

**БАЗИ ДАНИХ**  
**Лабораторна робота №9**

**I. Тема**

Нормалізація схеми бази даних

**II. Мета**

1. Оцінка відповідності розробленої схеми БД нормальним формам.
2. Способи підвищення ефективності операцій отримання та пошуку даних шляхом денормалізації схеми БД.
3. Застосування SQL-інструкцій для зміни структури таблиць, перевизначення типів даних колонок.
4. Композитні типи даних для визначення цілісних інформаційних одиниць в обраній ПО.
5. Пакетне оновлення структури БД із збереженням інформації.

**III. Завдання**

№	Склад роботи	Форма звіту
1	<p>Проаналізувати розроблену схему БД стосовно її відповідності першим 3 нормальним формам.</p> <p>Навести 3 обставини різного типу, що порушують ці вимоги (не враховувати можливість NULL-значень для деяких колонок) та запропонувати методи нормалізації таблиць</p>	<p>Визначити функціональні та транзитивні залежності в обраних таблицях, що не задовольняють вимогам нормальних форм.</p> <p>Навести оновлену структуру відповідних таблиць</p>
2	<p>Встановити деякі структурні недоліки існуючого опису таблиць (зайві ключі, надлишкові дані, некоректні типи, колонки або таблиці із порушенням уніфікації назв, неповнота встановлених обмежень цілісності, невідповідність типів особливостям значень та інше).</p> <p>Навести 3 обставини, виправлення яких приведе схему БД до більшої відповідності даним ПО</p>	<p>Описати, які принципи реляційного подання даних порушує колонка чи таблиця у поточній схемі БД. Зазначити, якщо необхідно, особливості даних ПО.</p> <p>Навести оновлену структур відповідних таблиць</p>
3	<p>Визначити можливі шляхи реорганізації структури таблиць для підвищення ефективності доступу до певних даних.</p> <p>Навести 1 спосіб денормалізації таблиць, що забезпечить суттєво ефективніший спосіб отримання або пошуку даних</p>	<p>Привести 1 запит на вибірку, якій працює менш ефективно для старої структури, та буде більш ефективним для нової.</p> <p>Обґрунтувати висновок за відповідними планами</p>

		виконання або обставинами, що характеризують дані ПО
4	<p>Розробити набір SQL-інструкцій для модифікації опису таблиць, що виправляє в поточній структурі БД раніше встановлені недоліки.</p> <p>Застосувати команди ALTER, DROP, CREATE для таблиць, колонок, обмежень або індексів</p>	<p>Подати набір команд, які при послідовному виконанні в транзакції перетворюють схему та дані, виправляючи визначені недоліки.</p> <p>Навести скрін із результатом виконання такого пакету команд.</p>
5	<p>Розширити створений набір SQL-інструкцій та визначити їхню послідовність виконання, щоб трансформувати структуру БД з урахуванням наявності даних в усіх таблицях. Важливо не допустити втрати інформації, що вже зберігається.</p> <p>Додатково при оновленні таблиць слід застосувати команди SELECT, UPDATE, INSERT, DELETE для перенесення даних між створюваними таблицями чи колонками, якщо у виправленій схемі БД змінено спосіб зберігання даних</p>	<p>Навести скріни структури та зразки даних 1 зміненої таблиці, які були до та стали після оновлення схеми</p>
6	<p>Визначити особливості представлення деяких даних обраної ПО, що подаються занадто складно з урахуванням вимог реляційної моделі.</p> <p>Обґрунтувати, в чому полягає складність чи незручність такої структури опису інформації з точки зору очікуваних способів її використання (внесення, вибірки, пошуку)</p>	<p>Привести приклад значення композитного типу або фрагмент структурованого документу, якій більш зручно (компактно, цілісно, ефективно, виразно) подає визначений блок даних</p>

#### IV. Додаткові вказівки

Склад таблиць якісно розробленої БД включає задані в єдиному стилі назви колонок із зрозумілим призначенням. Використані типи даних (розмір, точність) повинні відповідати мінімальним інформаційним одиницям ПО, мати визначені діапазони значень або встановлені домени.

Сурогатні первинні ключі застосовуються для спрощення зв'язку із іншими таблицями або у разі, якщо немає жодного атрибуту чи пари із вимогою унікальності.

Нормальні форми задають додаткові вимоги до структури таблиць, враховуючи природу інформації. Можна вважати, що таблиця представляє

дані однієї сутності. Додаткові таблиці можуть визначати багатозначні атрибути, складні та взаємовиключні відношення між сутностями, спільні для декількох сутностей характеристики, ієрархію.

Передумова для нормалізації – наявність додаткових взаємозв'язків між колонками в таблиці. Встановлений первинний ключ має визначати решту даних, які самі по собі між собою не пов'язані. Будь-яка колонка, що обчислюється від інших колонок цієї таблиці, або залежить від даних зв'язаних таблиць, порушує вимоги нормалізації.

При розробці схеми БД слід враховувати майбутню експлуатацію моделі в реальній інформаційній системі. Формальний підхід до нормалізації таблиць інколи вступає в протиріччя вимогам щодо продуктивності вибірки даних, зручності опрацювання та самодостатності описаних структур.

Наприклад, комплексні індекси створюються лише на колонки однієї таблиці, хоча пошук може відбуватись за декількома параметрами одночасно, що описані в різних таблицях, зв'язаних реляційними ключами.

Процесу денормалізації характерне об'єднання в 1 таблицю різних сутностей, дублювання інформації із додаванням тригерів для автоматичної підтримки узгодженості даних. Наприклад, замість використання рекурсивної вибірки при реалізації багаторівневого зв'язку “батько-нащадок”, визначеного 1 колонкою, створюється додаткова таблиця, яка для “батьків” зберігає ключі “нащадків” на всіх рівнях.

Для зменшення кількості таблиць та очікуваних з'єднань в запитах, коли модель реалізує зв'язок 1-1, дані можна включити в 1 спільну таблицю. Якщо такий зв'язок був необов'язковим, то слід дозволити NULL на відповідну частину колонок. При цьому коректним є обмеження CHECK з умовою: або всі значення додаткових колонок визначені, або не визначено жодного.

Різнорідні дані зазвичай вимагають складної організації таблиць для коректного представлення в реляційній моделі. Якщо немає вимог до ефективного пошуку або такий блок даних завжди використовуються як цілісне значення, то виправдано застосування підтримуваних СУБД композитних типів (ARRAY, RECORD, JSON). При значній кількості різних атрибутів сутності можна виконати уніфікацію домену даних, що дозволяє зберігати значення в окремій таблиці “ключ-атрибут-значення”.

Перед модифікацією поточної БД слід зробити дамп структури та всіх даних у файл. Після розробки окремих команд, що виправляють встановлені недоліки, їх необхідно перевірити в пакетному режимі. Дамп із файлу можна завантажити в створену пусту БД, виконати в ній всі команди 1 транзакцією і порівняти оновлену базу (таблиці, дані, обмеження) із базою, на якій перед цим виконувалась розробка запитів.

Заключна частина звіту містить висновки по темі роботи, власні думки або важливі спостереження, отримані при виконанні завдань.

## **V. Контрольні запитання**

Основні теми відповідно до лекцій №№ 11-12:

- об'єкти аналізу при розробці БД

- особливості проектування реляційної БД
- вибір первинного ключа
- цілі нормалізації схеми БД
- передумови для денормалізації БД
- декомпозиція, відтворення відношення
- вимоги 1 нормальної форми
- вимоги 2 та 3 нормальних форм
- об'єктно-реляційне відображення
- композитні типи даних