SQL CLAUSES:

GROUP BY - Group by is used to group the data based on specified conditions. It group the values of all rows of specified condition.

ORDER BY - Order by is a clause it arranges the data in ascending or Descending order based on specified column.

HAVING - Having is a clause, it perform operations after the GROUP BY, this means the values consists after grouped data will be retrived based on having condition.

WHERE - "WHERE" does go to particular rows of columns to check the give specified condition wheather its "True" or "False". If "True" it retrives the data of given condition.

INFOSYS TABLE

Employee ID	Salary	Department	Working hours	Manager ID
1001	20,000	Python	8	4520
1002	30,000	Azure	8	4520
0011	25000	AWS	10	4520
0022	40000	SQL	7	4522
2011	15000	SQL	7	4523
2012	50000	Data Science	12	4522

Note: GROUP BY is often used with functions like COUNT(), MIN(), MAX(), AVG(), SUM().

GROUP BY:

```
SELECT Department, sum(salary)
FROM INFOSYS
GROUP BY Department
```

Department	sum(salary)
Python	20,000
Azure	30000
AWS	25000
SQL	55000
Data science	50000

GROUP BY with where clause:

```
SELECT Depar AVG(salary), Working hours
FROM INFOSYS
WHERE Working hours < 8
GROUP BY Working hours.
```

Working hours	AVG (SALARY)
7	27500

Note: Group by is followed after where clause

--> WHERE ---> GROUP BY.

Group by with "where" and "Having clause":

```
SELECT EmployeeID, Salary, Department, count(*)
From INFOSYS
WHERE Working hours ≥ 8
GROUP BY Employee ID
HAVING Salary > 20,000
```

Count(Empolyee ID)	Salary	Department.
1	30000	Azure
1	25000	AWS
1	50000	Data science.

Note: Group by is followed between Where and Having clause

--> WHERE --> GROUP BY ---> HAVING

Logical Operators:

AND - AND operators compare's the condition values and if True it retrives the following data.

> Note: ~~When~~ Retrives data only when both the conditions are "True".

OR - OR operators check the condition values if any one of the condition satisfied (True) it retrives the following data of "True".

NOT - NOT operators excludes the given specified list values and retrives the data which are not in list.

```
SELECT Department, salary, Employee ID
FROM INFOSYS
WHERE Department = "SQL" AND salary>= 40000
```

Department	Salary	Employee ID
SQL	40000	0002

```
SELECT Department, salary, Employee ID
FROM INFOSYS
WHERE Department = "python" OR salary < 20000
```

Department	salary	Employee ID
Python	20000	1001

Note: Even though salary is False it retrives the Department values.

```
SELECT Department, salary, Employee ID
FROM INFOSYS
WHERE salary NOT salary = [20000, 30000, 50000]
```

Department	salary	Employee ID
AWS	25000	0011
SQL	40000	0022
SQL	15000	2011

Between & IN Operators

Between - Between operators displays the rows values in the following range.

IN - IN operators returns the values which are matching in the lists.

```
SELECT Department, salary
FROM INFOSYS
WHERE salary BETWEEN 25000 AND 40000
```

Department	salary
25000	AWS
30000	Azure
40000	SQL

```
SELECT Department, salary
FROM INFOSYS
WHERE salary NOT BETWEEN 25000 AND 40000
```

Department	salary
20000	python
15000	SQL
50000	Data science

```
SELECT Department, salary
FROM INFOSYS
WHERE salary IN (25000, 30000)
```

Department	salary
AWS	25000
Azure	30000

```
SELECT Department, salay.
FROM INFOSYS
WHERE salary NOT IN (25000, 30000, 40000, 20000)
```

Department	salary
SQL	15000
Data science	50000

LIKE clause : "Like" is a logical operator that search & matches the pattern by using wildcard operators.

```
SELECT Department WHERE Department LIKE 'S%'
```

output:

Department
SQL
SQL

```
SELECT Department WHERE Department LIKE '%e'
```

Department
Azure
Data science

```
SELECT Department WHERE Department LIKE 'S%L'
```

Department
SQL
SQL

```
SELECT Department WHERE Department LIKE '%zu_'
```

Department
Azure