# Single Row Functions
# &
# Group Functions

This presentation will be divided into two different areas.

It is important to remember that a function DOES NOT change the table or database. It will only adjust the returned data set.

1. Single Row Functions
2. Group Functions

# Types of Functions

There are two different type of functions. Both are powerful features of SQL.

Single row functions operate on a SINGLE row only and returns ONE results per row.

Multiple row functions manipulates GROUPS of rows to give ONE results per group of rows.

Single row function are used to manipulate date items. They accept one or more arguments and return one value for each row returned by the query.

An argument can be one of the following:

- User-supplied constant
- Variable Name
- Column Name
- Expression

# Features of a single row function

Acts on each row returned in the query.

Returns one result per row.

May return a data value of a different type than is referenced.

May accept one or more arguments.

Can be used in SELECT, WHERE and ORDER BY clauses and they can be nested.

Single row functions fall into five different categories.

| | |
|---|---|
| Character | General |
| Number | Date |
| Conversion | |

The types of categories shown is (are) a partial list.

# Character Functions

There are several functions that fall into this category.

| | | |
|---|---|---|
| Lower | Upper | Initcap |
| Concat | Substr | Length |
| Instr | Lpad | Trim |

## Characteristics of each function:

| | |
|---|---|
| LOWER | Coverts alpha characters to lowercase. |
| UPPER | Converts alpha characters to uppercase. |
| INITCAP | Coverts alpha characters values to uppercase for the first letters of each word, all other letters in lowercase. |
| CONCAT | Concatenates the first character value to the second character value, equivalent to using the concatenation operator. |
| SUBSRT | Returns specified characters from character value starting at the character position M, N characters long. |

## More characteristics of each function:

| | |
|---|---|
| LENGTH | Returns the number of characters in value. |
| INSTR | Returns the numeric position of a named character. |
| LPAD | Pads the characters value right-justified to a total width of *n* character positions. |
| TRIM | Enables you to trim leading or trailing character, or both, from a character string. If trim-character or trim-source is a character literal, you must enclose it in quotes. |

# Number Functions:

Number function accept numeric input and return numeric output.

| | |
|---|---|
| Round | Rounds the column, expression, or value to N decimal places or IF N is omitted, no decimal places. |
| TRUNC | Truncates the column, expression, or value to N decimal places or IF N is omitted, no decimal places. |
| MOD | Returns the remainder only when we divide one number by another. |

## Date functions:

When working with dates it is important to know the format of the dates stored in the table.

| | |
|---|---|
| MONTHS_BETWEEN | Number of months between two dates. |
| ADD_MONTHS | Add calendar months to a date. |
| NEXT_DAY | Next day of the date specified. |
| LAST_DAY | Last day of the month. |
| ROUND | Rounds the date. |
| TRUNC | Truncates the date. |

# Aggregating Data using Group Functions.

Unlike single-row functions, group function operate on sets of rows to give one result per group.

| | |
|---|---|
| AVG | COUNT |
| MAX | MIN |
| STDDEV | SUM |
| VARIANCE | |

## Descriptions of group functions:

| | |
|---|---|
| AVG | Average value of N, ignoring null values. Numeric only. |
| COUNT | The number of rows, where expression evaluates to something other than null. |
| MAX | Maximum value of expression, ignoring null values. Can be used on all data types. |
| MIN | Minimum value of expression, ignoring null values. Can be used on all data types. |
| STTDEV | Standard deviation of n, ignoring null values. Numeric only. |
| SUM | Sum values of n, ignoring null values. Numeric only. |
| VARIANCE | Variance of n, ignoring null values. Numeric only. |

## Guidelines for using Group Function:

DISTINCT makes the function consider only non-duplicated values.

ALL makes it consider every value including duplicates. This is the default.

Data types of values can be: CHAR, VARCHAR, NUMBER, or DATE.

All group function except COUNT ignore null values.

# Creating Groups of data:

When we need to create grouped of data into small groups we have to use the GROUP BY clause. The rules for using the group by clause are as follows:

If you include a group function in a SELECT clause, you cannot select individual results as well UNLESS the individual column appears in the GROUP BY clause. Sorted by default in ascending order.

Using a WHERE clause, we can pre-exclude rows before dividing them into groups.

You must include the column(s) in the GROUP BY clause.

You cannot use a column alias in the GROUP BY clause.

## Using the GROUP BY clause:

When using the group by clause, make sure that all columns in the select list that are not in the group function are included.

The SELECT clause specifies the columns to be retrieved.

The FROM clause specifies the table(s).

The WHERE clause specifies the rows to be retrieved. If the WHERE clause is not used, all rows will be returned.

The GROUP BY clause specifies how the rows should be grouped.

## Other characteristics of the GROUP BY clause:

The Group by column does not have to be in the SELECT clause.

We can have groups within groups. We do this by adding the multiple column names in the GROP BY clause.

We can restrict or exclude the rows returned. Since it is a group function we have to use the HAVING clause.

The order in which SQL evaluates the statement is:

1. The rows are grouped.
2. The group function is applied to the group.
3. The groups that match the criteria in the HAVING clause are displayed.