

1. BETWEENAND Operator

- This Operator is used to filter records based on given range of values.
- It includes both the start and end values of range.

- SYNTAX:

```
SELECT column_Name(s)
FROM table_name
WHERE column_name BETWEEN value1 AND value2;
```

- Features:

- Works with numeric, text and date values

Ex. Selecting employees hire between two dates.

```
SELECT ename, hiredate
FROM EMP
WHERE HIREDATE BETWEEN '1981-04-01' AND '1981-
05-01';
```

Ex. Finding products priced between 10 and 50

Columns: Product_Name, PRICE

Table : products

```
SELECT Product_Name, PRICE
FROM Products
WHERE PRICE between 10 and 50;
```

Ex. Retrieving students with grades between 80 to 90

2. IN Operator:

It allows us to specify multiple values in WHERE clause.

It filters rows that match any value in a given list.

SYNTAX:

```
SELECT column_name(s)
FROM table_name
WHERE column_name IN (value1, value2, ....., valueN);
```

Advantage:

Reduces the need of typing multiple OR conditions

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

```
SELECT ENAME, JOB
FROM EMP
WHERE JOB=' MANAGER' OR JOB = 'CLERK' ;
```

```
SELECT ENAME, JOB
FROM EMP
WHERE JOB IN ( 'MANAGER' , ' 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)

```
SELECT ORDERID, STATUS  
FROM ORDER  
WHERE STATUS IN ( 'SHIPPED' , ' PENDING' );
```

Ex. Retrieving students enrolled in specific courses.

3. IS NULL Operator

- This Operator tests for NULL values in a column.
- NULL represents missing 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;
```

Ex. Select the employee names and salaries where commission is not null (i.e., employees who receive a commission).

```
SELECT ENAME, SAL
FROM EMP
WHERE COMM IS NOT NULL;
```

Ex. Finding orders without a ship date.

```
SELECT ORDER_ID, SHIP_DATE
FROM ORDERS
WHERE SHIP_DATE IS NULL;
```

Ex. Finding products without any description.

4. LIKE Operator

- Used to search for patterns in column
- It works with two wildcards
 - % : Represents zero or more characters
 - _ : Represents Single character

Ex. Selecting employees whose name starts with 'S' ;

```
SELECT ENAME
FROM EMP
WHERE ENAME LIKE 'S%' ;
```

Ex. Retrieve name ending with 'M' .

```
SELECT ENAME
FROM EMP
WHERE ENAME LIKE '%M' ;
```

Ex. Names containing 'E' .

```
Select ename
From emp
Where ename like '%E%' ;
```

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

```
SELECT ENAME
```

```
FROM EMP
```

```
WHERE ENAME LIKE '__L%';
```

Ex. 5 character name:

```
SELECT ENAME
```

```
FROM EMP
```

```
WHERE ENAME LIKE '_____'; (5 underscores)
```

5. ORDER BY clause

- The order by clause sorts the resultset in ascending order or descending order.

SYNTAX:

```
SELECT column_name(s)
```

```
From table_name
```

```
ORDER BY column_name[ASC|DESC];
```

Ex. Sorting employees by salary in ascending order

```
SELECT ENAME, SAL
```

```
FROM EMP
```

```
ORDER BY SAL ASC;
```

Ex. Sorting employees by salary in DESCENDING order

```
SELECT ENAME, SAL  
FROM EMP  
ORDER BY SAL DESC;
```

Ex. Sorting enames alphabetically

```
SELECT ENAME, SAL  
FROM EMP  
ORDER BY ENAME ASC;
```

Ex. Sorting the products by name alphabetically

```
SELECT PRODUCT_NAME, PRICE  
FROM PRODUCTS  
ORDER BY PRODUCT_NAME ASC;
```

6. LIMIT clause

The LIMIT clause specifies the number of rows to return in the resultset.

SYNTAX:

```
SELECT column_name(s)  
From table_name  
LIMIT number_of_records;
```

Ex. Retrieving the top 5 highest paid employees.

```
SELECT ename, sal
FROM emp
ORDER BY sal DESC
LIMIT 5;
```

Ex. Selecting the first 10 students

```
SELECT STUDENT_NAME, GRADE
FROM STUDENTS
LIMIT 10;
```

Operator/Clause	Use Case	Features
BETWEEN.....AND	Filters records in a range	Includes start and end values
IN	Matches multiple values	Better instead of OR
IS NULL	TEST FOR MISSING VALUES	Checks for null
LIKE	Search for patterns	Uses % and _ (wildcards)
ORDER BY	SORT RESULTS	SORT IN ASC OR DESC ORDER
LIMIT	Limit the number of rows in resultset	Controls number of rows we can display

Arithmetic operators:

- Add a fixed amount(300) to each employees salary and display the result.

```
SELECT ENAME, SAL, SAL+300 AS NEW_SALARY
FROM EMP;
```

- Subtract a fixed amount(200) from each employees salary and display.

```
SELECT ENAME, SAL, SAL-200 AS UPDATED_SALARY
FROM EMP;
```

- Calculate a 10% bonus for each employee' s salary and display.

```
Select ename, sal, sal*0.10 AS BONUS
```

```
From emp;
```

- calculate the employee' s half salary

```
SELECT ENAME, SAL, SAL/2 AS HALF_SALARY from emp;
```

- Add 100 rupees to salary, subtract 50, multiply the result by 2 and divide it by 3.*
- Show employees whose updated salary(after adding 300) is greater than 2000.*