Database Systems

Spring 2019

Lab Manual 5

Instructor: Muhammad Junaid Zaffar
Lab Assistant: Misha Zaheer

Purpose:

- Subqueries

Reading Content: T-SQL fundamentals Chapter 4

Instructions:- Do comment your code properly

Things to be covered in Subqueries:

- Self Contained Subqueries
    - Scaler
    - Multivalued
- Correlated subqueries
- CAST function
- EXSITS predicate

A subquery is a query that is nested inside a SELECT, INSERT, UPDATE, or DELETE statement, or inside another subquery. A subquery can be used anywhere an expression is allowed.

Subquery Fundamentals:-

A Subquery is also called an inner query or inner select, while the statement containing a subquery is also called an outer query or outer select.

Many Transact-SQL statements that include subqueries can be alternatively formulated as joins. Other questions can be posed only with subqueries. In Transact-SQL, there is usually no performance difference between a statement that includes a subquery and a semantically equivalent version that does not. However, in some cases where existence must be checked, a join yields better performance. Otherwise, the nested query must be processed for each result of the outer query to ensure elimination of duplicates. In such cases, a join approach would yield better results.

A subquery nested in the outer SELECT statement has the following components:

- A regular SELECT query including the regular select list components.
- A regular FROM clause including one or more table or view names.
- An optional WHERE clause.
- An optional GROUP BY clause.
- An optional HAVING clause.

The SELECT query of a subquery is always enclosed in parentheses. It cannot include a COMPUTE or FOR BROWSE clause, and may only include an ORDER BY clause when a TOP clause is also specified.

A subquery can be nested inside the WHERE or HAVING clause of an outer SELECT, INSERT, UPDATE, or DELETE statement, or inside another subquery. Up to 32 levels of nesting is possible, although the limit varies based on available memory and the complexity of other expressions in the query. Individual queries may not support nesting up to 32 levels. A subquery can appear anywhere an expression can be used, if it returns a single value.

If a table appears only in a subquery and not in the outer query, then columns from that table cannot be included in the output (the select list of the outer query).

Statements that include a subquery usually take one of these formats:

- WHERE expression [NOT] IN (subquery)
- WHERE expression comparison_operator [ANY | ALL] (subquery)
- WHERE [NOT] EXISTS (subquery)

In some Transact-SQL statements, the subquery can be evaluated as if it were an independent query. Conceptually, the subquery results are substituted into the outer query (although this is not necessarily how SQL Server actually processes Transact-SQL statements with subqueries).

There are three basic types of subqueries. Those that:

- Operate on lists introduced with IN, or those that a comparison operator modified by ANY or ALL.
- Are introduced with an unmodified comparison operator and must return a single value.
- Are existence tests introduced with EXISTS.

Subquery rules

A subquery is subject to the following restrictions:

- The select list of a subquery introduced with a comparison operator can include only one expression or column name (except that EXISTS and IN operate on SELECT * or a list, respectively).
- If the WHERE clause of an outer query includes a column name, it must be join-compatible with the column in the subquery select list.
- The **ntext**, **text**, and **image** data types cannot be used in the select list of subqueries.
- Because they must return a single value, subqueries introduced by an unmodified comparison operator (one not followed by the keyword ANY or ALL) cannot include GROUP BY and HAVING clauses.
- The DISTINCT keyword cannot be used with subqueries that include GROUP BY.
- The COMPUTE and INTO clauses cannot be specified.
- ORDER BY can only be specified when TOP is also specified.
- A view created by using a subquery cannot be updated.
- The select list of a subquery introduced with EXISTS, by convention, has an asterisk (*) instead of a single column name. The rules for a subquery introduced with EXISTS are the same as those for a standard select list, because a subquery introduced with EXISTS creates an existence test and returns TRUE or FALSE, instead of data.