Advanced SQL Techniques Summary

Day 15 & 16 of 30 DaysDuckDBChallenge

ABOSEDE IGHARO A.

Overview of Day 15 & 16

Introduction to Subqueries and Window Functions

In days 15-16 of the 30 DaysDuckDBChallenge, we delved into advanced SQL techniques, focusing on Subqueries and Window Functions.

- **Subqueries:**
  - Discussed the meaning of Subqueries and the three types : Scalar, Multiple-row, and Correlated Subqueries.
  - Identified SQL clauses (e.g., WHERE, FROM) and SQL commands that allow Subqueries.
  - Highlighted the advantages of using the WITH clause in Subqueries.
- **Window Functions:**
  - Introduced the concept of Window Functions.
  - Explored specific Window Functions such as NTILE, Row number, Rank, Dense Rank, Lead, and Lag.

Understanding Subqueries

Meaning and Types of Subqueries

What is a Subquery?

A Subquery is often called an inner query or nested query and it is basically a query within a query. A subquery can be used to retrieve data that will be used in the main query or the outer query as a condition.

Types of Subqueries

1. **Scalar Subquery:** Returns just one row and one column.
2. **Multiple-row Subquery:** Returns multiple rows of data. There are also two(2) types of Multiple rows; Subquery which returns multiple column and subquery which returns multiple rows.
3. **Correlated Subquery:** This is a query that is related to the outer query.

SQL Clauses and Subqueries

Clauses Allowing Subqueries

The following clauses allow the integration of subqueries:

- WHERE Clause: Allows subqueries to filter and conditionally retrieve data based on specified criteria.
- FROM Clause: Enables subqueries to be used as a data source, returning results that can be utilized in the main query.
- HAVING Clause**:** Permits subqueries to filter grouped results when used with aggregate functions.

SQL Commands Supporting Subqueries

Several SQL commands support the integration of subqueries, allowing for a seamless combination of queries:

- SELECT Command: Allows subqueries to retrieve data that will be used in the main query's SELECT statement.
- INSERT, UPDATE, DELETE Commands: Subqueries can be used within these commands to conditionally modify or delete records based on the results of the subquery.

Advantages of WITH Clause in Subqueries

1) **Enhances Readability:** The WITH clause improves code organization, making it more readable and understandable.
2) **Reuse of Subquery Results:** Results from the WITH clause can be reused within the main query, avoiding redundancy.
3) **Simplifies Complex Queries:** Complex queries become more manageable and comprehensible with the use of the WITH clause.
4) **Avoidance of Repetition:** Reduces redundancy by preventing the need to repeat the execution of the same subquery.

Window Functions

Introduction

Window Functions in SQL are like special calculators that work on a specific group of rows at a time, kind of like looking at a window of data. This "window" is defined by an OVER clause and helps us analyse data in more detail. Unlike regular functions that look at all the rows, window functions let us focus on a smaller set of rows related to the current one, making our analysis more precise and flexible.

Types of Window Functions

1) NTILE Function: Divides the data into groups, like creating folders to organize information.

2) Row Number Function: Assigns a unique number to each row, like giving each row a special ID.

3) Rank Function: Ranks rows based on values, like giving them a position in a competition.

4) Dense Rank Function: Similar to Rank but without any gaps between the rankings, like a continuous list.

5) Lead Function: Looks ahead to the next row, like seeing what's coming next in a series.

6) Lag Function: Looks back at the previous row, like remembering what happened before.