МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНА МЕТАЛУРГІЙНА АКАДЕМІЯ УКРАЇНИ

В. В. Кузьменко, Г. Г. Швачич, Н. С. Романова

КОМП'ЮТЕРНІ ТЕХНОЛОГІЇ В ДОКУМЕНТОЗНАВСТВІ

РОЗДІЛ "ОРГАНІЗАЦІЯ ТА УПРАВЛІННЯ БАЗАМИ ДАНИХ"

Затверджено на засіданні Вченої ради академії
як конспект лекцій

Дніпропетровськ НМетАУ 2005

УДК 004(075.8)

Кузьменко В. В., Швачич Г. Г. Романова Н. С. Комп’ютерні технології в документознавстві. Розділ “Організація та управління базами даних”: Конспект лекцій. – Дніпропетровськ: НМетАУ, 2005. – 42с.

Викладені основні поняття які надають можливість конструювання баз даних. Як засіб розглянута оболонка Microsoft Access

Призначений для студентів спеціальності 6.020100 – документознавство та інформаційна діяльність.

Іл. 9. Бібліогр.: 9 найм.

Відповідальний за випуск Г. Г. Швачич, канд. техн. наук, проф.

Рецензенти: Б. І. Мороз, д-р техн. наук, проф. (Академія митної Служби України)

Т. М. Пашова, канд. техн. наук, доц. (Дніпропетровський Державний аграрний університет)

ВСТУП

Поняття бази даних й інформаційної системи

Віками людство накопичувало спеціальні знання, відомості про навколишній світ, іншими словами – збирало інформацію. З появою обчислювальної техніки значно спростилися способи її зберігання й обробки. З'явилося програмне забезпечення, що дає можливості систематизувати великі потоки інформації й створювати інформаційні системи. Метою інформаційних систем є зберігання й надання структурованої інформації.

При розгляді великих сукупностей об'єктів виникає необхідність згрупувати об'єкти, які мають однакові властивості. Такі сукупності об'єктів виділяються в окремі класи. Усередині кожного виділеного класу об'єкти впорядковуються за правилами класифікації, наприклад, за алфавітом, або й по деяких конкретних ознаках, наприклад, по кольорах, матеріалу та ін. Угруповання об'єктів значно полегшує пошук інформації в системі.

Інформаційні системи (ІС) можна умовно розділити на фактографічні й документальні.

У фактографічних ІС реєструються факти – конкретні значення властивостей (атрибутів) об'єктів реального миру. Основна ідея таких систем полягає в тім, що всі відомості про об'єкти (прізвища людей, назви предметів, числа, дати) повідомляються комп'ютеру в визначеному заздалегідь обумовленому форматі (наприклад, дата – у вигляді комбінації ДД. ММ. РРРР). Інформація, з якої працює фактографічна ІС, має чітку структуру, що дозволяє машині відрізняти одні дані від інших, наприклад, прізвище від посади людини, дату народження від зросту й т.п. Фактографічна система здатна давати однозначні відповіді на поставлені питання.

Документальні ІС обслуговують принципово інший клас задач, які не припускають однозначної відповіді на поставлене питання. Базу даних таких систем утворюють сукупності слабко структурованих об'єктів (текстових документів: статей, книг, рефератів й т.і, і графічних об'єктів). Бази даних документальних ІС постачені тим або іншим формалізованим апара-

том пошуку. Ціль системи, як правило, – видати список документів або об'єктів, що задовольняють сформульованим у запиті умовам.

Зазначена класифікація ІС у відомій мірі умовна, тому що сучасні фактографічні системи часто працюють із неструктурованими блоками інформації (текстами, графікою, звуком, відео), постаченими структурованими описувачами. Щоб пояснити, як фактографічна система може перетворитися в документальну (і навпаки), розглянемо умовний приклад.

Нехай об'єктом обробки фактографічної ІС є список вчених-економістів. Для кожного вченого існують наступні дані:

- ім'я;
- дата народження у форматі ДД. ММ. РРРР;
- національність;
- біографія (довільний текст);
- назви праць.

Маючи у своєму розпорядженні структуровані описувачі (ім'я, дата, стать), система може видати строгі відповіді на питання:

1) про будь-який ученого персонально;
2) про розподіл учених по даті народження й статі (у будь-яких сполученнях).

Якщо видалити зі списку структуровані описувачі, система перетвориться в документальну й, якщо не прийняти заходів, втратить здатність знаходити й класифікувати вчених. Помітимо, що ті ж дані в тій або іншій формі дублюються в біографії, наприклад: «Вільям Стаффорд народився в 1554 році в родині…»,«Іван Тихонович Ціпків мешкав з 1652 по 1726 рік…»і т.д. Але, на відміну від нас, комп'ютер не знає, що Стаффорд – іноземець, а Ціпків – росіянин, що «народитися» й «жити с..… по...» – синоніми й т.д.

У нашому курсі розглянемо фактографічні ІС, які використовуються буквально у всіх сферах людської діяльності, а практика роботи з ними буде представлена в середовищі системи управління базами даних (СУБД) Microsoft Access.

ТЕМА 1 Загальні відомості про бази даних

Основа будь-якої інформаційної системи, об'єкт її обробки – база даних, або сукупність відомостей про конкретні об'єкти реального світу в якій-небудь предметній галузі. Синонім терміна «база даних» – «банк даних». Для забезпечення швидкості і якості пошуку даних у базі, цей процес повинен бути автоматизований.

Комп'ютерну базу даних можна створити декількома способами:

- за допомогою алгоритмічних мов програмування, таких як Basic, Pascal, C++ і т. і. Даний спосіб застосовується для створення унікальних баз даних;
- за допомогою прикладного середовища. Наприклад, Visual Basic надає можливість створювати бази даних, що вимагають індивідуальних особливостей побудови;
- за допомогою спеціальних програмних середовищ, які називаються системами управління базами даних (СУБД).

У нинішній час існує кілька видів систем управління базами даних. Найбільш відомими й популярними СУБД є Access, FoxPro й Paradox.

1.1 Моделі баз даних

База даних може бути заснована на одній моделі або на сукупності декількох моделей. Будь-яку модель даних можна розглядати як об'єкт, що характеризується своїми властивостями (параметрами). Існують три основних типи моделей даних: реляційна, ієрархічна й мережева.

1.1.1 Реляційна модель

Термін «реляційний» (від латинського relatio – відношення) указує, насамперед, на те, що реляційна модель зберігання даних побудована на взаємовідношенні частин, котрі складають базу даних. У найпростішому випадку реляційна модель являє собою двомірний масив або двомірну таблицю, а при створенні складних інформаційних моделей – сукупність взаємозалежних таблиць. Кожен рядок такої таблиці називається записом, а стовпець – полем.

Реляційна модель даних має такі властивості:

- кожен елемент таблиці – один елемент даних;
- всі поля в таблиці є однорідними, тобто мають один тип;
- кожне поле має унікальне ім'я;
- однакові записи в таблиці відсутні;
- порядок записів у таблиці може бути довільним і може характеризуватися кількістю полів, типом даних.

1.1.2 Ієрархічна модель

Ієрархічна модель бази даних являє собою сукупність елементів, які розташовані у підпорядкуванні, від загального до часткового, й утворюють перевернене дерево (граф). Дана модель характеризується такими параметрами, як рівні, вузли і зв'язки. Принцип роботи моделі такий, що декілька вузлів більше низького рівня з'єднуються за допомогою зв'язку з одним вузлом більше високого рівня. Вузол – інформаційна модель елемента, який перебуває на даному рівні ієрархії.

Властивості ієрархічної моделі даних:

- кілька вузлів нижчого рівня зв'язані тільки з одним вузлом вищого рівня;
- ієрархічне дерево має тільки одну вершину (корінь), не підлеглу ніякій іншій вершині;
- кожен вузол має своє ім'я (ідентифікатор);
- існує тільки один шлях від кореневого запису до більш приватного запису даних.

1.1.3 Мережева модель

Мережева модель баз даних схожа на ієрархічну. Вона має ті ж основні складові (вузол, рівень, зв'язок), однак характер їх відносин принципово інший. У мережевій моделі прийнятий вільний зв'язок між елементами різних рівнів.

1.2 Реляційна структура баз даних

Будь-яку структуру даних можна перетворити в просту двомірну таблицю. Таке подання інформації є найбільш зручним як для користувача, так і для машини. Переважна більшість сучасних інформаційних систем працює з реляційними базами даних.

Основна ідея реляційного підходу полягає в тому, щоб представити довільну структуру даних у вигляді двомірної таблиці, тобто нормалізувати структуру.

Кожен запис у таблиці повинен мати первинний ключ (рис.1), тобто ідентифікатор (або адресу), Кожен ключ визначає тільки один запис. Кожен запис, а відповідно до цього й кожен ключ у базі даних є унікальними. Якщо повторюється ключ у будь-якій таблиці, то цілісність даних порушується.

Створення ключових полів в таблиці бази даних (рис.1)

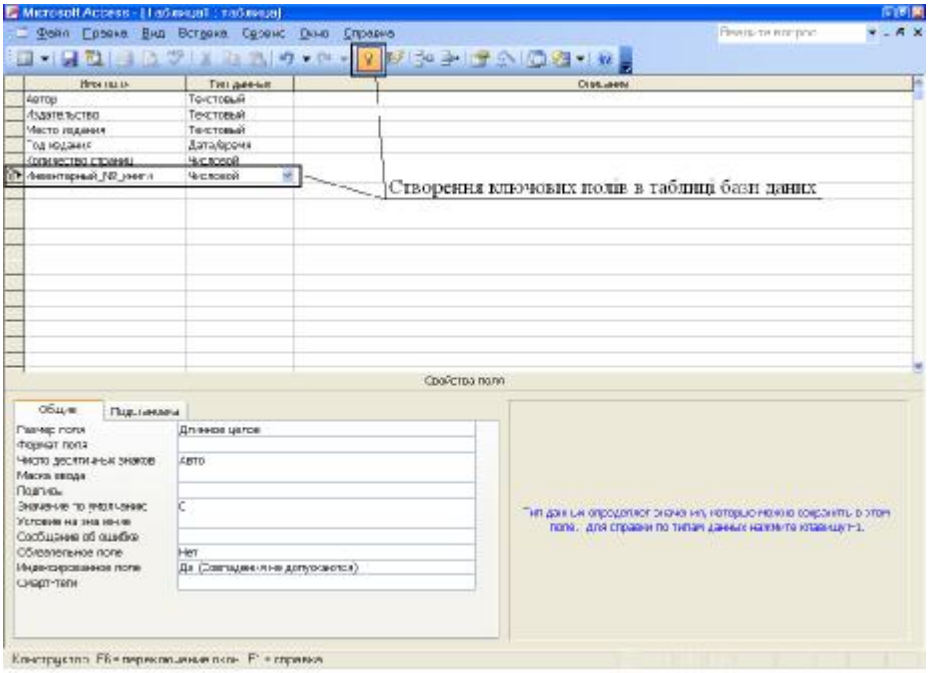

Рис.1

Первинний ключ повинен володіти двома властивостями:

1) однозначна ідентифікація запису: запис повинен визначатися одним ключем і навпаки, кожному ключу відповідає єдиний запис;
2) відсутність надмірності: ніяке поле не можна видалити із запису (із ключа), не порушуючи при цьому властивості однозначної ідентифікації.

Кожне значення первинного ключа в межах таблиці повинно бути унікальним. У противному випадку неможливо відрізнити один запис від іншої. Вказівка ключа – це єдиний спосіб відрізнити один запис від іншого. Як ключове поле використовують унікальні цифрові значення – код, табельні номери й т. і.

Крім первинного ключа, можуть використатися прості (або вторинні) ключі таблиці. Простих ключів може бути безліч. Вони використаються при упорядкуванні (індексуванні) таблиць.

1.3 Нормалізація в базі даних

Процес перетворення ієрархічної або мережевої структури даних у реляційну структуру, називається нормалізацією. Зовні ця операція дуже проста, але містить деякі особливості. Вони полягають у тому, що навіть у простих двомірних структурах доводиться правити склад полів. Наприклад, ми включимо у таблицю поле, значення якого не залежить від первинного ключа. У такому випадку з'являється можливість втрати інформації.

Важливіше інше: повторюючи багаторазово ті самі дані, ми не тільки переробимо масу зайвої роботи, але й неминуче помилимося. Варто прагнути до виключення з таблиці полів, які не зв'язані безпосередньо з первинним ключем таблиці. Для цього, крім оперативної таблиці, необхідно створити кілька довідкових таблиць. Оперативна таблиця змінюється часто, а довідкові – рідко, їх легко виправити, вносячись лише невеликі зміни.

При проектуванні таблиць рекомендуються три «золоті правила»:

1. необхідно усвідомити, що є первинний ключ таблиці (тобто переконатися, що двох записів з однаковим значенням ключа в таблиці бути не може);
2. якщо первинний ключ не проглядається, переконатися в правильності підбора складу полів;
3. якщо первинний ключ бездоганний, до нього можна дописувати будь-які атрибути, що залежать тільки від ключа.

Якщо при перегляді підготовленої бази даних у парі таблиць виявиться однойменне поле, яке не входить у первинний ключ ні однієї із цих таблиць, – це помилка нормалізації. Система не зможе контролювати погодженість значень таких полів.

1.4 Вірогідність інформації в базі даних

Оскільки первинне заповнення таблиць й введення їх у машину провадить людина, помилки в даних є не виключенням, а правилом, тому будь-яка інформаційна система повинна мати засіб для діагностики й виправлення помилок.

Порушення логічного взаємозв'язку – це логічні (семантичні) помилки, помилки змісту, які можуть бути виявлені апаратом формального логічного контролю, побудованим інформаційної системи. Стандартні засоби не можуть охопити всі можливі випадки логічного контролю, тому кожна конкретна інформаційна система має власні засоби додаткового («нестандартного») контролю. У сучасних системах управління базами даних (СУБД) є засоби підтримки цілісності даних. Крім того, в інформаційній системі (ІС) можна вказати умови, яким повинні задовольняти значення деяких полів (умови верифікації даних).

Набагато складніша справа з помилками в припустимих значеннях даних. Такі помилки умовно називаються арифметичними, хоча це не зовсім точно, тому що помилково може бути записане значення текстового даного: наприклад, Іванов І. П. замість Іванов А. П.

Існує ряд засобів для виявлення арифметичних помилок, але на користувальницькому рівні обмежуються простим візуальним контролем.

Тема 2 Основи розробки бази даних

Перш, ніж приступати до роботи із базою даних, необхідно вибрати модель подання даних. Вона повинна відповідати наступним вимогам:

- наочність подання інформації;
- простота вводу інформації;
- зручність пошуку й відбору інформації;
- можливість використання інформації, котра введена в іншу базу;
- можливість швидкого перенастроювання бази даних (додавання нових полів, нових записів, видалення записів).

При розробці бази даних виділимо наступні етапи роботи.

І етап Постановка проблеми

Визначається ціль створення бази даних, формуються завдання по її створенню, широко описуються складові частини майбутньої бази даних,

перераховуються види робіт, які передбачається здійснювати в цій базі (відбір, доповнення, зміна даних, друк або вивід звіту й т. і.).

II етап Аналіз об'єкта

Перш за все визначаються об'єкти майбутньої бази даних. Після розбивки бази даних на окремі об'єкти розглядаються властивості кожного з них, установлюється, якими параметрами описуються об'єкти. Відомості про об’єкти розташовуються у вигляді окремих записів і таблиць. Далі визначається тип даних кожного окремого запису (текстовий, числовий і т. і.).

III етап Синтез моделі

На цьому етапі по проведеному вище аналізу вибирається певна модель бази даних. У зв’язку з цим розглядаються достоїнства й недоліки реляційного й ієрархічного підходів. Вони зрівнюються з вимогами й завданнями в базі даних, яка створюється. Вибирається модель, що максимально забезпечить реалізацію поставлених завдань. Після вибору моделі графічно виконується її схема з вказівками зв'язків між таблицями або вузлами.

IV етап Способи подання інформації, програмний інструментарій

Після створення моделі, залежно від обраного програмного продукту, визначається форма подання інформації. У більшості СУБД дані можна представляти у двох видах:

- з використанням екранних форм;
- без використання екранних форм.

Форма – створений користувачем графічний інтерфейс для вводу даних у базу.

V етап Синтез комп'ютерної моделі об'єкта й технологія його створення

Після розгляду інструментальних можливостей обраного програмного продукту, варто приступити до реалізації БД на комп'ютері. Виділимо стадії, типові для будь-якої СУБД.

1. *Запуск СУБД, створення нового файлу бази даних або відкриття створеної раніше бази* (рис.2). У процесі виконання даної стадії необхідно запустити СУБД, створити новий файл (нову базу) або відкрити існуючу.

Запуск СУБД, створення нового файлу бази даних (рис.2)

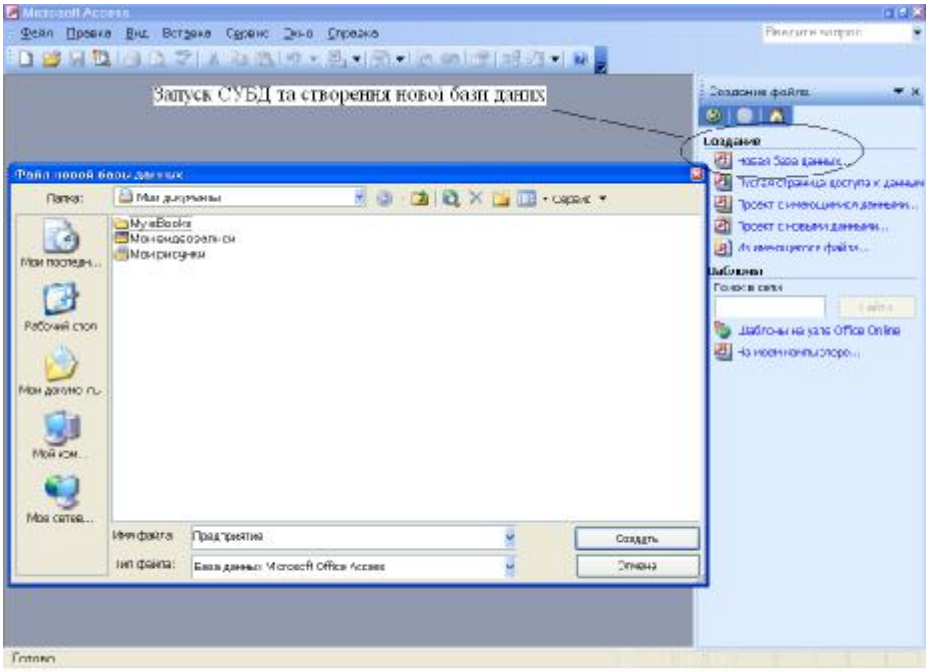

Рис.2

2. *Створення вихідної таблиці або таблиць* (рис.3). Створюючи вихідну таблицю, необхідно вказати ім'я й тип кожного поля. Імена полів не повинні повторюватися усередині однієї таблиці. У процесі роботи із БД можна доповнювати таблицю новими полями. Створену таблицю необхідно зберегти, давши їй ім'я, унікальне, в межах створюваної бази.

Створення вихідної таблиці (рис.3)

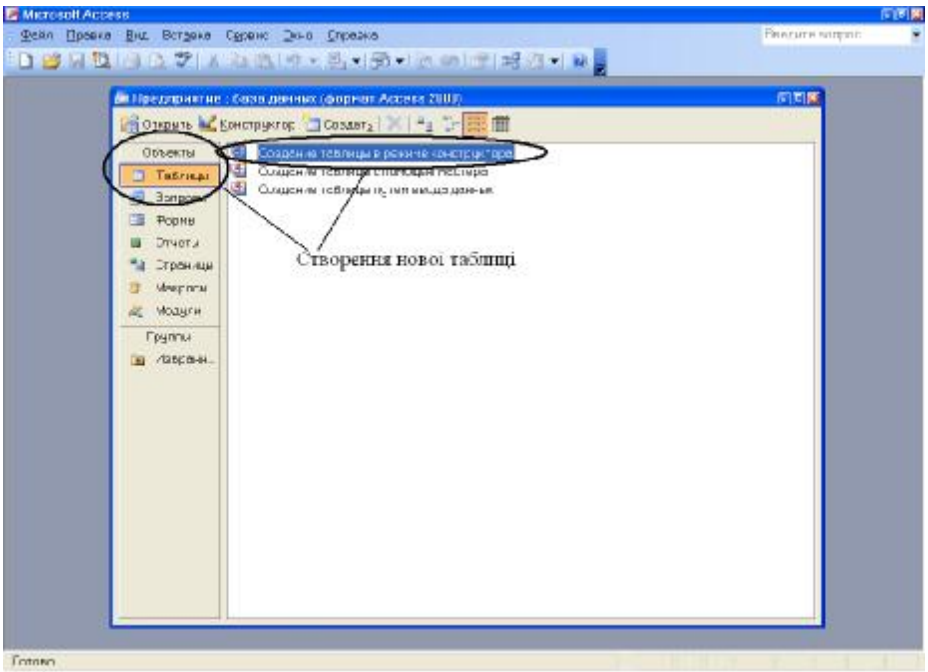

Рис.3

3. *Створення екранних форм* (рис.4). Спочатку необхідно вказати таблицю, на базі якої буде створюватися форма. Її можна створювати за допомогою «Майстра форм» або самостійно, вказавши, який вид вона повинна мати (наприклад, у вигляді стовпця або таблиці). При створенні форми можна вказувати не всі поля, які містить таблиця, а тільки деякі з них. Ім'я форми може збігатися з ім'ям таблиці, на базі якої вона створена. На основі однієї таблиці можна створити кілька форм, які можуть відрізнятися виглядом або кількістю використаних з даної таблиці полів. Після створення форму необхідно зберегти. Створену форму можна редагувати, змінюючи місце розташування, розміри й формат полів.

Створення екранних форм (рис.4)

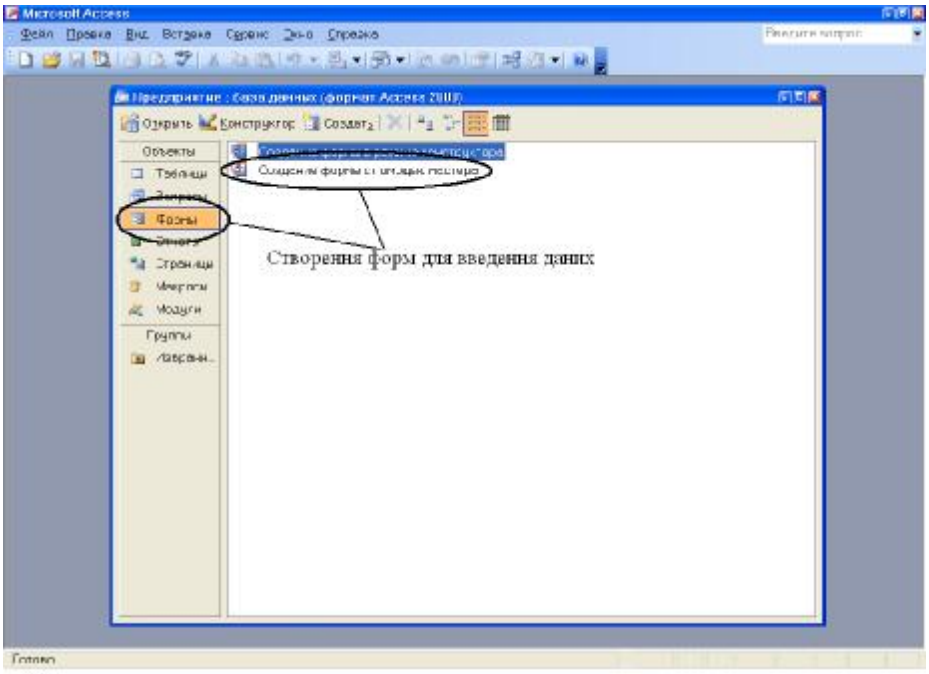

Рис.4

4. *Заповнення БД.* Процес заповнення БД може проводитися у двох видах: у вигляді таблиці й у вигляді екранної форми. Числові й текстові поля можна заповнювати у вигляді таблиці, а поля типу MEMO й OLE – у вигляді форми.

VI етап Робота зі створеною базою даних

Робота із БД містить у собі такі дії, як:

- пошук необхідних відомостей;
- сортування даних;
- відбір даних;

- вивід на друк;
- зміна й доповнення даних.

Розглянемо етапи створення й принципи роботи з базами даних на прикладі СУБД Microsoft Access.

ТЕМА 3 Робота з таблицями бази даних на прикладі СУБД Microsoft Access

Таблиці – фундаментальні об'єкти реляційної бази даних. У них зберігається основна частина даних додатка Microsoft Access. Окрема таблиця завжди зберігає інформацію з конкретної теми (наприклад, відомості про службовців компанії, або адреси замовників). Інформація в таблиці організується в рядки (записи) і стовпці (поля). Таблиці притаманні два компоненти: структура таблиці й дані таблиці.

3.1 Структура таблиці

Структура таблиці є визначенням таблиці. Вона повинна бути спроектована й створена перед вводом яких-небудь даних у таблицю. Структура таблиці зумовлює типи даних, які буде зберігати таблиця, а також правила або обмеження, асоційовані з вводом, зміною, видаленням даних. Структура таблиці доступна через вікно конструктора таблиць. Щоб відкрити таке вікно для існуючої таблиці, потрібно відкрити вкладку «**Таблиці»** вікна бази даних, вибрати таблицю й нажати кнопку «**Конструктор».**

Структура таблиці включає наступну інформацію:

- **ім'я таблиці.** Ім'я, по якому до таблиці можна звернутися у властивостях, методах та операторах SQL;
- **стовпці таблиці.** Категорії інформації, збереженої в таблиці. Кожен стовпець має ім'я й тип свого даного;
- **табличні й стовпцеві обмеження. О**бмеження цілісності, визначені на рівні таблиці або на рівні стовпця.

Вікно конструктора таблиць використовується як для визначення структури таблиці при її створенні, так і для наступної зміни структури таблиці.

3.2 Дані в таблиці

Дані таблиці - це інформація, що збережена в ній. Всі дані таблиці структуровані, зберігаються в рядках, кожен з яких містить порцію інформації. Дані - це та частина таблиці, до якої мають доступ користувачі додатка Microsoft Access. Дані таблиці можуть виводитися в елементах управління, наприклад, у формах і звітах, або надаватися в режимі таблиці.

3.3 Створення таблиці

Таблиці – це об'єкти, які зберігають більшу частину даних додатка Microsoft Access, тому підходити до їх проектування необхідно з усією старанністю. Правильна розробка таблиць включає багато аспектів, які гідні глибокого розгляду. Нижче наводяться деякі базові принципи розробки таблиць.

1. Необхідно уникати дублювання інформації. Для кожної категорії даних слід використовувати окрему таблицю. Наприклад, не варто зберігати опис відділів у таблиці, що зберігає інформацію про службовців. Процес, що дозволяє виключати дублювання даних у таблиці, називається нормалізацією. Нормалізація також надає можливість заощаджувати простір бази даних, допомагає запобігти помилок, які можуть виникати при наявному дубльовані інформації. В Microsoft Access процедури нормалізації допомагає виконати майстер аналізу.

2. В таблиці не слід зберігати значень, які можуть бути легко обчислені з існуючих даних. Наприклад, не потрібно зберігати суму всіх позицій товарного замовлення, тому що її можна обчислити за допомогою простої формули.

3. Для всіх полів необхідно вибирати відповідний тип даних. Це допоможе зменшити розміри бази даних і збільшить швидкість виконання операцій. При описі поля варто встановлювати для нього тип даних найменшого розміру, який дозволяє зберігати потрібні дані.

4. У кожну створювану таблицю необхідно включати стовпець або набір стовпців первинного ключа. Первинні ключі потрібні для встановлення відносин «один-до-багатьох» між таблицями. Крім того, багато баз даних підтримують обмеження по первинному ключу, використовуючи індекс, який може значно підвищувати швидкість пошуку й операції сор-

тування даних. У складовому ключі, що містить кілька полів, потрібно використати рівно стільки полів, скільки для нього необхідно.

Є спеціальні випадки, коли первинний ключ доцільніше не створювати. Наприклад, для деяких таблиць індекс, асоційований з первинним ключем, може неприйнятно знижувати продуктивність вводу й модифікації даних. Після того як проект таблиці готовий, можна приступати до її фізичного створення.

Створити таблицю можна двома способами. Для вводу нових даних можна створити порожню таблицю. Можна також створити таблицю, використовуючи вже існуючі дані з іншого джерела.

3.3.1 Створення нової порожньої таблиці

В Microsoft Access існує кілька способів створення нової таблиці:

1. Використання майстра баз даних для створення нової стандартної бази даних із числа наданих в Microsoft Access.

Створена за одну операцію база даних буде містити всі необхідні таблиці, форми й звіти. Майстер баз даних створює нову базу даних цілком. Його не можна використати для додавання нових таблиць, форм і звітів у вже існуючу базу даних.

2. Використання майстра таблиць для створення нової таблиці.

Майстер таблиць дозволяє вибрати поля для даної таблиці із числа визначених раніше таблиць.

3. Ввід даних безпосередньо в порожню таблицю в режимі таблиці.

При збереженні нової таблиці в Microsoft Access дані аналізуються, кожному полю привласнюється необхідний тип даних і формат.

4. Визначення всіх параметрів структури таблиці в режимі конструктора.
5. Імпорт у поточну базу даних структур таблиць і даних із зовнішнього джерела.
6. Створення в поточній базі даних таблиць, пов'язаних з таблицями зовнішнього джерела.

Незалежно від методу, застосованого для створення таблиці, завжди є можливість використати режим конструктора для подальшої зміни структури таблиці, наприклад для додавання нових полів, установки значень за замовчуванням або для створення масок вводу.

3.3.2 Створення таблиці в режимі конструктора таблиць

Режим конструктора таблиць дозволяє найбільше гнучко управляти всіма створюваними й уже наявними компонентами таблиці (рис.5). Щоб створити таблицю в режимі конструктора необхідно:

1. Перейти у вікно бази даних. Переключитися з іншого вікна у вікно бази даних можна, нажавши клавішу «**F11**».
2. Вибравши вкладку «**Таблиці»,** натиснути кнопку «**Створити».**

Створення таблиць в режимі конструктор (рис.5)

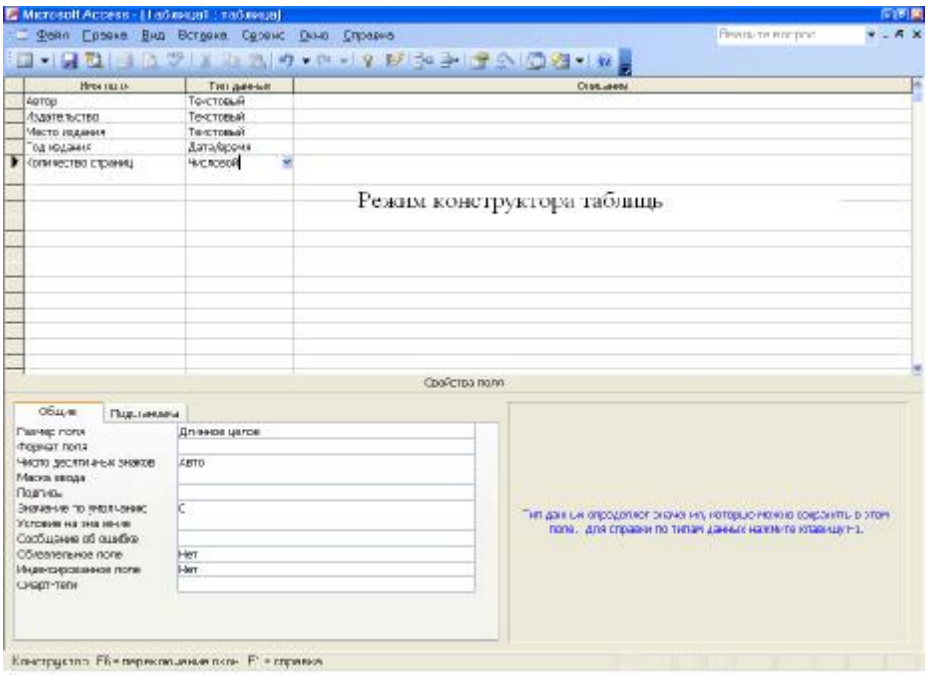

Рис.5

Якщо дана таблиця є зв'язаною, то додавати в неї нове поле в поточній базі даних неможливо. Якщо зв'язана таблиця є таблицею Microsoft Access, то для додавання поля необхідно відкрити вихідну базу даних. Якщо зв'язана таблиця є таблицею іншого додатка, то для додавання поля необхідно відкрити вихідний файл у цьому додатку.

3. У вікні «**Нова таблиця»** вибрати елемент «**Конструктор».**
4. Визначити в таблиці кожне поле.

Поля таблиці містять дані, що представляють порції інформації. Користувач має можливість визначати формат відображення даних, указувати значення за умовчанням, прискорювати операції пошуку й сортування, за-

даючи значення властивостей полів у розділі «**Властивості поля»** у режимі конструктора таблиці.

В Microsoft Access властивості полів необхідні при перегляді або зміні даних користувачем. Наприклад, задані користувачем значення властивостей «**Формат поля», «Маска вводу»** й «**Підпис»** визначають вид таблиці в базі даних й запиту з неї. Елементи управління в нових формах й звітах, приєднані до полів таблиці. Вони успадковують головні властивості полів базової таблиці за умовчанням. Додаток Microsoft Access буде автоматично перевіряти виконання цих умов при кожному додаванні або зміні даних у таблицю.

Для додавання поля в кінець структури таблиці потрібно вибрати останній рядок структури. Для вставки поля в середину структури варто вибрати рядок, перед яким потрібно додати нове поле, і нажати кнопку «**Додати рядок»** на панелі інструментів. У стовпець «**Ім'я стовпця»** ввести ім'я поля; у стовпці «**Тип даних»** вибрати необхідний тип даних. У стовпці «**Опис»** можна ввести необов'язковий короткий опис поля. Текст опису буде виводиться в рядку стану, при додаванні даних у поле, а також буде включений в опис об'єкта таблиці. При необхідності можна задати значення властивостей поля в бланку властивостей у нижній частині вікна.

5. Призначити ключові поля таблиці.

Наявність у таблиці ключових полів не обов'язково. Якщо вони не були визначені, то при збереженні таблиці видається питання, потрібно чи їх створювати.

Для збереження таблиці нажати кнопку «**Зберегти»** на панелі інструментів, ввівши припустиме ім'я таблиці.

Вибір типів даних для полів таблиць

Тип даних для полів таблиць слід вибирати в меню **«Тип даних»** (рис.6).

При виборі типу даних необхідно враховувати наступні умови:

1. Відповідність значень, які повинні зберігатися в полі обраному типу даних.

Наприклад, не можна зберігати текст у полі, що відповідає числовому типу даних, і недоцільно зберігати числові дані в текстовому полі.

2. Збіг виділеного розміру поля і необхідного для зберігання в ньому даних.

3. Операції, які повинні провадитися з даними в полі.

Наприклад, підсумовування, сортування даних або індексування поля. Сортувати й індексувати гіперпосилання й об'єкти OLE в полі MEMO неможливо.

4. Використання групування записів у запитах або звітах.

Поля MEMO, гіперпосилання й об'єкти OLE використати для групування записів не можна.

5. Спосіб відсортування даних у полі.

Вибір типів даних для полів таблиць (рис.6)

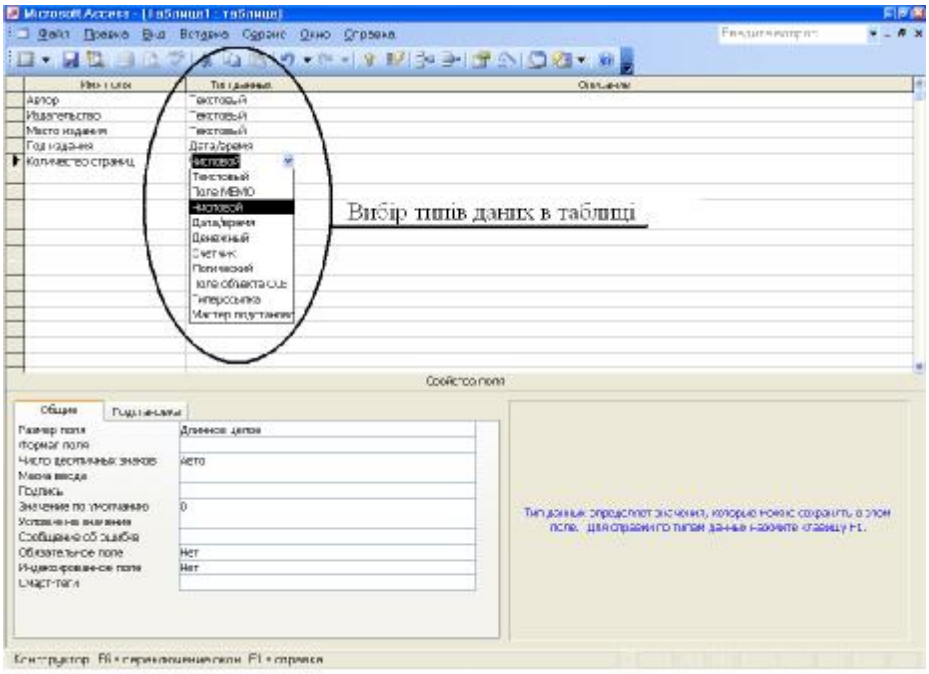

Рис.6

Числа в текстових полях сортуються як строкові значення (1, 10, 100, 2, 20, 200 і т. і.), а не як числові значення. Для сортування чисел як числових значень необхідно використати числові поля або поля, що мають грошовий формат. Формати дат неможливо належним чином відсортувати, якщо вони введені в текстове поле. Для забезпечення сортування дат і часів варто використовувати поле типу «Дата/Час».

Розглянемо типи даних, припустимі для зберігання й обробки в полях додатка Microsoft Access.

Текстовий тип даних

Поле текстового типу містить текст або комбінацію тексту й чисел, наприклад, адреси або числові значення, що не вимагають обчислень. У цій якості можуть бути розглянуті номери телефонів або поштові індекси. Розмір текстового поля складає до 255 символів. У текстовому форматі зберігаються тільки введені в поле символи. Для управління максимальним числом символів, які вводять у поле, необхідно визначити ознаку **«Розмір поля»**.

Тип даних МЕМО

Поле МЕМО містить довгий текст або числа, наприклад, коментарі або опис. Його розмір 64 000 символів. У цьому полі дані не можуть бути індексовані або відсортовані. Для зберігання форматованого тексту або довгих документів, замість поля МЕМО, доцільно створити поле об'єкта OLE.

Числовий тип даних

У числовому полі зберігаються дані, які використовуються для математичних обчислень, за винятком обчислень, що включають грошові операції. Для них використовується поле грошового типу. Розмір числового поля від 0 до 255 символів.

Дані типу «Дата/Час»

Поле типу Дата/Час зберігає значення дат і часу. Воно забезпечує правильне сортування даних такого типу. Всі зміни, внесені у формати дат і часу у вікні «**Мова й стандарти»** Панелі управління Windows, будуть автоматично відбиті в полях типу Дата/Час. Розмір поля 8 байт.

Грошові значення даних

Тип поля **«Грошовий»** зберігає значення валют. У цьому полі запобігають округлення під час обчислень, передбачається до 15 символів у цілій частині числа й 4 у дробовій. Розмір поля 8 байт.

Поле лічильник

У цьому полі здійснюється автоматична вставка послідовних (що відрізняються на 1) або випадкових чисел при додаванні записів. Для створення зростаючого лічильника варто залишити всі настроювання й властивості в нижній частині вікна за замовчуванням. При створенні лічильника

випадкових чисел потрібно встановити значення «**Випадкові**». Розмір поля 4 байти. Для кодів реплікації 16 байт.

Тип даних «Логічні»

Логічний тип поля містить тільки одне або два значення, такі як «**Так/Ні**», «**Істина/Неправда»**. Розмір поля 1 байт.

Об'єкти OLE

Поле OLE містить об'єкти (наприклад, документи Microsoft Word, електронні таблиці Microsoft Excel, малюнки, звуки й інші дані), створені в інших програмах, що використовують протокол OLE. Об'єкти можуть бути зв'язаними або впровадженими в таблицю Microsoft Access. Для відображення об'єкта OLE у формі або звіті необхідно використати елемент управління. Розмір поля до 1 гігабайта.

Дані типу «Гіперпосилання»

Поле, у якому зберігаються гіперпосилання, має формат UNC (стандартний формат шляху файлу із включенням адреси мережного сервера), або URL (адреса об'єкта у внутрішній мережі із вказівкою типу протоколу доступу). Розмір поля до 64 000 символів.

Поле «Майстер підстановок»

Поле «**Майстер підстановок»** дозволяє вибрати значення з інших таблиць або зі списку значень, використовуючи поле зі списком. При виборі необхідного параметра в списку типів даних для їх визначення завантажується майстер підстановок. Розмір поля **«Майстер підстановок»** такий самий, як і розмір ключового поля.

Числові, грошові й логічні поля, а також Дата/Час забезпечують стандартні формати відображення типів даних. Для вибору формату кожного типу даних варто визначити властивість «**Формат».** Для всіх даних, крім об'єктів OLE, можна також створити користувальницький формат відображення даних.

Розмір поля

Розмір поля визначає максимальний розмір даних, які зберігаються в полях типу **«Текстове»**, **«Числове»** або **«Лічильник»**.

Якщо тип даних **«Текстовий»**, значенням розміру даного поля є ціле число в діапазоні від 0 до 255. За замовчуванням задається розмір 50. Для

поля **«Лічильник»** припустимими значеннями розміру поля є **«Довге ціле»** або **«Код реплікації»**. Для числового типу даних припустимі значення розміру представлені в таблиці 1.

Припустимі значення розміру числового поля

Таблиця 1

Значення	Опис	Дробова частина	Розмір
Байт	Числа від 0 до 255	Відсутня	1 байт
Ціле	Числа від - 32 768 до 32 767	Відсутня	2 байти
Довге ціле	Значення за замовчуванням Числа від -2 147 483 648 до 2 147 483 647	Відсутня	4 байти
Із плаваючою крапкою (4 байти)	Числа від -3.402823E38 до -1.401298E-45 для негативних значень і від 1.401298E-45 до 1.402823E38 для позитивних.	7 знаків	4 байти
Із плаваючою крапкою (8 байт)	Числа від -1.79769313486232E308 до -4.94065645841247E для негативних значень і від 1.79769313486231E308 до 4.94065645841247E-324 для позитивних.	15 знаків	8 байт
Код реплікації	Глобальний унікальний ідентифікатор (GUID) при реплікації об'єктів даних	Не визначено	16 байт

Користувач має можливість указати стандартні розміри текстових і числових полів у вкладці **«Таблиці/запити»** у групі **«Розміри полів за замовчуванням»** (у діалоговому вікні **«Параметри»,** що відкривається командою **«Параметри»** у меню **«Сервіс»).**

Рекомендується задавати мінімально припустиме значення властивості «**Розмір поля»,** оскільки обробка даних меншого розміру виконується швидше й вимагає меншого обсягу пам'яті.

Перетворення більшого значення властивості «**Розмір поля»** на менше, у таблиці, що вже містить дані, може привести до втрати даних. Наприклад, при зменшенні розміру текстового поля від 255 до 50 всі значення, довжина яких перевищує 50 символів, будуть загублені. Дані в числовому полі, які виходять за межі діапазону, що відповідає новому розміру поля, округляються або заміняються порожніми значеннями. Наприклад, при заміні значення «Із плаваючою крапкою (4 байти)» на «Ціле» дробові числа будуть округлені до найближчого цілого числа, а значення поза діапазоном від -32 768 до 32 767 будуть перетворені в порожні значення.

*Скасувати зміни даних, що відбулися при модифікації властивості «**Розмір поля»,** після його збереження в режимі конструктора таблиці неможливо*.

Для полів, у яких планується зберігати числові значення від одного до чотирьох знаків у дробовій частині, рекомендується використовувати грошовий тип даних. При обробці числових значень із полів типу **«Із плаваючою крапкою (4 байти)»** й **«Із плаваючою крапкою (8 байт)»** застосовуються обчислення із плаваючою крапкою. При обробці числових значень із грошових полів використаються більше швидкі обчислення з фіксованою крапкою.

Поле типу «Лічильник»

Для створення полів, у які при додаванні запису автоматично вводиться число, в Microsoft Access існує тип даних **«Лічильник»**. При цьому, вже створений для запису номер не може бути вилучений або змінений. У поле лічильника можуть бути використані три типи чисел: послідовно зростаючі на одиницю, випадкові числа, коди реплікації (які також називаються GUID – глобальні унікальні ідентифікатори).

Найбільше часто використається лічильник послідовно зростаючих чисел. Такий тип лічильника зручно використати в ключовому полі табли-

ці. Лічильник випадкових чисел створює унікальний номер для кожного запису в таблиці.

Поле лічильника й реплікація

Код реплікації використовується при реплікації бази даних, для створення унікальних ідентифікаторів, які забезпечують синхронізацію реплік. При реплікації бази даних необхідно визначити підходящий розмір для поля типу **«Лічильник»**, використовуваного як ключове поле таблиці.

При використанні поля типу **«Лічильник»,** як ключового поля для таблиць у реплікованій базі даних, його розмір можна встановити або як **«Довге ціле»**, або **«Код реплікації»**. Якщо між операціями синхронізації реплік додається менше 100 записів, то з метою економії дискового простору як розмір поля варто встановити значення **«Довге ціле»**. Якщо між операціями синхронізації додається більше 100 записів, то з метою запобігання повторення значень у ключових полях у різних репліках необхідно використати значення **«Код реплікації»**.

Варто мати на увазі, що в полі типу **«Лічильник»** з розміром **«Код реплікації»** генеруються 128-байтові значення, які вимагають достатнього місця на жорсткому диску.

Властивість «Формат поля»

Властивість **«Формат поля»** дозволяє виставити формати виводу тексту, чисел, дат і значень часу на екран і на печатку. Наприклад, для поля **Ціна** розумно вказати у властивості **«Формат поля» Грошовий** формат. Установити для його властивості – **Число десяткових знаків –** значення 2 або **«Авто»**. У цьому випадку введене в поле значення 4321,678 буде відображатися як 4 321,68 гр.

Припустимо використання як вбудованих, так і спеціальних форматів, створених за допомогою символів форматування. Для елементів управління значення властивості **«Формат поля»** задається у вікні властивостей. Для поля в таблиці або запиті значення даної властивості задається в режимі конструктора таблиці (у розділі властивостей поля) або у вікні запиту (у вікні властивостей поля). Формати можна вибирати зі списку вбудованих форматів для полів, що мають числовий, грошовий, логічний типи даних, а також типи даних лічильника та дати/часу. Для будь-яких

типів даних полів, відмінних від об'єктів OLE, є можливість створення власних спеціальних форматів. Крім того, значення даної властивості можна задати в макросі або в програмі.

Властивість **«Формат поля»** визначає тільки спосіб відображення даних, не надаючи впливу на спосіб їх збереження. В Microsoft Access визначені стандартні формати для полів з типами даних **«Числовий»**, **«Дата/Час»**, **«Логічний»**, **«Текстовий»** і **«Поле МЕМО»**. У якості стандартних використаються національні формати, обирані у вікні **«Мова й стандарти»** панелі управління Windows.

Набір форматів визначається настроюваннями для конкретної країни. Наприклад, якщо на вкладці **«Мова й стандарти»** указати «Англійський» (США), то число 1234.56 у грошовому форматі буде виглядати як $1,234.56. Але якщо вказати на цій вкладці вказати «Український» , то це число буде виглядати так: 1 234,56 гр. Настроювання **«Формат поля»,** задано в режимі конструктора таблиці, використовується для відображення даних у режимі таблиці. Це ж настроювання застосовується при створенні пов'язаних із обраним полем нових елементів управління у формі або звіті.

У таблиці 2 перераховані символи, використовувані при визначенні спеціальних форматів для будь-якого типу даних.

Символи спеціальних форматів

Таблиця 2

Символ	Значення
(Пробіл)	Виводить пробіл як символьну константу
«ABC»	Всі символи усередині лапок вважаються символьними константами
!	Вирівнює символи по лівому краю
*	Заповнює доступний порожній простір вказаним символом
\	Виводить символ як символьну константу. Для цієї ж мети можна використати лапки
[кольори]	Задає колір, назву якого зазначено в дужках. Припустимі імена квітів: Чорний, Синій, Зелений, Бірюзовий, Червоний, Ліловий, Жовтий, Білий

Не дозволяється змішувати в одному форматі спеціальні символи, призначені для визначення числових форматів, а також форматів дати/часу й текстових форматів. Якщо для поля обрано конкретне значення властивості **«Маска вводу»**, а у властивості **«Формат поля»** задається інше форматування тих же даних, то пріоритет мають настроювання, що задають у властивості **«Формат поля»**, а значення **«Маска вводу»** ігнорується. У властивості **«Формат поля»** задаються різні настроювання для різних типів даних. Нижче приводиться опис конкретних настроювань.

Властивість «Формат поля» для дати/часу

Властивість «**Формат поля»** дозволяє вказати використання вбудованих або спеціальних числових форматів для полів дати/часу. У таблиці 3 приводяться вбудовані значення властивості «**Формат поля»** для полів дати/часу.

Формати полів для даних типу дата/час

Таблиця 3

Значення	**Опис**
Повний формат дати	(Значення за умовчанням). Якщо значення містить тільки дату, то час не відображається, та навпаки. Даний формат є комбінацією двох: «Короткий формат дати» й «Довгий формат часу». Приклади: 01.11.95 1:07:19 й 23.01.96 23:01:04
Довгий формат дати	Збігається з настроюванням **«Повний формат»**, що задається у вікні **«Мова й стандарти»** Панелі керування Windows. Приклад: 1 Червень 1995 р.
Середній формат дати	Приклад: 03-апр-95
Короткий формат	Збігається з настроюванням «Короткий формат дати», що задається у вікні **«Мова й стандарти»** Панелі управління Windows. Приклад: 11.06.95. Значення короткого формату дати припускають, що дати з діапазону 01.01.00 й 31.12.29 ставляться до двадцять першого століття. Передбачається також, що дати із проміжку 01.01.30 й 31.12.99 ставляться до двадцятого століття

Значення	Опис
Довгий формат часу	Збігається з форматом часу, що задається у вікні **«Мова й стандарти»** на вкладці **«Час»** панелі керування Windows. Приклад: 20:58:10
Середній формат часу	Приклад: 05:34
Короткий формат часу	Приклад: 17:34

Існують спеціальні формати дати й часу. Вони виводяться у відповідності зі значеннями, установленими у вікні **«Мова й стандарти»** Панелі управління **Windows.** Спеціальні формати, що суперечать настроюванням вікна «**Мова й стандарти»,** ігноруються.

Властивість Формат поля для числових і грошових полів

Властивість **«Формат поля»** дозволяє вказати використання вбудованих і спеціальних числових форматів для числових і грошових типів даних. У таблиці 4 наводяться вбудовані значення властивості **«Формат поля»** для числових полів.

Формати числових полів

Таблиця 4

Значення	Опис
Основний	Числа відображаються так, як вони були введені
Грошовий	Використовуються роздільники груп розрядів; негативні числа виводяться в круглих скобках; властивість **«Число десяткових знаків»** за умовчанням одержує значення 2
Фіксований	Виводиться, принаймні один розряд; властивість **«Число десяткових знаків»** за умовчанням одержує значення 2
З роздільниками	Числа виводяться з роздільниками груп та розрядів
Процентний	Значення множиться на 100; додається символ відсотків (%); властивість Число десяткових знаків за умовчанням одержує значення 2
Експонентний	Числа виводяться в експонентній (наукової) нотації

Спеціальні числові формати (таблиця 5) можуть містити в собі від одного до чотирьох розділів, відділених друг від друга крапкою з комою (;). Кожен формат містить специфікацію для різних розділів (типів) числових даних.

Таблиця спеціальних числових форматів

Таблиця 5

Розділ	Опис
Перший	Формат позитивних чисел
Другий	Формат негативних чисел
Третій	Формат нульових значень
Четвертий	Формат порожніх значень

Властивість «Формат поля» для текстових і МЕМО полів

Властивість **«Формат поля»** дозволяє створювати спеціальні формати для текстових і МЕМО полів (таблиця 6) за допомогою спеціальних символів. Для цього використовуються наступні символи:

Таблиця властивостей текстових і МЕМО полів

Таблиця 6

Символ	Опис
@	Обов'язковий текстовий символ або пробіл
&	Необов'язковий текстовий символ
<	Перетворить всі символи в рядкові
>	Перетворить всі символи в прописні

Спеціальні формати для текстових полів і полів МЕМО (таблиця 7) можуть включати один або два розділи, поділених крапкою з комою (;). Ці розділи описують специфікації формату різних порцій даного поля.

Таблиця спеціальних форматів текстових і МЕМО полів

Таблиця 7

Розділ	Опис
Перший	Формат відображення тексту
Другий	Формат відображення рядків нульової довжини й порожніх значень

Формат поля й маска вводу даних

В Microsoft Access до схожих результатів приводить зміна двох властивостей полів: властивість **«Формат поля»** і властивість **«Маска вводу».** Властивість **«Формат поля»** використовується для відображення даних у постійному форматі. Наприклад, якщо властивість **«Формат поля»** для поля типу **«Дата/Час»** установлений як **Середній** формат дати, то всі введені дані будуть відображатися в наступному форматі: 12-янв-97. Якщо ж користувач бази даних введе число у вигляді 12.01.97 (або в іншому визначеному виді), то при збереженні запису формат дати буде перетворений у **Середній** формат дати.

При установці властивості **«Формат поля»** змінюється тільки відображення значення, однак, дана властивість ніяк не впливає на зберігання значення в таблиці. Зміни у форматі відображення застосовуються тільки після збереження введених даних, до цього моменту визначити, у якому форматі були введені дані в поле, неможливо. Якщо ж вводом даних необхідно керувати, як додаток до формату відображення даних або замість нього використається маска вводу. Якщо потрібно, щоб дані відображалися так, як вони були введені, властивість **«Формат поля»** взагалі не встановлюється.

Маска вводу забезпечує відповідність даних у визначеному форматі, а також у заданому типу значень, що вводять у кожну позицію. Наприклад, для поля **«Номер телефону»** потрібно, щоб всі введені значення, щодо, телефонного номера містили точне число тільки цифрових знаків і становили повний номер телефону (наприклад, у США це код штату, код міста й номер абонента). Якщо для поля визначені як формат відображення, так і маска вводу, то при додаванні й редагуванні даних використається маска вводу, а параметр **«Формат поля»** визначає відображення даних при збереженні запису. Якщо використовується як властивість **«Формат поля»,** так і властивість **«Маска вводу»,** необхідно забезпечити, щоб результати їх дії не суперечили один-одному. Маска вводу для поля таблиці створюється в режимі конструктора за допомогою майстра.

3.4 Ключі й індекси

Могутність реляційних баз даних полягає в тому, що можна швидко знайти й зв'язати дані з різних таблиць за допомогою запитів, форм та звітів. Для цього кожна таблиця повинна містити одне або кілька полів, які однозначно ідентифікують кожен запис у таблиці. Такі поля називаються ключовими полями таблиці. Якщо для таблиці позначені ключові поля, то ядро бази даних запобігає дублюванню або вводу порожніх значень у ключове поле.

3.4.1 Типи ключових полів

В Microsoft Access можна виділити три типи ключових полів: лічильник, простий ключ і складовій ключ (рис.7).

Створення первинних ключових полів (рис.7)

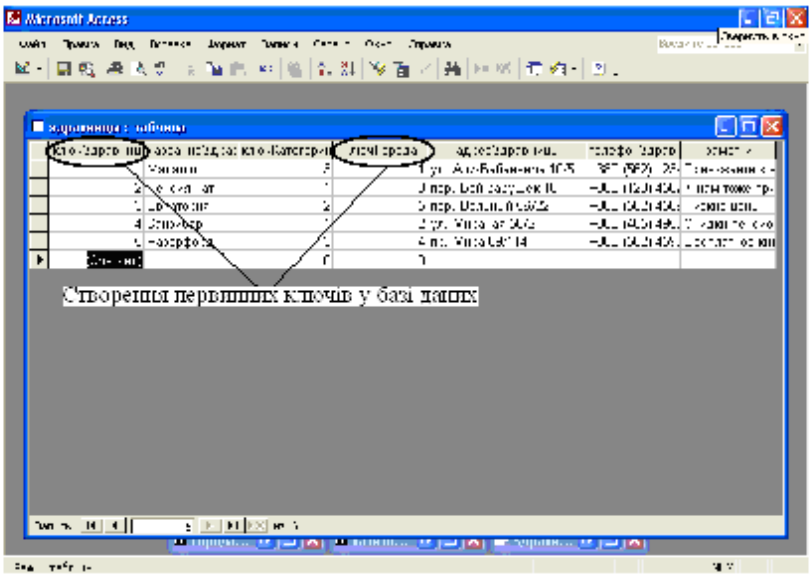

Рис.7

Ключові поля лічильника

Поле лічильника можна задати таким чином, щоб при додаванні кожного запису в таблицю в це поле автоматично вносилося порядкове число. Вказівка такого поля в якості ключового – найбільш простий спосіб створення ключових полів. Якщо до збереження створеної таблиці ключові поля не були визначені, то при збереженні буде видане повідомлення про створення ключового поля. При натисканні кнопки **«ОК»** буде автоматично створене ключове поле лічильника.

Простий ключ

Якщо поле містить унікальні значення, такі як коди або інвентарні номери, то це поле можна визначити як ключове. Якщо обране поле містить повторювані або порожні значення, то воно не буде визначено як ключове. Для визначення записів, що містять повторювані дані, можна виконати запит на пошук повторюваних записів. Якщо усунути повтори шляхом зміни значень неможливо, то треба або додати в таблицю поле лічильника й зробити його ключовим, або визначити складовий ключ.

Складовий ключ

У випадках, коли неможливо гарантувати унікальність значень кожного окремого поля, слід створити ключ, який складається з декількох полів. Найчастіше така ситуація виникає для таблиці, використовуваної для зв'язування двох таблиць у відношенні **«багато-до-багатьох»**. Прикладом такої таблиці може служити таблиця **«Книги»** бази даних **«Картотека»**, яка зв'язує таблиці **«Автори»** й **«Видавці»**. У цій таблиці можна призначити ключ, що складається із двох полів: **«№ Автора»** й **«№ Видавця»**. У таблиці **«Книги»** може бути представлено багато авторів і багато видавців, але кожна книга присутня в картотеці тільки один раз, тому комбінація значень полів **«№ Автора»** й **«№ Видавця»** достатня для утворення ключа.

Іншим прикладом може служити складська база даних, в інвентарній книзі, в якій використаються один основний й один або кілька допоміжних інвентарних номерів.

Важливо: Якщо визначити підходящий набір полів для складового ключа складно, можна просто додати поле лічильника й зробити його ключовим. Наприклад, не рекомендується визначати ключ по полях **«Імена»** й **«Прізвища»**, оскільки не можна виключити повторення цієї пари значень для різних людей.

3.4.2 Індекси

Індекси – об'єкти бази даних, які забезпечують швидкий доступ до окремих рядків у таблиці. Індекс створюється з метою підвищення продуктивності операцій запитів і сортування дані таблиці. Індекси також використовуються для підтримки в таблицях деяких типів ключових обмежень. Ці індекси часто створюються автоматично при визначенні обмеження.

Індекс – незалежний об'єкт, логічно окремий від індексованої таблиці. Створення або видалення індексу ніяк не впливає на дані індексованої таблиці. Індекс зберігає високо оптимізовані версії всіх значень одного або кількох стовпців таблиці. Коли значення запитується з індексованого стовпця, процесор (ядро) бази даних використовує індекс для швидкого знаходження необхідного значення.

Індекси повинні постійно відновлюватися, щоб відбивати останні зміни індексованих стовпців таблиці. Процедури відновлення індексу при вставці, модифікації або видаленні значення в індексований стовпець автоматично виконуються процесором бази даних. Хоча ці операції не вимагають ніяких дій з боку користувача, вони знижують ефективність деяких операцій маніпулювання даними (крім запитів на вибірку). Зменшення продуктивності зв’язане з підтримкою індексу, у більшості випадків компенсується підвищенням швидкості доступу до даних.

Індекси забезпечують найбільші вигоди для відносно статистичних таблиць, по яких часто виконуються запити.

3.4.3 Створення й зміна ключових полів

Правила створення ключових полів таблиці:

1. У режимі конструктора виділити одне або кілька полів, які необхідно визначити як ключові.
2. Для виділення одного поля потрібно натиснути область виділення рядка потрібного поля (кнопка ліворуч рядка). Виділити кілька полів можна, утримуючи при виборі кожного поля клавішу «Ctrl».
3. Нажати кнопку «**Ключове поле»** на панелі інструментів.

Створення індексу

Створити індекси, як і ключі, можна по одному або декількох полях. Складові індекси дозволяють при відборі даних групувати записи, у яких перші поля можуть мати однакові значення. Індексувати поля потрібно для виконання частих пошуків, сортувань або об'єднань із полями з інших таблиць у запитах. Ключові поля таблиці індексуються автоматично.

Не можна індексувати поля з типом даних MEMO, гіперпосилання або об'єкт OLE. Для інших полів індексування використовується, якщо по-

ле має текстовий, числовий, грошовий тип або тип дати/часу й потрібно здійснювати пошук і сортування значень у поле.

Якщо передбачається, що буде часто виконуватися сортування або пошук одночасно по двох і більше полях, можна створити складовий індекс. Наприклад, якщо для того самого запиту часто встановлюється критерій для полів Ім'я та Прізвище, то для цих двох полів має сенс створити складовий індекс. При сортуванні таблиці по складовому індексу спочатку здійснюється сортування по першому полю, визначеному для даного індексу. Якщо в першому полі містяться записи з повторними значеннями, то сортування здійснюється по другому полю й т. і.

Щоб створити індекс для одного поля необхідно:

1. У режимі конструктора в панелі структури таблиці (верхня частина вікна) вибрати поле, для якого потрібно створити індекс.
2. У панелі властивостей (нижня частина вікна) для властивості **«Індексоване поле»** встановити значення **«Допускаються збіги»** або **«Збіги не допускаються»**.
3. Переконатися, що в даному полі співпадаючих записів не існує
4. Вибрати значення **«Збіги не допускаються»**.

Щоб створити складовий індекс:

1. У режимі конструктора на панелі інструментів нажати кнопку **«Індекси».**
2. У першому порожньому рядку поля **«Індекс»** ввести ім'я індексу.

Для індексу можна використовувати або ім'я одного з індексованих полів, або інше придатне ім'я.

3. У поле **«Ім'я поля»** нажати стрілку й вибрати в списку перше поле, для якого потрібно створити індекс.
4. У наступному рядку поля **«Ім'я поля»** вказати друге індексоване поле (для даного рядка поле **«Індекс»** доцільно залишити порожнім).
5. Повторити ці дії для всіх полів, які необхідно включити в індекс. У складеному індексі може бути до 10 полів.

За умовчуванням, установлений порядок сортування **«По зростанню»**. Для сортування даних полів по убуванню в поле **«Порядок сортування»** у вікні індексів потрібно вказати значення **«По убуванню»**.

Відзначимо, що поля індексу можуть не бути ключовими.

Обмеження «Унікальний індекс»

Обмеження «**Унікальний індекс»** запобігає ввід в поле повторюваних значень. Цей тип обмеження може бути встановлений як для одного поля, так і для декількох полів складового ключа. Призначення ключового поля (для одного поля) автоматично забороняє ввід в нього повторюваних значень, тим самим, забезпечуючи для кожного запису унікальний ідентифікатор. Заборона на ввід повторюваних значень може знадобитися й для інших, не ключових, полів.

Щоб установити обмеження «**Унікальний індекс»** для одного поля таблиці необхідно:

1. У режимі конструктора в панелі структури таблиці вибрати поле, у якому допускається ввід тільки унікальних значень.
2. У панелі властивостей, для властивості **«Індексоване поле»** встановити значення **«Збіги не допускаються»**.

Щоб установити обмеження «**Унікальний індекс»** для декількох полів таблиці необхідно:

1. У режимі конструктора відкрити вікно індексів й створити складовий індекс, включивши в нього поля, у які повинен бути, заборонений ввід повторюваних значень.
2. Вибравши ім'я індексу, у панелі властивостей індексу в осередку властивості «**Унікальний індекс»** установити значення **«ОК»**.

3.5 Обмеження й підтримка цілісності даних

Обмеження – необхідні умови управління базою даних, що охоплюють широке коло аспектів: ключі, значення, типи й формати даних і т. і. Обмеження встановлюють для користувача деякі рамки при вводі, зміні або видаленні даних додатків. Вся система обмежень при створенні додатка бази даних будується з метою забезпечення цілісності даних.

Цілісність даних являє собою набір правил, використовуваних процесором бази даних для підтримки зв'язків між записами у зв'язаних таблицях, а також для захисту від випадкового видалення або зміни зв'язаних даних. Наприклад, обмеження можна використати, щоб гарантувати, що кожен «службовець», таблиці «Службовці» занесеної до бази даних буде

ставитися до якого-небудь «відділу» таблиці «Відділи», або щоб користувач, не зміг випадково ввести негативне значення для ціни товару, відповідної таблиці бази даних.

Обмеження можна визначати на двох рівнях:

1. **У базі даних.** Обмеження в базі даних асоціюються з визначеннями об'єктів-таблиць. Наприклад, для таблиці може бути встановлене обмеження, яке вимагає, щоб кожне значення в стовпці було унікальним.
2. **У додатку Access (у програмному коді або властивостях об'єктів).** Обмеження в додатку асоціюються з об'єктами додатка, які формують інтерфейс до інформації бази даних. Наприклад, текстове поле може мати обмеження, котре вимагає, щоб всі значення, які вводять у нього, були більше ніж 20.

3.5.1 Обмеження в базі даних

Обмеження в базі даних – декларативно визначене правило, що лімітує значення, які можуть бути введені в стовпець або набір стовпців у таблиці. Обмеження бази даних є декларативно обумовленими, тому що визначаються як частина структури таблиці при її створенні або зміні. Будучи один раз асоційоване з таблицею, обмеження завжди підтримується, якщо його явно не видалити або не деактивувати.

Розстановка обмежень у базі даних надає наступні переваги:

- **Централізація.** Обмеження бази даних визначається тільки один раз і може автоматично використатися всіма клієнтами, котрі звертаються до бази даних. Визначення обмеження в базі даних звільняє розроблювача від необхідності вносити ті ж самі обмеження в кожну форму, яка використає дану інформацію. При необхідності модифікувати обмеження, зміни вносяться тільки в один об'єкт.
- **Захист.** Обмеження бази даних завжди підтримуються, незалежно від того, який інструмент доступу до даних використовується. Обмеження, визначені в додатку, можуть бути порушені користувачем, котрий приміняє для доступу до тих же таблиць інший додаток або інструмент.
- **Простота.** Обмеження бази даних прості у визначенні й не вимагають ніякого програмного коду.

3.5.2 Типи обмежень в базі даних

Типи обмежень, котрі можна асоціювати з таблицею, варіюються залежно від бази даних, у якій зберігається таблиця. Описані нижче категорії обмежень підтримуються більшістю реляційних баз даних, у тому числі й Microsoft Access.

Обмеження «Порожні рядки» (Not Null)

Обмеження Not Null забороняє ввід в стовпець таблиці порожніх значень. Воно завжди застосовується до окремих стовпців. Обмеження Not Null використовуються, щоб гарантувати, що для важливих даних завжди є значення. Наприклад, це обмеження можна використати, щоб гарантувати, що в записі про кожного службовця в базі даних проставлена його платня.

При визначенні структури таблиці це обмеження задається установкою значень властивостей **«Обов'язкове поле»** й **«Порожні рядки»** поля таблиці.

Необхідно розрізняти два типи порожніх значень: **порожні значення** й **порожні рядки.** У деяких ситуаціях поле може бути залишено порожнім тому, що дані для нього існують, але поки невідомі, або їх не існує зовсім.

Розрізняють два типи порожніх рядків. Наприклад, якщо в таблиці є поле **«Номер факсу»**, то воно може бути порожнім, тому що користувач не знає, чи є в клієнта номер факсу, чи ні. Якщо ж введений порожній рядок то це означає, що строкового значення немає.

Обмеження «Унікальний індекс» (Unique)

Обмеження Unique забороняє користувачу ввід в стовпець або набір стовпців дубльованих значень. Обмеження Unique може активуватися для окремого стовпця або для комбінації стовпців. В останньому випадку обмеження Unique іноді називається складеним обмеженням Unique.

Обмеження Unique використовуються, щоб гарантувати, що в таблиці не буде дубльованих значень стовпців. Наприклад, воно може гарантувати, що кожному службовцеві в базі даних буде привласнений унікальний номер. Обмеження Unique не забороняє користувачу ввід в таблицю деякі-

лькох порожніх значень. Порожнє значення в стовпці завжди задовольняє обмеженню Unique.

В Access обмеження Unique ініціюється установкою значення **«Збіги не допускаються»** для властивості **«Індексоване поле»,** або установкою значення **«ОК»** для властивості **«Унікальний індекс».**

Обмеження «Первинний ключ» (Primary Key)

Обмеження Primary Key гарантує, що кожен рядок у таблиці буде унікально ідентифікований значенням у стовпці або наборі стовпців первинного ключа. Обмеження по первинному ключу поєднує риси обмежень Unique й Not Null.

Рекомендується включати обмеження Primary Key у кожну створювану таблицю. Використання первинного ключа може значно підвищити швидкодію доступу до рядків таблиці.

Обмеження Primary Key також використається для підтримки цілісності даних, коли в базі визначені відносини **«один-до-багатьох»**. Установка цілісності дозволяє підтримувати відповідність між головною й підлеглою таблицями.

Обмеження «Зовнішній ключ» Foreign Key

Обмеження Foreign Key (зовнішній ключ) гарантує, що кожне значення, введене в стовпець, вже існує в деякому іншому стовпці (звичайно, в іншій таблиці). Обмеження Foreign Key використовуються для підтримки цілісності даних, коли в базі визначені відношення **«один-до-багатьох»**.

При відношенні **«один-до-багатьох»** зовнішній ключ - це стовпець у підлеглій таблиці, що містить ідентифікатор рядка в головній таблиці. Значення в стовпці зовнішнього ключа дорівнює значенню в стовпці первинного ключа в іншій таблиці. При відношенні **«один-до-одного»** кожен рядок у підлеглій таблиці відповідає унікальному рядку в головній таблиці, одному рядку в головній таблиці може відповідати будь-яка кількість рядків у підлеглій таблиці.

3.5.3 Підтримка цілісності даних

При підтримці цілісності даних між головною й підлеглою таблицями використовуються наступні правила:

- підлеглий рядок не може бути вставлений, поки не існує головний рядок. Наприклад, не можна ввести записи позицій рахунку-фактури, поки в головній таблиці не з'явиться запис рахунку. У поле зовнішнього ключа можливе введення порожніх значень, які показують, що записи тільки підготовляються й поки не є зв'язаними;
- головний рядок не може бути вилучений до видалення всіх підлеглих рядків. Наприклад, не можна видалити запис рахунку-фактури, якщо в підлеглій таблиці є записи позицій рахунку;
- якщо значення первинного ключа в головному рядку змінено, всі значення зовнішнього ключа, які звертаються до цього значення, повинні бути також обновлені; і навпаки, не можна змінити значення ключового поля в головній таблиці, якщо існують зв'язані записи.

Щоб накласти ці правила на конкретний зв'язок, при його створенні варто встановити прапорець «**Забезпечення цілісності»** даних у вікні «**Зв'язок»**. Якщо даний прапорець установлений, то будь-яка спроба виконати дію, що порушує одне з перерахованих вище правил, приведе до виводу на екран попередження, а сама дія виконана не буде.

Каскадне відновлення й каскадне видалення зв'язків між таблицями

Для зв'язків записів у головній та підлеглій (зв'язаній з нею) таблицях, відносно яких визначена цілісність даних, користувач має можливість указати, чи варто автоматично виконувати операції каскадного відновлення й каскадного видалення зв'язаних записів. Якщо включити дані параметри, то стануть можливими операції видалення й відновлення зв'язаних записів, у противному випадку ці операції заборонені умовами цілісності даних.

Щоб забезпечити цілісність даних при видаленні записів, або зміні значення ключового поля в головній таблиці, автоматично вносяться необхідні зміни у зв'язані таблиці. Якщо при визначенні зв'язку у вікні **«Зв'язок»** встановити прапорець **«Каскадне відновлення зв'язаних по-**

лів», то будь-яка зміна значення в ключовому полі головної таблиці приведе до автоматичного відновлення відповідних значень у всіх зв'язаних записах. Наприклад, при зміні коду клієнта в таблиці «Клієнти» буде автоматично обновлене поле «Код Клієнта» у всіх записах таблиці «Замовлення» для замовлень кожного клієнта, тому цілісність даних не буде порушена. Access виконає каскадне відновлення без введення попереджуючих повідомлень.

Важливо: Якщо в головній таблиці ключовим полем є поле лічильника, то встановлення прапорця **«Каскадне відновлення зв'язаних полів»** не приведе до яких-небудь результатів.

Якщо при визначенні зв'язку встановити прапорець **«Каскадне видалення зв'язаних записів»,** будь-яке видалення запису в головній таблиці приведе до автоматичного видалення зв'язаних записів у підлеглій таблиці. Наприклад, при видаленні з таблиці «Клієнти» запису конкретного клієнта будуть автоматично вилучені всі зв'язані записи в таблиці «Замовлення». Якщо записи віддаляються з форми або таблиці при встановленому прапорці **«Каскадне видалення зв'язаних записів»,** Access виводить попередження про можливості видалення зв'язаних записів. Якщо ж записи віддаляються за допомогою запиту на видалення записів, то видалення здійснюється автоматично, без виводу попередження.

3.6 Запити

Запити використовуються як для безпосереднього перегляду на екрані, так і для формування звітів. Для одержання звітів необхідно сформувати запит на вибірку за допомогою конструктора запитів.

Запити можуть використовуватися як джерела інформації для форм та звітів. У цих випадках в запиті використовуються дані з кількох таблиць. Access виконує запит кожного разу, коли відкривається форма або звіт, тому інформація, що відображена на екрані, завжди нова.

При виконанні в Access звичайного запиту (запиту на вибірку, який просто вибирає потрібні дані) результати відображаються у формі динамічного набору, який має такий самий вигляд, як і таблиця, але фактично є динамічним набором записів базованих на структурі запиту. Записів у динамічному наборі фактично не існує, тому коли цей набір закривається, за-

писи зникають (дані, на яких базований набір, звичайно, залишаються в початкових таблицях).

Запит можна зберегти, але при цьому одержані при його виконанні дані не зберігаються. При збереженні запитів зберігається тільки їх структура. Динамічний набір в Access містить живі дані, а не статичну копію даних первинних таблиць. Тому при модифікації даних у записках динамічного набору запиту модифікуються і записи в первинних таблицях. Дані в динамічному наборі можна змінювати так, як і в таблиці: пересувати і ховати стовпці, змінювати висоту рядків і ширину стовпців.

Приклад виконання запиту (рис.8)

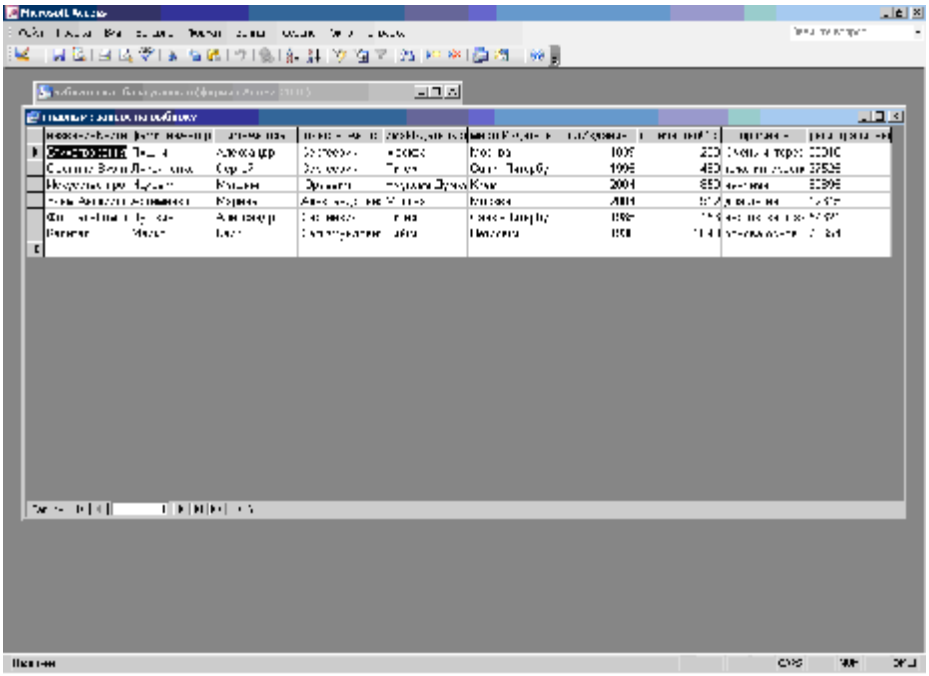

Рис.8

Для виконання запиту (рис.9) необхідно:

- натиснути кнопку “Запит”;
- натиснути кнопку “Створити”;
- в діалоговому вікні “Новий запит” вибрати “Конструктор”;
- зробити вибір елементів запиту (таблиць, данні з яких необхідні);
- після вибору необхідних таблиць у розгорнутому списку натиснути клавішу “Додати”;
- за допомогою діалогового інтерфейсу програми вибрати необхідні поля.

Діалогове вікно створення запиту у базі даних MS ACCESS (рис.9)

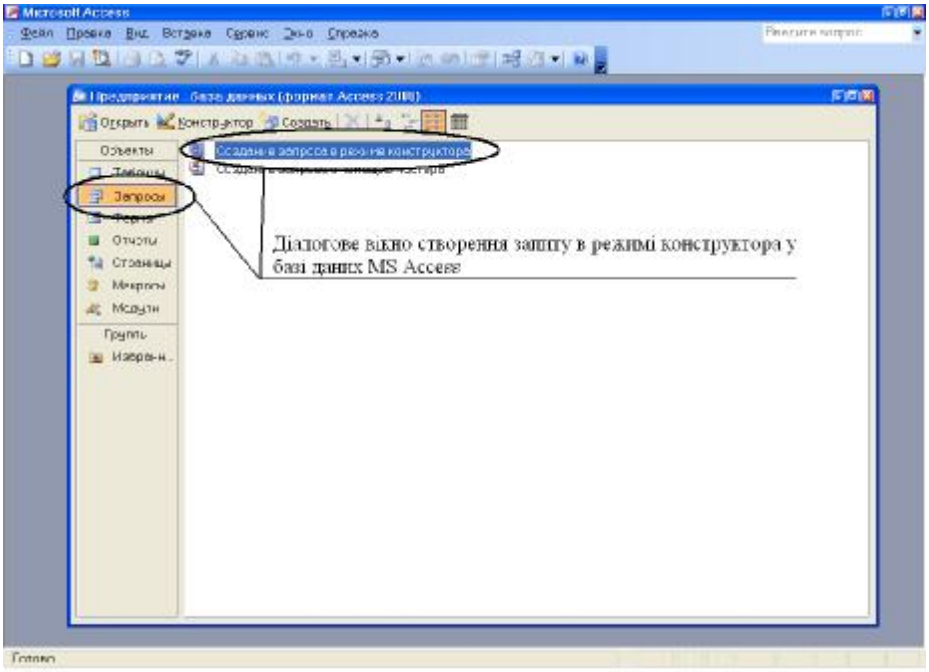

Рис.9

Література

1. Баженов В. А., Венгерський П. С., Горлач В. М., Левченко О. М., та ін. Інформатика. Комп'ютерна техніка. Комп'ютерні технології: Підручник. – К.: Каравела, 2003. – 464 с.
2. Береза А. М. Основи створення інформаційних систем: Навч. посібник. – К.:КНЕУ,1998.–140 с.
3. Дейт К.Дж. Введение в системы баз данных – К.; М.; СПб.: "Вильямс", 1999. – 848с.
4. Економічна інформатика. Підручник для вузів / Під ред. проф. В.В. Евдокимова. – СПб.: Питер, 1997. – 592 с.
5. Куправа Т. А. Самоучитель Access 97/2000. – СПб.: Наука и техника, 2001. – 144 с.
6. Попель Г., Гоулстмайн Б. Информационная технология – миллионные прибыли: Пер. с англ. / Науч. ред. и авт. предисл. В. В. Симанов. – М.: Экономика, 1990. – 238 с.
7. Праг К.Н., Ирвин М.Р. Біблія користувача Access для Windows 98. – К.: Діалектика, 1999. – 576 с.
8. Хансен Г., Хансен Дж. Базы даних: розробка і керування. – М.: БІНОМ, 1999. – 704 с.
9. Microsoft Access 2000: наглядно и конкретно / перевод с английского. – М.: Издательский отдел «Русская редакция» ТОО "Channel Trading Ltd.", 2000. – 256 с.

Зміст

Навчальне видання

Кузьменко В’ячеслав Віталійович
Швачич Геннадій Григорович
Романова Наталія Сергійовна

Комп’ютерні технології в документознавстві
Розділ “Організація та управління базами даних”

Конспект лекцій

Тем. план 2005, поз.

Підписано до друку 18.05.05. Формат 60х84 1/16. Папір друк. Друк плоский. Облік.-вид. арк. 2,47. Умов.-друк. арк. 2,44. Тираж 100 пр. Замовлення №

Національна металургійна академія України
49600, Дніпропетровськ – 5, пр. Гагаріна, 4

Редакційно – видавничий відділ НМетАУ