

БАЗИ ДАНИХ.

СИСТЕМА КЕРУВАННЯ БАЗАМИ ДАНИХ MICROSOFT ACCESS

1. Основні поняття і терміни

У повсякденному житті ми часто користуємось записниками, довідниками, календарями, картотеками тощо. Операції, пов'язані з опрацюванням великих обсягів інформації, сьогодні виконують за допомогою комп'ютера і спеціальних програм.

Довгий час для розв'язання кожної задачі використовувалась тільки своя сукупність даних. Скільки розв'язувалось задач, стільки використовувалось даних. Сукупність даних, що використовувалась в одній програмі, як правило, не могла використовуватись в іншій.

Дані, призначені для розв'язування кожної задачі, оформляли у вигляді так званих наборів даних або файлів даних. Такий метод використання даних має певні недоліки, зокрема, надмірність і дублювання даних.

Чому дублювання відноситься до недоліків використання даних? Це зумовлено двома основними факторами:

- 1) дублювання даних призводить до зайвих витрат пам'яті комп'ютера;
- 2) зміна даних в одній задачі зумовлює необхідність їх зміни в інших задачах.

Ці обставини і були основною причиною появи в кінці 60-х років баз даних (БД). Інформацію потрібно зберігати у БД.

База даних – це єдине централізоване сховище предметної області, до якої мають доступ багато програм.

Тематики баз даних побутового характеру можуть бути, наприклад, такі: телефонний довідник, каталог книг у бібліотеці, розклад руху транспорту, а виробничого – такі: анкетні дані про співробітників, облік випущеної продукції, препаратів на складах в аптеці тощо.

Бази даних використовуються в інформаційно-пошукових системах, наприклад, електронних довідниках, картотеках, каталогах. В аеропорту або на залізничному вокзалі можна отримати на дисплеї інформацію про вільні місця, розклад руху літаків чи поїздів у потрібному напрямку. Сьогодні за допомогою комп'ютера можна зробити покупку не виходячи з дому. Переглядаючи ілюстровану базу даних – електронний каталог, можна вибрати продукти, одяг чи модель автомобіля й організувати через Інтернет відповідне замовлення, оплату та доставку.

Переваги БД:

- в БД дані розробляються і зберігаються як єдине ціле;
- кожна програма вибирає із бази лише ті дані, які потрібні тільки для даної задачі;
- забезпечується незалежність прикладних програм від даних, тобто зміни в даних не викликають за собою необхідність змін у програмі (цю функцію реалізує СУБД)

Дані у базі повинні бути повними, розумно організованими, зручними у користуванні. Даних може бути багато. Їх регулярно потрібно оновлювати. Тому їх зберігають у зовнішній пам'яті у **файлах даних**. На екран комп'ютера дані з файлів зазвичай подають у вигляді таблиць. Рядки таблиці називають **записами**, а стовці – **полями**. Отже, як і будь-який файл даних, БД складається із записів, які діляться на поля.

Запис містить інформацію про якийсь один об'єкт БД, наприклад, інформацію про одну людину, одну книжку, один рейс тощо. Один запис може складатися з даних різних типів. Якщо об'єктом дослідження є людина, то запис про неї може містити такі дані (їх називають **атрибутами**): прізвище, ім'я та по батькові, домашню адресу (це дані текстового типу), рік народження (типу дата), номер телефону (числового цілого типу).

Існують такі типи полів:

- **Текстовий** (може містити до 255 символів, за замовчуванням 50);
- **Поле мемо** використовується для збереження тексту великого обсягу (від 255 до 65 535 символів). Це може бути додаткова інформація до тієї, що є в таблиці;
- **Числової** для збереження числових даних, з якими планується виконувати розрахунки; Для визначення кількості десяткових значень числових полів у розділі **Свойства поля** (внизу вікна конструктора в таблиці) встановлюються такі параметри **Раз мер поля (Одинарное с плавающей точкой; Формат поля (Фиксированный); Число десятичных знаков** (авто (2 знаки) або вводять числове значення);
- **Дата/Время** для збереження значень у вигляді дати і часу;
- **Денежний** для введення даних із грошовими одиницями;
- **Счетчик** – автоматично збільшується на одиницю, коли в таблицю додається новий запис.
- **Поле об'єкта OLE** – розміщуються графічні об'єкти;
- **Гіперссылка** – містить гіперпосилання, які вказують шлях до файла на жорсткому диску або адресу в мережі Інтернет
- **Логічний** - поля, в яких зберігаються значення Так або Ні

2. Системи керування базами даних

Системи керування базами даних (СКБД) – це спеціальні програми, дякуючи яким кожна програма має доступ до конкретних баз даних. СУБД – це програмний засіб, призначений для введення, запису, зберігання, пошуку, сортування, обробки, аналізу і виведення інформації, що міститься в базі даних.

Банк даних – це БД і СУБД.

Основні операції, які можна виконати за допомогою СУБД:

- ❖ Створити новий файл у вигляді таблиці;
- ❖ Створити структуру таблиці (тобто структуру запису);

- ❖ Ввести дані у таблицю;
- ❖ Зберегти таблицю на диску;
- ❖ У разі потреби модифікувати (змінити) структуру;
- ❖ Доповнити базу даних записами;
- ❖ Вилучити зайві записи;
- ❖ Впорядкувати записи за зростанням чи спаданням значень у деякому полі;
- ❖ Відшукати записи, що задовольняють деякому критерію за допомогою фільтра чи звернувшись до бази даних із запитом;
- ❖ Виконати обчислення;
- ❖ Подати потрібні дані у вигляді форми чи звіту.

Отже, в усіх наведених вище операціях переважають функції зберігання і маніпуляції даними, а не функції обчислень. Хоча в БД можна виконати багато фінансових та статистичних розрахунків, ці обчислення виконуються не так швидко, як, наприклад, у табличних процесорах.

3. Хто користується базою даних ?

Осіб, які використовують відомості, що містяться у базах даних, називають **користувачами**.

Серед багатьох категорій користувачів особливе місце займає **адміністратор бази даних (АБД)**. АБД є ідеологом і головним конструктором БД. Ніхто, крім адміністратора, не має права вносити зміни до структури БД, тобто змінювати назву поля, видалити або ввести нове поле.

Головні завдання **системних програмістів** полягають у контролі за функціонуванням бази даних, а також розробці програм, що розширюють можливості СУБД.

Прикладні програмісти розробляють програми обробки даних, що містяться в БД, відповідно до потреб, що виникають.

Крім перерахованих осіб, БД використовують спеціалісти, які як правило не мають хорошої підготовки в області програмування, але які користуються БД у повсякденній роботі. Для цієї категорії людей і розробляються БД (працівники бібліотек, підприємств, відділів кадрів, лікарів, фармацевтів тощо). Цю категорію користувачів називають **кінцевими користувачами**.

4. Поняття реляційної бази даних

Між елементами бази даних (полями та записами) існують певні зв'язки. Залежно від характеру цих зв'язків розрізняють три типи організації баз даних: ієрархічний, мережний та реляційний. Для персональних ЕОМ використовують в основному **реляційні бази даних**.

Концепція реляційної бази даних розроблена Е.Ф. Коддом у 1970р. (Вчений опублікував революційну за змістом статтю, яка серйозно похитнула стале уявлення про бази даних. Йому належить ідея про те, що дані слід пов'язувати відповідно до їх внутрішніх логічних взаємовідносин, а не фізичних покажчиків. Завдяки цьому користувачі зможуть комбінувати дані з різних джерел, якщо логічна інформація, необхідна для такого комбінування, присутня в початкових даних. У своїй статті Кодд запропонував просту модель даних, згідно з якою всі дані зведено в таблиці, які складаються з рядків і стовпців. Ці таблиці одержали назву реляцій, а модель стала називатися реляційною). В основі цієї бази даних лежить математичне поняття відношення (від англ. relation). Відношення представляються у вигляді двовимірних таблиць.

Відношення (таблиця) подається в комп'ютері у вигляді файлу даних. Рядок таблиці відповідає запису у файлі даних, а стовпчик – полю. В теорії реляційних баз даних рядки називають кортежами, а стовпчики атрибутами:

Таблиця	Рядок	Стовпчик
Відношення	Кортеж	Атрибут
Файл	Запис	Поле

Над відношеннями (таблицями) можуть виконуватися різні операції, подібно виконанню арифметичних операцій. Це дає можливість одержувати з одних відношень, що зберігаються в комп'ютері, інші відношення.

Для виконання операцій над відношеннями в СУБД є спеціальні алгоритмічні мови.

5. СКБД Microsoft Access

Однією з найпростіших у користуванні, а отже – найкращою для опанування, є СУБД Access, розроблена фірмою Microsoft Corporation у 1989-1994рр. Вона призначена для роботи з реляційними базами даних, не потребує знання мов програмування, дає змогу працювати в мультизадачному середовищі. Вона входить до набору програмних продуктів Microsoft Office і працює в операційних системах Windows 9X і вищих.

Для її запуску потрібно виконати послідовність команд :

Пуск → Все программы → Microsoft Office → Microsoft Access

Одна з основних переваг СУБД Access полягає в тому, що вона має прості та зручні засоби обробки кількох таблиць у одній базі даних. Таблиця – це місце зберігання даних. Вона є основним об'єктом бази даних.

СКБД Access має значну кількість спеціальних програм, які отримали назву «**майстри**». Так є майстер таблиць, майстер форм і т.д. **Майстри** здійснюють діалог з користувачем, у процесі якого визначаються дані, необхідні для розв'язування відповідної задачі. Для зручності роботи «майстри» поділяють на кроки (етапи). Користувач може пропускати деякі кроки чи повертатися до попередніх.

У роботі з СКБД можна визначити такі етапи (створення та використання):

1. Створення структури БД, таблиць та їх редагування.
2. Редагування даних у БД (записування, дозаписування, вилучення).
3. Створення запитів, форм, звітів.
4. Виведення інформації.

Будь-яка БД створюється спочатку на папері, а потім вводиться в комп'ютер. Створення БД починається з визначення переліку, змісту і типу даних що використовуються, тобто зі створення структури (визначають кількість та тип, надають назви та описують властивості всіх полів).

Кроки щодо створення БД (структури файла бази даних) :

- **Пуск** → **Все программы** → **Microsoft Office** → **Microsoft Access**;
- **Файл** → **Создать** → в області **Доступные шаблоны ЛКМ Новая база данных** → в області **Новая база данных**, в полі **Имя файла ЛКМ по значку папки** → у вікні **Файл новой базы данных** вказуємо де буде зберігатися БД і задаємо ім'я;
- **ЛКМ по вкладці Создание** → в області **Таблицы ЛКМ Конструктор таблиц**;
- **Вводимо конкретні дані в Імя поля та Тип даних**;
- **Вибираємо ключове поле: виділяємо одне з полів** → на вкладці **Работа с таблицами** ► **Конструктор в полі Сервис ЛКМ по Ключевое поле**;
- **Закриваємо вікно Таблица1**
- **На запит Сохранить изменения макета или структуры таблицы 'Таблица1' ЛКМ по Да**
- **У вікні Сохранение вводимо Імя таблиці:** → **ЛКМ ОК**;

Таблиця повинна мати **ключове поле**. Значення таких полів використовуються для створення зв'язків між двома таблицями. Ключ – є ідентифікатором кожного запису БД і не може мати нульових значень або таких, що повторюються. Якщо користувач не створив ключового поля, то під час запису нової таблиці на диск система пропонує створити поле-лічильник.

Коли структура файла бази даних створена, в базу даних можна вводити конкретні дані у відповідні поля.

Створену таблицю відкривають в області навігації 2р. ЛКМ по її значку. Нова таблиця немає записів – тільки назви стовпчиків, що характеризують структуру таблиці. Заповнення таблиці відбувається звичайним способом. Курсор встановлюють в потрібну комірку вказівником миші. Перехід в наступну комірку можна виконати клавішею **Tab**. Перехід до чергового запису виконується після заповнення останньої комірки.

Після наповнення таблиці даними зберігати її не потрібно – все зберігається автоматично. Але, якщо під час роботи з таблицею відбулось редагування її макету (наприклад, змінювалась ширина стовпчика), СКБД попросить підтвердити збереження цих змін.

Якщо виникне необхідність змінити структуру таблиці (склад полів чи їх властивості), таблицю потрібно відкрити в режимі **Конструктор**. Для цього її виділяють в області навігації і виконують команду **ЛКМ Главная** → в області **Режимы ЛКМ Режим** ► **Конструктор** або **ЛКМ по таблиці** → **ЛКМ Конструктор**.

Для внесення змін у структуру будь-якої таблиці :

- **На вкладці Главная** → **ЛКМ Режимы** → із випадального списку вибрати **Конструктор**;
- **Виділити поле, перед яким хочемо вставити нове поле**;
- **На вкладці стрічки Конструктор** ► **Работа с таблицами в полі Сервис ЛКМ по инструменту Вставить строки**;
- **ЛКМ по значку Закреть**;
- **На запит Сохранить изменения макета или структуры таблицы? - ЛКМ по Да**

Над елементами таблиці можна виконувати ряд дій:

Сортування записів:

I спосіб:

Виділити поле → ЛКМ по кнопці випадального списку біля назви поля → ЛКМ **Сортировка от А до Я**

II спосіб:

Викликати контекстне поля → ЛКМ **Сортировка от А до Я**

Сховати поле:

Виділити поле → ПКМ по ньому → **Скрыть поля**

Показати поле (сховане):

- Викликати контекстне меню поля;
- ЛКМ **Отобразить поля**
- У вікні **Отображение столбцов** поставити прапорець навпроти потрібного поля ;
- ЛКМ **Закреть**

Для зміни ширини поля:

- Викликати контекстне меню поля;
- ЛКМ **Ширина поля;**
- У вікні **Ширина столбца** ЛКМ **По ширине данных**

Створення міжтабличних зв'язків

В міжтабличних зв'язках два основні призначення: забезпечення цілісності даних і автоматизація задач обслуговування бази. Тобто зв'язок між таблицями дозволяє:

- ✓ виключити можливість видалення чи зміну даних в ключовому полі головної таблиці, якщо з цим полем зв'язані які-небудь поля інших таблиць;
- ✓ зробити так, що при видаленні (чи зміні) даних в ключовому полі головної таблиці автоматично (і абсолютно коректно) відбудеться видалення чи зміна відповідних даних в полях зв'язаних таблиць.

Таким чином, суть створення зв'язків між таблицями полягає, з одного боку, в захисті даних, з іншого боку – в автоматизації внесення змін відразу в декілька таблиць при змінах в одній таблиці.

Якщо структура бази даних продумана, а зв'язки намічені, то створення реляційних відношень між таблицями виконується просто:

- Переконайтесь, що на панелі навігації присутні потрібні Вам таблиці;
- Виконайте послідовність команд: ЛКМ по вкладці **Работа с базами данных** → в області **Отношения** ЛКМ по **Схема данных** /одночасно з відкриттям вікна **Схема данных** відкриється діалогове вікно **Добавление таблицы**/;
- На вкладці вікна **Таблицы** виберіть по черзі всі таблиці, закрийте вікно /якщо ви не вибрали всі таблиці, то це можна зробити і пізніше – ПКМ по області вікна →ЛКМ **Добавить**;
- Перетягніть від повідне поле з однієї таблиці на список полів іншої таблиці;
- У вікні що з'явиться **Изменение связей** встановіть прапорець **Обеспечение целостности данных** і перевірте правильність вибраних полів, ЛКМ по **Создать**;
- Аналогічно перетягніть інше поле з однієї таблиці в іншу;
- Змініть вигляд розташування таблиць так, щоб вони якнайкраще візуально представляли зв'язки між полями.

Робота з формами

Вводити дані можна шляхом заповнення рядків створеної таблиці. Однак середовище БД дає змогу вводити дані, попередньо створивши форму.

Форма – це аналог картки, в якій введені дані щодо одного конкретного об'єкту.


Введення даних безпосередньо у таблицю не дуже зручне, тому що довжина деяких полів дуже велика і всі стовпчики одночасно не видно на екрані. Другий недолік полягає в тому, що в таблиці видно дані одразу по всіх записах, а це не завжди бажано, особливо в тих випадках, коли необхідно забезпечити конфіденційність. Форма – дуже зручний спосіб заповнення нових записів, схожий на заповнення картки. Вона дає можливість вводити записи по одному, у суворо визначеній послідовності полів і легко контролювати дані під час введення. Після заповнення поточної форми MS Access запам'ятовує її і виводить черговий незаповнений бланк, у рядку стану якого показано кількість введених записів.

Якщо форма базується тільки на одному об'єкті вона називається **простою**. Якщо форма базується на полях з кількох зв'язаних таблиць, то вона називається **складною** і являє собою композицію із кількох форм.

Microsoft Access містить різні види форм. Так, з допомогою **Майстра форм** можна створювати форми наступних видів: **в один стовбец, ленточные, табличне, выровненные**.

Крім того, програма підтримує деякі спеціальні види форм. Наприклад, так звані розділені форми. Ці форми підтримують одночасний вивід даних в двох представленнях – у вигляді форми та у вигляді таблиці. Дві частини форми синхронізуються одна з одною.

Створення форм (стовпцева форма):

- В області навігації ЛКМ по випадяючому списку **Все объекты Access** → ЛКМ **Формы**;
- ЛКМ по вкладці **Создание в области Формы** ЛКМ по **Мастер форм**;
- В діалоговому вікні **Создание форм в полі Таблицы и запросы** із випадяючого списку вибрати потрібну таблицю;
- **Перемістити всі поля із поля Доступные поля: в поле Выбранные поля:**, для цього ЛКМ по  ЛКМ **Далее**;
- В наступному вікні виберіть зовнішній вигляд форм, поставивши перемикач навпроти **в один столбец** → ЛКМ **Далее**;
- В наступному вікні задайте ім'я форми та поставте перемикач **Открыть форму для просмотра и ввода данных**. → ЛКМ по **Готово**;
- У вікні створеної форми пролистайте усі записи, і введіть через форму декілька нових записів.

Елементи управління форми. Елементи управління, якими може користуватися розробник форми представлені на вкладці **Конструктор** ► **Элементы управления**. Ця вкладка належить групі **Инструменты конструктора форм**, яка стає доступною під час роботи з формою в режимі **Конструктора**. Вибір елемента управління виконується клацанням миші на відповідному значку, після чого потрібно наступним клацанням у полі форми відмітити місце, куди його потрібно вставити. Разом з елементом в поле форми вставляється його приєднаний стандартний напис.