Reference > SQL data types reference > Logical

Logical data types

This topic describes the logical data types supported in Snowflake.

Data types

Snowflake supports a single logical data type (BOOLEAN).

BOOLEAN

BOOLEAN can have TRUE or FALSE values. BOOLEAN can also have an UNKNOWN value, which is represented by NULL. BOOLEAN columns can be used in expressions (for example, a SELECT list), as well as predicates (for example, a WHERE clause).

The BOOLEAN data type enables support for Ternary logic.

BOOLEAN conversion

Snowflake supports conversion to and from BOOLEAN.

Conversion to BOOLEAN

Non-BOOLEAN values can be converted to BOOLEAN values explicitly or implicitly.

Explicit conversion

You can explicitly convert specific text string and numeric values to BOOLEAN values by using the TO_BOOLEAN or CAST functions:

String conversion	• Strings converted to TRUE: `'true'`, `'t'`, `'yes'`, `'y'`, `'on'`, `'1'`. • Strings converted to FALSE: `'false'`, `'f'`, `'no'`, `'n'`, `'off'`, `'0'`. • Conversion is case-insensitive. • Other text strings can't be converted to BOOLEAN values.
Numeric conversion	• Zero (`0`) is converted to FALSE. • Any non-zero value is converted to TRUE.

Implicit conversion

Snowflake can implicitly convert specific text string and numeric values to BOOLEAN values:

String conversion   • `'true'` is converted to TRUE.

Numeric conversion
- Zero (`0`) is converted to FALSE.
- Any non-zero value is converted to TRUE.

Conversion from BOOLEAN

BOOLEAN values can be converted to non-BOOLEAN values explicitly or implicitly.

Explicit conversion

You can explicitly cast BOOLEAN values to text string or numeric values:

String conversion
- TRUE is converted to `'true'`.
- FALSE is converted to `'false'`.

Numeric conversion
- TRUE is converted to `1`.
- FALSE is converted to `0`.

Implicit conversion

String conversion
- TRUE is converted to `'true'`.
- FALSE is converted to `'false'`.

Examples

Create a table and insert values:

```sql
CREATE OR REPLACE TABLE test_boolean(
  b BOOLEAN,
  n NUMBER,
  s STRING);

INSERT INTO test_boolean VALUES
  (true, 1, 'yes'),
  (false, 0, 'no'),
  (NULL, NULL, NULL);

SELECT * FROM test_boolean;
```

```
+-------+------+------+
| B     |    N | S    |
|-------+------+------|
| True  |    1 | yes  |
| False |    0 | no   |
```

The following query includes a BOOLEAN-typed expression:

```sql
SELECT b, n, NOT b AND (n < 1) FROM test_boolean;
```

```
+-------+------+-------------------+
| B     |    N | NOT B AND (N < 1) |
|-------+------+-------------------|
| True  |    1 | False             |
| False |    0 | True              |
| NULL  | NULL | NULL              |
+-------+------+-------------------+
```

The following example uses a BOOLEAN column in predicates:

```sql
SELECT * FROM test_boolean WHERE NOT b AND (n < 1);
```

```
+-------+---+----+
| B     | N | S  |
|-------+---+----|
| False | 0 | no |
+-------+---+----+
```

The following example casts a text value to a BOOLEAN value. The example uses the SYSTEM$TYPEOF to show the type

```
SELECT s,
       TO_BOOLEAN(s),
       SYSTEM$TYPEOF(TO_BOOLEAN(s))
  FROM test_boolean;
```

```
+------+--------------+-----------------------------+
| S    | TO_BOOLEAN(S) | SYSTEM$TYPEOF(TO_BOOLEAN(S)) |
|------+--------------+-----------------------------|
| yes  | True         | BOOLEAN[SB1]                |
| no   | False        | BOOLEAN[SB1]                |
| NULL | NULL         | BOOLEAN[SB1]                |
+------+--------------+-----------------------------+
```

The following example casts a number value to a BOOLEAN value:

```
SELECT n,
       TO_BOOLEAN(n),
       SYSTEM$TYPEOF(TO_BOOLEAN(n))
  FROM test_boolean;
```

```
+------+--------------+-----------------------------+
| N    | TO_BOOLEAN(N) | SYSTEM$TYPEOF(TO_BOOLEAN(N)) |
|------+--------------+-----------------------------|
| 1    | True         | BOOLEAN[SB1]                |
| 0    | False        | BOOLEAN[SB1]                |
```

In this example, Snowflake implicitly converts a BOOLEAN value to a text value:

```
SELECT 'Text for ' || s || ' is ' || b AS result,
       SYSTEM$TYPEOF('Text for ' || s || ' is ' || b) AS type_of_result
  FROM test_boolean;
```

```
+----------------------+------------------------+
| RESULT               | TYPE_OF_RESULT         |
|----------------------+------------------------|
| Text for yes is true | VARCHAR(16777216)[LOB] |
| Text for no is false | VARCHAR(16777216)[LOB] |
| NULL                 | VARCHAR(16777216)[LOB] |
+----------------------+------------------------+
```