

MySQL Overview & Architecture

1.1 Introduction to MySQL & Ecosystem

What is MySQL?

MySQL is an open-source, relational database management system (RDBMS) that uses Structured Query Language (SQL) for database access and management. It's widely used in web applications, data warehouses, and enterprise environments.

Key Characteristics:

- **Open Source:** Free to download and use, with source code available
- **Reliability:** ACID-compliant with InnoDB engine
- **Performance:** Optimized for read-heavy workloads
- **Scalability:** Supports large databases and high concurrent users
- **Security:** User authentication and privilege system
- **Platform Independence:** Runs on Linux, Windows, macOS, Unix

MySQL Ecosystem Components

Core Components:

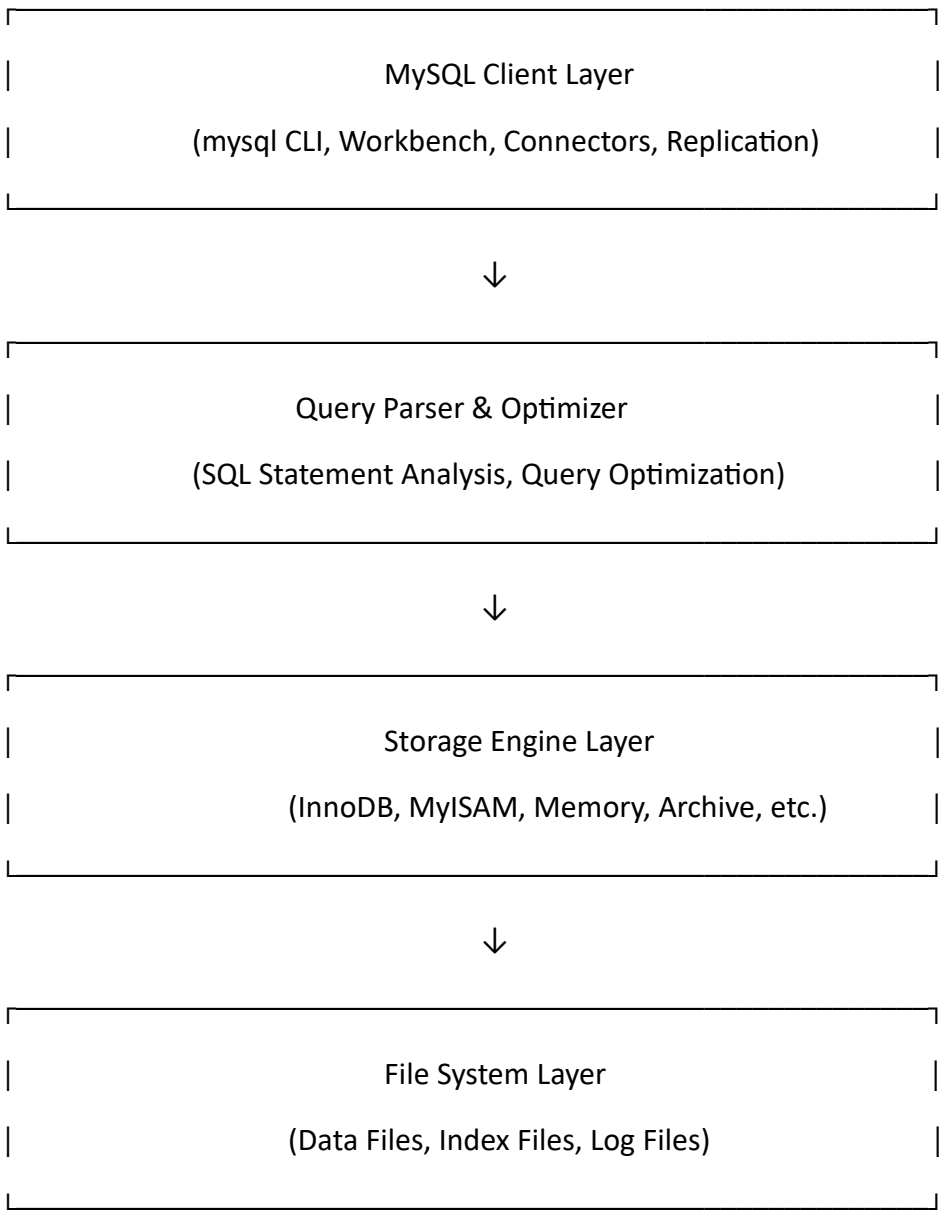
1. **MySQL Server:** The database engine that stores and retrieves data
2. **MySQL Client:** Command-line interface to interact with the server
3. **Storage Engines:** Pluggable backend that handles data storage
4. **Replication:** Master-slave data synchronization
5. **MySQL Shell:** Advanced command-line interface (MySQL 8.0+)
6. **MySQL Workbench:** GUI tool for database development and administration

Related Tools:

- **Percona XtraBackup:** Fast, non-blocking backup tool
- **Percona Toolkit:** Advanced MySQL utility scripts
- **MySQL Router:** Connection routing and load balancing
- **InnoDB Cluster:** MySQL high-availability solution

1.2 MySQL Architecture & Components

High-Level Architecture



Key Architectural Components

1. Connection Layer

- Handles client connections using TCP/IP or Unix sockets
- Manages user authentication and SSL connections
- Tracks connection state and resources

2. SQL Layer

- Parses SQL statements
- Checks syntax and privileges
- Optimizes query execution plans
- Manages transactions and locks

3. Storage Engine Layer

- Executes queries against storage engine
- Manages indexes, caching, and data retrieval
- Handles transaction logging and recovery
- Pluggable architecture allows multiple engines

4. File System Layer

- Stores database files on disk
- Manages InnoDB tablespaces
- Handles binary logs and relay logs
- Manages undo and redo logs

MySQL Data Dictionary

In MySQL 8.0+, the data dictionary is:

- Stored in InnoDB tables in mysql schema
- Centralized metadata repository
- Replaces previous file-based storage
- Improves reliability and consistency

Key Data Dictionary Tables:

- mysql.tables: Table definitions
- mysql.columns: Column information
- mysql.indexes: Index metadata
- mysql tablespaces: Tablespace definitions

1.3 Storage Engines Overview

What is a Storage Engine?

A storage engine is the component of MySQL that handles the storage and retrieval of data in the database. MySQL uses a pluggable architecture, allowing multiple storage engines.

InnoDB Engine

Overview:

- Default storage engine in MySQL 5.7+
- ACID-compliant with transaction support
- Implements row-level locking
- Crash recovery capabilities

Key Features:

- **ACID Properties:** Atomicity, Consistency, Isolation, Durability
- **Foreign Keys:** Referential integrity support
- **Transactions:** BEGIN, COMMIT, ROLLBACK support
- **Crash Recovery:** Automatic recovery after crashes
- **Row-level Locking:** Better concurrency than table locks

When to Use:

- Production environments
- Applications requiring transactions
- Data integrity is critical
- Concurrent access is high

MyISAM Engine

Overview:

- Older default engine (pre-5.7)
- Fast but lacks transaction support
- Table-level locking

Key Features:

- **Fast I/O:** Optimized for read operations

- **Table-level Locking:** Entire table locked during writes
- **Full-text Search:** Built-in full-text indexing
- **No Transactions:** No transaction support

When to Use:

- Read-only tables
- Data warehousing scenarios
- Full-text search requirements
- Legacy applications

Memory Engine

Overview:

- Stores data in RAM
- Extremely fast access
- Data lost on server restart

Key Features:

- **Speed:** Data stored in RAM
- **Fixed-length Records:** More efficient memory usage
- **Hash Indexes:** Default index type
- **No Persistence:** Data not written to disk

When to Use:

- Session storage
- Temporary tables
- Caching layers
- Real-time analytics

Archive Engine

Overview:

- Optimized for compressed, historical data
- High compression ratio
- Very slow INSERT/UPDATE operations

Key Features:

- **Compression:** Automatic compression to save space
- **Read-only Queries:** Optimized for SELECT
- **Slow Modifications:** INSERT/UPDATE very slow
- **Non-indexed:** Full table scans required

When to Use:

- Historical data archival
- Log storage
- Data backup and compliance
- Space-constrained environments

NDBCluster Engine

Overview:

- Distributed, in-memory storage engine
- Used in MySQL Cluster
- Supports geographic redundancy

Key Features:

- **Distributed:** Data partitioned across nodes
- **In-Memory:** Primary storage in RAM
- **Synchronous Replication:** Real-time data consistency
- **High Availability:** Automatic failover

When to Use:

- Distributed systems requiring HA

- Real-time applications with minimal downtime
- Geographic redundancy needed

Storage Engine Comparison

Feature	InnoDB	MyISAM	Memory	Archive
Transaction Support	Yes	No	Yes	No
Locking	Row-level	Table-level	Table-level	Table-level
ACID Compliance	Yes	No	No	No
Foreign Keys	Yes	No	No	No
Full-text Index	Yes (MySQL 5.7+)	Yes	No	No
Compression	Limited	No	No	Yes
Speed	Good	Very Fast	Fastest	Slow
Use Case	General	Read-heavy	Caching	Archive

1.4 Key Concepts

Databases and Tables

Database: A collection of related tables and objects

```
CREATE DATABASE company;
```

```
USE company;
```

Table: A structured collection of rows and columns

```
CREATE TABLE employees (
    emp_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100),
    salary DECIMAL(10, 2),
    department VARCHAR(50)
```

);

Rows and Columns

Row (Record): A single data entry with values for each column **Column (Field):** A named attribute of data with a specific data type

Primary Keys and Indexes

Primary Key: Uniquely identifies each row in a table

PRIMARY KEY (emp_id)

Index: Data structure for fast data retrieval

CREATE INDEX idx_name ON employees(name);

CREATE UNIQUE INDEX idx_email ON users(email);

Relationships and Foreign Keys

Foreign Key: Maintains referential integrity between tables

CREATE TABLE departments (

dept_id INT PRIMARY KEY,

dept_name VARCHAR(100)

);

CREATE TABLE employees (

emp_id INT PRIMARY KEY,

dept_id INT,

FOREIGN KEY (dept_id) REFERENCES departments(dept_id)

);

1.5 MySQL Versions and Features

MySQL Version History

MySQL 5.7 (Released 2013)

- InnoDB becomes default engine
- Group Replication introduced
- JSON support added
- Improved Performance Schema

MySQL 8.0 (Released 2018)

- Window functions support
- Common Table Expressions (CTEs)
- Data Dictionary in InnoDB
- Enhanced security (caching_sha2_password default)
- Role-based access control
- Instant DDL operations

Key Differences MySQL 5.7 → 8.0:

-- MySQL 8.0+ Window Functions

```
SELECT employee_id, salary,  
       ROW_NUMBER() OVER (PARTITION BY department ORDER BY salary DESC) as rank  
FROM employees;
```

-- MySQL 8.0+ CTEs

```
WITH salary_stats AS (  
    SELECT department, AVG(salary) as avg_sal  
    FROM employees  
    GROUP BY department  
)  
SELECT * FROM salary_stats WHERE avg_sal > 50000;
```

-- MySQL 8.0+ Roles

```
CREATE ROLE 'data_analyst';
```

```
GRANT SELECT ON company.* TO 'data_analyst';
```

1.6 Summary: Key Takeaways

1. **MySQL Basics:** Open-source RDBMS with ACID-compliant InnoDB engine
2. **Architecture:** Layered design with connection, SQL, storage engine, and file system layers
3. **Storage Engines:** InnoDB (default), MyISAM (fast), Memory (RAM), Archive (compression)
4. **Engine Selection:** Choose based on use case (transactions, speed, storage)
5. **Version Considerations:** MySQL 8.0 with advanced features like window functions and CTEs