Лабораторна робота 5: Нормалізація бази даних

У цій лабораторній роботі ви проаналізуєте та вдосконалите схему бази даних, яку створили в попередніх лабораторних роботах. Нормалізація має бути частиною процесу проектування бази даних. Вивчаючи існуючу схему (ER-діаграму та таблиці) і зразки даних, ви визначите надлишковість або аномалії та реорганізуєте таблиці, щоб зменшити дублювання. Мета полягає в покращенні цілісності даних шляхом структурування таблиць таким чином, щоб кожен факт зберігався лише один раз.

Цілі

- **Пошук надлишковості та аномалій:** Виявлення потенційної надлишковості даних (наприклад, повторювані значення) або аномалій оновлення (проблеми вставки/оновлення/видалення) у поточній схемі.
- **Перелік функціональних залежностей:** Визначте та перелічіть функціональні залежності (ФЗ) для кожної проблемної таблиці.
- **Перевірка нормальних форм:** Оцініть поточну нормальну форму кожної таблиці (1NF, 2NF, 3NF) на основі її функциональних диференціалів (FD) та структури ключа.
- **Застосування нормалізації:** перетворення таблиць у вищі нормальні форми (до 3НФ) для усунення часткових та транзитивних залежностей.

Інструкції

Відкрийте схему вашої лабораторної роботи 2 у PostgreSQL та перегляньте кожну таблицю, її стовпці та зразки даних. Пам’ятайте, що ваша ER-діаграма надає *макрозображення* сутностей та зв’язків, тоді як нормалізація досліджує деталі кожної таблиці.

Знайдіть одну або кілька таблиць з погано структурованими даними. Шукайте повторювані групи або дублікати даних у рядках, а також будь-які ознаки аномалій (наприклад, одне й те саме значення зберігається в кількох місцях). Це ознаки того, що таблиця може порушувати 1NF або вищі форми.

Для кожної вибраної таблиці запишіть функціональні залежності (ФЗ). Нагадаємо, що ФЗ – це твердження виду «X → Y», що означає, що X однозначно визначає Y. Визначте всі ключі та те, як від них залежать неключові стовпці.

Для кожної таблиці перевірте критерії 1NF, 2NF та 3NF:

1. *1NF:* Усі атрибути є атомарними (без повторюваних груп). Якщо таблиця має стовпець, який може містити кілька значень (наприклад, список або повторюваний атрибут), це порушує 1NF.
2. *2NF:* Таблиця знаходиться в 1NF і (якщо первинний ключ складений) жоден неключовий атрибут не залежить лише від частини ключа (часткової залежності немає). Якщо складений ключ існує, переконайтеся, що кожен інший стовпець залежить від усього ключа, а не від його підмножини.

3. *3NF:* Таблиця знаходиться в 2NF, і жоден неключовий атрибут не залежить від іншого неключового атрибута (немає транзитивної залежності). Іншими словами, кожне неключове поле повинно залежати лише від первинного ключа.

У вашому `normalization.md` опишіть крок за кроком, як виправити кожну таблицю. Поясніть, як видалити повторювані групи (для досягнення 1NF) шляхом розділення стовпців на нові таблиці, потім як усунути часткові залежності (для досягнення 2NF) шляхом подальшої декомпозиції таблиць, і, нарешті, як видалити транзитивні залежності (для досягнення 3NF) Для кожного кроку покажіть оригінальний дизайн таблиці та запропоновані нові таблиці (з ключами) і поясніть, чому зміна усуває аномалію.

Опишіть перероблені SQL-інструкції `CREATE TABLE` для вашого нового дизайну. Відобразіть усі нові таблиці, стовпці, первинні ключі та зовнішні ключі, щоб схема була в 3NF. За потреби ви можете додати сурогатні ключі або таблиці пошуку. Перевірте ці визначення в pgAdmin (або psql), щоб переконатися, що вони завантажуються правильно.

Використовуючи бажаний інструмент для побудови діаграм, намалюйте нову схему (після нормалізації). Експортуйте ER-схему у зображення або PDF-файл і додайте її до своїх результатів. Оновлена діаграма має відображати всі нові таблиці та їхні зв'язки.

Поради

- Часткові та транзитивні залежності: Часткова залежність — це коли неключовий атрибут залежить лише від частини складеного ключа. Транзитивна залежність — це коли неключовий атрибут залежить від іншого неключового атрибута. Зверніть увагу на це під час перевірки 2NF та 3NF.
- Збалансуйте нормалізацію та продуктивність: нормалізація зменшує надмірність та аномалії, але надмірна декомпозиція може негативно вплинути на продуктивність. На практиці більшість схем нормалізуються до 3NF (або BCNF), щоб збалансувати цілісність даних з ефективністю. Прагніть до 3NF, якщо у вас немає конкретної причини зупинитися раніше.
- Після модифікації схеми перезавантажте всі зразки даних та виконайте запити в pgAdmin або psql. Переконайтеся, що обмеження цілісності даних працюють (наприклад, зовнішні ключі застосовуються) та що ваші запити все ще повертають правильні результати з новою структурою.

Результати

- Оновлена діаграма ER: зображення або PDF-файл оновленої діаграми ER, що відображає ваші нормалізовані таблиці та зв'язки.
- Звіт Markdown (`normalization.md`): Включіть у цей документ такі розділи:
- Оригінальний та перероблений дизайн таблиць (показати команду `ALTER TABLE` для кожної зміненої таблиці).
- Функціональні залежності, визначені для кожної таблиці.

— Покрокове пояснення нормалізації (як кожна таблиця переходить з 1НФ до 2НФ та до 3НФ).
— SQL-скрипти: Файл(и) `.sql`, що містять інструкції `CREATE TABLE` для переглянутої схеми (і будь-які необхідні команди `DROP TABLE`).

Критерії оцінювання

Зафіксуйте та надішліть усі результати до вашого групового репозиторію Git: оновлене зображення/PDF-файл ER-діаграми, звіт `normalization.md` та файли SQL-скриптів. Переконайтеся, що все добре організовано та чітко задокументовано.

— Потрібно навести всі функціональні залежності для початкової (ненормалізованої) схеми. Всі залежності мають бути коректними та повними (мінімальний набір).

— Треба вказати найвищу нормальну форму (1NF, 2NF, 3NF тощо) початкової схеми та пояснити її. Потрібно пояснити порушення (наприклад, часткові чи транзитивні залежності), що заважають нормалізації до 2NF чи 3NF.

— Треба показати всі етапи нормалізації: перехід до 1NF, потім 2NF і 3NF. Кожен проміжний етап має включати таблиці зі атрибутами та ключами. Кінцева таблиця повинна бути у 3NF.

— Треба надати SQL DDL-скрипти для фінальних таблиць у 3NF, з правильними визначеннями стовпців та типів даних, а також з ключами (первинними та зовнішніми).

Розділ 11. Функціональні залежності – Проектування баз даних – 2-ге видання

https://opentextbc.ca/dbdesign01/chapter/chapter-11-functional-dependencies/

Розділ 12. Нормалізація – Проектування баз даних – 2-ге видання

https://opentextbc.ca/dbdesign01/chapter/chapter-12-normalization/

Основи функціональних залежностей та нормалізації для реляційних баз даних

https://www.tutorialspoint.com/basics-of-functional-dependencies-and-normalization-for-relat ional-databases

Нормалізація проти денормалізації: компроміси, які вам потрібно знати

https://celerdata.com/glossary/normalization-vs-denormalization-the-trade-offs-you-need-to-know