

# REFERENCIA RÁPIDA DE MYSQL

Bases de datos, tablas, CRUD, joins, índices, usuarios

## Conexión

### Línea de comandos

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

### Cadena de conexión

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

### Comandos de estado

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

## Bases de datos y tablas

### Operaciones de base de datos

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

### Operaciones de tabla

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

### Información de tabla

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

### Modificar tabla

```
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;
```

## Tipos de datos

### Numéricos

**INT** Entero de 4 bytes (-2B a 2B)  
**BIGINT** Entero de 8 bytes  
**DECIMAL(p,s)** Numérico exacto (p. ej., DECIMAL(10,2))  
**FLOAT / DOUBLE** Punto flotante aproximado  
**BOOLEAN** Alias de TINYINT(1)

### Cadenas

**VARCHAR(n)** Cadena de longitud variable hasta n chars  
**TEXT** Texto de hasta 65 KB  
**MEDIUMTEXT** Texto de hasta 16 MB  
**CHAR(n)** Cadena de longitud fija, rellena  
**ENUM('a','b')** Un valor de un conjunto definido

### Fecha y hora

**DATE** YYYY-MM-DD  
**DATETIME** YYYY-MM-DD HH:MM:SS  
**TIMESTAMP** Fecha y hora almacenada en UTC  
**TIME** HH:MM:SS  
**JSON** Tipo de documento JSON nativo

## CRUD

### Insertar

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

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

### Seleccionar

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

### Actualizar

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

### Eliminar

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

### Upsert

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

## Joins

### Tipos de join

**INNER JOIN** Filas coincidentes en ambas tablas  
**LEFT JOIN** Todas las filas izquierdas + coincidentes derechas  
**RIGHT JOIN** Todas las filas derechas + coincidentes izquierdas  
**CROSS JOIN** Producto cartesiano  
**SELF JOIN** Tabla unida consigo misma

### Ejemplos de join

```
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;
```

## Índices

### Crear y eliminar

```
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;
```

### Índice de texto completo

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

### Información de índices

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

## Funciones

### Funciones de cadena

**CONCAT(a, b)** Concatenar cadenas  
**SUBSTRING(s, pos, len)** Extraer subcadena  
**UPPER(s) / LOWER(s)** Cambiar mayúsculas/minúsculas  
**TRIM(s)** Eliminar espacios al inicio/final  
**LENGTH(s)** Longitud de cadena en bytes  
**REPLACE(s, from, to)** Reemplazar ocurrencias

### Funciones de fecha

**NOW()** Fecha y hora actual  
**CURDATE()** Fecha actual  
**DATE\_ADD(d, INTERVAL n DAY)** Agregar intervalo a la fecha  
**DATEDIFF(d1, d2)** Diferencia en días  
**DATE\_FORMAT(d, fmt)** Formatear fecha (p. ej., '%Y-%m-%d')

### Funciones de agregación

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

## Usuarios y permisos

### Gestión de usuarios

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

### Permisos

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

### Ver permisos

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

## Backup y restauración

### mysqldump

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

### Restaurar

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

### Registro binario

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

## Patrones comunes

### Paginación

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

### Transacciones

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

### Insertión condicional

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

### Variables y sentencias preparadas

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