

Тема уроку: Поняття моделі даних, бази даних. Поняття й призначення систем керування базами даних. Огляд реляційної моделі даних. Модель «сутність-зв'язок». Поняття відношення, атрибута, ключа, зв'язку. Класифікація зв'язків за множинністю та повнотою. Правила побудови моделі даних предметної області. Поняття таблиці, поля, запису. Основні етапи роботи з базами даних у середовищі системи керування базами даних. Відображення моделі «сутність-зв'язок» на базу даних. Властивості полів, типи даних. Введення даних у таблиці. Сортуння, пошук і фільтрація даних.

План:

- 3.1. Загальні відомості про системи управління базами даних
- 3.2. Access для Windows
 - 3.2.1. Що таке база даних
 - 3.2.2. Випадки, коли доцільно використовувати бази даних
 - 3.2.3. Завдання, що розв'язуються з допомогою Access
 - 3.2.4 Об'єкти бази даних
 - 3.2.5. Проектування бази даних
 - 3.2.6. Робота з вікном бази даних
 - 3.2.7. Створення таблиць
 - 3.2.8. Визначення імен полів
 - 3.2.8.1. Вибір типів даних
 - 3.2.8.2. Використання полів опису
 - 3.2.8.3. Використання властивостей полів
 - 3.2.9. Додавання первинного ключа
 - 3.2.10. Реорганізація і видалення полів при розробці таблиць
 - 3.2.11. Збереження таблиці
 - 3.2.12. Додавання даних в таблицю
 - 3.2.13. Редагування і видалення даних
 - 3.2.14. Друкування таблиці

3.1. Загальні відомості про системи управління базами даних

При розв'язуванні багатьох інформаційних задач сьогодні широко використовуються системи управління базами даних (СУБД) (DATA BASE) реляційного типу. В цих СУБД реалізується реляційна модель даних - зображення їх у табличному вигляді. Рядок такої таблиці еквівалентний запису файлу бази даних (БД), а графа - полю запису. Доступ до елемента даних здійснюється з допомогою зв'язку потрібного рядка (запису) з потрібною графою (полем).

Існує два типи систем управління БД: *файлами* і *реляційними базами даних*. В *системах управління файлами* (*File management systems*), іноді їх ще називають *плоскими файловими базами даних*, дані вводяться і зберігаються без індексування, а при пошуку і створенні вихідних форм обробляються послідовно. Тому ці системи недостатньо гнучкі при маніпулюванні даними. *Системи управління реляційними базами даних* (*Relational database management systems*) дозволяють користувачам маніпулювати даними більш вільно завдяки збереженню всіх даних тільки один раз і опису зв'язків між ними. Зв'язки являють собою загальні елементи даних, таких як прізвище покупця або номер його чекової книжки. Дані кожного набору інформації можна знайти, відобразити і відновити на підставі даних іншого набору.

До складу багатьох СУБД, призначених для роботи на персональних комп'ютерах, входять три основні компоненти: командна мова, інтерпретуюча система або компілятор для перетворення

команд до виду, придатного до виконання, і засоби взаємодії користувача із СУБД (інтерфейс користувача).

Командна мова служить для виконання і забезпечення потрібних операцій з даними, дає змогу створювати прикладні програми, екранні форми введення і виведення інформації, структури файлів БД і маніпулювати даними. Структура і можливості мови визначають вигляд і можливості конкретної СУБД.

У СУБД можна виконувати команди по одній або групами в авто-матичному режимі, заздалегідь записаними у спеціальний (програмний) файл. Команди мови СУБД записуються в текстовій формі, близькій до звичайної мови. Для того щоб персональний комп'ютер міг виконати таку команду, її треба перетворити на команду, що виконується комп'ютером.

Структура і вміст компонентів СУБД визначає її призначення і коло потенційних користувачів. Залежно від рівня підготовки і діапазону розв'язуваних завдань користувачів персональних комп'ютерів можна поділити на три групи:

тих, яким для розв'язання своїх завдань потрібен порівняно невеликий набір функцій пакета СУБД (створення файлів даних, введення і поновлення даних, виведення на друк нескладних вихідних документів тощо);

тих, яким необхідно знати багато функцій пакета СУБД, у тому числі і елементи командної мови, але не потрібно створювати складні програми;

програмістів, в завдання яких входить розробка складних прикладних програм з управління і обробки файлів БД.

Для користувачів першої групи необхідно включати до складу СУБД режим, що керується за допомогою меню. Користувачі другої групи можуть освоювати командний режим. Третя група користувачів чекає від СУБД достатньо повного набору зручних засобів для розробки складних програм.

Останні версії сім'ї СУБД **Access for Windows**, розроблені фірмою Microsoft, відповідають вимогам усіх перелічених груп користувачів.

3.2. Access для Windows

3.2.1. Що таке база даних

База даних - це систематизоване сховище інформації. Телефонний довідник - прекрасний приклад БД (рис. 3.1)

.

Рядки $\Rightarrow$ записи

Графи $\Rightarrow$ поля



Прізвище	Ім'я	Робочий телефон	Факс
Андрющенко	Іван	211-34-90	222-78-00
Белов	Андрій	442-65-98	441-09-65
Петренко	Ірина	216-77-99	221-45-09
Тарасенко	Юрій	274-55-85	441-32-98

Рис. 3.1. База даних абонентів

3.2.2. Випадки, коли доцільно використовувати бази даних

Комп'ютер і комп'ютерні програми слід сприймати як робочий інструмент, який щоденно використовується в роботі. До пакету Microsoft Office входить набір таких програм-інструментів, скориставшись якими, користувач зможе автоматизувати виконання щоденних завдань. Тому корисно знати, для яких завдань необхідно використовувати СУБД Access. Бази даних призначені для збереження інформації. Розглянемо кілька типових прикладів інформації, яка може зберігатися в комп'ютерних БД:

інформація про працівників установи;

облік продукції;

робочі плани;

відомості про покупців;

облік продажів;

постачальники;

накладні.

Зверніть увагу на те, що в усіх наведених прикладах підкреслюється функція зберігання і маніпулювання даними, а не розрахунки. В БД можна виконувати багато фінансових, статистичних та інших розрахунків, але останні здійснюються не так швидко, як, приміром, в електронних таблицях. Наприклад, СУБД - не найкраща програма для автоматизації розрахунків податку з прибутку: для розв'язування цієї задачі більш придатна електронна таблиця. Однак якщо користувачеві необхідно відслідковувати, аналізувати і зберігати інформацію про податки за кілька років, то БД краще знадобиться для збирання та збереження цієї інформації.

3.2.3. Завдання, що розв'язуються з допомогою Access

Microsoft Access - це система управління реляційними БД, призначена для роботи в середовищі Windows. Працюючи з Access, користувач може розв'язувати наступні завдання:

розбивати дані на логічно пов'язані між собою частини;

вводити, змінювати і знаходити необхідні дані;

знаходити підмножини даних за заданими умовами;

створювати форми і звіти;

автоматизувати виконання стандартних завдань;

графічно встановлювати зв'язки між даними;

вставляти рисунки у форми і звіти;

створювати особисті, готові до роботи з БД програми, які вміщують меню, діалогові вікна і командні кнопки.

3.2.4 Об'єкти бази даних

В Access поняття БД віднесено до одного файлу, що вміщує всю інформацію. Кожна БД Access складається з таких об'єктів: таблиць, запитів, форм, звітів, макрокоманд і модулів. У табл. 3.1 наведені основні об'єкти в Access.

Таблиця 3.1

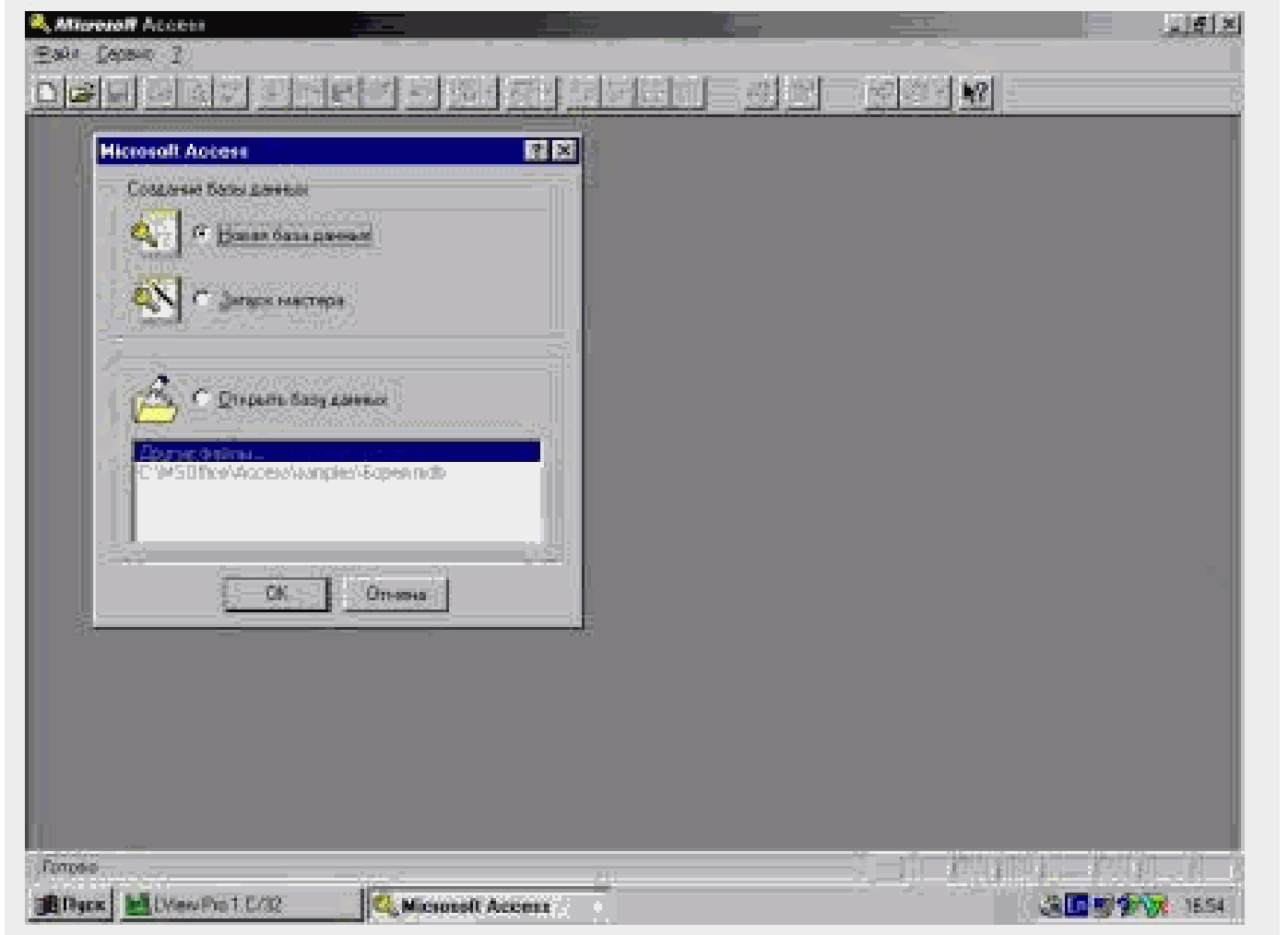
Об'єкти БД

Об'єкт	Опис
Таблиця	Вміщує дані в форматі таблиці; схожа на електронну таблицю
Запрос	Вибирає дані з таблиці на підставі заданої умови. Запити дозволяють бачити дані з кількох таблиць в одній
Форма	Відображує дані з таблиці або запиту на підставі описаного користувачем формату. Форми дозволяють проглядати, редагувати і друкувати дані. У формі можна відображати дані, взяті з кількох таблиць або запитів
Отчет	Відображує і друкує дані з таблиці або запиту на підставі описаного користувачем формату. В звіті дані редагувати не можна. Звіти можуть вміщати дані, взяті з кількох таблиць або запитів
Макрос	Автоматизує стандартні дії на підставі вибраних користувачем команд і подій
Программный модуль	Автоматизує складні операції, які не можна описати з допомогою макрокоманд. Програмні модулі - це процедури, написані мовою програмування Access Basic

3.2.5. Проектування бази даних

Перш ніж приступати до створення таблиць, форм і звітів, користувачеві необхідно витратити якийсь час на проектування БД. Далі подані деякі важливі поради, яких слід дотримуватися при розробці БД.

- Почніть з аналізу існуючих БД (створених вручну або за допомогою комп'ютера). Прогляньте



Після запуску Access з'явиться діалогове вікно (рис. 3.2).

Рис. 3.2. Головне вікно Access і початкове діалогове вікно

Головне вікно Access - це робоча область, де користувач маніпулює різними об'єктами Access (наприклад, таблицями). В цьому вікні виберіть БД, з якою будете працювати, або створіть нову. Створити нову БД можна з допомогою одного з двох методів.

Активізуйте пункт **Новая база данных (Blank Database)** і клацніть мишкою на кнопці **ОК**, щоб відкрити нову БД.

Активізуйте пункт **Запуск мастера (Database Wizard)** і клацніть мишкою на кнопці **ОК**, щоб запустити *Мастер базы данных*.

Створити або відкрити БД можна, закривши діалогове вікно (клацнувши мишкою на кнопці **Отмена (Cancel)**).

Для створення БД виконайте команду **файл** та її опцію **Создать базу данных (File → New Database)**, або клацніть мишкою на кнопці **Создать базу данных (New Database)** на панелі інструментів.

Для відкриття раніше створеної БД виконайте команду **Файл** та її опцію **Открыть базу данных (File → Open Database)**, або клацніть мишкою на кнопці **Открыть базу данных (Open Database)** на панелі інструментів.

Якщо буде активізовано пункт **Новая база данных**, програма зробить запит про ім'я створюваної БД (рис. 3.3). У вікно *Имя файла* необхідно ввести ім'я (наприклад, **probal**).

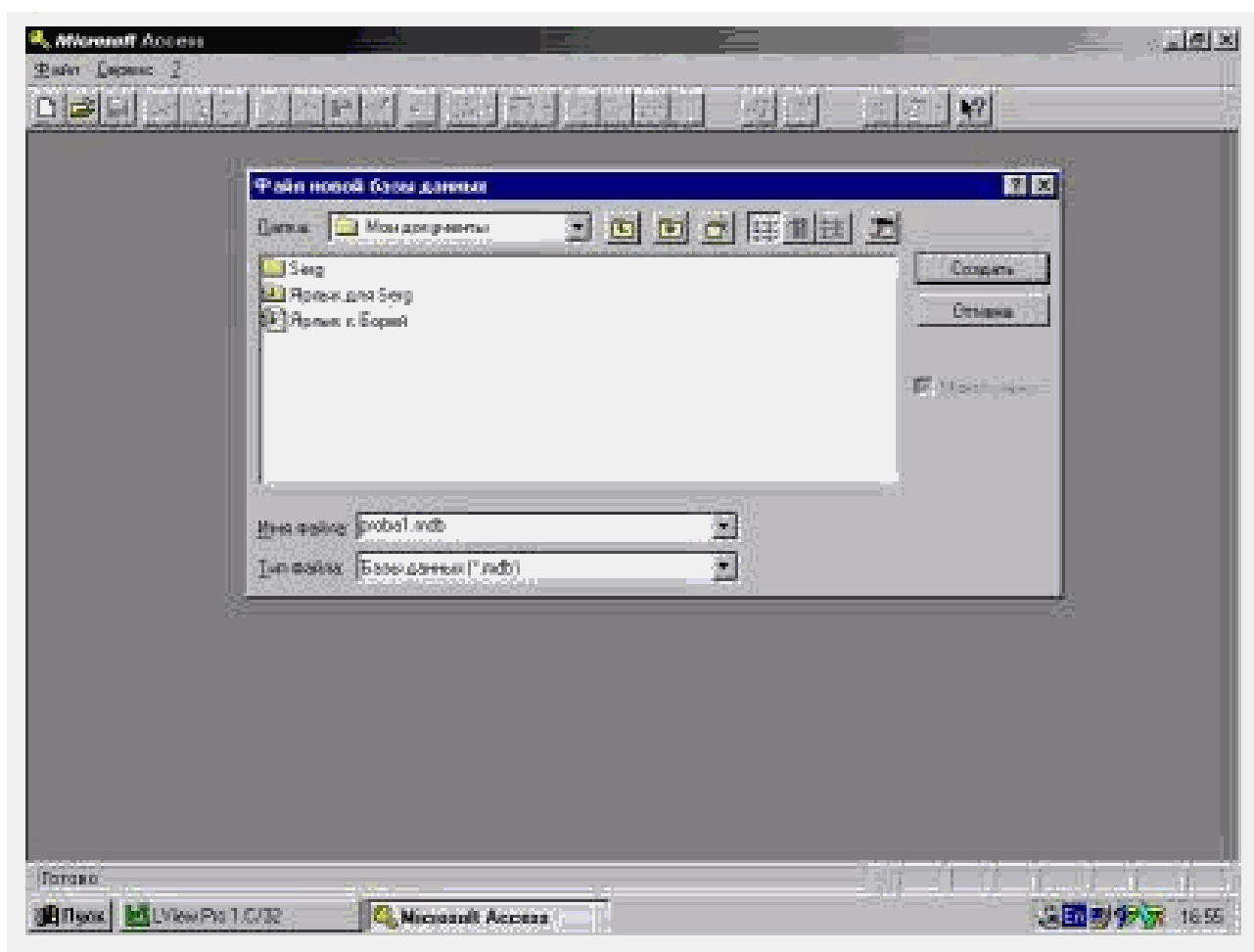


Рис. 3.3. Діалогове вікно *Файл новой базы данных*

3.2.7. Створення таблиць

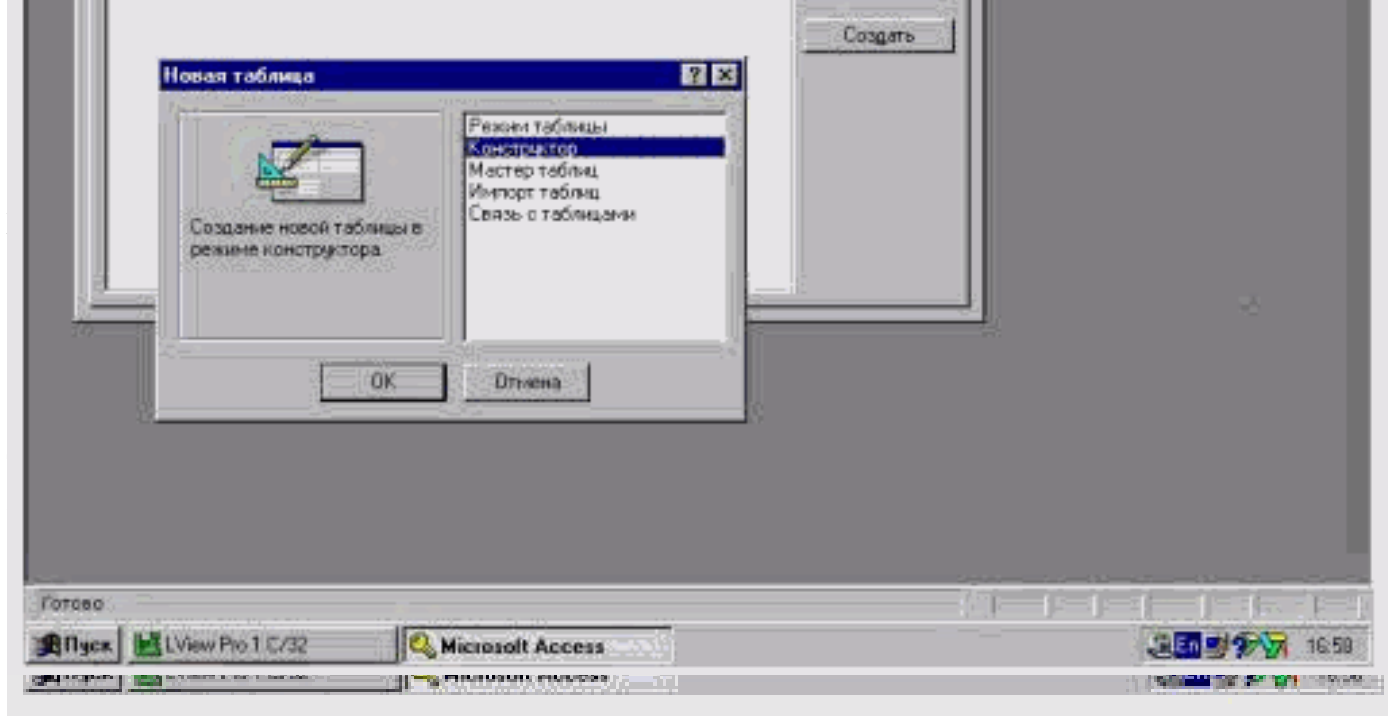


Рис. 3.4. Діалогове вікно *База данных*

Клацніть мишкою на кнопці **Создать (New)**. З'явиться вікно *Новая таблица (New Table)*, відтворене на рис. 3.5.

Рис. 3.5. Діалогове вікно *Новая таблица*

В правій частині діалогового вікна *Новая таблица (New Table)* виберіть потрібний пункт і клацніть мишкою на кнопці **ОК**.

В правому полі діалогового вікна *Новая таблица (New Table)* відображені п'ять пунктів, які дозволяють вибрати шлях створення таблиці:

Режим таблицы (Datasheet View). Відкривається таблиця, схожа на електронну таблицю, в яку можна вводити дані;

Конструктор (Design View). Якщо користувач хоче керувати процесом створення таблиці, то необхідно вибрати цю опцію і клацнути мишкою на кнопці **ОК**. Відкриється вікно конструктора, показане на рис. 3.6;

Мастер таблиц (Table Wizard). Запускає майстра, який ставить низку запитань і створює таблицю на підставі відповідей на них;

Импорт таблиц (Import Table). Запускає майстра, який дозволяє використовувати таблицю, що зберігається в другому форматі, як основу для нової таблиці;

Связь с таблицами (Link Table). Запускає майстра, який дозволяє встановлювати зв'язки з іншими таблицями.

3.2.8. Визначення імен полів

У вікні конструктора таблиць є три графи (рис. 3.6): **Имя поля (Field Name)**, **Тип данных (Data Type)** і **Описание (Description)**. В графу **Имя поля (Field Name)** введіть ім'я поля. Імена полів можуть мати довжину не більше від 64 символів і вміщувати букви, цифри, пропуски і знаки пунктуації.

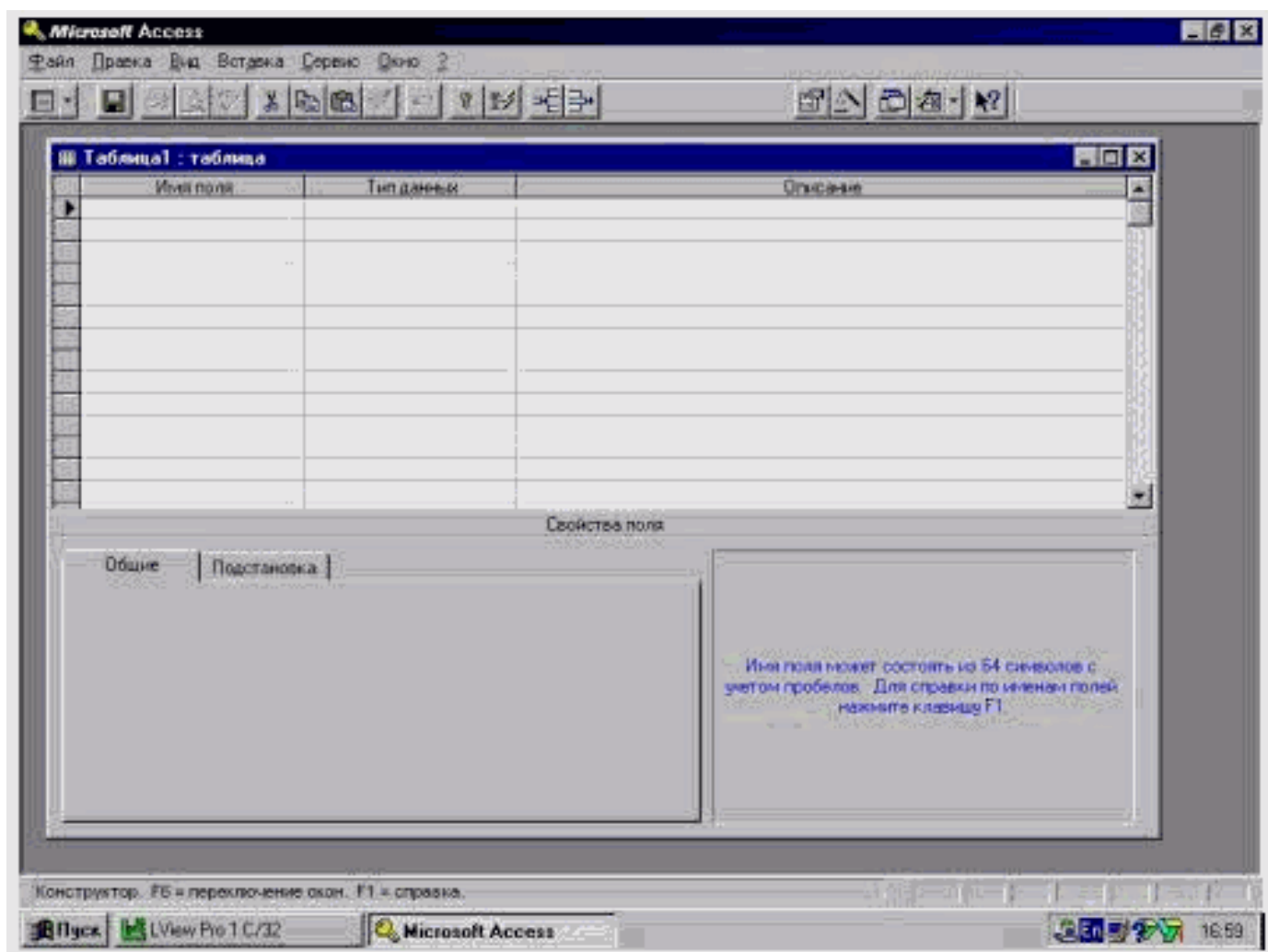


Рис. 3.6. Вікно *Конструктор таблиць*

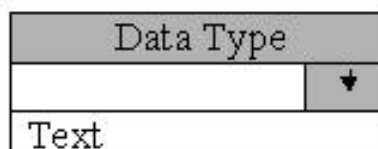
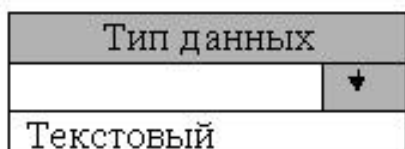
Примітка. Access дає змогу створювати імена полів більш описові, ніж старі БД для DOS, але бажано віддавати перевагу коротким іменам. Оскільки за замовчуванням імена полів використовуються як мітки полів у формах і як заголовки граф - у звітах, це може створити деякі незручності при розробці звіту, бо графи для розміщення полів будуть занадто широкими.

3.2.8.1. Вибір типів даних

Після введення імені поля натисніть клавішу <Tab>, щоб перемістити курсор в графу **Тип данных (Data Type)**. В цій графі необхідно вибрати тип даних, які будуть зберігатися в полі. За замовчуванням вважається, що дані є текстові; щоб змінити тип даних, клацніть мишкою на стрілці в правій частині клітинки і з вікна списку, що з'явилося (рис. 3.7), виберіть необхідний тип.

В Access є вісім типів даних і **Мастер подстановок (Lookup Wizard)**.

Текстовый (Text). Використовується для збереження будь-якої послідовності символів, включаючи букви, цифри, розділові знаки, пропуски і спеціальні символи. Текстові поля можуть вміщувати до



255 символів. За замовчуванням довжина текстового поля дорівнює 50 символам. Для зміни цієї величини клацніть мишкою на полі *Размер поля (Field Size)* вкладення *Общие (General)* в нижній частині вікна і введіть потрібне значення.

Рис. 3.7. Вікно *Список типа данных*

Поле МЕМО (Memo). Використовується для збереження тексту великого обсягу (наприклад, цілих параграфів). В полях **МЕМО** може вміщуватися до 32000 символів.

Числовой (Number). Застосовується для збереження числових значень. В числових полях звичайно вміщуються значення, які можуть брати участь у розрахунках. Якщо ці числа не будуть брати участь у розрахунках (наприклад, поштові індекси або номери телефонів), то краще використовувати для збереження текстові поля.

Дата/Время (Date/Time). Слугує для збереження дати і/або часу. Значення дати й часу можна зберігати і в текстових полях, однак використання цього поля має свої переваги. Наприклад, перевірка правильності дат і часу виконується автоматично (дата 13/42/97 і час 25:47 недопустимі), з вмістом цих полів можна робити обчислення.

Денежный (Currency). Використовується для збереження числових значень, які слугують виключно для указання грошових сум. Автоматично додається фіксована кількість цифр після десяткової коми, щоб запобігти помилкам округлення.

Счетчик (AutoNumber). Використовується для автоматичної нумерації запису, що додається. Наприклад, полю першого запису присвоюється значення 1, полю наступного запису - 2 тощо. Зверніть увагу на те, що після додавання запису значення цих полів змінити не можна.

Логический (Yes/No). Використовується для збереження логічних даних типу Да/Нет.

Поле объекта OLE (OLE Object). Вміщує об'єкт OLE. Об'єкт OLE - це документ іншого додатка Windows, що підтримує OLE (Object linking and embedding - зв'язування і впровадження об'єктів). Об'єкт OLE може вміщувати звук, відео, робочі таблиці або документи текстового процесора.

Мастер подстановок (Lookup Wizard). Використовується для задання значень, яких може набувати це поле. Наприклад, якщо необхідно додати в поточну таблицю поле для збереження назв областей,

можна відкрити іншу таблицю, яка вміщує назви всіх областей, і в *Мастере подстановок* вказати, що це поле може вмішувати значення, які зберігаються в цій таблиці.

Зауваження. Нехай необхідно з'ясувати, що використовувати - поле **МЕМО** чи поле **об'єкта OLE**.

Для збереження документа, створеного з допомогою текстового процесора, замість поля **МЕМО** можна використовувати поле **об'єкта OLE**. Поля **МЕМО** придатні для невеликого за обсягом тексту, що вміщує кілька абзаців. Поля **об'єкта OLE** більш придатні для збереження текстів великого обсягу, наприклад, багатосторінкових документів, створених в текстовому процесорі Windows. В текстовому процесорі *Mord for Windows* який підтримує **OLE**, можна використати команду **Правка** та її опцію **Копировать (Edit & Copy)** і команду **Правка** та її опцію **Вставить (Edit & Paste)** в *Access for Windows* для вставляння документів в поля типу **Поле Объекта OLE (OLE Object)**. Пам'ятайте, що при використанні *впровадження* (замість *зв'язування*) даних БД *Access* може швидко розростися до дуже великих розмірів і займати величезний обсяг дискової пам'яті.

3.2.8.2. Використання полів опису

Вибравши необхідний тип даних, натисніть клавішу <Tab> для переміщення курсора в графу **Описание (Description)**. В цю графу вводиться необов'язковий опис поля.

Після введення опису натисніть клавішу <Tab>. Курсор переміщується на наступний рядок (запис). Введіть імена полів, типи даних та (якщо потрібно) опис для всіх полів.

3.2.8.3. Використання властивостей полів

В нижній частині вікна конструктора таблиці розміщений розділ **Свойства поля (Field Properties)**, в якому визначаються властивості кожного поля таблиці. В процесі розробки таблиці на цей розділ можна не звертати уваги, оскільки *Access* не вимагає обов'язкових змін у ньому. Але після створення таблиці визначення властивостей полів значно спроститься контроль за введенням даних. Наприклад, у текстових полях можна змінювати розмір поля, який за замовчуванням дорівнює 50 символам, а також визначати інші характеристики, наприклад, формат даних поля або правила, які мають задовольняти дані. Властивості залежать від типу даних, вибраних для конкретного поля. В табл. 3.2 наведені властивості полів та їх призначення.

Таблиця 3.2

Властивості полів та їх призначення

Властивість	Призначення
Размер поля (Field Size)	Визначає максимальну довжину текстового поля. Для числових полів обмежує допустимі значення
Формат (Format)	Визначає формат відображення даних. (Його можна вибрати з уже створених форматів або створити самому)
Маска ввода (Input Mask)	Визначає шаблон, якому мають відповідати дані, що вводяться в поле
Дробная часть (Decimal Places)	Визначає кількість цифр після десяткової коми
Заголовок (Caption)	Визначає мітку, яка буде з'являтися у формах і звітах
Значение по умолчанию (Default Value)	Визначає встановлене за замовчуванням значення, яке автоматично з'являється в полі при додаванні нових записів.
Условие на значение (Validation Rule)	Визначає умови, яким мають відповідати дані, що вводяться в поле

Можна встановити будь-яку властивість поля, якщо виконати такі дії.

В режимі **Конструктор таблиць** клацніть мишкою на полі, властивості якого бажано встановити.

В розділі **Свойства поля (Field Properties)** виберіть вкладення *Общие (General)* і клацніть мишкою на потрібній властивості.

Якщо з правого боку від властивості з'явиться кнопка зі стрілкою, клацніть мишкою на цій кнопці, щоб відкрити список властивостей, і виберіть необхідне значення.

Зауваження. Можливо, користувач вирішить зменшити значення **Размер поля (Field Size)**, яке за замовчуванням дорівнює 50 символів, щоб зекономити дисковий простір. Але робити це зовсім не обов'язково, оскільки Access для збереження тексту використовує так звані *поля змінної довжини*, так, що кожне текстове поле використовує саме той обсяг пам'яті, який потрібен для збереження введених даних. Такий підхід відрізняється від підходу, що застосовується у БД, створених раніше.

Якщо ж потрібно вводити рядки тексту, що перевищують 50 символів, то збільшіть значення властивості **Размер поля (Field Size)**.

Одними з найбільш могутніх (і таких, що найчастіше використовується) властивостей полів є **Формат** і **Условие на значение**. Властивість поля *Формат (Format)* слугує для визначення зовнішнього вигляду даних у полі таблиці. (Вказана властивість використовується для будь-якого типу поля, крім *объекта OLE*.) Вибравши поле, клацніть мишкою на властивості **Формат (Format)** і/або виберіть стандартний формат, або створіть особистий. В Access є цілий набір стандартних форматів для числових полів, полів дати і часу та логічних полів. Для текстових полів і полів МЕМО можна використовувати символи, наведені в табл. 3.3.

Таблиця 3.3. Символи для форматів текстових полів і полів МЕМО

Символ	Значення
@	Текстовий символ або пропуск
&	Необов'язковий символ
<	Перетворює всі символи поля в малі
>	Перетворює всі символи поля у великі

Для числових або грошових полів можна скористатися символами, наведеними в табл. 3.4.

За допомогою властивості **Условие на значение** відбувається контроль за даними, що вводяться в поля таблиці. Ці умови (встановлюються користувачем) визначають, припустиме чи ні значення, введене в поле. Для введення умови спочатку виберіть потрібне поле. Потім у полі *Условие на значение (Validation Rule)* розділу **Свойства поля (Field Properties)** введіть умову. В полі *Сообщение об ошибке (Validation Text)* введіть текст повідомлення, яке буде відтворюватися при введенні даних, що не відповідають визначеній умові.

Таблиця 3.4

Условия на значение (Validation Rule) задаються як вирази. Вирази вміщують в собі стандартні математичні символи, а дані у виразах вміщують у символи #. Приклади деяких виразів, що використовуються в умовах, наведені в табл. 3.5.

Таблиця 3.5 Приклади виразів, що використовуються в умовах

Вираз	Значення
Between #15/4/97# and #31/12/97#	Дата, що вводиться, має бути між 15/4/97 і 31/12/97
<=#15/12/97#	Дата, що вводиться, має бути до 15/12/97
Like 112M####	Значення повинно мати вісім символів, які починаються з "112M"
>=4,50and<=12,00	Значення має бути між 4,50 і 12,00

Умови можуть включати вирази, що вміщують інші поля цієї таблиці. Наприклад, якщо є таблиця з грошовими полями "Основний заробіток" і "Заробіток за понаднормові" і необхідно, щоб значення в полі "Заробіток за понаднормові" не перевищувало вдвічі значення в полі "Основний заробіток", можна використати таку умову:

<=[Основний заробіток]*2

3.2.9. Додавання первинного ключа

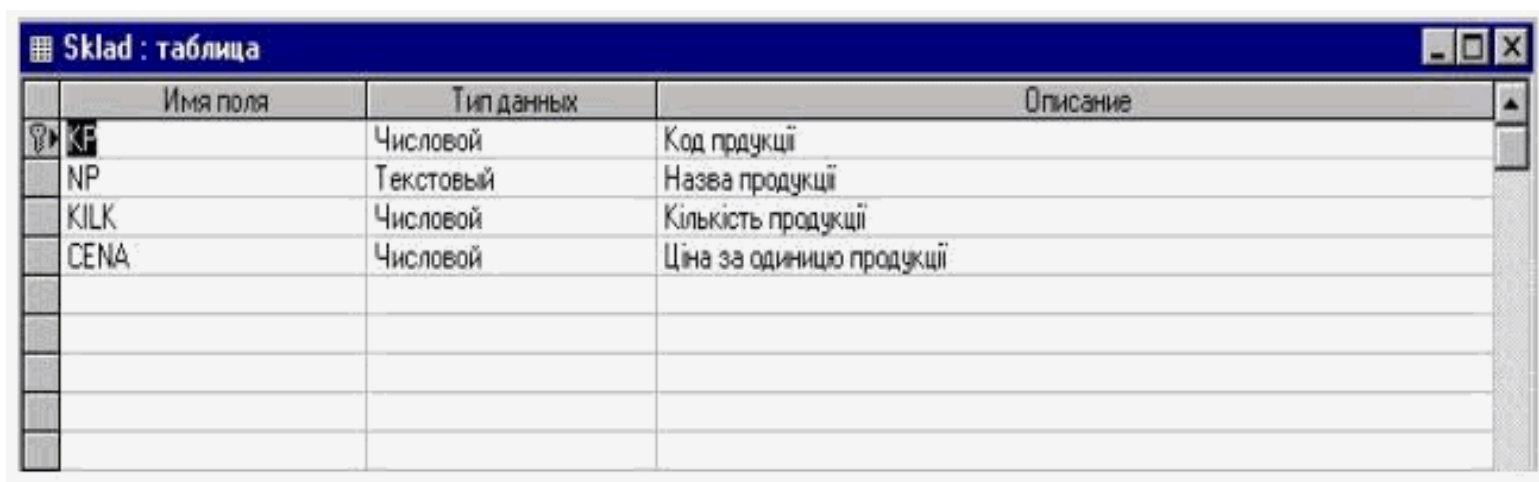
При розробці таблиці Access дозволяє визначити одне або більше полів як первинний ключ. *Первинний ключ* - це одне або кілька полів, за якими однозначно ідентифікуються записи в таблиці.

Первинні ключі досить корисні (настільки, що якщо спробувати зберегти таблицю без визначення первинного ключа, на екрані з'явиться діалогове вікно, яке вимагатиме його створення).

За замовчуванням первинні ключі в списку даних таблиці відтворюються першими. Для первинних ключів завжди необхідно використовувати дані, які ніколи не будуть повторюватися. Як первинний ключ не можна використовувати таке поле, як прізвище, оскільки неможливо буде мати два записи для одного прізвища. Необхідно також пам'ятати, що інколи є сенс використати як первинний ключ комбінацію полів. Наприклад, в таблиці, в якій ведеться облік щотижневого робочого часу, комбінація полів ідентифікаційного номера службовця і кінця тижня може бути використана для первинного ключа, оскільки жоден службовець не в змозі подати дві заявки про кількість відпрацьованих годин за один тиждень. Для встановлення первинного ключа необхідно виконати такі дії.

В режимі **Конструктор таблиць (Design View)** виберіть поле (або поля), яке бажано зробити первинним ключем. Щоб вибрати кілька полів, спочатку виберіть перше, клацнувши мишкою на кнопці ліворуч від назви поля, а потім, утримуючи натиснутою клавішу <Ctrl>, - решту полів.

Виконайте команду **Правка** та її опцію **Первичный ключ (Edit > Primary Key)** або клацніть мишкою на кнопці **Первичный ключ (Primary Key)** панелі інструментів (на ній зображено ключ). Поруч з вибраним полем відобразиться піктограма ключа (рис. 3.8).



	Имя поля	Тип данных	Описание
☛	KF	Числовой	Код прдукції
	NP	Текстовый	Назва продукції
	KILK	Числовой	Кількість продукції
	CENA	Числовой	Ціна за одиницю продукції

Рис. 3.8. Таблица з іменем Sklad, яка має первинний ключ

3.2.10. Реорганізація і видалення полів при розробці таблиць

Змінювати структуру таблиці, що розробляється, неважко. Для зміни імені поля, типу або опису необхідно клацнути мишкою в межах необхідного поля і внести потрібні зміни. Щоб видалити поле, спочатку необхідно його виділити, клацнувши мишкою на кнопці ліворуч від назви поля, потім виконати команду **Правка** та її опцію **Удалить строку (Edit > Delete Row)** або натиснути клавішу <Delete>.

Для вставляння нового поля між двома, що вже існують, необхідно виділити те поле, перед яким бажано вставити нове, після цього виконати команду **Вставка** та її опцію **Строка (Insert > Row)** або натиснути клавішу <Insert>.

Для переміщення поля в нове місце потрібно виділити поле, клацнути мишкою на ньому ще раз і, утримуючи натиснутою кнопку мишки, перетягнути рядок у потрібне місце.

3.2.11. Збереження таблиці

Після внесення всіх змін таблицю необхідно зберегти, виконавши для цього такі дії.

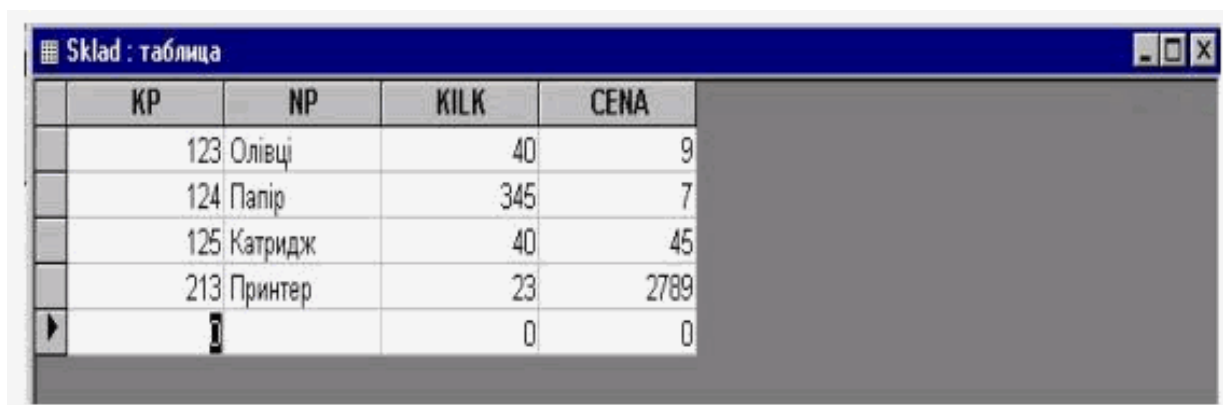
Виконати команду **Файл** та її опцію **Закрити (File ☒ Close)** або двічі клацнути мишкою на піктограмі таблиці в лівому верхньому кутку вікна. З'явиться діалогове вікно із запитом про те, чи необхідно вносити зміни в таблицю.

Клацнути мишкою на кнопці **Да (Yes)**, і з'явиться діалогове вікно *Сохранить как (Save As)*.

Ввести ім'я таблиці і клацнути мишкою на кнопці **ОК**.

3.2.12. Додавання даних в таблицю

Додавання записів в існуючу таблицю здійснюється в режимі таблиці. Щоб відкрити таблицю в цьому режимі, необхідно двічі клацнути мишкою на імені потрібної таблиці у вікні БД або виділити ім'я таблиці, а потім клацнути мишкою на кнопці **Открыть (Open)**. Вибрана таблиця відкриється в режимі таблиці, який використовується для додавання нових або редагування вже введених в таблицю даних. Для додавання нових даних необхідно просто ввести їх у поля робочої таблиці. У кінці кожної таблиці завжди є ще один порожній запис, помічений зірочкою. Коли починається введення даних у порожній запис, під ним з'явиться новий порожній запис. Тепер можна помістити курсор в будь-яке поле і ввести дані. Натискання клавиші <Tab> після закінчення введення переміщує курсор в наступне поле. Натискання клавиші <Tab> в кінці останнього поля дає змогу перейти до наступного порожнього запису (Рис. 3.9).



KP	NP	KILK	CENA
123	Олівці	40	9
124	Папір	345	7
125	Катридж	40	45
213	Принтер	23	2789
		0	0

Рис. 3.9. Таблиця з іменем Sklad, заповнена даними

3.2.13. Редагування і видалення даних

Для редагування даних необхідно активізувати потрібне поле таблиці і ввести або відредагувати дані. Під час редагування ліворуч від запису з'явиться зображення олівця як ознака того, що запис редагується. Як тільки користувач залишає рядок, зміни записуються в таблицю. В процесі редагування для переміщення по таблиці використовуються клавіші та їх комбінації, наведені в табл. 3.6.

Таблиця 3.6

Клавiші та їх комбiнації для перемiщення по таблиці

Клавiша	Призначення при перемiщенні
<←>	На один символ лiворуч
<→>	На один символ праворуч
<↑>	На попередній запис
<↓>	На наступний запис
<Tab>	На наступне поле
<Shift + Tab>	На попереднє поле
<Ctrl + Home>	На перше поле першого запису
<Ctrl + End>	На останнє поле останнього запису
<PgUp>	На одну сторiнку ввeрх
<PgDn>	На одну сторiнку вниз

В Access не можна редагувати поля, що перелічені нижче.

Поля счетчика. Значення в цих полях збільшуються і зберігаються автоматично. Якщо запис є, Access не дозволить його змінити.

Недоступные поля. Якщо опцію **Доступно (Enabled)** розділу **Властивість поля** встановлено у режим **Нет (No)** або опцію **Заблокировано (Locked)**, розділу **Властивість поля** встановлено в режим **Да (Yes)**, вміст поля редагувати не можна.

Поля в некоторых типах запросов. Неприпустиме редагування полів у запитах, що встановлюють зв'язки більш ніж з однією таблицею. Також не можна редагувати підсумкові значення в деяких запитах.

Вычисляемые поля. Оскільки поля, що розраховуються, не існують як окремі поля (вони розраховуються на підставі інших полів), їх редагувати не можна.

3.2.14. Друкування таблиці

Якщо таблиця перебуває в режимі таблиці, її можна роздрукувати.

Якщо роздруковуються всі записи у вікні БД, необхідно вибрати таблицю. Якщо потрібно роздрукувати вибрану групу записів, необхідно відкрити таблицю в режимі таблиці.

Для друкування вибраної групи записів необхідно їх виділити, клацнувши мишкою на першому записі і перетягнувши курсор мишки до останнього запису.

Виконати команду **Файл** та її опцію **Печать (File ● A Print)** або клацнути мишкою на кнопці **Печать (Print)** панелі інструментів.

Для друкування всіх записів клацнути мишкою на кнопці **OK**.

Для друкування вибраної групи записів клацнути мишкою на опції **Выбранные записи (Selected Record(s))**, а потім - на кнопці **ОК**.