Feature	Where	HAVING
Usage	Filters rows before grouping	Filter groups after Group by
Works With	SELECT, UPDATE, DELETE	SELECT (only with GROUP BY)
Used On	Individual rows	Grouped data (aggregates)
Can use Aggregates?	☒ No (cannot use SUM, COUNT, etc.)	☑ Yes (can use SUM, COUNT, etc.)
Example	Select * From sales where price > 100;	Select category, SUM(price) From sales Group By category Having sum(price) > 500;

Order of Execution of SQL Statement :-

FROM → Determines the tables involved

JOIN → Combines tables if necessary.

WHERE → Filters rows based on columns.
Group by → Groups data based on columns.
Having → Filters grouped data.
Select → Chooses the columns to display.
Order by → Sorts the results.
LIMIT → Limits the number of rows

In SQL, keys are used to identify & ensure the integrity of data in tables. They play a crucial role in managing relationships b/w tables & ensuring that data remains consistent.

1. **Primary key** :- A primary key is a unique identifier for each record in a table. It ensures that each row has a unique value & cannot be NULL.

- It is used to uniquely identify records in a table.
- Only one primary key can exist in a table
- Automatically enforces uniqueness & non-nullability.

2. **Candidate key** :- A candidate key is a set of one or more attributes that can uniquely identify a record in a table. Every candidate key is a potential primary key.

- A table can have mutiple candidate keys.
- A candidate key can be chosen as the primary keys.
- They must be unique & non-null.

Ex:- In a table, if EmployeeID & Email are both unique attributes that could serve as the primary keys, the both EmployeeID & Email are Candidate keys.

5. **Unique key** :- Ensures all values in the column are unique.
Allows NULL values (but only one NULL per column)

3. **Super key** :- A super key is a set of one or more attributes that can uniquely identify a record in a table. It may contain extra attributes that are not necessary for uniqueness (i.e., it could be a ~~sus~~ superset of a candidate key).

- Every primary key is a super key, but not every super key is a primary key.
- A Super key can have additional attributes beyond the minimal unique identifiers.

Ex:- In the case of EmployeeID & Email, a Super key could be EmployeeID, Name.
(since EmployeeID alone is enough to uniquely identify, but adding Name makes it a super key.)

4. **Foreign key** :- A foreign key is an attributes (or set of attributes) in one table that is used to link to the primary key in another table. It establishes & enforces a relationship b/w two tables.

- Ensure referential integrity b/w 2 tables (i.e., ensures that data in one table corresponds to valid data in another)
- It can have duplicate values & NULLs, depending on the constraints.

Here, EmployeeID in the Orders table is a foreign key that links to the EmployeeID in the Employees tables.

ex:- create table Orders (OrderID int Primary key, EmployeeID int, foreign key (EmployeeID) Reference Employees (EmployeeID));