

Тема:

Бази даних.

Мета:

Вивчити основні характеристики і засоби роботи в базі даних.

План

6.1 Основні поняття бази даних.

6.2 Види моделей баз даних.

6.3 Нормалізація відношень у таблицях.

6.4 ER-діаграми.

6.5 Системи управління базами даних.

6.6 СУБД Microsoft Access.

6.7 Створення бази даних у Access.

6.1. Основні поняття бази даних.

У контексті інформаційних систем і технологій особливого значення набуває таке поняття, як ***дані*** - інформація, подана у формалізованому вигляді, придатному для обробки автоматизованими засобами за можливої участі людини. Сучасною формою організації даних на машинних носіях є автоматизовані бази і банки даних.

Автоматизований банк даних - це система інформаційних, математичних, програмних, мовних, організаційних і технічних засобів, необхідних для інтегрованого накопичення, зберігання, ведення, актуалізації, пошуку та видачі даних. Основними складовими автоматизованого банку даних є база даних (БД) і система управління базою даних (СУБД).

База даних - це іменована цілісна структурована сукупність взаємозв'язаних даних, що відбиває стан об'єктів та відношень між ними в певній предметній галузі, організована за певними правилами, які передбачають загальні принципи опису, зберігання і обробки даних.

Особливості роботи з даними дуже специфічні для різних предметних областей. База даних допомагає систематизувати і зберігати інформацію з певної предметної області, полегшує доступ до даних, пошук і надання необхідних відомостей. Сучасні бази даних оперують інформацією, представленою в самому різному форматі - від звичайних чисел і тексту до графічних і відеоданих.

БД тільки надає можливості для зберігання структурованої інформації, будучи однією зі складових відповідної інформаційної системи. Тому, кажучи про правові бази даних, по суті, мають на увазі правові інформаційні системи, що включають крім інформації з відповідної предметної області ще й засоби для її обробки.

Найважливішим компонентом інформаційної системи є особливий програмний продукт - система управління базою даних (СУБД).

СУБД - комплекс програмних і мовних засобів загального і спеціального призначення, необхідних для створення бази даних, підтримки

її в актуальному стані, маніпулювання даними й організації доступу до них різних користувачів чи прикладних програм в умовах застосовуваної технології обробки інформації.

До основних **функцій СУБД** належать:

- 1) опис БД (вказати назви полів, їх довжину, тип та інше);
- 2) введення в БД підготовлених даних;
- 3) перевірка правильності введення даних (контроль за типом);
- 4) редагування даних (вилучення, заміна, коректування, вставка, доповнення);
- 5) обробка запитів від користувачів (пошук певної інформації);
- 6) забезпечення одночасної роботи декількох користувачів з однією БД;
- 7) захист даних.

Організація баз даних є необхідною передумовою для створення правових інформаційних систем і належного забезпечення правовою інформацією суспільства.

Створюючи базу даних, користувач прагне упорядкувати інформацію за різними ознаками і швидко витягувати вибірку з довільним поєднанням ознак. Це можливо зробити, якщо дані структуровані. Структуризація - введення угод про способи представлення даних.

6.2. Види моделей баз даних.

Ядром будь-якої БД є модель даних - сукупність структур даних і операцій їх обробки. Між елементами БД існують певні зв'язки. Залежно від характеру цих зв'язків розрізняють три типи організації БД: ієрархічний, мережний та реляційний.

Ієрархічна модель даних є деревовидною моделлю. Об'єкти зв'язані ієрархічними відносинами подібно переверненому дереву. До основних понять ієрархічної структури відносять: рівень, елемент (вузол), зв'язок. Вузол - це сукупність атрибутів даних, що описують деякий об'єкт. Кожний вузол на нижчому рівні пов'язаний тільки з одним вузлом, що знаходиться на вищому рівні. Ієрархічне дерево має тільки одну вершину (коріння дерева), не підлеглу жодній іншій вершині і що знаходиться на самому верхньому (першому) рівні. Залежні (підлеглі) вузли знаходяться на другому, третьому і т.д. рівнях. До кожного запису БД існує тільки один (ієрархічний) шлях від кореневого запису. Дана модель реалізує зв'язок «один до багатьох».

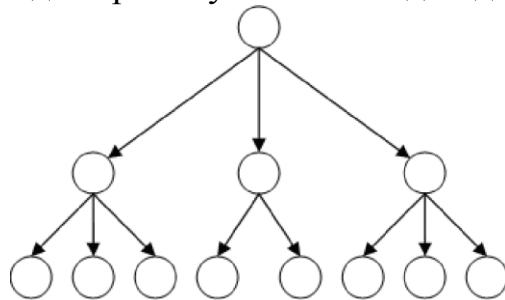


Рис. 6.1. Ієрархічна модель даних

У *мережній моделі* даних при тих же основних поняттях (рівень, вузол, зв'язок) кожен елемент може бути пов'язаний з будь-яким іншим елементом. Для цієї моделі допустимі будь-які види зв'язків між записами, відсутні обмеження на кількість зворотних зв'язків.

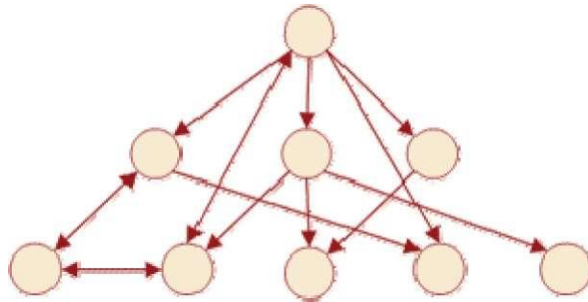


Рис. 6.2. Мережна модель даних

Реляційна модель даних (англ. relation - відношення) орієнтована на організацію даних у вигляді набору двовимірних таблиць, між якими встановлюються зв'язки. Рядки таблиці відповідають записам, а стовпці - полям. Поля утворюють структуру бази даних, а записи складають інформацію, яка в ній міститься. Кожне поле відображає певну характеристику сутності, а відповідний стовпець містить дані одного типу.

Кожна реляційна таблиця є двовимірним масивом і має наступні властивості:

- кожен елемент таблиці - один елемент даних;
- всі стовпці в таблиці однорідні, тобто всі елементи в стовпці мають однаковий тип (числовий, символьний і т.д.);
- кожен стовпець має унікальне ім'я;
- однакові рядки в таблиці відсутні;
- порядок проходження рядків і стовпців може бути довільним.

Таблиці реляційної моделі даних зв'язані між собою ключами.

Ключ (простий ключ) - це поле, що однозначно визначає відповідний запис (значення в цьому полі не повинні повторюватись). Якщо записи однозначно визначаються значеннями декількох полів, то така таблиця БД має складений ключ. Щоб зв'язати дві реляційні таблиці, необхідно ключ першої таблиці ввести до складу другої таблиці.

У реляційній базі даних таблиця, яка включає один або декілька стовпців властивостей об'єкта і містить первинний ключ, який однозначно визначає цей об'єкт, називається базовою (головною, основною). *Первинний ключ* складається з набору значень, які однозначно визначають запис базової таблиці. Будь-якому значенню первинного ключа повинен відповідати один і лише один рядок таблиці. Первинний ключ включає одне поле тільки в тому випадку, якщо це поле не містить значень, що повторюються. *Зовнішній ключ* - це поле або набір полів, значення яких повинні належати множині значень первинного ключа деякої іншої таблиці. Кажуть, що зовнішній ключ посиляється на таблицю, значення ключа якої він містить. Зовнішні ключі використовують для моделювання зв'язків між таблицями та окремими об'єктами. Якщо певні дві таблиці з'єднані зв'язком, то та, яка містить

зовнішній ключ, називається зв'язаною (підпорядкованою).

Приклади реляційної структури БД представлено на рисунках 6.3. та 6.4.

База даних "Шкільний похід", таблиця "Учасники"

Код учасника	Прізвище	Ім'я	Клас	Адреса	Телефон
1	Арбузов	Іван	8-А	вул. Піщана, 4	3-15-98
2	Бірюкова	Ольга	9-Б	вул. Леніна, 45	2-65-47
3	Крамаренко	Ігор	10-А	вул. Горького, 12	3-54-78

База даних "Шкільний похід", таблиця "Знаряддя"

Код знаряддя	Код учасника	Знаряддя
10	1	Палатка
20	3	Ніж
30	1	Ліхтарик

Рис. 6.3. Приклад організації реляційної структури БД 97

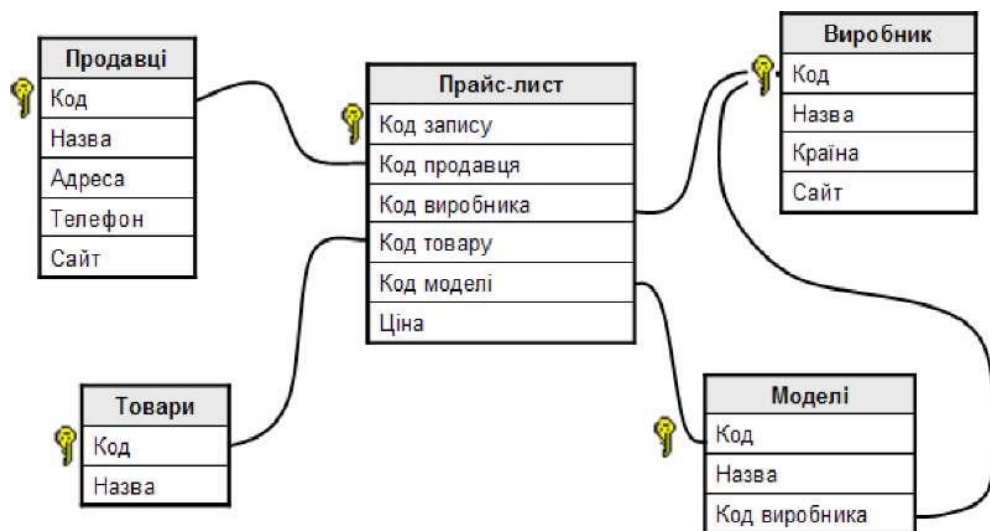


Рис. 6.4. Приклад реалізації реляційної структури БД

6.3. Нормалізація відношень у таблицях.

Нормалізація відношень - формальний апарат обмежень на формування відношень, який дозволяє усунути дублювання, забезпечує несуперечність даних, що зберігаються в базі, зменшує трудовитрати на ведення (введення, коректування) БД.

Принципи нормалізації:

- 1) у кожній таблиці БД не повинно бути полів, що повторюються;
- 2) в кожній таблиці повинен бути унікальний ідентифікатор (первинний ключ);
- 3) кожному значенню первинного ключа повинна відповідати достатня інформація про тип сутності або про об'єкт таблиці;

4) зміна значень у полях таблиці не повинна впливати на інформацію в інших полях (крім змін в полях ключа).

Створення БД зазвичай починається з проектування, а тільки потім можна перейти до її програмної реалізації. Проектування бази даних передбачає розробку структури таблиць та встановлення зв'язків між ними. Внесення даних та їх опрацювання не стосується етапу проектування бази даних.

Під час проектування таблиць спочатку краще розробити їх структуру на папері, при цьому доцільно користуватися основними правилами:

1) дані в таблиці не повинні дублюватися, також не має бути повторень між таблицями. Якщо деякі дані зберігатимуться тільки в одній таблиці, то і змінювати їх доведеться тільки в одному місці;

2) кожна таблиця має містити дані лише з однієї теми. Дані з кожної теми опрацьовуються значно легше, якщо вони містяться в незалежних одна від одної таблицях.

У реальному проектуванні структури бази даних можна використовувати семантичне моделювання. Семантичне моделювання є моделюванням структури даних, спираючись на значення цих даних. Як інструмент семантичного моделювання використовуються різні варіанти діаграм сутність-зв'язок (ER-Entity-Relationship).

6.4. ER-діаграми.

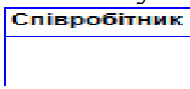
ER-діаграми є стандартним способом визначення даних і відношень між ними. ER-діаграма містить інформацію про сутності системи і способи їх взаємодії, включає ідентифікацію об'єктів, важливих для предметної області (сутностей), властивостей цих об'єктів (атрибутів) і їх відношень з іншими об'єктами (зв'язків). У багатьох випадках інформаційна модель дуже складна і містить безліч об'єктів.

Основними поняттями ER-діаграм є наступні.

Сутність - це клас однотипних об'єктів, інформація про які повинна бути врахована в моделі. Кожна сутність повинна мати найменування, виражене іменником в однині.

Прикладами сутностей можуть бути такі класи об'єктів, як «Постачальник», «Співробітник», «Накладна».

Кожна сутність у моделі зображена у вигляді прямокутника з найменуванням:



Співробітник

Екземпляр сутності - це конкретний представник даної сутності.

Наприклад, представником сутності «Співробітник» може бути «Співробітник Іванов».

Сутності повинні мати певні властивості, унікальні для кожного екземпляра цієї сутності.

Атрибут сутності - це іменована характеристика, що є деякою властивістю сутності.

Найменування атрибута повинно бути виражене іменником в однині

(можливо, з прикметниками, що його характеризують).

Прикладами атрибутів сутності «Співробітник» можуть бути такі атрибути, як «Табельний номер», «Прізвище», «Ім'я», «По батькові», «Посада», «Зарплата» і т.п.

Співробітник	Співробітник
Табельний номер Прізвище Ім'я По батькові Посада Зарплата	<u>Табельний номер</u> Прізвище Ім'я По батькові Посада Зарплата

Атрибути зображені в межах прямокутника, що визначає сутність.

Ключ сутності - це ненадмірний набір атрибутів, значення яких в сукупності є унікальними для кожного екземпляра сутності. Ненадмірність полягає в тому, що видалення будь-якого атрибута з ключа порушує його унікальність.

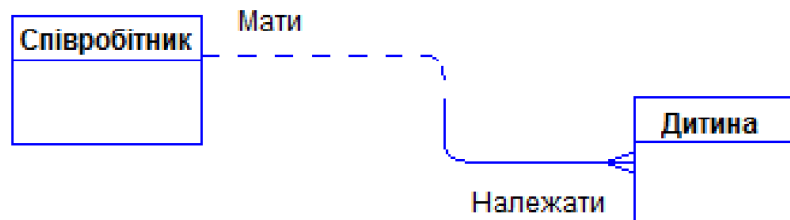
Сутність може мати декілька різних ключів.

Ключові атрибути зображені на діаграмі підкресленням.

Зв'язок - це деяка асоціація між двома сутностями. Одна сутність може бути пов'язана з іншою сутністю.

Зв'язки дозволяють за однією сутністю знаходити інші сутності, пов'язані з нею.

Графічно зв'язок зображений лінією, що сполучає дві сутності:



Кожний зв'язок має два кінці і одне або два найменування. Найменування звичайно виражається в невизначеній дієслівній формі: «мати», «належати» і т.п. Кожне з найменувань відноситься до свого кінця зв'язку. Іноді найменування не пишуться внаслідок їх очевидності.

Кожний зв'язок може мати один з наступних типів зв'язку:

- Зв'язок типу «один до одного» означає, що один екземпляр першої сутності (лівої) пов'язаний з одним екземпляром другої сутності (правої).
- Зв'язок типу «один до багатьох» означає, що один екземпляр першої сутності (лівої) пов'язаний з декількома екземплярами другої сутності (правої). Це тип зв'язку, що використовується найчастіше. Ліва сутність (із сторони «один») називається батьківською, права (із сторони «багатьох») - дочірньою. Характерний приклад такого зв'язку приведений на попередньому малюнку.
- Зв'язок типу «багато до багатьох» означає, що кожний екземпляр першої сутності може бути пов'язаний з декількома екземплярами другої

сутності, і кожний екземпляр другої сутності може бути пов'язаний з декількома екземплярами першої сутності. Тип зв'язку «багато до багатьох» є тимчасовим типом зв'язку, допустимим на ранніх етапах розробки моделі. Надалі цей тип зв'язку повинен бути замінений двома зв'язками типу «один до багатьох» шляхом створення проміжної сутності.

Кожний зв'язок може мати одну з двох модальностей зв'язку.

Модальність «може» означає, що екземпляр однієї сутності може бути пов'язаний з одним або декількома екземплярами іншої сутності, а може бути і не пов'язаний з жодним екземпляром.

Модальність «повинен» означає, що екземпляр однієї сутності зобов'язаний бути пов'язаний не менше ніж з одним екземпляром іншої сутності.

Зв'язок може мати різну модальність з різних кінців.

Описаний графічний синтаксис дозволяє однозначно читати діаграми, користуючись наступною схемою побудови фраз:

<Кожний екземпляр сутності 1> <модальність зв'язку> <найменування зв'язку> <тип зв'язку> <екземпляр сутності 2>.

Кожний зв'язок може бути прочитаний як зліва направо, так і справа наліво. Зв'язок на одному з попередніх малюнків читається так:

Зліва направо: «кожний співробітник може мати декількох дітей».

Справа наліво: «Кожна дитина зобов'язана належати тільки одному співробітнику».

Приклад простої ER-моделі (рис. 6.5.):

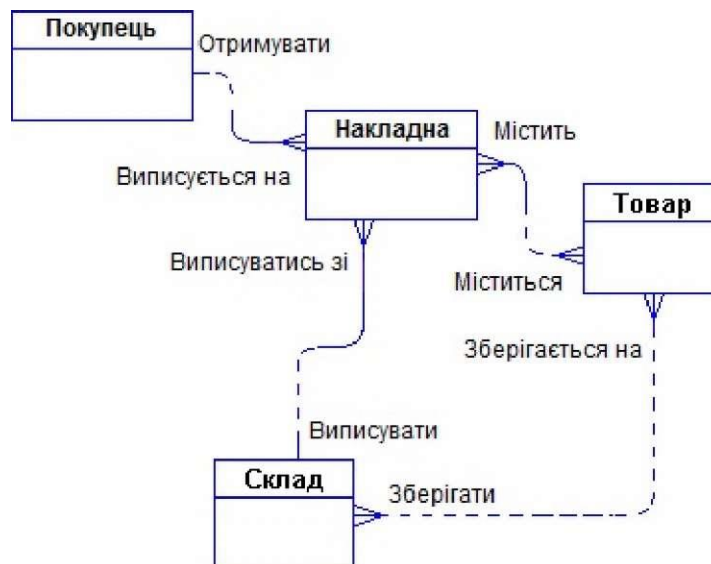


Рис. 6.5. Приклад ER-моделі

6.5. Системи управління базами даних.

За ступенем універсальності розрізняють два класи СУБД:

- системи загального призначення;
- спеціалізовані системи.

СУБД загального призначення не орієнтовані на яку-небудь сферу або на інформаційні потреби якої-небудь групи користувачів. Кожна система

такого роду реалізується як програмний продукт, здатний функціонувати на деякій моделі ЕОМ в певній операційній системі і поставляється багатьом користувачам як комерційний продукт. Такі СУБД володіють засобами настройки на роботу з конкретною базою даних. Зазначеним СУБД властиві розвинені функціональні можливості і навіть певна функціональна надмірність. Використовування СУБД загального призначення як засіб для створення автоматизованих інформаційних систем, заснованих на технології БД, дозволяє істотно скорочувати терміни розробки, економити ресурси.

Спеціалізовані СУБД створюються в окремих випадках при неможливості або недоцільності використання СУБД загального призначення.

СУБД загального призначення - це складні програмні комплекси, призначені для виконання всієї сукупності функцій, пов'язаних зі створенням і експлуатацією БД.

Ринок програмного забезпечення персональних комп'ютерів пропонує різноманітні за своїми функціональними можливостями СУБД загального призначення для різних операційних систем. Приклади: СУБД Access з пакета Microsoft Office, Base з пакета LibreOffice, Kexi-Kexi - інтегрований додаток для управління даними з пакета KOffice (може використовуватись для проектування баз даних, обробки даних, виконання запитів).

LibreOffice - це якісний і повністю безкоштовний офісний пакет зі всіма необхідними компонентами. Побудований на основі популярного офісного пакета OpenOffice і розвивається спільнотою незалежних розробників. Основною метою Libre Office є створення повністю вільного і незалежного офісного пакета. Пакет LibreOffice сумісний з ОС: Windows, MacOSX, Linux.

LibreOffice Base - це повнофункціональна оболонка для роботи з базами даних, розроблена для задоволення потреб широкого кола користувачів в різних ситуаціях.

Base надає користувачам допоміжну програму - Майстра, що допомагає в створенні нових баз даних або нового дизайну БД; Майстер допомагає створювати таблиці, запити, форми і звіти; він поставляється з набором типових таблиць, орієнтованих на контроль активів, облік клієнтів і замовлень, створення рахунків-фактур і багатьох інших корисних речей, що використовуються досить часто.

Якщо необхідна проста, розрахована на одного користувача БД, то у складі Base є вбудований движок реляційної бази даних HSQL. Для людей, що потребують простої для розуміння і використання системи, дані зберігаються прямо у файлі Base.

Base стане в нагоді і досвідченим користувачам з потребами корпоративного рівня; вона забезпечує підтримку драйверів для декількох широко поширених, розрахованих на багатьох користувачів СУБД: MySQL, Adabas D, MS Access і PostgreSQL. Крім того, вбудована підтримка драйверів стандартів JDBC і ODBC дозволяє підключатися практично до будь-якої іншої бази даних.

Base - це не просто інструмент для автономної роботи з базами даних,

він також чудово інтегрується з іншими додатками пакета LibreOffice. Наприклад, за допомогою Base ви можете передавати дані адресної книги майстру створення листів в Writer, використовуючи стандартний протокол LDAP; також можна експортувати дані в такі поширені формати адресних книг, як Microsoft Outlook, Microsoft Windows і Mozilla. Нарешті, Base може служити джерелом даних для побудови зв'язаних діапазонів даних (linked data ranges) для файлів Calc, а також для аналізу даних або як основа для діаграм. Тому, хоча Base і є оболонкою (front end) для баз даних, і сама по собі - потужний інструмент, вона також забезпечує додаткову функціональність іншим додаткам пакета LibreOffice. Якщо ж Base використовується в поєднанні з написанням макросів (а ця можливість включена в усі компоненти LibreOffice), то реальні можливості по обробці даних обмежені лише уявою.

Додаток Base призначений для роботи з невеликими за обсягами базами даних і сам по собі не запускає ніякі процеси, тому для більш досконалої багаторівневої роботи доцільно використовувати Access.

Далі на прикладі однієї з найпоширеніших систем управління базами даних загального призначення - Microsoft Access, що входить до складу популярного пакета Microsoft Office, познайомимося зі способами створення баз даних і з прийомами роботи з базами даних.

6.6. СУБД Microsoft Access.

Microsoft Access - це набір інструментальних засобів для створення та експлуатації реляційних БД.

Microsoft Access складається з багатьох об'єктів - таблиць, форм, сторінок доступу до даних, запитів та звітів, що використовуються для представлення даних та управління ними.

Об'єкти БД Access:

Таблиці. Об'єкт, який визначається і використовується для зберігання даних. Таблиця містить поля (стовпці) і записи (які називаються також рядками). В записі зібрана вся інформація про деякий об'єкт (людину, зразок продукції і т.п.). Для кожної таблиці можна визначити первинний ключ (одне або декілька полів, що містять унікальні для кожного запису значення) і один або декілька індексів, що допомагають прискорити доступ до даних.

Форми. Об'єкт, призначений в основному для введення даних, відображення їх на екрані або управління роботою додатка. Форми використовуються для того, щоб реалізувати вимоги користувача до представлення даних із запитів або таблиць. Форми можна також роздрукувати. За допомогою форми можна у відповідь на деяку подію, наприклад, зміна значення певних даних, запустити макрос або процедуру VBA.

Запити. Об'єкт, який дозволяє користувачу одержати потрібні дані з однієї або декількох таблиць. Для створення запиту можна використовувати бланк QBE (запит за зразком) або інструкції SQL (структурована мова запитів). Можна створити запити на вибірку, оновлення, видалення або

додавання даних. За допомогою запитів можна також створювати нові таблиці, використовуючи дані з однієї або декількох існуючих таблиць.

Звіти. Об'єкт, призначений для створення форматова-ного представлення даних, які згодом можуть бути виведені на екран, роздруковані або включені в документ іншого додатка.

Звіти можна створювати на основі таблиць та запитів.

Звіти дозволяють відібрати з БД потрібні зведення і представити їх у вигляді, зручному для сприйняття, а також надають широкі можливості для узагальнення та аналізу даних.

Макрос - це набір спеціальних макрокоманд, який забезпечує об'єднання розрізнених операцій оброблення даних у програму. Макроси автоматизують виконання певної послідовності команд.

Модуль - це програма мовою VBA, яку використовують для реалізації нестандартних процедур при створенні програм. Програмування в Access базується на об'єктах, які містять дані та код (програму), що зберігаються у спеціальних модулях.

Сторінки доступу до даних слугують для забезпечення доступу до даних, що містяться в базі, віддаленій від споживача (наприклад, через Інтернет).

Таблиці та форми використовують для обслуговування змісту БД, а запити та звіти виконують основну функцію БД - перетворення та подання інформації, яка необхідна користувачу.

Істотною перевагою СУБД Access є можливість використання її без спеціальних мов програмування та можливість розв'язувати велике коло завдань.

6.7. Створення бази даних у Access.

Робота з БД підрозділяється на дві фази: *фазу конструювання* і *фазу управління даними*.

До того, як за допомогою БД можна буде управляти інформацією, її потрібно сконструювати. Потрібно спланувати кожен об'єкт БД. Цей процес полягає в створенні об'єкта і подальшому налаштуванні його у відповідності з потребами. Після фази конструювання можна користуватись різними об'єктами БД для управління реальними даними: вводити інформацію, редагувати її, створювати запити з метою видобування певних даних, друкувати звіти, які представляють інформацію у бажаному форматі, або якось інакше обробляти дані.

Технологія розробки СУБД містить декілька етапів, основними з яких є:

- проектування структури БД і зв'язків між таблицями;
- розробка структури окремих таблиць і введення даних в таблиці;
- розробка схеми даних, що реалізовує запроектовані зв'язки між таблицями і запитам;
- розробка форм для реалізації інтерфейсу управління БД;
- розробка запитів;

- розробка макросів і програмних модулів для управління БД;
- розробка звітів для друку документів.

Наведена послідовність етапів не є жорсткою. Звичайно, розробнику СУБД доводиться багато разів повертатися до одних і тих же етапів, поступово уточнюючи проект.

При створенні нової БД доцільно дотримуватись наступної логічно обгрунтованої послідовності дій.

1 етап. Проектування структури БД і зв'язків між таблицями

Для створення БД спочатку необхідно з'ясувати її структуру. Якщо вона заздалегідь спланована за допомогою ER - діаграми, процес конструювання відбувається значно простіше.

Створення структури БД складається з таких етапів:

- 1) складання генерального списку полів;
- 2) визначення типу даних для кожного поля (залежно від характеру інформації, що розміщена в цьому полі);
- 3) розподіл полів генерального списку за базовими таблицями. Важливо визначити, які поля повинні знаходитись в кожній таблиці. На цьому етапі розподіл проводять за функціональною ознакою;
- 4) встановлення ключового поля в кожній таблиці. Як ключове поле необхідно вибрати поле, дані в якому не повторюються. Якщо в таблиці взагалі немає таких полів, завжди можна додатково ввести поле типу «лічильник» - воно не може містити повторів даних за визначенням. Ключове поле індексується автоматично, що надалі надає можливості підвищити швидкість пошуку та сортування;
- 5) встановлення міжтабличних зв'язків. Зв'язок між таблицями організовується на основі загального поля, причому в одній із таблиць воно обов'язково повинно бути ключовим. Найпоширеніші зв'язки: «один до багатьох» та «один до одного». Створення зв'язаних таблиць бажано в тому випадку, якщо інформація в деяких полях таблиці повторюється в декількох записах. Переміщення таких полів в окрему таблицю і встановлення зв'язків між таблицями дозволить суттєво зменшити обсяг БД та підвищити точність обробки;

- 6) налагодження складної схеми згідно з завданням.

Типи даних Access

СУБД Access дозволяє працювати з даними таких типів:

- 1) текстові - звичайний неформатований текст обмеженої довжини (до 255 символів);
- 2) числові - тип даних для збереження дійсних чисел, що допускають використання в математичних обчисленнях;
- 3) грошові - тип даних для збереження грошових сум;
- 4) дата/час - тип даних для збереження календарних дат та поточного часу;
- 5) лічильник - тип даних для натуральних чисел з автоматичним накопиченням, що застосовується для нумерації записів;
- 6) логічний тип - тип даних для збереження логічних даних, що набувають значення «так» або «ні»;

- 7) гіперпосилання - це поле для збереження адрес URL Web-документів мережі Інтернет;
- 8) майстер підстановок - об'єкт, за допомогою якого можна автоматизувати введення даних в поле;
- 9) поле об'єкта OLE - спеціальний тип даних, призначений для зберігання об'єктів OLE, наприклад, мультимедійних;
- 10) вкладення - використовується для збереження двійкових файлів, які неможливо прочитати за допомогою текстового редактора (наприклад, цифрові зображення) або файлів, створених за допомогою інших додатків пакета Microsoft Office.

Властивості полів БД Access

Поля БД визначають структуру БД та групові властивості даних, які записуються в комірці кожного поля. Розглянемо основні властивості полів таблиць БД в СУБД Access:

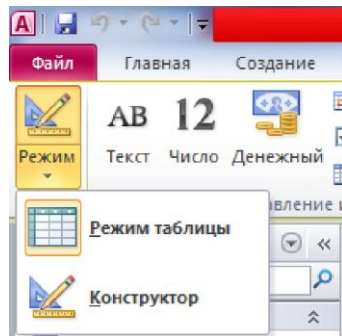
- 1) ім'я поля визначає як потрібно звертатися до даних цього поля при автоматичних операціях з БД і використовується як заголовок відповідного стовпця таблиці;
- 2) тип поля визначає тип даних для даного поля;
- 3) розмір поля визначає граничну довжину для даних, які будуть розміщуватися в полі;
- 4) формат поля визначає спосіб форматування даних у комірках, які належать до поля;
- 5) маска введення - засіб для автоматизації введення даних (наприклад, дат), визначає форму, за якою вводиться дані в поле;
- 6) підпис - визначення заголовка стовпця таблиці для даного поля (якщо підпис не вказаний, то як заголовок використовується ім'я поля);
- 7) значення за замовчуванням - значення, яке вводиться до комірок поля автоматично;
- 8) умова на значення - обмеження, що використовується для перевірки правильності введення даних;
- 9) повідомлення про помилку - текстове повідомлення, яке видається автоматично при спробі введення в поле помилкових даних;
- 10) обов'язкове поле - властивість, що визначає, чи потрібно обов'язково заповнювати дане поле при заповненні БД;
- 11) порожні рядки - властивість, яка дозволяє введення до комірки порожніх рядків;
- 12) індексоване поле - таке поле, в якому всі записи та операції, пов'язані з пошуком або сортуванням, істотно прискорюються;
- 13) смарт-теги - дозволяє обирати смарт-теги, що застосовуються до поля;

Слід зауважити, що властивості полів можуть відрізнятися від наведених вище залежно від типу даних.

2 етап. Створення таблиць БД

Будь-яка таблиця БД має унікальне ім'я і складається з рядків і стовпців. Її структура визначається характером функцій, які має виконувати база. На цій основі задають поля таблиці та призначають їх властивості.

Таблиці БД можна створювати або на основі існуючих шаблонів (Access має набір шаблонів типових баз даних для автоматизації вирішення найбільш поширених задач) або вручну (нову порожню БД можна створити в Режимі таблиці або в Режимі конструктора).



Також таблицю можна створити за допомогою додавання нової таблиці до існуючої БД, імпорту або створення посилання на таблицю з іншого джерела даних (Книга Excel, текстовий файл або інша БД).

Готову таблицю можна редагувати: змінювати, додавати або вилучати поля, змінювати їх параметри. Краще всі виправлення робити до початку заповнення БД, оскільки спроба змінити параметри полів заповненої таблиці може призвести до втрати та перекручування даних.

Редагування таблиць баз даних

У разі необхідності таблиці можуть бути відредаговані. Це можна зробити в Режимі Конструктора або у Режимі таблиць.

Режим «Конструктор» забезпечує:

- зміну імені полів, їх типи і параметри;
- вилучення полів з таблиці та додавання нових;
- змінювати порядок проходження полів;
- змінювати або задавати нові ключові поля і т. д.

Для виконання різноманітних операцій над даними, які зберігаються у таблицях, можна використовувати контекстне меню (контекстне меню стовпців, рядків, таблиці, окремої комірки таблиці, заголовка таблиці).

Редагування даних у будь-якій відкритій таблиці здійснюється відповідно до загальноприйнятих правил роботи в системі Windows. Для цього необхідно підвести курсор мишки до відповідної комірки і клацнути лівою кнопкою мишки. Далі використовуються звичайні засоби редагування.

2 етап. Введення даних в таблицю

Дані в таблицю можна ввести безпосередньо у режимі її відкриття або через спеціально створену форму. Другий спосіб переважний, оскільки форми забезпечують зручніший інтерфейс і можливості контролю введення.

Дані, що вводяться, повинні відповідати типу даних і формату, визначеному в структурі для кожного поля таблиці. При невідповідності Access видає попередження і не дозволяє продовжувати введення. Слід або ввести дані необхідного формату, або відмінити введення.

3 етап. Визначення зв'язків між таблицями

В Access можуть пов'язуватися між собою окремі таблиці БД.

Зв'язок - спосіб, за допомогою якого інформація з однієї таблиці пов'язується з інформацією іншої таблиці.

Як правило, зв'язують ключове поле однієї таблиці (яке є унікальним ідентифікатором кожного запису) з відповідним йому полем іншої таблиці, яке називають полем зовнішнього ключа. Для встановлення зв'язків між таблицями доцільно звернувшись до «Схемы данных» (на вкладці «Работа с базами данных» або на вкладці «Таблица» у групі «Связи»). Схема даних є графічним відображенням БД. Вона використовується різними об'єктами Access для визначення зв'язків між декількома таблицями. Наприклад, при створенні форми, що містить дані з декількох взаємозв'язаних таблиць, схема даних забезпечує автоматичний злагоджений доступ до полів цих таблиць. «Схема данных» забезпечує цілісність взаємозв'язаних даних при коректуванні таблиць.

Зв'язок між таблицями встановлює відносини між співпадаючими значеннями в ключових полях, зазвичай між полями, що мають однакові імена в обох таблицях (але імена співпадають не завжди).

Обов'язковою умовою при встановленні зв'язку є збіг полів, що зв'язуються, по типу і формату.

Стосовно різновидів зв'язків мова йшла раніше (при розгляданні ER-діаграм).

Забезпечення **цілісності даних** визначає систему правил для підтримки зв'язків між записами у зв'язаних таблицях та захищає від випадкового видалення або зміни зв'язаних даних і означає виконання для взаємозв'язаних таблиць таких умов коригування даних:

- 1) у підпорядковану таблицю не можна додати запис з неіснуючими у головній таблиці значеннями ключа зв'язку;
- 2) у головній таблиці не можна вилучити запис, якщо не вилучені пов'язані з нею записи в підпорядкованій таблиці;
- 3) зміна значень ключа зв'язку в головній таблиці повинна приводити до зміни відповідних значень у записах підпорядкованої таблиці;

Встановлювати зв'язки між таблицями типу «один до одного» або «один до багатьох» і задавати для них параметри цілісності даних можна тільки за таких умов:

- зв'язані поля можуть мати різні імена, але тип даних і значення характеристик повинні бути однаковими;
- обидві таблиці повинні зберігатися в одній базі даних;
- головна таблиця зв'язується з підпорядкованою за первинним ключем.

Цілісність даних - це набір правил, що гарантують, що Access працюватиме тільки з несуперечливими даними і дозволеними операціями.

5 етап. Використання запитів

У СУБД Access на рівні з найпростішими засобами пошуку і фільтрації даних з однієї або кількох зв'язаних таблиць використовують зручні та потужні засоби - запити.

Запит - об'єкт, за допомогою якого можна отримати потрібні дані з однієї чи кількох таблиць.

За їх допомогою можуть бути виконані обчислення, відновлення даних у таблицях, додані або вилучені записи. При цьому результати виконання запиту подаються у зручному вигляді - у формі таблиці. Запит може будуватися з використанням тимчасової таблиці, створеної за допомогою іншого запиту.

Засобами запиту можна виконати таку дію:

- 1) вибрати записи з кількох таблиць, що задовольняють умові вибору;
- 2) включити в підсумкову таблицю додаткові поля і в разі необхідності виконати обчислення для них (обчислювальні поля);
- 3) згрупувати записи з однаковими значеннями в деякому полі;
- 4) на основі зв'язаних таблиць створити нову таблицю;
- 5) вилучити зі зв'язаних записи, що відповідають деяким умовам і т. д.

У СУБД Access залежно від задачі, яка розв'язується, можна створити кілька видів запитів:

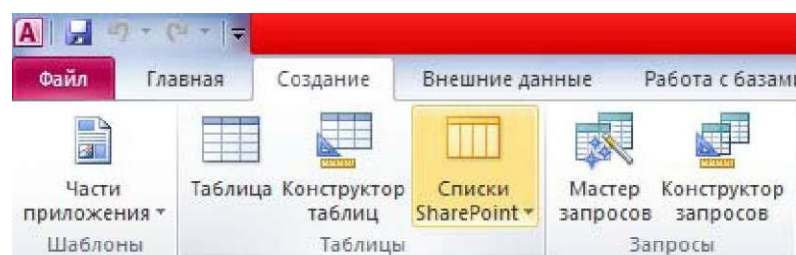
1. *Запрос на выборку таблицы* - запит, що забезпечує вибір даних зі зв'язаних таблиць і таблиць, побудованих під час реалізації інших запитів.
2. *Запрос на создание таблицы* - запит, що ґрунтується на запиті на вибирання і забезпечує формування та заповнення нової таблиці.
3. *Запрос на обновление* - запит, що дає можливість вносити зміни в групу записів, які відбираються за допомогою запиту на вибирання.
4. *Запрос на добавление* - це запит, за допомогою якого записи з таблиці результату запиту додаються в таблиці бази даних.
5. *Запрос на удаление* - запит, що забезпечує виключення записів з однієї або кількох зв'язаних таблиць.

Основою будь-якого запиту є запит на вибирання. Результати виконання запиту відображаються у вигляді таблиці, яка формується на підставі умов відбору. Поля таблиці визначаються користувачем і зазначаються на бланку запиту.

Фактично, запит - це уявлення користувача про потрібні дані з різних таблиць або інших запитів. У процесі відкриття запиту в режимі таблиці або використання його у формах та звітах створюється новий набір записів з поточного змісту бази даних. Дані в запитах можна редагувати. Всі зміни фіксуються у таблицях, дані з яких використовуються у запиті.

Користувач має можливість створювати запити за досить складними умовами, навіть з використанням розрахунків за деякими формулами. У запиті можна навіть створювати значення, які розраховуються за допомогою заданого виразу, тобто створювати обчислювальне поле.

Запити в Access можна створювати за допомогою *Майстра* або за допомогою *Конструктора* (вкладка «Создание», група «Запросы»).



Ще одним із засобів обробки даних є мова структурних запитів SQL, яка використовується під час аналізу, обробки реляційних баз даних. Мова SQL - це структурована мова запитів. Мова SQL складається з інструкцій, речень, операцій та агрегатних функцій, які поєднуються в інструкції для створення, модифікації та маніпулювання базами даних.

У Microsoft Access існує також поняття фільтра, який, в свою чергу, є набором умов, що дозволяють відбирати підмножину записів або сортувати їх. Схожість між запитом на вибірку і фільтрами полягає в тому, що і в тих, і в інших проводиться витягання підмножини записів з базової таблиці або запиту. Проте між ними існують відмінності, які потрібно розуміти, щоб правильно зробити вибір, в якому випадку використовувати запит, а в якому - фільтр.

Основні відмінності запитів і фільтрів полягають у наступному:

- фільтри не дозволяють в одному рядку відображати дані з декількох таблиць, тобто об'єднувати таблиці;
- фільтри не дають можливості вказувати поля, які повинні відображатися в результуючому наборі записів, вони завжди відображають всі поля базової таблиці;
- фільтри не можуть бути збережені як окремий об'єкт у вікні бази даних (вони зберігаються тільки у вигляді запиту);
- фільтри не дозволяють обчислювати суми, середні значення, підраховувати кількість записів і виконувати інші розрахунки.

Запити можуть використовуватися тільки із закритою таблицею або запитом. Фільтри звичайно застосовуються при роботі в режимі Форми або в режимі Таблиці для перегляду або зміни підмножини записів.

6 етап. Застосування форм

Здебільшого дані відображаються у вікні Access у вигляді таблиці. Це зручно для одночасного перегляду великої кількості записів. Але іноді користувачу зручніше працювати з одним записом. Це здійснюється за допомогою форми.

Форма - це об'єкт, призначений для введення нових даних, перегляду існуючих, здійснення обчислень над даними та управління роботою бази.

Можна створити форми з різноманітними цілями.

Використання форм:

1) введення та редагування. Найпростіший засіб використання форм - це введення та редагування даних. При введенні даних значно спрощується внесення змін, вилучення та додавання даних до БД. За допомогою форми можна зробити доступними тільки для читання всі дані або їх частину, обчислювати значення, що виводяться, приховувати значення та інше;

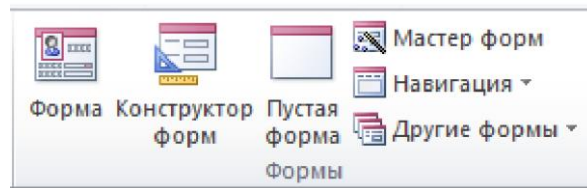
2) виведення повідомлень. Форми можуть надавати відповідну інформацію про роботу додатка або виконувати дію;

3) друк інформації. У формі можна визначати опції для друку;

4) керування ходом виконання додатка. Для керування ходом виконання додатка, автоматизації введення певних даних або виконання певної послідовності дій можна створити форми для роботи з макросами або

функціями VBA.

Форму можна створити декількома способами (в залежності від типу та складності створюваної форми), доступ до яких можна знайти на вкладці «Создание» у групі «Формы»:



7 етап. Створення звітів

Звіти нагадують форми, однак мають інше функціональне призначення - слугують для форматованого виведення даних на пристрій друку. Тому звіти повинні враховувати параметри принтера та паперу.

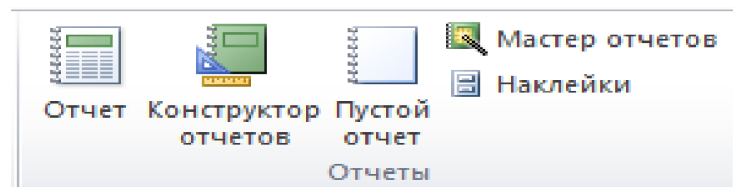
Звіт є кінцевим результатом виконання багатьох процедур із базою даних і забезпечує видачу даних у будь-якому форматі з різними рівнями деталізації. В нього можуть бути включені лінії рамки, рисунки, графіки та інші.

Звіт - особлива форма подання даних, призначених для виведення на друк у зручній для сприйняття формі.

Перед початком створення звіту користувач повинен провести підготовчу роботу, в результаті чого необхідно визначити потрібний макет звіту. У процесі підготовки визначають склад і зміст звіту, а також розміщення в ньому значень, які виводяться з полів таблиць баз даних.

Засоби створення звіту дають можливість групувати дані за кількома рівнями. Для кожного рівня можна обчислити підсумки, визначити заголовки та примітки за кожною групою. Як правило, для створення звіту використовують багатотабличний запит, у який збирають дані з різних таблиць.

Звіт можна створювати декількома способами в залежності від потреби (вкладка «Создание», група «Отчеты»).



8 етап. Створення макросів

Для виконання дій, що повторюються, використовують макроси.

Макрос - певна послідовність дій, що виконується автоматично. Макроси можуть виконувати такі дії: відкривання та закривання таблиць, запитів, форм, звітів; фільтрування, пошук та перехід до певного запису БД; виведення на екран інформаційних повідомлень; подачу звукового сигналу; запуск та вихід із додатків та багато інших.

9 етап. Створення модулів

Для програмування у СУБД Access використовується процедурна мова Visual Basic For Application (VBA) з додаванням елементів об'єктно-орієнтованого програмування, інструкцій SQL та макрокоманд. Програмування в Access базується на об'єктах, які містять дані та код (програму). Програми зберігаються у модулях, які поділяються на:

1) загальні модулі - це окремі об'єкти БД, які використовуються для зберігання програм, що доступні з будь-якого місця додатка;

2) модулі, що зв'язані з формами та звітами. Кожний формат та звіт містять вбудовані модулі з процедурами обробки подій. Ці модулі також можуть містити інші процедури, доступні лише з даної форми або звіту.

Модулі поділяють на модулі класу та стандартні модулі. Стандартний модуль містить тільки код. Модуль класу містить код і дані. Модулі форми та модулі звітів - це модулі класів. Кожний модуль містить розділ описів та процедури, додані користувачем.

Процедури поділяють на два види:

1) процедури - функції FUNCTION (функції користувача) містять інструкції мови VBA та завжди повертають значення, тому їх можна використовувати у виразах;

2) процедури - підпрограми SUB виконують дії, але не повертають

Контрольні запитання

1. Визначте поняття БД і системи управління базами даних (СУБД).
2. Перерахуйте основні функції СУБД.
3. Назвіть види моделей БД, надайте їх стислу характеристику.
4. У чому полягає правило нормалізації відношень у БД?
5. Що таке ER-діаграми?
6. Надайте стислу характеристику основним поняттям ER-діаграм: сутність, екземпляр сутностей, атрибути сутностей, ключ сутності.
7. Що таке зв'язок в ER-діаграмі?
8. Охарактеризуйте зв'язки «один до одного», «один до багатьох», «багато до багатьох».
9. Назвіть відомі СУБД загального призначення та перерахуйте їх основні можливості.
10. Назвіть основні об'єкти БД Microsoft Access та надайте їх стислу характеристику.
11. Перерахуйте етапи створення нової БД в Microsoft Access.
12. Які типи даних існують в Access? Для чого і як створюються міжтабличні зв'язки у СУБД Access?
13. Якими діями забезпечується цілісність даних у таблицях?
14. У чому полягає доцільність розробки і використання форм у БД?
15. Що таке запит у СУБД? Перерахуйте і охарактеризуйте типи запитів при роботі з СУБД Access.
16. Що собою являє звіт у СУБД Access? Мета його створення.
17. У чому полягає мета створення та застосування макросів