LESSON 5 LAB

SUBQUERIES IN THE SELECT CLAUSE

Ihor Liutak

Content

1. Enhancing Queries with Nested Data Retrieval
2. Understand the concept of subqueries
3. Learn how to use subqueries in the SELECT clause
4. Explore practical examples using db.loc database
5. Practice hands-on tasks to strengthen learning

Short description

Introduce the concept of subqueries and explain their use in sql. Each how to use subqueries in the select clause to fetch additional data or perform calculations.

Kurzbeschreibung

Stellen Sie das Konzept von Unterabfragen vor und erklären Sie deren Verwendung in SQL. Erfahren Sie, wie Sie Unterabfragen in der Select-Klausel verwenden, um zusätzliche Daten abzurufen oder Berechnungen durchzuführen.

Syntax Overview

General Syntax:

```
SELECT column1,
       (SELECT subquery_column
        FROM subquery_table
        WHERE condition) AS alias
FROM main_table;
```

Key Components:

- Outer Query: Main query fetching results.
- Inner Query: Subquery providing additional data.
- Alias: Name assigned to the subquery result.

Example 1 - Simple Subquery

Scenario: Fetch fruits with their country of origin.

Query:

```
SELECT name AS fruit_name,
       (SELECT name
        FROM countries
        WHERE countries.id = fruits.origin)
           AS country_name
FROM fruits;
```

Output:

- Fruit Name | Country Name
- ---------- | ----------
- Mango | India

Example 2 - Aggregated Data

Scenario: Find the average storage period for fruits by their seasonality.

Query:

```
    SELECT seasonality,
           (SELECT AVG(storage_period)
            FROM fruits AS f
            WHERE f.seasonality = fruits.seasonality) AS
avg_storage_period
    FROM fruits;
```

Output:

- Seasonality | Avg Storage Period
- ---------- | ----------
- Summer | 14 days

Hands-On Task 1

Task: List each fruit's name and the total number of images associated with it.

Query:

```
SELECT name AS fruit_name,
       (SELECT COUNT(*)
        FROM fruit_images
        WHERE fruit_images.fruit_id = fruits.id) AS image_count
FROM fruits;
```

Objective: Practice using scalar subqueries.

Hands-On Task 2

Task: Find fruits stored in the same storage place as the fruit "Mangosteen".

Query:

```
SELECT name AS similar_storage_fruits
FROM fruits
WHERE storage_place = (
      SELECT storage_place
      FROM fruits
      WHERE name = 'Mangosteen'
      LIMIT 1);
```

Objective: Use subqueries for matching conditions.

Recap and Q&A

Recap:

- Subqueries enhance SQL query flexibility.
- Can be used in `SELECT`, `WHERE`, or `FROM` clauses.

Best Practices:

- Keep Subqueries Simple