SET Operators

SET operators in SQL combine the results of multiple queries into a single result set. As we know to combine table with columns, we use JOINS and when it comes to rows, we use SET Operators.

Types of SET operators:

UNION

UNION ALL

EXCEPT

INTERSECT

Syntax and Rules:

SET operator syntax combines the 2 SELECT statement.

*‘SELECT \* FROM CUSTOMERS*

*UNION*

*SELECT \* FROM ORDERS’*

Rules:

1. SET operator can be used almost in all clauses. Example – WHERE, JOIN, GROUP BY, HAVING. ORDER BY is allowed only once at the end of the query.

Example –

*‘SELECT \* FROM CUSTOMERS*

*JOIN*

*WHERE*

*GROUP BY*

*UNION*

*SELECT \* FROM ORDERS*

*JOIN*

*WHERE*

*GROUP BY*

*ORDER BY’*

1. Number of columns in each column must be same.

Example –

*‘SELECT FirstName, LastName FROM CUSTOMERS*

*UNION*

*SELECT First Name, LastName FROM ORDERS’*

1. Data types of the columns must be same.
2. The order of column in each query must be same.

Example –

*‘SELECT CustomerID, LastName FROM CUSTOMERS*

*UNION*

*SELECT EmployeeID, LastName FROM ORDERS’*

1. The column names in the result set are determined by the column names specified in the first query.

Example –

*‘SELECT CustomerID AS ID, LastName FROM CUSTOMERS*

*UNION*

*SELECT EmployeeID, LastName AS Last\_Name FROM ORDERS’*

1. Even if all rules are met and SQL shows no errors, the result may be incorrect. Incorrect column selection leads to inaccurate results.

Example –

*‘SELECT FirstName, LastName FROM CUSTOMERS*

*UNION*

*SELECT First Name, LastName FROM ORDERS’*

Error SQL:

*‘SELECT FirstName, LastName FROM CUSTOMERS*

*UNION*

*SELECT LastName, FirstName FROM ORDERS’*

UNION Operator –

Returns all district rows from both queries. Removes duplicates.

Example –

*‘SELECT userid, gold\_signup\_date FROM goldusers\_signup*

*UNION*

*SELECT userid, created\_date FROM sales’*

UNION ALL Operator –

Return all rows from the queries. This will not remove duplicates. UNION ALL is generally faster than UNION.

Example –

*‘SELECT userid, gold\_signup\_date FROM goldusers\_signup*

*UNION ALL*

*SELECT userid, created\_date FROM sales’*

EXCEPT Operator –

Returns distinct rows from the first query which are not found in the second query.

Example –

*‘SELECT userid, gold\_signup\_date FROM goldusers\_signup*

*EXCEPT*

*SELECT userid, created\_date FROM sales’*

INTERSECT Operator –

Returns only rows which are common. This removes duplicates.

Example –

*‘SELECT userid, gold\_signup\_date FROM goldusers\_signup*

*INTERSECT*

*SELECT userid, created\_date FROM sales’*

Combine Information

Combining multiple table information into one table before analysing the data. This is performed to make sure that the reporting is done with only one SQL query rather than multiple queries.

Example –

*‘SELECT \* FROM sales;*

*SELECT \* FROM users;*

*SELECT userid, created\_date FROM sales*

*UNION*

*SELECT userid, signup\_date FROM users’*

Never use asterisk when combining the tables and use column name instead because when the schema of the data is changed it won’t be noticed if used asterisk.

Use the step as ‘Source Table’ to make sure where the data is coming from.

Delta Detection

Data engineers use the most this technique to make sure no duplicates are getting inserted in the data warehouse. This is done with the use of EXCEPT operator.

Example –

If on day 1, two customers visit the shop the data warehouse will record 2 customers with ID 1 & 2. On day 2, two more customers came to visit the shop but this time one of the customers is with ID 1. This will now get duplicated in the warehouse but if we use EXCEPT this will eliminate the duplicates and the new customer with ID 3 will get in.

EXCEPT Use Case – Data Completeness Check –

This is the best practice to check the data. If the data from DB A and DB B are identical or needs to be migrated to another DB, the EXCEPT is used to check whether there is missing data or not. This is checked between the two DB and the result should be empty for confirmation. Now we have to also check other way around.