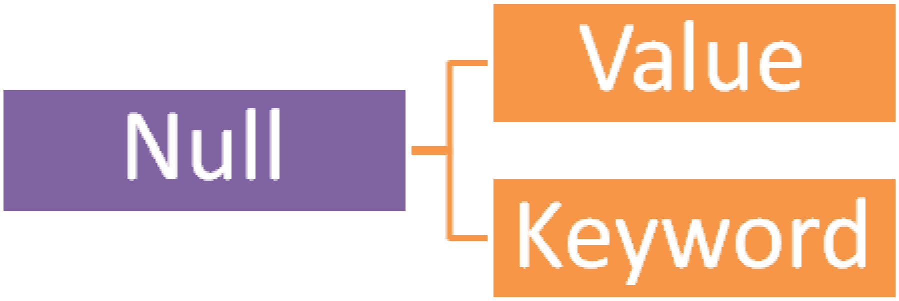
MySQL IS NULL & IS NOT NULL Tutorial with EXAMPLES

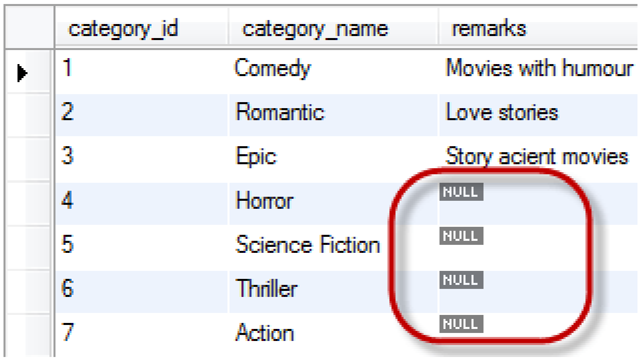
In SQL Null is both a value as well as a keyword. Let's look into NULL value first -



Null as a Value

In simple terms, NULL is simply a placeholder for data that does not exist. When performing insert operations on tables, they will be times when some field values will not be available.

In order to meet the requirements of true relational database management systems, MySQL uses NULL as the placeholder for the values that have not been submitted. The screenshot below shows how NULL values look in the database.



category_id	category_name	remarks
1	Comedy	Movies with humour
2	Romantic	Love stories
3	Epic	Story acient movies
4	Horror	NULL
5	Science Fiction	NULL
6	Thriller	NULL
7	Action	NULL

Let's now look at some of the basics for NULL before we go further into the discussion.

- **NULL is not a data type -** this means it is not recognized as an "int", "date" or any other defined data type.
- **Arithmetic operations** involving **NULL** always **return NULL** for example, 69 + NULL = NULL.
- All **aggregate functions affect only rows that do not have NULL values**.

Let's now demonstrate how the count function treats null values. Let's see the current contents of the members table-

```
SELECT * FROM `members`;
```

Executing the above script gives us the following results

membership_number	**full_names**	**gender**	**date_of_birth**	**physical_address**	**postal_address**	**contact_number**	**email**
1	Janet Jones	Female	21-07-1980	First Street Plot No 4	Private Bag	0759 253 542	janetjones@yagoo.cm
2	Janet Smith Jones	Female	23-06-1980	Melrose 123	NULL	NULL	jj@fstreet.com
3	Robert Phil	Male	12-07-1989	3rd Street 34	NULL	12345	rm@tstreet.com
4	Gloria Williams	Female	14-02-1984	2nd Street 23	NULL	NULL	NULL

5	Leonard Hofstadter	Male	NULL	Woodcrest	NULL	845738767	NULL
6	Sheldon Cooper	Male	NULL	Woodcrest	NULL	976736763	NULL
7	Rajesh Koothrappali	Male	NULL	Woodcrest	NULL	938867763	NULL
8	Leslie Winkle	Male	14-02-1984	Woodcrest	NULL	987636553	NULL
9	Howard Wolowitz	Male	24-08-1981	SouthPark	P.O. Box 4563	987786553	lwolowitz[at]email.me

Let's count all members who have updated their contact_number

```
SELECT COUNT(contact_number)  FROM `members`;
```

Executing the above query gives us the following results.

COUNT(contact_number)
7

Note: Values that are NULL have not been included

What is NOT?

The NOT logical operator is used to test for Boolean conditions and returns true if the condition is false. The NOT operator returns false if the condition been tested is true

Condition	NOT Operator Result
True	False
False	True

Why use NOT null?

There will be cases when we will have to perform computations on a query result set and return the values. Performing any arithmetic operations on columns that have the NULL value returns null results. In order to avoid such situations from happening, we can employ the use of the NOT NULL clause to limit the results on which our data operates.

NOT NULL Values

Let's suppose that we want to create a table with certain fields that should always be supplied with values when inserting new rows in a table. We can use the NOT NULL clause on a given field when creating the table.

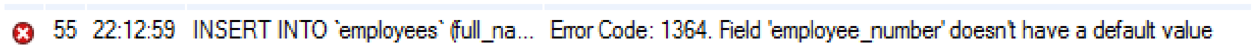
The example shown below creates a new table that contains employee's data. The employee number should always be supplied

```
CREATE TABLE `employees`(
  employee_number int NOT NULL,
  full_names varchar(255) ,
  gender varchar(6)
);
```

Let's now try to insert a new record without specifying the employee name and see what happens.

```
INSERT INTO `employees` (full_names,gender) VALUES
('Steve Jobs', 'Male');
```

Executing the above script in MySQL workbench gives the following error -



```
55 22:12:59 INSERT INTO `employees` (full_na... Error Code: 1364. Field 'employee_number' doesn't have a default value
```

NULL Keywords

NULL can also be used as a keyword when performing Boolean operations on values that include NULL. The "IS/NOT" keyword is used in conjunction with the NULL word for such purposes. The basic syntax when null is used as a keyword is as follows

```
`comlumn_name'  IS NULL
`comlumn_name' NOT NULL
```

HERE

- ***"IS NULL"*** is the keyword that performs the Boolean comparison. It returns true if the supplied value is NULL and false if the supplied value is not NULL.
- ***"NOT NULL"*** is the keyword that performs the Boolean comparison. It returns true if the supplied value is not NULL and false if the supplied value is null.

Let's now look at a practical example that uses the NOT NULL keyword to eliminate all the column values that have null values.

Continuing with the example above , suppose we need details of members whose contact number is not null . We can execute a query like

```
SELECT * FROM `members` WHERE contact_number IS NOT NULL;
```

Executing the above query gives only records where contact number is not null.

Suppose we want member records where contact number is null. We can use following query

```
SELECT * FROM `members` WHERE contact_number IS NULL;
```

Executing the above query gives member details whose contact number is NULL

membership_number	full_names	gender	date_of_birth	physical_address	postal_address	contact_number	email
1	Janet Jones	Female	21-07-1980	First Street Plot No 4	Private Bag	0759 253 542	janetjones@yagoo.cm
3	Robert Phil	Male	12-07-1989	3rd Street 34	NULL	12345	rm@tstreet.com
5	Leonard Hofstadter	Male	NULL	Woodcrest	NULL	845738767	NULL
6	Sheldon Cooper	Male	NULL	Woodcrest	NULL	976736763	NULL
7	Rajesh Koothrappali	Male	NULL	Woodcrest	NULL	938867763	NULL
8	Leslie Winkle	Male	14-02-1984	Woodcrest	NULL	987636553	NULL
9	Howard Wolowitz	Male	24-08-1981	SouthPark	P.O. Box 4563	987786553	lwolowitz[at]email.me

Comparing null values

Three-value logic - performing Boolean operations on conditions that involve NULL can either return **"Unknown", "True" or "False".**

For example, **using the "IS NULL" keyword** when doing comparison operations **involving NULL** can either return **true** or **false**. Using other comparison operators returns **"Unknown"(NULL).**

Suppose you compare number five with 5

```
SELECT 5 =5;
```

The query result is 1 which means TRUE

5 =5
1

Let's do the same operation with NULL

```
SELECT NULL = NULL;
```

NULL = NULL
NULL

Let's look at another example

```
SELECT 5 > 5;
```

5 > 5
0

The query result is 0 which means FALSE

Let's look at same example using NULL

```
SELECT NULL > NULL;
```

```
NULL > NULL
```

```
NULL
```

Let's use the IS NULL keyword

```
SELECT 5 IS NULL;
```

```
5 IS NULL
```

```
0
```

The query result is 0 which is FALSE

```
SELECT NULL IS NULL;
```

```
NULL IS NULL
```

```
1
```

The query result is 1 which is TRUE

Summary

- NULL is a value placeholder for optional table fields.
- MySQL treats the NULL value differently from other data types. The NULL value when used in a condition evaluates to the false Boolean value.
- The NOT logical operation is used to test for Boolean values and evaluates to true if the Boolean value is false and false if the Boolean value is true.
- The NOT NULL clause is used to eliminate NULL values from a result set

- Performing arithmetic operations on NULL values always returns NULL results.
- The comparison operators such as [, =, etc.] cannot be used to compare NULL values