

MYSQL QUICK REFERENCE

Databases, Tables, CRUD, joins, indexes, users

Connecting

Command Line

```
mysql -u root -p
mysql -u user -p -h 127.0.0.1 -P 3306
mysql -u user -p mydb < dump.sql
```

Connection String

```
mysql://user:password@host:3306/dbname
mysql -u user -p -e "SELECT VERSION();"
```

Status Commands

```
STATUS;
SHOW VARIABLES LIKE 'port';
SHOW PROCESSLIST;
```

Databases & Tables

Database Operations

```
CREATE DATABASE mydb CHARACTER SET utf8mb4;
SHOW DATABASES;
USE mydb;
DROP DATABASE mydb;
```

Table Operations

```
CREATE TABLE users (
  id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(100) NOT NULL,
  email VARCHAR(255) UNIQUE,
  created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

Table Info

```
SHOW TABLES;
DESCRIBE users;
SHOW CREATE TABLE users;
```

Alter Table

```
ALTER TABLE users ADD COLUMN age INT;
ALTER TABLE users MODIFY COLUMN name VARCHAR(200);
ALTER TABLE users DROP COLUMN age;
ALTER TABLE users RENAME TO customers;
```

Data Types

Numeric

INT 4-byte integer (-2B to 2B)
BIGINT 8-byte integer
DECIMAL(p,s) Exact numeric (e.g., DECIMAL(10,2))
FLOAT / DOUBLE Approximate floating-point
BOOLEAN Alias for TINYINT(1)

String

VARCHAR(n) Variable-length string up to n chars
TEXT Up to 65 KB text
MEDIUMTEXT Up to 16 MB text
CHAR(n) Fixed-length string, padded
ENUM('a','b') One value from a defined set

Date & Time

DATE YYYY-MM-DD
DATETIME YYYY-MM-DD HH:MM:SS
TIMESTAMP UTC-stored datetime
TIME HH:MM:SS
JSON Native JSON document type

CRUD

Insert

```
INSERT INTO users (name, email)
VALUES ('Alice', 'alice@example.com');

INSERT INTO users (name, email) VALUES
('Bob', 'bob@ex.com'),
('Carol', 'carol@ex.com');
```

Select

```
SELECT * FROM users WHERE id = 1;
SELECT name, email FROM users
ORDER BY name LIMIT 10 OFFSET 20;
```

Update

```
UPDATE users SET email = 'new@ex.com'
WHERE id = 1;
UPDATE users SET active = 0
WHERE last_login < '2025-01-01';
```

Delete

```
DELETE FROM users WHERE id = 1;
TRUNCATE TABLE users; -- fast, resets AUTO_INCREMENT
```

Upsert

```
INSERT INTO users (id, name, email)
VALUES (1, 'Alice', 'a@ex.com')
ON DUPLICATE KEY UPDATE
name = VALUES(name), email = VALUES(email);
```

Joins

Join Types

INNER JOIN Rows matching in both tables
LEFT JOIN All left rows + matching right
RIGHT JOIN All right rows + matching left
CROSS JOIN Cartesian product
SELF JOIN Table joined with itself

Join Examples

```
SELECT u.name, o.total
FROM users u
INNER JOIN orders o ON u.id = o.user_id;

SELECT e.name, m.name AS manager
FROM employees e
LEFT JOIN employees m ON e.manager_id = m.id;
```

Indexes

Create & Drop

```
CREATE INDEX idx_name ON users(name);
CREATE UNIQUE INDEX idx_email ON users(email);
CREATE INDEX idx_composite
ON orders(user_id, created_at);
DROP INDEX idx_name ON users;
```

Fulltext Index

```
ALTER TABLE posts ADD FULLTEXT(title, body);
SELECT * FROM posts
WHERE MATCH(title, body) AGAINST('mysql');
```

Index Info

```
SHOW INDEX FROM users;
EXPLAIN SELECT * FROM users WHERE name = 'Alice';
```

Functions

String Functions

CONCAT(a, b) Concatenate strings
SUBSTRING(s, pos, len) Extract substring
UPPER(s) / LOWER(s) Change case
TRIM(s) Remove leading/trailing spaces
LENGTH(s) String length in bytes
REPLACE(s, from, to) Replace occurrences

Date Functions

NOW() Current datetime
CURDATE() Current date
DATE_ADD(d, INTERVAL n DAY) Add interval to date
DATEDIFF(d1, d2) Difference in days
DATE_FORMAT(d, fmt) Format date (e.g., '%Y-%m-%d')

Aggregate Functions

```
SELECT COUNT(*), AVG(price), SUM(qty),
       MIN(price), MAX(price)
FROM products
GROUP BY category HAVING COUNT(*) > 5;
```

Users & Permissions

User Management

```
CREATE USER 'app'@'localhost'
IDENTIFIED BY 'secret';
ALTER USER 'app'@'localhost'
IDENTIFIED BY 'newsecret';
DROP USER 'app'@'localhost';
```

Grants

```
GRANT ALL ON mydb.* TO 'app'@'localhost';
GRANT SELECT, INSERT ON mydb.users
TO 'reader'@'%';
REVOKE INSERT ON mydb.users
FROM 'reader'@'%';
FLUSH PRIVILEGES;
```

View Permissions

```
SHOW GRANTS FOR 'app'@'localhost';
SELECT user, host FROM mysql.user;
```

Backup & Restore

mysqldump

```
mysqldump -u root -p mydb > backup.sql
mysqldump -u root -p --all-databases > all.sql
mysqldump -u root -p mydb users > users.sql
```

Restore

```
mysql -u root -p mydb < backup.sql
mysql -u root -p -e "SOURCE /path/backup.sql"
```

Binary Log

```
SHOW BINARY LOGS;
SHOW BINLOG EVENTS IN 'binlog.000001';
mysqlbinlog binlog.000001 | mysql -u root -p
```

Common Patterns

Pagination

```
SELECT * FROM users
ORDER BY id LIMIT 20 OFFSET 40; -- page 3
```

Transactions

```
START TRANSACTION;
UPDATE accounts SET balance = balance - 100
WHERE id = 1;
UPDATE accounts SET balance = balance + 100
WHERE id = 2;
COMMIT; -- or ROLLBACK;
```

Conditional Insert

```
INSERT IGNORE INTO users (email, name)
VALUES ('a@ex.com', 'Alice');
-- skips if email already exists (UNIQUE)
```

Variables & Prepared Statements

```
SET @name = 'Alice';
PREPARE stmt FROM
"SELECT * FROM users WHERE name = ?";
EXECUTE stmt USING @name;
DEALLOCATE PREPARE stmt;
```