SQL SERVER FULL COURSE PART - 3

ALL YOU NEED TO KNOW TO GET GOING WITH MS SQL SERVER

SELECT Statement

- SELECT statement is used to fetch data from one or multiple database tables bases on one or multiple business criteria.
- SELECT * FROM TableName
- SELECT **Column_List** FROM TableName
- SELECT **SomeExpression** FROM Table Name
- SELECT ***/Column_List/SomeExpression** FROM TableName **WHERE** SomeCondition

SELECT Statement - WHERE Clause

- WHERE clause is used to fetch data based on some condition. Only those rows will be returned from database for which the condition will be true.

- SELECT * FROM TableName WHERE Condition

SELECT Statement – * (ALL)

- We use * symbol to represent all. When we write SELECT * FROM TableName, all the rows/records from that table will be returned.
- When we need to select all rows based on some condition then we can use where clause with SELECT statement, like SELECT * FROM TableName WHERE Condition.

SELECT Statement – DISTINCT

- When we want to remove duplicate records from the result set in that case, we use DISTINCT.
- If there are some duplicate data in the table. It removes duplicates and gives us the unique records only.
- **Unique data is identified based on combination of column list in the select statement.**

SELECT Statement – TOP

- TOP clause will return the top number of rows in the result based on the top_value.
- SELECT TOP (1) * FROM TableName
- SELECT TOP 10 ColumnList FROM TableName
- SELECT TOP 7 ColumnList FROM TableName WHERE Condition

SELECT Statement – ORDER BY

- The ORDER BY clause is used to sort the result set in some order.
- The default order is ascending.
- **ASC** keyword is used to sort the result set in ascending order.
- **DESC** keyword is used to sort the result set in descending order.

SELECT Statement – GROUP BY

- The GROUP BY Clause is used to divide(group) similar types of records as a group and then return.
- Generally, we use GROUP BY clause in the query when we use aggregate functions such as count(), sum(), max(), min(), and avg() etc.
- When we implement group by clause first the data of the table will be divided into the separate group as per the column.
- Then aggregate function will execute on each group data to get the result.
- Then final result will be returned.

SELECT Statement – HAVING Clause

- HAVING Clause is an additional filter that is applied to the result set produced by GROUP BY Clause.
- Logically, the HAVING Clause filters the rows from the intermediate result set that is built by using the FROM, WHERE, or GROUP BY clauses in the SELECT statement.

Difference Between HAVING and WHERE Clause

- WHERE clause cannot be used with aggregate functions whereas HAVING clause can be used with aggregate functions.
- WHERE clause is used for filtering individual rows on a table whereas the HAVING clause is used to filter groups.
- WHERE comes before GROUP BY. This means the WHERE clause filters rows before aggregate calculations are performed.
- HAVING comes after GROUP BY. This means the HAVING clause filters groups after aggregate calculations are performed.
- So, from a performance standpoint, HAVING is slower than WHERE and should be avoided when possible.
- WHERE clause can be used with – Select, Insert, and Update statements whereas HAVING clause can only be used with the Select statement.

Variables and Data Types

- Variable is an object that can hold a single data value of a specific type.
  - Local Variable
    - A user declares the local variable.
    - By default, a local variable starts with @.
    - Every local variable scope has the restriction to the current batch or procedure within any given session.
  - Global Variable
    - The system maintains the global variable. A user cannot declare them.
    - The global variable starts with @@
    - It stores session related information.
- Data Types are the attribute that specifies what types of data can be entered by the user such as integer, character, decimal, date time, etc.
- In SQL Server Database, each column of a table, all the local variables, and parameters must have a data type.

Types of Data Type

- Integer Data Types
- Decimal Data Types
- Money / Currency Data Types
- Date and Time Data Types
- Character Data Types
- Binary Data Types
- Special Data Types

Operators

- An operator is a symbol that performs some specific operation on operands.
- These operators are classified as follows in SQL Server
  - Assignment operator
  - Arithmetic operator
  - Comparison operator
  - Logical operator

Assignment Operator

- Equal to (**=**) sign is known as the assignment operator.
- It is the only assignment operator in MS SQL Server.
- SQL SERVER 2008 has introduced a new concept of Compound Assignment.

Arithmetic Operators

- + (Addition Operator)
- – (Minus Operator)
- * (Multiplication Operator)
- / (Division Operator)
- % (Modulo Operator)

Comparison Operators

- Equal (=) Operator
- Not Equal (!= or <>) Operator
- Greater Than (>) Operator
- Less Than (<) Operator
- Greater Than or Equal To (>=) Operator
- Less Than or Equal To (<=) Operator
- Not Greater Than (!<) Operator
- Not Less Than (!>) Operator

Logical Operators

- AND – TRUE if both Boolean expressions are TRUE.
- OR – TRUE if either Boolean expression is TRUE.
- NOT – Reverses the value of any other Boolean operator.

BETWEEN Operator

- BETWEEN operator is used to get the values within a range.
  - Between Operator returns true if the operand is within a range.
  - Between Operator will return records including the starting and ending values.
  - This operator support only the AND operator.
  - The BETWEEN Operator takes the values from small to big range in the query.
- If we use the NOT keyword along with the BETWEEN operator, then it will return data where the column values not in between the range values.

IN Operator

- We use IN operator in WHERE clause to compare column or variable values with a set of multiple values.
- If we use the NOT keyword along with the IN operator, then it will return data where column value not in the set of values

LIKE Operator

- The LIKE operator in SQL Server is used to search for character string with the specified pattern using wildcards in the column.
- Pattern means its specific string of characters with wildcards to search for matched expressions.
- Wild Card Characters
  - **%** symbol represents any no of characters in the expression.
  - **_** will represent a single character in the expression.
  - **[]** symbol indicates a set of characters in the expression.
  - **[^]** will represent any single character, not within the specified range
-

LIKE Operator

- Employee Name contains 3 characters.
  - SELECT * FROM TableName WHERE Name LIKE ‘___’
- Employee Name contains ‘A’ character.
  - SELECT * FROM TableName WHERE Name LIKE’%A%’
- Employee Name starts with ‘P’ character and ends with ‘A’ character.
  - SELECT * FROM TableName WHERE Name LIKE ‘P%A’
- Employee Name starts with J, H, K, U characters.
  - SELECT * FROM TableName WHERE Name LIKE'[J, H, K, U]%’
- Employee Names start with A to Z characters.
  - SELECT * FROM TableName WHERE Name LIKE'[A-Z]%’
- Employee Name not start with A to Z characters.
  - SELECT * FROM TableName WHERE Name NOT LIKE'[A-Z]%’

THANK YOU