# Database SQL Language Reference

## Nulls

If a column in a row has no value, then the column is said to be **null**, or to contain null. Nulls can appear in columns of any data type that are not restricted by `NOT NULL` or `PRIMARY KEY` integrity constraints. Use a null when the actual value is not known or when a value would not be meaningful.

Oracle Database treats a character value with a length of zero as null. However, do not use null to represent a numeric value of zero, because they are not equivalent.

> **Note:**
>
> Oracle Database currently treats a character value with a length of zero as null. However, this may not continue to be true in future releases, and Oracle recommends that you do not treat empty strings the same as nulls.

Any arithmetic expression containing a null always evaluates to null. For example, null added to 10 is null. In fact, all operators (except concatenation) return null when given a null operand.

### Nulls in SQL Functions

For information on null handling in SQL functions, see "Nulls in SQL Functions" (functions001.htm#CJAEFDCE) .

### Nulls with Comparison Conditions

To test for nulls, use only the comparison conditions `IS NULL` and `IS NOT NULL`. If you use any other condition with nulls and the result depends on the value of the null, then the result is `UNKNOWN`. Because null represents a lack of data, a null cannot be equal or unequal to any value or to another null. However, Oracle considers two nulls to be equal when evaluating a `DECODE` function. Refer to DECODE (functions049.htm#i1017437) for syntax and additional information.

Oracle also considers two nulls to be equal if they appear in compound keys. That is, Oracle considers identical two compound keys containing nulls if all the non-null components of the keys are equal.

### Nulls in Conditions

A condition that evaluates to `UNKNOWN` acts almost like `FALSE`. For example, a `SELECT` statement with a condition in the `WHERE` clause that evaluates to `UNKNOWN` returns no rows. However, a condition evaluating to `UNKNOWN` differs from `FALSE` in that further operations on an `UNKNOWN` condition evaluation will evaluate to `UNKNOWN`. Thus, `NOT FALSE` evaluates to `TRUE`, but `NOT UNKNOWN` evaluates to `UNKNOWN`.

Table 3-20 (#g194888) shows examples of various evaluations involving nulls in conditions. If the conditions evaluating to `UNKNOWN` were used in a `WHERE` clause of a `SELECT` statement, then no rows would be returned for that query.

***Table 3-20 Conditions Containing Nulls***

| Condition | Value of A | Evaluation |
|---|---|---|
| `a IS NULL` | `10` | `FALSE` |
| `a IS NOT NULL` | `10` | `TRUE` |
| `a IS NULL` | `NULL` | `TRUE` |
| `a IS NOT NULL` | `NULL` | `FALSE` |
| `a = NULL` | `10` | `UNKNOWN` |
| `a != NULL` | `10` | `UNKNOWN` |
| `a = NULL` | `NULL` | `UNKNOWN` |
| `a != NULL` | `NULL` | `UNKNOWN` |
| `a = 10` | `NULL` | `UNKNOWN` |
| `a != 10` | `NULL` | `UNKNOWN` |

For the truth tables showing the results of logical conditions containing nulls, see Table 7-5 (conditions004.htm#g1068986) , Table 7-6 (conditions004.htm#g1068810) , and Table 7-7 (conditions004.htm#g1068834) .

✖ (https://docs.oracle.com/cd/E11882_01/server.112/e41084/sql_elements004.htm)

(#)

About Oracle (http://www.oracle.com/corporate/index.html) | Contact Us (http://www.oracle.com/us/corporate/contact/index.html) | Legal Notices (http://www.oracle.com/us/legal/index.html) | Terms of Use (http://www.oracle.com/us/legal/terms/index.html) | Your Privacy Rights (http://www.oracle.com/us/legal/privacy/index.html)
Copyright © 2016, Oracle and/or its affiliates. All rights reserved.