

## Лабораторна робота 8

### Тема: Обробка даних на основі регулярних виразів

#### 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) 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. SELECT-запити з простою вибіркою на основі регулярних виразів

1. Одна з колонок таблиць повинна містити строкове значення з трьома різними буквами у першій позиції. Створіть запит, який отримає три рядки таблиці з урахуванням трьох букв, використовуючи оператор LIKE.
2. Повторіть завдання 1, використовуючи регулярні вирази з альтернативними варіантами.
3. Одна з колонок таблиць повинна містити строкове значення з цифрами від 3 до 8 у будь-якій позиції. Створіть запит, який отримає рядки таблиці з урахуванням присутності у вказаній колонці будь-якої цифри від 3 до 8.
4. Створіть запит, який отримає рядки таблиці з урахуванням відсутності в зазначеній колонці будь-якої цифри від 3 до 8.
5. Створіть запит, який отримає рядки таблиці з урахуванням присутності в раніше вказаній колонці поєднання будь-яких трьох цифр розміщених підряд від 3 до 8.

#### Етап 2. SELECT-запити зі складною вибіркою на основі регулярних виразів

1. Створіть запит, який отримає рядки таблиці з урахуванням присутності в раніше вказаній колонці поєднання будь-яких двох підряд розташованих цифр, або трьох підряд розташованих букв.
2. Одна з колонок таблиць повинна містити строкове значення з двома однаковими буквами, що повторюються підряд. Створіть запит, який отримає рядки таблиці з таким значенням колонки.
3. Одна з колонок таблиць повинна містити строкове значення з двома однаковими словами, що повторюються підряд. Створіть запит, який отримає рядки таблиці з таким значенням колонки.
4. Одна з колонок таблиць повинна містити строкове значення з номером мобільного телефону у форматі +XX(XXX)XXX-XX-XX, де X – цифра. Створіть запит, який отримає рядки таблиці з таким значенням колонки.
5. Одна з колонок таблиць має містити строкове значення з електронною поштовою адресою у форматі EEE@EEE.EEE.UA, де E – будь-яка латинська буква. Створіть запит, який отримає рядки таблиці з таким значенням колонки.

### Етап 3. Запити на вилучення підрядків та перетворення даних з використанням регулярних виразів

1. Одна з колонок таблиць повинна містити строкове значення з двома словами, розділеними пробілом. Виконайте команду UPDATE, помінявши місцями ці два слова.
2. Одна з колонок таблиць має містити строкове значення з електронною поштовою адресою у форматі EEE@EEE.EEE.UA, де E – будь-яка латинська буква. Створіть запит, вилучення логіна користувача з електронної адреси (підстрока перед символом @).
3. Одна з колонок таблиць повинна містити строкове значення з номером мобільного телефону у форматі +XX(XXX)XXX-XX-XX, де X – цифра. Виконайте команду UPDATE, додавши перед номером телефону фразу «Mobile:».
4. Додайте до колонки з електронною адресою обмеження цілісності, що забороняє вносити дані, відмінні від формату електронної адреси, використовуючи команду ALTER TABLE таблиця ADD CONSTRAINT обмеження CHECK (умова). Перевірте роботу обмеження на двох прикладах UPDATE-запитів із правильними та неправильними значеннями колонки.
5. Видаліть зайві дані з колонки з номером мобільного телефону, залишивши тільки номер телефону в заданому форматі.
6. Додайте в колонку з мобільним телефоном обмеження цілісності, що забороняє вносити дані, відмінні від формату, записаного в завданні 3. Перевірте роботу обмеження на двох прикладах UPDATE-запитів із правильними та неправильними значеннями колонки.

### 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-8».

- 1) створити *Issue* з назвою «tasks-of-laboratory-work-8»;
- 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-8» (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 «8-RegExp» (кнопка «Add file» - «Create new file»), при створенні якого одночасно створити файл README.md з першим рядком «8 Обробка даних на основі регулярних виразів» зі стилем «Заголовок 3-го рівня» мови розмітки Markdown (три символи решітка ###).

4.4 Розмістити в каталозі «8-RegExp» *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.