Grouping

- Grouping in SQL is done using GROUP BY. GROUP BY statement groups rows that have the same values into summary rows, like “find the accounts in each branch”
- After using the GROUP BY clause, all the rows with the same values of the specified column name will get summarized into a single row.
- Note that all the column names mentioned after SELECT statement shall be repeated in GROUP BY, in order to successfully execute the query.
- **Aggregate functions** - These are used to perform calculations on multiple values of a column and return the result into a single value. The aggregate functions are explained below:

Function	Description
COUNT()	The number of rows returned based on condition.
AVG()	The average of the value is returned in the selected column.
MAX()	The maximum value of a column is returned.
MIN()	The minimum value of a column is returned.
SUM()	The sum of the values in a specified column is returned.

- **General form**:-

  Query:-

```
SELECT column_name(s)
FROM T_name
WHERE  condition
GROUP BY column_name(s);
```

Ex-

Table : Ninjas

ID	Ninja's Name	City
101	Lokesh Ninja	Kolkata
102	Kuldeep Ninja	Bhopal
103	Raju Ninja	Kolkata
104	Ojasv Ninja	Shimla
105	Abhi Ninja	Bhopal
106	Tarun Ninja	Bhopal

Find the number of Ninjas in each city ?

```
SELECT COUNT(ID), City
FROM Ninjas
GROUP BY City
ORDER BY COUNT(ID) DESC;
```

Output:

COUNT(ID)	City
1	Shimla
2	Kolkata
3	Bhopal

Note: In the above query we have used the aggregate function COUNT().

Now if we want only CITY with COUNT = 2 (lets say), we have to use the HAVING clause.

```
SELECT COUNT(ID), City
FROM Ninjas
GROUP BY City
HAVING  COUNT(ID) = 2;
```

Output:

COUNT(ID)	City
2	Kolkata

- **General form**:-

Query:-

```
SELECT column_name(s)
FROM T_name
WHERE condition
GROUP BY column_name(s)
HAVING condition;
```

- **COUNT(*) function -**

This function counts (sums up) the total number of rows that satisfy the specified condition, and if there is no condition, then it simply returns the number of records in a table.

Syntax:

```
SELECT COUNT(*)
FROM <table-name>;
```

Example: Find the total number of records in the table Ninjas.

Table: Ninjas

ID	Ninja's Name	City
101	Lokesh Ninja	Kolkata
102	Kuldeep Ninja	Bhopal
103	Raju Ninja	Kolkata
104	Ojasv Ninja	Shimla
105	Abhi Ninja	Bhopal
106	Tarun Ninja	Bhopal

QUERY -
SELECT COUNT(*)
FROM Ninjas;

Output:
6

- **HAVING clause :**

The HAVING clause is used to filter group of data as per specified conditions. It is used with GROUP BY clause to get the results of column operations.

The HAVING and WHERE clause serve similar purpose, but there is slight difference regarding the usage in the query and output in the result.

Syntax :

SELECT *<column-name>*
FROM *<table-name>*
GROUP BY *<column-name>*
HAVING *<condition>*

Example: Find the total number of ninjas who belongs to Bhopal, also display the City column.

Table: Ninjas

ID	Ninja's Name	City
101	Lokesh Ninja	Kolkata
102	Kuldeep Ninja	Bhopal
103	Raju Ninja	Kolkata
104	Ojasv Ninja	Shimla
105	Abhi Ninja	Bhopal
106	Tarun Ninja	Bhopal

QUERY –

```
SELECT City, COUNT(*)
FROM Ninjas
GROUP BY City
HAVING City = "Bhopal";
```

Output:

City	COUNT(*)
Bhopal	3

- **Usage of HAVING and WHERE clause.**

WHERE	HAVING
WHERE clause is used to filter data based on certain specific condition.	HAVING clause is used to filter group of data based on certain conditions.
It can be used without use of GROUP BY clause.	It cannot be used without using GROUP BY clause.
It is used before GROUP BY clause.	It is used after GROUP BY clause.
It can be used with SELECT, UPDATE and DELETE statements.	It can only be used with SELECT statement.
It is implemented in row operations.	It is implemented in column operations.

Syntax with position of various clauses -

```
SELECT <column-name>
FROM <table-name>
WHERE <condition>
GROUP BY <column-name>
HAVING <condition>
ORDER BY <column-name>
```