

# Référence rapide MySQL

Bases de données, tables, CRUD, jointures, index, utilisateurs

## Connexion

### Ligne de commande

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

### Chaîne de connexion

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

### Commandes de statut

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

## Bases de données et tables

### Opérations sur les bases de données

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

### Opérations sur les tables

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

### Informations sur les tables

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

### Modifier une 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;
```

## Types de données

### Numérique

<b>INT</b>	Entier 4 octets (-2 milliards à 2 milliards)
<b>BIGINT</b>	Entier 8 octets
<b>DECIMAL(p, s)</b>	Numérique exact (ex. DECIMAL(10,2))
<b>FLOAT / DOUBLE</b>	Virgule flottante approximative
<b>BOOLEAN</b>	Alias pour TINYINT(1)

### Chaîne

<b>VARCHAR(n)</b>	Chaîne de longueur variable jusqu'à n caractères
<b>TEXT</b>	Texte jusqu'à 65 Ko
<b>MEDIUMTEXT</b>	Texte jusqu'à 16 Mo
<b>CHAR(n)</b>	Chaîne de longueur fixe, complétée
<b>ENUM('a', 'b')</b>	Une valeur d'un ensemble défini

### Date et heure

<b>DATE</b>	AAAA-MM-JJ
<b>DATETIME</b>	AAAA-MM-JJ HH:MM:SS
<b>TIMESTAMP</b>	Datetime stocké en UTC
<b>TIME</b>	HH:MM:SS
<b>JSON</b>	Type de document JSON natif

## CRUD

### Insertion

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

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

### Sélection

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

### Mise à jour

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

### Suppression

```
DELETE FROM users WHERE id = 1;
TRUNCATE TABLE users; -- rapide, remet AUTO_INCREMENT à zéro
```

### Upsert

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

## Jointures

### Types de jointures

<b>INNER JOIN</b>	Lignes correspondantes dans les deux tables
<b>LEFT JOIN</b>	Toutes les lignes gauches + correspondances droites
<b>RIGHT JOIN</b>	Toutes les lignes droites + correspondances gauches
<b>CROSS JOIN</b>	Produit cartésien
<b>SELF JOIN</b>	Table jointe à elle-même

### Exemples de jointures

```
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;
```

## Index

### Créer et supprimer

```
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;
```

### Index plein texte

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

### Informations sur les index

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

## Fonctions

### Fonctions de chaînes

<b>CONCAT(a, b)</b>	Concaténer des chaînes
<b>SUBSTRING(s, pos, len)</b>	Extraire une sous-chaîne
<b>UPPER(s) / LOWER(s)</b>	Changer la casse
<b>TRIM(s)</b>	Supprimer les espaces en début/fin
<b>LENGTH(s)</b>	Longueur de chaîne en octets
<b>REPLACE(s, from, to)</b>	Remplacer les occurrences

### Fonctions de date

<b>NOW()</b>	Datetime actuelle
<b>CURDATE()</b>	Date actuelle
<b>DATE_ADD(d, INTERVAL n DAY)</b>	Ajouter un intervalle à une date
<b>DATEDIFF(d1, d2)</b>	Différence en jours
<b>DATE_FORMAT(d, fmt)</b>	Formater une date (ex. '%Y-%m-%d')

### Fonctions d'agrégation

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

## Utilisateurs et permissions

### Gestion des utilisateurs

```
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;
```

### Voir les permissions

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

## Sauvegarde et restauration

### mysqldump

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

### Restaurer

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

### Journal binaire

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

## Modèles courants

### Pagination

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

# Référence rapide MySQL

---

## Transactions

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

## Insertion conditionnelle

```
INSERT IGNORE INTO users (email, name)
VALUES ('a@ex.com', 'Alice');
-- ignoré si l'email existe déjà (UNIQUE)
```

## Variables et instructions préparées

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