

Лабораторна робота № 3

Тема: Проектування простих *DQL/DML*-команд мови *SQL* проекції, вибірки та модифікації

Завдання

В лабораторній роботі використовується БД, створена у лабораторній роботі №2 відповідно до вашого варіанта.

За кожним етапом (окрім етапу 3) створити файл-скрипт *N.sql*, де *N* – номер етапу, в який під час виконання завдань вказувати:

- 1) умова завдання у вигляді багаторядкового коментаря
- 2) *SQL*-команда
- 3) рядки із відповіддю на запит (для *SELECT*-команд) або реакція СКБД (для помилки) у вигляді багаторядкового коментаря

Для отримання рядків із відповіддю на *SQL*-команда зручно використовувати *SQLPlus*.

Приклад оформлення кожного рішення:

```
/* 1) отримання зарплат працівників за рік з урахуванням надбавки
*/

SELECT
    ename,
    sal,
    12*sal+100
FROM emp;

/*
ENAME          SAL  12*SAL+100
-----
KING            5000      60100
TOD             4000      48100
SCOTT           3000      36100
BLAKE           3500      42100
ALLEN           3000      36100
*/
```

Етап 1 Конструювання простих *INSERT*-команд

(під час *Online*-заняття бажано встигнути виконати 1-5 завдання з урахуванням 2-х таблиць)

- 1.1 Для кожної таблиці БД створити команди внесення даних, тобто внести по два рядки.
- 1.2 Для однієї з таблиць створити команду додавання колонки типу *date* з урахуванням предметної області.

1.3 Для зазначеної таблиці створити команду на внесення одного рядка, але з невизначеним значенням колонки типу date.

1.4 Створити команду налаштування формату date = dd/mm/yyyy.

1.5 Для задіяної в завданні 1.2 таблиці створити ще одну команду на внесення одного рядка з урахуванням значення колонки типу date.

1.6 Для однієї з таблиць, що містить обмеження цілісності потенційного ключа, виконати команду додавання нового рядка зі значенням колонки, що порушує це обмеження. Перевірити реакцію СКБД на таку зміну.

1.7 Для однієї з таблиць, що містить обмеження цілісності зовнішнього ключа, виконати команду додавання нового рядка зі значенням колонки зовнішнього ключа, який відсутній у колонці первинного ключа відповідної таблиці. Перевірити реакцію СКБД на подібне додавання, яке порушує обмеження цілісності зовнішнього ключа.

Етап 2 Конструювання простих SELECT-команд

(під час Online-заняття бажано встигнути виконати 1-4 завдання)

2.1 Для однієї з таблиць створити команду отримання значень всіх колонок (явно перерахувати) у всіх рядках.

2.2 Для однієї з таблиць створити команду отримання цілого числа колонки з використанням будь-якої арифметичної операції. При виведенні на екран визначити для зазначеної колонки нову назву псевдоніма.

2.3 Для однієї з таблиць, що містить колонку зовнішнього ключа створити команду отримання значення колонки без дублювання значень.

2.4 Для однієї з таблиць створити команду отримання результату конкатенації значень будь-яких двох колонок. При виведенні на початок рядка виведення додати літерал «UNION=».

2.5 Модернізувати рішення завдання 2.2, отримавши в порядку зростання значення псевдоніму.

2.6 Для однієї з таблиць створити команду отримання значення двох колонок, значення яких відсортовані в порядку зростання (для першої колонки) та в порядку зменшення (друга колонка).

Етап 3 Основи роботи з транзакціями

(під час Online-заняття бажано встигнути виконати всі завдання)

3.1 Відкрийте другий (додатковий) термінал з'єднання з сервером у будь-якому інструменті (*SQLPlus* або *SQLDeveloper*) та встановіть з'єднання з БД.

3.2 Створити команду внесення нового рядка до таблиці за прикладом рішення завдання 1.5 та виконайте послідовно цю команду у першому та у другому терміналах. Перевірте реакцію другого терміналу, в якому була виконана команда пізніше за часом.

3.3 У першому терміналі скасувати виконання команди та перевірте реакцію другого терміналу на цю команду.

3.4 У першому та у другому терміналах отримати вміст таблиці, в яку вносилися рядки.

За цим етапом рішення до загального звіту рішень не надаються.

Етап 4 Конструювання *SELECT*-команд з вибіркою (фільтруванням) даних

(під час Online-заняття бажано встигнути виконати 1-4 завдання)

4.1 Для однієї з таблиць створити команду отримання значень всіх колонок (явно перерахувати) за окремими рядками з урахуванням умови: цілочисельне значення однієї з колонок має бути більшим за якесь константне значення.

4.2 Для однієї з таблиць створити команду отримання значень всіх колонок (явно перерахувати) за окремими рядками з урахуванням умови: символічне значення однієї з колонок має співпадати з якимось константним значенням.

4.3 Для однієї з таблиць створити команду отримання значень всіх колонок (явно перерахувати) за окремими рядками з урахуванням умови: символічне значення однієї з колонок повинно містити в першому та третьому знаку якісь надані вами символи.

4.4 У завданні 1.2 було додано колонку типу *date*. Створити команду отримання значень всіх колонок (явно перерахувати) за окремими рядками з урахуванням умови: значення доданої колонки містить невизначене значення.

4.5 Створити команду отримання значень всіх колонок (явно перерахувати) за окремими рядками з урахуванням умови, що поєднує умови з рішень завдань 4.1 та 4.2

4.6 Створити команду отримання значень всіх колонок (явно перерахувати) за окремими рядками з урахуванням умови, що інвертує результат рішення 4.5

Етап 5 Конструювання простих *DELETE/UPDATE*-команд

(під час Online-заняття бажано встигнути виконати 1-4 завдання)

5.1 Для однієї з таблиць, що містить обмеження цілісності зовнішнього ключа, виконати команду зміни значення колонки зовнішнього ключа на значення, яке відсутнє в колонці первинного ключа відповідної таблиці. Перевірити реакцію СКБД на таку зміну.

5.2 Повторити рішення завдання 5.1, але таке, що не призводить до порушення цілісності зовнішнього ключа.

5.3 Для однієї з таблиць, що містить обмеження цілісності первинного ключа, пов'язане з колонкою зовнішнього ключа іншої таблиці, виконати одну команду видалення рядка зі значенням колонки первинного ключа, що є в колонці зовнішнього ключа іншої таблиці. Перевірити реакцію СКБД видалення.

5.4 Повторити рішення завдання 5.3, але таке, що не призводить до порушення цілісності зовнішнього ключа.

5.5 Для однієї з таблиць, що містить обмеження цілісності первинного ключа, виконати команду зміни значення колонки первинного ключа на значення, яке присутнє у колонці зовнішнього ключа відповідної таблиці. Перевірити реакцію СКБД на таку зміну.

5.6 Повторити рішення завдання 5.5, але таке, що не призводить до порушення цілісності зовнішнього ключа, тобто виконати команду зміни значення колонки первинного ключа на значення, яке вже не присутнє у колонці зовнішнього ключа відповідної таблиці.

Етап 6 Конструювання *SELECT*-команд з агрегацією даних

(під час Online-заняття бажано встигнути виконати 1-2 завдання)

6.1 Для однієї з таблиць створити команду отримання кількості рядків таблиці.

6.2 Для однієї з таблиць створити команду отримання суми значень однієї з цілих колонок.

6.3 Для однієї з таблиць створити команду отримання статистики появи значень однієї з колонок у таблиці, наприклад:

- значення 1, кількість появи значення 1
- значення 2, кількість появи значення 2

6.4 Модифікувати рішення попереднього завдання так, щоб у відповіді були відфільтровані рядки з урахуванням заданої умови, що включає використану функцію агрегації.

Етап 7 Документування результатів роботи на Веб-сервісі *GitHub*

7.1 Розпочинаючи роботу над документуванням рішень лабораторної роботи, необхідно у вашому *GitHub*-репозиторії:

- 1) створити *Issue* з назвою «*tasks-of-laboratory-work-3*»;
- 2) підключити до *Issue* ваш *GitHub-project* (правий розділ «*Projects*» сторінки з *Issue*);
- 3) змінити статус *Issue* з «*Todo*» на «*In progress*», автоматично перевівши *Scrum*-картку з цим *Issue* на *Scrum*-дошку «*In progress*»;
- 4) створити нову *Git*-гілку з назвою, яка відповідає назві *Issue*, наприклад, «*tasks-of-laboratory-work-3*» (використовується посилання «*Create a branch*» у правому розділі «*Development*» сторінки з *Issue*).

7.2 Після створення *Git*-гілки перейти до цієї гілки для створення оновлень файлів *Git*-репозиторію.

7.3 У новій гілці *Git*-репозиторію створити каталог з назвою «*3-Simple-DQL-DML*» (кнопка «*Add file*» - «*Create new file*»), при створенні якого одночасно створити файл *README.md* з першим рядком «*3 Проектування простих DQL/DML-команд мови SQL проєкції, вибірки та модифікації*» зі стилем «Заголовок 3-го рівня» мови розмітки *Markdown* (три символи решітка ###).

7.4 Розмістити в каталозі «*3-Simple-DQL-DML*» *GitHub*-репозиторія файли *1.sql*, *2.sql*, *4.sql*, *5.sql*, *6.sql* з рішеннями завдань відповідних етапів.

7.5 Виконати запит *Pull Request*, розпочавши процес *Code Review*.

Під час створення *Pull Request* необхідно вказати:

- *Reviewers* = *Oleksandr Blazhko*, *Maria Glava*;
- *Labels* = *enhancement (New feature or request)*;
- *Projects* = посилання на *GitHub-project*.

Завершення процесу *Code Review* відбудеться до початку нового заняття, після чого викладач закриє *Issue*, завершуючи процес виконання завдань з лабораторної роботи.

Під час консультації у найближчий понеділок ви зможете отримати більше коментарів щодо ваших рішень.