- If a storage engine does not support an attempted `ALTER TABLE` operation, a warning may result. Such warnings can be displayed with `SHOW WARNINGS`. See Section 13.7.7.42, “SHOW WARNINGS Statement”. For information on troubleshooting `ALTER TABLE`, see Section B.3.6.1, “Problems with ALTER TABLE”.
- For information about generated columns, see Section 13.1.9.2, “ALTER TABLE and Generated Columns”.
- For usage examples, see Section 13.1.9.3, “ALTER TABLE Examples”.
- `InnoDB` in MySQL 8.0.17 and later supports addition of multi-valued indexes on JSON columns using a *`key_part`* specification can take the form `(CAST json_path AS type ARRAY)`. See Multi-Valued Indexes, for detailed information regarding multi-valued index creation and usage of, as well as restrictions and limitations on multi-valued indexes.
- With the `mysql_info()` C API function, you can find out how many rows were copied by `ALTER TABLE`. See mysql_info().

There are several additional aspects to the `ALTER TABLE` statement, described under the following topics in this section:

- Table Options
- Performance and Space Requirements
- Concurrency Control
- Adding and Dropping Columns
- Renaming, Redefining, and Reordering Columns
- Primary Keys and Indexes
- Foreign Keys and Other Constraints
- Changing the Character Set
- Importing InnoDB Tables
- Row Order for MyISAM Tables
- Partitioning Options

Table Options

`table_options` signifies table options of the kind that can be used in the `CREATE TABLE` statement, such as `ENGINE`, `AUTO_INCREMENT`, `AVG_ROW_LENGTH`, `MAX_ROWS`, `ROW_FORMAT`, or `TABLESPACE`.

For descriptions of all table options, see Section 13.1.20, “CREATE TABLE Statement”. However, `ALTER TABLE` ignores `DATA DIRECTORY` and `INDEX DIRECTORY` when given as table options. `ALTER TABLE` permits them only as partitioning options, and requires that you have the `FILE` privilege.

Use of table options with `ALTER TABLE` provides a convenient way of altering single table characteristics. For example:

- If `t1` is currently not an `InnoDB` table, this statement changes its storage engine to `InnoDB`:

```
ALTER TABLE t1 ENGINE = InnoDB;
```

  - See Section 15.6.1.5, “Converting Tables from MyISAM to InnoDB” for considerations when switching tables to the `InnoDB` storage engine.