MySQL Storage Engine Overview

The storage engine is a dedicated software module in the MySQL RDBMS that is meant to ensure all essential SQL operations performed correctly. It cares about the primary tasks of creating, reading, updating, and deleting (CRUD), and the overall database performance. The good thing is that MySQL has predefined storage engines that are developed, adjusted, and optimized to meet requirements. On the other hand, there is always a catch.

Frequently, database developers just accept the default engine and proceed. But this choice is more essential than one might consider. The wrong storage engine may lead to many negative consequences, especially when your database grows with time. Even the default, most favored engines aren't perfect under all circumstances (though they serve well for the majority of standard cases).

MySQL supports the following storage engines that developers can use for their specific purposes:

- **InnoDB** is the most widely used and ACID-based storage engine set as default in MySQL versions 8.0 or higher. The main difference is that it supports foreign key referential integrity constraints.
- **MyISAM** can handle non-transactional tables and support table-level locking and full-text search indexes. It is mainly used on the Web.
- **Federated** can create a single, local database by connecting physical MySQL servers. It stores data only on the remote server.
- **MEMORY** can create tables and store data in memory for faster performance and data access. It supports table-level locking and non-transactional tables and can be used for creating temporary tables.
- **MERGE** can logically group a set of similar MyISAM tables into one table. The storage engine can manipulate large volumes of data.
- **EXAMPLE** is used to teach developers how to create new storage engines.
- **Archive** is used to store large volumes of unindexed data.
- **CSV** is a flexible storage engine that stores data in CSV files and can be integrated into multiple applications that support a CSV format in order to import and export data.
- **Blackhole** accepts but does not store data, and returns empty data during data retrieval.

To view which of them are available for your server, use the following command:

```
mysql> SHOW ENGINES\G
```

In the output, you will get a list of available and unavailable storage engines. It also will define which of them are already set as default.

```
      Engine: MyISAM
     Support: YES
     Comment: MyISAM storage engine
Transactions: NO
          XA: NO
  Savepoints: NO
*************************** 6. row ***************************
      Engine: PERFORMANCE_SCHEMA
     Support: YES
     Comment: Performance Schema
Transactions: NO
          XA: NO
  Savepoints: NO
*************************** 7. row ***************************
      Engine: InnoDB
     Support: DEFAULT
     Comment: Supports transactions, row-level locking, and foreign keys
Transactions: YES
          XA: YES
  Savepoints: YES
```

The next step is to define which one suits you best. In most cases, one storage engine is fine for all your goals. But sometimes applications use several engines. The choice depends on your preferences and work conditions.

To change the engine

*ALTER TABLE **my_table** ENGINE = InnoDB;*

Most Popular MySQL Storage Engines

Speaking of MySQL, we deal with two storage engines most frequently. These are MyISAM and InnoDB (you might have noticed them both in the illustration above). Let's have a closer look at them and see how they can be used to serve your goals best.

MyISAM

MyISAM was the MySQL default storage engine prior to version 5.5.1, and it still remains a popular choice due to its simplicity and speed. Most specialists consider it to be the best option for beginners who only start mastering MySQL. Unfortunately, one of the primary factors ensuring simplicity is the absence of foreign keys support. As a result, your work won't involve complicated configurations, but your options are limited.

Also, this engine requires less disk space and thus is suitable in cases of limited disk space. It provides extreme speed with the SELECT and INSERT statements, which is a valuable advantage, but it can be slow when dealing with the DELETE and UPDATE statements, as it does not support transactions with rollbacks and table-level locking.

If you consider MyISAM for your project, note that it applies best to the data warehousing applications or web apps that won't use transactions.

Though MyISAM is still used by many applications, it was not surprising that it stopped being the default one and was replaced by InnoDB.

InnoDB

If you work on applications based on MySQL now, InnoDB will most likely be your storage engine. It ensures all options that a database would require, and is the most popular choice for the absolute majority of developers.

InnoDB supports transactions and foreign keys constraints. Thus, it can check the INSERT, UPDATE, and DELETE statements' consistency much better. It is less speedy than MyISAM, but it is less vulnerable to crushes either. Besides, it offers such an essential advantage as row-level locking, ensuring a more efficient multi-user performance.

In addition, in the case with multiple storage engines configured, it would be better to disable the unused storage engines to improve server performance. For example, you have ten configured storage engines but use only one of them. In this case, nine storage engines stand idle and only consume server resources with bringing no benefit.

Therefore, MyISAM and InnoDB are the most widely used storage engines for MySQL.

However, this RDBMS supports more than a dozen of them. It's worth paying attention to one more engine that is called Federated.

Federated

Although not default, Federated is a well-known storage engine for MySQL. Its peculiarity is that it allows accessing the data from a remote MySQL database. At the same time, it does not need replication or cluster technologies. The key is the local Federated table. When a query addresses that table, it applies automatically to the remote federated table. The data aren't stored locally.

This technique is excellent when developers need to work with several databases in MySQL and remote MySQL servers. Remote servers store the data, while the local server only points to them with the connection string. The efficiency of this action does not depend on the storage engine type on the remote server.

However, Federated has a dangerous drawback. It is far from being the best solution when dealing with joined tables, as its speed of work decreases tremendously in that case. Besides, its treatment of transactions is rather weird. The engine has a narrow specialization and should be addressed with care.

MySQL Storage Engines Feature Comparison

The table displays the differences between the features of MyISAM, InnoDB, Memory, and Archive storage engines. Yes means that the feature is supported and No – not supported.

Feature	MyISAM	InnoDB	Memory	Archive
Storage limits	256 TB	64 TB	RAM	None
B-tree indexes	Yes	Yes	Yes	No
Backup / point-in-time recovery	Yes	Yes	Yes	Yes
Cluster database support	No	No	No	No
Clustered indexes	No	Yes	No	No
Compressed data	Yes	Yes	No	Yes
Data caches	No	Yes	N/A	No

Encrypted data	Yes	Yes	Yes	Yes
Foreign key support	No	Yes	No	No
Full-text search indexes	Yes	Yes	No	No
Geospatial data type support	Yes	Yes	No	Yes
Geospatial indexing support	Yes	Yes	No	No
Hash indexes	No	No	Yes	No
Index caches	Yes	Yes	N/A	No
Locking granularity	Table	Row	Table	Row
MVCC (Multiversion concurrency control)	No	Yes	No	No
Replication support	Yes	Yes	Limited	Yes
Spatial data types and functions	Yes	Yes	No	Yes
Spatial and non-spatial indexes (for spatial columns)	Yes	Yes	Yes	No

T-tree indexes	No	No	No	No
Transactions	No	Yes	No	No
Update statistics for data dictionary	Yes	Yes	Yes	Yes

To create table

CREATE TABLE t1 (a INT, b CHAR (20), PRIMARY KEY (a)) ENGINE=InnoDB;