І.Л. Тхір, Н.П. Тхір

КОНСПЕКТ ЛЕКЦІЙ

з предмету:

"Технології комп'ютерної обробки інформації"

на тему:

для учнів професії
4112 "Оператор комп'ютерного набору"

Тернопіль 2011

І.Л. Тхір, Н.П. Тхір. Конспект лекцій з дисципліни "Технології комп'ютерної обробки інформації" для поглибленого вивчення теми "Система керування базами даних Microsoft Access 2003". Тернопіль: Технічний коледж ТНТУ, 2011р. 76 с.

Розглянуто та затверджено на засіданні методичної комісії комп'ютерних дисциплін

Протокол №_____ від «__»_____20__ р.

Голова _____/ І.Л. Тхір/

© 2011. Комп'ютерна верстка Степан Тхір

ЗМІСТ

ЗМІСТ

Тема1 Поняття про бази даних **5**

Тема 2 Створення БД. робота з таблицями в Access **12**

2.1. Робота з головним вікном БД. Створення та відкриття БД 12

2.2. Створення таблиць. Робота із структурою таблиці 16

2.3 Заповнення та редагування таблиці БД. Пошук даних 27

2.4 Індексування, сортування та зв'язування таблиць БД 31

2.5. Фільтрування даних в таблиці БД 34

Тема 3 Робота з запитами до БД **37**

Тема 4 Робота з формами в Access **54**

4.1 Автоматичне створення форм. Майстри створення форм 54

4.2 Робота з конструктором форм 58

4.3 Робота із вікном форми 65

Тема 5 Робота з звітами в Access **67**

5.1 Типи звітів. Автозвіти. Створення звітів за допомогою майстра 67

5.2. Робота з конструктором звітів 72

5.3. Перегляд та друк звітів. Перетворення звітів в інший формат 74

ТЕМА1 ПОНЯТТЯ ПРО БАЗИ ДАНИХ

База даних (надалі **БД**) або **DataBase (DB)** — це набір впорядкованої інформації, об'єднаної в єдине ціле. База даних об'єднує в собі як набір самих даних, так і спосіб їх організації. Причому над БД можна проводити операції обробки даних (сортування, фільтрування і т.д.).

Термін база даних часто асоціюють з роботою на комп'ютері, хоча насправді це значно ширше поняття. Адже звичайний записник, скринька з картотекою, телефонний довідник також є своєрідними базами даних. Фактично бази даних з'явились ще задовго до появи електронно-обчислювальної техніки.

В комп'ютерній техніці із поняттям бази даних зв'язано дві складові, це власне бази даних (надалі БД) та системи керування базами даних.

Система керування базами даних (надалі **СКБД**) або **DataBase Managment System (DBMS)** — це програма, яка здійснює керування даними та їх обробку в БД.

Однією із характеристик БД є наявність зв'язків між певними даними в базі. За способом встановлення зв'язків між даними в БД (за структурою) їх поділяють на: ієрархічні, сіткові, реляційні, об'єктно-орієнтовані та об'єктно-реляційні.

Ієрархічна структура БД — це структура, в якій дані зв'язані між собою односторонніми залежними вертикальними зв'язками від старших вершин до молодших. Кожен елемент зв'язаний хоча б з одним вищестоячим та декількома нижчестоячими елементами даних. Це полегшує доступ до інформації, але тільки тоді, коли всі можливі запити відображені в структурі дерева. Будь-які інші запити в жодному разі не можуть бути реалізовані. Прикладом ієрархічної структури БД може бути організація каталогів на диску (папок у Windows), різного типу класифікації, структура державної влади і т.д.

На рисунку 1.1. зображено приклад структури ієрархічної бази даних, в яку може бути внесена інформація про складові комп'ютерної системи.

Рисунок 1.1 Приклад ієрархічної структури БД

Сіткова структура БД — це структура, в якій кожен елемент даних зв'язаний зі всіма сусідніми незалежними та рівноправними горизонтальними зв'язками (рис. 1.2.). Теоретично кожен елемент даних повинен бути зв'язаним одночасно зі всіма іншими елементами (зв'язок «всіх із всіма»), але оскільки практично реалізувати це неможливо, то доводиться встановлювати деякі обмеження. Така структура значно розширює можливості створення запитів до БД, але оскільки кожен елемент даних повинен містити посилання на дуже велику групу інших елементів, це призводить до надмірного перенасичення зв'язками. Отже основним недоліком сіткової структури БД є дуже великі затрати як дискової, так і оперативної пам'яті ПК, а, це в свою чергу, знижує швидкість обробки даних. Крім цього, для таких моделей характерна складність реалізації СКБД.

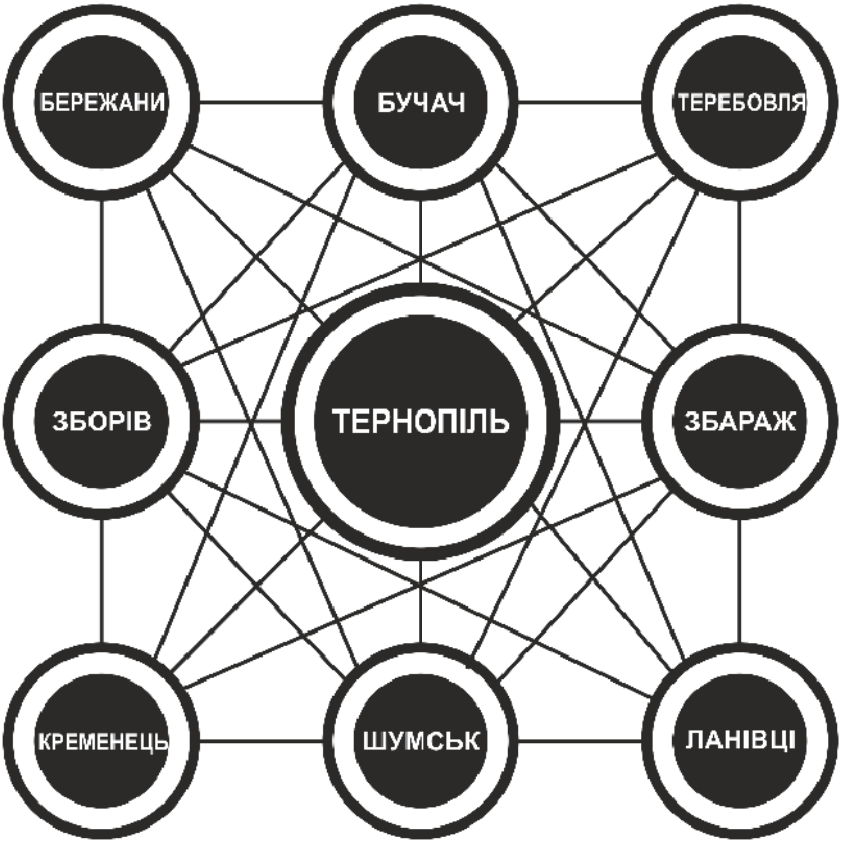
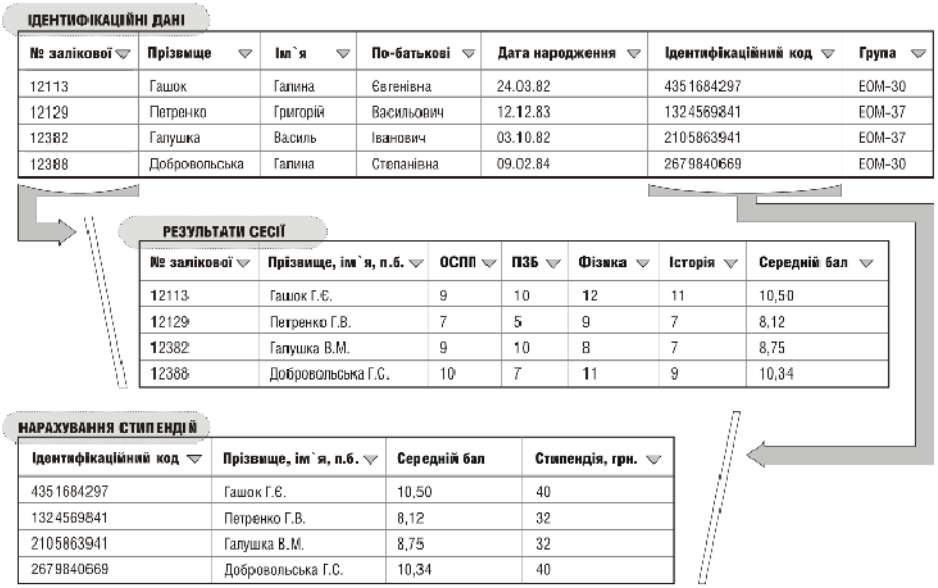


Рисунок 1.2 Приклад сіткової структури БД (Структура автобусних маршрутів)

Реляційна структура БД (РБД) – це структура, в якій дані організовані в формі двомірної таблиці по колонках і рядках (рис. 1.3). В теорії множин таблиці відповідає термін **відношення (relation)**, який і дав назву даному типу структури. Перевагою реляційної моделі є відносна простота інструментальних засобів її підтримки, а недоліком — жорсткість структури (неможливо, наприклад, задати рядки таблиці необмеженої довжини) і залежність швидкості її роботи від розміру БД.

Рядки в реляційній базі даних називають **записами (Record)**, а колонки — **полями (Field)**. Прикладом такої БД може бути база даних працівників певного підприємства. Якщо цю інформацію занести в двомірну таблицю в такому порядку: перша колонка — порядковий номер, друга — прізвище та ініціали, третя — посада, четверта — дата народження, п'ята — домашня адреса, шоста — номер телефону і т.д., то буде отримано звичайну реляційну БД. Полями цієї БД будуть: порядковий номер, прізвища, посада і т.д., а записами повна інформація про кожну особу зокрема.

Крім цього, одна таблиця, що є складовою реляційної БД може містити зв'язки з іншими таблицями. Це дозволяє розмежувати інформацію за певними категоріями і спростити пошук потрібних записів.



ІДЕНТИФІКАЦІЙНІ ДАНІ

№ залікової	Прізвище	Ім`я	По-батькові	Дата народження	Ідентифікаційний код	Група
12113	Гашок	Галина	Євгенівна	24.03.82	4351684297	ЕОМ-30
12129	Петренко	Григорій	Васильович	12.12.83	1324569841	ЕОМ-37
12382	Галушка	Василь	Іванович	03.10.82	2105863941	ЕОМ-37
12388	Добровольська	Галина	Степанівна	09.02.84	2679840669	ЕОМ-30

РЕЗУЛЬТАТИ СЕСІЇ

№ залікової	Прізвище, ім`я, п.б.	ОСПП	ПЗБ	Фізика	Історія	Середній бал
12113	Гашок Г.Є.	9	10	12	11	10,50
12129	Петренко Г.В.	7	5	9	7	8,12
12382	Галушка В.М.	9	10	8	7	8,75
12388	Добровольська Г.С.	10	7	11	9	10,34

НАРАХУВАННЯ СТИПЕНДІЙ

Ідентифікаційний код	Прізвище, ім`я, п.б.	Середній бал	Стипендія, грн.
4351684297	Гашок Г.Є.	10,50	40
1324569841	Петренко Г.В.	8,12	32
2105863941	Галушка В.М.	8,75	32
2679840669	Добровольська Г.С.	10,34	40

Рисунок 1.3 Приклад реляційної структури БД

Об'єктно-орієнтована структура БД (ООБД) або Object-Oriented DataBase (OODB) — це БД, основою якої є особливий тип даних — об'єкти (Object). Будь-який елемент реального світу в об'єктно-орієнтованих БД моделюється у вигляді об'єктів. В реальному світі все, що нас оточує, так або інакше взаємодіє між собою. Фактично, природа в реальному світі є одним великим об'єктом, що складається з багатьох менших, які в свою чергу складаються із ще менших. Кожен об'єкт при його створенні отримує унікальний ідентифікатор, який зв'язаний з об'єктом протягом всього періоду його існування і не змінюється при зміні стану об'єкта.

Кожен об'єкт має стан та поведінку. **Стан об'єкту** — це набір значень його атрибутів. **Поведінка об'єкту** — набір **методів** (програмний код), що керують станом об'єкту. **Методи** — це підпрограми (процедури), що викликаються для того, щоб виконати певні дії над об'єктом. Стан та поведінка об'єктів вбудовані в самі об'єкти. Значення атрибутів об'єкта — це також певний об'єкт або множина об'єктів. Множина об'єктів із однаковим набором атрибутів та методів утворює **клас об'єктів**. База даних — це набір елементів даних, що зв'язані відношеннями "входить в клас" та "є

атрибутом". Взаємодія між об'єктами здійснюється за рахунок передачі повідомлень і виконання відповідних методів (програмних кодів).

Однією із фундаментальних властивостей об'єктів є **наслідування**, тобто породження нового класу на основі вже існуючого. При цьому новий клас називають **підкласом (нащадком)**, а клас, що його породив — **суперкласом (батьківським класом)**. Нащадок наслідує всі атрибути та методи суперкласу, але не завжди має безпосередній доступ до нього. Крім цього, в підкласі можуть бути визначені додаткові атрибути та методи.

Декілька класів можуть мати однакові імена методів та властивостей, навіть якщо вони вважаються різними. Це дозволяє писати методи доступу, які будуть правильно працювати з об'єктами зовсім різних класів, лише б відповідні методи та властивості були визначені в цих класах. В цьому полягає поліморфізм об'єктно-орієнтованих бад даних.

Існує два рівні об'єктно-орієнтованої моделі даних: нижній — структурний та верхній — поведінковий. На структурному рівні підтримуються складні об'єкти та їх ідентифікація. На верхньому рівні визначаються складні методи та способи взаємодії між цими об'єктами.

Об'єктно-орієнтовані бази даних дозволяють представляти складні об'єкти більш безпосередньо аніж реляційні БД. Крім цього, ООБД дозволяють визначати об'єкти як сукупність інших об'єктів. Для сукупностей допускається декілька рівнів вкладеності. Властивості також можуть мати складну структуру і визначаються за допомогою конструктора колекцій. Більше того, вони можуть мати в якості значень не примітивні об'єкти, що дає можливість формувати глибоко вкладені об'єкти структури.

В об'єктно-орієнтованих БД підтримується засіб інверсних зв'язків між двома об'єктами (**бінарний зв'язок**). Така система забезпечує цілісність посилань шляхом встановлення відповідного оберненого посилання відразу після створення прямого посилання. Ще однією перевагою ООБД над реляційними є менша необхідність в зв'язках.

Кожен об'єкт, інформація про який зберігається в ООБД, вважається належним до певного класу, а зв'язки між класами встановлюються за допомогою властивостей та методів класів. Наприклад, розглянемо два класи верхнього рівня (рис. 1.4): "Відділ" та "Службовець". Визначимо також клас "Керівник", який є підкласом класу "Службовець".

Службовець	
Властивість	**Тип**
Ім'я Службовця	Текстовий
Номер Службовця	Цифровий
Місцезнаходження	Текстовий
Список відділів	Список об'єктів посилань:Відділ
Метод	**Встановлює**
Висунути (Відділ)	Посилання на (Керівник)

Керівник	
(Підклас класу Службовець)	
Властивість	**Тип**
Ім'я Службовця	Текстовий
Номер Службовця	Цифровий
Місцезнаходження	Текстовий
Список відділів	Список посилань на: Відділ
Ск. часу керує	Цілий
Звання	Текстовий
Метод	**Встановлює**
Підвищити	Безпосередньо не успадкований

Рисунок 1.4 Приклад об'єктно-орієнтованої структури БД

Об'єкти (екземпляри класу) можуть бути створені шляхом використання методу "Створити екземпляр", який є спільним для всіх класів. Об'єкт класу "Службовець",

що був раз створений зв'язується з "Відділом" за допомогою методу "Найняти" класу "Відділ" (із посиланням на об'єкт "Службовець" в якості аргумента).

Для того, щоб зробити службовця керівником, потрібно застосувати метод "Підвищити" до службовця (аргумент — посилання на "Відділ"). Цей метод створює новий об'єкт класу "Керівник" з дублікатами наслідуваних властивостей, а тоді знищує існуючий екземпляр "Службовця" (використовуючи метод "Знищити", що також спільний для всіх класів). Встановлюється посилання на новий об'єкт (дескриптор об'єкта). Відмітимо також, що метод повинен скорегувати властивість "Список службовців" кожного відділу, що входить в "Список відділів", в яких працює службовець. Це необхідно для забезпечення цілісності БД. Оскільки "Керівник" є підкласом "Службовця", цей "Список службовців" не буде мати ніяких конфліктів (посилання на "Службовця" насправді може бути посиланням на "Керівника", так як він є підкласом "Службовця", тоді як обернене твердження невірне).

Основними сферами застосування об'єктно-орієнтованих баз даних є мультимедійні системи, системи автоматизованого проектування CAD (Computer Aided Design), автоматизовані системи виробництва CAM (Computer Aided Manufacturing), системи автоматизованої розробки програмного забезпечення (CASE), системи керування складними документами, системи засновані на знаннях, інформаційно-пошукові системи (Information Retrieval) та інші бази даних в Internet, просторові бази даних (наприклад, картографічні програми) та інші.

Останніми роками швидкими темпами почала розвиватись гібридна **постреляційна** або **об'єктно-реляційна структура баз даних (ОРБД) (Object-Relation DataBase, ORDB)**, яка поєднує в собі переваги реляційної та об'єктно-орієнтованої моделі даних. При цьому дані у вигляді складних за структурою об'єктів поєднуються із використанням реляційних зв'язків між ними. Такі БД переважно використовують **темпоральну модель збереження даних**, тобто для будь-якого об'єкта даних, що створений в момент часу t_1 та знищений в момент часу t_2, в БД зберігаються всі його стани в часовому інтервалі між t_1 та t_2.

Поняття про СКБД

Як ми вже описували, для керування базами даних на ПК створюють спеціалізовані програми, які називаються **системами керування базами даних (СКБД)**. Будь-який пакет програм СКБД повинен містити:

- середовище користувача, що надає можливість безпосереднього керування даними БД за допомогою клавіатури та миші;
- алгоритмічна мова — для створення вузькоспеціалізованих програм обробки БД;
- компілятор — для надання створеній програмі закінченого комерційного продукту в формі незалежного exe-файлу чи програмного пакету;
- модулі швидкого програмування операцій, які вимагають великих затрат робочого часу користувача (створення звітів, меню користувача та ін.).

Наявність в СКБД мови програмування дозволяє створювати складні системи обробки даних, які орієнтовані на конкретні задачі і навіть для конкретного користувача. Існують також СКБД, які мають в своєму розпорядженні лише мову програмування і не мають оболонки. Вони призначені виключно для програмістів — це системи так званого компіляційного типу, які важко назвати СКБД і тому їх часто називають просто **компілятори баз даних**.

На сьогоднішній день СКБД, що підтримують сіткову модель БД майже не використовуються, оскільки реляційні системи менш вимогливі до апаратного забезпечення і повністю задовольняють більшість користувачів. Група реляційних СКБД представлена на ринку програмних продуктів дуже широко. Це, наприклад, такі системи як Paradox, R:Base, Clarion та ін.

Особливу увагу слід звернути на сім'ю так званих dBASE-подібних СКБД, оскільки вони є фактично родоначальниками цього напрямку розвитку комп'ютерної техніки. Фірма Ashton-Tate ще на початку 80-х років випустила свій програмний продукт dBASE та dBASEII. Ця та наступні версії програми стали настільки популярними, що досить швидко з'явилось відразу кілька фірм-послідовників, які запозичили не лише ідеологію Ashton-Tate, але і її мову. В цю сім'ю крім самої dBASE увійшли також FoxBASE, Clipper, Quick Silver, DBXL, Dbfast. Крім цього у 80-ті роки на основі цих програм створено цілу групу русифікованих програмних продуктів, таких як РЕБУС (русифікована версія dBASEIII-plus), КАРАТ/М-2.0 і 2.1 (русифікована версія FosBASE-plus 2.0), СуперКом-87 (русифікована версія Clipper-87).

Однією з найважливіших характеристик будь-якої СКБД є тип використовуваного транслятора (інтерпретатор чи компілятор). Програма, написана для системи-інтерпретатора, може використовуватись лише за наявності самої системи. На відміну від неї, скомпільована програма є самостійною і може використовуватись без СКБД. Отже, перевагою компіляторів є самостійність створених у них програм, а інтерпретаторів — більша зручність при розробці і налагодженні програм та простота освоєння.

В наступній версії dBASEIV були значно розвинуті засоби доступу до даних, в тому числі реалізовано мову керування запитами SQL. Крім цього було добавлено багато нових функцій і команд. Це привело до того, що досить значний проміжок часу dBASEIV займала провідну позицію на ринку інтерпретаторів СКБД. Поряд з нею розвивалась і СКБД Clipper, перші версії якої були просто компіляторами до dBASE.

Варто звернути увагу на те, що наприкінці 80-их провідну позицію серед СКБД починає займати FoxPro (фірма Fox Software). Це відбулось завдяки винятково високим швидкісним характеристикам системи, зручному графічному інтерфейсу користувача з можливістю детального налаштування параметрів екрану і підтримкою маніпулятора миша. Крім цього FoxPro має засоби для швидкої генерації екранів, звітів та меню, підтримується SQL і робота в мережі.

Переломним моментом в комерційному розвитку dBASE-сумісних СКБД стали 1991-1992 роки. Фірма-розробник dBASE Ashton-Tate злилась з Borland, Clipper (Nantucket) з Computer Association, а FoxPro (Fox Software) з Microsoft. Причому, для перших двох фірм цей крок був вимушеним, оскільки вони мали певні фінансові труднощі. Для Fox Software тоді буда протилежна ситуація — Microsoft зацікавилась FoxPro, враховуючи її популярність і перспективність.

Значною популярністю серед користувачів завжди користувалась СКБД Paradox фірми Borland. Її перша версія була створена в 1985 році і, на відміну від подібних програм, вже мала можливість виконувати пошук інформації в реляційній базі, використовуючи запити. В подальшому Borland ще посилила позиції своєї СКБД Paradox, а Microsoft випустила Microsoft FoxPro для Windows. При цьому Microsoft розвиває і підтримує власну СКБД Access. Зрозуміло, що з ростом популярності Windows ці СКБД перейшли на дану платформу і використовують її можливості, наприклад, технології об-

міну даними. Зверніть увагу, що Microsoft Access не мала DOS-версій і одразу була орієнтована на Windows.

Дана СКБД є першою програмою, яка підтримує метод доступу до даних **ODBC (Open DataBase Connectivity)**, який дозволяє, використовуючи відповідні драйвери і фільтри, працювати з БД практично довільних форматів. Використовуючи ODBC можна отримати доступ до баз даних з інших прикладних програм, наприклад з Excel.

В другій половині 90-х років акцент в розвитку СКБД починає зміщуватись в бік **об'єктно-орієнтованих систем керування базами даних (ООСКБД)**. Хоча такі СКБД вперше з'явились ще на початку 80-х, саме в останні роки XX століття вони набули широкої популярності на даний час нараховується близько 30 типів ООСКБД. Серед них в першу чергу потрібно відзначити Jasmine компанії Computer Associates, GemStone компанії Servio, ONTOS DB, ObjectStore компанії Object Design, O2 компанії O2 Technology, ODB-Jupiter російської компанії НПЦ "Интелтек Плюс" та інші.

В останні роки бурхливо почали розвиватись **об'єктно-реляційні системи керування базами даних (ОРСКБД)**, розвиток яких започаткувала система PosgreSQL. Пізніше на об'єктно-реляційну модель переробили свої ядра СКБД такі гіганти індустрії програмного забезпечення як IBM та Oracle. Інша відома компанія, розробник СКБД, Informix придбала досить потужну ОРСКБД Illustra та вбудувала її в свої продукти.

Щодо класифікації сучасних СКБД, то їх можна поділити за такими характеристиками.

За призначенням і галузями застосування СКБД поділяють на:

- спеціалізовані СКБД, наприклад, для зберігання графічної, геофізичної інформації;
- СКБД загального використання, які призначені для вирішення широкого кола користувацьких задач, наприклад: складський облік, відділ кадрів та ін.

За складністю і функціональними можливостями:

- СКБД для персонального використання (наприклад, Paradox, Access, FoxPro та ін.);
- СКБД, які функціонують в режимі клієнт-сервер (наприклад, Oracle, Informix, Ingres та інші).

СКБД для персонального використання працюють на одному окремо взятому ПК, або в локальній одноранговій мережі. При цьому використання спільної для багатьох користувачів БД практично неможливе.

На відміну від попередніх СКБД, які функціонують в режимі клієнт-сервер, дозволяють працювати з єдиною, спільною для всіх користувачів БД, яка поміщається на файл-сервері. Сервер також здійснює обробку запитів до БД, що поступають від клієнтів і передає їм готові результати. Це дозволяє значно зменшити потік даних, що проходять по мережі. Варто зауважити, що сервер здійснює обробку запитів до бази даних значно швидше, в порівняні з ПК, оскільки в ролі серверу використовують більш потужні комп'ютери.

Для знайомства з СКБД персонального використання автори посібника вибрали Microsoft Access, завдяки наявності всіх функціональних параметрів, необхідних для роботи з реляційними базами. Поряд з тим Microsoft Access є досить простою в освоєнні. Вона також входить в пакет Microsoft Office, що забезпечує додаткову інтегрованість з офісними програмами.

ТЕМА 2 СТВОРЕННЯ БД. РОБОТА З ТАБЛИЦЯМИ В ACCESS

2.1. Робота з головним вікном БД. Створення та відкриття БД

Під базою даних в Access ми розуміємо сукупність семи основних розділів та інших додаткових (створених користувачем):

- **таблиць (Tables)** — це власне таблиці із даними БД;
- **запитів (Queries)** — послідовності команд, збережені під одним іменем, що використовуються для перегляду, аналізу та змінювання даних в одній або декількох таблицях;
- **форм (Forms)** — представлення даних таблиць, або результатів роботи запитів у форматованому, спеціально оформленому, вигляді (картотека, діаграми, зведені таблиці і т.д.);
- **звітів (Reports)** — представлення даних таблиць, або результатів роботи запитів у підготовленому для друку вигляді, переважно для підведення певних підсумків. На відміну від форм, дані у звітах не можливо редагувати (лише переглядати та друкувати на принтері).
- **web-сторінок (Pages)** — представлення даних таблиць, або результатів роботи запитів у форматі HTML-документів для їх публікації в Internet. На відміну від попередніх розділів web-сторінки знаходяться в окремих файлах HTML-формату, а в самій БД розміщені лише своєрідні ярлики цих web-сторінок;
- **макросів (Macros)** — невеликі програми, написані мовою Visual Basic for Application, які автоматизують виконання певних операцій над даними Access. Вони представляють собою одну або декілька макрокоманд, які виконуються послідовно, або в порядку заданому певними умовними операторами, і дозволяють автоматично виконати практично будь-які дії доступні в Access над компонентами БД;
- **модулів (Modules)** — набір процедур та функцій, написаних користувачем мовою Visual Basic for Application (VBA), які використовуються для створення окремої прикладної програми з обробки даної конкретної БД. Ці модулі об'єднуються в **проекти (Project)**, з яких і створюються відповідні прикладні програми. Потрібно також зазначити, що для створення модулів користувач повинен мати достатні знання та навички програмування мовою VBA, вивчення якої не входить в курс даного посібника;

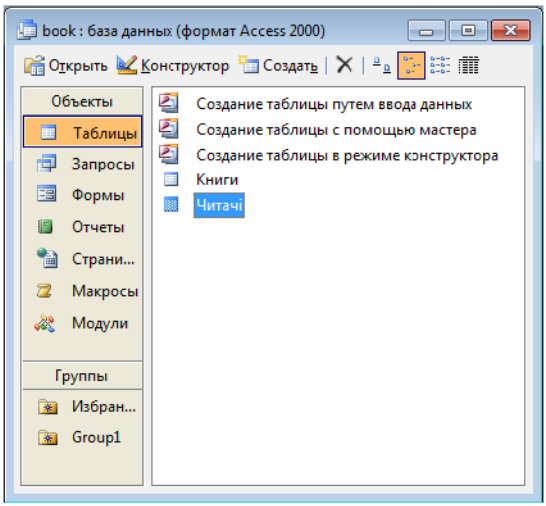


Рисунок 2.1 Головне вікно БД в Access

- **інших груп об'єктів (Groups)** — додаткові групи об'єктів, створені користувачем, куди можна поміщати ярлики до будь-яких об'єктів БД. Таким чином можна об'єднувати різні типи об'єктів у спільні групи.

В більшості випадків в файлі БД використовується не одна таблиця, а декілька. Так само, може бути декілька запитів, форм, звітів, web-сторінок, макросів та модулів. Між таблицями, як правило, встановлюють певні реляційні зв'язки.

Access зберігає базу даних в файлах з розширенням ***.mdb**, куди одночасно входять всі сім розділів (за виключенням хіба що самих web-сторінок). Для роботи із структурою бази даних в Access використовується головне вікно бази даних (рис. 2.1), яке з'являється при відкритті (створені) БД.

Як видно з рисунку 2.1 розділи, з яких складається БД в Access можна вибрати в спеціальній панелі, що знаходиться в лівій частині головного вікна БД. Після вибору відповідного розділу в правій частині відображається список всіх його об'єктів. Крім цього, в правій частині вікна можуть бути присутні спеціальні команди-посилання для створення нових об'єктів цього розділу (рис. 2.1). У верхній частині головного вікна БД знаходяться спеціальні піктограми для: Открыть — відкриття вибраного в активному розділі об'єкта (або двічі ліву клавішу миші на сам об'єкт), Конструктор — відкриття спеціальної підпрограми конструктора для редагування структури вибраного в розділі об'єкта, Создать — створення нового об'єкта в цьому розділі, — знищення вибраного об'єкта, — встановлення режиму відображення великих піктограм об'єктів в розділі, — малих піктограм, — встановлення режиму віідображення об'єктів розділу у вигляді списку і — детального списку із вказанням дати створенння об'єкту, його останньої модифікації та його типу.

Вікно бази даних можна використовувати не лише для того, щоб побачити які об'єкти входять в БД або щоб перейти до одного з них, але це вікно дає значні можливості для роботи з об'єктами. Наприклад, для друку даних, які знаходяться в будь-якій таблиці, потрібно вибрати розділ **Tables (Таблицы)**, а тоді викликати контекстне меню на назві відповідної таблиці і вибрати команду **Print (Печать)**.

Примітка

При роботі з Access головне вікно бази даних може перекриватись іншими вікнами програми. Для швидкого переходу до головного вікна бази даних потрібно натиснути клавішу F11.

Якщо користувач часто використовує один із об'єктів БД (наприклад таблицю БД), то, для спрощення доступу до нього потрібно викликати його контекстне меню і вибрати команду Create Shortcut (Создать ярлык). Відкривається вікно, де потрібно вказати шлях до створюваного ярлика БД та натиснути кнопку OK. За замовчуванням вказано шлях на робочий стіл Windows.

В Access можна виділити три основних способи створення баз даних:

- на основі порожнього бланку БД — це найбільш універсальний, але разом з тим найбільш трудомісткий спосіб, оскільки користувачу доводиться вручну створювати всі її компоненти;
- на основі існуючого файлу БД — використовується для створення точної копії існуючої бази даних. Використовується у випадку, якщо нова БД повинна містити однакові або подібні дані до існуючої БД або для резервного копіювання БД;

- на основі шаблонів БД — створення БД на основі готових шаблонів структури БД та її компонентів. Цей спосіб подібний до попереднього, але в новостворений БД буде використано лише структуру та параметри форматування компонентів шаблону, а самі дані автоматично вводитись не будуть. При виборі більшості із шаблонів завантажується спеціальна підпрограма-майстер створення БД, яка дозволяє покроково створювати базу даних шляхом вибору відповідних параметрів в діалогових вікнах.

Для створення нового файлу БД, потрібно в пункті меню **File (Файл)** вибрати команду **New... (Создать...)** або натиснути піктограму . Можна також скористатись комбінацією клавіш **Ctrl+N**. В будь-якому випадку відкриється панель задач (в правій частині вікна Microrsoft Access) з двох розділів.

В розділі **New (Создание)** можна вибрати один із наступних способів створення БД:

- **Blank Database... (Новая база данных...)** — створення порожнього бланку БД. При цьому спочатку відкривається вікно, де потрібно вказати назву файлу БД та натиснути кнопку **Create (Создать)**. Після цього відбувається перехід до головного вікна БД (рис. 2.1);
- **Blank data access page... (Пустая страница доступа к данным...)** — створення web-сторінки доступу до даних БД. При цьому відкривається вікно, де потрібно вибрати файл джерело даних та натиснути кнопку **Open (Открыть)**. Якщо вибрати командну кнопку **New Source... (Создать...)**, то відкриється майстер підключення даних, де можна вибрати інше джерело даних, наприклад SQL-сервер, Microsoft Jet-сервер та інші;
- **Project using existing data... (Проект с имеющимися данным...)** — створення проекту Microsoft Access із використанням існуючих зовнішніх даних. **Проект Microsoft Access** — це файл Access, який підключається до БД Microsoft SQL Server і використовується для створення прикладних програм обробки БД в структурі клієнт-сервер. Проекти не містять даних або об'єктів визначення даних, таких наприклад, як таблиці. Потрібно зазначити, що створення проектів не входить в обсяг матеріалу даного посібника і детально розглядатись нами не буде;
- **Project using new data... (Проект с новыми данными...)** — створення проекту Microsoft Access із створенням нових даних (див.: попередню команду);
- **From existing file... (Из имеющегося файла...)** — створення бази даних на основі існуючого файлу БД. При цьому відкривається вікно папки, де потрібно вибрати назву файлу БД, з якого потрібно зробити копію та натиснути командну кнопку **Create New (Создать новый)**. При цьому створюється новий файл БД з іменем, що дублює попереднє, але в його кінці додається цифра "1".

Другий розділ **Temlates (Шаблоны)**, панелі задач створення БД, призначений для створення БД на основі шаблонів баз даних. Він містить команди: **On my computer... (На моем компьютере...)** — вибрати шаблон в локальній папці шаблонів даного ПК або **On Microsoft.com (Шаблоны на Microsoft.com)** — завантажити шаблон із web-вузла Microsoft.

При виборі команди створення БД на основі шаблону із локальної папки ПК відкривається вікно із двох закладок. В закладці **General (Общие)** можна вибрати: **Blank Database... (Новая база данных...)** — для створення БД на основі порожнього бланк-

ку, **Blank data access page... (Пустая страница доступа к данным...)** — створення web-сторінки доступу до даних БД, або один із варіантів проекту Microsoft Access (див. попередній матеріал).

В другій закладці власне можна вибрати один із шаблонів створюваної БД. При цьому відкривається додаткове вікно, де потрібно вказати назву файлу БД та натиснути кнопку **Create (Создать)**. Після чого завантажується майстер створення БД, робота якого складається із шести етапів.

На першому етапі роботи майстра відкривається діалогове вікно інформаційного характеру, в якому описуються типи інформації, що будуть поміщені в БД. В даному вікні потрібно вибрати кнопку **Next (Далее)**.

В наступному вікні (рис. 2.2) виводиться список всіх таблиць (**Tables in the database: (Таблицы базы данных:)**), які будуть входити в БД. По черзі вибираючи таблиці в списку справа (**Fields in the table: (Поля таблицы:)**), можна переглянути поля, які пропонуються для кожної із таблиць. Список містить додаткові (не відмічені) поля, що позначені похилим шрифтом. Якщо користувачу потрібно додати до таблиці одне з таких полів, то його потрібно відмітити. Необхідно також зазначити, що поля, відмічені за замовчуванням, відключити неможливо.

Третє вікно майстра встановлює вигляд оформлення форм цієї БД, а на червертому — вигляд її звітів.

У п'ятому вікні необхідно вказати заголовок для БД. Якщо відмітити опцію **Yes, I'd like to include a picture (Добавить рисунок во все отчеты?)**, то у звіти буде поміщено малюнки, що вибираються за допомогою кнопки **Picture (Рисунок)**. Після натискування кнопки **Next (Далее)**, відкривається останнє вікно майстра створення БД, яке містить єдину опцію **Yes, start the database (Запустить базу данных?)** — відкрити новостворену базу.

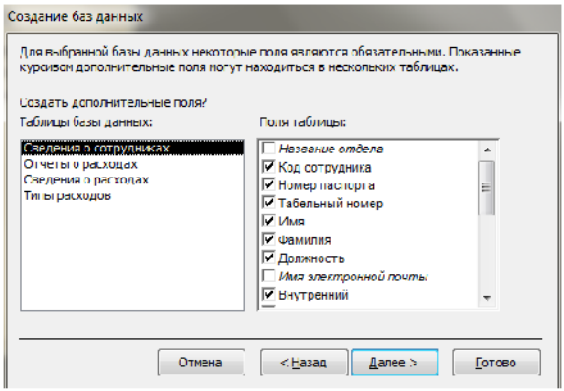


Рисунок 2.2 Друге вікно майстра створення БД

Якщо потрібно відкрити файл БД, то в пункті меню **File (Файл)** необхідно вибрати команду **Open (Открыть)**. Можна також скористатись комбінацією клавіш **Ctrl+O** або натиснути піктограму . В будь-якому випадку відкриється стандартне для Windows-програм вікно відкриття файлу.

Після того, як база даних буде відкрита, або створена нова БД, з'явиться вікно бази даних (рис. 2.1). В лівій частині вікна знаходиться панель з розділів, в яких представлені основні об'єкти БД (в новій БД списки будуть порожні).

Примітка

В Access одночасно не може бути відкрито дві бази даних. При створенні або відкритті іншої БД, база, що була відкрита до цього, автоматично закриється.

2.2. Створення таблиць. Робота із структурою таблиці

Перш ніж створювати нову таблицю, потрібно точно встановити, які вона буде містити поля, тобто намітити собі її структуру. **Структура таблиці** включає в себе назви всіх її полів разом із їх типами та довжиною. Користувач повинен вирішити наперед (ще до створення БД) скільки потрібно таблиць, які атрибути, або поля, вони будуть містити, які типи даних будуть записуватись в полях (символи, числа, грошові одиниці, графіка і т.д.).

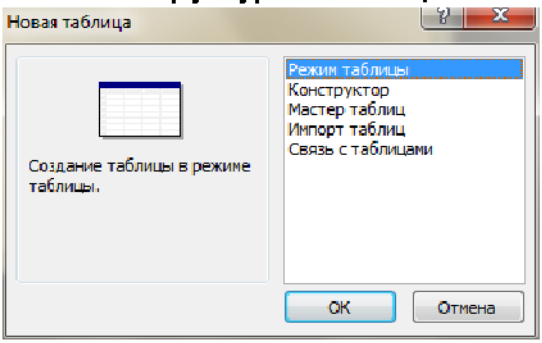


Рисунок 2.3 Діалогове вікно New Table

Для створення нової таблиці потрібно у вікні бази даних (рис. 2.1) вибрати розділ **Tables (Таблицы)** і в ній натиснути кнопку **New (Создать)**. З'являється діалогове вікно **New Table (Новая таблица)** (рис. 2.3), в правій частині якого потрібно вибрати один з п'яти режимів створення таблиці і натиснути кнопку Ok.

Режим Datasheet View (Режим таблицы)

В режимі **Datasheet View (Режим таблицы)** відкривається таблиця, схожа на звичайну електронну таблицю із десяти колонок, куди можна безпосередньо вводити дані. На відміну від аркуша електронної таблиці, заголовки колонок позначаються не латинськими літерами, а назвами полів. За замовчуванням назви полів позначаються **Field1 (Поле1)**, **Field2 (Поле2)** і т.д. Для того, щоб змінити назву поля, потрібно двічі натиснути ліву клавішу миші на його заголовку або викликати на ньому контекстне меню і вибрати команду **Rename Column (Переименовать стобец)**. В результаті цього в заголовку з'явиться курсор, який дозволяє відредагувати назву поля.

Примітка

Як назву поля можна використовувати будь-який набір символів загальною довжиною не більше 64, крім "." (крапка), "!" (знак оклику), "`" (апостроф) та "[ ]" (квадратні дужки), а також вона не повинна починатись із пропуска та керуючих символів. При цьому назви полів не повинні повторюватись в одній таблиці. Бажано, також, щоб імена полів не співпадали із назвами вбудованих в Access функцій та властивостей.

Як ми вже зазначили, таблиця за замовчуванням містить 10 полів. Якщо користувачу недостатньо такої кількості полів, то можна додати нове, викликавши на назві існуючого поля (перед яким потрібно додати нове) контекстне меню і, вибравши команду **Insert Columns (Добавить столбец)**.

Можна також додати спеціальне поле, в якому дані не вводяться, а вибираються із випадаючого списку (список підстановки). Наприклад, якщо необхідно в поточній таблиці використати поле із назвами міст області, а в іншій таблиці ці назви вже введені, то, вказавши даний тип, місто можна буде вибирати зі списку. Для цього потрібно в контекстному меню заголовка поля, перед яким слід вставити нове, вибрати

команду **Lookup Column (Столбец подстановок)**. Завантажується майстер створення списків підстановок, де на першому етапі (рис. 2.4) необхідно вибрати джерело отримання елементів списку.

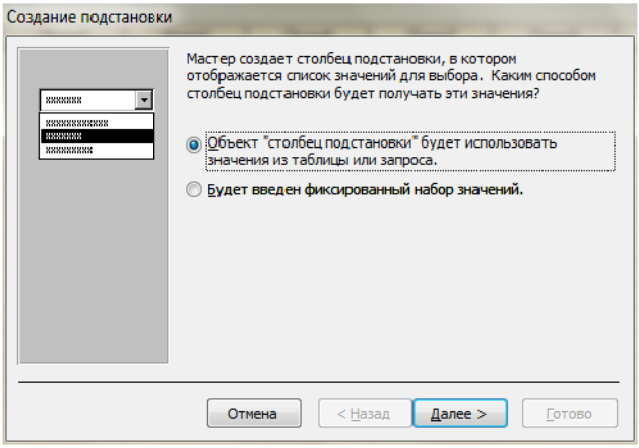


Рисунок 2.4 Перше вікно майстра створення списку підстановки

При встановленні опції **I want the lookup column to look up the values in a table or query (Объект "столбец подстановок" будет использовать значения из таблицы или запроса)** значення списку підстановки будуть вибиратись із поля іншої таблиці БД або запиту. Якщо натиснути **Next (Далее)**, відкривається вікно, де потрібно вказати назву таблиці або запиту та знову натиснути **Next (Далее)**. Після цього відкривається вікно (рис. 2.5), де в списку ліворуч потрібно вибрати назву поля, звідки будуть використовуватись дані для списку підстановки, та натиснути кнопку [>]. В результаті цього відповідна назва потрапляє в список праворуч. Таким чином можна послідовно вибрати декілька полів. Щоб вибрати всі поля одночасно слід натиснути командну кнопку [>>]. Зайві поля можна виключити із вибраного списку кнопками [<] або [<<] — виключити всі поля.

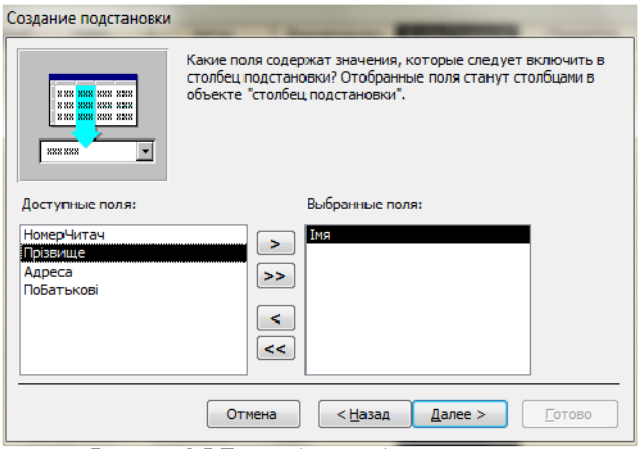


Рисунок 2.5 Третє вікно майстра створення списку підстановки

Після натискування **Next (Далее)** відбувається перехід до наступного кроку роботи майстра, де у випадаючому списку можна вибрати поле, по якому будуть сортуватись дані та вказати напрямок сортування: **Ascending (по возрастанию)** — у порядку зростання або **Descending (по убыванию)** — спадання.

У наступному вікні майстра створення списку підстановки можна вибрати ширину колонки, яка містить список, а в останньому — назву списку підстановки.

Якщо на першому кроці роботи майстра створення списку підстановки (рис. 2.4) вибрати опцію **I will type in the values that I want (Будет введен фиксированный набор значений)**, то елементи цього списку будуть вводитись вручну. Якщо натиснути **Next (Далее)**, відкривається вікно, де потрібно ввести створюваний список, а в

полі **Number of columns: (Число столбцов)** вказати кількість колонок значень цього списку. Після цього відбувається перехід до останнього вікна роботи майстра, де потрібно вказати назву списку підстановки та натиснути клавішу **Finish (Готово)**.

Для знищення у створюваній таблиці зайвих полів потрібно по черзі викликати на їх заголовках контекстне меню і вибрати команду **Delete Column (Удалить столбец)**.

Після вказання структури таблиці можна розпочинати введення даних. По завершенні заповнення даних в таблицю її потрібно зберегти, вибравши в пункті меню **File (Файл)** команду **Save (Сохранить)**. Можна також скористатись комбінацією клавіш **Ctrl+S** або натиснути піктограму . В будь-якому випадку відкриється вікно із єдиним полем, де потрібно вказати назву таблиці. При закритті таблиці виводиться додаткове вікно, де пропонується створити ключове поле. **Ключове поле (Primary Key)** — це поле, яке однозначно ідентифікує кожен запис, тобто в цьому полі дані не повторюються (ідентифікаційний код, номер паспорта, порядковий номер запису і т.д.). При виборі кнокпи **Yes (Да)**, в таблиці створюється ключове поле, в якому буде внесено порядковий номер запису в таблиці.

Примітка

Довжина імені таблиці не повинна перевищувати 64 символів. В імені таблиці можна використовувати пропуски (тільки не починати з них), але не можна використовувати крапки, знаки оклику і квадратні дужки.

Якщо в таблиці не визначене ключове поле, то після введення імені таблиці і натискування кнопки Ok, виводиться діалогове вікно, яке рекомендує вказати ключове поле і перепитує, чи добавити його в структуру.

Вікно створення таблиці за допомогою режиму Datasheet View (Режим таблицы) можна завантажити безпосередньо із головного вікна БД (рис. 2.1). Для цього потрібно у розділі Tables (Таблицы) вибрати команду-посилання Create table by entering data (Создание таблицы путем ввода данных).

Режим майстра таблиць

Режим **Table Wizard (Мастер таблиц)** завантажує підпрограму майстра створення таблиці. При цьому в першому вікні майстра (рис. 2.6) потрібно вибрати в розділ **Sample Tables: (Образцы таблиц:)** шаблон створюваної таблиці. Список шаблонів поділено на дві категорії: **Business (Деловые)** — ділові та **Personal (Личные)** — особисті. Вибравши потрібний шаблон таблиці в розділі **Sample Field: (Образцы полей:)** з'являться перелік рекомендованих полів, які можна добавити до створюваної таблиці. Для цього використовуються спеціальні кнопки: > — додати виділене поле, >> — додати всі доступні поля. Кнопки із стрілками в протилежному напрямку дозволяють забрати поле (поля) зі створюваної таблиці. Якщо користувач бажає змінити ім'я для вибраного поля, то потрібно натиснути командну кнопку **Rename Field (Переименовать поле)**. Вказавши всі необхідні поля, слід натиснути кнопку **Next (Далее)** для переходу до наступного кроку роботи майстра.

У вікні, що відкриється, вказують назву створюваної таблиці та необхідність задання **ключового поля (Primary Key)**. При виборі опції **Yes, set a primary key for me (Microsoft Access автоматически определяет ключ)** Acces буде автоматично додавати ключове поле у вигляді порядкових номерів записів. В іншому випадку ключове поле створене не буде і користувач зможе задати його вручну у конструкторі таблиць.

В наступному вікні майстра створення таблиць можна вибрати таблиці, з якими потрібно зв'язати дану. При цьому потрібно вибрати таблицю для зв'язування та натиснути командну кнопку **Relationships... (Связи...)**. Якщо в БД ще немає жодної таблиці, то вікно зв'язування таблиць з'являтись не буде.

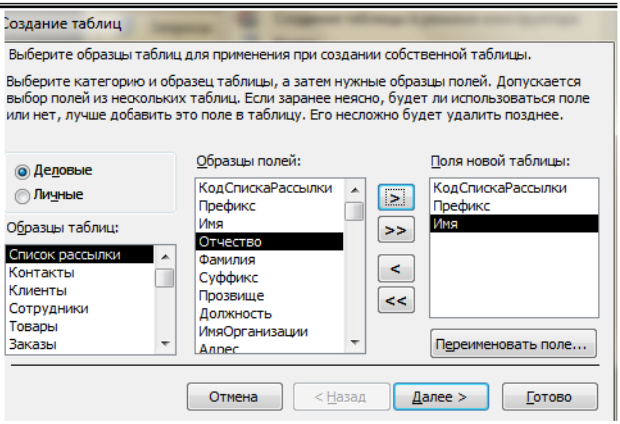


Рисунок 2.6 Третє вікно майстра створення таблиць

Останнє вікно майстра містить такі опції:

- **Modify the table design (Изменить структуру таблицы)** — відкрити вікно конструктора таблиць для відлагодження структури таблиці;
- **Enter data directly into the table (Ввести данные непосредственно в таблицу)** — відкрити вікно таблиці для безпосереднього введення даних;
- **Enter data into the table using a from the wizard creates for me (Ввести данные в таблицу с помощью формы, создаваемой мастером)** — відкрити вікно таблиці для введення даних за допомогою майстра.

Примітка

Вікно майстра створення таблиць можна завантажити безпосередньо із головного вікна БД (рис. 2.1). Для цього потрібно у розділі Tables (Таблицы) вибрати команду-посилання Create table by using wizard (Создание таблицы с помощью мастера).

Режим конструктора таблиць

Досить зручним та універсальним режимом створення БД є режим **Design View (Конструктор)**, який дозволяє керувати процесом створення структури таблиці. При виборі даного режиму і натискуванні Ок, відкривається вікно конструктора таблиці (рис. 2.7).

У вікні конструктора таблиці є три колонки: **Field Name (Имя поля)**, **Data Type (Тип данных)** та **Description (Описание)**. В колонці Field Name (Имя поля) потрібно вказати ім'я створюваного поля (див. попередній матеріал розділу). Якщо користувач ввів в імені поля недопустимі символи, то виводиться інформаційне вікно. Після введення імені поля потрібно натиснути клавішу Tab, щоб перемістити курсор в наступну колонку конструктора — **Data Type (Тип данных)**. Тут необхідно вибрати тип даних, які будуть зберігатися в даному полі. За замовчуванням встановлюється текстовий тип даних. Щоб змінити цей тип, потрібно натиснути на піктограму випадаючого списку, яка розміщена в правій частині комірки, і вибрати необхідний тип.

В Access є 10 типів даних:

- **Text (Текстовый)** — використовується для зберігання будь-якої послідовності символів, включаючи літери, цифри, пробіли і спеціальні символи. Текстові

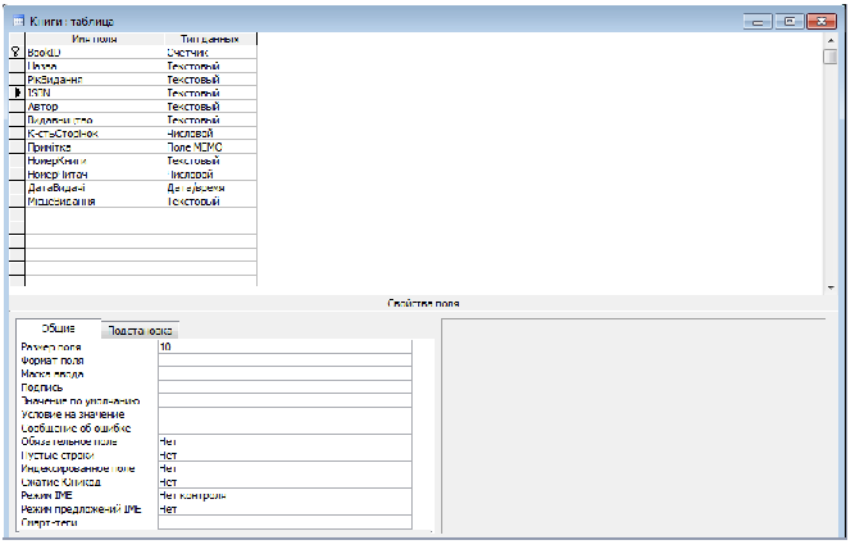


Рисунок 2.7 Вікно конструктора таблиць

поля можуть містити до 255 символів. За замовчуванням довжина текстового поля дорівнює 50-ти символам. Для зміни цієї величини потрібно встановити курсор в полі **Field Size (Размер поля)**, закладки **General (Общие)**, яка знаходиться в нижній частині вікна, і ввести відповідне значення;

- **Memo (Поле МЕМО)** — використовується для зберігання тексту великого об'єму (наприклад, цілих параграфів). В полях МЕМО може зберігатися до 32000 символів. Переважно в цих полях зберігається інформація про службовців, фрагменти юридичних звітів, історії хвороб і т.д. Якщо в полі потрібно зберегти текст цілого документу, то краще встановити тип **OLE Object (Поле объекта OLE)**;
- **Number (Числовой)** — використовується для зберігання числових значень, над якими виконують обчислення. Якщо обчислення над числами виконувати не потрібно (номери телефонів, поштові індекси і т.д.), то краще використовувати текстові поля;
- **Data/Time (Дата/время)** — використовується для зберігання дати або часу. Хоча значення дати і часу можна зберігати і в текстових полях, але використання цього поля має ряд переваг, адже це дозволяє встановити режим автоматичної перевірки правильності введення дати і часу. Крім цього над полями Data/Time (Дата/Час) можна виконувати певні обчислення;
- **Currency (Денежный)** — використовується для зберігання числових значень грошових сум. Автоматично добавляється фіксована кількість цифр для дробової частини числа, щоб попередити помилки заокруглення;
- **AutoNumber(Счетчик)** — використовується для автоматичної нумерації записів таблиці. Поля лічильника можна розглядати як ідентифікаційні номери записів таблиці. Значення цих полів редагувати неможливо;
- **Yes/No (Логический)** — використовується для зберігання логічних величин, тобто таких, які можуть набирати лише двох значеннь **Yes (Да)** або **No (Нет)**;

- **OLE Object (Поле объекта OLE)** — вміщує об'єкт OLE. Об'єкт OLE — це документ іншої прикладної програми Windows, яка підтримує технологію обміну даними OLE. Об'єкт OLE може містити звукову та відеоінформацію, електронні таблиці або документи текстового процесора та ін.;
- **Hyperlink (Гиперссылка)** — використовується для введення адрес (в форматі URL) гіпертекстових документів;
- **Lookup Wizard... (Мастер подстановок...)** — використовується для введення значень, які належать до певного діапазону або списку (список підстановок). При виборі даного типу відкривається вікно майстра створення списку підстановок (рис. 2.4), роботу з яким описано в попередньому матеріалі розділу.

Примітка

Для зберігання текстових повідомлень великого об'єму, замість поля MEMO можна використовувати поле об'єкту OLE. Поля MEMO зручні для невеликого за об'ємом тексту, який складається з кількох абзаців. Для текстів більших об'ємів краще використовувати поле OLE. Для того, щоб вставити OLE-об'єкт, можна скористатись буфером обміну даними. Для цього потрібно у вікні текстового процесора скопіювати фрагмент в буфер обміну даними, а тоді перейти у вікно БД Access і вставити його командою Paste (Вставить) з пункту меню Edit (Правка).

Після того, як буде вказано тип даних відповідного поля, потрібно перейти до третьої колонки конструктора таблиці — **Description (Описание)**. В даній колонці можна вказати опис поля (коментар до поля), який забезпечує користувачів БД довідковою інформацією про дане поле. Текст повідомлення з'являється тоді, коли курсор знаходиться у відповідному полі. Зверніть увагу на те, що цей параметр є необов'язковим.

Після завершення введення опису потрібно знову натиснути клавішу Tab. Курсор переміститься на наступний рядок конструктора, де потрібно ввести параметри наступного поля.

В нижній частині вікна конструктора таблиці знаходиться розділ **Field Properties (Свойства поля)**, де можна встановити властивості кожного поля таблиці. В процесі розробки таблиці на цей розділ можна не звертати уваги, оскільки Access не вимагає, щоб в ньому здійснювались будь-які зміни. Але визначення властивостей полів значно спростить контроль за введенням даних. Наприклад, в текстових полях можна змінити розмір поля, який за замовчуванням дорівнює 50 символам або встановити інші характеристики (формат даних поля, правила, яким повинні відповідати дані і т.д.). Список властивостей залежить від типу вибраного поля. При цьому цей список розділений на дві закладки. Закладка **General (Общие)** містить наступні властивості поля:

- **Field Size (Размер поля)** — встановлює максимальну довжину текстового поля. Для числових полів обмежує допустимі значення, при цьому можна вибрати цілі числа або дійсні;
- **New Values (Новые значения)** — використовується лише для полів типу AutoNumber (Счетчик) і визначає, як потрібно генерувати нові значення: послідовно чи у випадковому порядку;
- **Format (Формат поля)** — встановлює формат відображення даних. В Access існує цілий набір спеціальних шаблонів для створення користувацьких форматів;

- **Input Mask (Маска ввода)** — встановлює шаблон, якому повинні відповідати введені в поле дані;
- **Decimal Places (Дробная часть)** — встановлює кількість цифр в дробовій частині числа;
- **Caption (Подпись)** — встановлює мітку, яка буде з'являтися у формах і звітах;
- **Default value (Значение по умолчанию)** — встановлює значення, яке буде автоматично вводитись в полі (за замовчуванням) при добавлянні нових записів;
- **Validation Rule (Условие на значение)** — встановлює умову, якій повинні відповідати введені у поле дані;
- **Validation Text (Сообщение об ошибке)** — встановлює повідомлення, яке з'явиться у рядку статусу, якщо введені дані не відповідають умові;
- **Required (Обязательное поле)** — встановлює, чи обов'язково вводити дані у поле (**Yes/No (Да/Нет)**);
- **Allow Zero Length (Пустые строки)** — встановлює, чи допускаються в полі порожні текстові рядки;
- **Indexed (Индексированное поле)** — встановлює індекс для поля. Індекси використовуються для швидкого пошуку;
- **Unicode Compression (Сжатие Юникод)** — встановлює режим стискування для символів в двобайтовому кодуванні Unicode. При цьому коди всіх символів, в яких перший байт коду дорівнює 0 будуть архівуватись (див. частину 5 даного посібника) при записі у файл і автоматично розархівовуватись при перегляді даних. Дана характеристика використовується для текстових полів, полів МЕМО та гіперпосилань;
- **IME Mode (Режим IME)** — задає параметри, що керують роботою IME для поля таблиці. **IME (Input Method Editor)** — програма, яка забезпечує введення тексту східно-азіатськими мовами (китайською, корейською, японською) у прикладних програмах шляхом перетворення натискань клавіш на складні знаки цих мов. Її можна вважати додатковим різновидом розкладки клавіатури. Даний параметр використовується для текстових полів, МЕМО та гіперпосилань;
- **IME Sentence Mode (Режим предложений IME)** — встановлює метод визначення IME ділянок, коли фокус курсора переміщається у дане поле (див. попередній параметр);
- **Smart Tags (Самарт-теги)** — дозволяє до вмісту поля додати смарт-теги. Як правило, команди, що поміщаються в смарт-теги полів дозволяють додати безпосередньо з таблиці БД дані про ім'я користувача або дату проведення зборів для папки **Contacts (Контакты)** або **Calendar (Календарь)** програми Microsoft Outlook

Примітка

Приклади форматів поля	
ddd", "d\ mmmm\ yyyy"р."	**виводить "Сб, 8 жовтня 2005р."**
!000,000.00" грн."	**виводить "156 789,56 грн.", вирівнявши по лівому краю поля**
[Red]q"-й квартал "yyyy"р."	**виводить "4-й квартал 2005р." символами червоного кольору**

Примітка

В Access розрізняють поняття порожнього значення та рядка нульової довжини. Якщо поле має значення Null – порожнє, то це означає, що поле не має ніякого значення. Якщо значенням текстового або MEMO-поля є рядок нульової довжини, то значення цього поля відоме – порожній рядок. Порожнє значення не може дорівнювати ніякому значенню, в тому числі й іншому порожньому значенню, тоді як порожні рядки вважаються рівні один одному.

За допомогою умов на вводимі в поля значення можна встановити контроль за їх введенням. Для введення умови потрібно встановити курсор в поле **Validation Rule (Условие на значение)**, розділу **Field Properties (Свойства поля)** і вказати значення умови. Після цього в поле **Validation Text (Сообщение об ошибке)** можна ввести текст повідомлення, яке буде з'являтися при введенні даних, які не відповідають умові.

Умови на вводимі значення задаються як вирази, які включають в себе стандартні математичні символи, що можуть бути виділені з двох боків символом #.

Примітка

Приклади вказання умов:
"NY" or "NJ" or "PA" – введені дані повинні бути або "NY", або "NJ", або "PA".
Between #19/7/01# and 10/10/01# – введена дата повинна знаходитись в діапазоні між 19/7/01 і 10/10/01.
<= #10/11/01# – введена дата повинна бути до 10/11/01.
Like 112M##### – введене значення повинно мати вісім символів, що починаються з "112M".
>= 4.50 and <= 12.00 – введене значення повинно бути між 4.50 і 12.00

Умови можуть включати вирази, які зберігають інші поля даної таблиці. Наприклад, якщо є таблиця із грошовими полями "Основна зарплата" та "Зарплата за надстрокові" і потрібно, щоб значення в полі "Зарплата за надстрокові" не перевищувало в два рази значення в полі "Основна зарплата", то можна використати наступну умову: <=[Основна зарплата] * 2.

Для керування введенням даних і надання даним єдиного вигляду можна також використати шаблони, що встановлюються параметром **Input Mask (Маска ввода)** властивостей полів. При цьому шаблон введення може складатись з трьох розділів, розділених крапкою з комою (;). Перший розділ — власне шаблон (значення шаблонів показані в таблиці 9.5), другий вказує чи зберігати текстові символи з введеним значенням (0 — зберігати разом з введеним значенням, 1 або не вказано — зберігати лише введені символи), а третій — встановлює символ, що буде виводитись в полі введення замість порожніх (не введених) символів. При цьому для третього розділу допускається використання довільного символа, але для введення пропуску потрібно ввести його в подвійних лапках (" "). Якщо третій розділ шаблону не вказати, то замість порожніх символів виводиться "_" (підкреслення).

Примітка

При вказанні шаблону "ISBN 0-&&&&&&&&&-0" можна вводити такі значення полів як: "ISBN 2-34123-405-3"
або "ISBN 1-14-431678-3". Шаблон "AAAA\A00\A" дозволяє вводити значення 4372A23A або "ГРУПА10А"

Таблиця 2.1

Символи шаблонів для форматів числових і грошових полів

Символ	Значення
0	цифра від 0 до 9, введення обов'язкове, "+"/"-" (плюс/мінус) не допускаються
9	цифра або пропуск, введення не обов'язкове,"+"/"-" (плюс/мінус) не допускаються
#	цифра або пропуск, введення не обов'язкове, допускаються"+"/"-" (плюс/мінус)
L	літера від "A" до "Z" або від "А" до "Я", введення обов'язкове
?	літера від "A" до "Z" або від "А" до "Я", введення не обов'язкове
A	літера або цифра, введення обов'язкове
a	літера, цифра або пропуск, введення не обов'язкове
&	будь-який символ, введення обов'язкове
C	будь-який символ, введення не обов'язкове
. ,	розділювач дробової частини числа (залежить від національних налаштувань Windows)
: ; - /	розділювач дати та часу (залежить від національних налаштувань Windows)
<	перевести всі наступні символи в маленькі літери (нижній реєстр)
>	перевести всі наступні символи у великі літери (верхній реєстр)
!	встановлює заповнення вмісту поля справа наліво. Символ "!" в шаблоні можна поміщати в довільне місце
****	будь-який наступний символ буде відображатись як текстова константа (наприклад, \К — виводить символ "К")
« »	символи взяті в подвійні лапки є символьними константами
PASSWORD (ПАРОЛЬ)	будь-який введений символ буде сприйматись як звичайно, але відображатись у вигляді символа "*". Використовується для введеня полів із паролями

Закладка **Lookup (Подстановка)**, вікна конструктора таблиць (рис. 2.7), переважно містить лише один параметр **Display Control (Тип элемента управления)**, який встановлює тип елемента керування для даного поля таблиці: **Tex Box (Поле)** — звичайне поле для введення інформації, **List Box (Список)** — список вибору (використовується для полів підстановок) або **Combo box (Поле со списком)** — список із можливістю введення даних як в поле (теж використовується для полів підстановок). Для поля **Yes/No (Логический)** використовується ще один елемент керування **Check**

Box (Флажок) — опція вибору. Якщо вибрано елемент керування **List Box (Список)** або **Combo box (Поле со списком)**, то дана закладка містить значно ширший набір параметрів, де можна встановити:

- **Row Source Type (Тип источника строк)** — визначає тип джерела елементів списку: **Table/Query (Таблица или запрос)** — таблиця або запит, **Value List (Список значений)** — список значень, введених користувачем або **Field List (Список полів)** — список полів;
- **Row Source (Источник строк)** — визначає джерело даних елементів керування. Якщо вибрати **Value List (Список значений)** — список елементів, розділених ";" (крапкою з комою). В іншомк випадку — ім'я таблиці або запиту;
- **Bound Column (Присоединенный столбец)** — визначає номер колонки списку або поля списку, що містить значення елемента керування. Якщо тут введено число нуль, то в поле передається номер рядка списку;
- **Column Count (Число столбцов)** — визначає кількість колонок, що містить список;
- **Column Heads (Заглавия столбцов)** — визначає, чи використовувати перший елемент списку в якості його назви;
- **Column Widths (Ширина столбцов)** — визначає ширину колонок списку або поля із списком. При використанні декількох колонок значення їх ширини розділяються ";" (крапкою з комою);
- **List Rows (Число строк списка)** — визначає найбільшу кількість рядків, що може відображатись у списку. При перевищенні кількості елементів списку цього значення з'являється лінійка прокрутки;
- **List Widths (Ширина списка)** — визначає ширину списку, що розкривається;
- Limit to List (Ограничиться списком) - визначає чи можуть вводитись в поле із списокм нові значення, що не є елементами списку.

При роботі з таблицями БД важливу роль відіграють ключові поля. **Ключове поле (Primary Key)** — це одне або кілька полів, за якими однозначно ідентифікуються записи в таблиці. Ключові поля дуже корисні, адже вони збільшують швидкодію роботи з таблицею і дозволяють добавляти зв'язки між таблицями.

За замовчуванням ключові поля у списку даних таблиці відображаються першими. В ролі ключових полів можна використовувати лише дані, які ніколи не будуть повторюватись. Найчастіше в цій якості встановлюють номери покупців, соціальних страховок, рахунків, фондів, накладних та ін. Як ключове поле не можна використовувати прізвище, оскільки трапляються випадки коли в одну таблицю входять декілька записів з однаковим прізвищем.

Інколи в якості ключового поля є сенс використовувати комбінацію полів. Наприклад, в таблиці, де ведеться звіт щотижневого робочого часу, як ключове поле можна використати комбінацію полів ідентифікаційного номера службовця і кінця тижня, оскільки жоден службовець не може подати дві заяви про кількість відпрацьованих годин за один тиждень.

Для встановлення ключового поля необхідно в режимі конструктора таблиці вибрати поле (або поля), яке потрібно використати як ключове (для декількох потрібно виділяти при натиснутій клавіші **Ctrl**). Після цього в пункті меню **Edit (Правка)** потрібно вибрати команду **Primary Key (Ключевое поле)** або натиснути піктограму .

Якщо потрібно змінити назву поля, його тип або коментар до поля, досить помістити курсор всередині відповідної комірки вікна конструктора таблиць і зробити потрібні зміни.

Для видалення поля, спочатку необхідно його виділити (натиснути кнопку зліва від назви поля), а тоді вибрати команду **Delete Row (Удалить строки)** з пункту меню **Edit (Правка)** або натиснути клавішу **Delete**. Якщо потрібно добавити нове поле між існуючі, то необхідно відмітити поле, перед яким потрібно вставити нове, а тоді в пункті меню **Insert (Вставка)** вибрати команду **Rows (Строки)** або натиснути клавішу Insert. Для переміщення поля на нове місце потрібно виділити його і, натиснувши ліву клавішу миші на кнопці зліва від нього, перетягнути в інше місце.

Після того, як вся структура таблиці вже сформована, її потрібно зберегти на диску. Для цього потрібно вибрати в пункті меню **File (Файл)** команду **Save (Сохранить)**. Можна також скористатись комбінацією клавіш **Ctrl+S** або натиснути піктограму . В будь-якому випадку відкриється вікно із єдиним полем, де потрібно вказати назву таблиці. Якщо закрити конструктор будь-яким з відомих для вікон Windows-програм способів, то з'являється діалогове вікно, яке перепитує чи потрібно зберігати зміни, внесені в структуру таблиці.

Режим імпортування таблиць

При виборі режиму **Import Table (Импорт таблицы)** відкривається діалогове вікно **Import (Импорт)**, де потрібно вибрати файл для імпортування в таблицю Access. Для цього спочатку у випадаючому списку **Files of Type (Тип файла)** необхідно вибрати тип імпортованого файлу, а тоді папку та ім'я самого файлу і натиснути кнопку **Import (Импорт)**. В результаті цього завантажується майстер імпортування документів, в таблицю Access. Вигляд діалогових вікон даного майстра залежить від типу імпортованого файлу. Переважно, на першому кроці з'являється вигляд імпортованої таблиці та опції, що встановлюють елементи документу, які потрібно показувати в таблиці. На другому кроці встановлюються назви полів таблиці БД. Так, при імпортуванні робочих аркушів Excel пропонується вибрати джерело даних для імпортування: **Show Worksheets (листы)** — цілі робочі аркуші та **Show Named Ranges (именованные диапазоны)** — прийняти за назви полів перший рядок електронної таблиці (**First Row Contains Column Headings (Первая строка содержит заголовки столбцов)**). Після цього з'являється вікно, де потрібно встановити чи створювати нову таблицю, або поміщати інформацію даного файлу в одну з існуючих таблиць. В наступних вікнах необхідно вказати назви невизначених при імпортуванні полів, чи створювати поле із порядковими номерами записів та назву самої таблиці.

Режим зв'язування таблиць

Режим **Link Table (Связь с таблицами)** дозволяє зв'язати певну базу даних із однією або декількома таблицями іншої БД. При цьому зв'язані таблиці будуть використовуватись так, ніби вони належать даній БД. При виборі цього режиму відкривається вікно, де необхідно вказати назву файлу БД (з яким потрібно встановити зв'язок) і натиснути командну кнопку **Link (Связь)**. Відкривається ще одне вікно, де потрібно вибрати таблиці іншої БД, з якими необхідно встановити зв'язок. При виборі таблиць інших форматів (наприклад, таблиця Excel) порядок дій схожий на порядок дій при імпортуванні таблиць.

2.3 Заповнення та редагування таблиці БД. Пошук даних

Для того, щоб БД заповнити даними, потрібно відкрити вікно відповідної таблиці. Тому у головному вікні бази даних (рис. 2.1) потрібно відмітити назву таблиці і двічі натиснути ліву клавішу миші, або вибрати командну кнопку **Open (Открыть)**. В результаті чого вибрана таблиця відкриється в окремому вікні (режим таблиці).

Режим таблиці використовується для додавання нових або редагування вже існуючих записів. Для додавання нових даних потрібно просто ввести їх у поля робочої таблиці, в останній порожній запис (цей запис помічений зірочкою). Коли користувач розпочне введення даних в останній запис, то під ним з'явиться новий порожній запис.

Введення текстової та числової інформації нічим не відрізняється від аналогічних операцій в текстових редакторах та електронних таблицях.

Як правило, в МЕМО-полі не видно всього його вмісту, оскільки в ньому є досить великі об'єми інформації. Для того, щоб побачити весь його вміст, потрібно помістити в нього курсор і натиснути комбінацію клавіш **Shift+F2**. При цьому відкриється спеціальне вікно **Zoom (Область ввода)**, яке також зручно використовувати при введенні інформації в це поле. Після введення або редагування даних в цьому вікні, потрібно натиснути кнопку **Ok**.

Як ми вже говорили, інформацію в поле **OLE Object (Поле объекта OLE)** можна вводити шляхом копіювання OLE-об'єкту через буфер обміну даними. Крім цього, можна викликати на його комірці контекстне меню і вибрати команду **Insert Object (Вставить Объект)**. Відкривається однойменне діалогове вікно, в якому можна відмітити опцію **Create New (Создать новый)** і вибрати тип створюваного об'єкту. З'являється вікно програми в якій створюється даний тип об'єкту. Інша опція **Create from file (Создать с файла)** дозволяє вставити об'єкт із файлу на диску.

Введення даних в логічні поля переважно полягає у тому, щоб відмітити опцію, яка знаходиться в цьому полі. Якщо опція відмічена, то це відповідає правдивому значенню, а відключена — хибному.

Примітка

Якщо потрібно змінити параметри шрифту таблиці, то потрібно помістити курсор в одну з її комірок і в пункті меню Format (Формат) вибрати команду Font (Шрифт). При цьому відкривається стандартне вікно встановлення параметрів шрифту. Зверніть увагу, що параметри шрифту встановлюються відразу для цілої таблиці (в тому числі для назв полів), тобто неможливо змінити ці параметри для однієї окремо взятої комірки або поля.

Введення даних в таблицю можна також здійснювати шляхом копіюваня через буфер обміну даними. Для цього використовуються традиційні комбінації клавіш **Ctrl+X** — вирізати, **Ctrl+C** — скопіювати та **Ctrl+V** — вставити з буфера, або відповідними піктограмами та пунктами меню **Edit (Правка)**. Якщо потрібно вставити дані в кінець таблиці, то потрібно в пункті меню **Edit (Правка)** вибрати команду **Paste Append (Добавить из буфера)**.

В Access подібно до Excel можна виділяти діапазон комірок, що належать до різних полів та записів. Для цього потрібно змістити курсор до лівого краю комірки і утримувати його до тих пір, поки він не перетвориться в символ "+", а тоді натиснути ліву клавішу і перемістити на всі комірки діапазону.

Для редагування даних досить помістити курсор в потрібну комірку таблиці і відредагувати дані або ввести нові. Під час редагування запису зліва від нього з'являється зображення олівця, який вказує на те, що запис редагується. Як тільки перемістити курсор на інший запис, то зміни автоматично записуються в таблицю.

В процесі редагування для переміщення по таблиці використовуються комбінації клавіш, представлені в таблиці 2.2.

Таблиця 2.2

Комбінації клавіш для переміщення по таблиці БД

Клавіші	Дія
← (→)	переміщення на один символ вліво (вправо)
↑ (↓)	перейти до попереднього (наступного) запису
Tab або **Enter**	перейти до наступного поля
Home	перейти до першого поля активного запису
End	перейти до останнього поля активного запису
Ctrl+Home	перейти до першого поля першого запису
Ctrl+End	перейти до останнього поля останнього запису
PgUp	перейти на одну екранну сторінку вверх
PgDn	перейти на одну екранну сторінку вниз
Ctrl+PgDn	перейти на одну екранну сторінку вліво
Ctrl+ PgUp	перейти на одну екранну сторінку вправо

Для переміщення по записах таблиці також використовуються піктограми панелі навігації, що знаходяться в нижній частині вікна режиму таблиці (таблиця 2.3). Для активування кнопки вказання номера активного запису можна натиснути клавішу **F5**.

Таблиця 2.3

Комбінації клавіш для переміщення по таблиці БД

Клавіші	Дія
	перейти до першого запису таблиці
	перейти до попереднього запису
	вказує номер активного запису
	перейти до наступного запису
	перейти до останнього запису
	створити новий запис

В Access неможливо редагувати наступні поля:

- **поля типу AutoNumber (Счетчик)** — значення в полях лічильника збільшуються і зберігаються автоматично;

- **недоступні поля** — коли у властивостях поля для параметру **Enabled (Доступно)** встановити **No (Нет)** або для **Locked (Заблоковано)** встановити **Yes (Да)**;
- **поля в деяких типах запитів** — неможливо редагувати поля в запитах, які встановлюють зв'язки більше ніж з однією таблицею. Також у деяких запитах неможливо редагувати підсумкові значення;
- **обчислювані поля** — оскільки обчислювані поля не існують як окремі поля (вони обчислюються на основі інших полів), їх редагувати неможливо.

Для знищення запису в таблиці БД потрібно спочатку його виділити (натиснути кнопку зліва від рядка запису), або помістити курсор в довільну комірку запису і в пункті меню **Edit (Правка)** вибрати команду **Delete Record (Удалить запись)**. Відкривається діалогове вікно, де потрібно підтвердити свої дії, вибравши кнопку **Yes (Да)**.

В Access процес знищення записів незворотній. Єдиний спосіб відтворити знищений запис – це ввести його заново. Тому при знищенні записів потрібно бути дуже обережним, щоб випадково не видалити потрібні дані.

Під час роботи із таблицями БД можна легко змінювати вигляд даних: розширювати або звужувати колонки (поля), переносити їх у інше місце, ховати і фіксувати. Такі зміни впливають лише на зовнішній вигляд таблиці, внутрішній порядок даних не змінюється.

Для зміни ширини колонки потрібно встановити курсор миші на її правій межі так, щоб він набув форми горизонтальної двонаправленої стрілки. Потім натиснути ліву клавішу і перетягнути цю межу.

Для зміни висоти рядка запису потрібно помістити курсор миші зліва від нього між двома записами так, щоб він набув форми вертикальної двонаправленої стрілки. Потім натиснути ліву клавішу і перетягнути межу. Цю ж операцію можна виконати вибравши команду **Row Height (Высота строки)** з пункту меню **Format (Формат)**. Відкривається вікно, де потрібно вказати числове значення висоти.

Якщо потрібно перемістити колонку в інше місце таблиці, то її необхідно виділити (натиснувши на її заголовку), а тоді натиснути ліву клавішу на назві поля і, не відпускаючи її, перетягнути в інше місце.

Для того, щоб сховати колонку, потрібно виділити її, а тоді в пункті меню **Format (Формат)** вибрати команду **Hide Columns (Скрыть столбцы)**. Коли потрібно показати заховану колонку, необхідно в пункті меню **Format (Формат)** вибрати команду **Unhide Columns... (Отобразить столбцы...)** і в діалоговому вікні, що з'явиться, вибрати відповідну назву колонки. За допомогою даного вікна можна також і заховати колонки, знявши відмітку із назви відповідного поля.

В таблиці БД існує поняття фіксованої колонки. **Фіксовані колонки** — це колонки, які залишаються на місці під час операції горизонтальної прокрутки. Для виконання операції фіксування відповідні колонки потрібно спочатку виділити, а тоді в пункті меню **Format (Формат)** вибрати команду **Freeze Columns (Закрепить столбцы)**. Якщо необхідно зняти фіксацію, то потрібно вибрати команду **Unfreeze All Columns (Освободить все столбцы)** з пункту меню **Format (Формат)**. При цьому фіксація знімається зі всіх колонок.

В Access можна створювати зв'язані таблиці. При цьому одному запису таблиці може одночасно відповідати декілька записів іншої (підлеглої таблиці). Якщо таблиця має підлеглу, то в її лівій частині буде відображатись додаткова колонка (колонка підлеглих таблиць) в якій відображаються піктограми «+». При натискуванні на таку піктограму відображається вміст зв'язаних даних із підлеглої таблиці.

Щоб відобразити підлеглі записи для всіх записів основної таблиці, потрібно в пункті меню **Format (Формат)** вибрати підпункт **Subdatasheet (Подтаблица)**, а в ньому **Expand All (Развернуть все)**. Для виконання зворотньої операції потрібно в цьому ж пункті меню вибрати команду **Collpase All (Свернуть все)**. Щоб забрати колонку відображення підлеглих таблиць, необхідно в меню **Format (Формат)** вибрати підпункт **Subdatasheet (Подтаблица)**, а в ньому **Remove (Удалить)**.

Для того, щоб вставити колонку відображення підлеглих таблиць, потрібно в меню **Insert (Вставка)** вибрати команду **Subdatasheet (Подтаблица)**. Відкривається вікно із трьох закладок із списками типів джерела підлеглих даних: **Tables (Таблицы)** — таблиці, **Queries (Запросы)** — запити та **Both (Таблицы и запросы)** — таблиці та запити. Вибравши у відповідній закладці потрібну таблицю або запит, у випадаючих списках в нижній частині вікна необхідно вказати: **Link Child Fields (Подчиненные поля)** — поле для зв'язку у підлеглій таблиці та **Link Master Fields (Основные поля)** — в основній таблиці.

При спробі закрити вікно таблиці після змін її зовнішнього вигляду, Access запропонує зберегти ці зміни. При цьому потрібно вибрати кнопку **Yes (Да)**, щоб зберегти зміни, або **No (Нет)** — залишити вигляд таблиці без змін.

Пошук та заміна даних в таблиці БД

Із збільшенням об'єму таблиці БД виникає проблема із пошуком потрібної інформації, що вимагає використання додаткових функцій пошуку.

Якщо потрібно знайти певний окремий запис, то необхідно використати команду **Find (Найти)** з пункту меню **Edit (Правка)**, або натиснути комбінацію клавіш **Ctrl+F**. У випадку, коли потрібно знайти велику кількість записів, краще використати можливості створення запитів.

При виборі команди **Find (Найти)** з'являється діалогове вікно із активною закладкою **Find (Найти)** (рис. 2.8), де в полі **Find What: (Образец:)** потрібно ввести шуканий текст. У випадаючому списку **Search: (Просмотр:)** необхідно вибрати напрямок пошуку: **All (Все)** — по всій таблиці БД, **Up (Вверх)** — від курсору до початку таблиці, **Down (Вниз)** — від курсору до кінця таблиці.

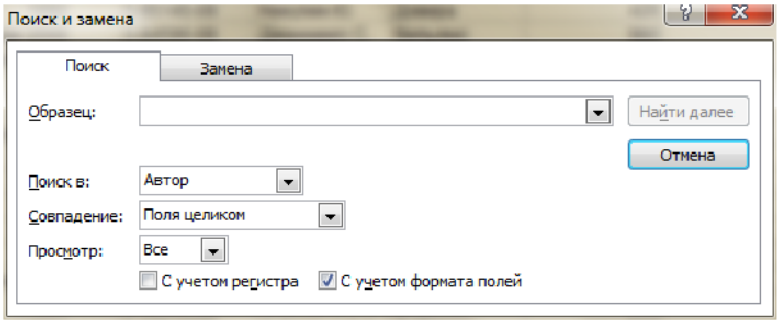


Рисунок 2.8 Закладка Find вікна пошуку та заміни даних

Випадаючий список **Match: (Совпадение:)** встановлює яку частину поля БД повинен становити текст, який ми шукаємо: **Any Part of Field (С любой частью поля)** — будь-яку частину вмісту комірки поля, **Whole Field (Поля целиком)** — весь вміст поля, або **Start of Field (С начала поля)** — вміст поля починається з даного тексту.

В іншому випадаючому списку **Look In (Поиск в)** можна вибрати назву поля, в якому потрібно здійснювати пошук або вказати пошук по цілій таблиці.

Крім цього у вікні пошуку можна встановити додаткові параметри пошуку:

- **Match Case (С учетом регистра)** — розрізняти під час пошуку малі та великі літери тексту;
- **Search Fields as Formatted (С учетом формата полей)** — здійснювати пошук на основі формату відображеного поля.

Після вибору всіх параметрів пошуку потрібно вибрати кнопку **Find Next (Найти далее)**.

Інколи виникає потреба в таблиці змінити один фрагмент тексту на інший. Для цього необхідно натиснути комбінацію клавіш **Ctrl+H**, або в пункті меню **Edit (Правка)** вибрати команду **Replace (Заменить)**. Можна також у вищеописаному вікні пошуку вибрати закладку **Replace (Заменить)**. Вигляд цієї закладки відрізняється лише наявністю додаткового поля **Replace with: (Заменить на:)**, де вказується фрагмент тексту, на який потрібно замінити той, що шукаємо. Крім цього, у закладці **Replace (Заменить)** є дві командні кнопки: **Replace (Заменить)** — замінити знайдений фрагмент та **Replace All (Заменить все)** — замінити всі входження даного фрагменту в поле (таблицю).

2.4 Індексування, сортування та зв'язування таблиць БД

Для таблиць великих розмірів значно ускладнюється процес пошуку даних, навіть якщо використовувати команду пошуку даних. Для пришвидшення цього процесу сучасні СКБД мають можливість сортування та індексування даних.

Виконання операції сортування таблиць в Access подібний до аналогічної операції у Word та Excel, але при цьому не відкривається ніяких додаткових вікон. Для виконання операції сортування даних по відповідному полю потрібно встановити курсор на одній із комірок поля і в пункті меню **Records (Записи)** вибрати підпункт **Sort (Сортировка)**, а в ньому **Sort Ascending (Сортировка по возрастанию)** — для сортування даних в порядку зростання або **Sort Descending (Сортировка по убыванию)** — спадання. Можна також скористатись піктограмою — для сортування в порядку зростання або — спадання.

Індексні поля — це поля, в яких інформація посортована в певному порядку. При цьому в результаті індексування самі записи не пересортовуються. Сортується лише інформація в певному полі, а оскільки до проіндексованого поля прив'язується адреса самого запису, то це дозволяє підтримувати записи впорядкованими по цьому полі. Така структура дозволяє одночасно проіндексувати будь-яку кількість полів. При звертанні до будь-якого індексного поля порядок чергування записів в таблиці змінюється в залежності від порядку встановленого в цьому полі.

Індекси корисні для збільшення швидкості виконання багатьох операцій над БД, особливо при виконанні пошуку і при складанні звітів. Наприклад, маємо таблицю із записами про службовців із ключовим полем за порядковими номерами працівників,

а в звітах записи потрібно впорядкувати за прізвищами. У цьому випадку, якщо таблиця має великий об'єм, можна пришвидшити процес створення звіту, створивши індекс по полю "Прізвище".

Для того, щоб створити індекс по певному полю, потрібно викликати вікно конструктора таблиць (рис. 2.7) та відмітити поле, по якому потрібно створити індекс. Після цього активізувати закладку **General (Общие)**, розділу **Field Properties (Свойства поля)**, і в полі **Indexed (Индексированное поле)** вибрати у випадаючому списку **Yes (Duplicates OK) (Да (Допускаются совпадения))** або **Yes (No Duplicates) (Да (Совпадения не допускаются))**.

Якщо вибрати перший параметр, то допускається присутність більше ніж одного запису з однаковими даними в полі. Коли вибрати **Yes (No Duplicates) (Да (Совпадения не допускаются))**, то Access не дозволяє зберігати декілька записів із однаковими значеннями в полі.

Щоб додати індекс одночасно по декількох полях, потрібно виконати наступну послідовність дій:

1. Відкрити потрібну таблицю в режимі конструктора таблиці.
2. Відмітити поле, по якому потрібно створити індекс.
3. Вибрати піктограму з піктографічного меню Access або в пункті меню **View (Вид)** виконати команду **Indexes (Индексы)**. При цьому відкривається діалогове вікно (рис. 2.9).
4. В колонці **Index Name (Индекс)** потрібно ввести ім'я індексу.
5. Натиснути в колонці **Field Name (Имя поля)** ліву клавішу миші. При цьому з'являється кнопка випадаючого списку. Якщо натиснути цю кнопку, то відкриється список всіх полів таблиці, де потрібно вибрати назву відповідного поля (яке потрібно індексувати).
6. В колонці Sort **Order (Порядок сортировки)** необхідно вказати напрямок сортування: **Ascending (По возрастанию)** — за зростанням або **Descending (По убыванию)** — за спаданням даних.
7. В наступних рядках вікна потрібно повторити пункти 4-6 для всіх інших полів, які необхідно проіндексувати.

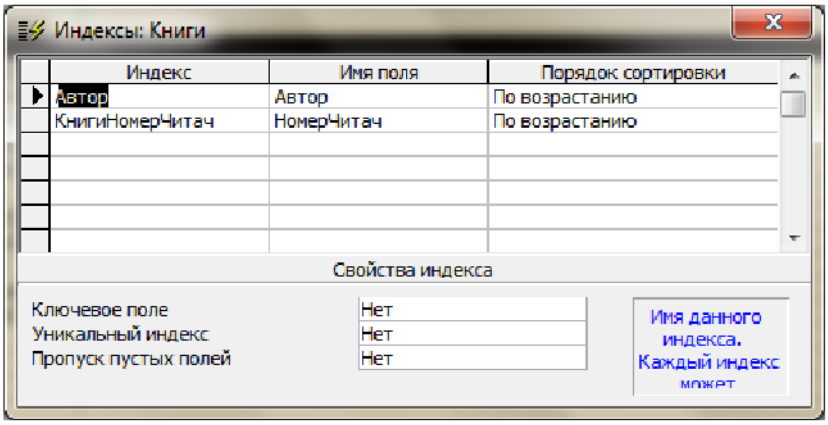


Рисунок 2.9 Вікно індексування полів таблиці

Встановлення зв'язків між таблицями

Одним із найбільш потужніших засобів Access є можливість встановлювати **зв'язки (relation)** між кількома таблицями. Для реалізації цієї можливості потрібно вибрати в пункті меню **Tools (Сервис)** команду **Relationships (Схема данных)**. При цьому встановлюються зв'язки по подібних полях таблиць БД, коли вони містять хоча б частину однакових даних. В результаті виконання команди автоматично відкривається однойменне вікно (рис. 2.10), де можна переглянути всі зв'язки, що встановлені між таблицями БД.

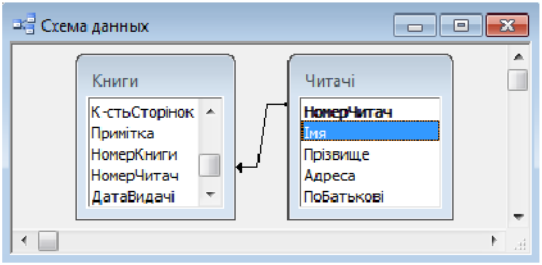


Рисунок 2.10 Вікно Relationships

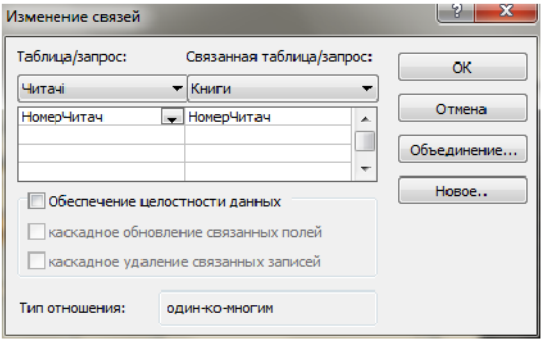


Рисунок 2.11 Діалогове вікно встановлення зв'язку

Встановлені зв'язки автоматично використовуються для з'єднання полів в створюваних запитах, формах і звітах. Хоч встановлювати зв'язки між таблицями зовсім не обов'язково, цей засіб економить час і гарантує отримання правильних результатів при розробці складних форм і звітів.

Другою особливістю встановлення зв'язків між таблицями є підтримка цілісності бази даних. При підтримці цілісності дані автоматично захищаються від змін і видалень, які можуть порушити зв'язки між записами в таблицях.

Якщо користувачу потрібно зв'язати поля двох таблиць, то у вікні **Relationships (Схема данных)** (рис. 2.10), необхідно виділити поле першої таблиці (з тих що зв'язуються). Потім мишею перетягнути його на поле в іншій таблиці (з яким потрібно встановити зв'язок). При цьому відкривається вікно (рис. 2.11), де потрібно переконатись в тому, що правильно встановлено назви полів, які зв'язують. Якщо поля вибрані неправильно, то у випадаючих списках їх можна змінити.

Далі, вибравши командну кнопку **Join Type... (Объединение...)**, потрібно вказати тип зв'язку:

- **Only include rows where the joined fields from both tables are equal (Объединение только тех записей, в которых связанные поля обеих таблиц совпадают)** — зв'язування "один з одним", коли кожному елементу поля першої таблиці повинен відповідати лише один елемент відповідного поля іншої таблиці;
- **Include ALL records from <таблиця1> and only those records from <таблиця2>. where the joined fields are equal (Объединение ВСЕХ записей из <таблиця1> и только тех записей из <таблиця2> в которых связанные поля совпадают)** — зв'язування "багато з одним", коли кільком елементам поля першої таблиці відповідає лише один елемент відповідного поля іншої таблиці;

- **Include ALL records from <таблиця2> and only those records from <таблиця1> where the joined fields are equal (Объединение ВСЕХ записей из <таблица2> и только тех записей из <таблица1> в которых связанные поля совпадают)** — зв'язування "одним з багатьма", коли один елемент поля першої таблиці відповідає декільком елементам відповідного поля іншої таблиці.

Крім цього, в нижній частині вікна зв'язування таблиць (рис. 9.15) можна встановити опції:

- **Enforce Referential Integrity (Обеспечение целостности данных)** — автоматична підтримка цілісності бази даних. Якщо встановити дану опцію, то можна використати наступні опції;
- **Cascade Update Related Fields (Каскадное обновление связанных полей)** — зміни, зроблені у зв'язаному полі першої таблиці, автоматично будуть перенесені в поля зв'язаних таблиць, які містять ті ж самі дані;
- **Cascade Delete Related Fields (Каскадное удаление связанных полей)** — видалення записів у першій таблиці автоматично призводить до видалення відповідних записів в зв'язаних таблицях.

Встановивши всі необхідні параметри, потрібно натиснути кнопку **Create (Создать)**. Якщо необхідно додати до зв'язаних таблиць ще одну, потрібно вибрати кнопку **Create New (Новое)**. Відкривається вікно із чотирьох випадаючих списків, де необхідно вказати назви таблиць, що зв'язують, та полів по яких іде зв'язування.

Примітка

Для знищення зв'язку між таблицями, потрібно у вікні Relationships (Схема данных) відмітити лінію відповідного зв'язку і в контекстному меню вибрати команду Delete (Удалить), або натиснути клавішу Delete.

Щоб змінити тип зв'язку для полів двох таблиць, потрібно відмітити лінію відповідного зв'язку і в контекстному меню вибрати команду Edit RelationShip (Изменить связь).

2.5. Фільтрування даних в таблиці БД

Для вибору із таблиці окремих записів, що відповідають певній умові в Access, подібно до електронних таблиць Excel, використовується функція фільтрування даних. Дана функція за призначенням дуже нагадує запити на вибірку, але, на відміну від запитів, фільтри неможливо зберегти в БД під окремим іменем, для їх повторного використання в майбутньому. Крім цього, фільтри можна застосувати лише для однієї активної таблиці, тоді як запит на вибірку можна створити для декількох зв'язаних таблиць, використовуючи одночасно їх поля. У фільтрах також неможливо вибрати відображення окремих полів таблиці.

В Microsoft Access передбачено чотири способи відбору даних за допомогою фільтрів:

- **фільтр за виділеним фрагментом** — дозволяє вибрати всі записи, що містять певне значення у виділеному полі;
- **поле фільтр для (Filter For)** — дозволяє вибрати всі записи, в яких вміст активного поля відповідає введеному в полі фільтра значенню або виразу;
- **звичайний фільтр** — дозволяє вибрати записи за значенням декількох полів;
- **розширений фільтр** — використовується для складних умов відбору.

Фільтр за виділеним фрагментом, звичайний фільтр та поле фільтр для (Filter For) є дуже простим способом відбору записів. Набір записів, які були відібрані в процесі фільтрування, називають результуючим набором.

Для створення фільтра за виділеним фрагментом потрібно в полі таблиці знайти значення, яке повинні містити записи із результуючого набору даних та виділити його. Після цього, вибрати в пункті меню **Records (Записи)** підпункт **Filter (Фильтр)**, а вньому команду **Filter by Selection (Фильтр по выделенному)**. Фільтр за виділеним фрагментом дозволяє відбирати записи, що не містять вибраного значення. Для цього необхідно виділити значення поля, натиснути на ньому праву клавішу миші і вибрати команду **Filter Exluding Selection (Исключить выделенное)**.

Примітка

Фільтри в Microsoft Access зберігаються автоматично при збереженні таблиці або форми, тому при повторному відкритті таблиці або форми можна знову використати збережений фільтр

Ознакою того, що в таблиці дані відфільтровані, є поява в рядка стану вікна Microsoft Access індикатора FLTR (ФЛТР)

Для того, щоб відмітити застосований фільтр потрібно в піктографічному меню вибрати піктограму , або в пункті меню **Records (Записи)** команду **Remove Filter (Удалить фильтр)**.

Access запам'ятовує останній використовуваний фільтр, тому, якщо знову натиснути на піктограму , або в пункті меню **Records (Записи)** вибрати команду **Apply Filter (Применить фильтр)**, то знову будть відібрані записи останнього фільтра.

Подібним до фільтрування за виділеним фрагментом є застосування поля **Filter For (Фильтр для)**. Його можна застосовувати, якщо не вдається швидко знайти значення для фільтрування, або в якості значення використовується обчислюваний вираз. Для цього потрібно викликати контекстне меню на полі, по якому необхідно відфільтровувати записи, і в ньому вибрати команду **Filter For (Фильтр для)**, а тоді в полі справа вказати значення або обчислюваний вираз (починаючи з символа "=") для фільтрування та натиснути клавішу **Enter**. Щоб застосувати фільтр не закриваючи контекстного меню, слід натсинути клавішу **TAB**. Для відключення даного фільтра також можна скористатись піктограмою , або командою **Remove Filter (Удалить фильтр)** із пункту меню **Records (Записи)**. Отже, піктограма може мати різне призначення в залежності від стану (натиснута або відпущена), а її назва також змінюється в залежності від того, чи встановлений на даний момент фільтр для таблиці.

Для того, щоб використати звичайний фільтр потрібно натиснути піктограму або в пункті меню **Records (Записи)** вибрати підпункт **Filter (Фильтр)**, а в ньому команду **Filter by Form (Изменить фильтр)**. В будь-якому випадку з'явиться форма **Filter by Form (фильтр)** — спеціальне вікно для вказання параметрів фільтра. Воно зовні нагадує вікно самої таблиці, але містить лише один рядок із випадаючих списків, де можна вибрати або ввести з клавіатури значення, які будуть умовами відбору. Якщо умови ввести в декілька полів, то вони будуть об'єднані за допомогою логічного оператора **"І" ("AND")** — для виконання загальної умови, необхідно дотримання одночасно всіх умов, введених в кожному полі зокрема. Для того, щоб застосувати логічний оператор **"АБО" ("OR")**, потрібно вибрати однойменну закладку в нижній частині вікна форми і ввести у відповідні поля наступні умови. Вказавши всі умови фільтрування, слід натиснути піктограму , або вибрати команду **Apply Filter (Применить фильтр)** із пункту меню **Records (Записи)**. Для відміни

звичайного фільтра можна скористатись тією ж піктограмою та командою, що й у випадку двох попередніх типів фільтра.

Звичайний фільтр можна зберегти в базі даних як запит на вибірку даних (див. розділ 3 даної частини посібника), щоб пізніше можна було його неодноразово застосовувати. Для цього потрібно відкрити вікно форми **Filter by Form (фильтр)**, вказати всі умови фільтрування та в пункті меню **File (Файл)** вибрати команду **Save As Query (Сохранить как запрос)** або натиснути піктограму .

Користувач також може імпортувати критерій вибірки із запиту для його використання у звичайному фільтрі. Для цього потрібно завантажити вікно форми Filter by **Form (фильтр)** і в пункті меню **File (Файл)** вибрати команду **Load from Query... (Загрузить из запроса...)**. Відкривається вікно із списком запитів на вибірку, де потрібно вибрати назву відповідного запиту та натиснути кнопку **OK**.

Розширений фільтр є найбільш складним типом фільтрування, що дуже нагадує створення запитів на вибірку даних за допомогою конструктора. Для його встановлення потрібно в пункті меню **Records (Записи)** вибрати підпункт **Filter (Фильтр)**, а в ньому команду **Advanced Filter/Sort... (Расширенный фильтр...)**. При цьому відкривається вікно конструктора розширеного фільтрування, що практично нічим не відрізняється від вікна конструктора запитів (рис. 3.3), роботу з яким описано в третій темі даного посібника. Єдиною відмінністю інтерфейсу вікна конструктора розширеного фільтрування є відсутність рядка **Table: (Таблица:)**, оскільки, як ми вже зазначали, фільтри застосовуються лише для однієї активної таблиці.

Потрібно також відмітити, що створений критерій фільтрування можна зберегти як запит, для того щоб в майбутньому можна було його використовувати в будь-який момент часу. Для цього потрібно вказати всі умови фільтрування та в пункті меню **File (Файл)** вибрати команду **Save As Query (Сохранить как запрос)** або натиснути піктограму .

Як і у випадку звичайного фільтрування, користувач може імпортувати критерій вибірки із запиту для його використання у розширеному фільтрі. Для цього потрібно завантажити вікно конструктора фільтрів і в пункті меню **File (Файл)** вибрати команду **Load from Query... (Загрузить из запроса...)**.

Відфільтровані дані будь-яким із вищеперерахованих способів можна зберегти в іншому файлі або відправити електронною поштою. В першому випадку потрібно в пункті меню **File (Файл)** вибрати команду **Export (Экспорт)**. Відкривається традиційне вікно збереження документу, де в полі **File name: (Имя файла:)** потрібно вказати назву створюваного файлу, а у випадаючому списку **Save as Type: (Тип файла:)** — його тип. Після цього натиснути командну кнопку **Export All (Сохранить все)**.

Для відправки відфільтрованих даних електронною поштою в пункті меню **File (Файл)** слід вибрати підпункт **Send To (Отправить)**, а в ньому команду **Mail Recipient (as Attchment) (Почтовый клиент)**. Відкривається вікно **Send (Послать)**, де необхідно вибрати тип файлу та натиснути **OK**. Після цього завантажується вікно поштового клієнта, в якому потрібно виконати стандартні дії при відправці повідомлення електронної пошти.

ТЕМА 3 РОБОТА З ЗАПИТАМИ ДО БД

Після створення БД та заповнення таблиць даними, можна приступити до наступного етапу розробки бази даних — задання запитів до даних таблиць.

Запити (Query) — це послідовності команд, збережені під одним іменем, що забезпечують простий доступ до певної сукупності полів і записів однієї або декількох таблиць і використовуються для перегляду, аналізу та змінювання даних БД. Вони не містять в собі даних БД, але дозволяють вибрати дані з таблиці та виконати над ними певні дії. Наприклад, можна використати запит для відображення даних, що відповідають певній умові із однієї або декількох таблиць та відсортувати їх у вказаному порядку або здійснити обчислення над групою записів БД. Запити будуються на основі команд спеціальної мови структурованих запитів **SQL (Structure Query Language)**, яка є стандартом для більшості реляційних СКБД. Діючим на даний час стандартом SQL є прийнята Американським національним інститутом стандартів ANSI (American National Standarts Institute) версія **SQL-92**. Фірми-розробники СКБД при реалізації мови SQL можуть вносити в нього розширення, але зобов'язані реалізовувати базовий набір команд **ANSI SQL**. Наприклад, в Access використовується різновид цієї мови **Jet SQL**.

В Microsoft Access використовується декілька способів створення запитів:

- **за допомогою вікна конструктор запитів (Design View)** — один із найбільш універсальних та зручних способів створення запитів більшості типів. Конструктор запитів представляє собою графічний інструмент створення запитів за зразком **QBE** (**Query By Example** — запит за зразком). Цей метод створення запитів розроблений IBM ще в 70-і роки;
- **за допомогою майстра створення запитів (Query Wizard)** — простий та зручний спосіб, що дозволяє створити деякі типи запитів за допомогою спеціальної підпрограми-майстра. При цьому для кожного типу запиту використовується специфічна підпрограма-майстер;
- **шляхом імпортування запиту із фільтра** — створити фільтр та зберегти його як запит;
- **за допомогою вікна SQL View** — безпосереднє введення з клавіатури команд запиту у вікні SQL View. Цей спосіб є найбільш трудомістким та вимагає від користувача достатньо глибоких знань синтаксису мови Jet SQL. Тому його переважно використовують лише для створення специфічних запитів, які неможливо ввести іншими способами, доступними засобами Microsoft Access.

Стандартні запити в Access можна поділити на наступні типи:

- **прості запити (Simple Query)** — дозволяють вибрати із таблиці декілька полів та обчислити для них функції підведення підсумків;
- **запити на вибірку (Select Query)** — дозволяють відобразити лише ті записи таблиці (кількох зв'язаних таблиць), які відповідають певному критерію вибору (фільтрування записів);
- **запити на створення таблиць (Make-Table Query)** — створюють нову таблицю на основі даних однієї або кількох існуючих таблиць;
- **запити на додавання записів (Append Query)** — додають в кінець таблиці дані іншої таблиці;

- **запити на обновлення (Update Query)** — змінюють значення полів в записах таблиці;
- **запити на знищення (Delete Query)** — знищують в таблиці записи, що відповідають певній умові;
- **перехресні запити (Crosstab Query)** — здійснюють статистичну обробку даних, результати якої відображаються у вигляді таблиці, яка дуже нагадує зведену таблицю Microsoft Excel;
- **запити на пошук записів, які повторюються (Find Duplicates Query)** — здійснюють пошук та відображення записів таблиці, які містять однаковий вміст певного поля (наприклад, однакові прізвища);
- **запити на пошук записів, які не мають зв'язків в іншій таблиці (Find Unmatched Query)** — здійснюють пошук та відображення записів, які не мають зв'язаних даних в іншій зв'язаній таблиці;
- **запити на об'єднання записів (Union Query)** — об'єднують в одному результуючому наборі результати декількох запитів та таблиць;
- **запити до сервера (Pass-Through Query)** — використовуються для вибору даних із сервера. На відміну від попередніх типів запитів вони не обробляються безпосередньо процесором Jet SQL, а передаються на сервер БД, до якого виконується звертання, наприклад Microsoft SQL Server;
- **керуючі запити (Data Definition Query)** — дозволяють створювати, змінювати та знищувати таблиці та створювати індекси використовуючи інструкції SQL. Даний тип запиту створюється лише шляхом безпосереднього введення SQL-команд у вікні SQL View;
- інші специфічні запити, що створюються лише у вікні SQL View.

Інколи, при створенні запитів, необхідно використовувати поля, які належать відразу кільком таблицям БД. Для цього між БД слід встановити реляційні зв'язки. Запити які використовують дані відразу з кількох таблиць БД називаються **реляційними запитами**.

Реляційні запити на вибірку можна використовувати для створення **віртуальних таблиць**, тобто набору даних, що належать різним таблицям, але виглядають як одна таблиця. Віртуальні таблиці дозволяють керувати даними, об'єднуючи їх в єдину структуру.

Розглянемо тепер процес створення кожного із типів запитів більш детально. Для створення будь-якого із типів запитів потрібно у головному вікні БД (рис. 2.1) перейти до розділу **Queries (Запросы)** і вибрати в ньому командну кнопку **Создать** **New (Создать)**. При цьому відкривається вікно створення запиту (рис. 3.1), де потрібно вибрати спосіб створення та його тип.

Створення простих запитів (Simple Query)

Прості запити використовуються для вибору з таблиці або декількох таблиць окремих полів для перегляду записів, що в них знаходяться. Даний тип запиту також дозволяє виконувати операції підведення підсумків над полями таблиці. Умови для фільтрування записів в даному випадку не встановлюються.

Створення простого запиту відбувається шляхом вибору команди **Simple Query Wizard (Простой запрос)** у вікні створення запиту (рис. 3.1).

У вікні майстра створення простого запиту (рис. 3.2) з випадаючого списку **Tables/Queries (Таблицы и Запросы)** потрібно вибрати назву таблиці або запиту на

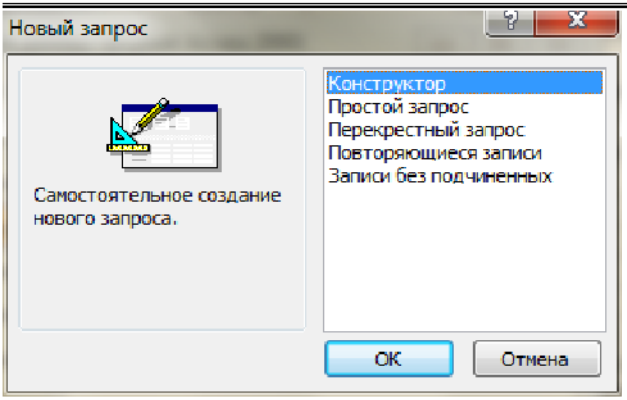


Рисунок 3.1 Вікно створення запитів

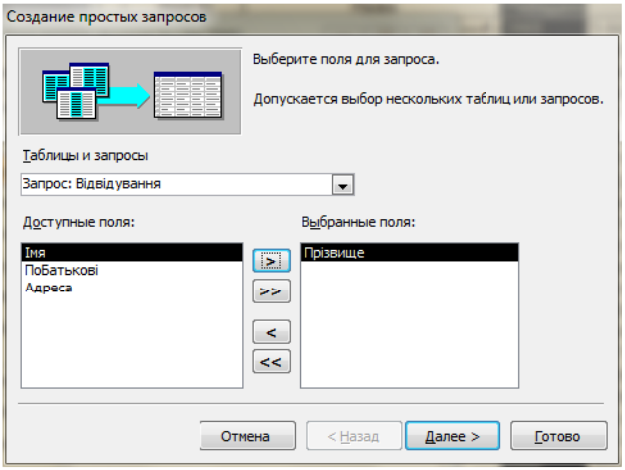


Рисунок 3.2 Перше вікно майстра створення простих запитів

вибірку, на основі якого створюється новий запит. При цьому в нижній частині вікна відображається список доступних полів — **Available Fields (Доступные поля:)**. Вибір полів, які будуть використані в запиті здійснюється за допомогою кнопки > — додати виділене поле, >> — вибрати одночасно всі поля таблиці. Якщо користувачу потрібно додати до даного списку ще й поля із іншої таблиці, то, після вибору полів попередньої, у випадаючого списку **Tables/Queries (Таблицы и Запросы)** слід вказати наступну таблицю і таким самим способом вибрати її поля.

В другому вікні майстра потрібно встановити тип простого запиту: **Detail (подробный)** — в запиті лише відображається вміст вибраних полів, або **Summary (итоговый)** — відображається колонка, в якій виводяться обчислені підсумки. При виборі другого типу потрібно натиснути командну кнопку **Summary Options... (Итоги...)**, для вказання у додатковому вікні функції яка буде визначати підсумки: Sum — сума, Avg — середнє арифметичне, Min - мінімальне значення, Max — максимальне значення та Count records in ... (Подсчет числа записей в ...) — кількість записів.

На останньому етапі роботи майстра необхідно ввести назву створеного запиту та вибрати одну із опцій: Open the query to view information **(Открыть запрос для просмотра данных)** — відкрити запит для перегляду та **Modify the query design (Изменить макет запроса)** — завантажити конструктор запитів для редагування його параметрів. Після цього, натиснути командну кнопку Finish (Готово).

Створення запитів на вибірку (Select Query)

Запити на вибірку використовуються для вибору із таблиці або декількох таблиць записів, що відповідають вказаному критерію фільтрування. При виконанні в Access даного типу запиту результати відображаються в формі **динамічного набору**, який виглядає майже так само як звичайна таблиця, але фактично містить динамічну

інформацію, тобто таку, що змінюється в залежності від ситуації. Записів в динамічному наборі фактично не існує, тому як тільки вікно запиту закривається — всі вони пропадають. Причому сам запит можна зберегти, але дані, що отримані при його виконанні не зберігаються. При збереженні запитів у вигляді операторів SQL зберігається лише їх структура.

Даний тип запитів в Access створюється за допомогою конструктора запитів. Для цього потрібно у вікні створення запитів (рис. 3.1) вибрати команду **Design View (Конструктор)**. При цьому відкривається наступне діалогове вікно **Show Table (Добавление таблицы)**, в якому є список таблиць, що входять в БД. В цьому вікні потрібно встановити над якими таблицями буде виконуватись запит. При цьому необхідно вибрати таблицю в списку і натиснути кнопку **Add (Добавить)**. Цю ж операцію необхідно повторити для всіх наступних таблиць. Нагадаємо, що запити на вибірку можна використати для створення віртуальних таблиць.

Примітка

Потрібно відмітити, що в Access можна створювати запит не лише до таблиці але й до іншого вже існуючого запиту. Тобто із відфільтрованих даних існуючого запиту, можна вибрати ще більш вузький діапазон записів (фільтрування у відфільтрованому). Для цього у вікні Show Table (Добавление таблицы) потрібно вибрати закладку Queries (Запросы) і вказати в ній запити, що будуть джерелом для нового запиту. Закладка Both (Все) виводить спільний список всіх існуючих таблиць та запитів для вибору.

Після вибору всіх необхідних таблиць та запитів у вікні **Show Table (Добавление таблицы)** його потрібно закрити. В результаті чого відбувається перехід безпосередньо до вікна конструктора запитів (рис. 3.3).

Примітка

Для створення реляційних зв'язків між полями таблиць у вікні конструктора запитів (рис. 3.3) потрібно перетягнути поле однієї таблиці (зв'язуване) на відповідне ім'я поля іншої таблиці (з яким необхідно зв'язати). В результаті цього між даними полями з'явиться лінія зв'язку (join lines). Інколи, при добавлянні таблиць у вікно конструктора запитів, реляційні зв'язки між ними додаються автоматично. Це відбувається у випадку, коли зв'язки встановлено на рівні БД, або коли дві таблиці мають однакові поля, сумісні за типом, і одне з них є ключовим. При цьому досить часто виникає багато зайвих зв'язків, які потрібно видалити, відмітивши лінію зв'язку лівою клавішею миші та натиснувши клавішу Delete.

У вікні конструктора запитів необхідно вказати всі поля що будуть брати участь в запиті (як ті, що задають умову для фільтрування так і ті, що лише будуть відображатись). Для цього потрібно встановити курсор миші на назву поля, в списку полів добавлених таблиць у верхній частині вікна конструктора. Тоді натиснути ліву клавішу миші і, не відпускаючи її, перетягнути відповідне поле в першу вільну комірку рядка **Field (Поле)**, що знаходиться в спеціальній таблиці нижньої частини вікна, або двічі натиснути ліву клавішу миші на назві поля у списку полів. Можна також натиснути ліву клавішу миші безпосередньо в комірці цього рядка і вибрати поле з випадаючого списку. В рядку **Table (Имя таблицы)** (рис. 3.3) автоматично з'явиться назва таблиці БД, якій належить дане поле.

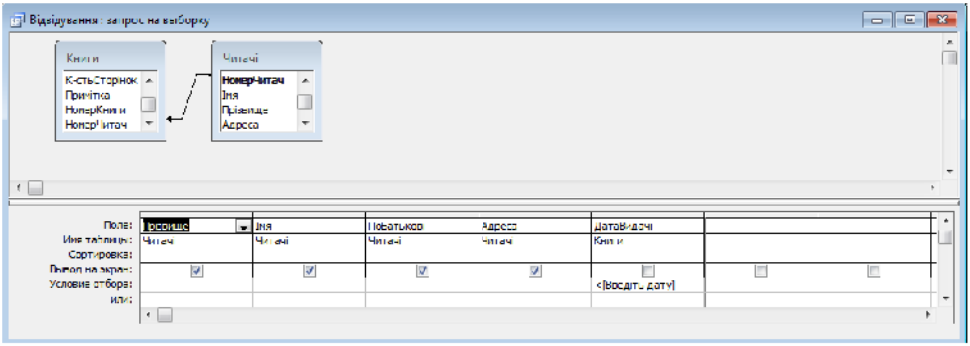


Рисунок 3.3 Вікно конструктора запитів

Якщо в комірку нижньої частини вікна перетягнути замість назви поля символ "*", який знаходиться над списком полів, то в комірку буде поміщено всі поля відповідної таблиці. Для даного набору полів неможливо буде встановити жодної умови.

У відповідній даному полю комірці рядка **Sort (Сортировка)** при необхідності можна вказати порядок сортування записів по цьому полю. Для цього потрібно помістити курсор у відповідну комірку, в результаті чого з'являється випадаючий список, де можна вибрати: **Ascending (по возрастанию)** — сортувати в порядку зростання, **Descending (по убыванию)** — в порядку спадання, **not sorted (отсутствует)** — не сортувати.

В рядку **Show (Вывод на экран)** потрібно встановити, чи буде дане поле виводитись на екран при перегляді таблиці БД. Якщо необхідно, щоб дане поле відображалось, то слід відмітити мітчик опції у відповідній комірці.

Наступні рядки таблиці конструктора запиту (**Criteria (Условие отбора)**) використовують для вказання критерію (умови) відбору даних у відповідному полі. При цьому тип умови залежить від типу поля, для якого ця умова встановлюється. При цьому часто використовуються текстові, числові умови та умови для роботи із датами. В ролі текстової умови вказується текст із вмістом поля потрібних записів. Наприклад, на рисунку 3.4 створено запит на представлення лише записів, в яких поле Name містить текст — «Деркач». Цю умову можна вводити в наступних форматах: «Деркач», Деркач, =Деркач, =»Деркач», Like "Деркач". В будь-якому випадку програма Access автоматично переводить цей текст в формат Like «Деркач». Like — це спеціальний оператор, що визначає що вміст поля починається із вказного фрагменту. Він автоматично додається до текстової умови.

Якщо користувачу зовсім або частково невідомий текст потрібного поля, то можна скористатись шаблонами: * — заміняє будь-яку кількість будь-яких символів, ? — за-

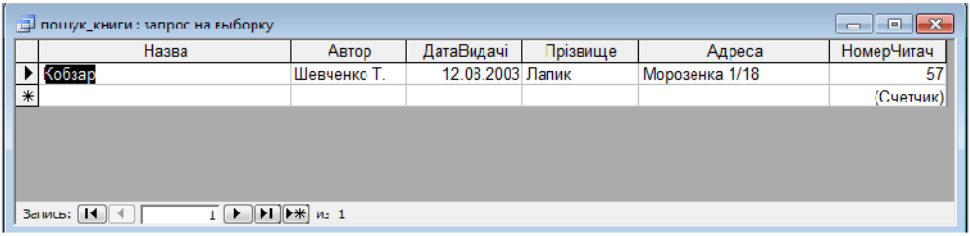


Рисунок 3.4 Приклад вікна з динамічним набором

міняє не більше одного символу (наприклад, «Та*о» - будь-який текст, що починається на «Та» і закінчуються на «о»). Інколи в текстових умовах використовують логічний оператор **not**, який заперечує наявність, в полі таблиці тексту вказаного справа від оператора (наприклад, not «К*» - будь-який текст, крім такого, що починається на літеру «К»).

Числова умова та умова типу дата записується в форматі:

<логічний оператор><вираз>,

де як логічний оператор можна використовувати:

= дорівнює виразу,
< менше від виразу,
> більше від виразу,
<= менше або дорівнює виразу,
>= більше або дорівнює виразу,
<> не дорівнює виразу.

В якості виразів в умовах запитів можна використовувати:

- **константи** — постійні величини, що вводяться в умови як конкретні значення. Їх часто використовують для створення виразів за замовчуванням та для порівняння значень в полях таблиці;
- **ідентифікатори** — імена об'єктів в Access (наприклад, полів таблиць або запитів), які при обчисленні виразів замінюються їх поточними значеннями. Імена полів, таблиць та запитів беруться в квадратні дужки. Назва таблиці відділяється від назви поля символом "!". Наприклад, [Книги]![Ціна] — вміст поля "Ціна" із таблиці "Книги". Для розділення об'єктів та їх властивостей використовується символ "." (крапка). Іменовані константи та змінні, що використовуються в програмах Visual Basic Application, також відносяться до ідентифікаторів. Існує декілька вбудованих констант, що є ідентифікаторами: **True, False, Yes, No** та **Null**;
- **функції** — підпрограми, що виконують певні обчислення і видають в умову результат замість свого імені. Після назви функції в круглих дужках вводяться вхідні параметри для обчислення. Якщо вхідних параметрів функція немає, то записуються порожні дужки. Наприклад, функція **Date()**, що має порожній список аргументів, визначає поточну дату;
- **оператори** — символи та абревіатури, які здійснюють в умові математичні та логічні обчислення.

До математичних операторів відносяться (Таблиця 3.1):

Таблиця 3.1

Математичні оператори

Оператор	Значення
+	додавання
-	віднімання

*	множення
/	ділення
\	цілочисельне ділення
mod	визначає остачу від ділення двох цілих чисел
^	піднесення до ступеня (наприклад, "2^3" дорівняє 8)
&	конкатенація — об'єднання текстових рядків

При використанні оператора ділення націло операнди з десятковими дробами заокруглюються до цілого, а дробові частини відкидаються;

Крім цього, для виконання логічних операцій порівняння використовуються логічні оператори (таблиця 3.2, таблиця 3.3):

Таблиця 3.2

Логічні оператори

Оператор	Значення
and	кон'юнкція (логічне "І") — видає результат True, лише коли обидва вирази дорівнюють True
or	диз'юнкція (логічне "АБО") — видає результат True, коли хоча б один вираз дорівнює True
not	логічне заперечення — змінює значення True на False або навпаки
xor	заперечення "АБО" — видає результат True, якщо порівнювані значення різні
eqv	логічна еквівалентність — видає результат True, якщо порівнювані значення однакові
imp	логічна імплікація — видає результат False, якщо перший вираз True, а другий False. В інших випадках результат True

Таблиця 3.3

Залежності значень логічних операторів від значень операндів

	A=True B= False	A=True B= True	A=False B= False	A=False B= True
A and B	False	True	False	False
A or B	True	True	False	True
Not A	False	True	True	True
A xor B	True	False	False	True
A eqv B	False	True	True	False
A imp B	False	True	True	True

Серед інших операторів в запитах можна використовувати:

Таблиця 3.4

Інші оператори в запитах

Оператор	Значення
between ... and ...	в діапазоні між (наприклад - between 2.20 and 10.20)
is Null	визначає чи значення поля є порожнім значенням
is Not Null	визначає чи значення поля не є порожнім значенням
in(..., ..., ..., ...)	визначає чи значення поля є елементом списку. Наприклад, умова In("Тернопіль", "Збараж", "Львів", "Київ") визначає всі записи, із значенням поля "Тернопіль", "Збараж", "Львів" або "Київ"
Like "...."	визначає, чи починається вміст поля із вказаних символів

Примітка

Приклади задання умов:
«Київ» - записи, для яких текстове поле містить текст «Київ».
Not «Львів» – записи, для яких текстове поле містить будь-який текст крім «Київ».
Like [A-K]* – записи, для яких текстове поле починається з будь-яких літер, які ідуть в алфавіті від «А» до «К».
not in("Львів", "Київ", "Рівне", "Луцьк", "Донецьк") - записи, для яких текстове поле не містить записів "Львів", "Київ", "Рівне", "Луцьк" або "Донецьк".
«Тк??» - записи, для яких текстове поле починається з літер «Тк», а далі в слові може бути ще до двох літер.
>=300 - записи, для яких вміст числового поля більший ніж 300.
Between 2/10/05 and 10/10/05 – записи, для яких поле дати лежить в межах від 2 жовтня 2005 року до 10 жовтня 2005 року.
=([Ціна]>=30) and ([Ціна]<>150) – записи, для яких числове поле [Ціна] має значення більше 30, але не дорівнює 150
<=Date()-60 – записи, для яких вміст поля дати відрізняється від поточної не більше ніж на 60 днів.

Примітка

В Access для запитів можна вказувати умови з параметрами, коли при завантаженні запиту спочатку виводиться діалогове вікно, в якому порібно задати умову для фільтрування. Це дозволяє користувачу використати один запит у різних ситуаціях. Для реалізації даної можливості потрібно в полі для вказання умови у квадратних дужках записати фразу, що буде здійснювати діалог з користувачем для вказання параметрів. Наприклад, можна записати умову: "<=[Задайте максимальне значення зарплати]". При цьому відкривається діалогове вікно із даною фразою і полем для вказання умови.

Примітка

Створення запитів цікаве не лише тим, що можна у вигляді однієї таблиці представити та відібрати дані кількох таблиць, але й дозволяє створювати обчислювані поля в запиті. Тобто в запиті можна створювати поля, які є результатом обчислення над значеннями інших полів. Найпростішим прикладом обчислюваного поля в запиті може бути поле, що об'єднує ім'я та прізвище особи. Щоб створити обчислюване поле, потрібно в рядок Field (Поле) вільної колонки конструктора запитів (рис. 9.18) ввести потрібний вираз. В нашому прикладі це буде "ПІБ: [Прізвище]&" "&[Ім'я]&" "&[ПоБатькові]".

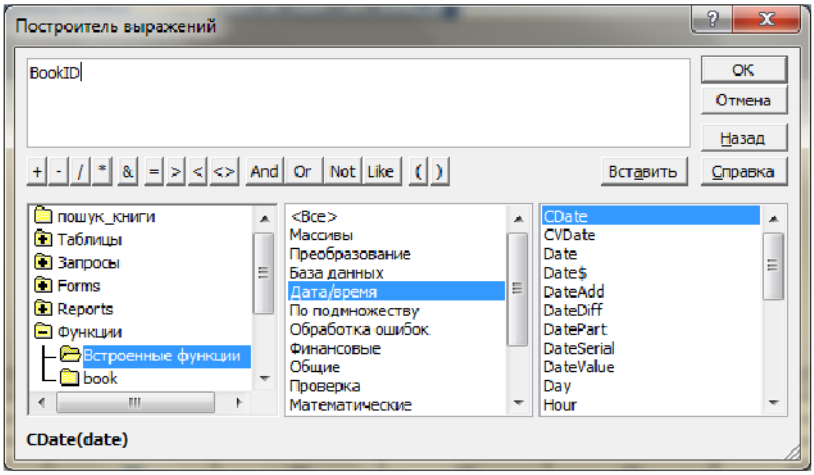


Рисунок 3.5 Вікно побудови виразів

Для автоматичного створення умов можна скористатись спеціальним вікном побудови виразів, яке викликається комбінацією клавіш **Ctrl+F2** або піктограмою . В будь-якому випадку відкривається вікно (рис. 3.5), розділене на декілька розділів. У верхньому розділі міститься спеціальне поле вводу, куди вводиться потрібний вираз. Для введення математичних та логічних операторів можна скористатись спеціальними кнопками під цим полем (рис. 3.5).

Нижня частина вікна розділена на три розділи. Спочатку потрібно в розділі зліва вибрати категорію об'єкта, що повинен бути елементом виразу (таблиці, запити, форми, звіти, функції і т.д.), та сам об'єкт (наприклад, назву таблиці). Після цього в середньомурозділі слід вибрати підлеглий об'єкт (наприклад, поле в таблиці) або категорію функції (фінансові функції, функції для роботи з датою та часом і т.д.). В кінці потрібно вибрати елемент в правому розділі (наприклад, назву потрібної функції). Ввівши всі елементи виразу слід натиснути командну кнопку **OK**.

Досить часто запит не обмежується умовою для одного поля, а об'єднує відразу кілька умов, які відносяться до різних полів. Такі запити використовують для об'єднання умов в комплексні, за допомогою логічних операцій **AND (И)** — логічне «і» та **OR (ИЛИ)** — логічне «або».

Операція логічного об'єднання «і» використовується в тому випадку, коли потрібно, щоб виконувались одночасно обидві умови. В запитах це використовується переважно тоді, коли умова створюється по двох полях одночасно (тобто, коли потрібно щоб задовольнялись умови одночасно для двох полів). В Access логічне «і» між умовами запитів реалізується шляхом їх введення в комірки одного рядка **Criteria (Условие отбора)**. Наприклад, на рисунку 9.18 встановлюється запит на відфільтровування записів, де поле "Назва" починається з літери «К», а поле "РікВидання" - «2000». Записи, для яких вміст хоча б одного з перерахованих полів не містить відповідного тексту, не відповідають умові, а тому не відфільтровуються.

Операція логічного об'єднання «або» використовується в тому випадку, коли потрібно, щоб виконувалась хоча б одна з перерахованих умов. В запитах це використовується переважно тоді, коли прості умови створюються по одному і тому ж полю. При

цьому прості умови записуються у рядки **Criteria (Условие отбора)** в одну колонку (яка відповідає потрібному полю). Наприклад, на рисунку 9.18 запит відфільтровує записи, для яких поле "Назва" починається з літер «К» або «Д». На практиці запит переважно складається із комбінації одночасно двох логічних операцій, як **AND (И)** так і **OR (ИЛИ)** (так, як це показано на рис. 3.3).

Після створення запиту його необхідно зберегти. Для цього потрібно в пункті меню **File (Файл)** вибрати команду **Save (Сохранить)**. Можна також натиснути комбінацією клавіш **Ctrl+S** або вибрати піктограму . В будь-якому випадку відкриється вікно із єдиним полем, де слід вказати назву створюваного запиту. Якщо запит не збережено, то при закриті вікна конструктора запитів виводиться про це попередження і пропонується зберегти його.

Зберігши запис, можна безпосередньо з вікна конструктора завантажити його на виконання, натиснувши піктограму . Можна також закрити вікно конструктора і завантажити запис двічі натиснувши ліву клавішу миші на його назві у розділі **Queries (Запрсы)**, головного вікна БД (рис. 2.1).

В будь-якому випадку відкривається вікно з **динамічним набором даних**, які отримані в результаті виконання запиту. Динамічний набір зовні схожий на звичайну таблицю БД (наприклад, як це показано на рис. 3.4). При роботі з динамічним набором можна змінювати порядок чергування полів, встановлювати нові поля і знищувати існуючі. Для виконання цих операцій необхідно спочатку виділити відповідні поля, натиснувши ліву клавішу миші на їх заголовку (при натиснутому **Ctrl**). Переміщення відмічених полів виконується за технологією Drag-and-Drop. Для знищення запису в динамічному наборі даних потрібно його виділити і в пункті меню **Edit (Правка)** вибрати команду **Delete Records (Удалить запись)**.

Примітка

Нагадуємо, що всі запити, які користувач створює в Access візуально (за допомогою конструктора запитів та іншими способами), автоматично транслюються в текст спеціальних програм, написаних мовою баз даних SQL. Ці програми завантажуються при виконанні запитів. Для того, щоб побачити текст програми запитів мовою SQL, потрібно в пункті меню View (Вид) вибрати команду SQL View (Режим SQL).

Створення запитів на створення таблиці (Make-Table Query)

Крім вищеописаних **запитів на вибірку (Selection Query)** в Access можна створювати інші типи запитів, які ще називають **виконуючими запитами (Action Query)**. Якщо запити на вибірку відфільтровують дані і відображають їх в динамічному наборі, то виконуючі запити виконують певні дії над даними таблиці або іншого запиту (наприклад, на основі цих даних можна створити нову таблицю БД).

Виконуючі запити, як і інші типи запитів, відображаються в закладці **Queries (Запросы)** головного вікна БД. Але їх можна відрізнити від інших запитів за наявністю знаку оклику справа від піктограми.

Одним із різновидів виконуючих запитів є запити на створення таблиці. Вони використовуються для створення нової таблиці на основі існуючої (або декількох існуючих). Як правило, такий запит виконують не для того, щоб створити точну копію існуючої таблиці, а щоб нова таблиця мала частину вибраних (відфільтрованих) даних із попередньої, або містила поля, що належать декільком існуючим таблицям.

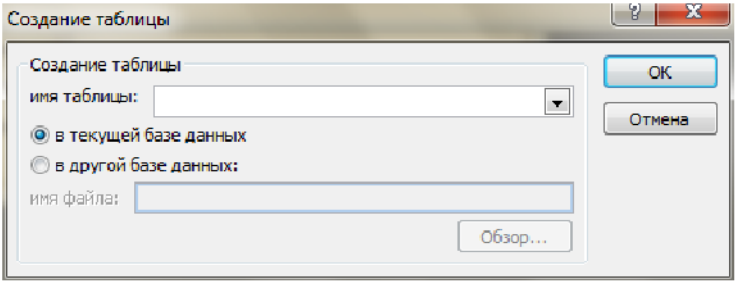


Рисунок 3.6 Вікно запиту на створення таблиці

Для створення запиту на створення таблиці потрібно завантажити конструктор запитів, помістивши в нього таблиці та запити на вибірку (див. попередній тип запиту), на основі яких ствоюється нова таблиця. Потім в пункті меню **Query (Запрос)** вибрати команду **Make-Table Query... (Создание таблицы)**. При цьому з'являється однойменне діалогове вікно (рис. 3.6), де в полі **Table Name (Имя таблицы)** потрібно вказати назву створюваної таблиці. Крім цього в даному вікні є дві опції-перемикачі:

- **Current Database (в текущей базе данных)** — помістити створену таблицю в активну БД;
- **Another Database (в другой базе данных)** — помістити створену таблицю в іншу БД. При цьому в полі File Name (Имя) потрібно вказати ім'я її файлу.

В результаті вибору кнопки OK відбувається повернення до конструктора запитів, де потрібно вказати всі необхідні параметри для створюваної таблиці подібно до створення запиту на вибірку, що описано попередньо. Після створення запит потрібно зберегти та завантажити на виконання одним із вищеперерахованих способів (див. попередній тип запиту).

Створення запитів на додавання записів (Append Query)

Запити на додавання записів використовуються для додавання даних однієї таблиці в кінець іншої. При цьому можна як копіювати з таблиці в таблицю активної БД, так і між таблицями різних БД. Таблиці не обов'язково повинні мати ідентичну структуру, але типи полів цих таблиць повинні бути однаковими.

Для створення запиту на додавання записів спочатку потрібно відкрити конструктор запитів (див. запити на вибірку), вказавши у вікні **Show Table (Добавление таблицы)** таблицю або запит на вибірку з яких будуть додаватись дані в іншу таблицю (див. попередній матеріал). Після цього в пункті меню **Query (Запрос)** вибрати команду **Append Query... (Добавление...)**. В результаті чого з'явиться вікно створення запиту на додавання записів, яке повністю аналогічне до вікна представленого на рисунку 3.6, але в полі **Table Name (Имя таблицы)** потрібно вказати назву таблиці, до якої потрібно доповнити дані.

Після натискування **OK** відбувається повернення до конструктора запитів, в якому з'являється додатковий рядок **Append To (Добавить к)**. В комірках цього рядка потрібно вибирати поля, до яких будуть дописуватись дані з полів вказаних у відповідних комірках рядка **Field (Поле)**. Після цього запит потрібно зберегти та завантажити на виконання одним із перерахованих в попередньому матеріалі способів.

Створення запитів на обновлення (Update Query)

Запити на оновлення є виконуючими запитами, які змінюють вміст певних полів таблиці, в залежності від вказаної умови. Запити на обновлення дуже корисні у випадку, коли потрібно внести глобальні зміни в певний набір даних.

Для створення запитів на обновлення потрібно відкрити конструктор запитів (див. запити на вибірку), вибравши у вікні **Show Table (Добавление таблицы)** таблиці, що будуть брати участь в запиті. Причому, необхідно вибирати не лише таблиці, які містять поля, які потрібно змінити але й таблиці із полями, що використовуються у формулах для обчислення нового значення змінюваного поля (якщо такі використовуються).

Після закриття вікна **Show Table (Добавление таблицы)** в пункті меню **Query (Запрос)** слід вибрати команду **Update Query (Обновление)**. В заголовку вікна конструктора запитів з'явиться повідомлення **Update Query (Запрос на обновление)**, а в самому бланку конструктора з'явиться рядок обновлення. Потім необхідно помістити курсор у відповідну комірку рядка **Update To (Условие отбора)** конструктора запитів і вказати формулу, за якою будуть змінюватись дані (наприклад, **[Ціна]*1,1** - значення в полі "Ціна" помножити на 1,1 (збільшити на 10%)).

Процес заповнення всіх інших параметрів запиту повністю аналогічний до процесу створення запитів на вибірку, що описаний попередньо.

Примітка

При використанні запитів на обновлення обов'язково потрібно створювати резервну копію таблиці, оскільки ці запити вносять безповоротні зміни в таблиці. Тому при помилковому завантаженні запиту або неправильному заданні його параметрів можна отримати неправильні дані таблиці

Створення запитів на знищення (Delete Query)

Запити на знищення знищують в таблиці всі записи, які відповідають певній умові. Досить часто такі запити використовують для знищення із зв'язаних таблиць записів, що не мають зв'язків в іншій таблиці.

Для створення запиту на знищення слід завантажити конструктор запитів (див. запити на вибірку), вибравши у вікні **Show Table (Добавление таблицы)** таблицю, в якій потрібно знищити записи. Після цього, необхідно в пункті меню **Query (Запрос)** вибрати команду **Delete Query (Удаление)** і у вікні конструктора вказати критерій для знищення записів. При цьому в комірках рядка **Delete (Удаление)**, що з'являється в конструкторі запитів для даного типу виконуючого запиту, автоматично буде вказано значення **Where (Условие)**.

Якщо потрібно створити запит на знищення незв'язаних записів у зв'язаних таблицях, потрібно завантажити конструктор запитів, вказавши в ньому відповідні таблиці, і вибрати команду **Delete Query (Удаление)**. Тоді для таблиці, з якої необхідно знищити записи в рядку **Field (Поле)** конструктора, потрібно вибрати символ "*" (можна перетягти мишею із списку полів). В результаті чого в комірці **Delete (Удаление)** з'явиться значення **From (Из)**. Щоб вказати умову відбору знищуваних записів, потрібно у наступних комірках рядка **Field (Поле)** вибрати поля, для яких вона задається, а у відповідних комірках критерію вказати умову.

Примітка

Якщо таблиці зв'язані відношенням "один-до-багатьох", то не можна знищити записи із таблиці "один", коли в таблиці "багато" присутні відповідні їм записи.Спочатку потрібно знищити записи в таблиці "багато", а лише тоді - "один". Для того, щоб спростити цей процес, Access дозволяє при визначенні зв'язків між таблицями встановити опцію Cascade Delete Related Records (каскадное удаление связанных записей) (рис. 2.11). Оскільки результат роботи запиту на знищення записів не можливо відмінити, то перед його виконанням потрібно створити резервну копію таблиці.

Створення перехресних запитів (Crosstab Query)

Перехресні запити використовують для відображення результатів певних статистичних розрахунків (суми, кількості записів, середніх значень, найбільшого та найменшого значення), що виконувались над даними з певного поля таблиці. Ці дані групуються по двох наборах, один з яких розміщений в лівій колонці, а інший у верхньому рядку двомірної таблиці. На перетині відповідних рядків та колонок знаходяться результати статистичних розрахунків. Отже з допомогою перехресних запитів можна створювати динамічні перехресні таблиці, які дуже нагадують зведені таблиці Microsoft Excel.

Перехресні запити варто використовувати для полів таблиць, що мають підтаблиці, в яких значенню запису одного поля відповідає декілька значень іншого.

В Access перехресні запити можна створити двома способами: за допомогою майстра перехресних запитів та конструктора запитів.

Для створення запиту за допомогою майтра перехресного запиту потрібно у головному вікні БД перейти в розділ **Queries (Запросы)** і вибрати командну кнопку **New (Создать)**. У вікні, що відкриється (рис. 3.1), необхідно встановити курсор на **Crosstab Query Wizard (Перекрестный запрос)** і натиснути кнопку **OK**.

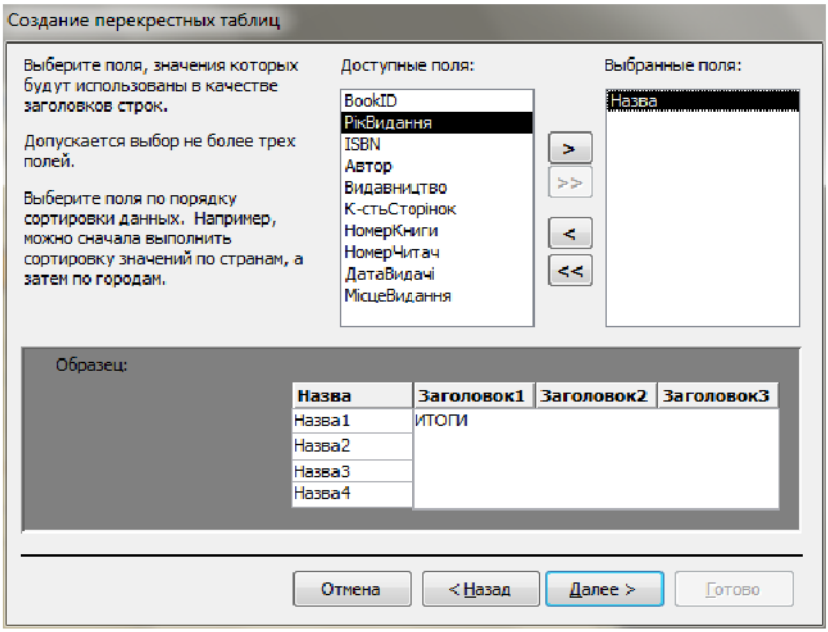


Рисунок 3.7 Друге вікно майстра створення перехресних запитів

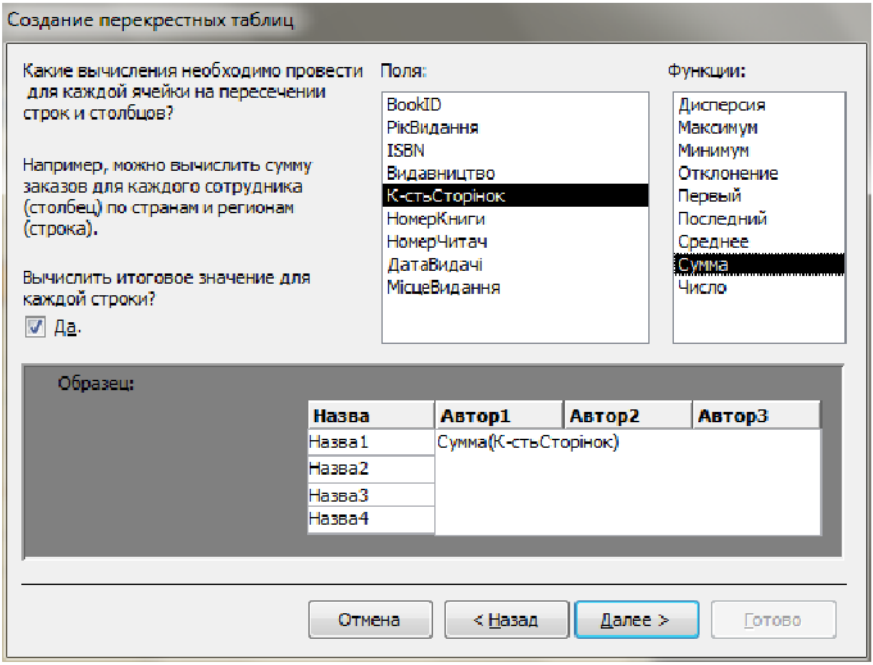


Рисунок 3.8 Четверте вікно майстра створення перехресних запитів

Відкривається вікно майстра створення запитів, де потрібно вибрати таблицю (опція **Tables (Таблицы)**) або запит на вибірку (**Queries (Запросы)**), на основі яких буде створюватись перехресний запит.

Після натискування **Next (Далее)** відкривається наступне вікно (рис. 3.7), де за допомогою кнопки: > потрібно вибрати поля (Access в перехресній таблиці дозволяє використати декілька полів, які вибираються із випадаючого списку), що будуть використовуватись як назви рядків перехресної таблиці (кнопка >> добавляє всі поля).

У третьому вікні необхідно виділити поле, що буде використовуватись як назва колонок перехресної таблиці.

На четвертому кроці роботи майстра створення перехресних таблиць відкривається вікно (рис. 3.8), в якому в списку **Fields (Поля)** потрібно вибрати поле, для якого буде проводитись статистичне обчислення, а у списку **Functions (Функции)** — назву відповідної статистичної функції. Якщо відмітити опцію **Yes, include row sums (Вычислить итоговое значение для каждой строки?)**, то в останній колонці таблиці буде виводитись сумарне значення для кожного рядка.

Останнє вікно майстра створення перехресного запиту, в якому необхідно вказати назву створюваного запиту та відмітити одну з двох опцій: **View the query (Посмотреть результаты запроса)** — відкрити запит для перегляду та **Modify the design (Изменить структуру запроса)** — завантажити конструктор запитів для редагування його параметрів.

Майстер створення перехресних запитів зручно використовувати для створення не дуже складних запитів. Для більш складніших перехресних запитів необхідно скористатись конструктором запитів.

Щоб створити перехресний запит за допомогою конструктора запитів, потрібно одним із вищеописаних способів завантажити його вікно і вибрати таблицю (таблиці) або запит на вибірку, по яких буде створюватись перехресний запит. Тоді в пункті меню **Query (Запрос)** вибрати команду **Crosstab Query (Перекрестный запрос)**. Після цього одним із вищеописаних способів добавити у вікно конструктора всі поля, що ввійдуть в запит — назви колонок та рядків перехресної таблиці та поле, по якому будуть проводитись статистичні обчислення. У випадаючих списках відповідних комірок рядка **Crosstab (Перекрестная таблица)** потрібно вибрати призначення полів: **Row Heading (Заголовки строк)** — назви рядків перехресної таблиці, **Column Heading (Заголовки столбцов)** — назви її колонок та **Value (Значение)** — поля, над якими проводяться статистичні обчислення. В рядку **Total (Групповая операция)** для полів, що будуть заголовками рядків та колонок потрібно залишити значення за замовчуванням — **Group By (Группировка)**, а для поля, по якому будуть проводитись обчислення — назву відповідної функції. Призначення параметрів інших полів описані вище (див. запити на вибірку).

Робота із перехресними таблицями майже нічим не відрізняється від аналогічних операцій над зведеними таблицями Microsoft Excel.

Створення запитів для пошуку полів з ідентичним вмістом (Find Duplicates Query)

Інколи виникає потреба вивести список записів, в яких деякі поля містять однакові дані. Для цього у вікні створення нового запиту потрібно вибрати пункт Find Dublicates Query Wizard (Повторяющиеся записи). Відкривається вікно майстра створення запиту даного типу, в якому потрібно вибрати таблицю (опція Tables(Таблицы)) або запит на вибірку (опція Queries (Запросы)), для яких створюється даний запит.

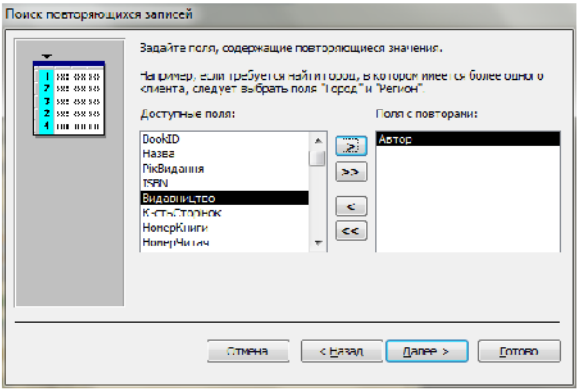


Рисунок 3.9 Друге вікно майстра створення пошуку полів з ідентичним вмістом

На другому етапі роботи майстра (рис. 3.9) необхідно за допомогою кнопки > вибрати поля (>> — для вибору всіх полів), в яких буде здійснюватись пошук однакових даних.

У вікні наступного етапу роботи майстра таким же способом, як і в попередньому вікні, потрібно вибрати інші поля, що відображатимуться при перегляді запиту.

В останньому вікні майстра необхідно ввести назву створеного запиту та вибрати одну із опцій: **View the results (Посмотреть результат запроса)** — відкрити запит для перегляду та **Modify the design (Изменить структуру запроса)** — завантажити конструктор запитів для редагування його параметрів.

Запити на пошук записів, що повторюються, можна створювати і в конструкторі запитів. При цьому в рядку встановлення умови **Criteria (Условие отбора)** колонки, по якій шукаємо однаковий вміст поля потрібно ввести складний SQL-вираз із застосуванням оператора **in(...)** та SQL-конструкції **SELECT**.

Створення запитів для пошуку не зв'язаних записів в іншій таблиці (Find Unmatched Query)

При зв'язуванні двох різних таблиць може виникнути ситуація, коли у полі, по якому відбувається зв'язування, дані однієї таблиці не мають відповідності в іншій. В результаті чого деякі записи залишаються не зв'язаними. Для пошуку таких записів використовується даний тип запиту. Він створюється шляхом вибору у вікні створення нового запиту (рис. 3.1) пункту **Find Unmatched Query Wizard (Записи без подчиненных)**. Відкривається вікно майстра створення запиту даного типу, в якому, подібно до попереднього типу потрібно вибрати таблицю або запит на вибірку, для якого створюється новий.

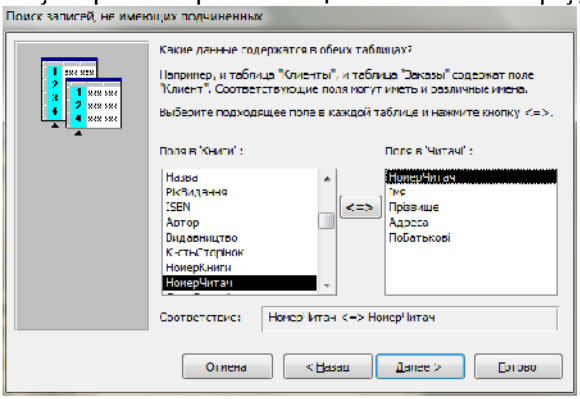


Рисунок 3.10 Третє вікно майстра створення пошуку полів з ідентичним вмістом

У другому вікні майстра необхідно вибрати зв'язану таблицю, з якою відбувається порівняння та знову натиснути кнопку **Next (Далее)**.

Відкривається третє вікно роботи майстра (рис. 3.10), де в списку зліва потрібно вибрати назву поля першої таблиці, а в списку справа — другої для перевірки їх відповідності. Після цього натиснути командну кнопку <=>.

На наступному етапі відкривається вікно, подібне до зображеного на рисунку 3.9, де за допомогою кнопки потрібно вибрати інші поля, що будуть відображатись у запиті.

Останнє вікно майстра повністю ідентичне до відповідного вікна створення запиту для пошуку полів з ідентичним вмістом.

Даний тип запиту також можна створювати за допомогою конструктора запитів. Для цього в рядку задання умови для відповідного поля слід ввести вираз **Is Null**.

Створення запитів на об'єднання записів (Union Query)

Даний тип запиту дозволяє об'єднати в одному результуючому наборі результати декількох запитів, таблиць та інструкцій **SELECT** (встановлює перелік полів таблиці для вибору у запиті). Запити на об'єднання в Access можна створити лише одним способом: шляхом безпосереднього введення інструкцій SQL у вікні **SQL View**.

Для його завантаження потрібно спочатку відкрити вікно конструктора запитів (див. запити на вибірку), а тоді в пункті меню **Query (Запрос)** вибрати підпункт **SQL Specific (Запрс SQL)**, а в ньому команду **Union (Объединение)**. В результаті чого відкривається порожнє вікно **SQL View (Режим SQL)**, в яке потрібно ввести відповідну інструкцію SQL. Синтаксис запиту на об'єднання наступний:

[Table] запит_1 UNION [ALL] [TABLE] запит_2 UNION [ALL] [TABLE] запит_3 ...

де **запит_1, запит_2, запит_3** — інструкція **SELECT**, назва збереженого запиту або таблиці, перед якою стоїть зарезервоване слово **[TABLE]**.

В кожному аргументі **запит_1, запит_2, запит_3** допускається використання параметра **GROUP BY** або **HAVING** для групування вихідних даних. Параметр **ALL** в запиті дозволяє включити в нього всі записи. Крім цього такі записи виконуються швидше.

Створення керуючих запитів (Data Definition Query)

Керуючі запити також відносяться до запитів, що створюються тільки в режимі введення SQL-інструкцій у вікні **SQL View**. Вони дозволяють створювати, змінювати та знищувати таблиці, створювати індекси та ін.

Для створення керуючого запиту потрібно завантажити вікно конструктора запитів (див.: запити на вибірку), а тоді в пункті меню **Query (Запрос)** вибрати підпункт **SQL Specific (Запрс SQL)**, а в ньому команду **Data Definition (Управление)**. В результаті чого відкривається порожнє вікно **SQL View (Режим SQL)**, в яке потрібно ввести відповідну інструкцію SQL.

Інструкції, які при цьому використовуються, відносяться до підмножини SQL, що називають DDL — мова визначення даних. При цьому можна використати наступні інструкції:

- **CREATE TABLE** — створення таблиці;
- **CONSTRAINT** — створення унікального індекса, в томі числі за первинним ключем, а також використовується для встановлення відношень між таблицями;
- **ALTER TABLE** — змінити структуру таблиці, що створена за допомогою інструкції **CREATE TABLE**;
- **DROP** — знищити таблицю із БД або індекси в таблиці;
- **CREATE INDEX** — створення індекса для поля або групи полів.

Примітка

Запити даного типу переважно використовуються в програмах мовою Visual Basic Application тоді, коли потрібно виконати відповідну операцію програмно. В іншому випадку для цього набагато краще використати засоби користувацького інтерфейсу Microsoft Access. В межах даного посібника ми не описуємо детально синтаксис всіх інструкцій SQL. Для того, щоб ознайомитись із ним детальніше автори посібника рекомендують звернутись до довідкового матеріалу з мови Microsoft Jet SQL.

Створення запитів до сервера (Pass-Through Query)

Даний тип запитів використовується для вибору даних із сервера. На відміну від попередніх типів запитів вони не обробляються безпосередньо процесором Jet SQL, а передаються на сервер БД, до якого виконується звертання, наприклад Microsoft SQL Server. Запити до сервера створюються подібно до керуючих запитів, тобто із використанням інструкцій SQL у вікні **SQL View**.

Для цього потрібно завантажити вікно конструктора запитів (див. запити на вибірку), а тоді в пункті меню **Query (Запрос)** вибрати підпункт **SQL Specific (Запрс SQL)**, а в ньому команду **Pass-Through (К серверу)**. В результаті чого відкривається порожнє вікно **SQL View (Режим SQL)**, в яке потрібно ввести відповідну інструкцію SQL.

ТЕМА 4 РОБОТА З ФОРМАМИ В ACCESS

4.1 Автоматичне створення форм. Майстри створення форм

В попередніх розділах даної частини посібника описано порядок роботи з БД в режимі таблиці. Але в Access крім таблиць для роботи з даним в БД часто використовують режим форм.

Форми — це представлення даних таблиць, або результатів роботи запитів у форматованому, спеціально оформленому, вигляді (картотека, діаграми, зведені таблиці і т.д.). Зверніть увагу, що форми Access мають значно ширші можливості, ніж просте відображення набору полів одного запису. В форми можна поміщати графічні зображення, в тому числі і у вигляді об'єктів OLE. В формах можна виводити дані з кількох зв'язаних таблиць. Можна також створювати форми, які забезпечують різне подання одних і тих же даних.

Використання форм для введення та редагування даних має ряд переваг в порівнянні з використанням таблиць. Так, при перегляді даних в режимі таблиці часто виявляється, що важко побачити всі поля окремого запису (а інколи і неможливо). Крім цього, форми дозволяють вибрати зручне розміщення полів і забезпечити перегляд графічних об'єктів, які зберігаються в полях OLE-об'єктів.

Access дозволяє швидко створювати форми, використовуючи **майстер форм**. Можна також створити форму на основі існуючої: відкривши порожню та базову і перетягнувши за технологією Drag-and-Drop потрібні об'єкти. Інший швидкий спосіб створення форм — це **автоформи**, тобто автоматичне створення форми на основі останньої створеної форми такого типу.

Крім цього, в Access існує і універсальний спосіб створення форм за допомогою **конструктора форм**. Цей спосіб дозволяє детально налаштувати всі елементи форми, але є найбільш трудомістким у реалізації.

Для створення форми потрібно у головному вікні БД (рис. 2.1) перейти до закладки **Forms (Формы)** і натиснути командну кнопку **New (Создать)**. Відкривається діалогове вікно **New Form (Новая форма)** (рис. 4.1). В даному вікні спочатку у випадаючому списку потрібно вибрати назву таблиці або запиту, по якому будуємо форму, а тоді один з режимів створення форми:

- **Design View (Конструктор)** — створення форми за допомогою конструктора форм;
- **Form Wizard (Мастер форм)** — створення форми за допомогою майстра форм;
- **AutoForm: Columnar (Автоформа: в столбец)** — створення форми з використанням можливостей автоформатування.

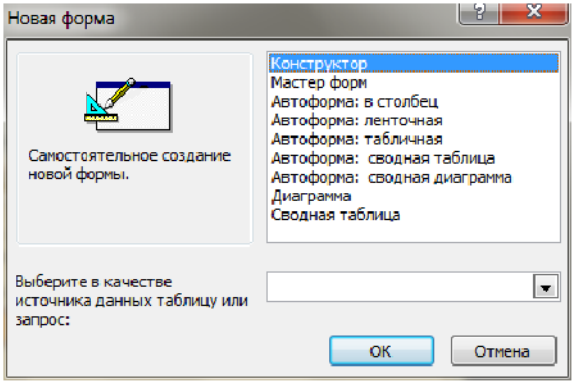


Рисунок 4.1 Вікно створення форм

При цьому поля розміщуються в колонку;

- **AutoForm: Tabular (Автоформа: ленточная)** — створення форми з використанням можливостей автоформатування. При цьому поля розміщуються в рядок;
- **AutoForm: Datasheet (Автоформа: табличная)** — створення форми з використанням можливостей автоформатування. При цьому поля розміщуються у вигляді звичайної таблиці БД;
- **AutoForm: PivotTable (Автоформа: сводная таблица)** — створення форми у вигляді зведеної таблиці Microsoft Excel;
- **AutoForm: PivotChart (Автоформа: сводная диаграмма)** — створення форми у вигляді зведеної діаграми Microsoft Excel;
- **Chart Wizard (Диаграмма)** — створення форми, яка відображає числові дані у вигляді діаграми, що будується за допомогою майстра діаграм;
- **Pivot Table Wizard (Сводная таблица)** — створення форми, що містить зведену таблицю Excel.

Після вибору режиму створення форми, у випадаючому списку в нижній частині вікна потрібно вибрати назву таблиці БД, до якої створюється форма і натиснути кнопку **OK**.

Найшвидший спосіб створення форм, це використання режимів **AutoForm (Автоформа)**. При цьому форма створюється на основі останньої створеної форми такого типу, а якщо вона ще не створювалась, то на базі стандартних бланків. Як ми вже говорили, у вікні **New Form (Новая форма)**, можна вибрати один з п'яти варіантів шаблону автоформи (див. попередній матеріал).

Створення форми за допомогою майстра форми

При виборі у вікні Создать **New Form (Новая форма)** (рис. 4.1) режиму створення форми **Form Wizard (Мастер форм)** завантажується підпрограма майстра створення форми. У першому вікні майстра (рис. 4.2), потрібно вибрати поля, які повинні увійти в форму. При цьому у списку **Available Fields: (Доступные поля:)**, потрібно вибирати поле, а тоді кнопкою > перемістити його у список **Selected Fields: (Выбранные поля:)**. Цю ж операцію повторити для інших полів (кнопка >> — для вибору одночасно всіх полів таблиці або запиту). Якщо форма повинна містити поля одночасно з декількох зв'язаних таблиць (запитів), то після вибору полів із першої, у випадаючому списку **Tables/ Queries (Таблицы и запросы)** необхідно вказати наступну таблицю (запит) та повторити операцію вибору її полів. Встановивши список всіх полів форми, потрібно натиснути **Next (Далее)**

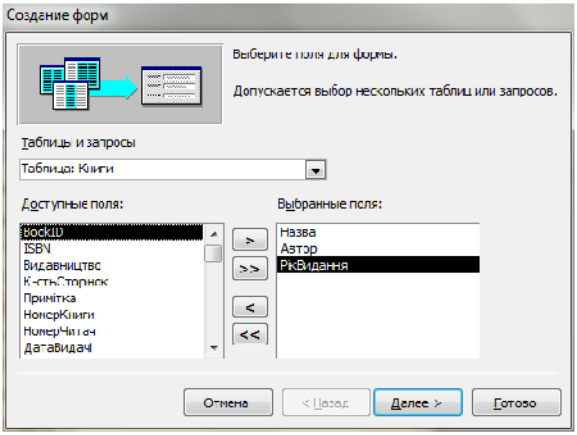


Рисунок 4.2 Перше діалогове вікно майстра форм

для переходу до наступного вікна роботи майстра. У випадку, якщо таблиці не зв'язані, після натискування **Next (Далее)** з'являється додаткове вікно, де пропонується зв'язати ці таблиці і розпочати створення форми спочатку.

Якщо для створення форми використано поля з декількох зв'язаних таблиць, то на другому етапі відкривається додаткове вікно, де потрібно встановити тип представлення даних: **Single form (Одиночная форма)** — дані обидвох таблиць відображаються в одному вікні форми, або **Linked forms (Связанные формы)** — у вікні форми відображаються дані першої у списку таблиці (запиту), а для представлення даних іншої — потрібно натиснути додаткову кнопку у формі. При цьому дані іншої таблиці (запиту) будуть відображені в іншому вікні зв'язаної форми. Замість першої опції на даному етапі може відображатись також опція **Form with subforms (Подчиненные формы)** — в одному вікні форми відобразити форму разом із із даними залежної підтаблиці (підформа). Після вибору типу представлення даних потрібно натиснути командну кнопку **Next (Далее)** для переходу до третього вікна майстра створення форм (рис. 4.3).

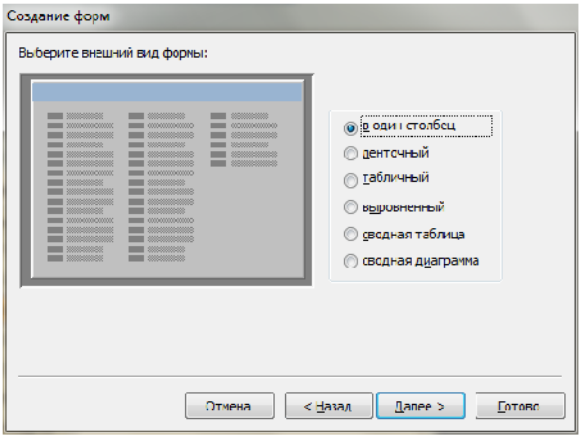


Рисунок 4.3 Третє діалогове вікно майстра форм

У випадку, якщо на першому етапі роботи майстра створення форм (рис. 4.2) вибрано дані із однієї таблиці, то автоматично відбувається перехід, до третього діалогового вікна майстра (рис. 4.3), в якому потрібно вибрати тип форми і натиснути командну кнопку **Next (Далее)**. При цьому можна вибрати наступні типи форм:

- **Columnar (в один столбец)** — всі поля запису розміщені в одну колонку і вирівняні по лівому краю;
- **Tabular (ленточный)** — всі поля запису розміщені в рядок і відділені між собою інтервалами;
- **Datasheet (табличный)** — форма у вигляді таблиці, що схожа на таблицю даних, але без виділення відступом;
- **Justified (выровненный)** — поля форми розміщуються по порядку, а їх ширина підганяється до ширини полів таблиці з фіксованим відступом від правого краю;
- **PivotTable (сводная таблица)** — форма у вигляді зведеної таблиці Micrososft Excel;
- **PivotCahrt (сводная диаграмма)** — форма у вигляді зведеної діаграми Micrososft Excel.

На четвертому етапі роботи майстра форм відкривається діалогове вікно (рис. 4.4), де потрібно вибрати стиль фону форми. В Access є десять стилів фону для форм, які створюються за допомогою майстра форм. Хочемо зауважити, що хоча деякі зі стилів фону форм є досить привабливими, не потрібно зловживати їх використанням, оскільки додавання до форми кольорових малюнків сповільнює роботу над БД. Це перш за все стосується малопотужних ПК з обмеженими ресурсами. Після

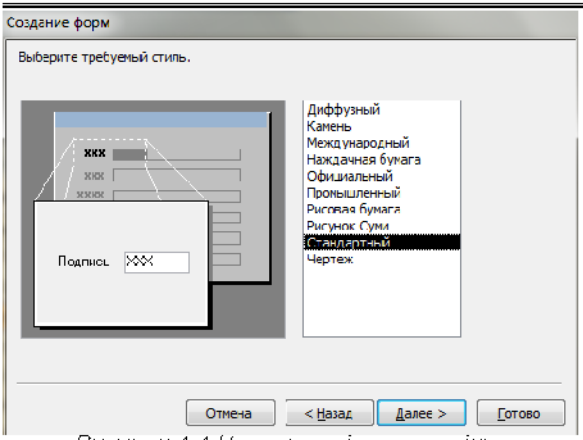


Рисунок 4.4 Четверте діалогове вікно майстра форм

того, як буде вибрано стиль форми, потрібно натиснути командну кнопку **Next (Далее)** для переходу до наступного етапу роботи майстра форм.

В останньому вікні роботи майстра форм потрібно вказати назву (заголовок) форми. Якщо форма заснована на кількох таблицях і на другому етапі вказано створення зв'язаних форм, то потрібно також вказати назви другорядних форм. Крім цього, в даному вікні можна вибрати одну з двох опцій перемикачів:

- **Open the form to view or enter information (Открыть форму для просмотра и ввода данных)** — після завершення роботи майстра форм відкрити вікно форми із активним першим записом;
- **Modify the form's design (Изменить макет формы)** — після завершення роботи майстра форм, відкрити вікно конструктора форм.

Роботу майстра потрібно завершити натискуванням кнопки **Finish (Готово)**.

Примітка

Якщо на другому етапі роботи майстра створення форм вибрано зв'язані форми, то замість опції Open the form to view or enter information (Открыть форму для просмотра и ввода данных) з'являється опція Open the main form to view or enter information (Открыть главную форму для просмотра и ввода данных) – відкрити головну форму із активним першим записом.

Створення форми у вигляді діаграми

Одним із специфічних видів форми в Access є форма у вигляді діаграми. Вона створюється за допомогою майстра побудови діаграми, що завантажується командою **Chart Wizard (Диаграмма)**, що знаходиться у вікні **New Form (Новая форма)** (рис. 4.1). Крім цього, у даному вікні потрібно вибрати назву таблиці або запиту на вибірку, по якому будується діаграма. Після вибору кнопки OK з'являється перше вікно майстра, в якому потрібно за допомогою кнопки > вибрати поля (>> — для вибору всіх полів), по яких буде будуватись діаграма. Дане вікно подібне, до першого вікна майстра створення форм (рис. 4.2), але в ньому немає випадаючого списку **Tables/Queries (Таблицы и запросы)**, отже діаграму можна будувати лише по полях однієї таблиці, або віртуальної таблиці, створеної за допомогою одного запиту.

На другому етапі роботи майстра потрібно вибрати тип діаграми і натиснути **Next (Далее)**. У наступному вікні (рис. 4.5) необхідно вказати назви осей діаграми. Для цього потрібно перетягнути прямокутну ділянку із назвою поля з правої частини вікна на спеціальну ділянку (білий прямокутник) осей. Якщо двічі натиснути ліву клавішу миші на назві осі, що містить числові дані (на рисунку 4.5 вісь Z), то відкривається вікно, в якому можна вибрати статистичну функцію обчислення результуючих даних, по яких буде будуватись

діаграма. У випадку вибору параметру **None (Нет)** діаграма будується власне по числових значеннях у даному полі.

В останньому вікні майстра створення діаграм потрібно ввести назву форми-діаграми та вказати необхідність встановлення до неї легенди. Нагадаємо, що в легенді вказуються типи заливки, кольори та візерунки, які використовуються для позначення даних на діаграмі. Крім цього в даному вікні можна вибрати одну з опцій: **Open the form with the chart displayed on it (Открытие формы с диаграмой)** — відкрити форму-діаграму для перегляду або **Modify the design of the form or the chart (Изменение структуры формы или диаграммы)** — відкрити вікно конструктора для зміни параметрів відображення діаграми.

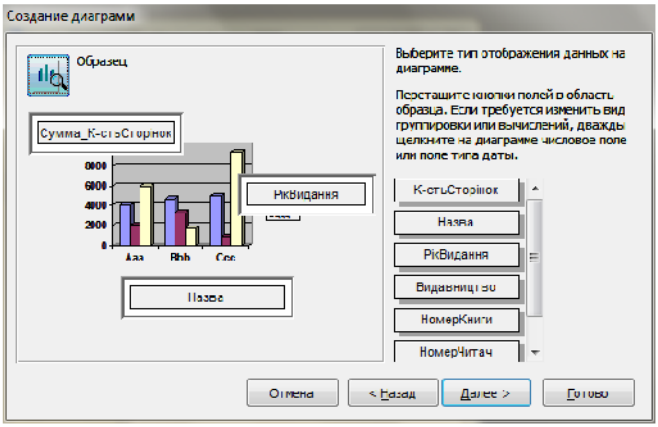


Рисунок 4.5 Третє вікно майстра діаграм

Створення форми у вигляді зведеної таблиці

Ще одним типом форм в Access є зведені таблиці. Для створення такої форми потрібно у вікні **New Form (Новая форма)** (рис. 4.1) вибрати пункт **PivotTable Wizard (Сводная таблица)** та у випадаючому списку назву таблиці або запиту на вибірку, по якому будується зведена таблиця. Відкривається перше вікно роботи майстра створення зведених таблиць. Воно містить лише інформацію про призначення даної програми.

Після натискування **Next (Далее)** відкривається друге вікно майстра, яке дуже подібне на перше вікно майстра створення форм (рис. 4.2). в якому за допомогою кнопки > потрібно вказати поля таблиці (запиту), що повинні міститись у формі і натиснути кнопку **Finish (Финиш)**. Якщо потрібно використати поля із декількох таблиць або запитів, то після встановлення полів першої з них потрібно у випадаючому списку **Tables/Queries (Таблицы и запросы)** вказати наступну таблицю (запит) та повторити операцію вибору полів.

По завершенні роботи майтра відкривається вікно із бланком зведеної таблиці та панель із списком вибраних полів (**PivotTable Field List**). Щоб добавити поле із списку у зведену таблицю, достатньо перетягнути його у відповідне місце таблиці за технологією Drag and Drop. Операції налаштування та форматування зведених таблиць в Access дуже подібні до аналогічних операцій в Microsoft Excel (див. розділ 8.1 восьмої частини посібника).

4.2 Робота з конструктором форм

Як ми вже говорили, форми можна створювати за допомогою конструктора форм. В цьому режимі розробка форми відбувається вручну, шляхом відкриття порожньої форми і додавання до неї об'єктів (текст, текстові вікна, графіка, лінії, рамки та ін.).

При створенні форми можна зекономити час. Для цього потрібно, використовуючи майстер форм, створити чорновий варіант потрібної форми, а потім відкрити форму в режимі конструктора і зробити всі необхідні зміни.

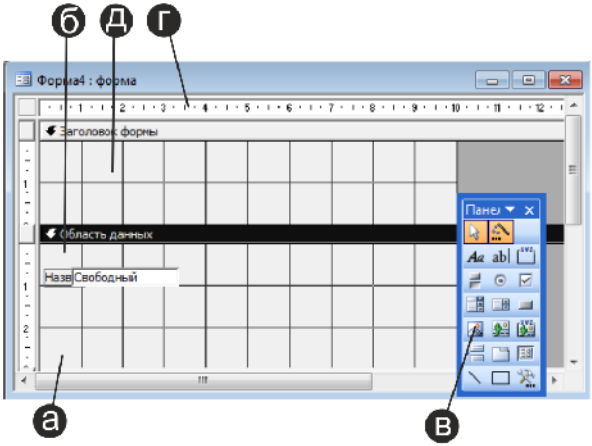


Рисунок 4.6 Вікно конструктора форм

Для створення форми за допомогою конструктора, потрібно у вікні створення нової форми (рис. 4.1) вибрати режим **Design View (Конструктор)**. Потім, у випадаючому списку вибрати назву таблиці, для якої створюється форма, і натиснути **OK**. При цьому відкривається вікно конструктора форм (рис. 9.31), в якому у вигляді прямокутної ділянки із сіткою (рис. 9.31.а) вказуються межі бланку створюваної форми. Сітка використовується для взаємного вирівнювання об'єктів, поміщених на форму. Бланк створюваної форми може складатись із п'яти розділів. Основний розділ **Details (Область данных)** (рис. 4.6.а) визначає робочу ділянку форми, в якій будуть відображатись її елементи (поля, командні кнопки і т.д.).

Вище цього розділу можуть відображатись, ще два розділи бланку створюваної форми це: **Form Header (Заголовок формы)** (рис. 4.6.д) — заголовок форми, тобто інформація, що буде відображатись на першій сторінці багатосторінкової форми, та **Page Header (Верхний колонтитул)** (рис. 4.6.е) — верхній колонтитул сторінки, тобто інформація, що буде повторюватись зверху кожної сторінки багатосторінкової форми.

Нижче розділу **Details (Область данных)** (рис. 4.6.а) також може відображатись ще два розділи: Page Footer (Нижний колонтитул) - нижній колонтитул сторінки, тобто інформація, що буде повторюватись внизу кожної сторінки багатосторінкової форми, та **Form Footer (Примечание формы)** — інформація, що буде відображатись в кінці останньої сторінки багатосторінкової форми.

Якщо ділянки заголовків (рис. 4.6.д) або колонтитулів (рис. 4.6.б) у вікні конструктора форм не відображаються, то в пункті меню **View (Вид)** потрібно вибрати команду **Form Header/Footer (Заголовок/примечание формы)** — відображення ділянки заголовка форми і ділянки в кінці останньої сторінки багатосторінкової форми, або **Page Header/Footer (Колонтитулы)** — ділянки верхнього і нижнього колонтитулів сторінки багатосторінкової форми.

Зліва та зверху від робочої ділянки вікна конструктора форм знаходяться розмірні лінійки (рис. 4.6.г), які дозволяють визначати розміри об'єктів на формі та відстані між ними. При роботі із конструктором форм з'являється панель інструментів із піктограмами об'єктів (рис. 4.6.в), що можуть розміщатись у формі. Якщо дана панель не виводиться, то потрібно в пункті меню **View (Вид)** вибрати команду **Toolbox (Панель элементов)**. Список інструментів цієї панелі та їх призначення представлений в таблиці 4.1.

Таблиця 4.1

Програми інструментів для розробки форм

Пікто-грама	Назва	Призначення
	Select Object (Выбор объектов)	вибирає, переміщує, змінює розміри об'єктів, і редагує їх
	Control Wizard (Мастера)	вмикає або вимикає майстри елементів керування
	Label (Надпись)	створює написи, заголовки, інструкції або описовий текст
	Text Box (Поле)	створює поля введення, які відображають вміст полів даних
	Option Group (Группа переключателей)	створює групу перемикачів, яка використовується для вибору одного значення із декількох можливих
	Toggle Button (Выключатель)	створює на формі вимикач, який використовується для вибору одного значення з декількох, або використовується як перемикач типу "Так"/"Ні"
	Option Button (Переключатель)	створює опції-перемикачі, які використовуються для вибору одного значення з декількох
	Check Box (Флажок)	створює звичайну опцію
	Combo Box (Поле со списком)	створює випадаючий список, в якому можна вибрати одне значення з декількох
	List Box (Список)	створює звичайний список
	Command Button (Кнопка)	створення командної кнопки
	Image (Рисунок)	створення малюнка. При цьому з'являється вікно Insert Picture (Вставка рисунка), де потрібно вибрати файл з малюнком
	Unbound Object Frame (Свободная рамка объекта)	створює фрейм (кадр з об'єктом, який можна вільно переміщати по документу і встановлювати обтікання основним текстом), який містить OLE-об'єкт
	Bound Object Frame (Присоединенная рамка объекта)	створює фрейм, що містить посилання на OLE-об'єкт
	Page Break (Разрыв страницы)	вставляє розрив сторінки
	Tab Control (Набор вкладок)	створює форму, яка містить кілька закладок
	Subform/Subreport (Подчиненная форма/отчет)	додає залежну зв'язану підформу або звіт
	Line (Линия)	створює лінію

	Rectangle (Прямоугольник)	створює прямокутник або квадрат
	More Control (Дополнительные элементы)	відкриває додаткове вікно для створення нестандартних елементів керування (наприклад, елементів керування Visual Basic)

Для того, щоб добавити елемент керування до бланку форми потрібно вибрати його піктограму в панелі інструментів, а тоді курсором миші обвести прямокутну ділянку, в якій повинен розміщуватись цей елемент.

Всі незмінні написи (назви полів, назви кнопок та інші повідомлення) створюються за допомогою інструмента **Label (Надпись)**. При цьому потрібно ввести прямокутну зону напису, а тоді помістити в неї курсор і ввести текст напису.

Для того, щоб помістити у вікно форми поле таблиці, потрібно встановити курсор на його ім'я у спеціальному вікні із списком полів і перетягнути в ділянку бланку створюваної форми. Якщо вікно із списком полів не виводиться, то в пункті меню **View (Вид)** необхідно вибрати команду **Field List (Список полей)**. Щоб помістити поле у вікно створюваної форми, можна також скористатись інструментом **Text Box (Поле)**. Вибравши цей інструмент, потрібно курсором миші утворити прямокутну ділянку і в контекстному меню об'єкта вибрати команду **Properties (Свойства)**. Відкривається вікно властивостей об'єкта, де у випадаючому списку **Control Source (Данные)** слід вибрати назву поля.

Інколи в поля, створені інструментом **Text Box (Поле)**, вводять формули, за якими обчислюються значення, що будуть виводитись в цих ділянках (обчислювані елементи керування). Наприклад, потрібно підрахувати і вивести податок, що становить 6% від величини, що вказана в полі "Ціна". Для того, щоб ввести в формулу обчислюваний елемент керування, потрібно спочатку ввести знак "=" ("дорівнює"), а тоді формулу, за якою буде вестись обчислення. Для вищеописаного прикладу обчислення податку потрібно ввести вираз "=Ціна*0,06".

Для створення у формах таких елементів керування, як командні кнопки, опції, перемикачі, вимикачі та ін., в Access використовуються спеціальні підпрограми майстрів. Для того, щоб вони автоматично завантажувались, коли об'єкт поміщається на форму потрібно в панелі інструментів (рис. 4.6.в) встановити піктограму **Control Wizard (Мастера)** у натиснутому вигляді.

При створенні командної кнопки, завантажується майстер, який на першому кроці відкриває вікно (рис. 4.7), що містить два списки: **Categories (Категории)** —

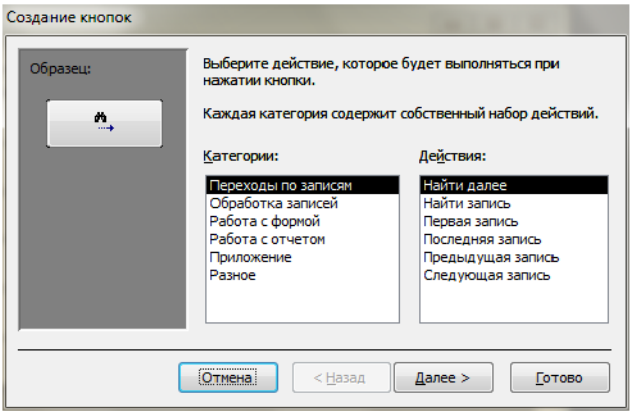


Рисунок 4.7 Перше вікно майстра встановлення на формі командної кнопки

типи (категорії) можливих команд (функцій) та дій і **Actions (Действия)** — назви цих команд. В списку категорій можна вибрати наступні типи:

- **Record Navigator (Переходы по записям)** — команди для переміщення по записах таблиці БД: **Find Next (Найти далее)** — знайти певну інформацію, **Find Record (Найти запись)** — знайти запис таблиці БД, **Go to First Record (Первая запись)** — перейти на перший, чи **Go to Last Record (Последняя запись)** — перейти на останній запис в таблиці БД, **Go to Next Record (Следующая запись)** — перейти на наступний запис, **Go to Previous Record (Предыдущая запись)** — перейти на попередній запис;
- **Record Operation (Обработка записей)** — команди для обробки записів БД: **Add New Record (Добавить запись)** — додати новий запис, **Delete Record (Удалить запись)** — знищити запис, **Duplicate Record (Дублировать запись)** — створити копію запису, **Print Record (Печать записи)** — роздрукувати запис на принтері, **Save Record (Сохранить запись)** — записати запис у файл, **Undo Record (Восстановить запись)** — відновити запис;
- **Form Operations (Работа с формой)** — команди для роботи із формами: **Apply Form Filter (Применить фильтр формы)** — добавити до форми фільтр, **Close Form (Закрыть форму)** — закрити вікно форми, **Edit Form Filter (Изменить фильтр формы)** — змінити умову (критерій відбору) для фільтру форми, **Open Form (Открыть форму)** — завантажити вікно зв'язаної форми, **Open Page (Открыть страницу)** — завантажити іншу сторінку форми, **Print a Form (Печать формы)** — роздрукувати вміст форми, **Print Current Form (Печать формы)** — роздрукувати вміст поточної підформи, **Refresh Form Data (Обновить данные формы)** — обновити дані в формі;
- **Report Operation (Работа с отчетом)** — команди для роботи зі звітами: **Mail Report (Отправить отчет по почте)** — відправити звіт електронною поштою, **Preview Report (Просмотр отчета)** — переглянути звіт, **Print Report (Печать отчета)** — роздрукувати звіт на принтері, **Send Report to File (Отправить отчет в файл)** — записати звіт у файл;
- **Application (Приложение)** — команди, що дозволяють завантажити інші прикладні програми. При цьому можна завантажити Excel, Word, Notepad та деякі допоміжні програми та утиліти для Access;
- **Miscellaneous (Разное)** — команди, що завантажують інші сервісні функції та програми: **AutoDialer (Автонабор номера)** — автоматичний набір телефонного номеру по модему, **Print Table (Печать таблицы)** — друк таблиці БД на принтері, **Run Macro (Выполнить макрос)** — завантаження макросу, **Run Query (Выполнить запрос)** — виконання запиту.

Після вибору потрібної команди (функції) відбувається перехід до вікна, де потрібно вибрати вигляд кнопки: **Text (Текст)** — у вигляді текстового повідомлення, **Picture (Рисунок)** — у вигляді малюнка піктограми (при цьому у списку потрібно вибрати назву малюнка). На останньому кроці роботи майстра потрібно вказати назву створюваної кнопки і натиснути **Finish (Готово)**.

При створенні у формі списків (випадаючих (Combo Box) або звичайних (List Box)) відкривається вікно майстра створення списків, де потрібно вказати спосіб складання списку:

- **I want the ... to look up the values in a table or query (Об'єкт "..." будет использовать значения из таблицы или запроса) (замість ... може бути combo box (поле со списком) або list box (список))** — створити комбінований список із записів однієї з таблиць БД або результатів роботи запиту. На наступних кроках роботи майстра потрібно вказати назву таблиці БД (запиту), на основі якої створюється список, а тоді (на третьому етапі) вибрати поля, які повинні входити в список. На четвертому етапі встановлюється чи виводити в списку ключове поле порядкових номерів, а на п'ятому чи сортувати записи в списку (якщо сортувати, то по якому полю). І на останньому кроці слід вказати назву списку;
- **I will type in the values that I want (Будет введен фиксированый набор значений)** — користувач сам заповнює список потрібними даними. При цьому, на наступному етапі відкривається вікно у вигляді таблиці, де потрібно ввести елементи списку. На наступних кроках роботи майстра, як і в попередньому випадку, необхідно вказати чи проводити сортування списку та назву списку (на останньому етапі);
- **Find a record on my form based on the value I selected in my combo box (Поиск записи в форме на основе значения, которое содержит поле со списком)** — як і в першому випадку, виводити комбінований список записів БД, але, на відміну від нього, при переміщенні по списку, автоматично змінюється відображення активного запису в формі. На наступному кроці роботи майстра потрібно вибрати поля, які будуть відображатись у цьому списку. На третьому кроці слід встановити, чи виводити в списку ключове поле порядкових номерів, а на четвертому (останньому) — вказати назву списку.

Однією із переваг Access є можливість створення **багатотабличних (зв'язаних) форм**, або форм, які в одному вікні показують дані з декількох зв'язаних таблиць. Для цього потрібно спочатку створити у вікні конструктора форм головну форму, а тоді перетягнути в нього з головного вікна БД (рис. 2.1) піктограму відповідної форми, яка повинна бути підформою.

Однією із переваг Access 2003 є можливість створення багатосторінкових форм. При цьому кожна сторінка, представляється окремою закладкою і може містити звичайні для форм поля та об'єкти.

Для створення багатосторінкової форми потрібно в панелі інструментів об'єктів керування вікна конструктора форм вибрати піктограму **Tab Control (Набор вкладок)** і курсором миші обвести прямокутну ділянку, в якій буде знаходитись вікно багатосторінкової форми.

При цьому створюється вікно форми з двох закладок. Якщо потрібно добавити ще одну закладку, то потрібно викликати контекстне меню на даному об'єкті і вибрати команду **Insert Page (Добавить вкладку)**. Для знищення закладки потрібно вибрати команду **Delete Page (Удалить вкладку)**.

За замовчуванням закладки форми називаються **Page1 (Вкладка1), Page2 (Вкладка2)** і т.д., але користувач може перейменувати назву закладки. Для цього потрібно викликати контекстне меню на відповідній закладці і вибрати в ньому команду **Properties (Свойства)**. Відкривається вікно, де слід вибрати закладку **All (Все)** і в полі **Name (Имя)** вказати її нове ім'я.

Подальший процес створення полів та об'єктів в закладках вікна форми нічим не відрізняється від описаного вище. Єдина відмінність полягає в тому, що тут об'єкти та поля потрібно створювати для кожної закладки окремо.

Якщо користувач бажає встановити стандартний стиль зовнішнього вигляду вікна форми, то в піктографічному меню конструктора форм потрібно натиснути піктограму , або в пункті **Format (Формат)** головного меню вибрати команду **AutoFormat... (Автоформат...)**. В будь-якому випадку відкривається вікно, подібне на четверте діалогове вікно майстра створення форм (рис. 4.4), де слід вибрати стиль зовнішнього вигляду форми. Якщо в даному вікні натиснути кнопку **Options>> (Параметры>>)**, то в нижній частині вікна з'являються додаткові опції, що встановлюють елементи стилю, які потрібно застосувати для даної форми: **Font (Шрифт)** — параметри шрифтів всіх елементів форми, **Color (Цвет)** — схему кольорів, **Border (Границы)** — стилі рамок та обводок навколо об'єктів. Крім цього, у вікні налаштування стилю оформлення форми пристуня командна кнопка **Customize... (Настройка...)**, яка відкриває додаткове діалогове вікно налаштування стилю. Дане вікно містить три опції: **Create a new AutoFormat based on the <назва> (создание нового стиля на основе стиля объекта: <назва>)** — на основі оформлення даної форми створити новий стиль оформлення, **Update <навза стилю> with values from the Form <назва> (обновление стиля <назва стилю> с атрибутами объекта: <назва>)** — обновити параметри стилю <назва стилю> згідно із змінами зробленими для даної форми, **Delete <навза стилю> (Удаление стиля <навза стилю>)** — знищити вибраний стиль.

Примітка

Access не перепитує необхідності знищення та зміни параметрів стилю оформлення форми, тому при виборі кнопки налаштування стилю Customize... (Настройка...) у вікні AutoFormat... (Автоформат...) потрібно бути дуже обережним, щоб не втрати відповідний стиль

Інколи, для надання формі більш привабливого вигляду у якості її фону застосовують різні графічні зображення. Для встановлення на формі фонового зображення потрібно викликати контекстне меню на спеціальному мітичку в лівому верхньому куті конструктора форми і вибрати команду **Properties... (Свойства)**. Відкривається вікно (рис. 4.8), де слід вибрати закладку **All (Все)**, а в ній поле **Picture (Рисунок)**. При цьому в полі справа необхідно вказати шлях та ім'я графічного файлу для того, щоб вибрати назву файлу в додатковому діалоговому вікні. Щоб зображення було вставлене в форму, а не зв'язане з нею у полі **Picture Type (Тип рисунка)** потрібно вибрати **Embended (Внедренный)**.

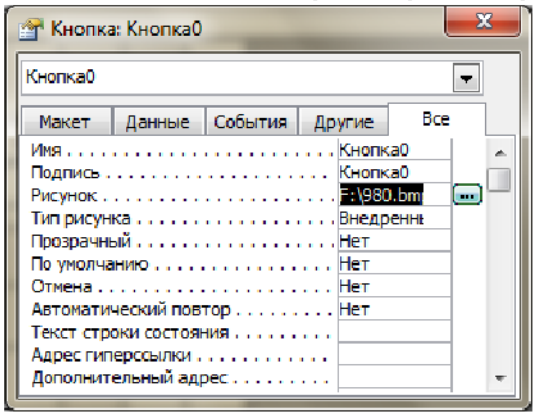


Рисунок 4.8 Вікно властивостей форми

Після цього, в пункті **Picture Alignment (Выравнивание рисунка)** (рис. 4.8) потрібно вказати розміщення малюнка відносно контура форми: **Top Left (Сверху слева)** — у лівому верхньому куті, **Top Right (Сверху справа)** — правому верхньому, **Center (По центру)** — по центру вікна форми, **Bottom Left (Снизу слева)** — лівому нижньому, **Bottom Right (Снизу справа)** — правому нижньому та **Form Center (По центру формы)** — по центру самої форми.

Крім цього, в пункті **Picture Size Mode (Масштабы рисунка)** (рис. 4.8) можна встановити масштаб цього зображення: **Clip (Фрагмент)** — встановити натуральні розміри зображення, **Stretch (Вписать в рамку)** — зображення розтягується або стискається до розмірів вікна форми або **Zoom (По размеру рамки)** — пропорційно розтягується або стискається зображення до максимально можливих розмірів, щоб не відбувалась його обрізка. Ще один пункт **Picture Tiling (Мозаичное заполнение)** встановлює режим дублювання зображення мозаїкою (параметр **Yes (Да)**).

4.3 Робота із вікном форми

Для того, щоб відкрити раніше створену форму, потрібно двічі натиснути ліву клавішу миші на її піктограмі у головному вікні БД або вибрати командну кнопку **Open (Открыть)**. В будь-якому випадку форма відкриється в режимі форми, де можна працювати з даними БД.

Технологія створення нових записів в БД з використанням форм нагадує технологію введення даних в таблицю. Тобто, для того, щоб добавити новий запис, потрібно перейти до порожнього запису, який завжди знаходиться в кінці таблиці, і ввести новий запис. При цьому після створеного запису автоматично з'являється ще один порожній запис.

Крім цього, новий запис можна створити, вибравши в пункті меню **Edit (Правка)** підпункт **Go To (Перейти)**, а в ньому команду **New Record (Новая запись)**, або скористатись піктограмою .

Примітка

Для того, щоб у вікні формн заборонити можливість додавання нових запнсів потрібно перейти в режим конструктора форм. Тоді на мітничку викликати контекстне меню і вибрати команду Properties (Свойства). Відкривається вікно властивостей форми (рис. 4.8), де в полі Allow Additions (Разрешить добавление) вибрати параметр No (Нет).

Для того, щоб знищити запис у формі потрібно перейти до нього і в пункті меню **Edit (Правка)** вибрати підпункт **Delete Record (Удалить запись)**, або скористатись піктограмою . Можна також натиснути ліву клавішу миші на ділянці виділення запису (в лівій частині форми) та нитиснути клавішу **Delete**. В будь-якому випадку відкривається попереджувальне діалоговое вікно, де потрібно підтвердити операцію кнопкою **Yes (Да)**. При цьому, якщо даний запис не має підлеглих записів в зв'язаних таблицях, або дозволено каскадне знищення записів, то дана операції завершиться успішно.

У формі можна також здійснювати редагування записів. Для цього потрібно перейти на відповідний запис і внести зміни в дані відповідного поля. Access не внесе змін у відповідну таблицю поки користувач не перейде до іншого запису, або поки запис не буде збережений примусово. Для цього потрібно натиснути комбінацію кла-

віш **Shift+Enter** або в пункті меню **Record (Запись)** вибрати команду **Save Records (Сохранить запись)**. Після збереження запису можна відмінити операцію редагування, якщо відразу після цього натиснути піктограму або в пункті меню **Edit (Правка)** вибрати команду **Undo typing (Отменить ввод)**.

Примітка

В Access є можливість заборонити у вікні форми редагування деяких полів. Для цього потрібно перейти в режим конструктора форм, відмітити це поле та натиснути піктограму або в контекстному меню поля вибрати Properties (Свойства). Відкривається вікно, де в пункті Locked (Блокировка) потрібно вибрати Yes (Да).

Примітка

Для того, щоб заборонити редагування будь-яких полів форми потрібно також перейти до конструктора форм і, не відмічаючи ніяких елементів, натиснути піктограму або в контекстному меню мітчика вибрати команду Properties (Свойства). В будь-якому випадку відкривається вікно властивостей форми (рис. 9.33), де в полі Allow Edits (Разрешить изменение) вибрати параметр No (Нет).

В Access з'явився новий засіб для роботи з даними в режимі форм — це **фільтр по формі**. Фільтр по формі дозволяє відфільтрувати записи з використанням форми для встановлення умов вибору. Наприклад, потрібно знайти всі записи з прізвищами "Баран".

Для цього потрібно скористатись фільтром по формі, виконавши наступну послідовність дій. Спочатку, в режимі форми, вибрати в пункті меню Records (Записи) підпункт **Filter (Фильтр)**, а в ньому команду **Filter by Form (Изменить фильтр)** або натиснути піктограму. При цьому зовнішній вигляд форми зміниться, її поля стануть порожніми.

У відповідне поле форми потрібно ввести умову відбору. При цьому можна вводити складні умови, використовуючи для їх об'єднання логічні оператори (див. таблицю 3.2, тема 3). Наприклад, якщо в поле "Прізвище" ввести запис "Баран or Сиротюк", то в формі залишаться лише записи в яких є прізвище "Баран" або "Сиротюк". Для завантаження створеного фільтру потрібно вибрати в пункті меню **Records (Записи)** команду **Apply Filter/Sort (Применить фильтр)** або натиснути піктограму.

Фільтр буде діяти доти, поки він не буде відмінений. Для цього потрібно в пункті меню **Records (Записи)** вибрати команду **Remove Filter/Sort (Удалить фильтр)**.

Іншим типом фільтру, що використовується в Access є **фільтр по виділеному значенню (Filter by Selection)**. Для реалізації цієї можливості необхідно у вікні форми встановити курсор в комірку поля, де знаходиться потрібне значення і в пункті меню **Records (Записи)** вибрати підпункт **Filter (Фильтр)**, а в ньому команду **Filter by Selection (Фильтр по выделенному)**.

Примітка

За допомогою фільтра можна також вибирати записи, що не містять певного значення у полі. Для цього необхідно встановити курсор в комірку поля, що містить значення, записи з яким не потрібно показувати і в пункті меню Records (Записи) вибрати підпункт Filter (Фильтр), а в ньому команду Filter Excluding Selection (Исключить выделенное).

ТЕМА 5 РОБОТА З ЗВІТАМИ В ACCESS

5.1 Типи звітів. Автозвіти. Створення звітів за допомогою майстра

Звіт (Report) — це кінцевий результат виконання багатьох задач керування БД. За допомогою звітів дані можна подати фактично в будь-якому форматі, з різними рівнями деталізування. Хоча для отримання потрібних даних можна використовувати запити, звіти краще підходять для візуального та друкованого подання даних.

У звітах також можуть відображатись обчислювані по вхідних даних значення, наприклад, функції підведення підсумків (сума, середнє арифметичне, кількість та ін.).

Практично всі звіти, які можна створити в Access, поділяються на три категорії:

- **рядкові звіти (Tabular)** — в цих звітах дані відображаються по рядках, причому кожне поле БД займає окрему колонку;
- **звіти в колонку (Columnar)** — в цих звітах всі поля відображаються в одну колонку і вирівнюються по лівому краю. Звіти, зверстані в колонку, дуже схожі на форми, але вони використовуються лише для відображення або друку даних, а не для їх введення та редагування;
- **діаграми** — звіти у вигляді діаграми. Вони також подібні на однойменні форми, але їх не можна редагувати;
- **поштові наклейки** — це звіти, які використовують для друку поштових наклейок. В Access є майстер звітів, який спеціально призначений для створення наклейок в різних форматах.

Крім цього, звіти можуть бути: **звичайні** — створені на основі однієї таблиці БД, та **реляційні** — створені на основі декількох зв'язаних таблиць.

Перш ніж створювати звіт, його необхідно спланувати. Рекомендується така послідовність розробки звіту.

1. Визначити розмітку звіту. Спочатку розмітку потрібно продумати і бажано чорновий варіант розробити на папері.

2. Зібрати потрібні дані. Як джерело даних для звіту Access можна використати одну або кілька таблиць або запитів. Для того, щоб відібрати і впорядкувати дані, краще за все використати запити.

3. Сконструювати структуру звіту. Використовуючи майстер звіту або вікно конструктора, потрібно розмістити необхідні поля, мітки та інші об'єкти.

4. Переглянути або роздрукувати звіт. Після завершення розробки звіту його можна переглянути, вибравши команду **Print Preview (Предварительный просмотр)** з пункту меню **File (Файл)**. Щоб роздрукувати звіт, потрібно вибрати в пункті меню **File (Файл)** команду **Print (Печать)** або натиснути комбінацію клавіш **Ctrl+P**.

Дані в звіті можна групувати по певному полю або таблиці. Наприклад, для звіту, що містить кілька десятків записів про покупців із різних міст.

В цьому випадку можна вибрати групування по полю "місто", в результаті чого дані будуть перегруповані таким чином, що назва міста буде зустрічатись лише по одному разу (всі записи з однаковим вмістом поля "Місто" будуть об'єднані в своєрідні групи).

В Access існує три основних методи створення звітів:

- **автозвіт (AutoReport)** — автоматичне створення звітів за шаблоном останнього створюваного звіту даного типу, або шаблону за замовчуванням;
- **за допомогою майстра звітів (Report Wizard)** — покрокове створення звіту за допомогою спеціальної підпрограми майстра;
- **за допомогою конструктора звітів (Disign View)** — універсальний спосіб створення звітів, який дозволяє детально налаштувати всі його елементи. Проте цей метод є найбільш трудомістким у реалізації. Переважно звіти створюють іншими способами (майстер, автозвіт), а після цього завантажують в конструктор для налаштування деяких його параметрів.

Для створення звіту потрібно в головному вікні БД (рис. 2.1) вибрати розділ **Reports (Отчеты)**, а в ньому командну кнопку Создать **New (Создать)**. При цьому відкривається вікно **New Report (Новый отчет)** (рис. 5.1), де потрібно вибрати спосіб створення звіту. Після цього у випадаючому списку, в нижній частині вікна, слід вказати назву таблиці БД, запиту або фільтра по формі, на основі якого буде створений звіт та натиснути командну кнопку **OK**. В списку режимів створення звіту можна вибрати:

- **Design View (Конструктор)** — створення структури звіту вручну, за допомогою конструктора звітів;
- **Deport Wizard (Мастер отчетов)** — створення звітів за допомогою майстра звітів;
- **AutoReport: Columnar (Автоотчет: в столбец)** — автоматичне створення звіту в колонку за зразком останнього створеного звіту такого типу;
- **AutoReport: Tabular (Автоотчет: ленточный)** — автоматичне створення рядкового звіту за зразком останнього створеного звіту такого типу;
- **Chart Wizard (Мастер диаграммы)** — створення звіту у вигляді діаграми;
- **Label Wizard (Почтовые наклейки)** — створення звітів у вигляді поштових наклейок.

Найбільш швидкий спосіб створення звітів — це використання функції автозвіт. Для цього у вікні **New Report (Новый отчет)** потрібно вибрати один з типів **AutoReport (Автоотчет)** і натиснути кнопку **OK**. При цьому автоматично створюється звіт за зразком останнього створеного звіту такого типу. Якщо звітів такого типу на ПК ще не було, то створиться звіт з параметрами, прийнятими в Access за замовчуванням.

Для створення звіту за допомогою майстра звітів потрібно в діалоговому вікні **New Report (Новый отчет)** вибрати режим **Report Wizard (Мастер отчетов)** (у випадаючому списку в нижній частині вікна потрібно вибрати таблицю БД, запит або фільтр по формі). На першому кроці роботи майстра звітів відкривається вікно, яке практично нічим не відрізняється від першого вікна майстра створення форм (рис. 4.2). В цьому вікні за

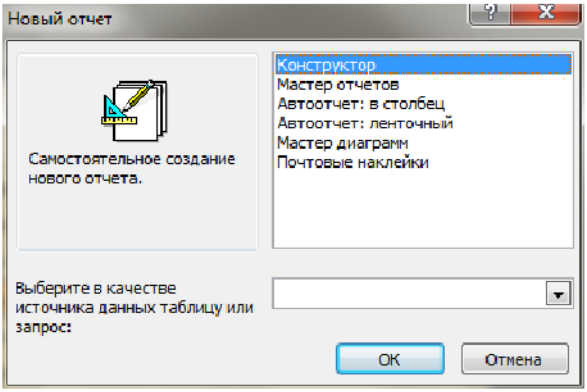


Рисунок 5.1 Вікно створення звітів

допомогою кнопки > потрібно вибрати поля (>> — для вибору всіх полів), які повинні ввійти в звіт. Якщо звіт повинен містити поля одночасно з декількох зв'язаних таблиць (запитів), то, після визначення полів першої з них, у випадаючому списку **Tables/Queries (Таблицы/запросы)** потрібно вибрати наступну таблицю (запит), і повторити операцію вибору полів.

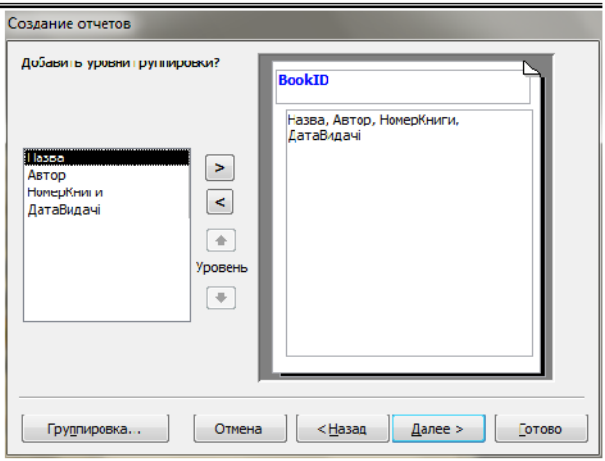


Рисунок 5.2 Друге вікно властивостей майстра звітів

Після натискання **Next (Далее)** відбувається перехід до другого етапу роботи майстра звітів. При цьому відкривається вікно, показане на рисунку 5.2. За допомогою піктограм цього вікна можна встановити поля, по яких буде відбуватись групування об'єктів, та рівні групування полів. Для цього потрібно вибрати в списку поле, по якому слід згрупувати записи і натиснути піктограму >. Якщо вибрати ще одне поле і натиснути піктограму >, то створиться другий рівень групування. Аналогічно створюється третій, четвертий рівень групування і т.д. Піктограмами із стрілками вверх та вниз можна змінювати порядок чергування рівнів групування. Кнопка **Grouping Options... (Группировка...)** відкриває додаткове вікно, де можна змінити відстань між записами поля, по якому відбувається групування. Встановивши список полів, навколо яких будуть групуватись дані, потрібно натиснути командну кнопку **Next (Далее)** для переходу до наступного етапу роботи майстра.

Примітка

Якщо на першому етапі роботи майстра створення звітів вибрано поля одночасно з декількох таблиць (запитів), то на другому кроці відкривається додаткове вікно, де необхідно вказати яким чином будуть структуровані дані в звіті. Майстер перевірить зв'язки між таблицями і, в залежності від вибраних елементів, пропонує декілька варіантів групування записів у звіті. Лише після натискування в ньому Next (Далее) з'являється вікно, зображене на рисунку 9.35. Якщо натиснути кнопку Show Me More Information (Вывести дополнительные сведения), то відкривається допоміжне інформаційне вікно, в якому будуть представлені приклади різних способів групування.

На третьому етапі роботи майстра звітів відкривається діалогове вікно (рис. 5.3), де у випадаючих списках потрібно вибрати поля, по яких потрібно сортувати дані в звіті. Вікно містить чотири списки (тобто можна сортувати до чотирьох полів одночасно), які пронумеровані в порядку черговості сортування. Тобто, спочатку йде сортування по полю, вказаному в першому списку, а вже коли в цьому полі є однаковий вміст (наприклад, однакові прізвища), то сортування відбувається по полю, вказаному в другому списку. Якщо і там однакові записи, то сортування йде по полю вказаному в третьому списку і т.д. Справа від кожного списку знаходиться кнопка, яка

визначає порядок сортування: **Ascending (по возрастанию)** — в порядку зростання (по алфавіту від "А" до "Я"), або **Descending (по убыванию)** — спадання (по алфавіту від "Я" до "А"). Крім цього, в даному вікні може бути присутня командна кнопка **Summary Options (Итоги)**, яка відкриває додаткове діалогове вікно, в якому можна встановити статистичну функцію для визначення підсумків по активному полю (сума, середнє арифметичне, найбільше значення або намйенше).

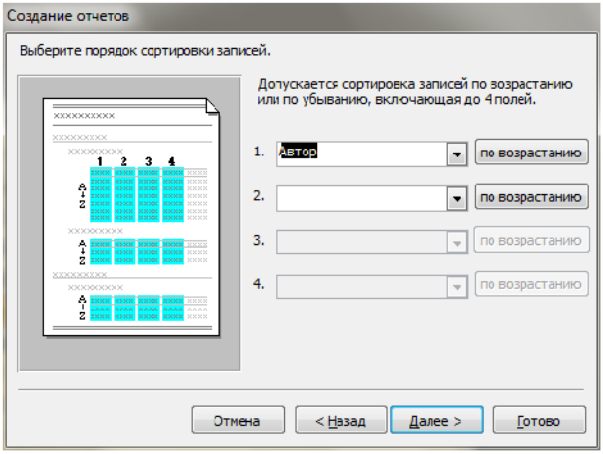


Рисунок 5.3 Третє вікно властивостей майстра звітів

Після натискування кнопки **Next (Далее)** відбувається перехід до четвертого вікна роботи майстра звітів. При цьому відкривається вікно, показане на рисунку 5.4. В даному діалоговому вікні потрібно створити макет майбутнього звіту. В розділі Layout (Макет) можна вибрати:

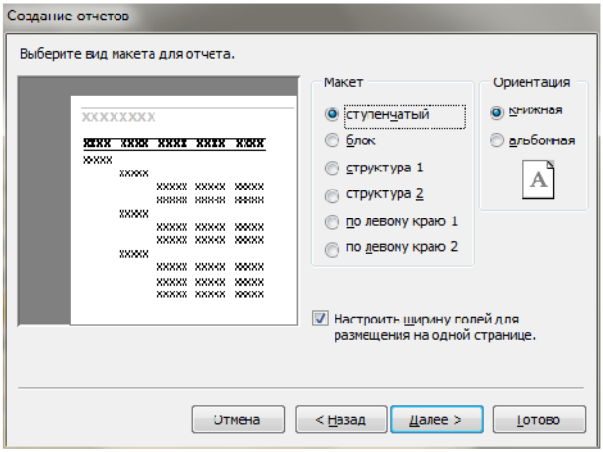


Рисунок 5.4 Четверте вікно властивостей майстра звітів

- **Stopped (в столбец)** — стовпчикове розміщення із виділенням полів, по яких відбувається групування в окремі рівні;
- **Block (блок)** — поблокове розміщення записів із виділенням полів, по яких відбувається групування в один рівень;
- **Outline 1 та Outline 2 (табличный)** — табличне розміщення записів із виділенням полів (таблиць), по яких відбувається групування;
- **Align Left 1 та Align Left 2 (выровненный)** — також табличне розміщення, але поля, по яких відбувається групування вирівнюються по лівому краю, разом із іншими полями.

Розділ **Orientation (Ориентация)** встановлює просторове розміщення таблиці звіту відносно аркуша паперу: **Portrait (книжная)** — вертикальне або **Landscape (альбомная)** — горизонтальне розміщення аркуша.

Якщо відмітити опцію **Adjust the field width so all fields fit on a page (Настроить ширину полей для размещения на одной странице)**, то ширина полів в звіті вста-

новлюється автоматично таким чином, щоб таблиця із звітом зайняла всю ширину аркуша паперу.

Після натискування **Next (Далее)** відбувається перехід до п'ятого етапу роботи майстра. Відкривається вікно, де потрібно вибрати стиль фону звіту. Дане вікно дуже схоже на аналогічне вікно роботи майстра створення форм (рис. 4.4). Вибравши стиль фону звіту, потрібно натиснути кнопку **Next (Далее)** для переходу до останнього вікна роботи майстра звітів.

На останньому етапі роботи майстра звітів відкривається діалогове вікно, повністю аналогічне до завершального вікна роботи майстра створення форм. Встановивши в даному вікні назву звіту, та відмітивши відповідний режим завершення роботи майстра, потрібно натиснути командну кнопку **Finish (Готово)**.

Створення поштових наклейок

В Access є можливість за даними таблиць БД створювати поштові та інші наклейки. Це дозволяє, наприклад, автоматично заповнити конверти поштових листівок, вибравши дані (прізвище, ім'я, по батькові, адреса і т.д.) на отримувача безпосередньо із таблиці БД.

Для створення поштових та інших наклейок в Access використовується спеціальна підпрограма — майстер створення наклейок, яка вибирається у вікні **New Report (Новый отчет)** (рис. 5.1) командою **Label Wizard (Почтовые наклейки)**. При цьому у випадаючому списку потрібно вибрати назву таблиці або запиту, по якому буде створена наклейка

Відкривається перше вікно майстра створення наклейок (рис. 5.5), в якому потрібно вибрати одиниці вимірювання розмірів (**Unit of Measure (Единицы измерения)**), тип друку наклейок (**Label Type (Тип наклеек)**) та самі розміри наклейки. Причому можна вибрати один із двох типів друку наклейок: **Sheet feed (на листах)** — на окремих аркушах паперу або **Continuous (рулонные)** — на рулоні паперу.

Розміри вибирають із списку стандартних типів, які поділені на декілька груп. Групи типів наклейок вибираються із випадаючого списку **Filter by manufacturer (Фильтр по изготовителю)**. Кнопка **Customize... (Настройка...)** дозволяє користувачу задати нестандартні розміри наклейки. При цьому відкривається вікно із списком наклейок із розмірами, заданими користувачем. Для встановлення нових нестандартних розмірів наклейки потрібно вибрати командну кнопку **New... (Создать)**, або **Duplicate... (Дублировать...)**, якщо в списку вже є наклейки. Після чого відкривається додаткове вікно (рис. 5.6), де також необхідно встановити одиниці вимірювання розмірів, тип друку

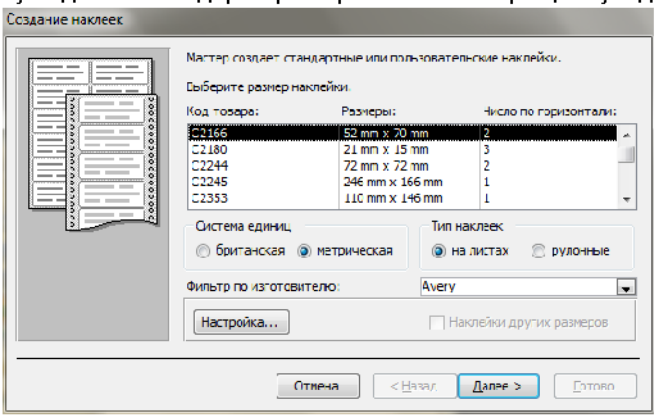


Рисунок 5.5 Перше вікно майстра створення наклейок

наклейок, та орієнтацію аркуша паперу. Після цього в спеціальних полях, в ділянці перегляду нижньої частини вікна (рис. 5.6), слід вказати кожен параметр розмірів. Встановши всі параметри, в полі **Label Name: (Название:)** потрібно ввести назву створеного формату наклейки на натиснути кнопку **OK**, а в наступному вікні **Close (Закрыть)**.

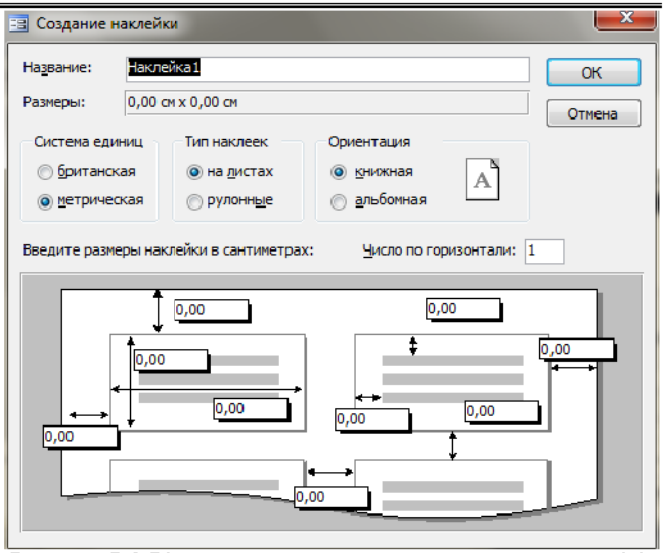


Рисунок 5.6 Вікно встановлення користувацьких розмірів

Після вибору командної кнопки **Next (Далее)** відбувається перехід до другого вікна майстра створення наклейок, в якому потрібно вибрати параметри шрифту для написів на наклейці (гарнітура, кегль, стиль, ширина літер та їх колір). Крім цього, в даному вікні можна вибрати ще один специфічний параметр шрифту — **Font weight: (насыщенность)**, який дозволяє встановити насиченість символів шрифту.

В третьому вікні майстра за допомогою кнопки > необхідно вибрати поля значення яких повинні виводитись в заголовку наклейки, а в четвертому всі інші поля, що будуть відображені на ній.

Останнє вікно підпрограми майстра створення наклейок дозволяє вказати назву звіту-наклейки та вибрати одну з опцій:

- **See the labels as they will look printed (Посмотреть наклейки в том виде, в котором они будут напечатаны)** — переглянути зображення наклейки та роздрукувати його на принтері;
- **Modify the label design (Изменить макет наклеек)** — відкрити вікно конструктора для редагування параметрів наклейки.

5.2. Робота з конструктором звітів

Якщо звіти, створені за замовчуванням або за допомогою майстра звітів користувача не задовольняють, то слід створити звіт вручну за допомогою конструктора звітів.

Для реалізації цієї можливості потрібно в головному вікні БД (рис. 2.1) вибрати розділ **Reports (Отчеты)**, а в ньому командну кнопку **New (Создать)**. При цьому відкривається вікно **New Report (Новый отчет)** (рис. 5.1), де необхідно вибрати режим **Design View (Конструктор)**, а тоді, у випадаючому списку, в нижній частині вікна вибрати назву таблиці БД, запиту або фільтра по формі і натиснути **OK**. Відкривається вікно конструктора звітів.

Звіт в Access складається з кількох розділів, тобто тут прийнято структурний стиль розробки звітів (кожен звіт складається з декількох структурних одиниць — розділів). Звіт може складатись з наступних розділів:

- **розділ заголовка звіту (Report Header)** — є необов'язковим розділом, який містить інформацію, що з'являється лише один раз - на початку звіту;
- **розділ верхнього колонтитула (Page Header)** — з'являється на початку кожної сторінки звіту. Він може містити заголовки колонок табличного звіту;
- **розділ ділянки даних (Detail)** — містить основну частину даних, які потрібно представити в звіті. Саме в цьому розділі знаходяться поля даних та назви полів;
- **розділ нижнього колонтитула (Page Footer)** — з'являється в кінці кожної сторінки звіту;
- **розділ приміток до звіту (Report Footer)** — є необов'язковим розділом, який містить інформацію, що з'являється лише один раз - в кінці звіту. Як правило, він використовується для розміщення підсумовуючих даних;
- **розділ заголовка групи сортування та групування (Sorting and Grouping)** — необов'язковий розділ, який з'являється на початку кожної групи записів згрупованих по певному полю (див. розділ 5.1 даної частини посібника).

Звіти стають більш зрозумілими та структурованими, якщо дані в них організовані в групи та підгрупи. Для створення цих елементів структури в конструкторі використовується діалогове вікно **Sorting and Grouping (Сортировка и группировка)** (рис. 5.7). Для його виклику потрібно в режимі конструктора натиснути піктограму або вибрати однойменну команду в пункті меню **View (Вид)**. В цьому вікні можна визначати поля або вирази, по яких Access буде групувати дані. Крім цього, згруповані дані можна сортувати по зростанню або спаданню. Символ **Sorting and Grouping (Сортировка и группировка)**, на кнопці виділення в лівій сірій колонці цього діалогового вікна вказує на те, що дане поле або вираз (що знаходиться праворуч від символа) використовується для групування записів. Рівнів групування може бути декілька, тобто можна створювати підгрупи всередині груп рядків.

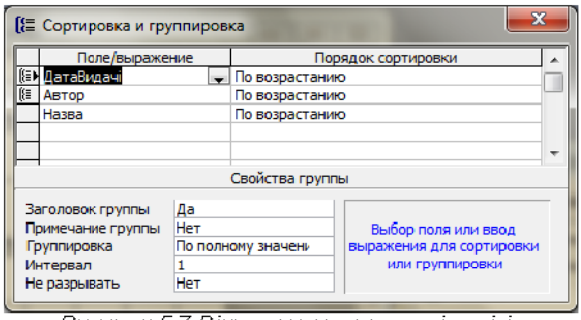


Рисунок 5.7 Вікно групування полів звітів

Для кожного поля групування записів потрібно в нижній частині вікна встановити параметри налаштування: **Group Header (Заголовок группы)** — вставити в звіт розділ із назвою групи у верхній колонтитул, **Group Footer (Примечание группы)** — у нижній колонтитул, **Group On (Группировка)** — вибрати спосіб групування, **Group Interval (Интервал)** — встановити кількість сиволів, що будуть використовуватись для нової групи, **Keep Together (Не разрывать)** — встановити, як група буде розміщуватись на сторінці. Параметри **Group On (Группировка)** та **Group Interval (Интервал)** пов'язані між собою і в сукупності визначають яким саме чином будуть групуватись дані.

Access дозволяє групувати дані двома способами: за категоріями (кожній категорії повинне відповідати унікальне значення), та за діапазоном значень, який може бути як числовим, так і алфавітним. Для групування даних можна використовувати не лише значення полів але й вирази.

Зліва та зверху від робочої частини вікна конструктора знаходяться розмірні **лінійки (Ruler)**, які призначені для визначення відстаней між об'єктами у звіті та їх розмірів. Якщо лінійки не відображаються, то в пункті меню **View (Вид)** вибрати команду **Ruler (Линейка)**.

Створення об'єктів та полів в розділах конструктора звітів нічим не відрізняється від створення аналогічних об'єктів для форм. При цьому використовується та ж панель інструментів із піктограмами об'єктів. Якщо ця панель не відображається, то в пункті меню **View (Вид)** необхідно вибрати команду **Toolbox (Панель элементов)**. Для розміщення у звіті полів таблиць можна використати вікно із списком полів, так само як воно виконується при створенні форм. Якщо це вікно відсутнє, то в пункті меню **View (Вид)** необхідно вибрати команду **Field List (Список полей)**.

Примітка

Потрібно відмітити, що такі елементи керування як командні кнопки, опції, перемикачі, вимикачі та списки використовувати недоречно, оскільки звіт після створення не можна використовувати для редагування даних.
Отже ці елементи жодних дій виконувати не будуть.

У звітах, як і в формах, можна додавати обчислювальні елементи керування для відображення даних, які базуються на обчисленнях. В обчисленнях можна використовувати будь-яку доступну Access функцію. Представимо кілька найпоширеніших функцій, що при цьому використовуються:

- **Avg (значення)** — знаходить середнє арифметичне набору значень;
- **Date ()** — виводить поточну дату;
- **Max (значення)** — знаходить максимальне значення із набору даних;
- **Min (значення)** — знаходить мінімальне значення із набору даних;
- **Sum (значення)** — знаходить суму значення із набору даних;
- **Time ()** — виводить поточний час.

Для створення формули, за якою буде проводитись обчислення, спочатку потрібно поставити знак "=" ("дорівнює"), а тоді ввести формулу.

5.3. Перегляд та друк звітів. Перетворення звітів в інший формат

На будь-якій стадії створення звіт можна переглянути, як він буде виглядати після друку на папері. Для цього потрібно в пункті в пункті меню **File (Файл)** вибрати команду **Print Preview (Предварительный просмотр)**. В будь-якому випадку відкриється вікно попереднього перегляду створеного звіту, що схоже на вікно попереднього перегляду документу Word.

Для перегляду частини звіту, що не поміщається на екрані, можна використовувати лінійки прокрутки або клавіші PageUp та PageDown.

Для того, щоб надрукувати звіт, потрібно відмітити його у вікні бази даних і в пункті меню **File (Файл)** вибрати команду **Print (Печать)**. При цьому відкривається вікно, в якому можна встановити діапазон сторінок звіту для друку, кількість копій і вибрати тип поточного принтеру. Це вікно схоже на вікно встановлення параметрів друку електронних таблиць, але є його дещо скороченим варіантом.

Роздрукувати звіт можна також викликавши контекстне меню на його назві у вікні бази даних і вибравши в ньому команду **Print (Печать)**. Можна також відмітити назву звіту у вікні бази даних і вибрати в піктографічному меню піктограму . Але зверніть увагу, що при цьому друк буде виконуватись з параметрами, які встановлені за замовчуванням.

Звіти в Access можна не лише друкувати на принтері, але й експортувати в інші формати для передачі користувачам, які не мають можливості працювати з цією СКБД. При цьому звіт можна перетворити в формат RTF, електронних таблиць Microsoft Excel, Web-сторінок в форматах HTML, XML, різні текстові формати (txt, csv, tab, asc) з можливістю його збереження в кодуванні Windows, MS-DOS або Unicode. Крім цього, можна використовувати спеціальний формат копій звітів **snapshot**. В цьому форматі звіти можна переглядати, гортати, друкувати окремо від бази даних Access. При цьому використовується невелика програма — переглядач звітів **Snapshot Viewer**. Ця програма повністю незалежна від Microsoft Access і може бути встановлена окремо. В цьому випадку її можна завантажити із офіційного web-вузла Microsoft.

Для експортування звіту у один із вищеперерахованих форматів потрібно відкрити звіт і в пункті меню **File (Файл)** вибрати команду **Export (Экспорт)**. Відкривається звичайне вікно збереження документа, де у випадаючому списку **Save as type (Тип файла)** потрібно вибрати формат файлу в який експортується звіт, а в полі **File name (Имя файла)** — його назву. При виборі текстового формату відкривається додаткове вікно, де слід вказати тип кодування символів. Якщо вибрати формат **Snapshot**, а програма **Snapshot Viewer** на ПК не встановлена, то відбудеться спроба її інсталяції. Тому можливо буде виведено повідомлення із пропозицією вставити інсталяційний диск із пакетом Microsoft Office.

КОНСПЕКТ ЛЕКЦІЙ

по темі:

"Система керування базами даних Microsoft Access 2003"

© 2011. І.Л. Тхір, Н.П. Тхір

© 2011. Технічний коледж ТНТУ ім. І.Пулюя