**ORDER BY clause**

It is useful to print column values in a specific order like ascending or descending order.

**Syntax:**

Select col1, col2, . . . or \* FROM table ORDER BY col1, col2, . . . [DESC];

Ex:

Get employee names in alphabetical order?

Select ename from emps order by ename;

Ex:

Get employee details like old employee to latest employee?

Select \* from emps order by joindt;

**Note:**

*We can also apply order by clause on multiple columns.*

Ex:

Get employee details like low salary to higher salary from each designation?

Select \* from emps order by disig, salary;

**Arithmetic Operators:**

These are useful to perform arithmetic operations on given values and also on column values.

**+ - \* / %**

Sample Examples:

Select 10+20

Select 100\*5

Select 200/4

Select 320%3

Select 300-200

**Relational Operators:**

These are useful to compare values by writing conditions.

We will write conditions on columns.

**< > = <= >= !=**

**WHERE clause:**

In ***SELECT, UPDATE*** and ***DELETE*** queries, we will write ***conditions*** under WHERE clause.

In SELECT query, we will write conditions to filter table data.

**Syntax:**

SELECT . . . . . FROM . . . . . WHERE <condition>;

Ex: Get the details of empid 1010?

Select \* from emps where empid=1010;

Ex: Get the details of all developers?

Select \* from emps where disig = ’developer’;

Ex: Get employee names and salaries with min salary 40000?

Select ename, salary from emps where salary >= 40000;

Ex: Get employee details who joined on '2020/01/22'?

select \* from emps where joindt='2020/01/22';

**Special Operators:**

**BETWEEN**

We will write a condition based on range of values.

Syntax:

Select . . . FROM . . . WHERE column name **BETWEEN** val1 **AND** val2;

Ex: Get employee details with min salary 40000 and max salary 50000?

Select \* from emps where salary between 40000 and 50000;

**NOT BETWEEN**

It is the negation of BETWEEN operator. It will not search for given range.

**IN**

We will write a condition based on multiple values in a column or we can search for multiple values in a column.

Syntax:

Select . . . FROM . . . WHERE column name **IN (** val1, val2, . . .);

EX:

-- display admin details and developer details also?

select \* from Emps where disig IN('admin','developer');

Ex: Display emp details with id 1010,1020,1030?

select \* from emps where empid in(1010,1020,1030);

**NOT IN**

It is the negation of IN operator. It will not search given list of values.

Syntax:

Select . . . FROM . . . WHERE column name **NOT IN (** val1, val2, . . .);

Ex: Display emp details other than 1010,1020,1030 emps?

select \* from emps where empid NOT IN (1010,1020,1030);