`FOR UPDATE` and `FOR SHARE` support `NOWAIT`, `SKIP LOCKED`, and `OF tbl_name` options. See Section 13.2.10, “SELECT Statement”.

`OF tbl_name` applies locking queries to named tables.

- `ADD PARTITION`, `DROP PARTITION`, `COALESCE PARTITION`, `REORGANIZE PARTITION`, and `REBUILD PARTITION` `ALTER TABLE` options are supported by native partitioning in-place APIs and may be used with `ALGORITHM={COPY|INPLACE}` and `LOCK` clauses.

  `DROP PARTITION` with `ALGORITHM=INPLACE` deletes data stored in the partition and drops the partition. However, `DROP PARTITION` with `ALGORITHM=COPY` or `old_alter_table=ON` rebuilds the partitioned table and attempts to move data from the dropped partition to another partition with a compatible `PARTITION ... VALUES` definition. Data that cannot be moved to another partition is deleted.

- The `InnoDB` storage engine now uses the MySQL data dictionary rather than its own storage engine-specific data dictionary. For information about the data dictionary, see Chapter 14, *MySQL Data Dictionary*.

- `mysql` system tables and data dictionary tables are now created in a single `InnoDB` tablespace file named `mysql.ibd` in the MySQL data directory. Previously, these tables were created in individual `InnoDB` tablespace files in the `mysql` database directory.

- The following undo tablespace changes are introduced in MySQL 8.0:

  - By default, undo logs now reside in two undo tablespaces that are created when the MySQL instance is initialized. Undo logs are no longer created in the system tablespace.

  - As of MySQL 8.0.14, additional undo tablespaces can be created in a chosen location at runtime using `CREATE UNDO TABLESPACE` syntax.

    ```
    CREATE UNDO TABLESPACE tablespace_name ADD DATAFILE 'file_name.ibu';
    ```

    Undo tablespaces created using `CREATE UNDO TABLESPACE` syntax can be dropped at runtime using `DROP UNDO TABLESPACE` syntax.

    ```
    DROP UNDO TABLESPACE tablespace_name;
    ```

    `ALTER UNDO TABLESPACE` syntax can be used to mark an undo tablespace as active or inactive.

    ```
    ALTER UNDO TABLESPACE tablespace_name SET {ACTIVE|INACTIVE};
    ```

    A `STATE` column that shows the state of a tablespace was added to the `INFORMATION_SCHEMA.INNODB_TABLESPACES` table. An undo tablespace must be in an `empty` state before it can be dropped.

  - The `innodb_undo_log_truncate` variable is enabled by default.

  - The `innodb_rollback_segments` variable defines the number of rollback segments per undo tablespace. Previously, `innodb_rollback_segments` specified the total number of rollback segments for the MySQL instance. This change increases the number of rollback segments available for concurrent transactions. More rollback segments increases the likelihood that