modify a few bytes. This enhancement builds upon support added in MySQL 8.0.4 for partial update of LOB data.

- As of MySQL 8.0.12, `ALGORITHM=INSTANT` is supported for the following `ALTER TABLE` operations:

  - Adding a column. This feature is also referred to as “Instant `ADD COLUMN`”. Limitations apply. See Section 15.12.1, “Online DDL Operations”.
  - Adding or dropping a virtual column.
  - Adding or dropping a column default value.
  - Modifying the definition of an `ENUM` or `SET` column.
  - Changing the index type.
  - Renaming a table.

  Operations that support `ALGORITHM=INSTANT` only modify metadata in the data dictionary. No metadata locks are taken on the table, and table data is unaffected, making the operations instantaneous. If not specified explicitly, `ALGORITHM=INSTANT` is used by default by operations that support it. If `ALGORITHM=INSTANT` is specified but not supported, the operation fails immediately with an error.

  For more information about operations that support `ALGORITHM=INSTANT`, see Section 15.12.1, “Online DDL Operations”.

- As of MySQL 8.0.13, the `TempTable` storage engine supports storage of binary large object (BLOB) type columns. This enhancement improves performance for queries that use temporary tables containing BLOB data. Previously, temporary tables that contained BLOB data were stored in the on-disk storage engine defined by `internal_tmp_disk_storage_engine`. For more information, see Section 8.4.4, “Internal Temporary Table Use in MySQL”.

- As of MySQL 8.0.13, the `InnoDB` data-at-rest encryption feature supports general tablespaces. Previously, only file-per-table tablespaces could be encrypted. To support encryption of general tablespaces, `CREATE TABLESPACE` and `ALTER TABLESPACE` syntax was extended to include an `ENCRYPTION` clause.

  The `INFORMATION_SCHEMA.INNODB_TABLESPACES` table now includes an `ENCRYPTION` column that indicates whether or not a tablespace is encrypted.

  The `stage/innodb/alter tablespace (encryption)` Performance Schema stage instrument was added to permit monitoring of general tablespace encryption operations.

- Disabling the `innodb_buffer_pool_in_core_file` variable reduces the size of core files by excluding `InnoDB` buffer pool pages. To use this variable, the `core_file` variable must be enabled and the operating system must support the `MADV_DONTDUMP` non-POSIX extension to `madvise()`, which is supported in Linux 3.4 and later. For more information, see Section 15.8.3.7, “Excluding Buffer Pool Pages from Core Files”.

- As of MySQL 8.0.13, user-created temporary tables and internal temporary tables created by the optimizer are stored in session temporary tablespaces that are allocated to a session from a pool of temporary tablespaces. When a session disconnects, its temporary tablespaces are truncated and released back to the pool. In previous releases, temporary tables were created in the global temporary