UNIT 4
SQL

AGGREGATE FUNCTIONS

WHAT ARE AGGREGATE FUNCTIONS ?

- Aggregate functions allow us to easily produce **summarized data from our database**.
- SQL aggregation function is used to perform the calculations on multiple rows of a single column of a table. **It returns a single value**.
- Aggregate Functions are all about :
  - Performing calculations on multiple rows
  - Of a single column of a table
  - And returning a single value - they operate on sets of rows and return results based on groups of rows.

TYPES OF AGGREGATE FUNCTIONS ?

- The ISO standard defines five (5) aggregate functions namely;

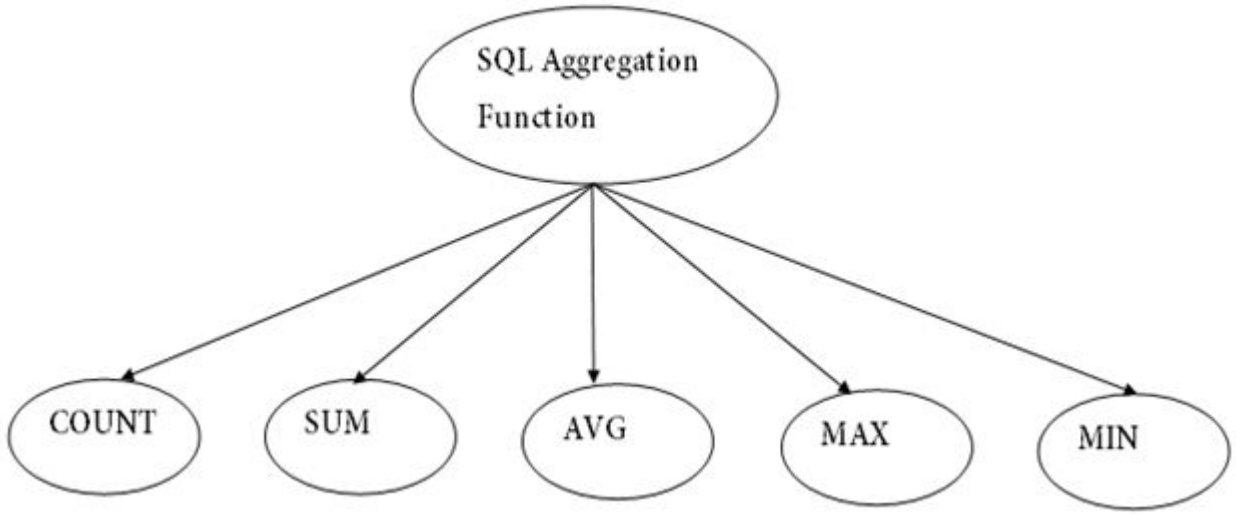
1) COUNT

2) SUM

3) AVG

4) MIN

5) MAX



Note : All aggregate functions by default exclude nulls values before working on the data.

SAMPLE TABLE

- Table shown below shows data in emp table:

eid	name	age	salary
401	Anu	22	9000
402	Shane	29	8000
403	Rohan	34	6000
404	Scott	44	10000
405	Tiger	35	8000

COUNT

- COUNT function is used to Count the number of rows in a database table.
- It works on both numeric and non-numeric data types.
- COUNT (*) is a special implementation of the COUNT function that returns the count of all the rows in a specified table. **COUNT (*) also considers Nulls and duplicates.**
- Its general **syntax** is,

```
SELECT COUNT(column_name) FROM table-name
```

COUNT

- SQL query to count employees is,

  SELECT COUNT(eid) FROM Emp;

- Result of the above query will be,

count(eid)
5

eid	name	age	salary
401	Anu	22	9000
402	Shane	29	8000
403	Rohan	34	6000
404	Scott	44	10000
405	Tiger	35	8000

COUNT

- SQL query to count employees, satisfying specified condition is,

  SELECT COUNT(name) FROM Emp WHERE salary = 8000;

- Result of the above query will be,

count(name)
2

eid	name	age	salary
401	Anu	22	9000
402	Shane	29	8000
403	Rohan	34	6000
404	Scott	44	10000
405	Tiger	35	8000

COUNT (**distinct**)

- SQL query is,

  SELECT COUNT(DISTINCT salary) FROM emp;

- Result of the above query will be,

count(distinct salary)

4

eid	name	age	salary
401	Anu	22	9000
402	Shane	29	8000
403	Rohan	34	6000
404	Scott	44	10000
405	Tiger	35	8000

MIN

- The MIN function **returns the smallest value in the specified table field**.
- MIN function is used to find the minimum value of a certain column.
- This function determines the smallest value of all selected values of a column.
- Its general **syntax** is:

```
SELECT MIN(column_name) from table-name;
```

MIN

- SQL query to find minimum salary is,

  SELECT MIN(salary) FROM emp;

- Result will be,

  MIN(salary)

  6000

eid	name	age	salary
401	Anu	22	9000
402	Shane	29	8000
403	Rohan	34	6000
404	Scott	44	10000
405	Tiger	35	8000

MAX

- Just as the name suggests, the MAX function is the opposite of the MIN function.
- It **returns the largest value from the specified table field**.
- Its general **syntax** is:

```
SELECT MAX(column_name) from table-name;
```

MAX

- SQL query to find MAXIMUM salary is,

  SELECT MAX(salary) FROM emp;

- Result will be,

  MIN(salary)

  10000

eid	name	age	salary
401	Anu	22	9000
402	Shane	29	8000
403	Rohan	34	6000
404	Scott	44	10000
405	Tiger	35	8000

SUM

- **SUM** function which **returns the sum of all the values in the specified column**.
- **SUM works on numeric fields only**.
- **Null values are excluded from the result returned.**
- Its general **syntax** is:

```
SELECT SUM(column_name) from table-name;
```

SUM

- SQL query to find sum of salaries will be,

  SELECT SUM(salary) FROM emp;

- Result of above query is,

SUM(salary)
41000

eid	name	age	salary
401	Anu	22	9000
402	Shane	29	8000
403	Rohan	34	6000
404	Scott	44	10000
405	Tiger	35	8000

AVG

- AVG function **returns the average of the values in a specified column**.
- Just like the SUM function, it **works only on numeric data types**.
- AVG function returns the average of all non-Null values.
- Its general **syntax** is:

```
SELECT AVG(column_name) FROM table_name
```

AVG

- SQL query to find average salary will be,

  SELECT avg(salary) from Emp;

- Result of the above query will be,

  avg(salary)

  8200

eid	name	age	salary
401	Anu	22	9000
402	Shane	29	8000
403	Rohan	34	6000
404	Scott	44	10000
405	Tiger	35	8000

RENAMING COLUMNS

- select sum(age) AS SUM, avg(age) AS AVG, min(age) AS MIN, max(age) AS MAX from customers;

SUM	AVG	MIN	MAX
105	21	20	22

- select sum(age) SUM, avg(age) AVG, min(age) MIN, max(age) MAX from customers;

SUM	AVG	MIN	MAX
105	21	20	22