# DBMS NOTE: 3

Aggregate Functions: In SQL, like in any other programming language we have some predefined functions that perform some specific tasks. Most of these functions are numeric in nature and aggregate numerical values, hence Aggregate Functions.

Some common aggregate functions are:

1. SUM: The SUM() function returns the total sum of a numeric column.

```
SELECT SUM(column_name)

FROM table_name

WHERE condition;
```

2. AVG: The AVG() function returns the average value of a numeric column.

```
SELECT AVG(column_name)
FROM table_name
WHERE condition;
```

3. COUNT: The COUNT() function returns the number of rows that matches a specified criterion.

```
SELECT COUNT(column_name)
FROM table_name
WHERE condition;
```

**Assignment:** Apply the MAX() and MIN() Functions is your database and show the output. Hint: The syntax is very similar to the other aggregate functions.

**Obtaining Results From Two Tables:** There might occur some scenarios where the information you need is spread across two or three tables. So to fetch those information we use something known as joins. Here we will only use **Equi Join**.

In Equi Join only the rows of the both tables are joined whose values of a particular column match.

Syntax:

```
SELECT column_list

FROM table1, table2....

WHERE table1.column_name =

table2.column_name;
```

**UNION:** The UNION operator is used to combine the result-set of two or more SELECT statements.

1. Every SELECT statement within UNION must have the same number of columns
2. The columns must also have similar data types
3. The columns in every SELECT statement must also be in the same order.

Syntax:

```
SELECT column_name(s) FROM table1

UNION

SELECT column_name(s) FROM table2;
```

**CARTESIAN Product (CROSS JOIN):** A Cartesian product combines each row from the first table with every row from the second table, resulting in a result set that contains all possible combinations of the rows from both tables.

Syntax:

```
SELECT *

FROM table1

CROSS JOIN table2;
```