

Лабораторна робота 6

Тема: Віртуальні таблиці (представлення) та інтеграція даних

Завдання

За кожним етапом 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 СУБД (для помилки) у вигляді багаторядкового коментаря

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

Етап 1. Створення віртуальних таблиць (VIEW) у реляційній БД

Під час *Online*-заняття бажано встигнути виконати два завдання етапу.

Використовується реляційна БД, створена та заповнена у лабораторних роботах №2-3.

При створенні віртуальних таблиць використовувати довільні назви таблиць та колонок.

1.1 Створити віртуальну таблицю, структура та вміст якої відповідає рішенням завдання 4.2 з лабораторної роботи №3: для однієї з таблиць створити команду отримання значень усіх колонок (явно перерахувати) за окремими рядками з урахуванням умови, в якій рядкове значення однієї з колонок має співпадати з якимось константним значенням. Отримати вміст таблиці.

1.2 Виконати команду зміни значення колонки створеної віртуальної таблиці на значення, яка входить в умову вибірки рядків із рішення попереднього завдання, при цьому нове значення має відрізнятися від поточного.

1.3 Створити віртуальну таблицю, структура та вміст якої відповідає рішенням завдання 3.2 з лабораторної роботи №4: для двох таблиць, пов'язаних через РК-колонку та FK-колонку, створити команду отримання двох колонок першої та другої таблиць з використанням екві-сполучення таблиць. Отримати вміст таблиці.

1.4 Виконати команду додавання нового рядка до однієї з таблиць, що входить до запиту з попереднього завдання.

Етап 2. Створення віртуальних таблиць у БД EAV

Під час Online-заняття бажано встигнути виконати два завдання етапу.

Використовується БД зберігання об'єктно-реляційної моделі EAV, створена та заповнена у лабораторній роботі № 5

2.1 Створити віртуальну таблицю, структура та вміст якої відповідає рішенням завдання 2.3 з лабораторної роботи №5, але враховує опцію «*WITH READ ONLY*»: отримати інформацію про атрибутивні типи. Отримати вміст таблиці.

2.2 Виконати видалення одного рядка з віртуальної таблиці, створеної у попередньому завданні. Прокоментувати реакцію СУБД.

2.3 Створити віртуальну таблицю, що містить дві колонки:

назва класу, кількість екземплярів об'єктів класу. Отримати вміст таблиці.

2.4 Перевірити можливість виконання операції зміни даних у віртуальній таблиці, створеної у попередньому завданні. Прокоментувати реакцію СУБД.

Етап 3. Інтеграція баз даних

3.1 Створити нового користувача, ім'я якого = «ваше_прізвище_латиницею»+'EAV', наприклад, blazhko_eav, з правами, достатніми для створення та заповнення таблиць БД EAV.

3.2 Створити таблиці БД EAV та заповнити таблиці об'єктних типів та атрибутивних типів, взявши рішення з лабораторної роботи №5.

3.3 Створити генератор послідовності таблиці OBJECTS БД EAV, ініціалізувавши його початковим значенням з урахуванням вже заповнених значень.

3.4 Налаштувати права доступу нового користувача до таблиць схеми даних із таблицями реляційної БД вашої предметної області, створеної в лабораторній роботі №2.

3.5 Створити множину запитів типу *INSERT INTO ... SELECT*, які автоматично заповнюють таблицю *OBJECTS*, взявши потрібні дані з реляційної бази даних вашої предметної області.

Етап 4. Документування результатів роботи на Веб-сервісі *GitHub*

4.1 Розпочинаючи роботу над документуванням рішень лабораторної роботи, необхідно у вашому *GitHub*-репозиторії створити *Issue* з назвою «*tasks-of-laboratory-work-6*».

- 1) створити *Issue* з назвою «*tasks-of-laboratory-work-6*»;
- 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-6*» (використовується посилання «*Create a branch*» у правому розділі «*Development*» сторінки з *Issue*).

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

4.3 У новій гілці *Git*-репозиторію створити каталог з назвою «*6-Views*» (кнопка «*Add file*» - «*Create new file*»), при створенні якого одночасно створити файл *README.md* з першим рядком «*6 Віртуальні таблиці (представлення) та інтеграція даних*» зі стилем «Заголовок 3-го рівня» мови розмітки *Markdown* (три символи решітка *###*).

4.4 Розмістити в каталозі «*6-Views*» *GitHub*-репозиторія файли *1.sql*, *2.sql*, *3.sql* з рішеннями завдань відповідних етапів.

4.5 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 в понеділок ви зможете отримати більше коментарів щодо ваших рішень.