

Лабораторна робота 13

Тема: Проектування активних баз даних на основі тригерів з використанням PL/SQL СУБД Oracle

Завдання

В лабораторній роботі використовується БД, створена у лабораторній роботі №2.

Примітка: кожне завдання для свого успішного виконання потребує наявності в окремих колонках таблиць заданих значень, тому якщо в таблиці немає потрібних значень, необхідно їх попередньо створити, використовуючи команди INSERT або UPDATE.

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

- 1) умову завдання у вигляді багаторядкового коментаря
- 2) *SQL*-команду
- 3) рядки із відповіддю на запит (для *SELECT*-команд) або реакцію СУБД (для помилки) у вигляді багаторядкового коментаря

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

Етап 1. Прості інформуюче-доповнюючі та обмежуючі тригери

1. Створити інформуючий тригер для виведення повідомлення на екран після додавання, зміни або видалення рядка з будь-якої таблиці Вашої бази даних із зазначенням у повідомленні операції, на яку зреагував тригер. Навести тест-кейс перевірки роботи тригера.

2. Повторити попереднє завдання лише під час роботи користувача, ім'я якого збігається з вашим логіном. Навести тест-кейс перевірки роботи тригера.

Етап 2. Тригери складного контролю даних

1. Створити тригер для реалізації каскадного видалення рядків зі значеннями РК-колонки, пов'язаних з FK-колоною. Навести тест-кейс перевірки роботи тригера.

2. Створити тригер для реалізації предметно-орієнтованого контролю спроби додавання значення FK-колонки, що не існує у РК-колонці. Навести тест-кейс перевірки роботи тригера.

Етап 3. Автоматичне оновлення даних

1. Розробити механізм журналізації DML-операцій, що виконуються над таблицею вашої бази даних, враховуючи такі дії:

- створити таблицю з ім'ям LOG_ім'я_таблиці. Структура таблиці повинна включати: ім'я користувача, тип операції, дата виконання операції, атрибути, що містять старі та нові значення.

- створити тригер журналювання.

Перевірити роботу тригера журналювання для операцій INSERT, UPDATE, DELETE.

2. Припустимо, що використовується СУБД до 12-ї версії, яка не підтримує механізм DEFAULT SEQUENCE, який дозволяє автоматично внести нове значення первинного ключа, якщо воно не задано при операції внесення. Для будь-якої колонки вашої бази даних створити тригер з підтримкою механізму DEFAULT SEQUENCE. Навести тест-кейс перевірки роботи тригера.

Етап 4. DDL-тригера та DB-тригера

1. Створити таблицю для реєстрації наступних DDL-подій: тип події, що спричинила запуск тригера; ім'я користувача; ім'я об'єкта БД. Створити тригер реєстрації заданих подій створення об'єктів. Подати тест-кейси перевірки роботи тригера.

Етап 5. Документування результатів роботи на Веб-сервісі GitHub

5.1 Розпочинаючи роботу над документуванням рішень лабораторної роботи, необхідно у вашому *GitHub*-репозиторії створити *Issue* з назвою «*tasks-of-laboratory-work-13*».

- 1) створити *Issue* з назвою «*tasks-of-laboratory-work-13*»;
- 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-13*» (використовується посилання «*Create a branch*» у правому розділі «*Development*» сторінки з *Issue*).

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

5.3 У новій гілці *Git*-репозиторію створити каталог з назвою «*13-PL-SQL-Trigger*» (кнопка «*Add file*» - «*Create new file*»), при створенні якого одночасно створити файл

README.md з першим рядком «13 Проектування активних баз даних на основі тригерів з використанням *PL/SQL* СУБД Oracle» зі стилем «Заголовок 3-го рівня» мови розмітки Markdown (три символи решітка ###).

5.4 Розмістити в каталозі «13-*PL-SQL-Trigger*» *GitHub*-репозиторія файли 1.sql, 2.sql, 3.sql, 4.sql з рішеннями завдань відповідних етапів.

5.5 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*.