Лекція 7. Цілі проектування. Нормалізація відношень.

Проектування БД

Насамперед відзначимо найбільш важливі цілі проектування БД [6].

1. Можливість збереження всіх необхідних даних у БД.

Ця мета є очевидною і ще раз підкреслює необхідність побудови концептуальної схеми БД, що включає всі сутності чи об'єкти предметної області, їхні атрибути і визначення відносин між ними.

2. Виключення надмірності даних.

Для розуміння цієї мети визначимо розходження між дублюванням даних і надлишковим дублюванням або надмірністю. Для цього розглянемо відношення *R* з попереднього розділу.

У цьому відношенні значення 'a' і 'b' атрибута 'C' повторюються (дублюються). Однак якщо видалити значення 'a', наприклад, з першого кортежу, то відновити цей кортеж ми вже не зможемо. Якщо ж видалити значення 'b' з одного чи навіть з усіх, крім одного, кортежів, то його можна відновити за значенням 'w' атрибута A, і навпаки. У першому випадку ми маємо необхідне дублювання, у другому — надлишкове.

R

A	B	C
x	1	a
y	2	a
z	3	a
~~w~~	4	~~b~~
~~w~~	5	~~b~~
w	6	b

Отже, якщо існує хоча б одна проекція на *неодиничну* підмножину схеми відношення, що приводить до *зменшення потужності* цього відношення, то таке відношення містить *надлишкові дані*.

Виключити надмірність можна, наприклад, шляхом розбивки вихідного відношення на декілька (2 чи більше):

A	B
x	1
y	2
z	3
w	4
w	5
w	6

A	C
x	a
y	a
z	a
w	b

У тому випадку, якщо зазначена у визначенні проекція зменшує потужність відношення не менш, ніж у два рази, то розбивка дозволяє *скоротити обсяг пам'яті*, необхідний для збереження інформації.

Однак більш важливим є те, що усунення надмірності дозволяє значною мірою спростити процедури додавання і видалення даних.

На вирішення цієї проблеми орієнтована і третя мета проектування:

3. Нормалізація відношень.

Під *нормалізацією* розуміється розбивка (у загальному випадку, перетворення) відношення на два чи більше для приведення його у відповідність до вимог так званих **нормальних форм**.

Розглянемо відношення R, представивши його таким чином:

R’

<table>
<tr><th colspan="2">AB</th><th>C</th></tr>
<tr><td>A</td><td>B</td><td rowspan="4">a</td></tr>
<tr><td>x</td><td>1</td></tr>
<tr><td>y</td><td>2</td></tr>
<tr><td>z</td><td>3</td></tr>
<tr><td>A</td><td>B</td><td rowspan="4">в</td></tr>
<tr><td>w</td><td>4</td></tr>
<tr><td>w</td><td>5</td></tr>
<tr><td>w</td><td>6</td></tr>
</table>

З математичної точки зору відношення *R’* цілком коректне: воно складається з двох кортежів, атрибут *AB* яких набуває значень, що, у свою чергу, дорівнюють деяким відношенням. Значення же атрибута С *атомарні*, тобто неподільні.

Відношення, *усі* значення якого атомарні, називається *нормалізованим* або представленими у *першій нормальній формі*: 1НФ. У даному випадку ці два поняття є синонімами.

Отже, *R* і *R’* — це, за суттю, однакові відношення, тільки *R* — нормалізоване, а *R’* — ненормалізоване.

Однак реляційна модель допускає тільки атомарні значення атрибутів. Тобто, ніякий домен не може містити значення-відношення. Тому БД, представлена в цій моделі, завжди повинна бути нормалізована.

Крім того, необхідно враховувати, що необмежений зріст числа відношень може привести до такого збільшення кількості і складності зв’язків між відношеннями, що зажадають значного ускладнення запитів до СБД, що, у свою чергу, знизить її ефективність. Щоб цього уникнути, необхідно знайти компроміс у досягненні третьої і четвертої цілей:

4. Зведення числа збережених у БД відношень до мінімуму.

Нормалізація відношень

Отже, для усунення надмірності даних, зниження імовірності виникнення проблем або *аномалій* при вставці, відновленні або видаленні інформації виконується приведення відношення до нормальної форми, що забезпечує розв’язання цих задач.

Для досягнення кожного рівня нормалізації, що поліпшує конструкцію бази даних, застосовуються визначені набори правил. Причому, реалізується це шляхом розбивки відношення або *декомпозиції*, в основі якої лежить концепція *функціональних залежностей* (ФЗ) між атрибутами відношення.

Нехай дане відношення *R*, а *X* та *Y* — правильні підмножини його схеми відношення. Говорять, що *Y функціонально залежить від X* ($X \to Y$ [*читається: „X функціонально визначає Y“ чи „X стрілка Y“*]) тоді і тільки тоді, коли для кожного припустимого значення множини *X* існує рівно одне значення множини *Y*.

Інакше кажучи, якщо два кортежі відношення R збігаються за значенням *X*, то вони збігаються за значенням *Y*. Ліву і праву частини ФЗ називають **детермінантом** і **залежною частиною** відповідно.

Якщо виконується зворотне співвідношення, то множини атрибутів є **взаємозалежними**: $X \leftrightarrow Y$. Приклад:

R: $A \to C, B \to C, \{A, B\} \to C, B \to A, \{B, C\} \to A$

S: $D \leftrightarrow E$

На сьогоднішній день розроблено 5 нормальних форм [4] — *1НФ - 5НФ*, що накладають на відношення зростаючі за жорсткістю обмеження.

Для переважної більшості систем досить привести відношення до *3НФ* чи її модифікації — *НФ Бойса-Кодда* (НФБК), що забезпечує *декомпозицію без втрат*.

Тому зупинимося детальніше на цій формі.

Нормальна форма Бойса-Кодда

Відношення знаходяться в *НФБК* тоді і тільки тоді, коли *кожен* детермінант є потенційним ключем.

Алгоритм декомпозиції без втрат, що приводить відношення до НФБК, такий:

1. Визначається ФЗ, що є причиною того, що відношення не знаходиться в НФБК.
2. Виконується проекція відношення на детермінант і залежну частину знайденої ФЗ.
3. Зі схеми вихідного відношення видаляються атрибути, що входять у залежну частину ФЗ.
4. Відношення, отримані в п. 2 і 3, і будуть результатом декомпозиції. Ці відношення перевіряються на відповідність НФБК, і, при необхідності, алгоритм повторюється.

Приклад. Розглянемо відношення *R*, ФЗ якого ми уже визначили вище. Випишемо детермінанти і потенційні ключі і визначимо, за якими елементами ці множини не збігаються. При цьому для наочності прикладу ми не будемо звертати увагу на властивість не надмірності потенційного ключа.

Детермінанти: $A, B, \{A, B\}, \{B, C\}$

Потенційні ключі: $B, \{A, B\}, \{B, C\}$

Видно, що атрибут *A* є детермінантом, але не є потенційним ключем. Йому відповідає ФЗ $A \to C$.

Виконаємо проекцію відношення R на атрибути A та C: $R'' = R\left[A, C\right]$. З відношення R видаляємо атрибут C: $R' = R\left[A, B\right]$.

R’

A	**B**
x	1
y	2
z	3
w	4
w	5
w	6

R"

A	**C**
x	a
y	a
z	a
w	b

Для перевірки того факту, що зроблено декомпозицію без втрат, виконується природне з’єднання результуючих відношень, що має дати початкове.

У тому випадку, якщо ФЗ, що відповідають першому пункту алгоритму, декілька, то виникає проблема вибору ФЗ, залежна частина якої чи її підмножина є детермінантом іншої ФЗ.

Однією з методик вирішення проблеми є пошук „*ланцюжків ФЗ*“ виду $A \to B \to C$ з наступним використанням для декомпозиції крайньої правої ФЗ.

При використанні цієї методики ФЗ зручно представити на *діаграмі ФЗ* (рис. 1).

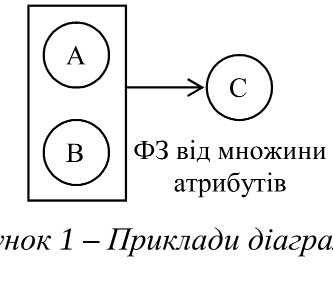

Рисунок 1 – Приклади діаграм ФЗ

Покажемо діаграму ФЗ відношення *R* (рис. 2):

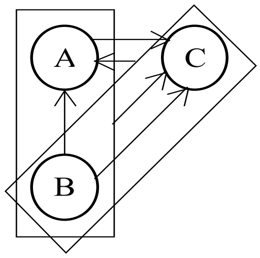

Рисунок 25 – Діаграма ФЗ відношення R1

Виділити на цій діаграмі ланцюжок ФЗ досить складно. Однак цю діаграму можна спростити, видаливши з усієї множини ФЗ *надлишкові ФЗ*. Тобто залежності, що не несуть інформації, яка *не могла б бути отримана* з інших ФЗ відношення.

Видалити надлишкові ФЗ можна, використовуючи два *правила виводу*. Нехай W, X, Y, Z — правильні підмножини схеми відношення R, а позначення XY відповідає $X \bigcup Y$.

1. **Транзитивність**. Якщо $X \to Y$ і $Y \to Z$, то $X \to Z$, яка називається *транзитивною* залежністю. Причому, **якщо всі три залежності входять у набір ФЗ відношення, то транзитивна залежність є надлишковою**.

2. Доповнення:

а) якщо $X \to Y$, то $XZ \to Y$;

б) якщо $X \to Y$, то $XZ \to YZ$.

Причому, **додаткові залежності в обох випадках є надлишковими.**

Повернемося до прикладу. Відповідно до правила *доповнення* виключаємо з набору ФЗ відношення R залежності $\{A, B\} \to C$ і $\{B, C\} \to A$. Отримаємо нову діаграму, на якій чітко видна *транзитивна* залежність $B \to C$. Її ми також виключаємо з розгляду (рис. 3).

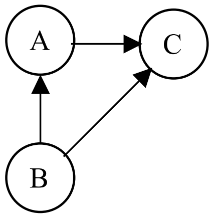

Рисунок 3 – Діаграма ФЗ транзитивної залежності

У результаті дістаємо ланцюжок ФЗ $B \to A \to C$, з якого беремо крайню праву залежність для декомпозиції.

Наведені правила є двома з трьох *правил* виводу або *аксіом Армстронга*.

3. Третє правило — **рефлексивність**: для будь-яких множин атрибутів X і Y: $XY \to X$ і $XY \to Y$.

Ця аксіома і кілька наступних правил виводу дозволяє спростити і зменшити початковий набір ФЗ.

4. **Декомпозиція**. Якщо $X \to YZ$, то $X \to Y$ і $X \to Z$.
5. **Об'єднання**. Якщо $X \to Y$ і $X \to Z$, то $X \to YZ$.
6. **Композиція**. Якщо $X \to Y$ і $Z \to W$, то $XZ \to YW$.