IN Operator

It allows us to specify multiple values in where clause.

It filters rows that match any value given in list.

SYNTAX:
Select column_name(s)

From table_name

Where column_name IN (value1, value2,……);

Ex.Selecting employees with specific job titles(Manager , Clerk)

Select ename, job

From emp

Where job = ‘Manager’ OR job = ‘Clerk’ ;

```
mysql> Select ename, job
    -> From emp
    -> Where job = 'Manager' OR job = 'Clerk';
+--------+---------+
| ename  | job     |
+--------+---------+
| SMITH  | CLERK   |
| JONES  | MANAGER |
| BLAKE  | MANAGER |
| CLARK  | MANAGER |
| ADAMS  | CLERK   |
| JAMES  | CLERK   |
| MILLER | CLERK   |
+--------+---------+
```

```
Select ename, job
From emp
Where job IN (’ Manager’ ,  ‘Clerk’ );
```

```
mysql> Select ename, job
    -> From emp
    -> Where job IN ('Manager', 'Clerk');
+--------+---------+
| ename  | job     |
+--------+---------+
| SMITH  | CLERK   |
| JONES  | MANAGER |
| BLAKE  | MANAGER |
| CLARK  | MANAGER |
| ADAMS  | CLERK   |
| JAMES  | CLERK   |
| MILLER | CLERK   |
+--------+---------+
```

ResultSet will be same in both queries but IN is faster one

Ex. Finding orders with specific status

Table: order

Columns: OrderID, Status(Shipped, Pending)

Ex. Retrieving students enrolled in specific courses.

- Is NULL Operator
  - This operator tests for null values in a column
  - NULL means a missing data or empty data.

SYNTAX:

```
SELECT column_name(s)
From table_name
Where column_name is null;
```

Ex. Selecting employees without any commission.

```
Select ename, comm
From emp
Where comm is null;
```

Select employee names and salaries where commission is not null(e.g employees who receives a commision)

- Finding orders without a ship date.

```
Select order_id, ship_date

From orders

Where ship_date is null;
```

- Finding products without any description.

LIKE Operator:

- Used to search for patterns in column
- It works with two wildcards
  1. % - represents zero or more characters
  2. _ - represents a single character.

Ex. Selecting employees whose name starts with ‘S’ .

```
Select ename

From emp

Where ename like ‘S%’ ;
```

```
mysql> Select ename
    -> From emp
    -> Where ename like ’S%’;
+--------+
| ename |
+--------+
| SMITH |
| SCOTT |
+--------+
```

Ex. Retrieving names ending with ‘S’.

```
mysql> select ename
    -> from emp
    -> where ename like ’%S’;
+--------+
| ename |
+--------+
| JONES |
| ADAMS |
| JAMES |
+--------+
```

Ex. Names containing E.

Ex. Names with 'L' at 3rd position.

Ex.5 character names.

- ORDER BY CLAUSE

It sorts the resultset in the ascending or descending order.

SYNTAX:

```
Select column_name(s)
From table_name
Order by column_name[ASC|DESC];
```

Ex. Sort employees by salary in ascending order.

```
Select ename, sal
From emp
Order by sal asc;
```

```
mysql> Select ename, sal
    -> From emp
    -> Order by sal asc;
+--------+---------+
| ename  | sal     |
+--------+---------+
| SMITH  |  800.00 |
| JAMES  |  950.00 |
| ADAMS  | 1100.00 |
| WARD   | 1250.00 |
| MARTIN | 1250.00 |
| MILLER | 1300.00 |
| TURNER | 1500.00 |
| ALLEN  | 1600.00 |
| CLARK  | 2450.00 |
| BLAKE  | 2850.00 |
| JONES  | 2975.00 |
| SCOTT  | 3000.00 |
| FORD   | 3000.00 |
| KING   | 5000.00 |
+--------+---------+
```

- Sort the employees in the descending order by their salary.

Select ename, sal

From emp

Order by sal desc;

- Sorting employees names alphabetically
- Sorting product names alphabetically
  - ➔ Select product_name,price
  - ➔ From products
  - ➔ Order by product_name asc;

LIMIT Clause

It specifies the number of rows to return in the resultset.

SYNTAX:

Select column_name(s)

From table_name

Limit number_of_records;

- To retrieve top 5 highest paid employees.

```
mysql> select ename, sal
    -> from emp
    -> order by sal desc
    -> limit 5;
+-------+---------+
| ename | sal     |
+-------+---------+
| KING  | 5000.00 |
| SCOTT | 3000.00 |
| FORD  | 3000.00 |
| JONES | 2975.00 |
| BLAKE | 2850.00 |
+-------+---------+
```

Ex. Selecting first 10 students.

```
Select student_name, grade
From student
Limit 10;
```

Operator/clause	Use case	Features
BETWEEN…AND	FILTERS RECORDS IN ROWS	INCLUDES START AND END VALUE
IN	MATCHES MULTIPLE VALUES	BETTER INSTEAD OF USING MULTIPLE ORs
LIKE	Search for a pattern	Uses % and _
IS NULL	TEST MISSING VALUES	CHECKS FOR NULL
ORDER BY	SORTS RESULT	SORTS IN ASC OR DESC ORDER
LIMIT	LIMITS THE NUBER OF ROWS IN A RESULTSET	CONTROLS NUMBER OF ROWS WE CAN DISPLAY

Offset:

It specifies how many rows to skip before starting to return the resultset.

SYNTAX:

```
select column_name(s)
From emp
LIMIT number_of_rows offset offset_value;
```

```
mysql> select ename,job,sal
    -> from emp
    -> order by sal desc
    -> limit 5 offset 2;
+-------+----------+---------+
| ename | job      | sal     |
+-------+----------+---------+
| FORD  | ANALYST  | 3000.00 |
| JONES | MANAGER  | 2975.00 |
| BLAKE | MANAGER  | 2850.00 |
| CLARK | MANAGER  | 2450.00 |
| ALLEN | SALESMAN | 1600.00 |
+-------+----------+---------+

mysql> select ename, job, sal
    -> from emp
    -> order by sal desc
    -> limit 2,5;
+-------+----------+---------+
| ename | job      | sal     |
+-------+----------+---------+
| FORD   | ANALYST   | 3000.00 |
| JONES  | MANAGER   | 2975.00 |
| BLAKE  | MANAGER   | 2850.00 |
| CLARK  | MANAGER   | 2450.00 |
| ALLEN  | SALESMAN  | 1600.00 |
+--------+----------+---------+
```

Ex.Add a fixed amount(400) to each employees salary and display the result.

Ex.Subtract a fixed amount 200 from each emp salary ‘Updated Salary’ .

Ex. Calculate a 10% bonus for each employee’ s salary and display.

TASK:

Add 100 rupees to salary, subtract 50, multiply by 2 and divide it by 3.

TASK:

Show employees whose updated salary(after adding 400) is greater than 2000.

Ex.Finding unique job roles in emp table.

Distinct*

Group By*

Aggregate functions*