

Тема лекції 7:

Логічне проектування реляційних баз даних

- ❑ Надлишковість даних та аномалії
 - ❑ Функціональні залежності
 - ❑ Процес нормалізації
 - ❑ Алгоритм декомпозиції
відношення з метою його
нормалізації
-

Надлишковість даних та аномалії

- ❑ Надлишковість – це дублювання даних в базі, яке може викликати додаткове використання ресурсів для зберігання даних і синхронізації дублікатів
 - ❑ Наявність надлишковості даних при виконанні операцій над ними призводить до різних аномалій – порушення цілісності БД.
-

Аномалії модифікації даних

- ❑ **Аномалія введення** даних означає, що при спробі додати у відношення новий кортеж, необхідно додати й інший кортеж, який містить дублікат, і зв'язані з ним значення інших атрибутів
 - ❑ **Аномалія видалення** означає, що при спробі видалення кортежу, необхідно видалити й інший кортеж, який містить дублікат, який необхідно видалити, і зв'язані з ним значення інших атрибутів (по аналогії зі вставкою). Однак, при видаленні обох кортежів можуть втратитись дані, пов'язані з дублікатами
 - ❑ **Аномалія оновлення** означає, що при спробі змінити певне значення атрибута, необхідно його змінювати для всіх дублікатів.
-

Вирішення проблеми аномалії модифікації даних

- Для усунення аномалії модифікації даних використовується метод нормалізації відношень, який базується на поняттях
 - функціональної залежності (ФЗ)
 - нормальних форм (НФ)
-

Функціональні залежності (ФЗ)

- ❑ ФЗ між атрибутами відношень – це основа аналізу коректності схеми
 - ❑ ФЗ визначають не поточний стан БД, а всі її можливі стани
 - ❑ ФЗ відображають ті зв'язки між атрибутами, які притаманні реальному об'єкту, що моделюється за допомогою відношень
 - ❑ ФЗ є семантичною властивістю атрибутів відношення
 - ❑ Семантика відношення вказує , які його атрибути можуть бути пов'язані один з одним, а також визначає ФЗ між атрибутами у вигляді обмежень, що накладаються на деякі атрибути.
-

Визначення функціональної залежності

- Якщо у відношенні, що містить набір атрибутів A та B , атрибут B функціонально залежить від атрибуту A , що позначається $A \rightarrow B$, то для будь-якого значення атрибуту A зв'язане тільки з одне значення атрибуту B .
-

Детермінант відношення

- Якщо у відношенні існує декілька функціональних залежностей, то кожен атрибут, або набір атрибутів, від якого залежить будь-який інший атрибут, називається **детермінантом відношення**.
 - Іншими словами, якщо існує ФЗ $A \rightarrow B$ і атрибут B не залежить функціонально ні від якої підмножини атрибуту A , то A – детермінант для B .
-

Поняття нормалізації

- Нормалізація – це формальний метод аналізу відношень, що базується на використанні їх первинних і потенційних ключів та функціональних залежностях
 - Метод включає ряд правил, які використовуються для перевірки окремих відношень таким чином, щоб уся БД була нормалізована до необхідного степеню нормалізації
 - Якщо деяке відношення не виконує певну вимогу, то це відношення декомпозується на відношення, кожне з яких відповідає усім вимогам нормалізації.
-

Процес нормалізації

- ❑ Нормалізація виконується в декілька послідовних етапів, кожен з яких відповідає деякій нормальній формі
 - ❑ Під час нормалізації формат відношень стає більш строгим і менш вразливим до аномалій модифікації даних
 - ❑ Процес нормалізації починається з перетворення даних з формату введення даних у формат таблиці
 - ❑ На вихідному етапі таблиця перебуває у ненормалізованій формі (ННФ)
-

Приклад 1. "Контакти"

- Розглянемо способи зберігання контактних телефонів людей. Прийнято рішення ідентифікувати людину за прізвищем та ініціалами. Передбачається, що в однієї людини є декілька контактних телефонів.
 - Нехай на початку таблиця «Контакти» має вигляд наступної таблиці, яка містить одну або декілька груп даних, що повторюються (в нашому випадку – "Телефони"):
-

Приклад 1. Таблиця вхідних даних "Контакти"

Прізвище, ініціали	Телефони
Комарницький Н.Л.	2645432 0505664556
Азопов А.А.	2764544
Дулько К.Є.	2983840 2758567 0505557565
Бутько В.В.	2346784
Герасим М.П.	2765544
Васильченко О.П.	2541434 0676876786 0956786876
Рущинський Р.О.	2645654

Приклад 1. (продовж.)

- Для внесення даних створимо відношення «Контакти» з ключем Прізвище_Ініціали, яке представлене у вигляд наступної таблиці
 - Це відношення теж є ненормалізоване:
 - у ньому велика кількість NULL значень
 - куди вносити тип телефону – невідомо
 - атрибути Телефон_1, Телефон_2, Телефон_3 утворюють групу даних
-

Приклад 1. Ненормалізована таблиця "Контакти"

Прізви_Ініц	Телефон_1	Телефон_2	Телефон_3
Комарницький Н.Л.	2645432	0505664556	
Азопов А.А.	2764544		
Дулько К.Є.	2983840	2758567	0505557565
Бутько В.В.	2346784		
Герасим М.П.	2765544		
Васильченко О.П.	2541434	0676876786	0956786876
Рущинський Р.О.	2645654		

Перша нормальна форма (1НФ)

- Відношення перебуває у **першій нормальній формі** тоді і тільки тоді, коли кожен атрибут відношення є атомарним, тобто неподільним і скалярним.
 - Для перетворення ненормалізованої таблиці у 1НФ необхідно знайти і усунути усі групи атрибутів.
 - На перетину кожного рядка і кожного стовпця повинне міститися лише одне значення атрибуту
 - Реляційне відношення називається нормалізованим, якщо воно перебуває у 1НФ
-

Приклад 1. (продовж.)

- Для приведення відношення «Контакти» до 1НФ перебудуємо його, додаючи атрибут Тип_Тел, а замість групи атрибутів Телефон_1, Телефон_2, Телефон_3 введемо атрибут Телефон і внесемо його у ключ. Отримуємо відношення:
Контакти (Прізви_Ініц, Телефон, Тип_Тел).
-

Приклад 1. (продовж.) Таблиця в 1НФ

Прізви_Ініц	Телефон	Тип_Тел
Комарницький Н.Л.	2645432	дом
Комарницький Н.Л.	0505664556	моб
Азопов А.А.	2764544	дом
Дулько К.Є.	2983840	дом
Дулько К.Є.	2758567	роб
Дулько К.Є.	0505557565	моб
Бутько В.В.	2346784	роб
Герасим М.П.	2765544	дом
Васильченко О.П.	2541434	дом
Васильченко О.П.	0676876786	моб
Васильченко О.П.	0956786876	моб
Рущинський Р.О.	2645654	дом

Приклад 1. (резюме)

- Проблему груп даних в атрибутах вирішено
 - Щодо пункту у визначенні 1НФ про неподільність значень атрибута
 - Розглянемо ситуацію з мобільними телефонами. Якщо запити до таблиці Контакти виконуються з метою побачити інформацію для здійснення дзвінка певній людині, то атрибут Телефон є неподільним і скалярним, тобто атомарним.
 - Якщо ж необхідно за кодом мобільного визначити, який оператор обслуговує номер телефону (наприклад, з використанням функцій над стрічками в умовах пошуку або з використанням операції з'єднання), то номер вже не є неподільним, і для його складової, яка містить код оператора, необхідно утворювати окреме поле. При цьому, якщо нормалізацію доводити до певного рівня, то код оператора треба забрати з поля Телефон.
 - Відношення Контакти(Прізви_Ініц, Телефон, Тип_Тел) перебуває у 1НФ.
 - Для нього не виконується 2НФ, оскільки існує ФЗ типу Телефон → Тип_Тел.
-

Приклад 1. (альтернативне рішення)

- ❑ Задачу контактних телефонів можна вирішити ненормалізованим відношенням (**Nested Relation**), в якому допускаються відношення у складі відношень.
 - ❑ В SQL:1999 і SQL:2003 уведено спеціальні типи даних:
 - ARRAY – масив
 - MULTISSET – мільтимножина
 - ROW – запис,
 - ❑ Це порушує принцип 1НФ, згідно з яким будь-який стовпець таблиці не може містити структурованих даних
 - ❑ на практиці не раз приходиться відступати від теорії на користь ефективності
-

Повна і часткова ФЗ

- У деякому відношенні атрибут B **повно функціонально залежить** від атрибуту A , якщо атрибут B функціонально залежить від повного значення атрибуту A і не залежить від будь-якої підмножини атрибуту A .
 - **Частковою функціональною залежністю** називається така залежність $A \rightarrow B$, якщо в A є деякий атрибут, при видаленні якого ця залежність залишається.
-

Друга нормальна форма (2НФ)

- Базується на понятті повної функціональної залежності.
 - Застосовується до відношень зі **складеними первинними ключами**, оскільки відношення з простим первинним ключем завжди перебуває, принаймні, у 2НФ.
 - Відношенню, яке не є в 2НФ, може загрожувати аномалія модифікації даних.
 - Відношення перебуває у **другій нормальній формі** тоді і тільки тоді, коли воно перебуває у 1НФ і не містить часткових функціональних залежностей атрибутів від атрибутів первинного ключа.
-

Приклад 2. "Здача сесії"

- Структура відношення :
(ПІБ, НомЗалКн, Гр, Дисципл, Оц).
 - Таке відношення містить дублювання даних у полях ПІБ, НомЗалКн, Гр. У цьому відношенні також присутні аномалії видалення даних
 - Оскільки кожен студент здає цілий набір дисциплін, то первинним ключем може бути
(НомЗалКн, Дисципл),
 - який однозначно визначає кожний рядок даного відношення:
(ПІБ, Гр, НомЗалКн, Дисципл, Оц).
 - З іншої сторони, атрибути ПІБ та Гр залежать лише від частини первинного ключа – від НомЗалКн.
 - Тому можна констатувати **наявність часткових функціональних залежностей** в цьому відношенні.
-

Приклад 2. Приведення до 2НФ

- Для приведення відношення до 2НФ, необхідно зробити його декомпозицію так, щоб нові відношення містили детермінант але не містили часткових функціональних залежностей
 - У нашому прикладі такими відношеннями можуть бути:
(НомЗалКн, ПІБ, Гр) та
(НомЗалКн, Дисципл, Оц).
-

Транзитивна залежність

- Якщо для атрибутів A , B і C деякого відношення існують залежності вигляду $A \rightarrow B$ і $B \rightarrow C$, то атрибут C **транзитивно залежить** від атрибуту A через атрибут B при умові, що атрибут A функціонально не залежить ні від атрибуту B , ні від C .
-

Третя нормальна форма (3НФ)

- ❑ Базується на понятті транзитивної залежності
 - ❑ Відношення перебуває у **третій нормальній формі** тоді і тільки тоді, коли воно перебуває у 2НФ і не містить транзитивних залежностей.
 - ❑ Це означає: якщо у відношенні є транзитивна залежність між атрибутами, то такі атрибути необхідно перемістити в нове відношення разом з копією детермінанта.
-

Приклад 3. "Контингент студентів"

-
- Розглянемо відношення, яке пов'язує студентів з інститутами, групами та спеціальностями, на яких вони вчаться:
(ПІБ, НомЗалКн, Інст, Гр, Спец, Каф)
 - Первинним ключем відношення є НомЗалКн, отже маємо наступні залежності:
 - $\text{НомЗалКн} \rightarrow \text{ПІБ}$,
 - $\text{НомЗалКн} \rightarrow \text{Інст}$,
 - $\text{НомЗалКн} \rightarrow \text{Гр}$,
 - $\text{НомЗалКн} \rightarrow \text{Спец}$,
 - $\text{НомЗалКн} \rightarrow \text{Каф}$.
-

Приклад 3. (продовж.)

- Розглянемо інші функціональні залежності. Група, в якій вчиться студент, однозначно визначає інститут, спеціальність і кафедру. Маємо наступні залежності:
 - $\text{Гр} \rightarrow \text{Інст}$,
 - $\text{Гр} \rightarrow \text{Спец}$,
 - $\text{Гр} \rightarrow \text{Каф}$.
 - Крім того, кафедра однозначно визначає інститут: $\text{Каф} \rightarrow \text{Інст}$.
 - І якщо припустити, що одну спеціальність можуть випускати декілька кафедр (а так воно, як правило, є), то спеціальність не визначає однозначно кафедру
-

Приклад 3. Приведення до 3НФ

-
- Наведені залежності утворюють **транзитивні групи**:
 - $\text{НомЗалКн} \rightarrow \text{Гр}, \text{Гр} \rightarrow \text{Каф},$
 - $\text{НомЗалКн} \rightarrow \text{Каф}, \text{Каф} \rightarrow \text{Инст}$
 - Якщо у відношенні є транзитивна залежність між атрибутами, то такі атрибути необхідно перемістити в нове відношення разом з копією детермінанта.
 - У нашому прикладі це реалізується наступним набором відношень:
 $(\underline{\text{НомЗалКн}}, \text{ПІБ}, \text{Гр}, \text{Спец}),$
 $(\underline{\text{Гр}}, \text{Каф}), (\underline{\text{Каф}}, \text{Инст})$
-

Нормальна форма Бойса-Кодда (НФБК)

- 2НФ – знайти та усунути часткові залежності від **первинного ключа**
 - 3НФ – знайти та усунути транзитивні залежності від **первинного ключа**
 - НФБК знайти та усунути такі ж залежності від **потенційних ключів**, якщо такі є.
 - Для відношення з єдиним потенційним ключем його 3НФ і НФБК є еквівалентними.
-

Нормальна форма Бойса-Кодда (НФБК)

- Відношення перебуває у **нормальній формі Бойса-Кодда** тоді і тільки тоді, коли кожен його детермінант є потенційним ключем.
 - НФБК є строгою версією 3НФ, оскільки кожне НФБК-відношення є 3НФ-відношенням, але не кожне 3НФ-відношення є НФБК-відношенням.
-

Приклад 4. "Здача іспитів"

- Припустимо, що студент може здавати іспит кілька разів. Щоб уникнути однофамільців, можна однозначно ідентифікувати студента номером його залікової книжки.
 - З іншої сторони, ведеться електронний облік успішності студентів, тому кожному студенту присвоюється його унікальний номер-ідентифікатор.
 - Відношення, яке моделює здачу іспитів має структуру:
(НомЗалКн, ІдСтуд, Дисципл, Дата, Оц)
-

Приклад 4. (продовж.)

- Можливими ключами відношення є
НомЗалКн, Дисципл, Дата та
ІдСтуд, Дисципл, Дата
 - Розглянемо існуючі функціональні залежності:
 - НомЗалКн, Дисципл, Дата \rightarrow Оц;
 - ІдСтуд, Дисципл, Дата \rightarrow Оц;
 - НомЗалКн \rightarrow ІдСтуд;
 - ІдСтуд \rightarrow НомЗалКн
-

Приклад 4. Приведення до НФБК

- Вихідне відношення перебуває у 3НФ, однак два детермінанта НомЗалКн та ІдСтуд не є можливими ключами відношення.
 - Для приведення відношення до НФБК необхідно розділити відношення на два з такими схемами:
 - (ІдСтуд, Дисципл, Дата, Оц) та
 - (НомЗалКн, ІдСтуд)або
 - (НомЗалКн, Дисципл, Дата, Оц) та
 - (НомЗалКн, ІдСтуд),що є рівнозначними.
-

Від ННФ до НФБК:

- ☐ Ключ (1НФ)
 - ☐ цілий ключ (2НФ)
 - ☐ і нічого, крім ключа (3НФ)
 - ☐ і нехай допоможе мені Кодд (НФБК)
-

Багатозначна залежність (БЗЗ)

- ❑ У більшості випадків досягнення 3НФ або НФБК є достатнім для реальних проектів БД.
 - ❑ Однак, окрім функціональних залежностей в ході досліджень були виявлені й інші залежності.
 - ❑ Багатозначна залежність (multi-valued dependence, MVD) при проектуванні відношень може викликати проблеми, пов'язані з надлишковістю даних.
 - ❑ Між атрибутами A , B і C деякого відношення існує **багатозначна залежність** (БЗЗ) (позначається $A \twoheadrightarrow B|C$), якщо для кожного значення атрибуту A існує набір значень атрибуту B і набір значень атрибуту C . Однак значення атрибутів B та C не залежать один від одного.
-

Четверта нормальна форма (4НФ)

- Базується на понятті багатозначної залежності
 - Відношення перебуває у **четвертій нормальній формі** тоді і тільки тоді, коли воно перебуває у НФБК і не містить нетривіальних багатозначних залежностей, тобто таких багатозначних залежностей, які не мають функціональних залежностей.
 - Відношення з нетривіальними БЗЗ утворюються, як правило в результаті природного з'єднання двох відношень за спільним полем, яке не є ключовим в жодному відношенні.
 - Фактично це приводить до спроби зберігати в одному відношенні інформацію про дві незалежні сутності.
-

Приклад 5. "Планування здачі іспитів"

- Схема відношення має вигляд:

Іспити (НомЗалКн, Гр, Дисципл)

- Оскільки воно повністю ключове, то в ньому не мітиться ФЗ.
 - Але певний зв'язок між атрибутами є.
 - Перелік дисциплін, які повинен здати студент, визначається не його номером залікової книжки, а групою, тобто спеціальністю.
 - Також група визначає список студентів, які в ній вчаться
 - Таким чином, існують дві БЗЗ:
Гр->>Дисципл|НомЗалКн
-

Приклад 5. Приведення до 4НФ

- В такому відношенні виникають аномалії модифікації.
 - Наприклад, внести нову дисципліну для усіх студентів групи досить трудомістка операція.
 - З іншої сторони така ж ситуація виникає при внесенні нового студента.
 - Необхідно зробити декомпозицію вихідного відношення на два:

Хто-Де (НомЗалКн, Гр),
Де-Що_здавати (Гр, Дисципл).
 - У цих відношеннях усунуто аномалії вставки і видалення. При цьому отримані відношення перебувають у 4НФ, оскільки є повністю ключовими і в них немає ФЗ.
-

Приклад 5. (зауваження)

- За вимогою **зберігання оцінки**, отриманої при здачі іспитів, відношення Іспити прийдеться залишити у попередньому вигляді, додавши в нього атрибут Оцінка, оскільки іншого місця для зберігання оцінки немає:

Іспити (НомЗалКн, Гр, Дисципл, Оцінка)

- Цей новий атрибут буде функціонально залежати від складеного ключа НомЗалКн, Гр, Дисципл.
-

Приклад 6. Приведення до 4НФ

- ❑ Це ситуація, коли людина має декілька місць роботи і декілька дітей.
 - ❑ Зберігання інформації про місця роботи і дітей в одному відношенні призводить до нетривіальної БЗЗ (Особа->>Робота|Діти).
-

Залежність з'єднання і 5НФ

- ❑ При будь-якій декомпозиції відношення на два інших отримані відношення володіють властивістю з'єднання без втрат.
 - ❑ Однак бувають випадки, коли потрібно зробити декомпозицію на більш ніж два відношення. У таких (досить рідких випадках) виникає необхідність враховувати залежність з'єднання (**project-join**), яка усувається за допомогою 5НФ.
 - ❑ **Залежність з'єднання** – це властивість декомпозиції, яка генерує хибні рядки при оберненому з'єднанні декомпозованих відношень.
 - ❑ Наявність PJ-залежності у відношенні робить його надлишковим та утруднює операції модифікації.
-

Приклад 7. Залежність з'єднання.

Відношення R зображено у таблиці:



X	Y	Z
1	1	2
1	2	1
2	1	1
1	1	1

Приклад 7. (продовж.)

Різні проекції відношення R, які включають по два атрибути:

Проекція $R1=R[X, Y]$			Проекція $R2=R[X, Z]$			Проекція $R3=R[Y, Z]$	
X	Y		X	Z		Y	Z
1	1		1	2		1	2
1	2		1	1		2	1
2	1		2	1		1	1

Приклад 7. (продовж.)

З'єднання відношень R1 і R2

R1 JOIN R2 WHERE R1.X=R2.X		
X	Y	Z
1	1	2
1	1	1
1	2	2
1	2	1
2	1	1

Приклад 7. (продовж.)

- Відношення R не відновлюється ні за одним з попарних з'єднань
 - $R1 \text{ JOIN } R2$
 - $R1 \text{ JOIN } R3$
 - $R2 \text{ JOIN } R3$.
 - Дійсно, з'єднання $R1 \text{ JOIN } R2$ містить лишній кортеж (позначено курсивом), який відсутній у відношенні R .
 - Аналогічно (через симетрію) інші попарні з'єднання не відновлюють відношення R .
 - Однак відношення R відновлюється з'єднанням усіх трьох проекцій $R1 \text{ JOIN } R2 \text{ JOIN } R3 = R$.
-

Приклад 7. (резюме)

- Між атрибутами цього відношення також є якась залежність, але вона не є функціональною і не є багатозначною.
 - Це є **залежність з'єднання**, що є узагальненим поняттям БЗЗ.
-

П'ята нормальна форма (5НФ)

- Відношення перебуває у **п'ятій нормальній формі** тоді і тільки тоді, коли будь-яка залежність з'єднання впливає з існування деякого можливого ключа, тобто є тривіальною.
 - Іншими словами, відношення не перебуває у 5НФ, якщо воно містить нетривіальну залежність з'єднання.
-

П'ята нормальна форма (5НФ)

- ❑ **Висновок:** не знаючи нічого про потенційні ключі відношення і як пов'язані атрибути, не можна робити висновок про те, чи знаходиться відношення у 5НФ.
 - ❑ Повертаючись до прикладу 7, можна лише припустити, що відношення R не перебуває у 5НФ. Якщо ж припустити, що відношення R є повністю ключовим, то воно перебуває в 4НФ, але не в 5НФ.
 - ❑ 5НФ рідко використовується на практиці. Вона є більш теоретичним дослідженням. Досить важко визначити саму залежність з'єднання для усіх можливих станів БД.
-

Алгоритм декомпозиції відношення з метою його нормалізації

- ❑ Декомпозиція – це процес розбиття відношення, яке не перебуває в потрібній НФ, на декілька відношень, кожне з яких перебуває у потрібній НФ, шляхом застосування до вихідного відношення оператора реляційної алгебри **Проекція**.
 - ❑ Якщо результат виконання операції **Проекція** містить кортежі, які повторюються, то залишається лише один кортеж з цієї групи.
-

SQL-код оператора реляційної алгебри **Проекція**

```
SELECT DISTINCT col1, col2 INTO T_2  
FROM T_1;  
ALTER TABLE T_1 DROP COLUMN col2;
```

або

```
SELECT col1, col2 INTO T_2  
FROM T_1  
GROUP BY col1, col2;  
ALTER TABLE T_1 DROP COLUMN col2;
```

- ❑ Оператор **Проекція** виконується над таблицею T_1 за її стовпцями col1, col2.
 - ❑ Є обов'язковим вказання або ключового слова DISTINCT, або фрази GROUP BY для автоматичного виключення дублювання кортежів в результатному відношенні.
 - ❑ Після виконання проекції необхідно визначити і задати ключ відношення T_2.
-

Алгоритм декомпозиції відношення

У реляційній БД для OLTP-систем як правило виконується декомпозиція до 3НФ, алгоритм якої містить наступні кроки

- ☐ Приведення до 1НФ
 - ☐ Виявлення ФЗ
 - ☐ Приведення до 2НФ
 - ☐ Приведення до 3НФ
-

Приведення до 1НФ

- ❑ Відношення, які складають модель БД, аналізуються на відповідність 1НФ.
 - ❑ З відношень вилучаються групи даних, уточнюється перелік доменів БД.
 - ❑ Перевіряється скалярність атрибутів з точки зору запитів до значень цих атрибутів.
 - ❑ Нескалярні атрибути представляються у вигляді декількох скалярних атрибутів.
-

Виявлення ФЗ

- ❑ Одна з проблем, пов'язана з виявленням ФЗ, полягає в тому, що переважно при використанні формального визначення можна виявити більше ФЗ, ніж потрібно для коректної нормалізації.
 - ❑ Тому залишають мінімально можливий набір ФЗ, який називається мінімальним покриттям, що забезпечує коректне проектування БД.
-

Приведення до 2НФ

- Якщо в деяких відношеннях виявлена ФЗ атрибутів від частини складеного ключа, то проводиться декомпозиція цих відношень на декілька відношень наступним чином:
 - ті атрибути, які залежать від частини складеного ключа, виносяться в окреме відношення разом з частиною ключ.
 - У вихідному відношенні лишаються усі ключові атрибути.
 - Наприклад, нехай вихідне відношення $R(K1, K2, A1, \dots, A_n, B1, \dots, B_m)$ має складений ключ $(K1, K2)$, а також містить дві ФЗ:
 $(K1, K2) \rightarrow (A1, \dots, A_n, B1, \dots, B_m)$ і $(K1) \rightarrow (A1, \dots, A_n)$.
Тоді декомпозовані відношення мають вигляд:
 $R1(K1, K2, B1, \dots, B_m)$ з ключем $(K1, K2)$ і
 $R2(K1, A1, \dots, A_n)$ з ключем $K1$.
-

Приведення до 3НФ

- Якщо в яких-небудь відношеннях виявлена залежність деяких неключових атрибутів від інших неключових атрибутів, то проводиться декомпозиція цих відношень наступним чином.
 - Ті неключові атрибути, які залежать від інших неключових атрибутів, виносяться в окреме відношення.
 - У новому відношенні ключем стає детермінант функціональної залежності.
 - Нехай вихідне відношення $R(K, A_1, \dots, A_n, B_1, \dots, B_m)$ має ключ K і дві ФЗ:
 - $K \rightarrow (A_1, \dots, A_n, B_1, \dots, B_m)$ – залежність усіх атрибутів від ключа відношення і
 - $(A_1, \dots, A_n) \rightarrow (B_1, \dots, B_m)$ – залежність деяких неключових атрибутів від інших неключових атрибутів.Декомпозовані відношення мають вигляд:
 - $R_1(K, A_1, \dots, A_n)$ – залишок від вихідного відношення з ключем K і
 - $R_2(A_1, \dots, A_n, B_1, \dots, B_m)$ з ключем (A_1, \dots, A_n) , куди винесені атрибути з вихідного відношення разом з детермінантом ФЗ.
-

Резюме

- На практиці при створенні логічної моделі даних, як правило, не використовують безпосередньо наведений алгоритм.
 - Досвідчені розробники БД відразу будують відношення у 3НФ.
 - Нагадаємо, що ER-діаграми відразу дозволяють створювати відношення у 3НФ, якщо використовувати правила моделі «сутність-зв'язок».
 - Тим не менше, наведений алгоритм важливий у двох випадках
 - По-перше, цей алгоритм показує, які проблеми виникають при розробці слабонормалізованих відношень.
 - По-друге, модель предметної області не завжди правильно розроблена з першого разу. Експерт предметної області може забути щось деталізувати, розробник може неправильно зрозуміти експерта, у процесі розробки можуть змінитись правила, прийняті у предметній області, і т.п. Усе це може призвести до появи нових функціональних залежностей, які були відсутні у першому варіанті моделі.
 - У таких ситуаціях і застосовується алгоритм нормалізації хоча б для того, що відношення залишились у 3НФ і логічна модель не погіршилась.
-

Дякую за увагу
