

Реляційна модель даних

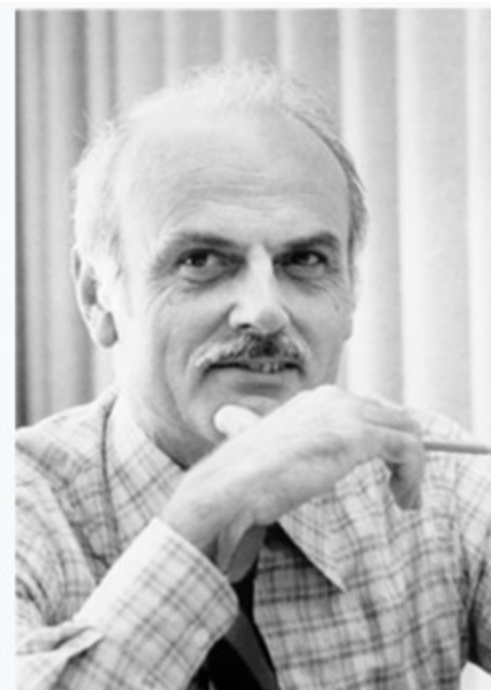
План

1. Базові поняття
2. 12 правил Кодда
3. Поняття ключа
4. Цілісність баз даних

Базові поняття

Базові поняття

- **Реляційна модель даних** — логічна модель даних.
- Вперше була запропонована британським ученим співробітником компанії IBM **Едгаром Франком Коддом (E. F. Codd)** в 1970 році в статті
- **«A Relational Model of Data for Large Shared Data Banks»**.



Народився	23 серпня 1923 ^[en] Острів Портленд ^[en] , Англія
Помер	18 квітня 2003 (79 років) Острів Вільямс, Флорида, США
Alma mater	Університет Мічигану
Галузь наукових інтересів	Комп'ютерні науки
Заклад	IBM
Відомий завдяки:	OLAP Реляційна модель
Нагороди	Премія Тюрінга

Базові поняття

- *Реляційна модель* даних заснована на математичному понятті відношення і представленні відношень у вигляді таблиць
- Сукупність схем реляційних відношень називають *схемою реляційного відношення*, або *реляційною схемою*.

Схема реляційного відношення має такі **властивості**:

- реляційне відношення має **ім'я**;
- імена атрибутів у межах схеми одного реляційного відношення мають бути **унікальними**;
- порядок атрибутів у схемі реляційного відношення **не є суттєвим**.

Базові поняття

Кортеж, що відповідає даній схемі відношення, -

- це множина пар {**назва атрибута, значення**}, яке містить одне входження кожної назви атрибута, що належить схемі відношення.

Відношення - це множина кортежів, які відповідають одній схемі відношення.

Відношення зазвичай записується у вигляді:

- $R(<A1: D1>, <A2: D2>, ..., <An: Dn>),$

Співробітники ($<\text{Номер}:12>, <\text{Прізвище}: \text{Устименко}>, <\text{Зарплата}: 1\$>$)

- або $R(A1, A2, ..., An).$

Співробітники (Номер, Прізвище, Зарплата)

Базові поняття

- **Екземпляр реляційного відношення** – є множина кортежів, а *кортеж* - це множина значень атрибутів.

Екземпляр відношення має такі **властивості**:

- порядок кортежів довільний;
- кортежі, як елементи множини, мають бути унікальними в межах реляційного відношення.
- **Домени** показують множину всіх можливих значень певного атрибута відношення.

Приклади доменів:

- домен «Курс» приймає значення від 1 до 4;
- домен «Вік» приймає значення від 14 до 100.

Базові поняття

- **Таблиця** - це поймаєменоване двовимірне зображення відношення;
- Вона складається з одного чи більше поймаєменованих стовпців і нуля або більше рядків.
- Назва таблиці відповідає імені реляційного відношення, імена стовпців - іменам атрибутів, а рядки - кортежам.

Базові поняття

Елементи реляційної моделі	Відповідний "табличний" термін
<i>База даних</i>	Набір таблиць
<i>Відношення</i>	Таблиця
<i>Кортеж</i>	Рядок таблиці
<i>Атрибут</i>	Заголовок стовпця таблиці
<i>Ключ</i>	Сукупність атрибутів, які унікально визначають кожен рядок таблиці, або виконують функції зв'язування таблиць, або дозволяють прискорити операції над таблицями
<i>Домен</i>	Множина значень атрибута
<i>Схема відношення</i>	Рядок заголовків стовпців таблиці
<i>Кардинальність</i>	Кількість рядків
<i>Степінь</i>	Кількість стовпці таблиці

Розклад руху автобусів як відношення



Базові поняття

Фундаментальні властивості відношень

- *відсутність кортежів-дублікатів*
- *відсутність упорядкованості кортежів*
- *відсутність упорядкованості атрибутів*
- *атомарність значень атрибутів*

Базові поняття

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).

Базові поняття

Реляційна БД включає в себе такі складові:

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

Базові поняття

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 предикату, 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.).

12 правил Кодда

12 правил Кодда

- **12 правил Кодда (0-12)** — набір 13 правил спроектовані для визначення того чи є СКБД реляційною.
- Правила настільки суворі, що всі популярні так звані «реляційні» **СКБД не відповідають багатьом критеріям**. Особливо складні 6, 9, 10, 11 і 12 правила.

12 правил Кодда

0. Фундаментальне правило (Foundation Rule)

- Система, яка позиціонується як реляційна система управління базами даних, повинна бути здатна управляти базами даних, використовуючи виключно свої реляційні можливості (зв'язки між даними)

12 правил Кодда

1. Інформаційне правило (Information Rule)

- Інформація має бути представлена у вигляді даних, що зберігаються в комірках.
- Дані, що зберігаються у комірках, мають бути атомарними.
- Порядок рядків у реляційній таблиці не повинен впливати на зміст даних і їх обробку.

12 правил Кодда

2. Правило гарантованого доступу (Guaranteed Access Rule)

- Доступ до даних має бути вільним від двозначності.
- До кожного елементу даних має бути гарантований доступ за допомогою комбінації імені таблиці, первинного ключа рядку й імені стовпця.

12 правил Кодда

3. Систематична обробка Null-значень

(Systematic Treatment of Null Values)

- Невідомі значення NULL, відмінні від будь-якого відомого значення, мають підтримуватись для всіх типів даних при виконанні будь-яких операцій.
- Наприклад, для числових даних невідомі значення не повинні розглядатись як нулі, а для символьних даних — як порожні рядки.

12 правил Кодда

4. Правило доступу до системного каталогу на основі реляційної моделі

(Dynamic On-line Catalog Based on the Relational Model)

- Словник даних має зберігатись у формі реляційних таблиць, і СУБД повинна підтримувати доступ до нього за допомогою стандартних мовних засобів, тих самих, що використовуються для роботи з реляційними таблицями, які містять дані користувача.

12 правил Кодда

5. Правило повноти підмови маніпулювання даними (Comprehensive Data Sublanguage Rule)

- Система управління реляційними базами даних має підтримувати хоча б одну реляційну мову, яка
 - а) має лінійний синтаксис,
 - б) може використовуватись інтерактивно і в прикладних програмах,
 - в) підтримує операції визначення даних, визначення уявлень, маніпулювання даними (інтерактивні та програмні), обмежувачі цілісності, управління доступом та операції управління транзакціями (begin, commit і rollback).

12 правил Кодда

6. Правило модифікації відображень (View Updating Rule)

- Кожне відображення має підтримувати усі операції маніпулювання даними, які підтримують реляційні таблиці: операції вибірки, вставки, модифікації і видалення даних.

12 правил Кодда

7. Правило високорівневих операцій модифікації даних (High-level Insert, Update, and Delete)

- Операції вставки, модифікації і видалення даних мають підтримуватись не тільки щодо одного рядку реляційної таблиці, але й будь-якої їх кількості.

12 правил Кодда

8. Правило фізичної незалежності даних (Physical Data Independence)

- Додатки не повинні залежати від використовуваних способів зберігання даних на носіях, від апаратного забезпечення комп'ютерів, на яких знаходиться реляційна база даних.

12 правил Кодда

9. Правило логічної незалежності даних (Logical Data Independence)

- Представлення даних в додатку не повинно залежати від структури реляційних таблиць.
- Якщо в процесі нормалізації одна реляційна таблиця розділяється на дві, подання має забезпечити об'єднання цих даних, щоб зміна структури реляційних таблиць не позначалась на роботі додатків.

12 правил Кодда

10. Правило незалежності контролю цілісності (Integrity Independence)

- Вся інформація, необхідна для підтримки цілісності, має бути у словнику даних.
- Мова для роботи з даними має виконувати перевірку вхідних даних і автоматично підтримувати цілісність даних.

12 правил Кодда

11. Правило незалежності від розміщення (Distribution Independence)

- База даних може бути розподіленою, може перебувати на кількох комп'ютерах, і це не повинно впливати на додатки, як і перенесення бази даних на інший комп'ютер.

12 правил Кодда

12. Правило узгодженості мовних рівнів (The Nonsubversion Rule)

- Якщо використовується низькорівнева мова доступу до даних, вона не повинна ігнорувати правила безпеки і правила цілісності, які підтримуються мовою більш високого рівня.

Поняття ключа

Поняття ключа

- Множина атрибутів, що однозначно ідентифікують кортежі реляційного відношення, називається **ключем**.
- Відношення може мати декілька різних ключів.

Наприклад

Студент(Код, ПІБ, Адреса)

- Код,
- пара (ПІБ. Адреса)
- три атрибути (Код, ПІБ. Адреса).
- Ключ називається **простим**, якщо складається з одного атрибута (Код), і **складеним** - якщо з кількох атрибутів, наприклад (ПІБ. Адреса).

Поняття ключа

- Ключ називають **надлишковим**, якщо певна його підмножина також є ключем.
- Наприклад, ключ (Код. ПІБ. Адреса) є надлишковим тому, що містить атрибут Код, який також є ключем.
- Ключ, що не є надлишковим, називають **мінімальним**.
- **Суперключ** — підмножина атрибутів відношення, яка задовольняє вимогу унікальності: не існує двох кортежів даного відношення, в яких значення цієї підмножини атрибутів збігаються (рівні).
- Реляційне відношення може мати багато можливих ключів, але тільки один із них є **первинним**.

Поняття ключа

Первинний ключ має такі **властивості**:

- кожне реляційне відношення має один і **лише один первинний ключ**;
- значення всіх атрибутів первинного ключа **не можуть бути невизначеними**, оскільки він має унікально ідентифікувати всі кортежі будь-якого екземпляра реляційного відношення;
- значення первинного ключа **не можуть повторюватися**, але допускаються повторення значень частини складеного первинного ключа;
- значення первинного ключа **не впливають на порядок кортежів** у табличному зображенні реляційного відношення;
- первинний ключ **не впливає на доступ до кортежів**, який може бути здійснено за значенням будь-якого атрибута чи набору атрибутів незалежно від того, чи є він первинним ключем.

Поняття ключа

- Сукупність атрибутів, що є первинним ключем іншого реляційного відношення, називається **зовнішнім (стороннім) ключем**.
- За допомогою зовнішніх ключів у реляційній моделі встановлюються зв'язки між реляційними відношеннями.

Зовнішні ключі мають такі **властивості**:

- значення зовнішнього ключа **завжди посилається** на певне значення відповідного **первинного ключа**, тобто будь-яке значення зовнішнього ключа має бути значенням первинного ключа іншого відношення;
- значення зовнішнього ключа, на відміну від значень первинного ключа, можуть бути **невизначеними й повторюватися** в межах реляційного відношення

Розглянемо приклад

- ВИКЛАДАЧ(Код_вик, ПІБ, Адреса, Код_від).
ВІДДІЛЕННЯ(Код_відділення, Назва, Керівник, Кабінет).
- Первинним ключем
 - відношення ВИКЛАДАЧ є Код_вик,
 - відношення ВІДДІЛЕННЯ – Код_відділення; цей самий атрибут у реляційному відношенні ВИКЛАДАЧ є зовнішнім ключем.

Цілісність баз даних

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.

Цілісність баз даних

Структурна цілісність передбачає :

- 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).

Цілісність баз даних

- **Обмеження реальних значень даних** вимагають, щоби значення поля належали деякому діапазону значень, або задовольняли певному арифметичному співвідношенню між значеннями різних полів.

Цілісність баз даних

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.

Правила вилучення і оновлення

Операція	Правило	Пояснення
Вилучення (DELETE)	RESTRICT Обмеження	Заборона вилучення рядка з батьківської таблиці, якщо в підлеглій таблиці цей рядок має нащадків
	CASCADE	При вилученні рядка з батьківської таблиці в підлеглій таблиці всі рядки-нащадки автоматично вилучаються
	SET NULL	При вилученні рядка з батьківської таблиці в підлеглій таблиці всім зовнішнім ключам рядків-нащадків автоматично присвоюється значення NULL
	SET DEFAULT	При вилученні рядка з батьківської таблиці в підлеглій таблиці всім зовнішнім ключам рядків-нащадків автоматично присвоюється певне значення, встановлене за замовчуванням

Правила вилучення і оновлення

Оновлення (UPDATE)	RESTRICT	Заборона зміни первинного ключа в рядку батьківської таблиці, якщо в підлеглий таблиці цей рядок має нащадки
	CASCADE	При зміні первинного ключа в рядку батьківської таблиці в підлеглий таблиці відповідні значення зовнішнього ключа також автоматично змінюються у всіх рядках-нащадках для того, щоби відповідати новому значенню первинного ключі
	SET NULL	При зміні первинного ключа в рядку батьківської таблиці в підлеглий таблиці відповідні значення зовнішнього ключа також автоматично змінюються у всіх рядках-нащадках і їм присвоюється значення NULL
	SET DEFAULT	При зміні первинного ключа в рядку батьківської таблиці в підлеглий таблиці відповідні значення зовнішнього ключа також автоматично змінюються у всіх рядках-нащадках і їм присвоюється певне значення, встановлене за замовчуванням

Правила вилучення і оновлення

- Правила NONE - не виконуються ніякі дії і правила NULL ALLOWED - дозволяються невизначені значення.
- При введенні нових рядків (INSERT) необхідно дотримуватися такої послідовності введення: спочатку дані вводяться в батьківську таблицю, а потім - в підлеглу.