Nulls Decoded

LEARN

1. Understanding NULL in SQL

In SQL, NULL is a special marker used to indicate that a data value does not exist in the database. It is fundamentally different from other values such as 0 or empty strings (''), which represent actual values. Instead, NULL means "unknown," "missing," or "not applicable."

Key Properties of NULL:

- **Unknown Value:** NULL indicates that the value is unknown or not yet assigned.
- **Not Comparable:** NULL is not equal to anything—not even another NULL.
- **Three-Valued Logic:** Any operation involving NULL returns one of three results: TRUE, FALSE, or UNKNOWN.

Example:

```
SELECT *
FROM Employees
WHERE ManagerID IS NULL;
```

This query returns employees who do not report to any manager (i.e., their ManagerID is unknown).

---

2. Checking for NULL Values

Since NULL represents an unknown, standard comparison operators such as = or != do not work. Instead, SQL provides special operators:

Syntax:

```
column_name IS NULL
column_name IS NOT NULL
```

Examples:

```
-- Find customers with missing phone numbers
SELECT *
FROM Customers
WHERE PhoneNumber IS NULL;

-- Find customers who have provided phone numbers
SELECT *
FROM Customers
WHERE PhoneNumber IS NOT NULL;
```

> Reminder: WHERE PhoneNumber = NULL will always return no rows because it evaluates to UNKNOWN.

---

3. Handling NULL with COALESCE and IFNULL

To display or work with alternate values when NULL is encountered, we can use built-in functions:

COALESCE (ANSI SQL Standard):

Returns the first non-null value from the list of arguments.

```
SELECT COALESCE(Email, 'No Email Provided') AS ContactEmail

FROM Users;
```

If Email is NULL, 'No Email Provided' is shown instead.

IFNULL (MySQL-Specific):

```
SELECT IFNULL(Salary, 0) AS AdjustedSalary

FROM Employees;
```

Replaces NULL salaries with 0.

---

4. NULL in Aggregate Functions

Aggregate functions like SUM(), AVG(), MIN(), MAX(), and COUNT() behave differently in the presence of NULL values.

Important Notes:

- COUNT(column) ignores NULL values.
- COUNT(*) includes all rows.
- AVG(column) calculates the average of non-NULL values.

Example:

```
SELECT

 COUNT(Salary) AS SalariesProvided,

 COUNT(*) AS TotalEmployees,

 AVG(Salary) AS AverageSalary
```

```
FROM Employees;
```

---

5. Sorting NULLs with ORDER BY

The position of NULLs in a sorted result can vary depending on the database engine:

- In PostgreSQL: NULLS are last by default in ascending sort.
- In MySQL: NULLS are first in ascending order.

Control Sorting of NULLs:

```
SELECT Name, Address

FROM Contacts

ORDER BY Address IS NULL, Address;
```

This moves NULLs to the bottom by first sorting on IS NULL (FALSE before TRUE), and then sorting the actual values.

Alternate Syntax (PostgreSQL):

```
ORDER BY column_name ASC NULLS LAST
```

---

6. Logical Expressions Involving NULL

When NULL is involved in any logical expression, the result is generally UNKNOWN.

Example:

```
SELECT *

FROM Products
```

```
WHERE Price > NULL;  -- Always returns no rows
```

Because any comparison with NULL results in UNKNOWN.

Use CASE to Handle NULL Logic:

```
SELECT ProductName,
  CASE
    WHEN Description IS NULL THEN 'No Description'
    ELSE Description
  END AS FinalDescription
FROM Products;
```

PRACTISE

Task 1: Identify Rows with Missing Emails

```
SELECT *
FROM Customers
WHERE Email IS NULL;
```

Task 2: Replace NULL Feedback with a Placeholder

```
SELECT COALESCE(Feedback, 'No Feedback Given') AS UserFeedback
FROM Reviews;
```

Task 3: Count Null and Non-Null Values in a Column

```
SELECT
  COUNT(PhoneNumber) AS ProvidedPhoneNumbers,
  COUNT(*) AS TotalCustomers

FROM Customers;
```

Task 4: Display Last Names with Fallback Message

```
SELECT COALESCE(LastName, 'Last Name Missing') AS FinalLastName

FROM Employees;
```

Task 5: Order Null Descriptions at the Bottom

```
SELECT ProductID, Description

FROM Products

ORDER BY Description IS NULL, Description;
```

FAQ

- **Q:** Is NULL equal to NULL in SQL?
  - **A:** No. NULL is not equal to anything, even another NULL. Use `IS NULL` instead.
- **Q:** Why doesn't `WHERE column = NULL` work?
  - **A:** Because NULL is unknown. Use `IS NULL` or `IS NOT NULL` to test.
- **Q:** Do aggregate functions ignore NULL?
  - **A:** Yes. Except for `COUNT(*)`, which includes all rows.
- **Q:** Can I replace NULL in SELECT output?
  - **A:** Yes. Use `COALESCE()` or `IFNULL()` or `CASE` expressions.
- **Q:** Are NULLs sorted at the top or bottom?
  - **A:** It depends on the DBMS. Use `IS NULL` or database-specific options like `NULLS LAST` to control it.