- Beginning with MySQL 8.0.21, a single-table `UPDATE` or `DELETE` statement can now in many cases make use of a semijoin transformation or subquery materialization. This applies to statements of the forms shown here:

  - `UPDATE t1 SET t1.a=value WHERE t1.a IN (SELECT t2.a FROM t2)`
  - `DELETE FROM t1 WHERE t1.a IN (SELECT t2.a FROM t2)`

  This can be done for a single-table `UPDATE` or `DELETE` meeting the following conditions:

  - The `UPDATE` or `DELETE` statement uses a subquery having a `[NOT] IN` or `[NOT] EXISTS` predicate.
  - The statement has no `ORDER BY` clause, and has no `LIMIT` clause.

    (The multi-table versions of `UPDATE` and `DELETE` do not support `ORDER BY` or `LIMIT`.)
  - The target table does not support read-before-write removal (relevant only for `NDB` tables).
  - Semijoin or subquery materialization is allowed, based on any hints contained in the subquery and the value of `optimizer_switch`.

  When the semijoin optimization is used for an eligible single-table `DELETE` or `UPDATE`, this is visible in the optimizer trace: for a multi-table statement there is a `join_optimization` object in the trace, while there is none for a single-table statement. The conversion is also visible in the output of `EXPLAIN FORMAT=TREE` or `EXPLAIN ANALYZE`; a single-table statement shows `<not executable by iterator executor>`, while a multi-table statement reports a full plan.

  Alo beginning with MySQL 8.0.21, semi-consistent reads are supported by multi-table `UPDATE` statements using `InnoDB` tables, for transaction isolation levels weaker than `REPEATABLE READ`.

- **Common table expressions.** MySQL now supports common table expressions, both nonrecursive and recursive. Common table expressions enable use of named temporary result sets, implemented by permitting a `WITH` clause preceding `SELECT` statements and certain other statements. For more information, see Section 13.2.15, “WITH (Common Table Expressions)”.

  As of MySQL 8.0.19, the recursive `SELECT` part of a recursive common table expression (CTE) supports a `LIMIT` clause. `LIMIT` with `OFFSET` is also supported. See Recursive Common Table Expressions, for more information.

- **Window functions.** MySQL now supports window functions that, for each row from a query, perform a calculation using rows related to that row. These include functions such as `RANK()`, `LAG()`, and `NTILE()`. In addition, several existing aggregate functions now can be used as window functions (for example, `SUM()` and `AVG()`). For more information, see Section 12.21, “Window Functions”.

- **Lateral derived tables.** A derived table now may be preceded by the `LATERAL` keyword to specify that it is permitted to refer to (depend on) columns of preceding tables in the same `FROM` clause. Lateral derived tables make possible certain SQL operations that cannot be done with nonlateral derived tables or that require less-efficient workarounds. See Section 13.2.11.9, “Lateral Derived Tables”.

- **Aliases in single-table DELETE statements.** In MySQL 8.0.16 and later, single-table `DELETE` statements support the use of table aliases.

- **Regular expression support.** Previously, MySQL used the Henry Spencer regular expression library to support regular expression operators (`REGEXP`, `RLIKE`). Regular expression support has been reimplemented using International Components for Unicode (ICU), which provides full Unicode support