**SQL Server NULL**.

Here are some interesting facts about NULL in SQL Server.

1. NULL can be defined as absence of value, undefined, or the value which is unknown at this point of time.
2. All data types can be defined with NULL constraint.
3. Direct usage of arithmetic or logical operations on NULL will not work as expected.
4. The system functions ISNULL, COALESE and NULLIF are used only to deal with NULL.
5. NOT NULL constraint cannot be defined for a computed column until it is also PERSISTED.
6. The only data types that will interpret NULL differently are rowversion and timestamp.

Run this code and see the output

declare @rv rowversion , @ts timestamp

select @rv=null, @ts=null

select @rv as rowversion,@ts as timestamp

Output is

rowversion timestamp

------------------ ------------------

0x 0x

1. NULL=NULL will never be true unless SET ANSI\_NULLS is OFF

While the first query returns nothing, the second will return 6

--Query 1

set ansi\_nulls on

select 6

where null=null

--Query 2

set ansi\_nulls off

select 6

where null=null

The condition WHERE col IS NULL will not be affected by the above setting

1. The default data type of NULL is INT. Refer [Default data type of NULL](http://beyondrelational.com/modules/2/blogs/70/posts/10982/default-datatype-of-null.aspx) for more information
2. Column with UNIQUE constraint will allow only one NULL value in SQL Server (But not true in other RDBMSs)
3. NULL will make SQL Server to use short circuit logic in some cases

Consider this example

select 6/0/null

select null/6/0

While the first query throws an error the second query returns NULL

1. The value NULL is not equal to string value 'NULL'.
2. By default NULL values come first when a column is ordered in ascending order and come last when ordered in descending order.
3. If you don’t use GROUP BY clause, the aggregate functions will always return single value (NULL) when the condition is false

select sum(col) as col from

(

select 45 as col

) as t

where 1=0

The above returns NULL

1. NULL values are by default omitted in all aggregate functions

Consider the following example

select sum(col) as col\_cum,count(col) as col\_count,avg(col\*1.0) as col\_avg from

(

select 1 as col union all

select null as col union all

select 2 as col union all

select 3 as col

) as t

The output is

col\_cum col\_count col\_avg

----------- ----------- -----------

6 3 2.000000

15. Aggregate functions cannot be directly applied over NULL value  
  
This code

select sum(null) as null\_sum

returns the following error

Msg 8117, Level 16, State 1, Line 1 .Operand data type NULL is invalid for sum operator.

Just CAST NULL AS INT and it will return the final result as NULL:

SELECT SUM(data)  
FROM (SELECT CAST(NULL AS INT) AS data) t