

## 11.6 Data Type Default Values

Data type specifications can have explicit or implicit default values.

A `DEFAULT value` clause in a data type specification explicitly indicates a default value for a column.

Examples:

```
CREATE TABLE t1 (  
  i      INT DEFAULT -1,  
  c      VARCHAR(10) DEFAULT '',  
  price DOUBLE(16,2) DEFAULT 0.00  
);
```

`SERIAL DEFAULT VALUE` is a special case. In the definition of an integer column, it is an alias for `NOT NULL AUTO_INCREMENT UNIQUE`.

Some aspects of explicit `DEFAULT` clause handling are version dependent, as described following.

- Explicit Default Handling as of MySQL 8.0.13
- Explicit Default Handling Prior to MySQL 8.0.13
- Implicit Default Handling

### Explicit Default Handling as of MySQL 8.0.13

The default value specified in a `DEFAULT` clause can be a literal constant or an expression. With one exception, enclose expression default values within parentheses to distinguish them from literal constant default values. Examples:

```
CREATE TABLE t1 (  
  -- literal defaults  
  i INT          DEFAULT 0,  
  c VARCHAR(10) DEFAULT '',  
  -- expression defaults  
  f FLOAT        DEFAULT (RAND() * RAND()),  
  b BINARY(16)   DEFAULT (UUID_TO_BIN(UUID())),  
  d DATE         DEFAULT (CURRENT_DATE + INTERVAL 1 YEAR),  
  p POINT        DEFAULT (Point(0,0)),
```

```
j JSON          DEFAULT (JSON_ARRAY())
);
```

The exception is that, for TIMESTAMP and DATETIME columns, you can specify the CURRENT\_TIMESTAMP function as the default, without enclosing parentheses. See Section 11.2.5, “Automatic Initialization and Updating for TIMESTAMP and DATETIME”.

The BLOB, TEXT, GEOMETRY, and JSON data types can be assigned a default value only if the value is written as an expression, even if the expression value is a literal:

- This is permitted (literal default specified as expression):

```
CREATE TABLE t2 (b BLOB DEFAULT ('abc'));
```

- This produces an error (literal default not specified as expression):

```
CREATE TABLE t2 (b BLOB DEFAULT 'abc');
```

Expression default values must adhere to the following rules. An error occurs if an expression contains disallowed constructs.

- Literals, built-in functions (both deterministic and nondeterministic), and operators are permitted.
- Subqueries, parameters, variables, stored functions, and loadable functions are not permitted.
- An expression default value cannot depend on a column that has the AUTO\_INCREMENT attribute.
- An expression default value for one column can refer to other table columns, with the exception that references to generated columns or columns with expression default values must be to columns that occur earlier in the table definition. That is, expression default values cannot contain forward references to generated columns or columns with expression default values.

The ordering constraint also applies to the use of ALTER TABLE to reorder table columns. If the resulting table would have an expression default value that contains a forward reference to a generated column or column with an expression default value, the statement fails.

## Note

If any component of an expression default value depends on the SQL mode, different results may occur for different uses of the table unless the SQL mode is the same

during all uses.

For CREATE TABLE ... LIKE and CREATE TABLE ... SELECT, the destination table preserves expression default values from the original table.

If an expression default value refers to a nondeterministic function, any statement that causes the expression to be evaluated is unsafe for statement-based replication. This includes statements such as INSERT and UPDATE. In this situation, if binary logging is disabled, the statement is executed as normal. If binary logging is enabled and binlog format is set to `STATEMENT`, the statement is logged and executed but a warning message is written to the error log, because replication slaves might diverge. When binlog format is set to `MIXED` or `ROW`, the statement is executed as normal.

When inserting a new row, the default value for a column with an expression default can be inserted either by omitting the column name or by specifying the column as `DEFAULT` (just as for columns with literal defaults):

```
mysql> CREATE TABLE t4 (uid BINARY(16) DEFAULT (UUID_TO_BIN(UUID())));
mysql> INSERT INTO t4 () VALUES();
mysql> INSERT INTO t4 () VALUES(DEFAULT);
mysql> SELECT BIN_TO_UUID(uid) AS uid FROM t4;
+-----+
| uid                                     |
+-----+
| f1109174-94c9-11e8-971d-3bf1095aa633 |
| f110cf9a-94c9-11e8-971d-3bf1095aa633 |
+-----+
```

However, the use of DEFAULT(*col name*) to specify the default value for a named column is permitted only for columns that have a literal default value, not for columns that have an expression default value.

Not all storage engines permit expression default values. For those that do not, an ER\_UNSUPPORTED\_ACTION\_ON\_DEFAULT\_VAL\_GENERATED error occurs.

If a default value evaluates to a data type that differs from the declared column type, implicit coercion to the declared type occurs according to the usual MySQL type-conversion rules. See Section 12.3, “Type Conversion in Expression Evaluation”.

## Explicit Default Handling Prior to MySQL 8.0.13

With one exception, the default value specified in a `DEFAULT` clause must be a literal constant; it cannot be a function or an expression. This means, for example, that you cannot set the default for a date column to be the value of a function such as `NOW()` or `CURRENT_DATE`. The exception is that, for `TIMESTAMP` and `DATETIME` columns, you can specify `CURRENT_TIMESTAMP` as the default. See Section 11.2.5, “Automatic Initialization and Updating for `TIMESTAMP` and `DATETIME`”.

The `BLOB`, `TEXT`, `GEOMETRY`, and `JSON` data types cannot be assigned a default value.

If a default value evaluates to a data type that differs from the declared column type, implicit coercion to the declared type occurs according to the usual MySQL type-conversion rules. See Section 12.3, “Type Conversion in Expression Evaluation”.

## Implicit Default Handling

If a data type specification includes no explicit `DEFAULT` value, MySQL determines the default value as follows:

If the column can take `NULL` as a value, the column is defined with an explicit `DEFAULT NULL` clause.

If the column cannot take `NULL` as a value, MySQL defines the column with no explicit `DEFAULT` clause.

For data entry into a `NOT NULL` column that has no explicit `DEFAULT` clause, if an `INSERT` or `REPLACE` statement includes no value for the column, or an `UPDATE` statement sets the column to `NULL`, MySQL handles the column according to the SQL mode in effect at the time:

- If strict SQL mode is enabled, an error occurs for transactional tables and the statement is rolled back. For nontransactional tables, an error occurs, but if this happens for the second or subsequent row of a multiple-row statement, the preceding rows are inserted.
- If strict mode is not enabled, MySQL sets the column to the implicit default value for the column data type.

Suppose that a table `t` is defined as follows:

```
CREATE TABLE t (i INT NOT NULL);
```

In this case, `i` has no explicit default, so in strict mode each of the following statements produce an error and no row is inserted. When not using strict mode, only the third statement produces an error; the implicit default is inserted for the first two statements, but the third fails because `DEFAULT(i)` cannot produce a value:

```
INSERT INTO t VALUES();  
INSERT INTO t VALUES(DEFAULT);  
INSERT INTO t VALUES(DEFAULT(i));
```

See Section 5.1.11, “Server SQL Modes”.

For a given table, the SHOW CREATE TABLE statement displays which columns have an explicit `DEFAULT` clause.

Implicit defaults are defined as follows:

- For numeric types, the default is 0, with the exception that for integer or floating-point types declared with the `AUTO_INCREMENT` attribute, the default is the next value in the sequence.
- For date and time types other than TIMESTAMP, the default is the appropriate “zero” value for the type. This is also true for TIMESTAMP if the explicit defaults for timestamp system variable is enabled (see Section 5.1.8, “Server System Variables”). Otherwise, for the first TIMESTAMP column in a table, the default value is the current date and time. See Section 11.2, “Date and Time Data Types”.
- For string types other than ENUM, the default value is the empty string. For ENUM, the default is the first enumeration value.