- **SQL standard explicit table clause and table value constructor.** Added table value constructors and explicit table clauses according to the SQL standard. These are implemented in MySQL 8.0.19, respectively, as the `TABLE` statement and the `VALUES` statement.

  The `TABLE` statement has the format `TABLE` *`table_name`*, and is equivalent to `SELECT * FROM` *`table_name`*. It supports `ORDER BY` and `LIMIT` clauses (the latter with optional `OFFSET`), but does not allow for the selection of individual table columns. `TABLE` can be used anywhere that you would employ the equivalent `SELECT` statement; this includes joins, unions, `INSERT ... SELECT`, `REPLACE`, `CREATE TABLE ... SELECT` statements, and subqueries. For example:

  - `TABLE t1 UNION TABLE t2` is equivalent to `SELECT * FROM t1 UNION SELECT * FROM t2`

  - `CREATE TABLE t2 TABLE t1` is equivalent to `CREATE TABLE t2 SELECT * FROM t1`

  - `SELECT a FROM t1 WHERE b > ANY (TABLE t2)` is equivalent to `SELECT a FROM t1 WHERE b > ANY (SELECT * FROM t2)`.

  `VALUES` can be used to supply a table value to an `INSERT`, `REPLACE`, or `SELECT` statement, and consists of the `VALUES` keyword followed by a series of row constructors (`ROW()`) separated by commas. For example, the statement `INSERT INTO t1 VALUES ROW(1,2,3), ROW(4,5,6), ROW(7,8,9)` provides an SQL-compliant equivalent to the MySQL-specific `INSERT INTO t1 VALUES (1,2,3), (4,5,6), (7,8,9)`. You can also select from a `VALUES` table value constructor just as you would a table, bearing in mind that you must supply a table alias when doing so, and use this `SELECT` just as you would any other; this includes joins, unions, and subqueries.

  For more information about `TABLE` and `VALUES`, and for examples of their use, see the following sections of this documentation:

  - Section 13.2.12, “TABLE Statement”

  - Section 13.2.14, “VALUES Statement”

  - Section 13.1.20.4, “CREATE TABLE ... SELECT Statement”

  - Section 13.2.6.1, “INSERT ... SELECT Statement”

  - Section 13.2.10.2, “JOIN Clause”

  - Section 13.2.11, “Subqueries”

  - Section 13.2.10.3, “UNION Clause”