

BOOLEAN data type

Firebird 3.0 BOOLEAN

Firebird 3 introduced a true BOOLEAN type (True/False/Unknown), complete with support for logical predicates, e.g.,

```
UPDATE ATABLE
SET MYBOOL = (COLUMN1 IS DISTINCT FROM COLUMN2)
```

The following is an excerpt from the [The Firebird 3.0 Release Notes \(27 January 2014 - Document v.0300-08 - for Firebird 3.0 Alpha 2\)](#):

Adriano dos Santos Fernandes

The SQL-2008 compliant BOOLEAN data type (8 bits) comprises the distinct truth values TRUE and FALSE. Unless prohibited by a NOT NULL constraint, the BOOLEAN data type also supports the truth value UNKNOWN as the null value. The specification does not make a distinction between the NULL value of this data type and the truth value UNKNOWN that is the result of an SQL predicate, search condition, or boolean value expression: they may be used interchangeably to mean exactly the same thing.

As with many programming languages, the SQL BOOLEAN values can be tested with implicit truth values. For example, field1 OR field2 and NOT field1 are valid expressions.

The IS operator

Predications use the operator `IS [NOT]` for matching. For example, `field1 IS FALSE`, or `field1 IS NOT TRUE`.

Note: Equivalence operators (`"=`", `"!=",` `"<>"` and so on) are valid in all comparisons.

Examples

```
CREATE TABLE TB00L (ID INT, BVAL BOOLEAN);
COMMIT;

INSERT INTO TB00L VALUES (1, TRUE);
INSERT INTO TB00L VALUES (2, 2 = 4);
INSERT INTO TB00L VALUES (3, NULL = 1);
COMMIT;

SELECT * FROM TB00L
      ID    BVAL
=====
```

```
1 <true>
2 <false>
3 <null>

-- Test for TRUE value
SELECT * FROM TBOOL WHERE BVAL
      ID    BVAL
=====
1 <true>

-- Test for FALSE value
SELECT * FROM TBOOL WHERE BVAL IS FALSE
      ID    BVAL
=====
2 <false>

-- Test for UNKNOWN value
SELECT * FROM TBOOL WHERE BVAL IS UNKNOWN
      ID    BVAL
=====
3 <null>

-- Boolean values in SELECT list
SELECT ID, BVAL, BVAL AND ID < 2
FROM TBOOL
      ID    BVAL
=====
1 <true>  <true>
2 <false> <false>
3 <null>  <false>

-- PSQL Declaration with start value
DECLARE VARIABLE VAR1 BOOLEAN = TRUE;

-- Valid syntax, but as with a comparison
-- with NULL, will never return any record
SELECT * FROM TBOOL WHERE BVAL = UNKNOWN
SELECT * FROM TBOOL WHERE BVAL <> UNKNOWN
```

Notes:

- Represented in the API with the `FB_BOOLEAN` type and `FB_TRUE` and `FB_FALSE` constants.
- The value `TRUE` is greater than the value `FALSE`.
- Although `BOOLEAN` is not implicitly convertible to any other data type, it can be explicitly converted to and from string with `CAST`.
- For compatibility reasons, the non-reserved keywords `INSERTING`, `UPDATING` and `DELETING` continue to behave as Boolean expressions when used in context in `PSQL`, while behaving as values if they are column or variable names in non-Boolean expressions.

The following example uses the word `INSERTING` in all three ways:

```
SELECT
  INSERTING, -- value
  NOT INSERTING -- keyword
FROM TEST
WHERE
  INSERTING -- keyword
  AND INSERTING IS TRUE -- value
```

Source: The Firebird 3.0 Release Notes by Helen Borrie (Collator/Editor): 27 January 2014 - Document v.0300-08 - for Firebird 3.0 Alpha 2.

Workaround in pre-Firebird 3.0 versions

Firebird/InterBase® does not offer a native BOOLEAN data type in pre-Firebird 3.0 versions. In pre-Firebird 3.0 versions they can be implemented using [domains](#).

The first step is to define a domain (which should logically be named BOOLEAN). The domain can be defined in one of two ways:

1. Using a SMALLINT (16 bits), defaulting to zero, with a check constraint to ensure only the values of zero or one are entered. i.e:

```
CREATE DOMAIN D_BOOLEAN AS SMALLINT DEFAULT 0
CHECK (VALUE BETWEEN 0 AND 1);
```

Once you have defined this domain you can forever use it as a BOOLEAN data type without further concern. It is particularly suitable from a Delphi point of view, as Pascal BOOLEANs work in a similar manner.

2. Alternatively, the domain can be defined as a CHAR(1) and appropriate single character values ensured using a check constraint. If T and F or Y and N are more meaningful for your application then use this approach.

We'd like to thank Paul Beach of IBPhoenix for this article about Boolean data types.

InterBase 2009 BOOLEAN data type

Please refer to the respective [InterBase® documentation](#) for InterBase-specific keywords.

From:
<http://ibexpert.com/docu/> - IBExpert

Permanent link:
<http://ibexpert.com/docu/doku.php?id=01-documentation:01-13-miscellaneous:glossary:boolean-data-type>

Last update: 2023/08/14 08:27

