

Лабораторна робота 11

Тема: Проектування бізнес-логіки мовою *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. Основи мовних конструкцій мови *PL/SQL* СУБД Oracle

1.1. Виберіть таблицю бази даних, що містить стовпець типу *date*. Оголосіть змінні, що відповідають стовпцям цієї таблиці, використовуючи посилальні типи даних. Надайте змінним значення, використовуйте SQL-функції для формування значень послідовності, перетворення дати до вибраного стилю. Виведіть на екран рядок.

1.2. Додати інформацію до однієї з таблиць, обрану у попередньому завданні. Використовувати формування нового значення послідовності та перетворення формату дати.

1.3. Для однієї з таблиць створити команду отримання середнього значення однієї з цілих колонок, використовуючи умову вибірки за заданим значенням іншої колонки. Результат присвоїти змінній і вивести на екран.

1.4. Виконайте введення 5 рядків у таблицю бази даних, використовуючи цикл з параметрами. Значення первинного ключа генеруються автоматично, решта даних дублюється.

Етап 2. Обробка виняткових ситуацій у *PL/SQL* СУБД Oracle

2.1. Повторити виконання завдання 3 етапу 1, забезпечивши контроль відсутності даних у відповіді на запит із використанням винятку.

2.2. Повторити виконання завдання 3 етапу 1, забезпечивши контроль отримання багаторядкової відповіді на запит.

2.3. Повторити виконання завдання 3 етапу 1, забезпечивши контроль за внесенням унікальних значень.

Етап 3. Використання курсорів у *PL/SQL* СУБД *Oracle*

3.1. Виконайте DELETE-запит із попередніх рішень, додавши аналіз даних із неявного курсору. Наприклад, кількість віддалених рядків

3.2. Повторіть виконання завдання 3 етапу 1 з використанням явного курсору.

3.3. У попередній лабораторній роботі розглядався приклад:

```
/* Завдання 5:
Отримати 5-ть найбільш високооплачуваних
працівників (прізвище, зарплата)

Рішення задачі зі стандартним застосуванням вкладених запитів
для СУБД Oracle версії < 12c.
Використання системної колонки ROWNUM
*/

WITH
emp_sort_sal AS (
    SELECT ename, sal
    FROM emp
    ORDER BY sal DESC
),
emp_sort_sal_rownum AS (
    SELECT
        ROWNUM AS r,
        ename,
        sal
    FROM emp_sort_sal
)
SELECT
    ename,
    sal
FROM emp_sort_sal_rownum
WHERE
    r <= 5;

/* Завдання 5:
Варіант 2 класичного рішення задачі
для СУБД Oracle версії >= 12c
з використанням опції "Top-N"-запиту:
[ OFFSET offset { ROW | ROWS } ]
[ FETCH { FIRST | NEXT } [ { rowcount | percent PERCENT } ]
  { ROW | ROWS } { ONLY | WITH TIES } ]
*/

SELECT
    ename,
    sal
FROM Emp
ORDER BY sal DESC
FETCH FIRST 5 ROWS ONLY;
```

З урахуванням вашої предметної області створити анонімний PL/SQL-блок, який викликатиме один із варіантів подібних SQL-запитів залежно від значення версії СУБД.

При вирішенні використовувати:

- значення змінної DBMS_DB_VERSION.VERSION;
- явний курсор із параметричним циклом.

Етап 4. Динамічні SQL-команди мови PL/SQL СУБД Oracle

4.1. Виконайте введення 5 рядків у таблицю бази даних із динамічною генерацією команди. Значення первинного ключа генеруються автоматично, решта даних дублюється.

4.2. Скласти динамічний запит створення таблиці, іменами колонок якої будуть значення будь-якої символічної колонки. Попередньо виконати перевірку існування таблиці з її видаленням.

4.3. Команда ALTER SEQUENCE може змінювати початкове значення генератора починаючи з СУБД версії 12. Для ранніх версій доводиться виконувати дві команди: видалення генератора та створення генератора з новим початковим значенням.

З урахуванням вашої предметної області створити анонімний PL/SQL-блок, який викликатиме один із варіантів SQL-запитів зміни початкового значення генератора залежно від значення версії СУБД.

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-11*».

- 1) створити *Issue* з назвою «*tasks-of-laboratory-work-11*»;
- 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-11*» (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 «*11-DBS_Model*» (кнопка «*Add file*» - «*Create new file*»), при створенні якого одночасно створити файл *README.md* з першим рядком «11 Проектування бізнес-логіки мовою PL/SQL СУБД Oracle» зі стилем «Заголовок 3-го рівня» мови розмітки *Markdown* (три символи решітка *###*).

5.4 Розмістити в каталозі «*11-DBS_Model*» *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*.