

ЛЕКЦІЯ №6,7

Тема: СИСТЕМИ УПРАВЛІННЯ БАЗАМИ ДАНИХ

ПЛАН

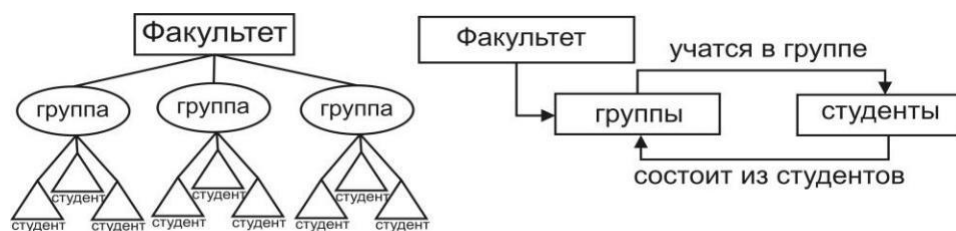
- 6.1 Поняття і види баз даних
- 6.2 Поняття системи управління базами даних (СУБД)
- 6.3 Проектування бази даних
- 6.4 СУБД Microsoft Access
 - 6.4.1 Загальні відомості
 - 6.4.2 Архітектура Microsoft Access
 - 6.4.3 Робота з вікном бази даних
 - 6.4.4 Створення нової таблиці
 - 6.4.4 Фільтрація даних

Час: 2 год.

Література: [1-3, 5].

6.1 Поняття і види баз даних

База даних – сукупність відомостей, які описують об'єкт. Істинною основою того, що ми намагаємося моделювати, є об'єкти і відношення. Об'єкти це елементи реального світу, інформацію про які ми збираємося зберігати в базі даних. Відношення - це зв'язки між об'єктами. Бази даних бувають:



Иерархическая

Сетевая



Реляционная

Плоскі або файлові – звичайно представлені у вигляді таблиці або файлу. Перевага - простота побудови, недолік - велике число повторень (надмірність);

Ієрархічні - складається з впорядкованого набору дерев; Перевага - логічність побудови, недолік - далеко не всі дані можна збудувати у вигляді дерев;

Сітьові - сітьова БД складається з набору записів і набору зв'язків між цими записами; Перевага - можливість скріплення великої кількості даних, недолік - з часом база даних перетворюється на важкоописувану структуру;

Реляційні - БД, що складається з декількох зв'язаних таблиць, де всі дані взаємозв'язані (англ. relation - відносини), цим досягається добра структурованість, за відсутності повторень;

Сучасні БД є реляційними, тобто підтримують бази даних, які складаються з безлічі залежностей. Залежність в даному контексті - це таблиця даних.

6.2 Поняття системи управління базами даних (СУБД)

Система управління базами даних – це прикладна програма, реалізована на електронній обчислювальній машині чи обчислювальному комплексі.

За допомогою СУБД можна:

1. Створювати структуру бази даних, вводити інформацію та зберігати її на зовнішніх носіях;
2. Виконувати певне коло операцій з даними;
3. Одержувати результати та зберігати їх на зовнішніх носіях або передавати на віддалені термінали;
4. Виводити інформацію на термінал у зручній для користувача формі або на друкувальні пристрої;
5. Давати можливість працювати з базами даних багатьом користувачам.

СУБД дає можливість позбутися ряду недоліків, які раніше притаманні були базам даних:

- може бути значно зменшена надмірність інформації через нормалізацію таблиць, у яких зберігаються дані;
- може бути збережена цілісність та достовірність даних;
- може бути збережена безпека (захист) даних;
- може бути досягнута незалежність даних від багатьох користувачів, що і є основною метою створення систем управління базами даних.

Будь-яка СУБД дозволяє виконувати чотири найпростіші операції з даними:

- *додавати* в таблицю одну або кілька записів;
- *видаляти* з таблиці одну або кілька записів;
- *оновляти* значення деяких полів в одній або декількох записах;

- **знаходити** одну або кілька записів, що задовольняють заданій умові.

СУБД Access зберігає всі дані в одному файлі, хоча й розподіляє їх по різних таблицях (файл.mdb), що служить для зберігання структурованих даних настроювань, що описують об'єкт й, СУБД стосовно до цієї БД і пов'язаній з нею завдання.

Зауваження: У базі даних потрібно зберігати тільки необхідну інформацію, і при цьому всі дані повинні зберігатися тільки в одному місці

Основна причина, по якій потрібно дотримуватися цього правила - це те, що такі дані довго вводити й важко аналізувати. Якщо випадково при введенні значення користувач зробило граматичну помилку, то при запитах й угрупованнях таке значення буде розглядатися як самостійне, і рядок, що містить це значення, не потрапить у потрібну групу або просто не буде виведена на екран.

6.3 Проектування бази даних

Перед тим як створювати таблиці, форми та інші об'єкти, потрібно задати структуру бази даних. Добра структура бази даних є основою для створення адекватної вимогам, ефективної бази даних. Сам процес проектування бази даних являє собою складний процес відображення опису предметної області у схемі внутрішньої моделі даних.

При проектуванні структури баз даних необхідно уникати повторення даних і створювати для них окремі таблиці. Цей процес називається **нормалізацією**.

Створення нової бази даних

Для створення нової бази даних необхідно скористатися областю завдань **Створення файлу** (New file) вікна додатка Microsoft Access. Ця область з'являється праворуч при першому запуску Access.

Варіант 1. Етапи проектування бази даних

Основні етапи проектування бази даних:

1. визначення мети створення бази даних;
2. визначення таблиць, що їх повинна містити база даних;
3. визначення необхідних у таблиці полів;
4. завдання індивідуального значення кожному полю;
5. визначення зв'язків між таблицями;
6. відновлення структури бази даних;
7. додавання даних і створення запитів, форм, звітів та інших об'єктів бази даних;
8. використання засобів аналізу в СУБД.

1. **Визначення мети створення бази даних.** На першому етапі проектування бази даних необхідно визначити мету створення бази даних, основні її функції та інформацію, яку вона повинна містити. Тобто необхідно визначити основні теми таблиць бази даних та інформацію, що міститимуть поля таблиць.

2. **Визначення таблиць, що їх повинна містити база даних.** Одним з найскладнішим етапом у процесі проектування бази даних є розробка таблиць, тому що результати, які повинна видавати база даних (звіти, вихідні форми тощо), не завжди дають повне уявлення про структуру таблиці. У разі проектування таблиць слід керуватися такими основними принципами:

- інформація в таблиці не повинна дублюватися. Не повинно бути повторень і між таблицями.
- кожна таблиця повинна містити інформацію лише на одну тему.

3. **Визначення необхідних у таблиці полів.** Кожна таблиця містить інформацію на окрему тему, а кожне поле в таблиці містить окремі дані по темі таблиці. Під час розробки полів для кожної таблиці необхідно пам'ятати:

- кожне поле має бути пов'язане з темою таблиці;
- не рекомендуються включати до таблиці дані, що є результатом виразу;
- у таблиці має бути вся необхідна інформація;
- інформацію варто розбивати на найменші логічні одиниці.

4. **Задання індивідуального значення кожному полю.** Кожна таблиця повинна містити поле чи набір полів, що задаватимуть індивідуальне значення кожного запису в таблиці. Таке поле чи набір полів називають *основним ключем*.

5. **Визначення зв'язків між таблицями.** Після розподілу даних по таблицях і визначення ключових полів необхідно вибрати схему для зв'язку даних у різних таблицях. Для цього необхідно визначити зв'язки між таблицями.

6. **Відновлення структури бази даних.** Необхідно на цьому рівні переглянути ще раз структуру бази даних і виявити можливі недоліки. Для перевірки необхідно створити декілька таблиць, визначити зв'язки між ними та ввести кілька записів у кожену таблицю, потім подивитися, чи відповідає база даних поставленим вимогам.

7. **Додавання даних і створення інших об'єктів бази даних.** Якщо структура таблиці відповідає поставленим вимогам, то можна вводити всі дані.

8. **Використання засобів аналізу в СУБД.** Наприклад в СУБД *Microsoft Access* є два інструменти для удосконалення структури бази даних: *Майстер аналізу таблиць* і *Аналізатор швидкості*.

Варіант 2. Розробка проекту бази даних

1. **Розробка логічної моделі даних.** Логічні моделі використовуються розробниками баз даних для формального представлення інформаційних потреб виробництва, економіки, бізнесу тощо. Розробка логічної моделі являє собою ітераційний процес, що складається з фаз аналізу, проектування та оцінювання. При цьому на кожній ітерації додаються нові правила.

2. **Підготовка звіту про логічну модель.** Для відстежування процесу проектування логічної моделі використовуються звіти. Вони також корисні для узгодження вимог із замовником. У звітах перераховуються сутності, їх атрибути, правила та обмеження, що вміщуються до бази даних.

3. **Перетворення логічної моделі у фізичну.** У процесі розробки фізичної моделі сутності, атрибути та зв'язки складають фізичну модель, відображаються у таблиці та стовпчиках. Щоб правильно і добре виконати цей етап проектування, засоби моделювання даних повинні працювати з кількома популярними СУБД SQL-типу, графічно відображати фізичні характеристики, дозволяти призначати та модифікувати тригери за замовчуванням, створювати власні тригери.

4. **Підготовка звіту про фізичну модель.** Такі звіти повинні надавати детальну інформацію про реалізацію обмежень, правил посилкової цілісності, включаючи призначення та зміст тригерів.

5. **Генерація схеми бази даних.** Схема описує реалізацію бази даних з урахуванням специфіки конкретної СУБД. Схема може створюватися або мовою визначення даних (файли DDL), або при прямому зверненні до СУБД.

6. **Супроводження розроблюваної моделі даних.** Більшість баз даних протягом свого життєвого циклу еволюціонує. Для того, щоб спростити цей процес, рекомендується синхронно змінювати модель та базу даних.

7. **Звернення проектування, що виходить з існуючої бази даних.** Відтворення схеми існуючої бази даних служить кільком цілям. Воно дає змогу побудувати модель цієї бази даних, перенести існуючі базу даних з однієї СУБД на іншу, а також дуже просто модифікувати схему бази даних, що функціонує. Ключовими параметрами для виконання цього є точність та гнучкість.

6.4 СУБД Microsoft Access

6.4.1 Загальні відомості

Microsoft Access – функціонально повна реляційна СУБД. В ній передбачені всі необхідні засоби для визначення та обробки даних, а також для керування ними при роботі з великими обсягами інформації.

СУБД Access - настільна (локальна) реляційна система управління базами даних. Всі дані зберігаються у файлах з розширенням *.mdb (Microsoft Data Base).

Micrisoft Access надає максимальну свободу в завданні типу даних (текст, числові дані, дати, час, грошові значення, малюнки, звук, документи, електронні таблиці тощо). Можна також задати формати збереження (довжина рядка, точність подання чисел і дати/часу) і подання цих даних у разі виведення на екран або друк.

Оскільки Micrisoft Access є сучасним додатком Windows, можна використовувати всі можливості DDE (Dynamic Data Exchange - Динамічний обмін даними) та OLE (Object Linking and Embedding, зв'язок та впровадження об'єктів).

За допомогою Micrisoft Access можна безпосередньо обробляти файли Paradox, dBASE III, dBASE IV, FoxPro та ін.

У Micrisoft Access для обробки даних таблиць використовується потужна мова SQL (Structured Query Language – структурована мова запитів).

Micrisoft Access переважно застосовується в чотирьох окремих сферах: на малому підприємстві, у роботі за контрактом, у великих корпораціях, а також у сфері домашнього застосування.

6.4.2 Архітектура Micrisoft Access

У Micrisoft Access *об'єктами* називається усе, що може мати ім'я.

Об'єкти СУБД Access:

Таблиці – це об'єкт, який визначається і використовується для збереження даних. Кожна таблиця включає інформацію про об'єкт певного типу, наприклад про клієнтів. Таблиця містить *поля (стовпчики)*, у яких зберігаються різноманітні дані, наприклад прізвище чи ім'я клієнта, і *записи* (які визначаються також *рядками*). Для кожної таблиці можна визначити *первинний ключ* (одне чи кілька полів, що мають унікальне значення для кожного запису) та один або кілька *індексів* з метою збільшення швидкості доступу до даного.

Запити – це об'єкт, який дає змогу користувачеві одержувати потрібні дані з однієї чи кількох таблиць. Для створення запиту можна використовувати QBE (запит за зразком) або інструкції SQL.

Форми - засіб для більш зручного представлення інформації користувачу. Форма є об'єктом, який призначений переважно для введення, відображення їх на екрані чи керування роботою додатка. Можна використовувати форми для того, щоб реалізувати вимоги користувача до введення даних із запитів або таблиць.

Звіти - засіб для формування документів з подальшим висновком на друк.

Сторінки - файли у форматі HTML, дозволяючі проглядати дані за допомогою браузера;

Макрос – є об'єктом, що являє собою структурований опис однієї або кількох дій, які повинен виконати Accessу відповідь на певну подію.

Модулі - програми для автоматизації і настройки баз даних, написані на мові програмування Visual Basic for Application.

Більшість об'єктів можна створювати за допомогою **Майстра** або **Конструктора**. Майстер більш зручний для непідготовленого користувача і для швидкого створіння об'єктів. Відповідаючи на питання Майстра ви без праці створите стандартний об'єкт за запропонованими шаблонами. Конструктор дозволяє професіональніше створити об'єкт, але вимагає багато часу і твердих знань.

6.4.3 Таблиці і поля

Основний структурний елемент Access – *таблиця*, в якій зберігається інформація. Об'єкт «таблиця» - це лише одна частина Access-системи, в якій справді зберігається інформація. Всі інші об'єкти (такі, як запити, форми та звіти) ґрунтуються на даних таблиць.

Для більшості користувачів операції, що виконуються в базі даних, починаються зі створення однієї чи більше таблиць. Структуровану інформацію, організовану в таблиці, легше зрозуміти і читати.

Усі бази даних мають двомірну структуру. Якщо структуру розуміти як таблицю, то, природно, використовуються терміни *стовпчик* і *рядок*. Загалом, поле є синонімом *стовпчика*, а запис – синонімом *рядка*.

Терміни *рядок* і *стовпчик* застосовуються, коли йдеться про фізичну структуру таблиці, що містить інформацію. *Поле* та *запис* слугують для вираження логічного зв'язку елементів даних, оскільки поля й записи не завжди постають у формі рядків і стовпчиків.

Поля визначають класифікаційні характеристики даних, за якою можна знайти кожен запис, такий, наприклад, як прізвище чи дата народження.

Запис містить інформацію про окрему особу, місце чи предмет.

Властивості полів таблиці

Підпис – за допомогою цієї властивості можна задати назви полів таблиці, які виводяться в різних режимах.

Обов'язкове поле – визначає необхідність уведення даних у це поле (значення Так/Немає) і забезпечує автоматичний контроль уведення даних у такі поля, тобто ні за яких умов користувач не зможе створити запис, у якій дане поле залишиться порожнім.

Формат поля – указує формат відображення даних з поля в режимі Таблиці.

Маска уведення – указується маска, що дозволяє автоматизувати перевірку уведення символів у поле. Вона застосовується до таких полів, як номер телефону, дата й т.д. Задавати маску уведення можна вручну або за допомогою Майстра.

Індексоване поле визначає, чи є дане поле індексованим, і якщо є, то в якому режимі. Існують два режими індексування: **Збіги допускаються** й **Збіги не допускаються**. У першому випадку поле може містити повторювані значення, у другому - немає.

Значення за замовчуванням – указується значення, додає автоматично в поле для кожного нового запису, якщо це значення не уведене користувачем.

3.4.4 Робота з вікном бази даних

Перш ніж перейти до створення таблиць, потрібно створити базу даних, у якій зберігатимуться таблиця та інші об'єкти Access. Із запуском Access з'являється діалогове вікно (рис. 3.1) Access.

Рисунок 3.1 – Діалогове вікно Access

Можна відкрити вже існуючу базу даних, створити нову базу даних або запустити **Мастера** для створення нової бази даних за зразком. Після відкриття наявної бази даних з'явиться вікно з такими вкладками: **Таблицы, Запросы, Формы, Отчеты, Макросы, Модули**. На кожній вкладці є кнопки: **Запуск, Конструктор, Создать**. Натиснувши на одну з цих кнопок, починаємо певну роботу з базою даних. Роботу можна почати також, натиснувши на об'єкті правою кнопкою миші. З'являється відповідне контекстне меню, в якому вибираємо відповідний пункт і починаємо роботу.

3.4.5 Створення нової таблиці

Кожна база даних може містити одну чи більше таблиць інформації. Нові таблиці створюються натисканням кнопки **Создать** у вікні бази даних, якщо обраний корінець **Таблицы**. У результаті цього відкривається вікно **Новая таблица**, показане на рисунку 3.2.

Рисунок 3.2

У правій частині вікна наведено п'ять пунктів, які дають можливість вибрати спосіб завдання таблиці **Режим таблицы**, **Конструктор**, **Мастер таблиц**, **Импорт таблиц** и **Связь с таблицями**. Вибираємо пункт **Конструктор** і натискаємо **ОК**.

Вікно конструктора має три стовпчика: **Имя поля**, **Типы данных** і **Описание**.

Имя поля – це ім'я, що присвоюється даному полю. Кожне поле в одній таблиці мусить мати унікальне ім'я, а також може мати протяжність не більше 64 символів і містити літери, цифри, проміжки та знаки пунктуації.

Описание – це записаний нами коментар, у якому мовиться про призначення даного поля. Ці нотатки заносити не обов'язково, але якщо проект виконується для замовника, то опис повинен бути зроблений, щоб не мати клопотів під час здавання проекту.

Тип полів:

1. **Текстовий (Text)** – використовується для збереження будь-якої послідовності символів, включаючи літери, цифри, знаки пунктуації, прогалини та спеціальні символи. Текстові поля можуть бути завдовжки до 255 символів.
2. **МЕМО (Memo)** – поля цього типу використовуються для збереження довгих текстових блоків до 65 000 символів (близько 16 сторінок інформації). Потрібно використовувати цей тип поля тоді, коли треба ввести чи імпортувати великий текстовий блок. Це поле змінної величини, і задавати розмір його не потрібно.
3. **Числовий (Number)** – використовується для збереження числових значень (крім грошових сум). У числових полях, як правило, містяться значення, над якими виконуватимуться обчислення.
4. **Дата/Время (Date/Time)** – використовується для зберігання дати/часу, значення вводяться в поля такого типу в стандартному форматі, наприклад /1/99/, 12:53:00 PM.
5. **Денежный (Currence)** – використовується для збереження числових значень. Автоматично додає фіксовану кількість цифр після десяткової коми, щоб уникнути помилок округлення.
6. **Счетчик (AutoNumber)** – автоматично присвоює унікальне числове значення для кожного запису, починаючи з 1 для першого запису, 2 для другого і т.д.
7. **Логический (Yes/No)** – цей тип поля використовується для збереження логічних величин, які можуть набувати лише двох значень типу «так» чи «ні».
8. **OLE Object** – цей тип поля містить дані з інших додатків, які підтримують технологію Object Linking and Embedding (OLE) – зв'язок та впровадження об'єктів. Цей тип поля дає змогу зберігати документи, малюнки, звукові сигнали, відео та ін.
9. **Гиперссылка (Hyperlink)** – поле цього типу містить адресу гіперпосилання, яка вказує шлях до іншого об'єкта, документа або WEB-сторінки.
10. **Мастер подстановок (Lookup Wizard)** – поле цього типу використовується для задання списку значень, які може приймати дане поле.

3.5 Фільтрація даних

В Microsoft Access передбачено чотири способи відбору записів за допомогою фільтрів: **фільтр по виділеному фрагменті**, **звичайний фільтр**, **поле Фільтр** для й **розширений фільтр**.

Фільтр по виділеному фрагменті, звичайний фільтр і поле **Фільтр** є дуже простими способами відбору записів, причому найпростішим є фільтр по виділеному фрагменті – він дозволяє знайти всі записи, що містять певне значення в обраному полі. Звичайний фільтр використовується для відбору за-

писів за значеннями декількох полів. Поле **Фільтр для** використовується, якщо фокус уведення перебуває в поле таблиці й потрібно ввести конкретне шука-не значення або вираження, результат якого буде застосовуватися як умова відбору. Для створення складних фільтрів варто використати вікно розширеного фільтра.

Набор записів, які були відібрані в процесі фільтрації, називається **результуючим набором**.

Контрольні запитання

- 1 Які відмінності існують між створенням таблиць Access вручну, за допомогою конструктора та за допомогою майстра таблиць?
- 2 Як полегшити пошук та ідентифікацію записів таблиць?
- 3 Як змінити зовнішній вигляд таблиці?
- 4 Пошук даних у таблицях.
- 5 Як встановлювати зв'язки між таблицями?
- 6 Як створювати нові запити в режимі конструктора?
- 7 Виведення запиту в режимі SQL на друк.
- 8 Перерахуйте основні функції СУБД Access.
- 9 Складові файла бази даних Access.
- 10 Типи полів даних.
- 11 Перелічить групи стандартних функцій Access.