Excellent topic 👍 — “**Working with NULL values in SQL**” is an important concept for handling missing or unknown data. Let’s go through **each function/operator** you mentioned with **clear examples and output**.

**🧩 Table Used for Examples**

Let’s take a simple table called Employee:

| **EmpID** | **Name** | **Salary** | **Commission** |
| --- | --- | --- | --- |
| 1 | Raj | 50000 | 2000 |
| 2 | Priya | 60000 | NULL |
| 3 | Kiran | NULL | 1000 |
| 4 | Meena | 45000 | NULL |
| 5 | Arjun | NULL | NULL |

**1️⃣ IS NULL – checks if a column value is NULL**

SELECT Name FROM Employee

WHERE Commission IS NULL;

✅ **Output:**

Priya

Meena

Arjun

👉 Returns all employees where **Commission** value is NULL.

**2️⃣ IS NOT NULL – checks if a column value is NOT NULL**

SELECT Name FROM Employee

WHERE Salary IS NOT NULL;

✅ **Output:**

Raj

Priya

Meena

👉 Returns employees having **Salary** value (not missing).

**3️⃣ IFNULL(expr, value) – replaces NULL with a specified value (MySQL)**

SELECT Name, IFNULL(Salary, 0) AS Salary

FROM Employee;

✅ **Output:**

| **Name** | **Salary** |
| --- | --- |
| Raj | 50000 |
| Priya | 60000 |
| Kiran | 0 |
| Meena | 45000 |
| Arjun | 0 |

👉 Replaces NULL salaries with 0.

**4️⃣ NULLIF(expr1, expr2) – returns NULL if both values are equal, otherwise returns expr1**

SELECT NULLIF(Salary, 50000) AS Result

FROM Employee

WHERE Name='Raj';

✅ **Output:**

NULL

👉 Since Salary = 50000, and compared with 50000 → returns NULL.  
If they were different, it would return Salary value itself.

**5️⃣ COALESCE(expr1, expr2, expr3, ...) – returns the first non-NULL value from the list**

SELECT Name,

COALESCE(Salary, Commission, 0) AS Effective\_Income

FROM Employee;

✅ **Output:**

| **Name** | **Effective\_Income** |
| --- | --- |
| Raj | 50000 |
| Priya | 60000 |
| Kiran | 1000 |
| Meena | 45000 |
| Arjun | 0 |

👉 It picks the **first non-null** value among Salary → Commission → 0.

**6️⃣ COUNT(NULL) – behavior with NULLs**

SELECT COUNT(Commission) AS Total\_Commission

FROM Employee;

✅ **Output:**

2

👉 Only **non-NULL** values are counted.  
Even though 5 rows exist, only 2 have Commission values.

If you want to count all rows:

SELECT COUNT(\*) FROM Employee;

-- Output: 5

**7️⃣ SUM(NULL) – NULLs are ignored in aggregate functions**

SELECT SUM(Salary) AS TotalSalary

FROM Employee;

✅ **Output:**

155000

👉 It adds only non-null salaries (50000 + 60000 + 45000).

If all values were NULL → result = NULL.

**🔹 Summary Table**

| **Function / Operator** | **Purpose** | **Example Output** |
| --- | --- | --- |
| IS NULL | Checks for null values | Commission IS NULL → Priya, Meena, Arjun |
| IS NOT NULL | Checks for not null values | Salary IS NOT NULL → Raj, Priya, Meena |
| IFNULL(col, val) | Replace NULL with value | IFNULL(Salary,0) |
| NULLIF(a,b) | Returns NULL if a=b | NULLIF(50000,50000) → NULL |
| COALESCE(a,b,c) | First non-null value | COALESCE(Salary,Commission,0) |
| COUNT(col) | Ignores NULL | Only counts 2 commissions |
| SUM(col) | Ignores NULL | Sums only non-null salaries |