Operators in MySQL are used in various situations to perform operations on data — such as calculations, comparisons, filtering rows, pattern matching, and combining conditions.

1. **Arithmetic Operators:**

**Used when performing calculations.**

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| **+** | Addition | salary + bonus |
| **-** | Subtraction | price - discount |
| **\*** | Multiplication | quantity \* price |
| **/** | Division | total / count |
| **%** | Modulus (remainder) | 10 % 3 → 1 |

* **Example use:** Generating bills, salary, discounts, totals.

**SELECT price \* quantity AS total\_cost FROM products;**

### ****2. Comparison Operators****

**Used in WHERE, HAVING, JOIN conditions** to compare values.

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| **=** | Equal to | salary = 50000 |
| **!= or <>** | Not equal to | name != 'John' |
| **>** | Greater than | marks > 70 |
| **<** | Less than | age < 30 |
| **>=** | Greater than or equal | salary >= 60000 |
| **<=** | Less than or equal | marks <= 35 |
| **BETWEEN** | Within range | age BETWEEN 20 AND 30 |
| **LIKE** | Pattern match | name LIKE 'A%' |
| **IN** | Matches a list of values | dept IN ('HR', 'IT') |
| **IS NULL** | Checks for NULL | bonus IS NULL |

* **Example use:** Filtering records.

**SELECT \* FROM employees WHERE salary > 50000;**

### ****3. Logical Operators (****AND****,**** OR****,**** NOT****)****

**Used to combine multiple conditions** in WHERE, HAVING.

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| **AND** | All conditions true | age > 25 AND dept = 'HR' |
| **OR** | Any condition true | salary < 40000 OR dept = 'IT' |
| **NOT** | Negates condition | NOT (status = 'Active') |

* **Example use:** Filtering with complex logic.

**SELECT \* FROM students WHERE course = 'Math' AND marks > 80;**

### ****4. Pattern Matching (****LIKE****,**** IN****,**** BETWEEN****)****

**Used to search specific patterns or values.**

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| **LIKE** | Matches a string pattern using wildcards (%, \_) | WHERE name LIKE 'A%' → names starting with "A" |
| **NOT LIKE** | Opposite of LIKE, matches strings that don't fit the pattern | WHERE name NOT LIKE '%son' → names not ending in "son" |
| **IN** | Checks if a value is in a given list of values | WHERE department IN ('HR', 'IT', 'Finance') |
| **NOT IN** | Checks if a value is **not** in a list | WHERE role NOT IN ('Intern', 'Trainee') |
| **BETWEEN** | Checks if a value is **within a range** (inclusive) | WHERE salary BETWEEN 30000 AND 60000 |
| **NOT BETWEEN** | Checks if a value is **outside** a given range | WHERE marks NOT BETWEEN 35 AND 75 |

* **Example use:** Search or match operations.

**SELECT \* FROM customers WHERE name LIKE 'Ravi%';**

### ****5. NULL Checking (****IS NULL****,**** IS NOT NULL****)****

**Used to identify missing or undefined values.**

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| **IS NULL** | Checks if a column value is NULL (i.e., no value) | SELECT \* FROM employees WHERE bonus IS NULL; |
| **IS NOT NULL** | Checks if a column has a **non-NULL** (defined) value | SELECT \* FROM employees WHERE bonus IS NOT NULL; |

* **Example use:** Finding incomplete data.

**SELECT \* FROM orders WHERE delivery\_date IS NULL;**

| **Type** | **Used For** |
| --- | --- |

|  |  |
| --- | --- |
| **Arithmetic** | Math operations on columns |

|  |  |
| --- | --- |
| **Comparison** | Filtering rows based on conditions |

|  |  |
| --- | --- |
| **Logical** | Combining multiple conditions |

|  |  |
| --- | --- |
| **Pattern Match** | Searching values or patterns |

|  |  |
| --- | --- |
| **NULL Check** | Finding missing values |

**Where clause**

Used for selecting the rows based on condition. (Filtering the rows using where condition)

SELECT \* FROM EMPLOYEES;

SELECT \* FROM EMPLOYEES **WHERE** SALARY>3000;

SELECT \* FROM EMPLOYEES **WHERE** SALARY<=3000;

SELECT \* FROM EMPLOYEES **WHERE** DEPARTMENT\_ID=30;

SELECT \* FROM EMPLOYEES **WHERE** COMMISSION\_PCT is null;

SELECT \* FROM EMPLOYEES **WHERE** FIRST\_NAME='Jennifer';

SELECT **DISTINCT** DEPARTMENT\_ID FROM EMPLOYEES;

SELECT **DISTINCT** \* FROM EMPLOYEES;

**Between & IN Operators**

**Between 🡪** Used to display the rows which is following in the range of values.

**Not Between**

SELECT FROM EMPLOYEES **WHERE** SALARY **BETWEEN** 10000 **AND** 12000;

SELECT FROM EMPLOYEES **WHERE** SALARY **NOT BETWEEN** 10000 **AND** 12000;

**IN** 🡪 IN operator return the rows when the values are matching in the list

**Not In**

SELECT FROM EMPLOYEES **WHERE** SALARY=3400 **OR** SALARY=2500 **OR** SALARY=3000;

SELECT FROM EMPLOYEES **WHERE** SALARY **IN** (3400,2500,3000);

SELECT FROM EMPLOYEES **WHERE** SALARY **NOT IN** (3400,2500,3000);

**Logical Operators (AND, OR, NOT)**

SET SQL\_SAFE\_UPDATES = 0;

SELECT \* From EMPLOYEES;

SELECT \* FROM EMPLOYEES **WHERE** SALARY>15000 **AND** JOB\_ID='AD\_VP';

SELECT \* FROM EMPLOYEES **WHERE** SALARY>15000 **OR** JOB\_ID='AD\_VP';

SELECT \* FROM EMPLOYEES **WHERE** **NOT** FIRST\_NAME='David';

**Pattern Matching operators (wild card characters)**

**%** --> many characters

**\_** --> single character

SELECT FROM EMPLOYEES **WHERE** FIRST\_NAME LIKE 'S%';

SELECT \* FROM EMPLOYEES **WHERE** FIRST\_NAME LIKE '%r';

SELECT FROM EMPLOYEES **WHERE** FIRST\_NAME LIKE 'S%r';

SELECT \* FROM EMPLOYEES **WHERE** FIRST\_NAME LIKE '%m%';

SELECT FROM EMPLOYEES **WHERE** FIRST\_NAME NOT LIKE 'S%';

SELECT FROM EMPLOYEES **WHERE** FIRST\_NAME LIKE '%e\_';

SELECT \* FROM EMPLOYEES **WHERE** FIRST\_NAME LIKE '\_\_\_';