

[MYSQL BASICS] (CheatSheet)

1. Database Operations

- **Create a database:** `CREATE DATABASE dbname;`
- **Drop a database:** `DROP DATABASE dbname;`
- **Select a database:** `USE dbname;`
- **List all databases:** `SHOW DATABASES;`

2. Table Operations

- **Create a table:** `CREATE TABLE tablename (id INT, data VARCHAR(100));`
- **Drop a table:** `DROP TABLE tablename;`
- **Rename a table:** `RENAME TABLE oldname TO newname;`
- **List all tables:** `SHOW TABLES;`
- **Describe table structure:** `DESCRIBE tablename;`
- **Truncate a table (remove all rows):** `TRUNCATE TABLE tablename;`
- **Add a column:** `ALTER TABLE tablename ADD column_name datatype;`
- **Drop a column:** `ALTER TABLE tablename DROP column_name;`
- **Rename a column:** `ALTER TABLE tablename CHANGE old_column_name new_column_name datatype;`
- **Modify a column type:** `ALTER TABLE tablename MODIFY column_name new_datatype;`
- **Add a primary key:** `ALTER TABLE tablename ADD PRIMARY KEY (column_name);`
- **Drop a primary key:** `ALTER TABLE tablename DROP PRIMARY KEY;`
- **Add a unique constraint:** `ALTER TABLE tablename ADD UNIQUE (column_name);`
- **Add a foreign key:** `ALTER TABLE tablename ADD CONSTRAINT fk_name FOREIGN KEY (column_name) REFERENCES other_table(column_name);`
- **Drop a foreign key:** `ALTER TABLE tablename DROP FOREIGN KEY fk_name;`
- **Create an index:** `CREATE INDEX index_name ON tablename (column_name);`
- **Drop an index:** `DROP INDEX index_name ON tablename;`

3. Data Manipulation

- **Insert a row:** `INSERT INTO tablename (column1, column2) VALUES (value1, value2);`
- **Update rows:** `UPDATE tablename SET column1 = value1 WHERE condition;`
- **Delete rows:** `DELETE FROM tablename WHERE condition;`
- **Select data:** `SELECT * FROM tablename;`

- **Select data with condition:** `SELECT * FROM tablename WHERE condition;`
- **Select and order data:** `SELECT * FROM tablename ORDER BY column ASC|DESC;`
- **Select distinct rows:** `SELECT DISTINCT column FROM tablename;`
- **Count rows:** `SELECT COUNT(*) FROM tablename;`
- **Sum a column:** `SELECT SUM(column) FROM tablename;`
- **Average a column:** `SELECT AVG(column) FROM tablename;`
- **Join tables (INNER JOIN):** `SELECT * FROM table1 INNER JOIN table2 ON table1.common_column = table2.common_column;`
- **Left Join:** `SELECT * FROM table1 LEFT JOIN table2 ON table1.common_column = table2.common_column;`
- **Right Join:** `SELECT * FROM table1 RIGHT JOIN table2 ON table1.common_column = table2.common_column;`
- **Full Join:** `SELECT * FROM table1 LEFT JOIN table2 ON table1.common_column = table2.common_column UNION SELECT * FROM table1 RIGHT JOIN table2 ON table1.common_column = table2.common_column;`
- **Group data:** `SELECT column, COUNT(*) FROM tablename GROUP BY column;`
- **Having clause:** `SELECT column, COUNT(*) FROM tablename GROUP BY column HAVING COUNT(*) > 1;`
- **Limit results:** `SELECT * FROM tablename LIMIT number;`
- **Offset results:** `SELECT * FROM tablename LIMIT number OFFSET number;`

4. Subqueries and Advanced Queries

- **Subquery in SELECT:** `SELECT (SELECT COUNT(*) FROM table2 WHERE condition) AS count FROM table1;`
- **Subquery in FROM:** `SELECT * FROM (SELECT * FROM table1) AS subtable;`
- **Subquery in WHERE:** `SELECT * FROM table1 WHERE column IN (SELECT column FROM table2);`

5. User and Permissions

- **Create a user:** `CREATE USER 'username'@'host' IDENTIFIED BY 'password';`
- **Grant permissions:** `GRANT ALL PRIVILEGES ON dbname.* TO 'username'@'host';`
- **Revoke permissions:** `REVOKE ALL PRIVILEGES ON dbname.* FROM 'username'@'host';`
- **Show grants:** `SHOW GRANTS FOR 'username'@'host';`
- **Drop a user:** `DROP USER 'username'@'host';`
- **Set password:** `SET PASSWORD FOR 'username'@'host' = PASSWORD('newpassword');`

6. Backup and Recovery

- **Backup a database:** `mysqldump -u username -p dbname > backupfile.sql;`
- **Restore a database:** `mysql -u username -p dbname < backupfile.sql;`

7. Performance and Maintenance

- **Analyze a table:** `ANALYZE TABLE tablename;`
- **Check a table:** `CHECK TABLE tablename;`
- **Optimize a table:** `OPTIMIZE TABLE tablename;`
- **Flush privileges:** `FLUSH PRIVILEGES;`
- **Show processlist:** `SHOW PROCESSLIST;`
- **Explain query:** `EXPLAIN SELECT * FROM tablename;`

8. Miscellaneous

- **Set SQL mode:** `SET sql_mode = 'modes';`
- **Show version:** `SELECT VERSION();`
- **Current database:** `SELECT DATABASE();`
- **List all indexes:** `SHOW INDEX FROM tablename;`
- **Change storage engine:** `ALTER TABLE tablename ENGINE = InnoDB;`
- **Change the delimiter:** `DELIMITER $$` (Useful for stored procedures)
- **Execute a script file:** `SOURCE /path/to/script.sql;`
- **List all triggers:** `SHOW TRIGGERS;`
- **Show create table statement:** `SHOW CREATE TABLE tablename;`
- **Show character set:** `SHOW CHARACTER SET;`
- **Show collation:** `SHOW COLLATION;`

9. Text and String Functions

- **Concatenate strings:** `SELECT CONCAT(column1, ' ', column2) AS full_name FROM tablename;`
- **Convert to uppercase:** `SELECT UPPER(column) FROM tablename;`
- **Convert to lowercase:** `SELECT LOWER(column) FROM tablename;`
- **Substring:** `SELECT SUBSTRING(column, 1, 10) FROM tablename;`
- **Replace text:** `SELECT REPLACE(column, 'old', 'new') FROM tablename;`
- **Length of a string:** `SELECT LENGTH(column) FROM tablename;`
- **Trim spaces:** `SELECT TRIM(column) FROM tablename;`

- **Find position of substring:** `SELECT INSTR(column, 'substring') FROM tablename;`

10. Numeric and Date Functions

- **Round a number:** `SELECT ROUND(column, decimals) FROM tablename;`
- **Get current date:** `SELECT CURDATE();`
- **Get current time:** `SELECT CURTIME();`
- **Extract year from date:** `SELECT YEAR(column) FROM tablename;`
- **Extract month from date:** `SELECT MONTH(column) FROM tablename;`
- **Date difference:** `SELECT DATEDIFF(date1, date2) FROM tablename;`
- **Add days to a date:** `SELECT DATE_ADD(column, INTERVAL 10 DAY) FROM tablename;`
- **Format date:** `SELECT DATE_FORMAT(column, '%Y-%m-%d') FROM tablename;`

11. Conditional Expressions

- **CASE statement:** `SELECT CASE WHEN condition THEN result ELSE default_result END FROM tablename;`
- **IF function:** `SELECT IF(condition, value_if_true, value_if_false) FROM tablename;`
- **IFNULL function:** `SELECT IFNULL(column, 'default_value') FROM tablename;`
- **NULL handling with COALESCE:** `SELECT COALESCE(column, 'default_value') FROM tablename;`

12. Advanced Join Techniques

- **Self Join:** `SELECT a.column, b.column FROM table a JOIN table b ON a.common_column = b.common_column WHERE condition;`
- **Cross Join:** `SELECT * FROM table1 CROSS JOIN table2;` - Produces Cartesian product of the tables.
- **Natural Join:** `SELECT * FROM table1 NATURAL JOIN table2;` - Automatically joins using columns with the same names.

13. Set Operations

- **Union:** `SELECT column FROM table1 UNION SELECT column FROM table2;` - Combines results and removes duplicates.
- **Union All:** `SELECT column FROM table1 UNION ALL SELECT column FROM table2;` - Combines all results, including duplicates.

- **Intersect (find matching rows, simulate using joins or exists):** `SELECT column FROM table1 WHERE EXISTS (SELECT column FROM table2 WHERE table1.column = table2.column);`
- **Except (find non-matching rows, simulate with NOT EXISTS):** `SELECT column FROM table1 WHERE NOT EXISTS (SELECT column FROM table2 WHERE table1.column = table2.column);`

14. Database Security

- **Encrypting data with AES:** `SELECT AES_ENCRYPT('text', 'key_string');`
- **Decrypting data with AES:** `SELECT AES_DECRYPT(column, 'key_string') FROM tablename;`
- **Generating MD5 hashes:** `SELECT MD5('text');`
- **Generating SHA2 hashes:** `SELECT SHA2('text', 256);`

15. Transaction Control

- **Start a transaction:** `START TRANSACTION;`
- **Commit a transaction:** `COMMIT;`
- **Rollback a transaction:** `ROLLBACK;`
- **Set transaction isolation level:** `SET TRANSACTION ISOLATION LEVEL READ COMMITTED;`

16. Database Administration

- **Viewing current user:** `SELECT USER();`
- **Changing the database engine:** `ALTER TABLE tablename ENGINE = MyISAM;`
- **Viewing table storage engine:** `SHOW TABLE STATUS LIKE 'tablename';`
- **Checking server status:** `SHOW STATUS LIKE 'Key%';`
- **Viewing active connections:** `SHOW PROCESSLIST;`
- **Flushing tables or caches:** `FLUSH TABLES; FLUSH QUERY CACHE;`

17. Debugging and Profiling

- **Show warnings after a query:** `SHOW WARNINGS;`
- **Show last query cost:** `SHOW STATUS LIKE 'Last_query_cost';`
- **Profile queries for performance:** `SET PROFILING = 1; SHOW PROFILES;`

18. Index and Performance Optimization

- **Creating a full-text index:** `CREATE FULLTEXT INDEX ft_index ON tablename (column);`
- **Using EXPLAIN to analyze queries:** `EXPLAIN SELECT * FROM tablename WHERE column = 'value';`
- **Optimizing queries with hints:** `SELECT /*+ MAX_EXECUTION_TIME(1000) */ * FROM tablename;`

19. Numeric and Date Functions

- **Add days to a date:** `SELECT DATE_ADD(date_column, INTERVAL 10 DAY) FROM tablename;`
- **Subtract days from a date:** `SELECT DATE_SUB(date_column, INTERVAL 5 DAY) FROM tablename;`
- **Format dates:** `SELECT DATE_FORMAT(date_column, '%Y-%m-%d') FROM tablename;`
- **Convert strings to dates:** `SELECT STR_TO_DATE('2024-01-01', '%Y-%m-%d') FROM tablename;`

20. Aggregate Functions Beyond Basics

- **Find the minimum value:** `SELECT MIN(column) FROM tablename;`
- **Find the maximum value:** `SELECT MAX(column) FROM tablename;`
- **Standard deviation:** `SELECT STDDEV(column) FROM tablename;`
- **Variance:** `SELECT VARIANCE(column) FROM tablename;`
- **Group concatenation:** `SELECT GROUP_CONCAT(column SEPARATOR ', ') FROM tablename GROUP BY group_column;`

21. Database Maintenance and Inspection

- **Show table status:** `SHOW TABLE STATUS LIKE 'tablename';`
- **Show engine status:** `SHOW ENGINE INNODB STATUS;`
- **View the error log:** `SHOW VARIABLES LIKE 'log_error';` (Then locate the file on the server)

22. Information Schema

- **Get table information:** `SELECT * FROM INFORMATION_SCHEMA.TABLES WHERE TABLE_SCHEMA = 'dbname';`
- **Get columns of a table:** `SELECT * FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME = 'tablename';`
- **Find specific table indexes:** `SELECT * FROM INFORMATION_SCHEMA.STATISTICS WHERE TABLE_NAME = 'tablename';`

23. Performance Schema for Diagnostics

- **Check slow queries:** `SELECT * FROM performance_schema.events_statements_summary_by_digest WHERE SCHEMA_NAME = 'dbname' ORDER BY SUM_TIMER_WAIT DESC;`
- **Monitor wait events:** `SELECT * FROM performance_schema.events_waits_summary_global_by_event_name;`