**Set Operators:** Set operators combine results from two or more queries into a

single result set. SQL Server provides the following set operators.

* UNION
* UNION ALL
* INTERSECT
* EXCEPT

To combine the results of two queries we need to follow the below basic rules.

* The number and the order of the columns must be the same in all queries.
* The data types must be compatible(Well-Matched)

**UNION:** it combines the result of two or more select statements into a single result

set that includes all the records belongs to all queries except duplicate values.

Create table sample (eno int, ename varchar (10), sal int)

Insert into sample values (10,'ganesh', 2400)

Create table sample1 (eno int, ename varchar (10), sal int)

Insert into sample1 values (10,'ganesh', 2400)

* Ex: Select ename from sample

Union

Select ename from sample1

**UNION ALL:** it is same as union but returns duplicate values

* Ex: Select ename from sample

Union All

Select ename from sample1

**INTERSECT:** INTERSECT returns any distinct values that are common in left

and right tables.

* Ex: Select ename from sample

Intersect

Select ename from sample1

**EXCEPT:** EXCEPT returns any distinct values from the left query that are not

found on the right query.

* Ex: Select ename from sample

Except

Select ename from sample1

**CLAUSES IN SQL:** We can add these to a query for adding additional

options like filtering the records, sorting records and grouping the records with in a table. These clauses contains the following clauses are,

**WHERE:** This clause is used for filter or restricts the records from the table.

Ex: SELECT \* FROM EMP WHERE SAL=10000

**ORDER BY:** The order by clause is used to sort or arrange the data in

ascending or descending order with in table. By default order by clause

arrange or sort the data in ascending order only.

* If we want to arrange the records in a descending order then we

use Desc keyword.

* We can apply order by clause on integer and string columns.

Ex: SELECT \* FROM EMP ORDER BY EID (For Ascending Order)

Ex: SELECT \* FROM EMP ORDER BY ENAME DESC (For Descending Order)

**TOP N CLAUSE:** This clause is used to fetch a top n number of records from a

table.

* Ex: SELECT TOP 3 \* FROM EMP

**GROUP BY:** Group by clause will use for to arrange similar data into groups.

when we apply group by clause in the query ten we use group functions like

count(),sum(),max(),min(),avg().

If we use group by clause in the query, first the data in the table will be divided

into different groups based on the columns and then execute the group f unction on

each group to get the result.

**Ex1:** WAQ to find out the number of employees working in the organization

**Sol:** SELECT COUNT (\*) FROM EMP

**Ex2:** WAQ to find out the number of employees working in each group in the

organization.

**Sol:** SELECT DEPT, COUNT=COUNT (\*) FROM EMP GROUP BY DEPT

**Ex3:** WAQ to find out the total salary of each department in the organization

**Sol:** SELECT DEPT, TOTALSALARY=SUM (SALARY) FROM EMP GROUP

BY DEPT (Like this we can find max, min, avg salary in the organization)

**HAVING CLAUSE:** Having clause is also used for filtering and restricting the

records in a table just like where clause.

Ex: WAQ to find out the number of employees in each department only if the

count is greater than 3

**Sol:** SELECT DEPT, COUNT=COUNT (\*) FROM EMP GROUP BY DEPT

HAVING COUNT (\*) >3

**Differences Between WHERE and HAVING Clause:**

|  |  |
| --- | --- |
| **WHERE** | **HAVING** |
| WHERE clause is used to filter and  restrict the records before grouping | HAVING clause is used to filter and  restrict the records after grouping |
| If restriction column associated with  A aggregative function then we cannot  use WHERE clause there | But we can use HAVING clause at this  situations |
| WHERE clause can apply without group  by clause | HAVING clause cannot be applied  without a group by clause |
| WHERE clause can be used for  restricting individual rows | Where as HAVING clause is used along  with group by clause to filter or restrict  groups |
| WHERE clause cannot support group  Functions | HAVING clause support group function |
|  |  |