

Chapter 26: MySQL Storage Engines

26.1 Storage Engine Concept

- * A storage engine is a component that handles how data is **stored**, **retrieved** and **managed** in MySQL.
- * MySQL is a **plugin based** database, which means it supports multiple storage engines.
- * Each storage engine has its **own features**, performance characteristics and use cases.
- * When you **create a table**, you can choose the storage engine for that table.



Key Points :

- ✓ Storage engines determine how data is stored and managed.
- ✓ Different engines offer different features and performance.
- ✓ Choosing the right engine is important for your application.
- ✓ You can specify the engine while creating a table.



26.2 InnoDB

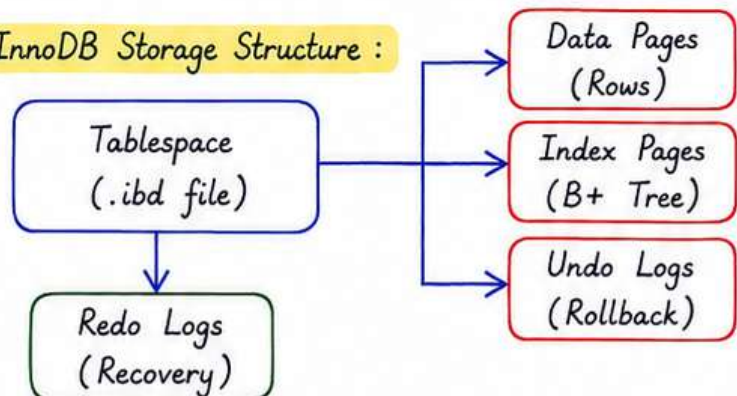
- * InnoDB is the **default** storage engine in MySQL.
- * It is a **transaction-safe** (ACID compliant) engine.
- * It supports **row-level locking** for high concurrency.
- * It supports **foreign keys** and referential integrity.
- * It uses a **clustered index** structure for data storage.
- * It provides **crash recovery** and support for commit, rollback.
- * InnoDB is suitable for applications with **high reliability** and **concurrent transactions**.



Features of InnoDB :

- ACID compliant
- Row-level locking
- Foreign key support
- Crash recovery
- MVCC (Multi-Version Concurrency Control)
- High performance and reliability

InnoDB Storage Structure :



Key Takeaway :



Storage engines define how MySQL stores and manages data. InnoDB is a **powerful**, **reliable** and **ACID compliant** engine best suited for modern applications.



26.3 MyISAM

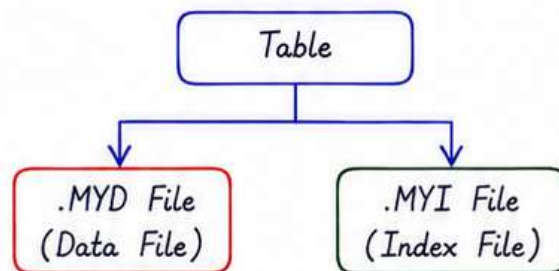
- * MyISAM is a **non-transactional** storage engine.
- * It was the **default engine** in older versions of MySQL.
- * It is **faster for read operations** and suitable for applications with mostly **SELECT** queries.
- * It supports **full-text indexing**.
- * It **does not support foreign keys**.
- * It is used in applications where **data integrity is not** a critical requirement.



Features of MyISAM :

- High speed for read-heavy workloads
- Full-text search support
- Table-level locking
- No support for transactions
- Uses less memory
- Stores data in .MYD file and index in .MYI file

MyISAM Storage Structure :



Key Takeaway :



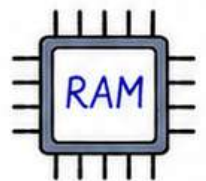
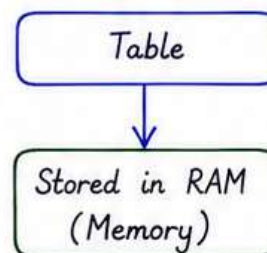
MyISAM is fast and efficient for **read-mostly applications** but lacks **transaction support** and **row-level locking**.



26.4 MEMORY

- * MEMORY is also known as **HEAP**.
- * It stores all data in **RAM** (memory).
- * It is **extremely fast** because it does not use disk storage.
- * It is best suited for **temporary tables** and data that changes frequently.
- * Data is **lost** when the server is restarted.

MEMORY Storage Structure :



Features of MEMORY :

- Very high performance
- Data stored in RAM
- Supports hash indexes
- Table-level locking
- Data is temporary (lost on restart)
- Suitable for temporary tables and small datasets

Best Use Cases :

- ✓ Temporary tables
- ✓ Session data storage
- ✓ Caching small lookup data
- ✓ High-speed data processing

Key Takeaway :

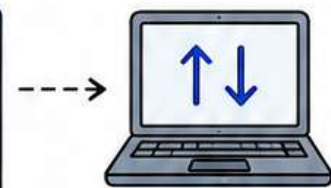
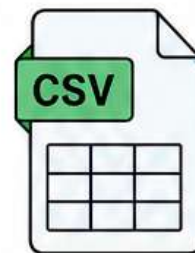


MEMORY engine offers **excellent performance** by storing data in RAM, but it is **non-persistent** and not suitable for large or permanent data.



26.5 CSV

- * CSV stands for **Comma Separated Values**.
- * It is a simple storage engine that stores data in **CSV files** (plain text files).
- * It is used for **data exchange** and **portability**.
- * It does **not** support indexing.
- * It is **not suitable** for **large datasets** and complex applications.
- * It is mainly used to **import** and **export** data.

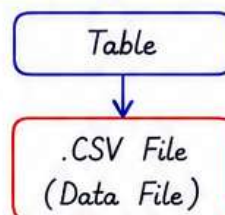


Data Import / Export

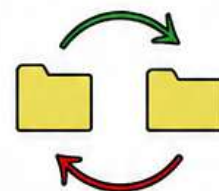
Features of CSV :

- Stores data in plain text (CSV files)
- Easy to read and edit
- Good for data import and export
- No indexing support
- No transactions support
- Best for small and simple datasets

CSV Storage Structure :



id, name, age, city
1, John, 25, Hyderabad
2, Anu, 22, Delhi
3, Ravi, 30, Mumbai
4, Sita, 27, Bangalore



Key Takeaway :

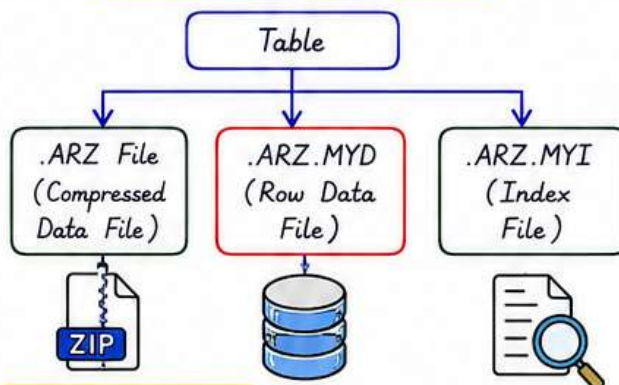


CSV engine is useful for **data exchange** and **portability** but lacks features like indexing, transactions and performance.

26.6 ARCHIVE

- * ARCHIVE is a storage engine that stores rows in **compressed format**.
- * It is used for **data archiving** and storage.
- * It supports indexing but only for a **limited number** of columns.
- * It does **not** support transactions.
- * It provides **high compression**, so it saves disk space.
- * It is suitable for **long-term data storage**.

ARCHIVE Storage Structure :



Features of ARCHIVE :

- High compression ratio
- Saves disk space
- Supports indexing (limited columns)
- No transaction support
- Data is read-only (insert and select allowed)
- Ideal for archiving and logging data



Best Use Cases :

- ✓ Archiving historical data
- ✓ Logging and audit data storage
- ✓ Data warehousing
- ✓ Storing large volume of infrequently accessed data



Key Takeaway :



ARCHIVE engine is best for **long-term, infrequently accessed data** because it provides **high compression** and saves storage space.



26.7 Storage Engine Selection

- * Choosing the right storage engine depends on your **application requirements**.
- * Each engine has its own **strengths** and **limitations**.
- * Consider factors like **performance**, **reliability**, **data integrity** and use case.
- * You can specify the storage engine while **creating a table**.



Storage Engine Comparison :

Storage Engine	Transaction Support	Locking	Speed (Reads)	Speed (Writes)	Data Integrity	Best Use Case
 InnoDB	Yes	Row-level	Good	Good	High	High reliability applications
 MyISAM	No	Table-level	Very Fast	Fast	Medium	Read-intensive applications
 MEMORY	No	Table-level	Very Fast	Very Fast	Low	Temporary tables, caching
 CSV	No	N/A	Slow	Slow	Low	Import/Export, Data exchange
 ARCHIVE	No	N/A	Slow	Slow	Low	Archiving, Long-term storage

Key Takeaway :



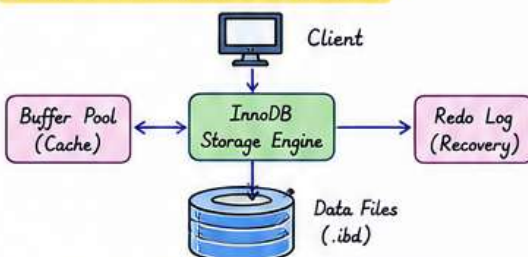
Use **InnoDB** for transactional and high reliability applications.
 Use **MyISAM** for read-heavy workloads.
 Use **MEMORY** for speed and temporary data.
 Use **CSV** for data exchange and **ARCHIVE** for long-term storage.



26.8 InnoDB Features

- * InnoDB is the **most advanced** storage engine in MySQL.
- * It is the **default** engine in MySQL 5.5+ versions.
- * It provides **high performance**, **reliability** and **rich feature set**.
- * It is widely used in modern web applications and enterprise systems.

InnoDB Architecture Overview :



Key Features of InnoDB :

- ✓ **ACID compliant** - Ensures reliable transactions. 
- ✓ **Row-level locking** - Improves concurrency. 
- ✓ **Foreign key support** - Maintains referential integrity. 
- ✓ **Crash recovery** - Recovers data after system failure. 
- ✓ **MVCC (Multi-Version Concurrency Control)** - Improves performance and consistency. 
- ✓ **Clustered index** - Stores data with primary key for faster access. 
- ✓ **Support for large datasets** - Handles large tables efficiently. 
- ✓ **High reliability** - Suitable for mission-critical applications. 

Key Takeaway :



InnoDB is a **powerful** engine with advanced features like **ACID compliance**, **row-level locking**, **crash recovery** and **foreign key support**, making it ideal for modern applications.



26.9 InnoDB Tablespaces

- * InnoDB stores data in **tablespaces**.
- * A tablespace is a **set of data files**.
- * It contains **tables, indexes** and other internal data.
- * InnoDB supports **two types** of tablespaces:

Types of Tablespaces :

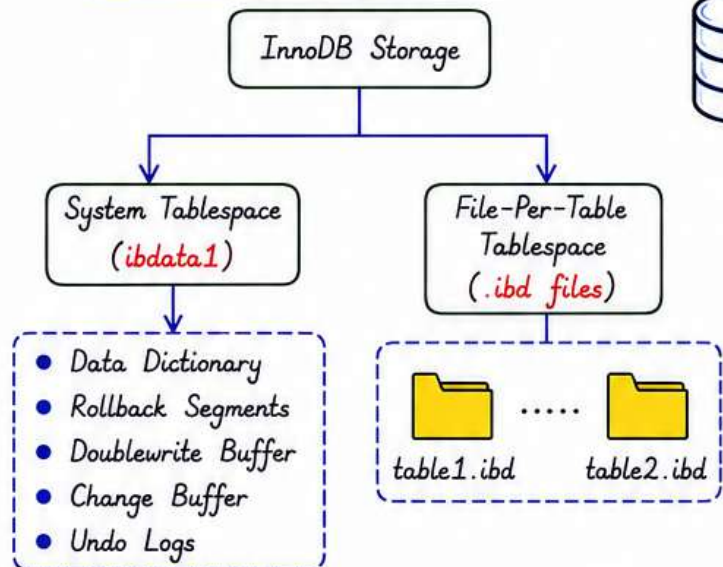
1. System Tablespace

- Default tablespace (ibdata1).
- Stores data dictionary, rollback segments and other internal data.
- Created automatically when MySQL starts.

2. File-Per-Table Tablespace

- Each table has its own tablespace.
- Stored in .ibd file.
- Better space management and performance.

InnoDB Tablespace Architecture :



Key Benefits :

- ✓ Better organization of data.
- ✓ Easy backup and recovery.
- ✓ Improved performance.
- ✓ Efficient space management.

Key Takeaway :



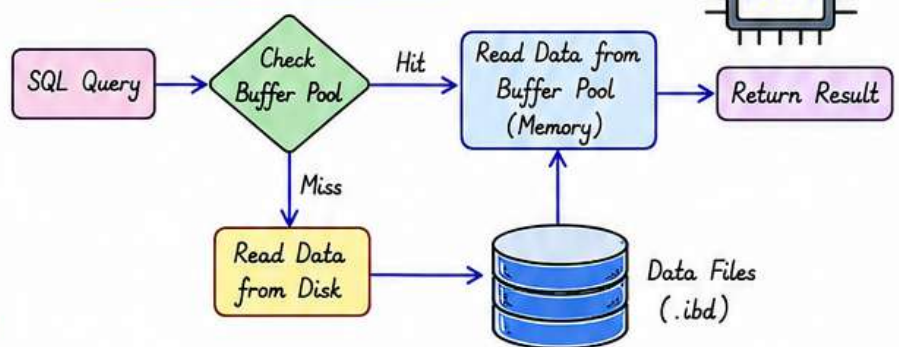
Tablespaces in InnoDB help in **organizing data efficiently** and improving performance. The file-per-table tablespace provides **better flexibility**.



26.10 InnoDB Buffer Pool

- * The **buffer pool** is the memory area used to **cache data and indexes**.
- * It **reduces disk I/O** by storing frequently used data in memory.
- * When data is requested, InnoDB **first looks in the buffer pool**.
- * If not found, it reads from disk and stores it in the buffer pool.
- * The buffer pool **improves performance** by reducing disk access.

Buffer Pool Architecture :



Buffer Pool Components :

- **Data Pages** - Store table and index data.
- **Index Pages** - Store index information.
- **Free Pages** - Used for modifications.
- **Dirty Pages** - Modified pages not yet written to disk.

Benefits of Buffer Pool :

- ✓ Reduces disk I/O.
- ✓ Improves query performance.
- ✓ Caches frequently accessed data.
- ✓ Supports larger working sets in memory.



Key Takeaway :



The InnoDB Buffer Pool **caches data and indexes** in memory, reducing disk I/O and **improving performance**. Proper buffer pool size is crucial for optimal results.

