difference between is and = in sql with different examples

In SQL, the operators **IS** and **=** are used for comparison but are applied in different contexts and serve distinct purposes. Let's explore the differences with examples:

**1. = Operator**

* Used for **value equality**.
* Compares two values to check if they are equal.
* Commonly used in WHERE, ON, or HAVING clauses.

**Examples:**

1. **Basic Equality Check:**
2. SELECT \*
3. FROM Employee
4. WHERE Department = 'HR';
   * Retrieves rows where the Department column is exactly 'HR'.
5. **Join Condition:**
6. SELECT e.name, d.name
7. FROM Employee e
8. JOIN Department d
9. ON e.departmentId = d.id;
   * Matches employees to their departments where departmentId in the Employee table equals id in the Department table.

**2. IS Operator**

* Used for **null comparisons**.
* Determines if a value is NULL or not.
* Cannot use = to compare NULL because NULL represents "unknown," and comparing NULL = NULL will always return FALSE.

**Examples:**

1. **Check for Null:**
2. SELECT \*
3. FROM Employee
4. WHERE Salary IS NULL;
   * Retrieves rows where the Salary column has a NULL value.
5. **Check for Not Null:**
6. SELECT \*
7. FROM Employee
8. WHERE Salary IS NOT NULL;
   * Retrieves rows where the Salary column is not NULL.
9. **Incorrect Use with =:**
10. SELECT \*
11. FROM Employee
12. WHERE Salary = NULL;
    * This will not return any results because NULL = NULL is not valid.

**Key Differences**

| **Feature** | **=** | **IS** |
| --- | --- | --- |
| **Purpose** | Compares equality of values. | Checks for NULL or NOT NULL. |
| **Use Case** | For non-null value comparisons. | For null checks. |
| **Behavior with NULL** | Returns no results when comparing with NULL. | Correctly identifies NULL values. |

**Combined Usage Example**

**Employee Table**

| **EmployeeId** | **Name** | **Department** | **Salary** |
| --- | --- | --- | --- |
| 1 | Alice | HR | 5000 |
| 2 | Bob | IT | NULL |
| 3 | Carol | HR | 7000 |

**Query 1 (Using =):**

SELECT \*

FROM Employee

WHERE Department = 'HR';

* Retrieves:
* EmployeeId | Name | Department | Salary
* -----------|-------|------------|-------
* 1 | Alice | HR | 5000
* 3 | Carol | HR | 7000

**Query 2 (Using IS):**

SELECT \*

FROM Employee

WHERE Salary IS NULL;

* Retrieves:
* EmployeeId | Name | Department | Salary
* -----------|------|------------|-------
* 2 | Bob | IT | NULL

**Incorrect Query with = for Null Check:**

SELECT \*

FROM Employee

WHERE Salary = NULL;

* Result:
  + No rows returned because NULL = NULL is invalid.

**Summary**

* Use **=** for comparing non-null values for equality.
* Use **IS** for checking NULL or NOT NULL values.

In SQL, **NULL** represents an **unknown value**. The key idea is that two unknown values cannot be definitively compared as equal, because SQL cannot assume that one unknown is the same as another unknown. As a result:

**Why NULL = NULL is FALSE**

1. **NULL Represents Unknown**:
   * When comparing two NULLs (NULL = NULL), you're essentially asking:
     + "Is an unknown value equal to another unknown value?"
   * Since SQL doesn't know what either NULL represents, it cannot say with certainty that they are equal.
2. **Three-Valued Logic**:
   * SQL uses **three-valued logic**: TRUE, FALSE, and UNKNOWN.
   * The result of NULL = NULL is **UNKNOWN**, not TRUE or FALSE.
   * In conditions like WHERE clauses, UNKNOWN is treated as FALSE, so no rows are returned.
3. **Consistency Across Operators**:
   * Other comparison operators like <, >, !=, or <> also yield UNKNOWN when applied to NULL. For example:
   * SELECT 1
   * WHERE NULL <> NULL;
     + This query will return no rows because NULL <> NULL evaluates to UNKNOWN.

**Example in Practice**

**Employee Table**

| **EmployeeId** | **Name** | **Salary** |
| --- | --- | --- |
| 1 | Alice | 5000 |
| 2 | Bob | NULL |
| 3 | Carol | NULL |

**Query Using =:**

SELECT \*

FROM Employee

WHERE Salary = NULL;

* Result: **No rows returned.**
* Explanation: Salary = NULL is evaluated as UNKNOWN for rows where Salary is NULL.

**Correct Way to Check for NULL**

To check if a value is NULL, you need to use the IS NULL operator:

**Query Using IS NULL:**

SELECT \*

FROM Employee

WHERE Salary IS NULL;

* Result:
* EmployeeId | Name | Salary
* -----------|-------|-------
* 2 | Bob | NULL
* 3 | Carol | NULL

**Additional Notes**

* **NULL = NULL is not TRUE because:**
  + SQL does not treat NULL as a specific value.
  + It represents "unknown," and unknown values cannot be confidently compared.
* **Practical Example of Misuse:**
* SELECT \*
* FROM Employee
* WHERE Salary = Salary;
  + Rows with NULL in the Salary column will not be returned because NULL = NULL is not TRUE.

**Three-Valued Logic Summary**

| **Expression** | **Result** |
| --- | --- |
| 5 = 5 | TRUE |
| 5 = NULL | UNKNOWN |
| NULL = NULL | UNKNOWN |
| NULL IS NULL | TRUE |

SQL requires the use of IS NULL and IS NOT NULL for handling null comparisons explicitly.