



MySQL Storage engine and Types



What is Storage Engine...

- A storage engine is a software module that a database management system uses to create, read, update data from a database.
- There are two types of storage engines in MySQL: transactional and non-transactional.
- When a database is created, one often overlooked but critical factor in performance is the storage engine (particularly as the database grows). In many instances, the temptation is to just accept the default and continue on developing your project.
- This can lead to unexpected negative impacts on performance, backups, and data integrity later in the application life cycle, such as when your team implements analytics and MySQL dashboards.
- To avoid these potential pitfalls, we are going to take a closer look at some of the most widely used storage engines supported by MySQL



InnoDB Storage engine

- The default option in MySQL 5.7, InnoDB is a robust storage engine that offers:
 - Full ACID compliance
 - Commit, rollback, and crash-recovery
 - Row-level locking
 - FOREIGN KEY referential-integrity constraints
 - Increase multi-user concurrency (via non-locking reads)
- It is an engine that performs well and offers many of the required attributes that any database would need.
- Oracle recommends using InnoDB for tables except for specialized use cases.



Memory Storage engine

- Creates tables in memory.
- It is the fastest engine.
- It provides table-level locking.
- It does not support transactions.
- Memory storage engine is ideal for creating temporary tables or quick lookups.
- The data is lost when the database is restarted.



CSV Storage engine

- A useful storage engine when data needs to be shared with other applications that use CSV formatted data.
- The tables are stored as comma separated value text files.
- Though this makes sharing the data with scripts and applications easier, one drawback is that the CSV files are not indexed.
- So, the data should be stored in an InnoDB table until the Import/Export stage of the process.



Blackhole Storage engine

- The *Blackhole* storage engine accepts but does not store data.
- Retrievals always return an empty set.
- The functionality can be used in distributed database design where data is automatically replicated, but not stored locally.
- This storage engine can be used to perform performance tests or other testing.



Federated Storage engine

- This storage engine is for creating a single, local, logical database by linking several different physical MySQL servers.
- No data is stored on the local server and
- Queries are automatically executed on the respective remote server.
- It is perfect for distributed data mart environments and can vastly improve performance when using MySQL for analytical reporting.



MyISAM Storage engine

- The functionality that sets MyISAM apart is its capability for:
 - full text search indexes
 - table-level locking
 - lack of support for transactions
- Though it is a fast storage engine, it is best suited for use in read-heavy and mostly read applications such as data warehousing and web applications that don't need transaction support or ACID compliance.



Choosing the right Storage Engine

- No storage engine is ideal for all circumstances. Some perform best under certain conditions and perform worse in other situations.
- There are tradeoffs than must be considered.
- A more secure solution takes more resources; it might be slower, take more CPU time, and disk space.
- MySQL is very flexible in the fact that it provides several different storage engines. Some of them, like the Archive engine, are created to be used in specific situations.
- In some cases the answer is clear. Whenever we are dealing with some payment systems, we are obliged to use the most secure solution. We cannot afford to loose such sensitive data. InnoDB is the way to go.
- If we want full-text search, then we can choose either MyISAM or InnoDB.
- Only InnoDB supports foreign key referential integrity constraint and if we plan to use this constraint, then the choice is clear.



Specifying and altering storage engines

- No storage engine is ideal for all circumstances. Some perform best under certain conditions and perform worse in other situations.
- There are tradeoffs than must be considered.
- A more secure solution takes more resources; it might be slower, take more CPU time, and disk space.
- MySQL is very flexible in the fact that it provides several different storage engines. Some of them, like the Archive engine, are created to be used in specific situations.
- In some cases the answer is clear. Whenever we are dealing with some payment systems, we are obliged to use the most secure solution. We cannot afford to loose such sensitive data. InnoDB is the way to go.
- If we want full-text search, then we can choose either MyISAM or InnoDB.
- Only InnoDB supports foreign key referential integrity constraint and if we plan to use this constraint, then the choice is clear.