

Міністерство освіти та науки України
Західноукраїнський національний економічний університет
Факультет комп'ютерних інформаційних технологій

ОПОРНИЙ КОНСПЕКТ ЛЕКЦІЙ
з дисципліни
«Проектування баз даних»

Тернопіль - 2021

ЗМІСТ

1. Життєвий цикл баз даних
2. Моделі даних. Реляційна модель
3. Концептуальне проектування баз даних
4. Логічне проектування баз даних
5. Нормалізація

ЖИТТЄВИЙ ЦИКЛ БАЗ ДАНИХ

Інформаційна система виконує функції збору та збереження даних, а також обробку та маніпулювання даними. Вона забезпечує структурування даних і управління ними.

Ефективність роботи інформаційної системи залежить від таких складових:

- проекту та реалізації бази даних;
- проекту та реалізації застосунків;
- супроводження інформаційної системи.

База даних є фундаментальним компонентом інформаційної системи і проектування БД виконується в рамках проектування інформаційної системи.

Інформаційна система має життєвий цикл (Systems Development Life Cycle, SDLC), який складається з таких етапів:

- планування;
- збір і аналіз вимог;
- проектування;
- реалізація;
- тестування;
- супроводження.

Ці етапи не є строго послідовними і передбачають повернення на попередні етапи за допомогою зворотних зв'язків. БД, як частина інформаційної системи, має свій життєвий цикл. Життєвий цикл БД складається з таких етапів:

- планування БД;
- аналіз вимог до БД;
- проектування БД (концептуальне, логічне, фізичне);
- розробка застосунків;
- реалізація і завантаження даних;
- тестування;
- експлуатація.

Конкретне наповнення кожного етапу значною мірою залежить від складності продукту, що розробляється. Для невеликих інформаційних систем кількість етапів може бути зменшена. Розглянемо більш детально зміст кожного етапу.

Планування бази даних

Етап планування бази даних передбачає розробку загального стратегічного плану, який дозволить ефективно реалізувати етапи життєвого циклу БД. Тут вирішуються такі питання:

- аналіз існуючих інформаційних систем;
- доцільність зміни існуючої інформаційної системи;
- обсяг робіт і ресурсів, вартість проекту;
- визначення технічного завдання для проекту бази даних;
- визначення технічних вимог;
- розробка методології збору даних, визначення їх формату;

- визначення необхідної документації;
- визначення послідовності проектування і реалізації застосувань.

Аналіз вимог до бази даних

На етапі аналізу вимог до бази даних вирішуються такі задачі:

- визначення діапазону дії і границь застосувань БД;
- визначення складу користувачів і областей застосування;
- визначення представлень користувачів, що підтримуються БД.

На цьому етапі також збираються і аналізуються вимоги користувачів:

- опис даних, що застосовуються (вхідні і вихідні документи);
- детальні відомості про транзакції;
- відомості про засоби застосування даних.

На основі всієї цієї інформації складаються специфікації вимог користувачів.

Проектування бази даних

Процес проектування БД являє собою послідовність переходів від неформального мовного опису інформаційної структури предметної області до формалізованого опису об'єктів предметної області в термінах деякої моделі.

Проектування БД складається з таких етапів:

- системний аналіз предметної області;
- концептуальне проектування;
- логічне проектування;
- фізичне проектування.

Системний аналіз передбачає мовний опис реальних об'єктів предметної області, визначення зв'язків між об'єктами, дослідження характеристик об'єктів і зв'язків. Результати дослідження використовуються при концептуальному проектуванні БД.

Для визначення складу і структури предметної області застосовуються або функціональний, або предметний підходи.

Функціональний підхід застосовує рух "від задач" і використовується у тих випадках, коли заздалегідь відомі функції майбутніх користувачів БД, а також відомі всі задачі, для інформаційних потреб яких створюються БД. В цьому випадку на основі виробничих документів, опитувань замовників можна чітко визначити мінімальний набір об'єктів предметної області та їх взаємозв'язок.

Предметний підхід застосовується у тому випадку, коли інформаційні потреби майбутніх користувачів чітко не визначені. В цьому випадку не можна чітко визначити мінімальний набір об'єктів предметної області. В опис предметної області включаються об'єкти та зв'язки, які є найбільш характерними та найбільш суттєвими для неї. БД називається предметною і може використовуватися для розв'язання задач, які заздалегідь не визначені.

У практичній діяльності використовується комплексний підхід, який з одного боку дозволяє розв'язувати конкретні інформаційні та функціональні задачі, а з іншого боку – враховує можливість додавання нових застосувань.

У загальному випадку існує два підходи до проектування БД: низхідне проектування і висхідне проектування.

Низхідне проектування починається з визначення наборів даних, потім визначаються елементи даних для кожного з таких наборів. Цей процес включає в себе ідентифікацію різних типів сутностей і визначення атрибутів кожної сутності. Низхідне проектування включає операції декомпозиції, що передбачає заміну вихідної множини відношень, що входять в схему БД, іншою множиною відношень, які є проєкціями вихідних відношень.

Цей підхід рекомендується застосовувати у тих випадках, коли кількість, різноманітність та складність сутностей, зв'язків і транзакцій значна за розмірами. Найбільш поширеними моделями для цього проектування є моделі "сутність – зв'язок" (ER-моделі, Entity-Relationship model).

Висхідне проектування починається з виявлення елементів даних, які потім групуються в набори даних. Спочатку визначаються атрибути, які потім об'єднуються в сутності. Висхідне проектування включає операції синтезу, що передбачає виконання компонування із заданої множини функціональних залежностей між об'єктами предметної області вихідних відношень схеми БД.

Цей підхід рекомендується застосовувати у тому випадку, якщо розробляється невелика БД з незначною кількістю об'єктів, атрибутів і транзакцій.

Концептуальне проектування полягає в створенні концептуальної моделі, яку відображає концептуальна схема БД. На цьому етапі визначаються об'єкти, зв'язки між об'єктами, атрибути, ключові атрибути.

Логічне проектування полягає в створенні логічної моделі на основі вибраної моделі даних. На цьому етапі необхідно вже знати яка СУБД буде застосовуватися в системі (ієрархічна, мережна, реляційна, об'єктно-орієнтована). Для перевірки вірності логічної моделі застосовується нормалізація. Крім того логічна модель перевіряється на умову забезпечення всіх транзакцій користувачів.

Фізичне проектування полягає в описі засобів фізичної реалізації логічного проекту БД. Фізичні моделі визначають засоби розміщення даних в середовищі зберігання і засоби доступу до цих даних, які підтримуються на фізичному рівні.

Розробка застосувань

Застосування – програма або програмна система, яка призначена для рішення деякої сукупності задач в даній предметній області, або яка являє собою типовий інструментарій, що застосовується в різних областях.

На цьому етапі вирішуються такі задачі:

- проектування транзакцій;
- проектування інтерфейсів користувачів.

Транзакція може складатися з декількох операцій по роботі з БД, які переводять БД з одного цілісного стану в інший. Розрізняють транзакції по отриманню певної інформації з БД і транзакції по зміні даних в БД (оновлення, вилучення, додавання). Транзакції також можуть бути змішані.

Інтерфейс користувача – сукупність функціональних компонентів, які забезпечують взаємодію користувача з системою.

Реалізація

На етапі реалізації вирішуються такі задачі:

- встановлюється технічне і програмне забезпечення СУБД;
- реалізується проект БД;
- реалізуються прикладні програми;
- реалізуються форми вводу/виводу даних і звіти;
- наповнення БД даними;
- захист БД від несанкціонованого втручання;
- підтримка цілісності БД.

Реалізація БД виконується за допомогою створення опису на мові визначення даних певної СУБД або з використанням графічного інтерфейса користувача. Застосування реалізуються на мовах третього та четвертого покоління або на розширеннях мов БД. Реалізація може виконуватися за допомогою інструментів автоматизованого проектування.

Тестування

На етапі тестування вирішуються такі задачі:

- перевіряється вірність роботи окремих модулів або функціональних компонентів (альфа-тестування);
- проводяться виміри продуктивності роботи системи, визначаються потреби в ресурсах;
- здійснюється дослідницька експлуатація (бета-тестування), при якій перевіряється відповідність розробленої системи її специфікаціям.

Для покращання роботи системи можлива модифікація логічного і фізичного проекту, оновлення або зміна програмного забезпечення СУБД, зміна технічного забезпечення. Також для покращання роботи виконується налагодження системних параметрів і параметрів СУБД.

Експлуатація

На етапі експлуатації вирішуються такі задачі:

- контроль продуктивності роботи системи і в разі потреби підвищення продуктивності (наприклад за рахунок створення додаткових індексів);
- супроводження і модернізація застосувань БД;
- профілактичне обслуговування (резервне копіювання);
- корегуюче обслуговування (відновлення БД);
- призначення прав доступу для нових користувачів;
- ведення статистики доступу до БД для підвищення ефективності роботи системи;
- періодична перевірка безпеки;
- періодичні зведення використання системи.

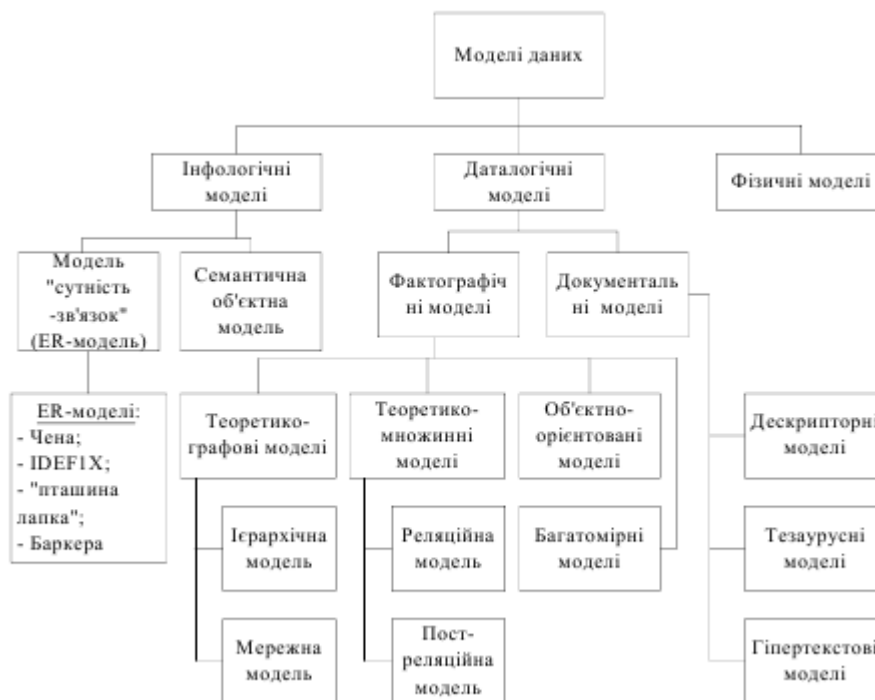
Моделі даних. Реляційна модель

Модель даних – це деяка абстракція, в якій знаходять своє відображення найбільш важливі аспекти функціонування визначеної предметної області, а другорядні – ігноруються.

Модель даних являє собою деяку цільову модель предметної області. У моделі даних розрізняють три головні складові:

- структурна частина, яка визначає правила породження допустимих для даної СУБД видів структур даних;
- управляюча частина, яка визначає можливі операції над такими структурами;
- класи обмежень цілісності даних, які можуть бути реалізовані засобами цієї системи.

Моделювання даних – це процес створення логічного представлення структури бази даних.



Класифікація моделей даних

Кожному рівню представлення інформації відповідає певна модель.

Інфологічна модель – відображає інформацію про предметну область у вигляді незалежного від СУБД, що використовується. Ця модель відображає інформаційно-логічний рівень абстрагування, який пов'язаний з описом об'єктів предметної області, їх властивостей і взаємозв'язків. Часто ці моделі ототожнюють з концептуальними моделями предметної області і називають концептуальними інфологічними моделями (внутрішня і зовнішня концептуальні інфологічні моделі).

Даталогічна модель – модель логічного рівня, яка відображає логічні зв'язки між елементами даних безвідносно до їх змісту і середовища збереження. Часто ці моделі ототожнюють з логічними моделями.

Фізична модель – описує те, як дані зберігаються в комп'ютері, представляючи інформацію про структуру записів, їх впорядкованість і про існуючі шляхи доступу до даних.

Модель "сутність-зв'язок" (ER-модель) – описує модель предметної області і складається з множини сутностей, множини зв'язків між сутностями, а також з атрибутів сутностей і зв'язків. В модель входить обмеження цілісності даних, що пов'язано з двома множинами сутностей і називається залежністю по існуванню. ER-моделі дозволяють графічно представляти моделі предметних областей. Вони є складовою частиною багатьох CASE-продуктів.

Семантична об'єктна модель – описує модель предметної області і являє собою модель даних. Ця модель складається з семантичних об'єктів, що містять сукупність атрибутів. Атрибути групуються у класи. Модель даних володіє більш розвиненими засобами відображення семантики у порівнянні з теоретико-множинними і теоретико-графовими моделями.

Теоретико-графова модель – модель даних, в якій дозволені структури даних можуть бути представлені у вигляді графа загального або спеціального виду, наприклад дерева. Необхідну групу операцій на мові маніпулювання даними, що засновані на цій моделі, представляють навігаційні операції. Операції над даними мають позаописовий характер.

Теоретико-множинна модель – модель даних, в якій використовується математичний апарат реляційної алгебри, реляційного обчислення, а операції над даними маніпулюють таблицями.

Фактографічні моделі – містять відомості, які представлені у вигляді спеціальним чином організованих сукупностей формалізованих записів даних.

Документальні моделі – передбачають, що в якості одиничного елемента інформації виступає неподільний на менші складові частини документ, а інформація про документ, як правило, не структурується, або структурується в обмеженому вигляді. В цих моделях в основному розглядаються тексти на природній мові, формати документів є вільними.

Ієрархічна модель – модель даних в основі якої використовується ієрархічна, деревоподібна структура даних. Вершинами цієї структури є записи, які складаються з простих елементів даних різних типів. Батьківському запису відповідає довільне число екземплярів підлеглих записів кожного типу.

Мережна модель – модель даних, в якій дозволені структури даних можуть бути представлені у вигляді графа загального вигляду. Вершинами такого графа можуть бути дані різних типів – від атомарних елементів даних до записів складної структури. На відміну від ієрархічної моделі наступник в цій моделі може мати довільне число батьків.

Реляційна модель – модель даних, яка заснована на математичному понятті відношення і представленні відношень у формі таблиць.

Постреляційна модель – розширена реляційна модель, яка знімає обмеження неподільності даних, що зберігаються в записах таблиць. Ця модель допускає багатозначні поля – поля, значення яких складається з підзначень. набір значень багатозначних полів вважається самостійною таблицею, яка вбудована в основну таблицю. Часто ці моделі ототожнюють з об'єктно-реляційними моделями.

Об'єктно-орієнтована модель – модель даних, яка базується на понятті об'єкта, тобто сутності, що володіє станом і поведінкою. Стан об'єкта визначається його атрибутами, а поведінка визначається сукупністю операцій, що визначені для цього об'єкта. Також передбачається можливість підтримки зв'язків між типами об'єктів.

Багатомірна модель – модель даних, яка оперує багатомірним представленням даних (у вигляді гіперкубу) і орієнтована на підтримку аналізу даних. Передбачається конструювання різноманітних агрегацій даних у межах гіперкубу, побудова різних його проєкцій – підмножин гіперкубу, деталізація і обертання даних, а також цілий ряд інших операцій.

Дескрипторна модель – описує кожен документ за допомогою дескриптора. Дескриптор має жорстку структуру і являє собою набори деяких лексичних одиниць (слів, словосполучень, термінів), які потрібні для роботи з документами. Дескриптори між собою не зв'язані.

Тезаурусна модель – описує кожен документ за допомогою дескрипторів, а також змістовних відношень між лексичними одиницями (ціле-частина, рід-вид, клас-підклас і т.ін.). Ці моделі дозволяють підвищити ефективність дескрипторних моделей за рахунок більш ефективного відображення предметної області.

Гіпертекстова модель – модель, що заснована на розмітці документа за допомогою спеціальних навігаційних конструкцій, які відповідають змістовим зв'язкам між різними документами, або окремими фрагментами одного документа. Такі конструкції утворюють деяку семантичну мережу в базі документів.

Реляційна модель. Базові поняття

Реляційна модель даних заснована на математичному понятті відношення і представленні відношень у вигляді таблиць. В будь-якій реляційній СУБД припускається, що користувач сприймає БД як набір таблиць. Це стосується тільки логічної структури БД, тобто відноситься до концептуального і зовнішнього представлень. На фізичному рівні БД реалізується за допомогою різних структур зберігання.

Елементи реляційної моделі:

відношення – таблиці;

кортеж – рядок таблиці;

атрибут – заголовок стовпця таблиці;

ключ - Сукупність атрибутів, які унікально визначають кожен рядок таблиці, або виконують функції зв'язування таблиць, або дозволяють прискорити операції над таблицями;

домен – множина значень атрибуту;

схема відношення – рядок заголовків стовпців таблиці.

Для однозначної ідентифікації рядків, для зв'язування таблиць між собою, для прискорення операцій над даними застосовують ключі:

потенційний ключ (Potential Key) - мінімальна підмножина атрибутів відношення, які єдиним чином ідентифікують кортеж даного відношення;

первинний ключ (Primary Key) - потенційний ключ, який обрано для унікальної ідентифікації кортежів відношення;

вторинний ключ (Secondary Key) - ключ, кожному значенню якого може відповідати більш ніж один екземпляр індексованих даних;

зовнішній ключ (Foreign Key) - сукупність атрибутів відношення, значення яких є одночасно і значеннями первинного або потенційного ключа іншого відношення.

Зовнішній і відповідний йому потенційний ключі повинні бути визначені на одному домені.



Порядок кортежів у відношенні не визначений. В реляційних СУБД для зручності кортежі впорядковують за допомогою ключів (первинних або вторинних).

Домені показують множину всіх можливих значень певного атрибута відношення.

Реляційна модель складається з таких частин:

- структурна (тут фіксується відношення як єдине ціле);
- маніпуляційна (тут використовуються два базових механізми маніпулювання реляційною БД – реляційна алгебра і реляційні обчислення);
- цілісність (тут використовується механізм, який запобігає руйнуванню даних).

Реляційній моделі даних властиві простота і природність використовуваних структур даних і операцій маніпуляції даними, повна незалежність від середовища зберігання даних, підтримка віртуальних, а не фізичних зв'язків між даними (на основі значень даних, а не покажчиків).

Реляційна БД включає в себе такі складові:

- інформаційні масиви (таблиці, індекси);
- системна інформація (структура БД, обмеження цілісності);
- прикладні програми (процедури, тригери).

Операційні можливості відношення мають дві еквівалентні форми – реляційна алгебра і реляційне обчислення.

У свою чергу реляційне обчислення поділяється на реляційне обчислення зі змінними кортежами, яке називається обчислення кортежів, і на реляційне обчислення зі змінними доменами, яке називається обчислення доменів.

Для виконання запитів до БД Е.Кодд запропонував відповідні принципи побудови трьох мов.

Мови запитів реляційної алгебри – це алгебраїчні мови, які дозволяють висловлювати запити засобами спеціалізованих операторів, що застосовуються до відношень.

Мови реляційного обчислення дозволяють висловлювати запити шляхом специфікації предиката, якому повинні відповідати потрібні кортежі (домени).

Реальні мови запитів (SQL, QBE і т.ін.) забезпечують не тільки функції відповідної теоретичної мови, але і реалізують деякі додаткові операції (арифметичні, друку і т.ін.).

Цілісність баз даних

Цілісність баз даних – властивість даних, що визначає повноту і правильність інформації, яка вміщується в БД.

Підтримка цілісності даних включає такі складові:

- структурна цілісність;
- обмеження реальних значень даних;
- посилкова цілісність.

Структурна цілісність передбачає виконання таких умов:

- наявність тільки однорідних структур даних типу "реляційне відношення";
- відсутність дублікатів кортежів;
- обов'язкова наявність у кожному відношенні первинного ключа;
- обмеження доменів, яке передбачає визначення кожного атрибуту на своєму домені;
- можливість застосування невизначених значень NULL (позначає відсутність будь-якого значення атрибуту).

Невизначене NULL значення розглядається, як значення невідоме на даний момент часу. Це значення при появі додаткової інформації може бути замінено на деяке конкретне значення.

Обмеження реальних значень даних вимагають, щоби значення поля належали деякому діапазону значень, або задовольняли певному арифметичному співвідношенню між значеннями різних полів. Обмеження значень можуть включати також визначення певних форматів для полів, задоволення значень полів певним статистичним умовам, бізнес правилам предметної області і т.ін.

Посилкова цілісність означає, що зміни в таблицях повинні виконуватися синхронно, а зміст двох пов'язаних таблиць має відповідати таким правилам:

- кожному запису основної таблиці відповідає нуль або більше записів підлеглої таблиці;
- в підлеглій таблиці немає записів, які не мають батьківських записів в основній таблиці;
- кожний запис підлеглої таблиці має тільки один батьківський запис основної таблиці.

Реляційна алгебра

Алгеброю називається множина об'єктів із заданою на ній сукупністю операцій, які замкнені відносно цієї множини.

Основною множиною в реляційній алгебрі є множина відношень. Варіант реляційної алгебри, запропонований Коддом, містить такі основні операції:

об'єднання – визначення відношення, яке вміщує всі кортежі, що належать відношенням R або S , при цьому виключаються з результату всі дублікати кортежів. Відношення R і S повинні бути сумісні за об'єднанням;

різниця - визначення відношення, яке вміщує всі кортежі, що належать відношенню R і відсутні у відношенні S . Відношення R і S повинні бути сумісні за об'єднанням;

перетин - визначення відношення, яке вміщує всі кортежі, що належать відношенням R і S . Відношення R і S повинні бути сумісні за об'єднанням;

декартовий добуток - визначення відношення, яке є результатом конкатенації кожного кортежа з відношення R з кожним кортежем з відношення S ;

проекція - визначення відношення, яке вміщує вертикальну підмножину відношення R , що утворюється шляхом отримання значень вказаних атрибутів і виключення з результату рядків-дублікатів;

селекція - визначення відношення, яке вміщує тільки ті кортежі з відношення R , які задовольняють заданому предикату;

ділення - визначення відношення, яке вміщує ті кортежі відношення R , які визначені на атрибуті C , що відповідає комбінації всіх кортежів відношення S , де C – множина атрибутів, які є у відношенні R , але відсутні у відношенні S .

Концептуальне проектування баз даних

З концептуального проектування починається створення концептуальної схеми БД, в основі якої лежить концептуальна модель даних. Концептуальна модель представляє загальний погляд на дані. Розрізняють два головних підходи до моделювання даних при концептуальному проектуванні:

- семантичні моделі;
- об'єктні моделі.

Семантичні моделі головну увагу приділяють структурі даних. Найбільш поширеною семантичною моделлю є модель "сутність – зв'язок" (Entity Relationship model, ER-модель).

ER-модель складається із сутностей, зв'язків, атрибутів, доменів атрибутів, ключів.

Моделювання даних відображає логічну структуру даних, аналогічно, як блок-схеми алгоритмів відображають логічну структуру програми.

Об'єктні моделі головну увагу приділяють поведінці об'єктів даних і засобам маніпуляції даними. Головне поняття таких моделей – об'єкт, тобто сутність, яка має стан і поведінку. Стан об'єкта визначається сукупністю його атрибутів, а поведінка об'єкта визначається сукупністю операцій специфікованих для нього.

Зближення цих моделей реалізується в розширеному ER-моделюванні (Extended Entity Relationship model, EER-модель).

Модель "сутність-зв'язок"

ER-моделювання являє собою низхідний підхід до проектування БД, який починається з визначення найбільш важливих даних, які називаються сутностями (entities), і зв'язків (relationships) між даними, які повинні бути представлені в моделі. Потім в модель заноситься інформація про властивості сутностей і зв'язків, яка називається атрибутами (attributes), а також всі обмеження, які відносяться до сутностей, зв'язків і атрибутів. ER-модель дає графічне представлення логічних об'єктів і їх відношень в структурі БД. Послідовність проведення ER-моделювання:

- визначення типів сутностей;
- визначення типів зв'язків;
- визначення атрибутів сутностей і зв'язків;
- визначення потенційних і первинних ключів;
- розширене моделювання;
- перевірка моделі на відсутність збитковості;
- перевірка моделі на відповідність транзакціям користувачів.

Сутності

Сутність дозволяє моделювати клас однотипних об'єктів. Сутність має унікальне ім'я у межах системи, що моделюється. Оскільки сутність відповідає деякому класу однотипних об'єктів, то передбачається, що в системі існує

багато екземплярів даної сутності. Об'єкт, якому відповідає сутність, має набір атрибутів, які характеризують його властивості. При цьому набір атрибутів повинен бути таким, щоби можна було розрізнити конкретні екземпляри сутності.

Приклад. Сутність **Викладач** може мати такі атрибути: Табельний номер, Прізвище, Ім'я, По батькові, Посада, Вчений ступінь.

Набір атрибутів, що однозначно ідентифікує конкретний екземпляр сутності, називають ключовим. У наведеному прикладі для сутності **Викладач** ключем буде Табельний номер, оскільки для всіх викладачів табельні номери різні.

Екземпляром сутності **Викладач** буде опис конкретного викладача. Загальноприйняте позначення сутності – прямокутник.

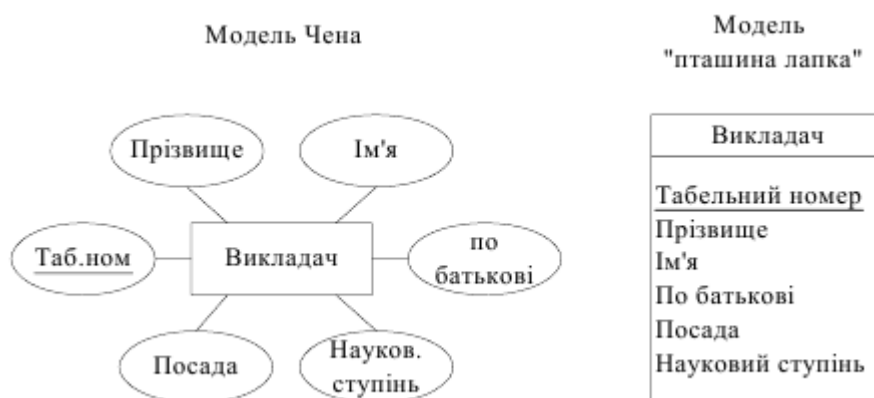


Рисунок. Представлення сутностей і атрибутів у ER-діаграмах П. Чена і ER-діаграмах "пташина лапка"

Зв'язки

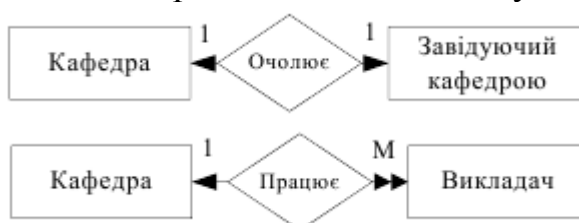
Між сутностями встановлюються зв'язки, які вказують яким чином сутності співвідносяться або взаємодіють між собою. Розрізняють такі зв'язки:

- між двома сутностями (бінарний зв'язок);
- між трьома сутностями (тернарний зв'язок);
- між N сутностями (N-арний зв'язок);
- між однією сутністю (рекурсивний зв'язок).

Найбільш поширеними є бінарні зв'язки. Зв'язок показує яким чином екземпляри сутностей зв'язані між собою. Бінарні зв'язки бувають:

- 1:1 (один до одного);
- 1:M (один до багатьох);
- N:M (багато до багатьох).

На рисунку показані відображення цих зв'язків у ER-моделях.





Атрибути

Атрибути являють собою властивості сутності. Значення кожного атрибута вибирають з відповідної множини значень, яка включає всі потенційні значення, які можуть бути присвоєні атрибуту. Ця множина значень називається доменом.

Приклад. Атрибут Оцінка може приймати чотири значення: 2, 3, 4, 5. Ці значення і складають домен цього атрибута.

Атрибути залежно від складності значень, які вони можуть приймати поділяються на певні категорії:

- простий - атрибут, який не може бути поділений на інші атрибути.
- складовий - атрибут, який може бути поділений на інші атрибути.
- однозначний - атрибут, який може приймати тільки одне значення.
- багатозначний - атрибут, який може приймати багато значень.
- похідний - атрибут, який не зберігається в БД, а обчислюється за допомогою певного алгоритму.



Атрибути номер залікової книжки, рік народження є простими.

Атрибути ПІБ і Адреса є складовими. ПІБ може бути поділений на атрибути: прізвище, ім'я, по батькові, а Адреса – на індекс, місто, вулиця, будівля, квартира.

Атрибут Вік є похідним: він обчислюється за значенням атрибута Рік народження (зображається пунктирною лінією).

Атрибут Номер залікової книжки є однозначним: він не може приймати два значення для одного студента.

Атрибут Номер телефону є багатозначним: він може приймати декілька значень для одного студента (зображається подвійною лінією).

Атрибут або набір атрибутів сутності, які застосовуються для ідентифікації екземпляра сутності, називаються **потенційним ключем**. Сутність може містити декілька потенційних ключів. В прикладі в якості потенційних ключів можуть бути такі атрибути: Номер залікової книжки, ПІБ.

Потенційний ключ, який вибрано для однозначної ідентифікації кожного екземпляра сутності певного типу, називається **первинним ключем**. Первинний ключ вибирається за умови гарантії унікальності його значень, а

також мінімальної довжини атрибутів, які входять в його склад. В прикладі в якості первинного ключа служить Номер залікової книжки.

Потужність зв'язків

Потужність зв'язку (кардинальність) відображає певне число екземплярів сутностей, які зв'язані з одним екземпляром зв'язаної сутності. В моделі Чена потужність зв'язку відображається вказівкою відповідних чисел поруч з сутностями у форматі (x, y). Перше число визначає мінімальне значення потужності зв'язку, а друге – його максимальне значення. Потужність вказує на число екземплярів у зв'язаній сутності.

Відомості про максимальне і мінімальне значення потужності зв'язку може застосовуватися у прикладному програмному забезпеченні або за допомогою тригерів; на рівні таблиць СУБД не може оперувати з потужністю зв'язків.

У моделі "пташина лапка" числовий діапазон значень потужності не відображається в ER-діаграмах.

Сильні і слабкі зв'язки

Якщо сутність може існувати незалежно від інших сутностей, то вона є незалежною від існування. Якщо сутність залежить від існування інших сутностей, то вона є залежною від існування. Наприклад, сутності Студент і Група можуть існувати незалежно одна від одної, а сутність Нагорода студента є залежною від сутності Студент й існувати без неї не може.

Якщо одна сутність незалежна від існування іншої сутності, то зв'язок між ними називається неідентифікаційним зв'язком або слабким зв'язком.

Ідентифікаційний зв'язок або сильний зв'язок має місце у тому випадку, коли одна зв'язана сутність залежить від існування іншої.

Обов'язкові і необов'язкові зв'язки

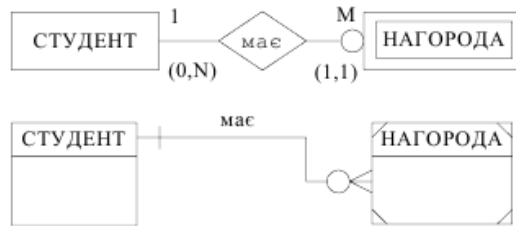
Участь сутності у зв'язку може бути обов'язковою або необов'язковою. Якщо один екземпляр сутності не потребує наявності відповідного екземпляра сутності в окремому зв'язку, то участь сутності у зв'язку є необов'язковою. Наприклад, у зв'язку сутностей Студент-Нагорода – не кожен студент має нагороди. Тобто не кожен екземпляр в таблиці Студент потребує обов'язкової наявності екземпляра сутності в таблиці Нагорода. Сутність Нагорода розглядається як необов'язкова по відношенню до сутності Студент. Необов'язкова сутність позначається невеликим колом з боку необов'язкової сутності.

Слабкі сутності

Слабкою сутністю називається сутність, яка задовольняє таким умовам:
– залежності від існування сутності з якою вона зв'язана;

– первинний ключ цієї сутності частково або повністю отримані з іншої сутності.

Слабка сутність на діаграмі Чена позначається подвійним прямокутником, а на діаграмі "пташина лапка" невеликими сегментами в кожному з кутів прямокутника.



Приклад. Сутність Нагорода є слабкою по відношенню до сутності Студент: вона залежить від існування цієї сутності і в її первинний ключ входить первинний ключ сутності Студент.

Складні зв'язки

Використання зв'язків більш високого порядку дає можливість у багатьох випадках краще відобразити семантику проблемної області.



Приклад. Сутності Викладач, Дисципліна і Екзамен утворюють тернарний зв'язок.

Проблеми побудови моделей "сутність – зв'язок"

При недостатньому розумінні суті встановлених зв'язків може бути створена модель, яка не буде повною мірою відображати зв'язки між реальними об'єктами. Визначають дефекти з'єднання, які виникають при невірній інтерпретації змісту деяких зв'язків: дефекти розгалуження і дефекти розриву.

Дефекти розгалуження мають місце, коли модель вірно відображає зв'язки між сутностями, але шлях між окремими сутностями визначений неоднозначно. Цей дефект виникає в тому випадку, коли два або більше зв'язків типу 1:М виходять з однієї сутності. Усунути цю проблему можна шляхом перебудови моделі для представлення вірної взаємодії цих сутностей.

Дефект розриву виникає у разі, коли існує один або декілька зв'язків з мінімальною потужністю рівною 0, яка визначає необов'язкову участь, і ці зв'язки складають частину шляху між взаємозв'язаними сутностями. Усунути цю проблему можна шляхом введення додаткового зв'язку між сутностями.

Логічне проектування баз даних

Логічне проектування виконується для певної моделі даних. Для реляційної моделі даних логічне проектування полягає у створенні реляційної схеми, визначенні числа і структури таблиць, формуванні запитів до БД, визначенні типів звітних документів, розробці алгоритмів обробки інформації, створенні форм для вводу і редагування даних в БД і рішенні цілого ряду інших задач. Концептуальні моделі за певними правилами перетворюються в логічні моделі даних.

Коректність логічних моделей перевіряється за допомогою правил нормалізації, які дозволяють переконатися в структурній узгодженості, логічній цілісності і мінімальній збитковості прийнятої моделі даних. Модель також перевіряється з метою виявлення можливостей виконання транзакцій, які будуть задаватися користувачами.

Проектування являє собою циклічний процес. Етапи логічного проектування наведені на рисунку.



Рисунок. Етапи логічного проектування бази даних

Спрощення концептуальної моделі

Першим кроком спрощення концептуальної моделі є попередні перетворення з метою усунення зв'язків, які є несумісними з реляційною моделлю.

На цьому етапі виконуються такі операції:

- вилучення двосторонніх зв'язків M:N;
- вилучення складних зв'язків;
- вилучення багатозначних атрибутів;
- вилучення рекурсивних зв'язків;
- вилучення зв'язків з атрибутами.

Вилучення двосторонніх зв'язків "багато до багатьох"

Перетворення зв'язку "багато до багатьох" виконується шляхом введення проміжної сутності із заміною одного зв'язку M:N двома зв'язками 1:N з новою сутністю.

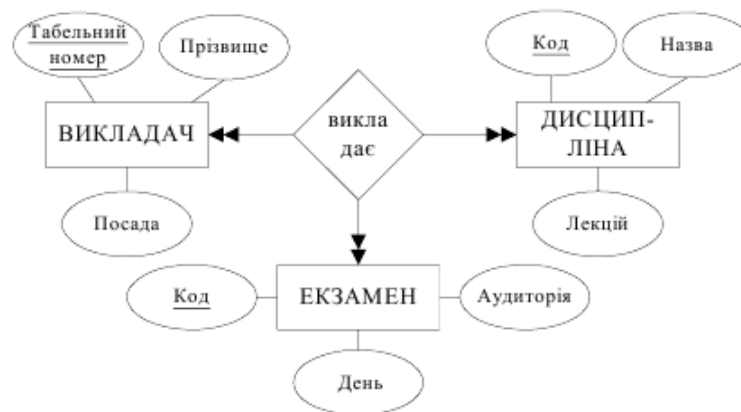
Приклад. **Викладач** може викладати багато **Дисциплін**, одну **Дисципліну** викладає багато **Викладачів**. У результаті перетворення отримується нова сутність **Викладання**, яка є слабкою і залежить від двох інших сутностей. Її первинний ключ складається з первинних ключів двох сутностей, а кожен атрибут окремо є вторинним ключем.

Вилучення складних зв'язків

Для вилучення складних зв'язків виконуються такі операції:

- у модель вводиться нова сутність;
- складний зв'язок замінюється бінарними зв'язками "один до багатьох" зі знов створеною сутністю;
- кількість бінарних зв'язків дорівнює ступеню складності зв'язку.

Приклад. **Викладач** може викладати багато **Дисциплін**, одну **Дисципліну** викладає багато **Викладачів**. З **Дисципліни** **Викладач** проводить **Екзамен**.



Вилучення багатозначних атрибутів

Якщо в концептуальній моделі даних присутній багатозначний атрибут, то може бути виконана декомпозиція цього атрибуту для визначення деякої сутності.

Приклад. Студент може мати декілька телефонів.

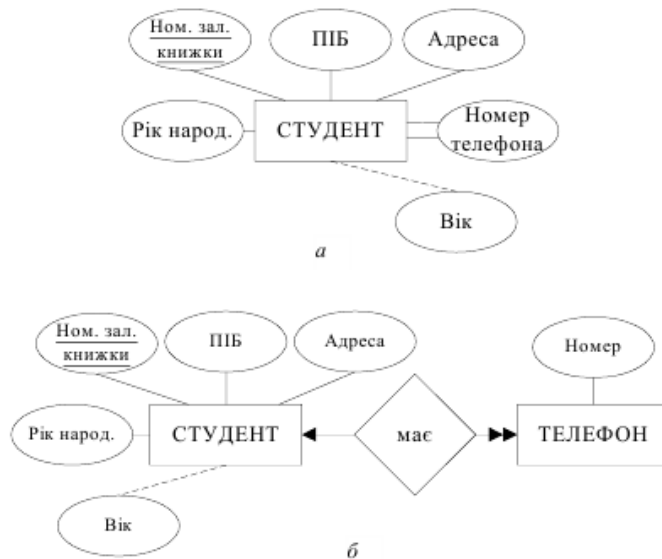


Рисунок. Вилучення багатозначного атрибуту: а – сутність Студент з багатозначним атрибутом Номер телефону; б – нова сутність Телефон

Вилучення рекурсивних зв'язків

На етапі спрощення концептуальної моделі рекурсивні зв'язки 1:1 і 1:М можуть бути перетворені у одне відношення. У випадку, коли є необов'язкова сутність з боку "багато" для зв'язку 1:М для зменшення пустих значень створюється нове відношення. Зв'язок М:М перетворюється на дві сутності.

Вилучення зв'язків з атрибутами

Вилучення зв'язків з атрибутами виконується шляхом додавання у модель нової сутності для відношення М:М з атрибутами зв'язку. Для відношення 1:М атрибути зв'язку передаються у сутність "багато" без створення нової сутності.

Методика перетворення ER-діаграм в реляційні структури

Для ER-моделі існує алгоритм однозначного перетворення її в реляційну модель даних.

Розглянемо правила перетворення ER-моделі в реляційну модель.

Сутності і атрибути

Для кожної сутності створюється відношення, кожен атрибут сутності стає атрибутом відповідного відношення.

Для сильних сутностей первинний ключ сутності стає PRIMARY KEY (PK) відповідного відношення.

Для слабких сутностей первинний ключ частково або повністю залежить від ключа сутності володаря (декількох володарів), тобто PK визначається тільки тоді, коли визначені всі PK сутностей володарів.

Зв'язки

Після перетворення концептуальної моделі залишаються такі типи зв'язків:

- "один до одного";
- "один до багатьох";

- рекурсивні зв'язки;
- суперклас – підклас.

Для кожного типу зв'язку залежно від умов зв'язування існують свої різновиди. Зв'язки між відношеннями в реляційній моделі реалізуються шляхом використання первинних і зовнішніх ключів.

Перевірка відношень за допомогою правил нормалізації

Створений на попередніх етапах набір відношень логічної моделі БД повинен бути перевірений на коректність об'єднання атрибутів у кожному відношенні. Перевірка виконується шляхом застосування до кожного відношення процедури послідовної нормалізації. Нормалізація гарантує, що отримана модель не буде мати протиріччя і буде мати мінімальну збитковість. Атрибути в результаті нормалізації будуть згруповані відповідно до існуючих між ними логічних зв'язків.

Для забезпечення коректності логічної моделі, у разі виявлення відношень, які не відповідають вимогам нормалізації, необхідно повернутися на попередні етапи проектування і перебудувати помилково створені елементи моделі.

Перевірка відповідності відношень вимогам транзакцій користувачів

Перевірка полягає в нанесенні безпосередньо на ER-діаграму всіх шляхів, які потрібні для виконання кожної з транзакцій. Якщо таким чином вдається виконати всі транзакції, то перевірка на цьому завершується. У протилежному випадку необхідно повернутися до попередніх етапів і перевірити, а у разі потреби і змінити ті фрагменти моделі, які не відповідають необхідній роботі транзакцій.

Якщо в результаті перевірки будуть виявлені області, які не беруть безпосередньої участі у роботі транзакцій, то можливо їх видалення з моделі.

Перевірка підтримки цілісності

Обмеження цілісності запобігають появі в БД суперечливих даних. Вирішення цієї проблеми на стадії проектування полягає у такому:

- наявність обов'язкових і необов'язкових значень даних для атрибутів (NULL, NOT NULL);
- наявність обмежень для доменів атрибутів (визначення області значень або діапазону значень);
- цілісність сутностей (обов'язкова наявність Primary Key в кожному відношенні);
- посилкова цілісність (зв'язування таблиць за допомогою Foreign Key);
- обмеження предметної області (бізнес правила), які реалізуються як засобами БД, так і на рівні застосувань.

Нормалізація баз даних

Нормалізація – це процедура визначення того, які атрибути зв'язані у відношенні. Одна з головних задач при розробці реляційної БД – об'єднання в одному відношенні тих атрибутів, які зв'язані між собою (між якими є функціональні залежності). Нормалізація являє собою поетапний процес заміни сукупності відношень іншою сукупністю (схемою), в якій відношення мають просту і регулярну структуру.

Результатом нормалізації є логічна модель БД.

Надлишковість даних в БД є небажаним явищем, оскільки призводить до збільшення об'єму пам'яті, уповільнює роботу БД. Надлишковість даних є результатом в першу чергу дублювання даних. Розрізняють незбиткове та збиткове дублювання даних. Повністю усунути надлишковість не потрібно, оскільки при цьому неможливо буде підтримувати БД як єдине ціле. Слід тільки мінімізувати надлишковість, залишивши необхідне дублювання даних.

Дублювання даних створює проблеми при виконанні операцій з БД. Ці проблеми виникають при спробі зробити операції: редагування, додавання або вилучення даних.

Аномаліями називається така ситуація в БД, яка призводить до протиріч у БД, або суттєво ускладнює обробку даних. Розрізняють аномалії модифікації, додавання і вилучення.

Приклад. Розглянемо відношення **Студент**.

Студент

Номер залікової книжки	Прізвище	Група	Факультет	Декан
1010	Бойко	ІТП-31	АІТ	Барков
2020	Лемешко	ІУСТ-22	АІТ	Барков
1030	Шевченко	ІТП-31	АІТ	Барков
1121	Петренко	БМО-32	АІТ	Барков
2231	Грицюк	ТБ-21	БТ	Тимчук

Аномалія модифікації виникає при спробі змінити прізвище декана. В цій ситуації необхідно переглянути всі кортежі. При великих розмірах БД це потребує значного часу, при цьому можливі помилки (у разі невірною введення прізвища), які порушують цілісність БД.

Аномалія додавання виникає при додаванні інформації про нового студента, при цьому необхідно вводити інформацію, яка вже є в БД: назва факультету, прізвище декана. Крім того неможливо створити нову групу поки не існує студентів, які в ній займаються.

Аномалія вилучення виникає при спробі вилучити дані про студента, який в групі поки ще один, наприклад Лемешко. В цьому випадку зникне інформація про групу ІУСТ-22.

Виконання декомпозиції наведеного відношення дозволяє позбутися вищезначених аномалій.

Номер залікової книжки	Прізвище	Група
1010	Бойко	ІТП-31
2020	Лемешко	ІУСТ-22
1030	Шевченко	ІТП-31
1121	Петренко	БМО-32
2231	Грицюк	ТБ-21

Група	Факультет
ІТП-31	АІТ
ІУСТ-22	АІТ
БМО-32	АІТ
ТБ-21	БТ

Факультет	Декан
АІТ	Барков
БТ	Тимчук

Процес проектування БД з використанням декомпозиції являє собою процес послідовної нормалізації схем відношень, при цьому кожна наступна ітерація відповідає нормальній формі більш вищого рівня і має кращі властивості у порівнянні з попередньою. Кожній нормальній формі (НФ) відповідає деякий набір обмежень.

Визначають такі нормальні форми: 1НФ, 2НФ, 3НФ, НФБК (нормальна форма Бойса-Кодда), 4НФ, 5НФ.

При виконанні декомпозиції зберігається множина вихідних функціональних залежностей між атрибутами і виконується зворотність. Зворотність означає можливість відновлення вихідної схеми. Функціональні залежності відображають зв'язки між атрибутами, які властиві реальному об'єкту.

Нормальні форми

Перша нормальна форма. Відношення знаходиться в 1НФ тоді і тільки тоді, коли всі його атрибути є атомарними.

Значення атрибуту вважається атомарним, якщо воно є неподільним у всіх застосуваннях.

Друга нормальна форма. Відношення знаходиться в 2НФ, якщо воно знаходиться в 1НФ і кожен його непервинний атрибут функціонально повно залежить від первинного ключа.

Неповною функціональною залежністю називається залежність неключового атрибуту від частини ключа, що складається з декількох атрибутів.

Повна функціональна залежність передбачає залежність неключового атрибуту від всіх атрибутів одночасно, що входять до складу ключа.

Третя нормальна форма. Відношення знаходиться в 3НФ, якщо воно знаходиться в 2НФ і жоден з непервинних атрибутів у відношенні не є транзитивно залежним від первинного ключа.

Атрибут С транзитивно залежить від атрибуту А, якщо для атрибутів А, В, С виконуються такі умови $A \rightarrow B$ і $B \rightarrow C$, але зворотня залежність відсутня.

Нормальна форма Бойса-Кодда. Відношення знаходиться в НФБК, якщо воно знаходиться в 3НФ і у ньому відсутні залежності атрибутів первинного ключа від неключових атрибутів.

Четверта нормальна форма. Відношення знаходиться в 4НФ тоді і тільки тоді, коли у випадку існування багатозначної залежності $A \twoheadrightarrow B$ всі інші атрибути відношення функціонально залежать від А.

П'ята нормальна форма. Відношення знаходиться в 5НФ тоді і тільки тоді, коли будь-яка залежність з'єднання у відношенні виходить з існування деякого можливого ключа у відношенні.

Денормалізація

Денормалізація – модифікація реляційної моделі, при якій ступінь нормалізації модифікованого відношення стає нижче, ніж ступінь нормалізації щонайменше одного з вихідних відношень.

Денормалізація застосовується у тих випадках, коли нормалізована БД не задовольняє вимогам, що висуваються до продуктивності системи. Денормалізація може застосовуватися у таких випадках:

- об'єднання таблиць зі зв'язками "один до одного";
- дублювання неключових атрибутів у зв'язках "один до багатьох" для зменшення кількості з'єднань;
- дублювання атрибутів зовнішнього ключа у зв'язках "один до багатьох" для зменшення кількості з'єднань;
- дублювання атрибутів "багато до багатьох" для зменшення кількості з'єднань;
- створення таблиць з даних, що містяться в інших таблицях;
- введення груп полів, що повторюються.

Застосовуючи денормалізацію слід враховувати, що цей процес має такі негативні наслідки:

- призводить до появи аномалій БД;
- знижує гнучкість системи;
- може зменшити час на відповіді до БД, але при цьому уповільнює операції оновлення даних;
- може ускладнити фізичну реалізацію системи.