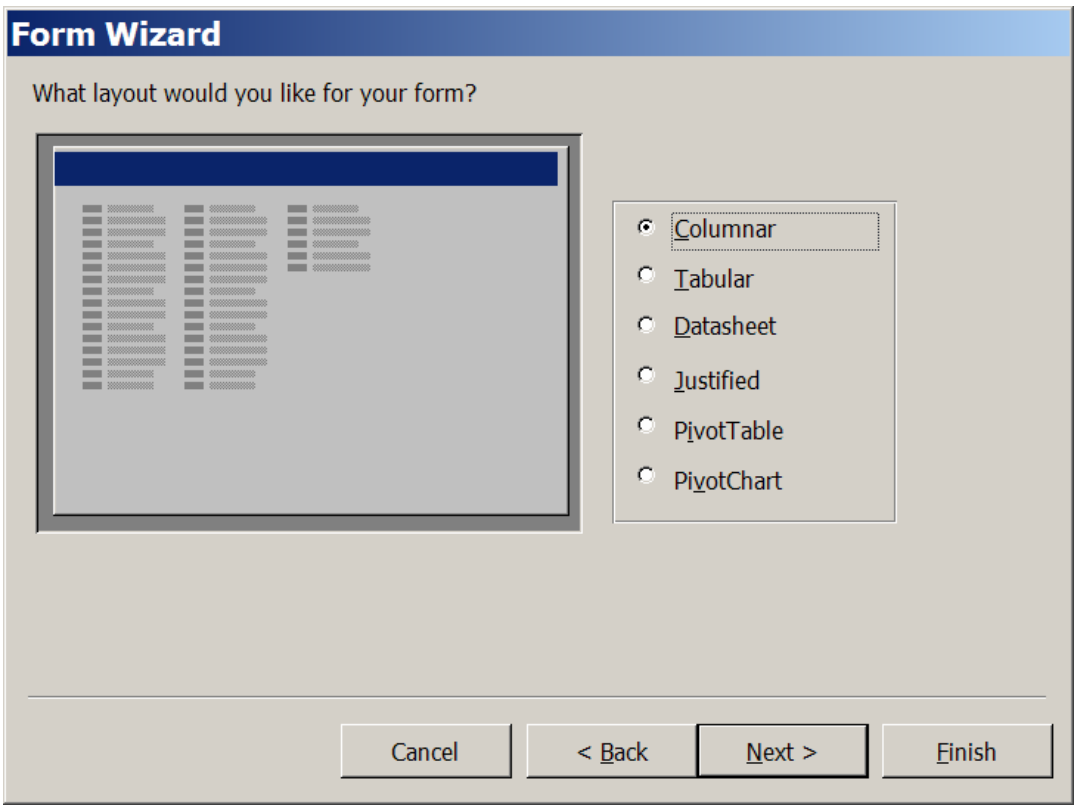


Рисунок 8.3 – Вибір дизайну форми

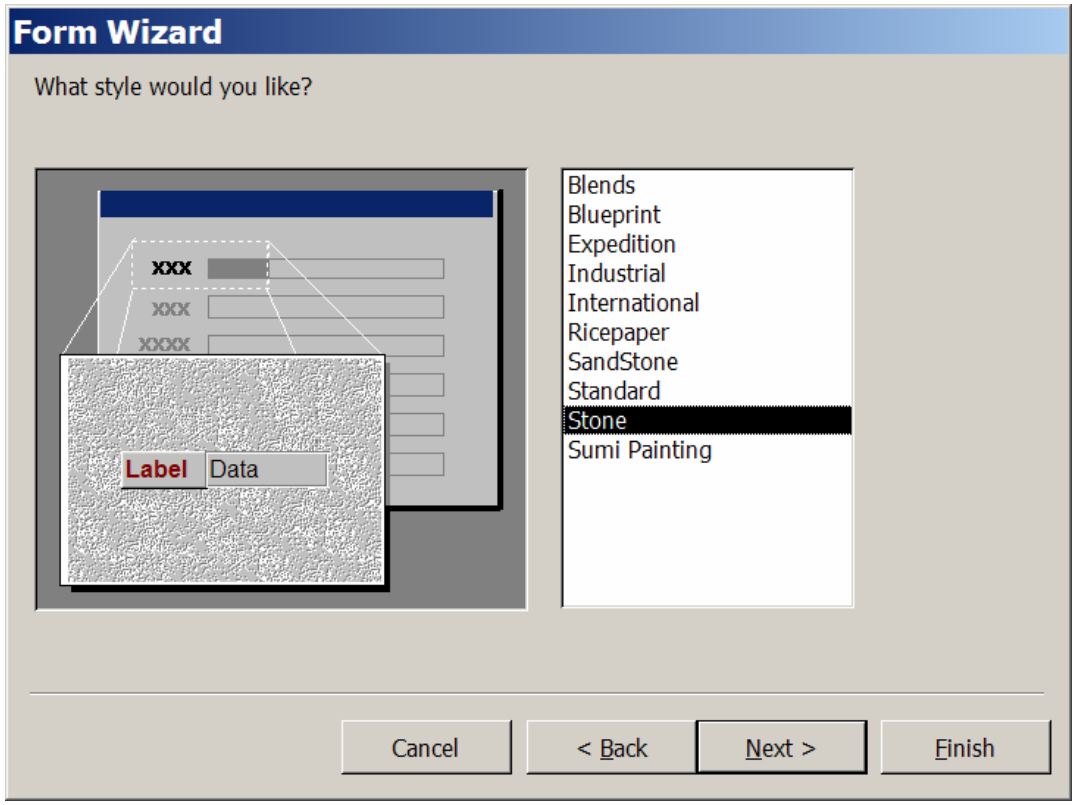


Рисунок 8.4 – Вибір стилю форми

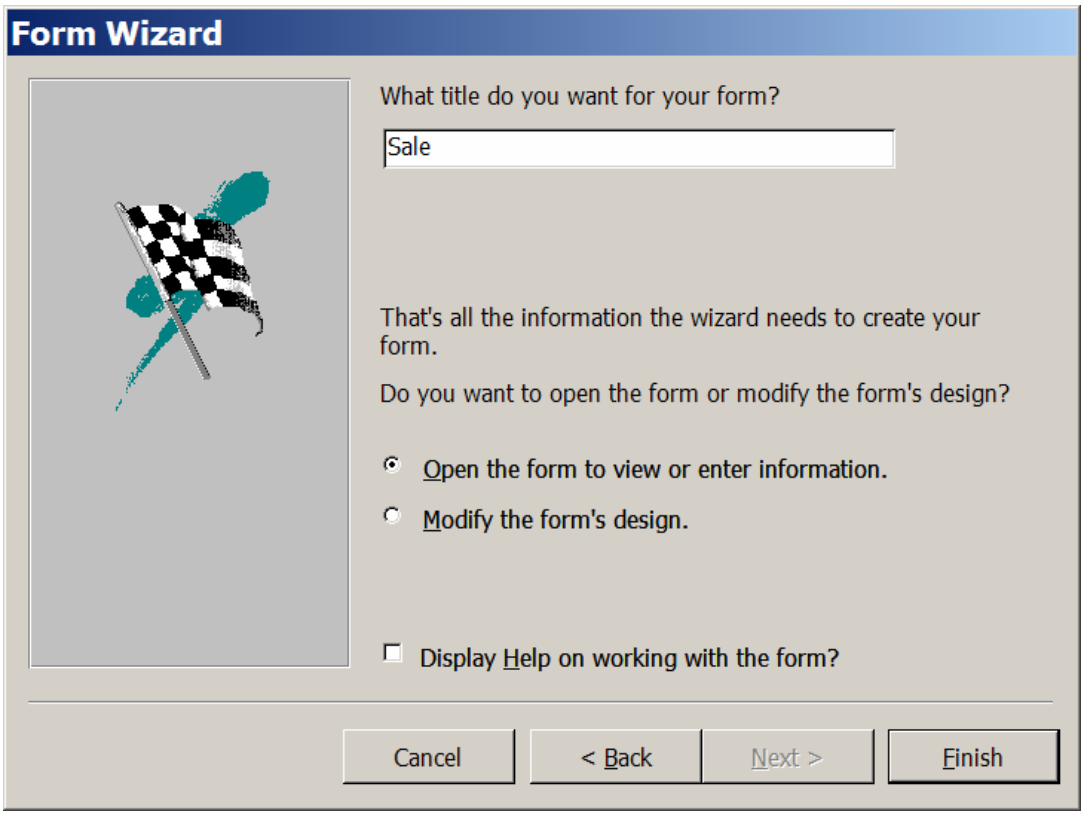


Рисунок 8.5 – Вибір імені форми

Створена екранна форма може не повністю відповідати потребам розробника. Щоб її підправити, можна відразу перейти в режим Конструктора, вибравши перемикач «Modify the form's design», а можна це зробити потім, перейшовши в режим Конструктора з режиму форми, або відкривши раніше створену форму в режимі Конструктора.

8.2. Доробка форм у режимі Конструктора

Як вже говорилося, форма, створена Майстром, може бути скоректована. Для цього треба одним із зазначених раніше способів перейти в режим **Конструктора форм**. Екранна форма, одержана за допомогою Майстра, у режимі Конструктора буде мати вигляд, показаний на рис. 8.6.

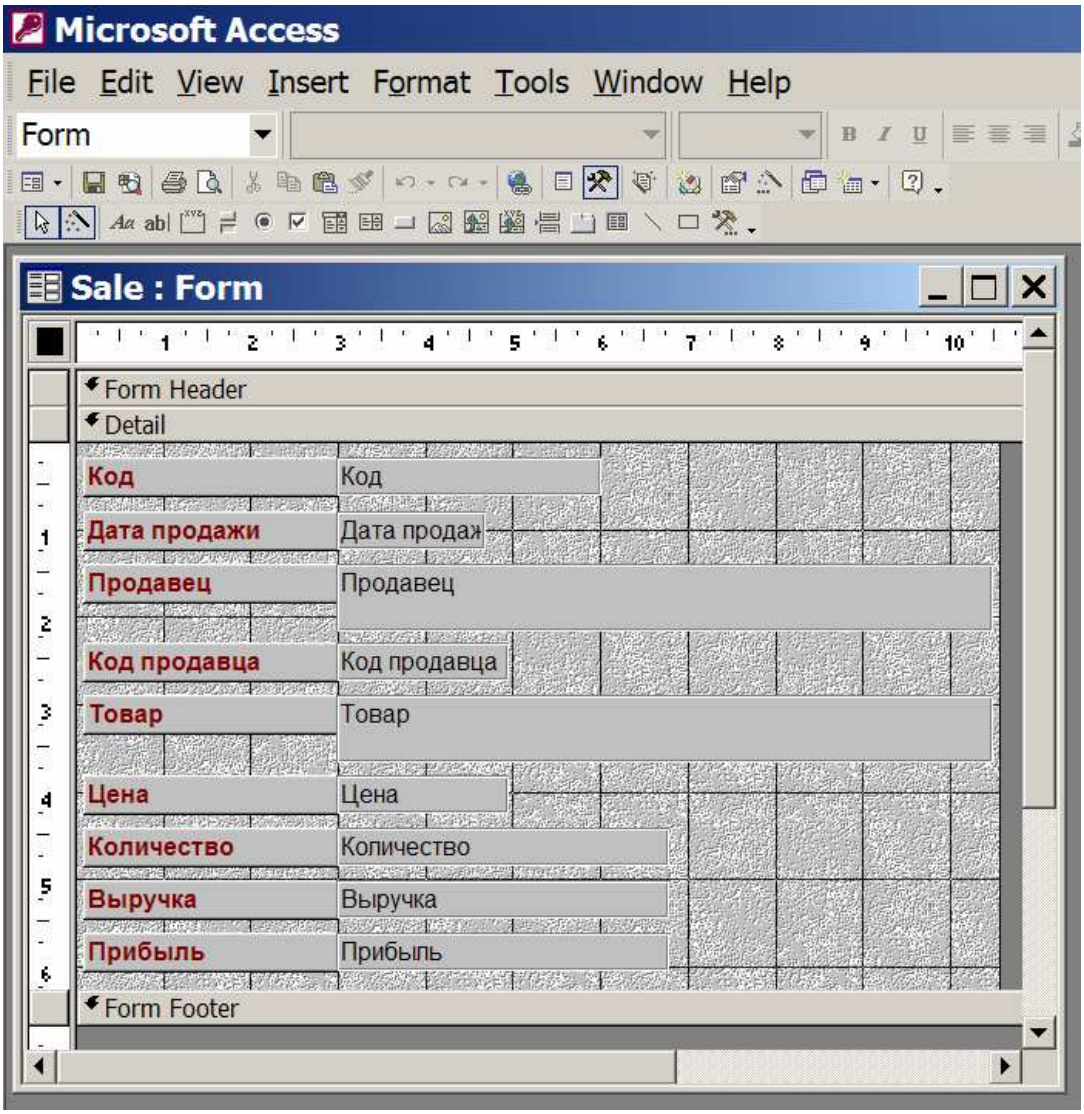


Рисунок 8.6 – Вид форми в режимі Конструктора

У верхній частині екрана, крім головного меню, видно три інструментальних панелі: форматування («Formatting»), конструктора форм («Form Design») і панелі елементів («Toolbox»). Перша з панелей стандартна для всіх офісних додатків Microsoft. Більшість з елементів другої панелі також зрозумілі. Деякі зі специфічних кнопок та їхнє застосування будуть пояснені пізніше.

Як видно з рис. 8.6, всі елементи одержаної форми розміщено в області даних. Область заголовка форми («Form Header») закрита, але її можна розкрити й ввести в неї заголовок форми й інші дані, що відносяться до всієї форми. Тому що призначення й способи роботи з областями форм і звітів одна-

кові, а їхнє використання у звітах більш актуальне, ці питання розглянуто в другій частині даного розділу навчального посібника, присвяченій звітам.

8.2.1. Зміна включених у форму елементів керування

Зміни, що вносяться у вихідну форму, можуть бути різноманітними. Насамперед, існує можливість переміщення, зміни розмірів і вирівнювання вже включених у форму елементів керування. Для цього елементи, які треба змінити, повинні бути виділені. Для виділення елемента керування треба встановити на нього покажчик і натиснути кнопку миші. Щоб виділити кілька елементів керування, варто натиснути клавішу Shift і, не відпускаючи її, виділити всі потрібні елементи. Якщо виділювані елементи перебувають поруч, і їх не розділяють ніякі елементи, які не повинні входити у виділювану групу, то можна натиснути ліву клавішу миші й, не відпускаючи її, охопити контуром, що з'явився, всі ті елементи, які треба виділити. Виділені елементи мають характерні маркери по кутах елемента й по серединах його сторін.

Щоб перемістити виділені елементи, треба позиціюватися на них мишею й домогтися, щоб покажчик набув форму долоні. Потім, тримаючи натиснутою ліву клавішу миші, перемістити елементи на необхідне місце. У цьому випадку переміщатися будуть всі виділені елементи. Якщо з пари «підпис-елемент керування» треба перемістити щось одне, то слід позиціюватися на лівий верхній кут потрібного елемента й домогтися, щоб покажчик прийняв форму вказівного пальця: у цьому випадку буде пересуватися тільки цей елемент.

Щоб змінити розмір елемента, треба позиціюватися на границі елемента таким чином, щоб покажчик прийняв форму двонаправленої стрілки. Вибір напрямку стрілок (вгору-вниз, вправо-вліво, по діагоналі) залежить від того, як ви хочете змінити розмір елемента.

Щоб вирівняти виділені елементи керування, у меню «Format» треба вибрати команду «Align», а потім у списку, що з'явився, вибрати спосіб вирівнювання.

Щоб видалити виділені елементи керування, треба натиснути клавішу «Del».

8.2.2. Включення нових елементів у форму

Якщо спочатку у форму були включені не всі поля з таблиці (або Ви випадково видалили потрібний елемент), неважко додати поля з таблиці, що є джерелом даних для форми. Для цього треба натиснути кнопку «Field List» на панелі інструментів. Із спадаючого списку, що з'явився, треба вибрати ім'я потрібного поля, позиціювавшись на ньому, натиснути ліву клавішу миші й, не відпускаючи її, перемістити елемент у потрібне місце форми.

Більш складним є варіант, коли для цих цілей використовується кнопка «Text Box» на панелі елементів, а потім у вставленого в результаті цієї дії вільного елемента міняється відповідним чином властивість «Control Source», а в його напису – властивість «Caption». Але такий спосіб краще використовувати тільки в тому випадку, коли інший шлях неможливий, наприклад, при виводі у форму обчислюваного поля.

При створенні поля, що обчислюється, у властивість «Control Source» треба ввести вираз для обчислення значення цього поля. Поля, що обчислюються, можуть вводитися не тільки в ті форми, які використовуються для виводу інформації, але й у ті, які використовуються для введення даних у базу даних. Наприклад, при введенні даних у таблицю «Sellers», можна на екран вивести поле, що обчислюється, «Вік»: це поле не буде зберігатися в таблиці (у неї буде вводитися тільки «Дата народження»), а на екран автоматично при введенні дати народження буде виводитися вік, що зручно, наприклад, для контролю даних, що вводять.

Щоб ввести у форму текст, треба натиснути кнопку «Label» на панелі елементів й, не відпускаючи кнопку миші, переміститися в те місце у формі, куди варто помістити текст. Після чого ввести потрібний текст і натиснути клавішу Enter. Крім тексту й полів, у форму можуть бути включені лінії, квадрати, малюнки.

8.2.3. Зміна типу елемента керування

В екранній формі можуть використовуватися різні елементи керування, у тому числі список, поле зі списком, які широко вживаються при створенні екранних форм. Якщо при побудові таблиці поля були створені як поля підстановки, то у формі, отриманій в результаті використання Майстри, цим полям будуть відповідати поля зі списком.

Якщо поля були створені як звичайно, а Ви хочете у формі використати, наприклад, поле зі списком, то можна підійти до бажаного результату різними способами. По-перше, можна змінити тип елемента керування у формі. Для цього в режимі Конструктора треба виділити відповідний елемент форми, клацнути правою клавішею миші, і в контекстному меню, що з'явився, вибрати позицію «Change to», як показано на рис. 8.7. Доступні перетворення будуть залежати від того, який тип має обраний елемент.

Існує кілька причин, з яких буває необхідно рознести дані, розташовані в екранній формі, по декількох сторінках (так звана **багатосторінкова форма**). Це може бути, наприклад, у випадку, коли елементів у формі занадто багато, і розміщення всіх їх на одному екрані надто захаращує його; або може бути викликано бажанням згрупувати поля й показувати кожну групу окремо і т.п.

Для досягнення цих цілей можна або, скориставшись елементом керування «Page Break», вказати, у яких місцях повинен бути перехід на наступну сторінку, або створити екранну форму з декількома вкладками. Елемент керування «Page Break» використовується для вказівки горизонтальних розри-

вів між елементами керування у формі. Для переходу до сторінки, що перебуває над або під зазначеним розривом, використовуються клавіші PgUp або PgDn.

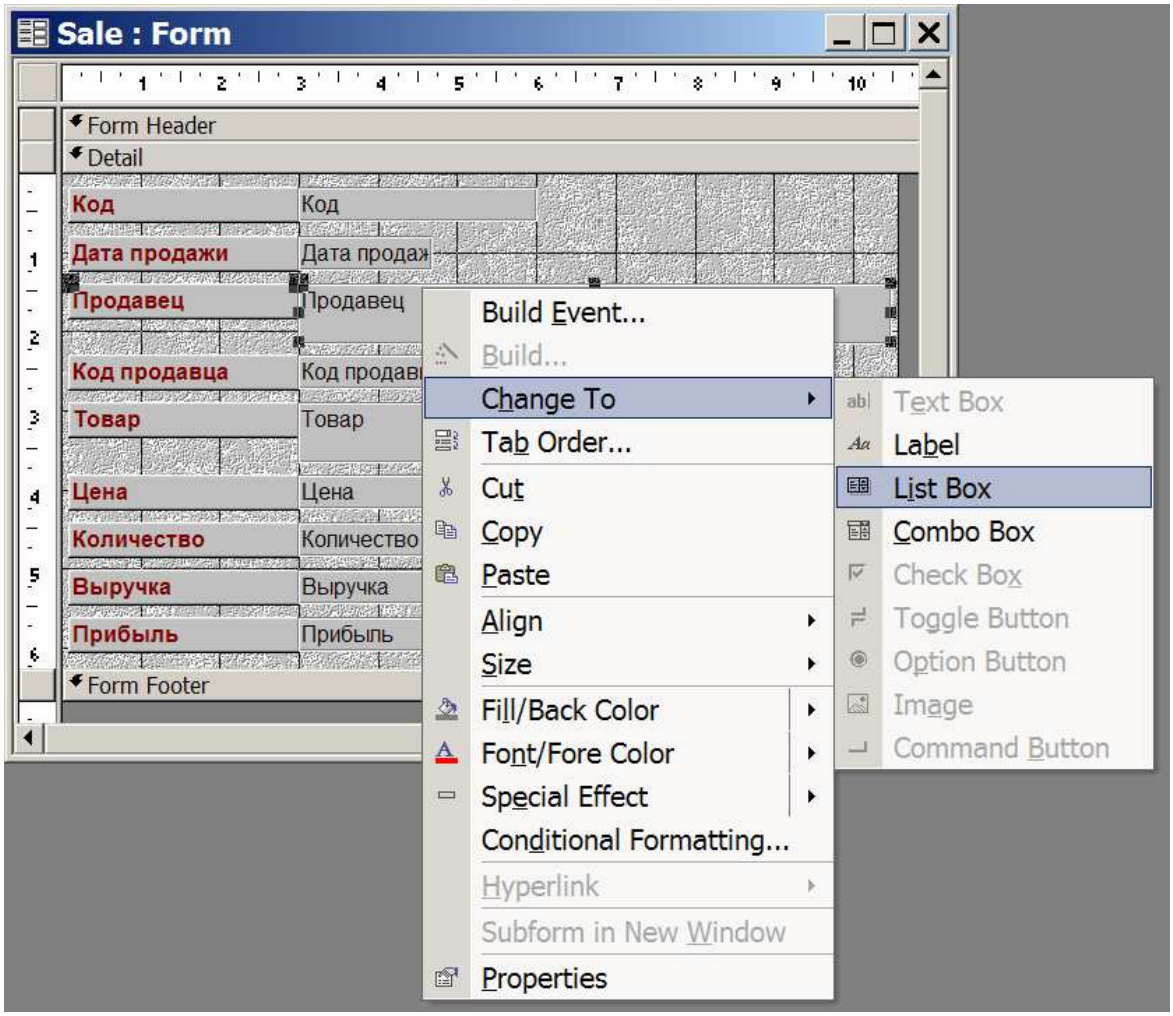


Рисунок 8.7 – Зміна типу елемента керування

8.2.4. Створення форм, що складаються з декількох сторінок

Для того, щоб створити форму з декількома вкладками, можна скористатися елементом керування «Tab Control». При виборі цієї можливості у формі створюються дві вкладки. У кожну з них можна перемістити ті елементи, які необхідно. Якщо необхідно створити більше число вкладок, то треба, перебуваючи в зоні вкладок натиснути праву клавішу миші й у контекстному меню, що з'явилося, вибрати позицію «Insert Page» (рис. 8.8).

Для включення елементів потрібно вибрати необхідну вкладку й натиснути кнопку «Field List» на панелі інструментів. Зі спадаючого списку треба вибрати ім'я потрібного поля, позиціювавшись на ньому, натиснути ліву клавішу миші й, не відпускаючи її, перетягти елемент у потрібне місце вкладки.

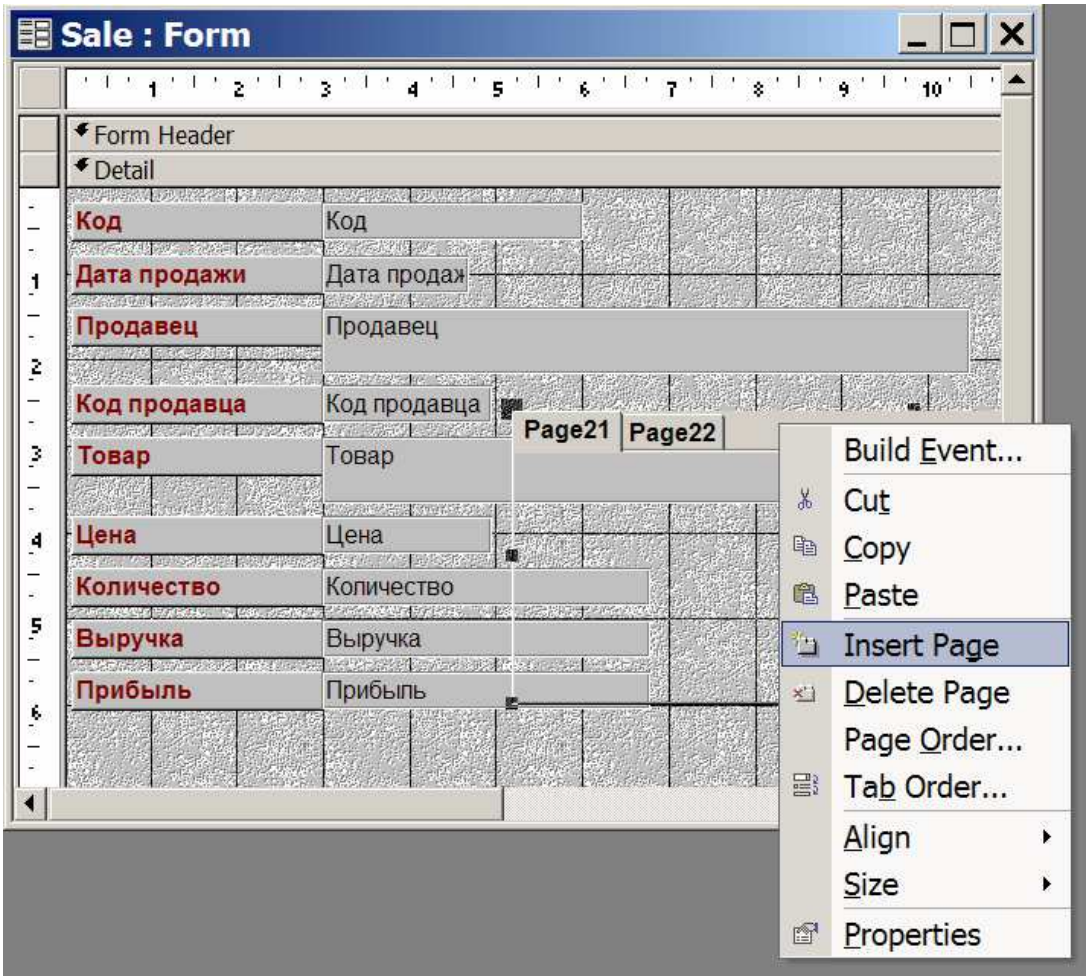


Рисунок 8.8 – Додавання вкладок

Для того, щоб змінити назву вкладки, треба скористатися правою кнопкою миші, у контекстному меню, що з'явилося, вибрати позицію «Properties» та у властивості «Name» записати необхідну назву (рис. 8.9).

8.2.5. Послідовність обходу полів

Послідовність обходу полів при роботі з формою може відрізнятися від їхнього розташування на екрані. Для встановлення послідовності обходу по-

лів можна, позиціювавшись на заголовок вікна форми в режимі Конструктора, натиснути на праву клавішу миші. При цьому з'явиться спливаюче вікно (рис. 8.10), у якому слід вибрати позицію «Tab Order». Після чого на екрані з'явиться вікно (рис. 8.11), у якому перераховано поля, включені у форму (якщо форма містить кілька вкладок, то будуть виводитися тільки ті елементи, які включено в «активну» вкладку). Залишається перетягнути їх, створивши потрібну послідовність, за допомогою миші.

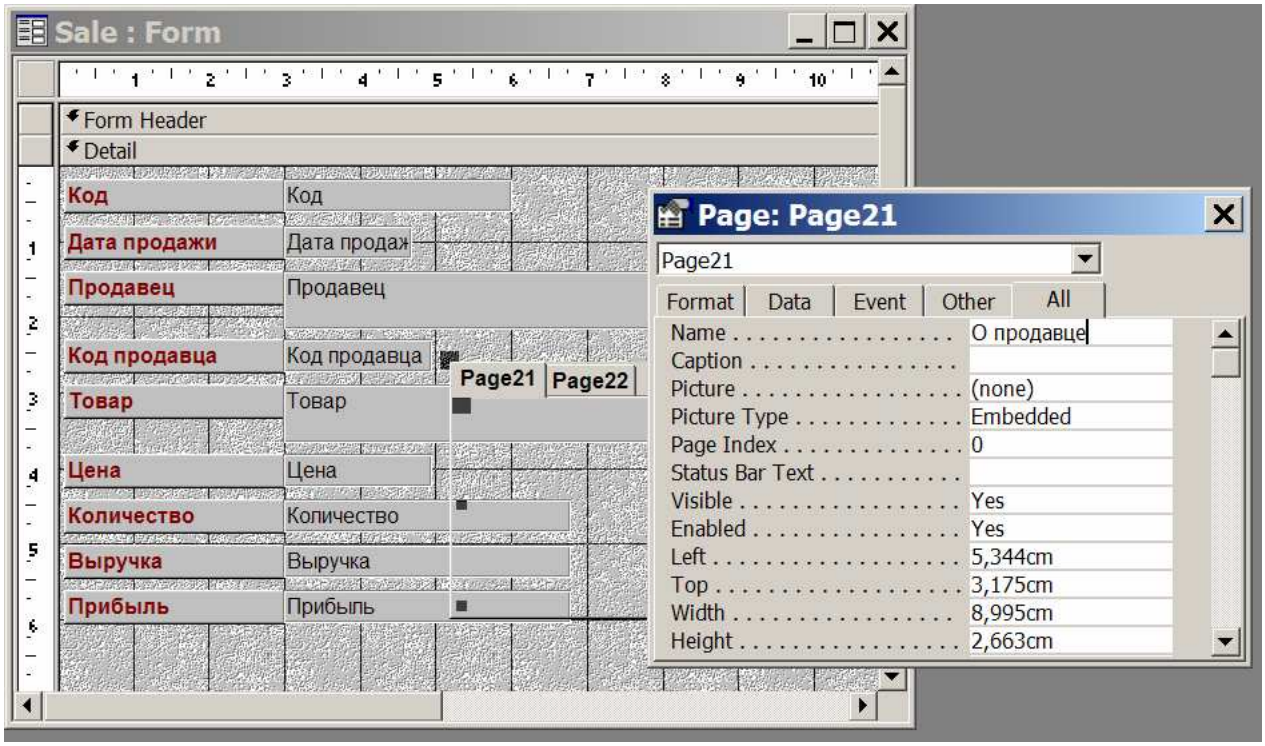


Рисунок 8.9 – Зміна назви вкладки

8.2.6. Властивості форми

Крім властивостей, що відносяться до кожного окремого елемента форми, є властивості, що відносяться до всієї форми. Їх можна з успіхом використовувати для створення додаткових зручностей при роботі з формою, для забезпечення цілісності бази даних та інших цілей. Для того, щоб подивитися й скорегувати властивості форми, треба відкрити форму в режимі Конструктора й подвійним натисканням кнопки миші на «тло» під формою відкрити

вікно властивостей форми.

Перелік властивостей форми досить великий. Розглянемо деякі з них. Так, на вкладці «Data» (рис. 8.12) є властивість «Data Entry». Якщо вибрати для нього значення «Yes», можна створити форму, що використовується тільки для введення даних. У цьому випадку в формі буде висвічуватися один порожній запис, у який можна вводити нові дані. Якщо Ви хочете, щоб виводилися всі записи, то значення цієї властивості повинне бути «No». Можна, створити форму тільки для перегляду, заборонивши всі коректування.

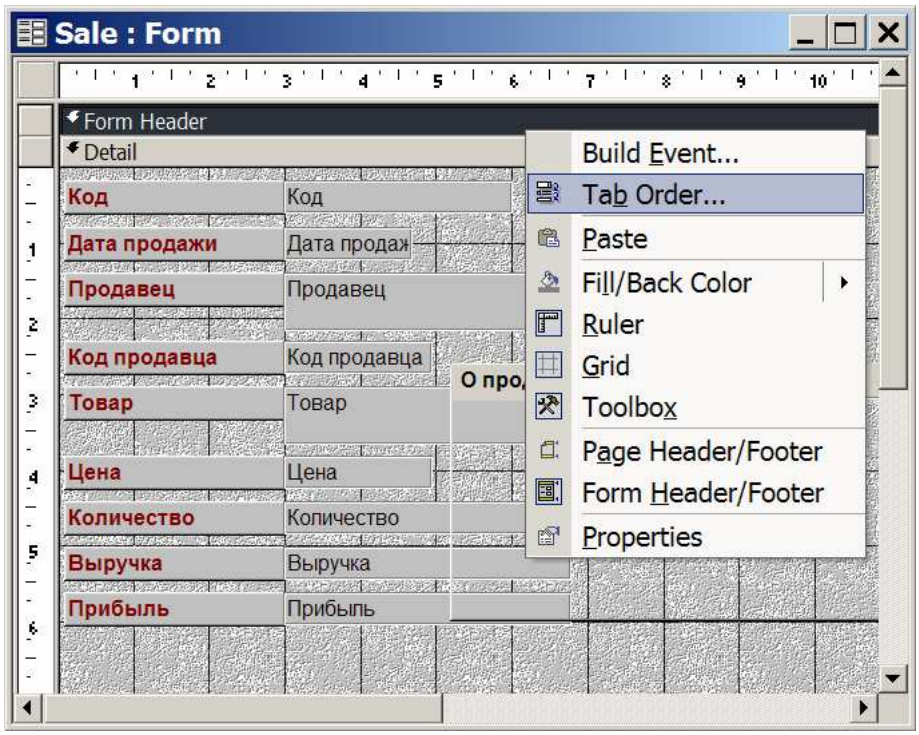


Рисунок 8.10 – Зміна послідовності обходу полів

8.3. Можливі причини помилок

Якщо в таблиці, що була джерелом даних для форми, буде перейменовано або вилучено поле, включене у форму, то при виводі форми у відповідному даному полю віконці буде виводитися рядок «#Name?». Це являє приклад порушення цілісності бази даних, коли був змінений один елемент бази даних і не змінений пов'язаний з ним інший елемент (у даному випадку – по-

ле таблиці бази даних і заснований на ньому елемент екранної форми). Якщо поле було перейменовано, то для відповідного елемента форми треба скорегувати властивість «Control Source». Якщо поле було вилучене з таблиці, елемент, що відповідав йому, треба видалити з форми.

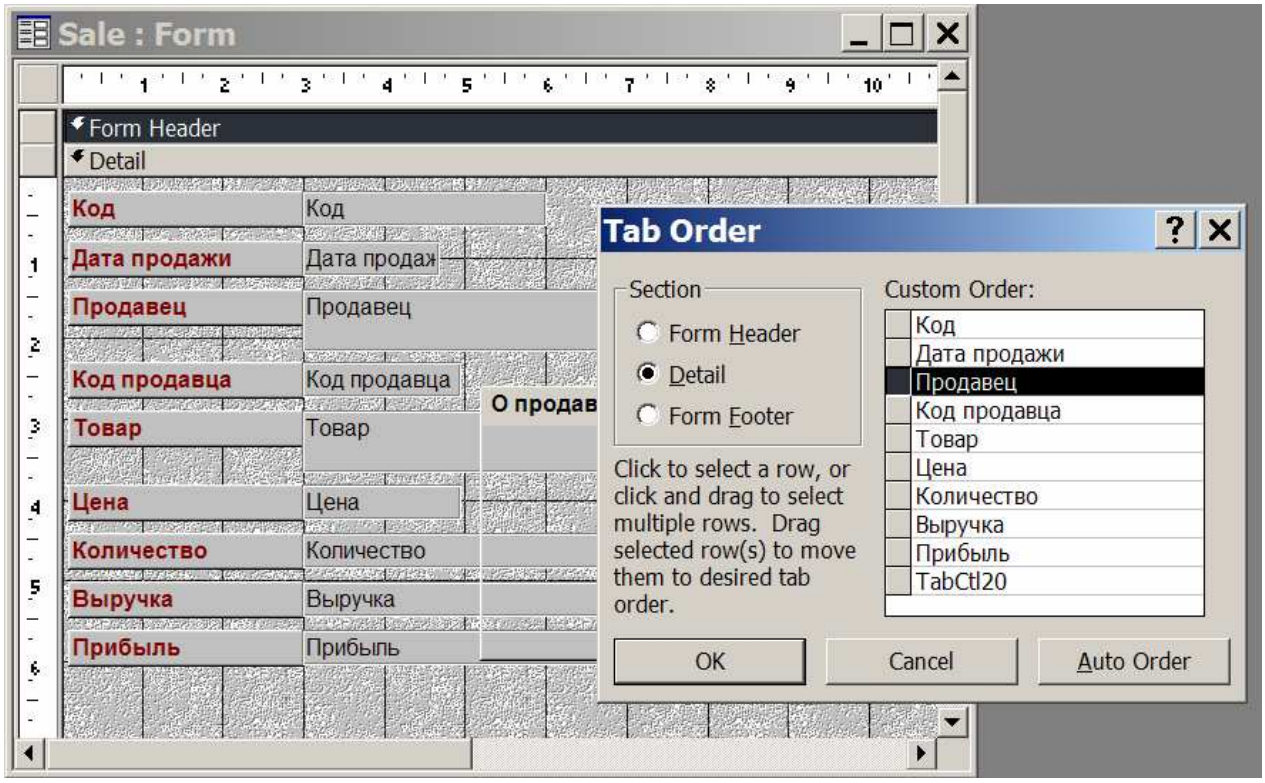


Рисунок 8.11 – Вибір послідовності обходу полів

Все вищесказане говорить про те, що бажано при первинному проектуванні й створенні таблиць максимально продумувати проектні вирішення та з обережністю підходити до подальшому коректуванню структури таблиць бази даних.

8.4. Створення звіту за допомогою Майстра

Для створення нового **звіту**, у вікні бази даних варто перейти на закладку «Reports», натиснути на кнопку «New» і у верхній частині вікна, що з'явилося (рис. 8.13) вибрати спосіб створення звіту, а в нижній – вказати таблицю або запит, дані з якого будуть виводитися у звіті. Після чого слід натис-

нути на кнопку «ОК».

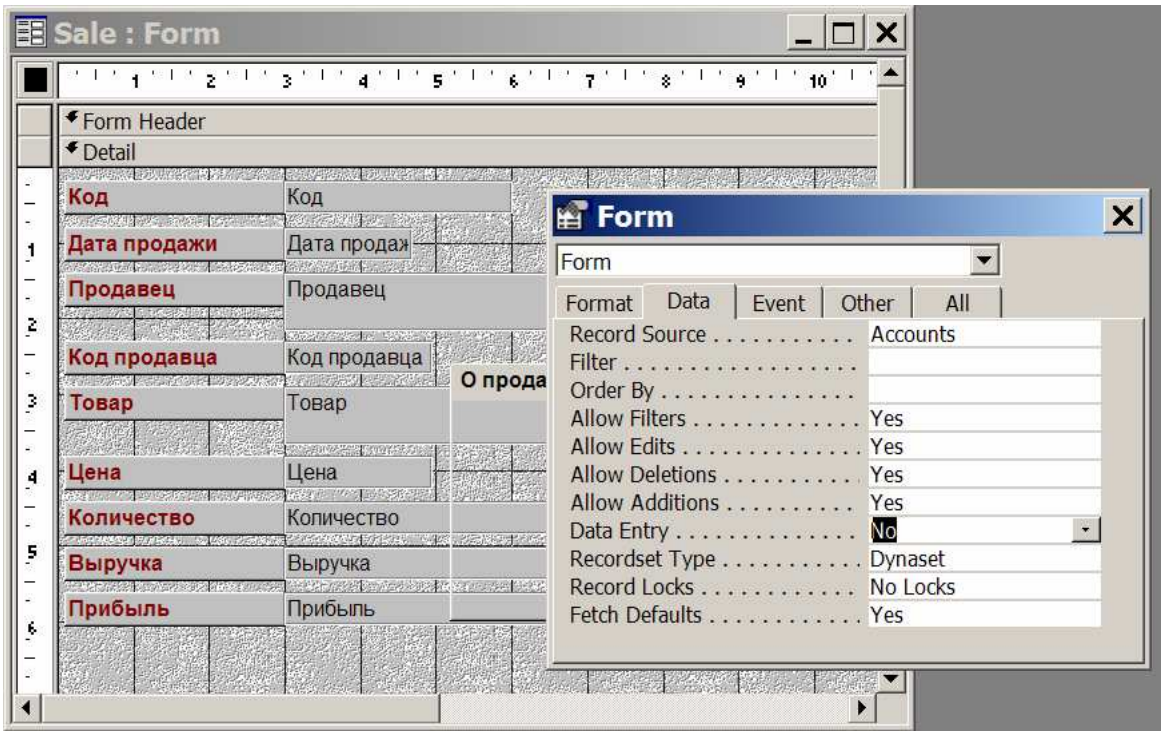


Рисунок 8.12 – Властивості форми

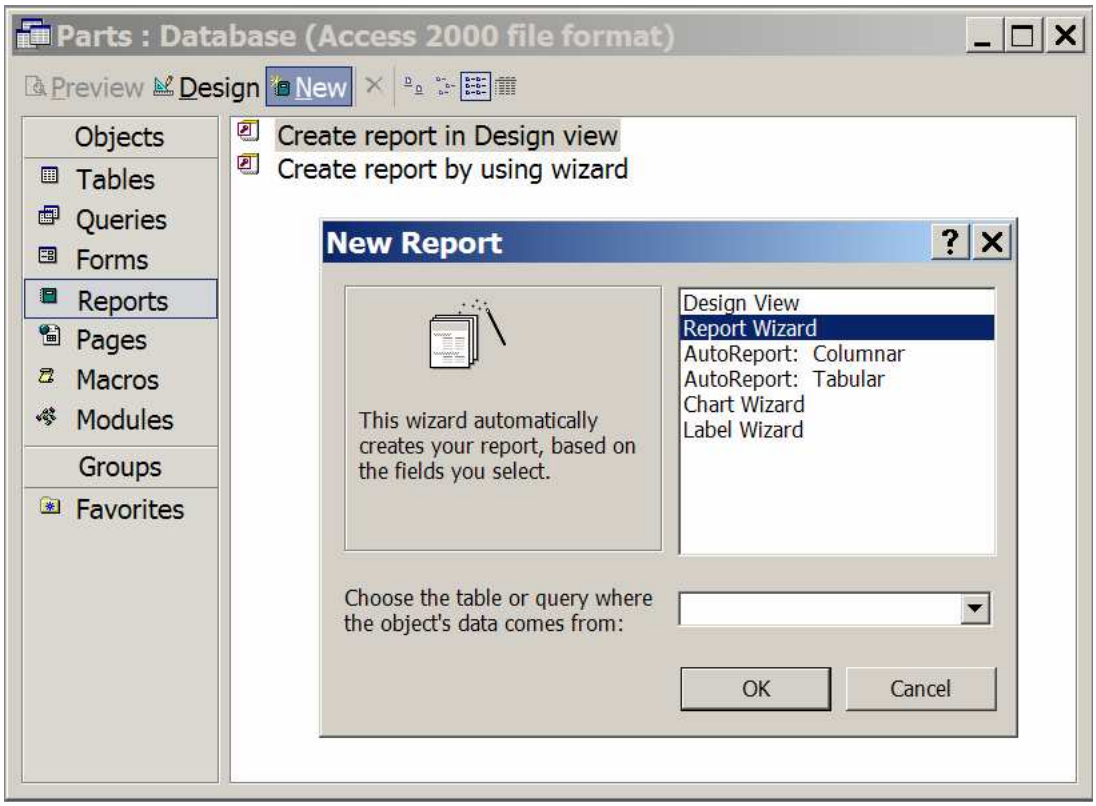


Рисунок 8.13 – Початкове вікно для створення звіту

Так само, як і при створенні форм, звіти краще створювати, користуючись одним з Майстрів, а потім, якщо буде потреба, зробити бажане коректування форми звіту. Майстрами в цьому випадку можна вважати всі можливості, перераховані в спадаючому меню, зображеному на рис. 8.13, крім позиції «Конструктор».

Найпростішим способом створення звітів є використання такзваних автозвітів («AutoReport»). При цьому у звіт виводяться всі поля обраного джерела даних (таблиці або запиту). Назвою звіту і його заголовком стає назва джерела даних. Ніяке втручання користувача в процес створення документа не передбачається. Очевидно, така можливість прийнятна тільки для найпростіших, епізодичних звітів.

Більш гнучкою можливістю є використання **Майстра звітів** («Report Wizard»). Саме цей інструмент найчастіше використовується при первісному створенні документа. При будь-якому способі створення звіту необхідно вибрати джерело даних, на основі якого він буде формуватися. Ним може бути одна або кілька таблиць або запит (рис. 8.14).

Якщо вибрати Майстер звітів, далі система запропонує визначити поля (рис. 8.14), які будуть входити у звіт. При виборі цих полів можна використовувати або кнопку із двома стрілками (у випадку, якщо у звіт будуть входити всі або більшість полів), або переносити поля по одному, використовуючи кнопку з однією стрілкою. При цьому варто звернути увагу на те, що поля слід переносити не в тому порядку, у якому вони розташовуються в структурі вихідної таблиці, а в тому, в якому вони будуть використовуватися у звіті. І хоча потім, у режимі Конструктора, можна міняти положення тих або інших елементів у звіті, краще продумувати ці питання відразу.

Припустимо, ми хочемо на основі таблиці «Accounts» одержати список товарів, проданих кожним з продавців, і принесений ними виторг. Інформацію згрупувати по продавцях. Для кожного продавця записи впорядкувати за алфавітом по полю «Товар», і далі – по датах продажу.

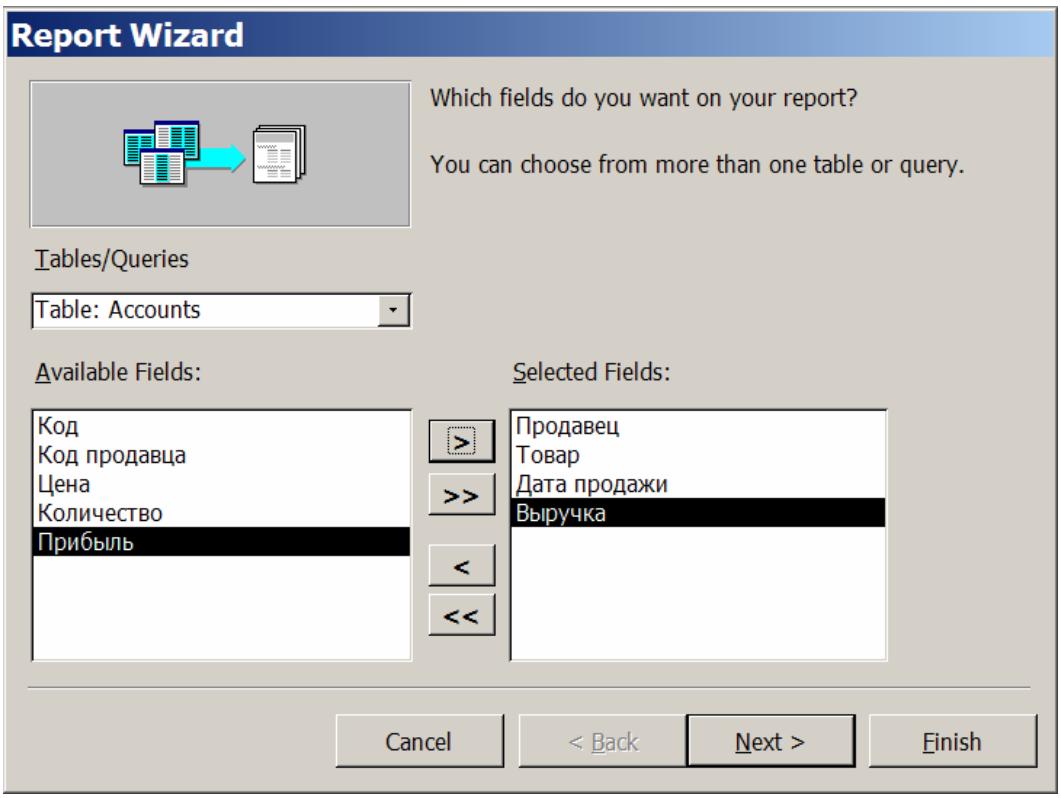


Рисунок 8.14 – Вибір полів, що будуть включені у звіт

Після визначення полів, що включають у звіт, система пропонує вибрати рівні групування (рис. 8.15). У нашому прикладі нас влаштовує запропоноване поле групування, і нових рівнів групування вводити не треба. У загальному випадку, для виділення рівнів групування треба позиціюватися на відповідне поле, що буде полем, по якому виконується групування, і натиснути на кнопку зі стрілкою. Якщо передбачається кілька рівнів групування, то поля повинні вибиратися в порядку старшинства групування.

Наступний екран (рис. 8.16) дозволяє задати порядок сортування та обчислення підсумків (рис. 8.17). Ми вибрали сортування за полями «Товар» та «Дата продажу» і одержання сумарних підсумків по полю «Виторг». Для того, щоб по кожному продавцеві обчислювалася сума виторгу, необхідно натиснути кнопку «Summary Options» і вибрати відповідну функцію «Sum».

Далі можна вибрати бажаний дизайн макета звіту, його стиль, а також задати ім'я звіту (рис.8.18 – 8.20) і натиснути на кнопку «Finish». Звіт виведеться на екран у режимі попереднього перегляду (рис. 8.21).

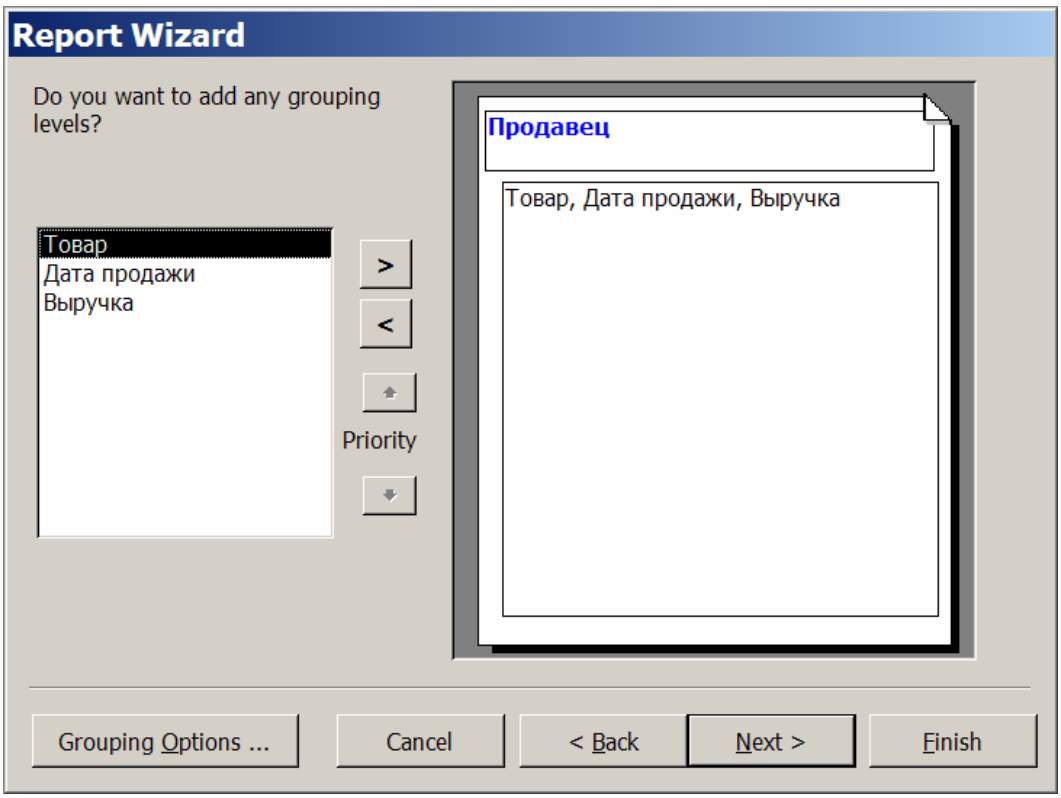


Рисунок 8.15 – Визначення рівнів групування

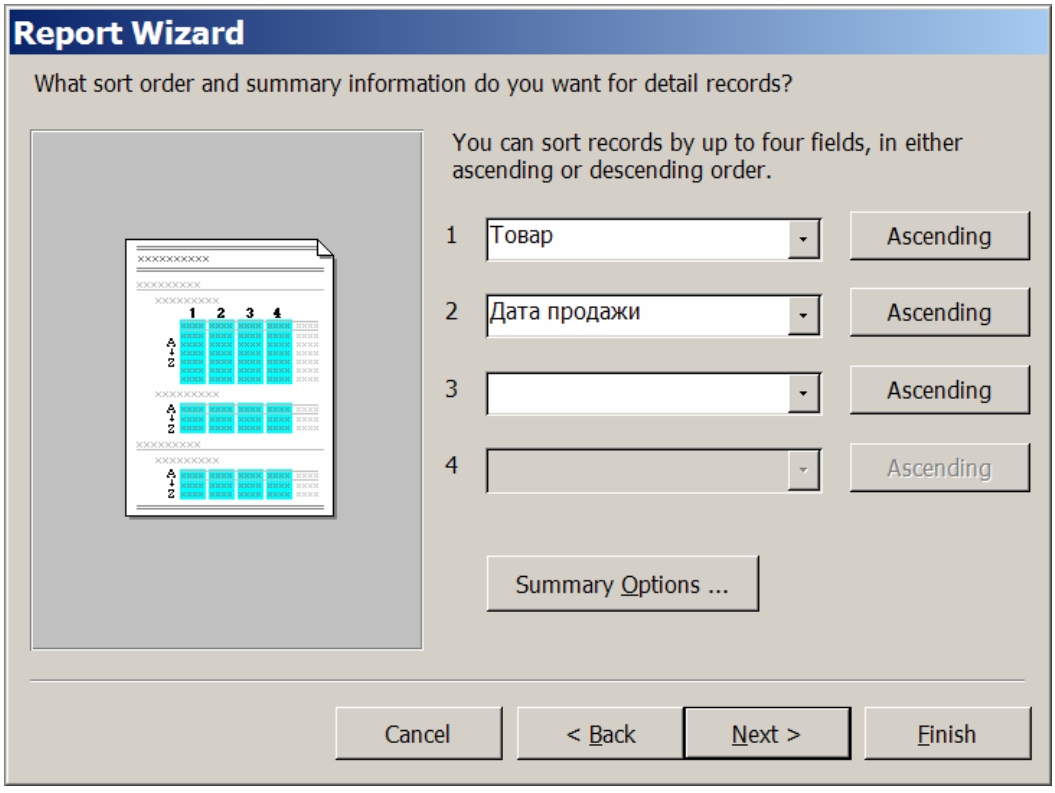


Рисунок 8.16 – Визначення порядку сортування

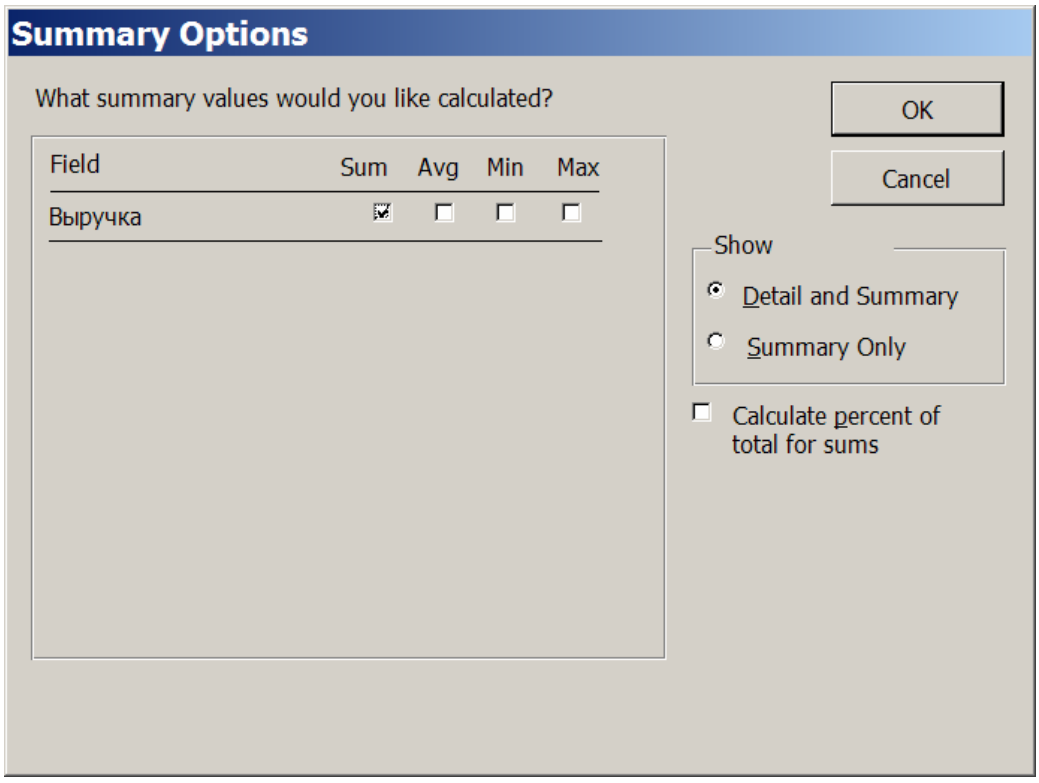


Рисунок 8.17 – Задавання обчислення підсумків

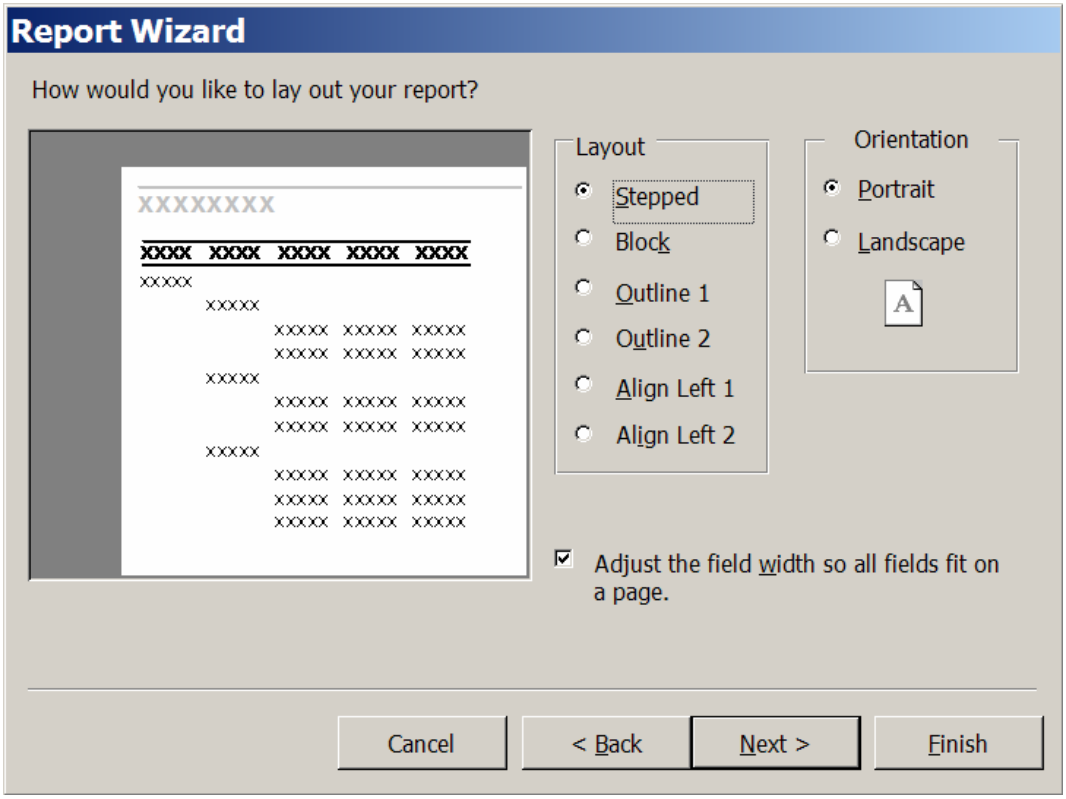


Рисунок 8.18 – Вибір дизайну звіту

Рисунок 8.19 – Вибір стилю звіту

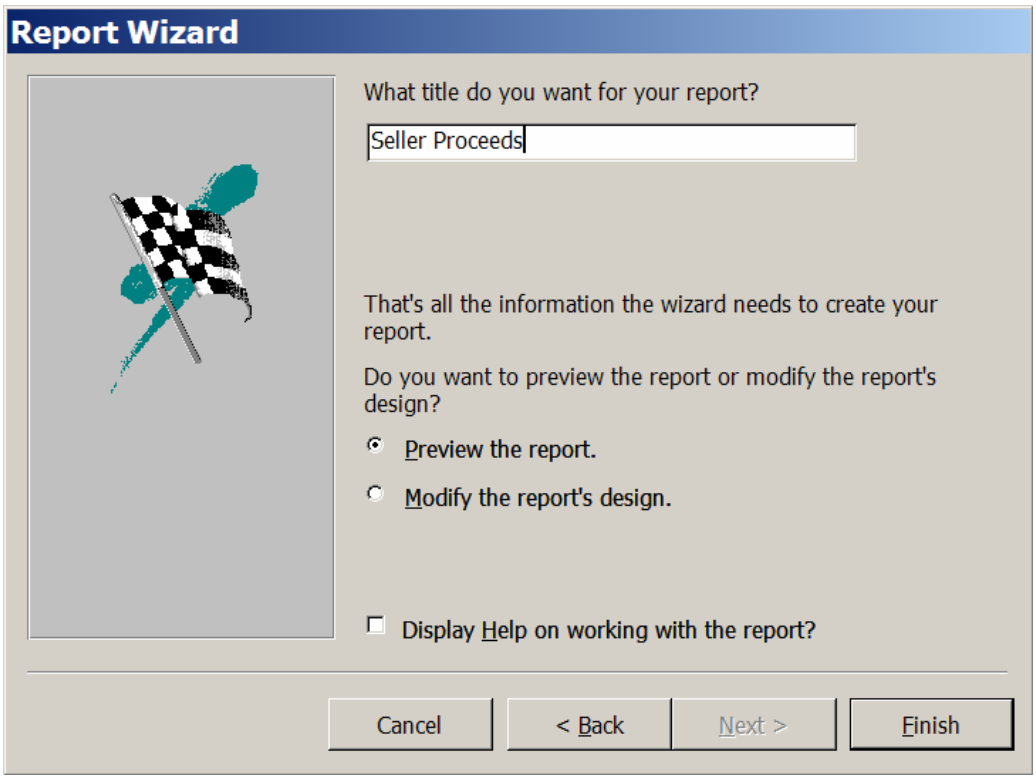
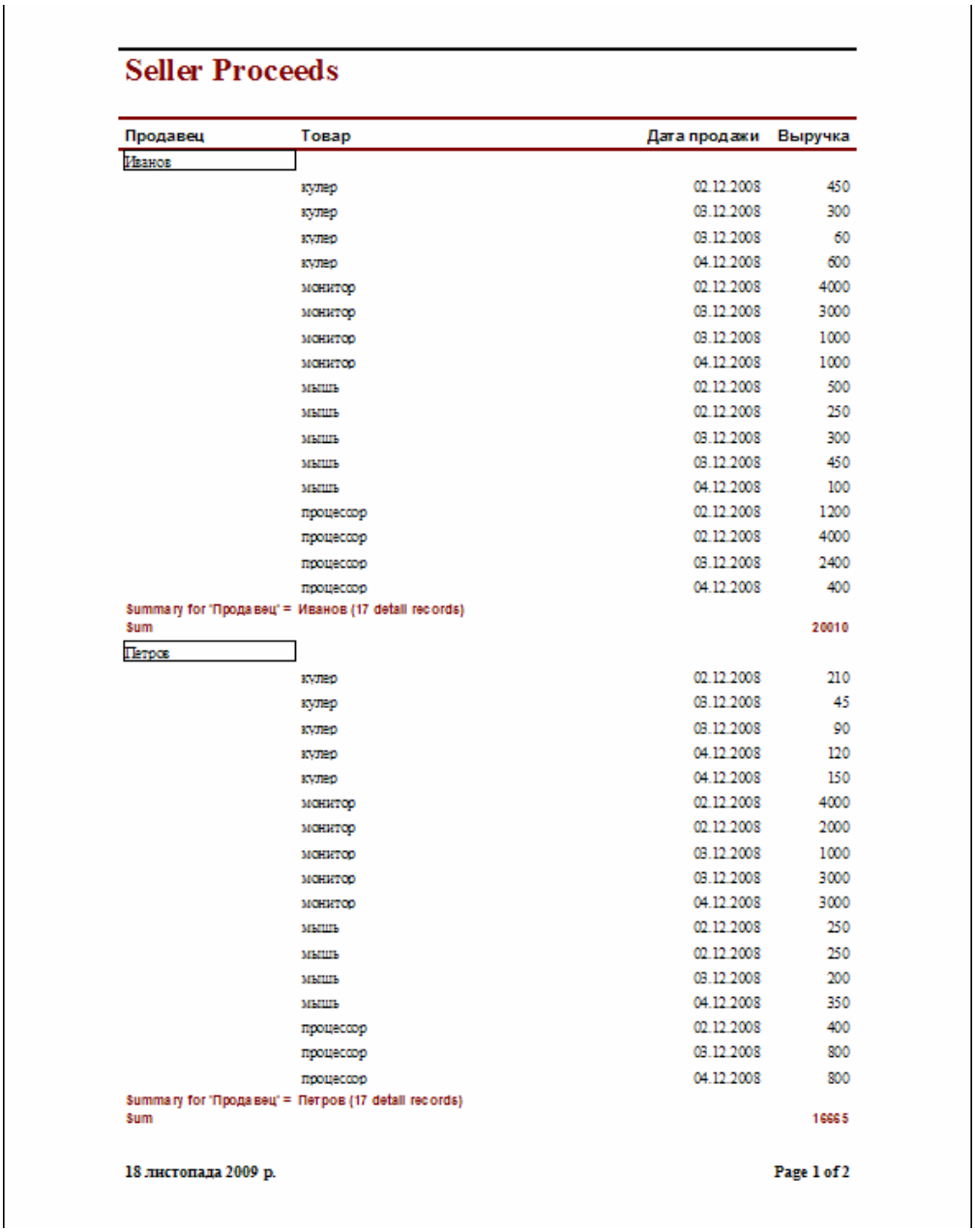


Рисунок 8.20 – Задавання імені звіту



Seller Proceeds

Продавец	Товар	Дата продажи	Выручка
Иванов			
	кулер	02.12.2008	450
	кулер	03.12.2008	300
	кулер	03.12.2008	60
	кулер	04.12.2008	600
	монитор	02.12.2008	4000
	монитор	03.12.2008	3000
	монитор	03.12.2008	1000
	монитор	04.12.2008	1000
	мышь	02.12.2008	500
	мышь	02.12.2008	250
	мышь	03.12.2008	300
	мышь	03.12.2008	450
	мышь	04.12.2008	100
	процессор	02.12.2008	1200
	процессор	02.12.2008	4000
	процессор	03.12.2008	2400
	процессор	04.12.2008	400
Summary for 'Продавец' = Иванов (17 detail records)			
Sum			20010
Петров			
	кулер	02.12.2008	210
	кулер	03.12.2008	45
	кулер	03.12.2008	90
	кулер	04.12.2008	120
	кулер	04.12.2008	150
	монитор	02.12.2008	4000
	монитор	02.12.2008	2000
	монитор	03.12.2008	1000
	монитор	03.12.2008	3000
	монитор	04.12.2008	3000
	мышь	02.12.2008	250
	мышь	02.12.2008	250
	мышь	03.12.2008	200
	мышь	04.12.2008	350
	процессор	02.12.2008	400
	процессор	03.12.2008	800
	процессор	04.12.2008	800
Summary for 'Продавец' = Петров (17 detail records)			
Sum			16665

18 листопада 2009 р. Page 1 of 2

Рисунок 8.21 – Перша сторінка одержаного звіту

При створенні звіту з використанням Майстра, на одному з останніх кроків система просить задати ім'я звіту. За замовчуванням звіту привласнюється ім'я, що збігається з ім'ям таблиці, на основі якої формується звіт (у нашому випадку це була таблиця «Accounts»). Однак краще відразу задати необхідне ім'я звіту. Варто звернути увагу, що задане ім'я виступає у двох ролях: як ім'я об'єкта-звіту, і як назва документа, виведена в заголовку звіту. У нашому прикладі документ буде називатися «Seller Proceeds». Такі довгі імена об'єктам звичайно не привласнюються.

Якщо перейменувати звіт після його створення, то його заголовок не зміниться. Це можна зробити, наприклад, виділивши назву звіту у вікні бази даних, після чого натиснути праву клавішу миші. У меню, що з'явилося, вибрати позицію «Rename» і набрати нове ім'я.

Необхідно звернути увагу на те, що в даному навчальному посібнику ми розглядаємо лише деякі можливості генератора звітів як інструментального засобу, але не розглядаємо питання проектування БД. Щоб приклади були доступними для огляду, реальні ситуації надзвичайно спрощені. Так, щоб реально визначити виторг, навіть якщо в цей період не проводилося ніяких акцій, треба перевіряти, чи не проданий даний товар з уцінкою, чи не мав покупець права на знижку, чи не проданий товар у кредит і т.п. Природно, що й самі таблиці повинні бути спроектовані трохи по іншому, ніж ті, що розглядалися в прикладах.

Джерелами для звітів можуть слугувати кілька таблиць. При визначенні полів, що входять у звіт, можна послідовно вибирати різні таблиці та запити й відбирати потрібні поля з них. Якщо звіт створюється без використання Майстра, то при необхідності застосувати кілька джерел варто спочатку створити відповідний запит, а потім вже на його основі формувати звіт.

8.5. Коректування та збереження звіту

Отриманий звіт може в чомусь не відповідати Вашим потребам. Наприклад, не влаштовують стандартні повідомлення, видавані за замовчуванням, не потрібно виводити дату створення звіту, або, навпаки, треба ввести якісь додаткові елементи і т.п. Щоб скорегувати форму звіту, треба перейти в режим **Конструктора звітів**. Для цього можна в пункті меню «View» вибрати команду «Design View», або вибрати відповідну кнопку на панелі інструментів. Можливості Конструктора звітів досить різноманітні й не розглядаються в даному посібнику. При необхідності читач може звернутися до більш докладної літератури [1].

Готовий звіт може бути роздрукований, або збережений у файлі. Щоб запам'ятати вміст звіту в текстовому файлі, можна використати позиції меню Tools – Office Links – Publish It with Microsoft Word, або кнопку «OfficeLinks», призначену для експорту даних в MS Word або MS Excel. Можна застосувати й інший шлях для збереження вмісту звіту, а саме використати позиції меню File – Save As.

Збереження вмісту звіту у файлі варто проводити не тільки тоді, коли друк звіту і його формування за якимись причинами рознесено в часі, або звіт повинен бути переданий в електронному виді користувачеві, що не має доступу до Вашого комп'ютера, але й для того, щоб зберегти звіт у тому вигляді, що відповідав вмісту бази даних на момент одержання даного звіту. В іншому випадку, при кожному відкритті звіту системою MS Access, його вміст буде відповідати стану БД на момент відкриття.

Питання й тести для самоконтролю

1. Що таке екранна форма?
2. Чи можна обійтися без використання екранних форм і чому?
3. Як створити нову форму за допомогою Майстра в СУБД MS Access?
4. Як доробити існуючу форму за допомогою Конструктора?
5. Розкажіть, що Ви знаєте про багатосторінкові форми.
6. Які Ви пам’ятаєте властивості екранних форм?
7. Що таке звіти й навіщо вони створюються?
8. Опишіть процедуру створення звіту за допомогою Майстра MS Access.
9. У чому подібність і розходження звітів MS Access і таблиць MS Excel?
10. Що можна зробити з готовим звітом?

Розділ 9. ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Матеріал даного розділу винесений у кінець навчального посібника не випадково. Справа в тому, що тільки після всебічного ознайомлення в попередніх розділах із сутністю інформаційних систем, з'являється можливість на досить високому теоретичному рівні розглянути загальнонаукові й методологічні поняття інформаційних технологій.

9.1. Поняття інформаційної технології

Як ми вже відзначали в розділі 1, слово «технологія» у буквальному перекладі із грецького означає «мистецтво», «майстерність», «уміння», а ці терміни описують не що інше, як стан процесу праці (роботу майстра навіть непосвячений легко зможе відрізнити від «праці» халтурника). Під процесом же варто розуміти певну сукупність дій, спрямованих на досягнення поставленої мети. Процес повинен визначатися обраною людиною стратегією й реалізуватися за допомогою сукупності різних засобів і методів.

9.1.1. Визначення інформаційної технології

Під технологією матеріального виробництва розуміють процес, обумовлений сукупністю засобів і методів обробки, виготовлення, зміни стану, властивостей, форми сировини або матеріалу. Технологія змінює якість або первісний стан матерії з метою одержання матеріального продукту.

Інформація також є одним з найцінніших ресурсів суспільства поряд з такими традиційними матеріальними видами ресурсів, як нафта, газ, корисні копалини та ін., а значить, процес її переробки за аналогією із процесами переробки матеріальних ресурсів можна сприймати як технологію. Тоді справедливе таке визначення.

Інформаційна технологія, процес, що використовує сукупність засобів і методів збору, обробки та передачі даних (первинної інформації) для одержання інформації нової якості про стан об'єкта, процесу або явища (інформаційного продукту).

Мета технології матеріального виробництва – випуск продукції, що задовольняє потреби людини або системи.

Мета інформаційної технології – виробництво інформації для її аналізу людиною й прийняття на його основі рішення з виконання якої-небудь дії.

Відомо, що, застосовуючи різні технології до того самого матеріального ресурсу, можна одержати різні вироби, продукти. Те ж саме буде справедливим і для технології переробки інформації.

Для наочності приведемо порівняльну таблицю основних компонентів обох видів технологій (табл. 9.1).

Таблиця 9.1. – Співставлення основних компонентів технологій

КОМПОНЕНТИ ТЕХНОЛОГІЙ ДЛЯ ВИРОБНИЦТВА ПРОДУКЦІЇ	
Матеріальної	Інформаційної
Підготовка сировини й матеріалів	Збір даних або первинної інформації
Виробництво матеріального продукту	Обробка даних і одержання результуючої інформації
Збут вироблених продуктів споживачам	Передача результуючої інформації користувачеві для прийняття на її основі рішень

9.1.2. Нові інформаційні технології

Інформаційна технологія є найбільш важливою складовою процесу використання інформаційних ресурсів суспільства. До теперішнього часу вона пройшла кілька еволюційних етапів, зміна яких визначалася, головним чином, розвитком науково-технічного прогресу, появою нових технічних засобів переробки інформації. У сучасному суспільстві основним технічним за-

собом технології переробки інформації є персональний комп'ютер, що істотно вплинув як на концепцію побудови й використання технологічних процесів, так і на якість результуючої інформації. Впровадження персонального комп'ютера в інформаційну сферу і застосування телекомунікаційних засобів зв'язку визначили новий етап розвитку інформаційних технологій та, як наслідок, зміну їхньої назви шляхом приєднання одного із синонімів: «нові», «комп'ютерні» або «сучасні».

Прикметник «нові» підкреслює новаторський, а не еволюційний характер цих технологій. Впровадження нових інформаційних технологій є новаторським актом у тому розумінні, що вони істотно змінюють зміст різних видів діяльності в організаціях. У поняття нової інформаційної технології включено також комунікаційні технології, які забезпечують передачу інформації різними засобами, а саме – телефоном, телеграфом, телебаченням, факсом та ін. Нижче наведено основні характерні риси **нових інформаційних технологій**:

– принципово нові засоби обробки інформації;

– вбудовування в технологію керування;

– нові технології комунікацій;

– цілісні технологічні системи;

– інтеграція функцій фахівців і менеджерів;

– нові технології обробки інформації;

– цілеспрямоване створення, передача, зберігання й відображення інформації;

– облік закономірностей соціального середовища;

– нові технології прийняття управлінських рішень.

Таким чином, нові інформаційні технології – це інформаційні технології з «дружнім» інтерфейсом роботи користувача, що використовують персональні комп'ютери та телекомунікаційні засоби.

Прикметник «комп'ютерні» підкреслює, що основним технічним засобом реалізації цих технологій є комп'ютер.

Відзначимо три основних принципи нових (комп'ютерних) інформаційних технологій:

– інтерактивний (діалоговий) режим роботи з комп'ютером;

– інтегрованість (стикування, взаємозв'язок) з іншими програмними продуктами;

– гнучкість процесу зміни як даних, так і постановок завдань.

Очевидно, більш точним варто вважати все-таки термін «нові», а не «комп'ютерні» інформаційні технології, оскільки він відбиває в їхній структурі не тільки технології, засновані на використанні комп'ютерів, але й технології, засновані на інших технічних засобах, особливо на засобах, що забезпечують телекомунікацію.

9.1.3. Інструментарій інформаційних технологій

Реалізація технологічного процесу матеріального виробництва здійснюється за допомогою різних технічних засобів, до яких відносяться: устаткування, верстати, інструменти, конвеєрні лінії і т.п.

За аналогією і для інформаційної технології повинно бути щось подібне. Такими технічними засобами виробництва інформації є апаратне, програмне й математичне забезпечення технологічного процесу. З їхньою допомогою виконується переробка первинної інформації в інформацію нової якості. Виділимо окремо з цих засобів програмні продукти й назвемо їх інструментарієм (toolbox). Для більшої чіткості можна його конкретизувати, назвавши програмним інструментарієм інформаційної технології. Визначимо це поняття.

Інструментарій інформаційної технології – один або кілька взаємозалежних програмних продуктів для певного типу комп'ютера, технологія роботи в якому дозволяє досягти поставленої користувачем мети.

Як інструментарій можна використати такі розповсюджені види програмних продуктів для персонального комп'ютера: текстовий процесор (ре-

дактор), настільні видавничі системи, електронні таблиці, системи керування базами даних, електронні записні книжки, електронні календарі, інформаційні системи функціонального призначення (фінансові, бухгалтерські, для маркетингу та ін.), експертні системи і т.д.

9.1.4. Співвідношення інформаційних систем та інформаційних технологій

Інформаційні технології тісно пов'язані з інформаційними системами, які є для них основним середовищем. На перший погляд може здатися, що введені в посібнику визначення інформаційної технології та системи дуже схожі між собою. Однак це не так.

Інформаційна технологія є процесом, що складається із чітко регламентованих правил виконання дій різного ступеня складності над даними, що зберігаються в комп'ютерах. Основна мета інформаційної технології – одержати необхідну для користувача інформацію в результаті цілеспрямованих дій з переробки первинної інформації.

Інформаційна система – це середовище, елементами якого є комп'ютери, комп'ютерні мережі, програмні продукти, бази даних, люди, різного роду технічні й програмні засоби зв'язку і т.д. Основна мета інформаційної системи – організація зберігання й передачі інформації. Інформаційна система є змішаною, людино-комп'ютерною системою обробки інформації.

Реалізація функцій інформаційної системи неможлива без знання орієнтованої на неї інформаційної технології. Інформаційна технологія може існувати й поза сферою інформаційної системи.

Таким чином, інформаційна технологія є більш ємним поняттям, що відбиває сучасні уявлення про процеси перетворення інформації в інформаційному суспільстві. У вмілому поєднанні двох інформаційних технологій – управлінської та комп'ютерної – полягає головна умова успішної роботи інформаційної системи.

Узагальнюючи все вищесказане, можна запропонувати трохи більш вузькі, ніж введені раніше, визначення інформаційної системи та технології, реалізованих засобами комп'ютерної техніки.

Інформаційна технологія – сукупність чітко визначених цілеспрямованих дій персоналу з переробки інформації на комп'ютері.

Інформаційна система – людино-комп'ютерна система для підтримки прийняття рішень і виробництва інформаційних продуктів, що використовує комп'ютерну інформаційну технологію.

9.1.5. Складові інформаційної технології

Такі технологічні поняття, використовувані у виробничій сфері, як норма, норматив, технологічний процес, технологічний етап і т.п., можуть застосовуватися і в інформаційній технології. Перш ніж розробляти ці поняття в будь-якій технології, у тому числі в інформаційній, завжди необхідно починати з визначення мети. Потім варто спробувати провести структурування всіх передбачуваних дій, що приводять до наміченої цілі, і вибрати необхідний програмний інструментарій.

Технологічний процес переробки інформації може бути визначений у вигляді ієрархічної структури по рівнях:

– перший рівень – етапи, де реалізуються порівняно тривалі технологічні процеси, що складаються з дій наступних рівнів;

– другий рівень – дії, у результаті виконання яких буде створено конкретний об'єкт в обраному на першому рівні програмному середовищі;

– третій рівень – сукупність стандартних для кожного програмного середовища прийомів роботи, що приводять до виконання поставленої у відповідній дії мети.

Необхідно розуміти, що освоєння інформаційної технології та подальше її використання повинні звестися до того, що спочатку варто добре опанувати набір елементарних прийомів, число яких обмежене. Із цього обме-

женого числа елементарних прийомів у різних комбінаціях складається дія, а з дій, також у різних комбінаціях, складаються етапи, які визначають той або інший технологічний процес. Сукупність технологічних процесів утворить власне інформаційну технологію.

9.2. Етапи розвитку інформаційних технологій

Існує кілька точок зору на розвиток інформаційних технологій з використанням комп'ютерів, які визначаються різними ознаками розподілу.

Загальним для всіх викладених нижче підходів є те, що з появою персонального комп'ютера почався новий етап розвитку інформаційної технології. Основною метою стає задоволення персональних інформаційних потреб людини як для професійної сфери діяльності, так і для побутової.

9.2.1. Етапи розвитку задач та процесів обробки інформації

Етап 1 (1960...70-і роки) – обробка даних в обчислювальних центрах у режимі колективного користування. Основним напрямком розвитку інформаційної технології була автоматизація рутинних дій людини.

Етап 2 (з 1980-х років) – обробка даних у розподілених обчислювальних системах. Створення інформаційних технологій, спрямованих на вирішення стратегічних завдань, у тому числі з використанням основ штучного інтелекту.

9.2.2. Етапи розв'язання проблем на шляху інформатизації суспільства

Етап 1 (до кінця 1960-х років) характеризується проблемою обробки більших обсягів даних в умовах обмежених можливостей апаратних засобів.

Етап 2 (до кінця 1970-х років) пов'язується з поширенням ЕОМ серії IBM/360, Проблема цього етапу – відставання програмного забезпечення від

рівня розвитку апаратних засобів.

Етап 3 (з початку 1980-х років) – комп'ютер стає інструментом непрофесійного користувача, а інформаційні системи – засобом підтримки прийняття його рішень. Проблеми – максимальне задоволення потреб користувача і створення відповідного інтерфейсу роботи в комп'ютерному середовищі.

Етап 4 (з початку 1990-х років) – створення сучасної технології міжорганізаційних зв'язків та інформаційних систем. Проблеми цього етапу досить численні. Найбільш істотними з них є:

– вироблення угод і встановлення стандартів, протоколів для комп'ютерного зв'язку;

– організація доступу до стратегічної інформації;

– організація захисту й безпеки інформації.

9.2.3. Етапи розвитку переваг, принесених комп'ютерними технологіями

Етап 1 (з початку 1960-х років) характеризується досить ефективною обробкою інформації при виконанні рутинних дій з орієнтацією на централізоване колективне використання ресурсів обчислювальних центрів. Основним критерієм оцінки ефективності створюваних інформаційних систем була різниця між витраченими на розробку та зекономленими в результаті впровадження засобами. Основною проблемою на цьому етапі була психологічна – погана взаємодія користувачів, для яких створювалися інформаційні системи, і розробників через розходження їхніх поглядів і розуміння розв'язуваних проблем. Як наслідок цієї проблеми, створювалися системи, які користувачі погано сприймали та, незважаючи на їх досить великі можливості, не використовували повною мірою.

Етап 2 (з середини 1970-х років) пов'язаний з появою персональних комп'ютерів. Змінився підхід до створення інформаційних систем – орієнтація зміщується в бік індивідуального користувача для підтримки прийнятих

ним рішень. Користувач зацікавлений у проведеній розробці, налагоджується контакт із розробником, виникає взаєморозуміння обох груп фахівців. На цьому етапі використовується як централізована обробка даних, характерна для першого етапу, так і децентралізована, заснована на вирішенні локальних завдань і роботі з локальними базами даних на робочому місці користувача.

Етап 3 (з початку 1990-х років) пов'язаний з поняттям аналізу стратегічних переваг у бізнесі та заснований на досягненнях телекомунікаційної технології розподіленої обробки інформації. Інформаційні системи мають своєю метою не просте збільшення ефективності обробки даних і допомогу керівникові. Відповідні інформаційні технології повинні допомогти організації вистояти в конкурентній боротьбі й одержати перевагу.

9.2.4. Етапи вдосконалення інструментарію інформаційної технології

Етап 1 (до другої половини XIX століття) – «ручна» інформаційна технологія, інструментарій якої становили: перо, чорнильниця, книга. Комунікації здійснювалися шляхом доставки через пошту листів, пакетів, депеш. Основна мета технології – відображення інформації в необхідній формі.

Етап 2 (з кінця XIX століття) – «механічна» технологія, інструментарій якого становили: друкарська машинка, телефон, диктофон, оснащена більш досконалими засобами доставки пошта. Основна мета технології – відображення інформації в потрібній формі зручнішими засобами.

Етап 3 (1940...60-і роки) – «електрична» технологія, інструментарій якої становили: великі ЕОМ і відповідне програмне забезпечення, електричні друкарські машинки, ксерокси, портативні диктофони. Змінюється мета технології. Акцент в інформаційній технології починає переміщуватися з форми відображення інформації на формування її змісту.

Етап 4 (з початку 1970-х років) – «електронна» технологія, основним інструментарієм якої стають великі ЕОМ і створювані на їхній базі автоматизовані системи керування й інформаційно-пошукові системи, оснащені ши-

роким спектром базових і спеціалізованих програмних комплексів. Центр ваги технології ще більше зміщається на формування змістовної сторони інформації для управлінського середовища різних сфер громадського життя, особливо на організацію аналітичної роботи. Безліч об'єктивних і суб'єктивних факторів не дозволили вирішити поставлені перед новою концепцією інформаційної технології завдання. Однак було надбано досвід формування змістовної сторони управлінської інформації та підготовлено професійну, психологічну й соціальну база для переходу на новий етап розвитку технології.

Етап 5 (з середини 1980-х років) – «комп'ютерна» або «нова» технологія, основним інструментарієм якої є персональний комп'ютер з широким спектром стандартних програмних продуктів різного призначення. На цьому етапі відбувається процес персоналізації автоматизованих систем управління, що проявляється в створенні систем підтримки прийняття рішень певними фахівцями. Подібні системи мають вбудовані елементи аналізу й інтелекту для різних рівнів керування, реалізуються на персональному комп'ютері й використовують телекомунікації. У зв'язку з переходом на мікропроцесорну базу, істотним змінам піддаються й технічні засоби побутового, культурного й іншого призначень. Починають широко використатися в різних областях глобальні й локальні комп'ютерні мережі.

9.3. Види інформаційних технологій

9.3.1. Інформаційна технологія обробки даних

Інформаційна технологія **обробки даних** призначена для вирішення добре структурованих завдань, для яких є необхідні вхідні дані й відомі алгоритми та інші стандартні процедури їхньої обробки. Ця технологія застосовується на рівні поточної (виконавської) діяльності персоналу невисокої кваліфікації з метою автоматизації деяких рутинних, постійно повторюваних алгоритмів управлінської праці. Тому впровадження інформаційних технологій

і систем на цьому рівні істотно підвищує продуктивність праці персоналу, звільняє його від рутинних дій, можливо, навіть іноді призводить до необхідності скорочення чисельності працівників.

На рівні поточної діяльності вирішуються такі завдання:

– обробка даних про дії, виконані установою;

– створення періодичних контрольних звітів про стан справ в організації;

– одержання відповідей на всілякі поточні запити й оформлення їх у вигляді паперових документів або звітів.

Приклад контрольного звіту: щоденний звіт про надходження й видачі наявних коштів банком, сформований з метою контролю балансу коштів.

Приклад запиту: запит до бази даних з кадрів, що дозволить одержати дані про вимоги, висунуті до кандидатів на зайняття певної посади.

Існує кілька особливостей, пов'язаних з обробкою даних, що відрізняють дану технологію від всіх інших:

– виконання необхідних установі завдань з обробки даних. Кожній організації запропоновано законом мати й зберігати дані про свою діяльність, які можна використати як засіб забезпечення й підтримки контролю в установі. Тому в будь-якій фірмі обов'язково повинна бути інформаційна система обробки даних і розроблено відповідну інформаційну технологію;

– вирішення тільки добре структурованих завдань, для яких можна розробити алгоритм;

– виконання стандартних процедур обробки. Існуючі стандарти визначають типові процедури обробки даних і пропонують їхнє дотримання організаціями всіх видів;

– виконання основного обсягу робіт в автоматичному режимі з мінімальною участю людини;

– використання деталізованих даних. Записи про діяльність фірми мають детальний (докладний) характер, що допускає проведення ревізій. У

процесі ревізії діяльність установи перевіряється хронологічно від початку періоду до його кінця й від кінця до початку;

– акцент на хронологію подій;

– вимога мінімальної допомоги в вирішенні проблем з боку фахівців інших рівнів.

Розглянемо основні компоненти інформаційної технології обробки даних. До таких компонентів відносяться:

– збір даних. У міру того, як фірма робить продукцію або послуги, кожна її дія супроводжується відповідними записами даних. Звичайно дії організації, що зачіпають зовнішнє оточення, виділяються окремо;

– обробка даних. Для створення з даних, що надходять, інформації, що відбиває діяльність фірми, використовуються типові алгоритми, на зразок класифікації або групування. Первинні дані звичайно мають вигляд кодів, що складаються з одного або декількох символів. Ці коди, що виражають певні ознаки об'єктів, використовуються для ідентифікації й групування записів;

– зберігання даних. Багато даних на рівні поточної діяльності необхідно зберігати для наступного використання або тут же, або на іншому рівні. Для їхнього зберігання створюються бази даних;

– створення звітів (документів). В інформаційній технології обробки даних необхідно створювати документи для керівників та виконавців установи, а також для зовнішніх партнерів. При цьому документи створюються як у зв'язку з проведеною фірмою діяльністю, так і періодично, наприкінці кожного місяця, кварталу або року.

9.3.2. Інформаційна технологія керування

Метою інформаційної технології керування є задоволення інформаційних потреб усіх без винятку співробітників установи, що мають справу із прийняттям рішень. Як наслідок, вона може бути корисна на будь-якому рівні керування.

Ця технологія орієнтована на роботу в середовищі інформаційної системи керування й використовується при меншій структурованості розв'язуваних завдань, якщо їх порівнювати із завданнями, розв'язуваними за допомогою інформаційної технології обробки даних.

Інформаційні системи керування ідеально підходять для задоволення подібних інформаційних потреб працівників різних функціональних підсистем (підрозділів) або рівнів керування установою. Інформація, що поставляється ними, містить відомості про минуле, сучасне та ймовірне майбутнє організації. Ця інформація має вигляд регулярних або спеціальних управлінських звітів.

Для прийняття рішень на рівні управлінського контролю, інформація повинна бути відображена в агрегованому вигляді так, щоб проглядалися тенденції зміни даних, причини відхилень і можливі рішення. На цьому етапі вирішуються такі завдання обробки даних:

– оцінка планованого стану об'єкта керування;

– оцінка відхилень від планованого стану;

– виявлення причин відхилень;

– аналіз можливих рішень і дій.

Інформаційна технологія керування спрямована на створення різних видів звітів.

Регулярні звіти створюються відповідно до деякого графіка, що визначає час їхнього створення, наприклад, місячний аналіз продажів компанії.

Спеціальні звіти створюються по запитах керівників або коли в організації відбулося щось незаплановане.

І ті, й інші види звітів можуть мати форму підсумовуючих, порівняльних і надзвичайних звітів.

У підсумовуючих звітах дані об'єднано в окремі групи, відсортовані й оформлені у вигляді проміжних та остаточних підсумків по окремих полях.

Порівняльні звіти містять дані, отримані з різних джерел або класифіковані по різних ознаках і використовувані для цілей порівняння.

Надзвичайні звіти містять дані виняткового (надзвичайного) характеру.

Використання звітів для підтримки керування виявляється особливо ефективним при реалізації так званого керування по відхиленнях.

Керування по відхиленнях припускає, що головним змістом одержуваних менеджером даних повинні бути відхилення стану господарської діяльності установи від деяких установлених стандартів (наприклад, від його запланованого стану). При використанні в організації принципів керування по відхиленнях до створюваних звітів пред'являються такі вимоги:

– звіт повинен створюватися тільки тоді, коли відхилення має місце;

– відомості у звіті повинні бути відсортовані за значенням критичного для даного відхилення показника;

– всі відхилення бажано показати разом, щоб менеджер міг уловити існуючий між ними зв'язок;

– у звіті необхідно показати кількісне відхилення від норми.

Основними компонентами інформаційної технології керування є інформація, що зберігається в базі даних, з виконаних фірмою дій і нормативні документи. Вхідна інформація надходить у БД із систем поточного рівня. Вона використовується при формуванні вищевказаних звітів. Вихідна інформація синтезується у вигляді управлінських звітів у зручному для прийняття менеджером рішення виді.

Вміст бази даних за допомогою відповідного програмного забезпечення перетворюється в періодичні й спеціальні звіти, що надходять до фахівців з прийнятті рішень в організації. База даних, використовувана для одержання зазначеної інформації, повинна складатися із двох елементів:

– даних, що накопичують на основі оцінки дій, проведених фірмою;

– планів, стандартів, бюджетів й інших нормативних документів, що визначають планований стан об'єкта керування (підрозділів установи).

9.4. Інфологічні моделі

Інфологічні моделі – це один з прогресивних напрямків в інформаційних технологіях. Мета інфологічного моделювання – забезпечення найбільш природних для людини способів збору й відображення інформації, що передбачається зберігати в створюваній базі даних. Тому інфологічну модель даних намагаються будувати за аналогією з природною мовою (остання не може бути використана у чистому вигляді через складність комп'ютерної обробки текстів і неоднозначність будь-якої природної мови).

9.4.1. Основні поняття й визначення

Основними конструктивними елементами інфологічних моделей є сутності, зв'язки між ними і їхні властивості (атрибути).

Сутність – будь-який помітний об'єкт (об'єкт, що можна відрізнити від іншого об'єкта), інформацію про який необхідно зберігати в базі даних. Сутностями можуть бути люди, предмети, автомобілі, рейси, смаки, кольори й т.п. Наприклад, типом сутності може бути «Автомобіль», а екземпляром сутності – «Таврія», «Мерседес», «БМВ» і т.д.

Атрибут – поіменована характеристика сутності. Його найменування повинно бути унікальним для конкретного типу сутності, але може бути однаковим для різного типу сутностей (наприклад, «Вага» може бути визначена для багатьох сутностей: «Автомобіль», «Жінка», «Ковбаса» і т.д.). Атрибути використовуються для визначення того, яка інформація повинна бути зібрана про сутності. Прикладами атрибутів для сутності «Автомобіль» є «Модель», «Ціна», «Номерний_Знак», «Колір» і т.д. Тут також існує розходження між типом і екземпляром. Тип атрибута «Колір» має багато екземплярів або значень: «Жовтий», «Сірий», «Рожевий», «Хакі» і т.д., однак кожному екземпляру сутності привласнюється тільки одне значення атрибута. Абсолютне розходження між типами сутностей й атрибутами відсутнє.

Ключ – мінімальний набір атрибутів, за якими можна однозначно знайти необхідний екземпляр сутності. Мінімальність означає, що виключення з набору будь-якого атрибута не дозволяє ідентифікувати сутність по атрибутам, що залишилися. Для сутності «Список_Маршрутів» ключем є атрибут «Номер_Рейса» або набір: «Початковий_Пункт», «Час_Відправлення», «Кінцевий_Пункт» за умови, що з пункту в пункт відправляється в кожен момент часу лише один автомобіль.

Зв'язок – асоціювання двох або більше сутностей. Якби призначенням бази даних було тільки зберігання окремих, не зв'язаних між собою даних, її структура могла б бути дуже простою. Однак одне з основних вимог до організації бази даних – це забезпечення можливості відшукання одних сутностей за значеннями інших, для чого необхідно встановити між ними певні зв'язки.

9.4.2. Опис зв'язків

При побудові інфологічних моделей можна використати мову ER-діаграм (Entity-Relationship, тобто сутність-зв'язок). У них сутності зображуються позначеними прямокутниками, асоціації – позначеними ромбами або шестикутниками, атрибути – позначеними овалами, а зв'язки між ними – ненаправленими ребрами, над якими може проставлятися ступінь зв'язку (1 або буква *M*, що замінює слово «many») і необхідне пояснення.

Між двома сутностям, наприклад, *A* и *B*, можливі чотири види зв'язків:

– зв'язок **«Один до одного»** (1:1). У кожен момент часу кожному екземпляру сутності *A* відповідає жодного або один екземпляр сутності *B* (рис. 9.1). Студент може не одержувати стипендію, одержувати звичайну або одну з підвищених стипендій;

– зв'язок **«Один до багатьох»** (1:*M*). Одному екземпляру сутності *A* відповідають жодного, один або кілька екземплярів сутності *B* (рис. 9.2). Квартира може бути порожньою, у ній може жити один або кілька мешканців.

Тому що між двома сутностями можливі зв'язки в обох напрямках, то сюди ж можна віднести зв'язки типу **«Багато до одного»** (*M*:1);

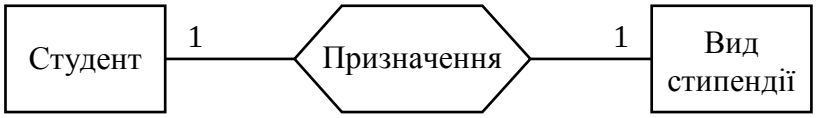


Рисунок 9.1 – Зв'язок «Один до одного»

Рисунок 9.2 – Зв'язок «Один до багатьох»

– зв'язок **«Багато до багатьох»** (*M*:*M*). Наприклад, якщо зв'язок між сутностями «Чоловік» і «Жінка» називається «Шлюб», існує чотири можливих реалізації такого зв'язку (рис. 9.3).

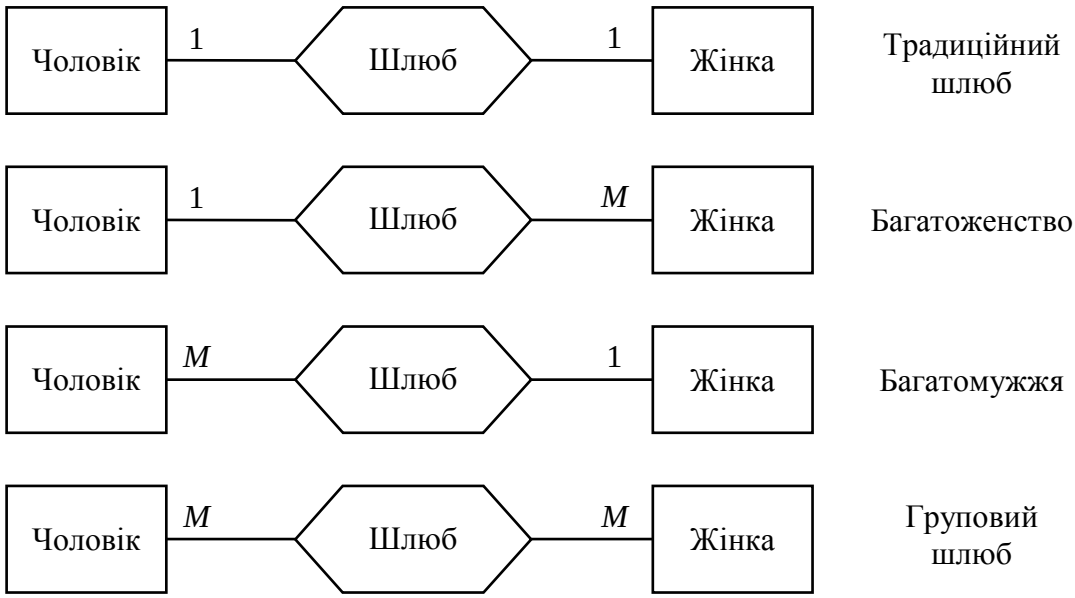


Рисунок 9.3 – Зв'язок «Багато до багатьох»

Характер можливих зв'язків між сутностями не обмежується перерахованими найпростішими. Існують і більш складні зв'язки [6].

Питання й тести для самоконтролю

1. Дайте визначення поняттю «інформаційна технологія».
2. Чим відрізняється інформаційна технологія від технології матеріального виробництва і що між ними спільного?
3. Перелічіть інструментарій інформаційних технологій.
4. Як співвідносяться інформаційна технологія та інформаційна система?
5. Опишіть технологічний процес переробки інформації.
6. Перелічіть етапи вирішення проблем на шляху інформатизації суспільства.
7. Як розвивалися переваги, привнесені комп'ютерними технологіями?
8. Які Ви знаєте види інформаційних технологій?
9. Опишіть основні поняття й визначення інфологічної моделі.
10. Які існують види зв'язку сутностей в інфологічній моделі?

БІБЛІОГРАФІЧНИЙ СПИСОК

1. Диго С.М. Базы данных. – М.: Московский международный институт эконометрики, информатики, финансов и права, 2004. – 177 с.
2. Кузьмин В. Microsoft Office Excel 2003. Учебный курс. – СПб.: Питер; Киев: Издательская группа BHV, 2004. – 493 с.
3. http://www.freecopy.ru/
4. Інформаційні системи в менеджменті: Навчальний посібник // Батюк А.Є., Двуліт З.П., Обельовська К.М., Огородник І.М. та ін. – К.: Інтелект-Захід, 2004. – 520 с.
5. Інформаційне забезпечення менеджменту // Новак В.О., Макаренко Л.Г., Луцький И.Г. – К.: Кондор, 2006. – 462 с.
6. Тлумачний словник з інформатики / Г.Г.Півняк, Б.С.Бусигін, М.М.Дівізінюк та ін. – Дніпропетровськ: Нац. гірн. ун-т, 2008. – 599 с.
7. Информационные системы: Учебное пособие для вузов / Под ред. В.Н.Волковой, Б.И.Кузина. – СПб.: Изд-во СПбГТУ, 1998. – 213 с.
8. Информационные системы в экономике / Под ред. В.В.Дика. – М.: Финансы и статистика, 1996. – 374 с.
9. Харитонова И.А. Microsoft Access 2007. Учебный курс. – СПб.: Питер; Издательская группа BHV, 2008. – 580 с.

ПРЕДМЕТНИЙ ПОКАЖЧИК

MS Access 125
MS Excel 73
MS Office 75

Атрибут 214

База даних 19, 34
– знань 20
Банк даних 36

Властивість об'єкта 104

Дані 19
–, можливі типи 130
–, обробка 209
Діаграма 97

Експорт таблиць 149
Електронна таблиця 74

Забезпечення 39
– адміністративне 49
– інформаційне 40
– лінгвістичне 42
– організаційно-методичне 49
– програмне 41
– технічне 46
Запит 152

Звідна таблиця 93
Звіт 190
Зв’язок 215
– «Багато до багатьох» 216
– «Багато до одного» 216
– «Один до багатьох» 215
– «Один до одного» 215

Зв’язування таблиць 138
Змінна 110
–, можливі типи 111
Знання 19

Імпорт таблиць 149
Інвертовані файли 63
Інформаційна система 9, 205
– автоматизована 9, 57
– автоматизованого проектування 59
– автоматична 57
– інтегрована (корпоративна) 59
– керування технологічними процесами 58
– організаційного керування 58
– ручна 57
Інформаційна технологія 201, 205
Інформаційне суспільство 12
Інформація 17

Керування 20
Керуючі конструкції 113
Класифікація 54
– баз даних 59
– банків даних 69
– інформаційних систем 55, 57
– СУБД 67
Ключ 215
Книга MS Excel 75
Комірка MS Excel 76
Конструктор запитів 154
– звітів 198
– таблиць 128
– форм 181
Конструкція If...Then 114
– If...Then...Else 115
– If...Then...Elself...Else 115
– Select Case 117
Користувач 37
– кінцевий 38

Майстер діаграм 98
– запитів 153
– звідних таблиць 93
– звітів 192
– таблиць 136
– форм 177
– функцій 89
Маска введення 143
Мова алгоритмічна 43
– VBA 102
Метод 125
Модель бази даних 61
– інфологічна 214
– ієрархічна 61
– мережна 61
– – з різнотипними файлами 62
– реляційна 64
Модуль 106

Об’єкт MS Access 126
Область видимості змінної 112
Обмеження цілісності 140
Оператори вводу-виводу 122

Підсистема 35
Подія 105
Поле таблиці 128
–, індексоване 144
Процедура 106
Процес 28

Розгалуження 114

Система 18
– інформаційно-пошукова 57
– інформаційно-вирішальна 58
– управління автоматизована 25
– управління базами даних 19, 35
– управління базою знань 20

Сутність 214

Таблиця MS Access 126
Таблиця MS Excel 75
Технологія 19
– інтелектуальна інформаційна 19
– інформаційна 201
– нова інформаційна 202

Умови на значення 142
Управління 21

Файл 33
Форма екранна 42, 175
– багатосторінкова 185
Формула 86
Функція 88
– InputBox 123
– MsgBox 122

Цикл 114, 119

ДОДАТКИ

Додаток А

Типові завдання для виконання РГР

Розрахунково-графічна робота (РГР) призначена для контролю засвоєння студентами матеріалу навчальної дисципліни. Завдання РГР побудовані таким чином, щоб охопити більшість розділів програми дисципліни. Для свого варіанта предметної області студент повинен виконати та документувати нижчеописані дії.

А.1. Робота з базою даних MS Excel

1. Створити книгу MS Excel за назвою <Ваше прізвище>.

2. У книзі з п. 1 створити аркуш «Вихідні дані».

3. На аркуші книги з п. 2 створити базу даних з такими полями:

– дата продажу (у діапазоні не менше 3-х днів);

– продавець або магазин або кіоск (не менше двох);

– проданий товар (не менше 4-х найменувань);

– кількість проданого товару;

– виторг від продажів.

4. Заповнити базу з п. 3 записами в кількості не менше 50.

5. За допомогою інструмента «Sort/Subtotals» («Сортування/Підсумки») відповісти на окремих аркушах книги на такі питання:

– який продавець краще торгує;

– який товар гірше всього продається.

6. За допомогою інструмента «Filter» («Фільтрація») відповісти на окремих аркушах книги на такі питання:

– який мінімальний виторг одного із продавців:

– які результати торгівлі за два дні підряд.

7. На окремих аркушах книги побудувати зведені таблиці по датах продажів для продавців і товарів.

8. Написати мовою VBA функцію робочого аркуша, що обчислює особистий прибуток продавця (магазина) залежно від суми продажу (виторгу). Якщо виторг становить від 0 до 99 грошових одиниць, то особистий прибуток становить 0 %; від 100 до 499 одиниць – <Половина номера Вашого варіанта> відсотків; 500 або більше одиниць – <Номер Вашого варіанта> відсотків. Ім'я функції – <Ваше ім'я>_<Номер варіанта>.

9. За допомогою функції з п. 8 на окремому аркуші книги з п. 1 (куди попередньо скопійовані записи з аркуша «Вихідні дані») виконати розрахунок особистого прибутку для всіх продажів.

А.2. Робота з базою даних MS Access

1. Створити базу даних MS Access з ім'ям <Ваше прізвище>. Імпортувати в нову таблицю цієї бази дані з аркуша «Вихідні дані» Вашої книги MS Excel з підрозділу А.1.

2. Для таблиці бази даних з п. 1 створити форму введення й доповнити таблицю 20 записами.

3. Для таблиці з п. 2 створити звіт про виторг по календарних числах за проданий товар.

4. Для таблиці з п. 2 створити не менш 3-х запитів на вибірку з використанням умов відбору для полів таблиці. Всі створені в базі дані об'єкти повинні мати розумні імена.

А.3. Побудова інфологічної моделі

1. Для запропонованого варіанта завдання розробити інфологічну модель, визначивши, сутності, атрибути, зв'язки, ключі.

2. За інфологічною моделлю з п. 1 побудувати асоціацію, що відбиває таблицю вихідних даних.

3. Зобразити отриману в п.п. 1, 2 інфологічну модель у вигляді ER-діаграми і мовою інфологічного моделювання.

А.4. Рекомендації з виконання РГР

1. Файли з результатами етапів, виконаних у підрозділах А.1 та А.2, здаються старості групи в електронному вигляді. Імена файлів – прізвища студентів. Староста збирає файли всіх студентів групи на своєму носії даних (рекомендується флеш-пам’ять). Не пізніше зазначеного викладачем строку староста переписує викладачеві файли всіх студентів, наявні в нього на даний момент. Відсутність електронної версії файлу РГР якого-небудь студента (навіть у випадку наявності паперового звіту) прирівнюється до нездачі цим студентом РГР із усіма наслідками, що випливають.

2. До одержання заліку з РГР кожному студенту настійно рекомендується зберігати в себе копію її файлу в електронному виді (для відновлення у випадку ушкодження вірусами або в результаті несправності носіїв даних).

3. РГР повинна бути виконана на аркушах формату А4 і містити:

– опис використаних засобів для підрозділів А.1 й А.2 (на початку кожної частини);

– роздруківку всіх аркушів книги MS Excel з підрозділу А.1;

– роздруківку тексту функції VBA з п. 8 підрозділу А.1;

– копії екранів форми і всіх бланків запитів з підрозділу А.2;

– звіт з п. 3 підрозділу А.2;

– граф і опис інфологічної моделі з підрозділу А.3;

– висновки (не більше 1 сторінки) про використання комп'ютерних технологій у проектуванні інформаційних систем.

4. Титульний аркуш РГР оформляється за звичайними правилами (зразок може бути взятий у викладача).

5. Варіанти предметних областей вибираються відповідно до вказівки викладача з наступного підрозділу.

А.5. Варіанти предметних областей

1. М'ясні продукти
2. Квіти
3. Сантехнічне обладнання
4. Хлібобулочні вироби
5. Бакалія
6. Автомобілі
7. Жіночий верхній одяг
8. Джинсовий одяг
9. Пиво
10. Жіноче взуття
11. Жіноча нижня білизна
12. Лікеро-горілчані вироби
13. Молочні продукти
14. Парфумерія
15. Рибопродукти
16. Постільна білизна
17. Комп'ютери й ноутбуки
18. Канцелярські товари
19. Тютюнові вироби
20. Комплектуючі до комп'ютерів
21. Чоловіча нижня білизна
22. Будівельні матеріали
23. Цукерки
24. Телеапаратура
25. Чоловічий верхній одяг
26. Вина
27. Чоловіче взуття
28. Електротовари
29. Меблі
30. Ліки
31. Радіоапаратура
32. Лакофарбові матеріали

Навчальне видання

ГРИЦУНОВ Олександр Валентинович

ІНФОРМАЦІЙНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

Навчальний посібник

Відповідальний за випуск *О. Б. Костенко*

Редактор *М. З. Аляб'єв*

Комп'ютерне верстання *Є. Г. Панова*

Дизайн обкладинки *І. П. Шелехов*

Підп. до друку 23.12.2009 р. Формат 60х84 1/16

Друк на ризографі Ум. друк. арк. 9,7

Зам. № Тираж 500 пр.

Видавець і виготовлювач:
Харківська національна академія міського господарства,
вул. Революції, 12, Харків, 61002
Електронна адреса: rectorat@ksame.kharkov.ua
Свідоцтво суб'єкта видавничої справи:
ДК №731 від 19.12.2001