MySQL Cheatsheet

What is a Database?

A **database** is a structured collection of interrelated data stored together to serve multiple applications.

MySQL Elements

Literals

Literals refer to fixed data values:

```
17        -- Numeric literal
'Harry'   -- Text literal
12.5      -- Real literal
```

Data Types

MySQL provides several data types:

```
# Numeric Types
TINYINT, SMALLINT, MEDIUMINT, INT, BIGINT
FLOAT(M,D), DOUBLE(M,D), DECIMAL(M,D)

# Date/Time Types
DATE        -- YYYY-MM-DD
DATETIME    -- YYYY-MM-DD HH:MM:SS
TIME        -- HH:MM:SS
```

```
YEAR        -- YYYY

# String/Text Types
CHAR(M)        -- Fixed-length string
VARCHAR(M)     -- Variable-length string
TEXT           -- Large text
BLOB           -- Binary large object (for files/images)
ENUM('x','y')  -- One value from a defined set
SET('x','y')   -- Multiple values from a defined set
```

NULL Values

NULL represents missing/unknown data.

Comments

```
/* Multi-line comment */
# Single-line comment
-- Single-line comment
```

MySQL Calculations

```
SELECT 5+8;   -- Addition
SELECT 15-5;  -- Subtraction
SELECT 5*5;   -- Multiplication
SELECT 24/4;  -- Division
```

> **Tip:** SQL is **case-insensitive**, but keywords are usually written in UPPERCASE for readability.

Accessing Databases

```
SHOW DATABASES;      -- List all databases
USE database_name;   -- Switch to a database
SHOW TABLES;         -- List all tables in the current database
```

Creating Tables

```
CREATE TABLE table_name (
  column1 datatype,
  column2 datatype,
  column3 datatype
);
```

Inserting Data

```
INSERT INTO table_name (col1, col2) VALUES (val1, val2);
INSERT INTO table_name VALUES (val1, val2, val3); -- All columns
INSERT INTO table_name (col1, col2, col3) VALUES (val1, val2, NULL); -- NULL insert
INSERT INTO table_name (date_col) VALUES ('2021-12-10'); -- Date insert
```

Retrieving Data (SELECT)

```
SELECT * FROM table_name;                       -- All columns
SELECT col1, col2 FROM table_name;              -- Specific columns
SELECT DISTINCT col1 FROM table_name;           -- Unique values
SELECT col1, col2 AS alias FROM table_name;     -- Column alias
SELECT * FROM table_name WHERE condition;       -- Filter rows
SELECT * FROM table_name WHERE col BETWEEN 10 AND 20;  -- Range filter
SELECT * FROM table_name WHERE col IN (1,2,3);  -- List match
SELECT * FROM table_name WHERE col NOT IN (1,2,3);
SELECT * FROM table_name WHERE col LIKE 'Ha%';  -- Pattern match
SELECT * FROM table_name WHERE col IS NULL;     -- NULL search
```

Constraints

```
NOT NULL   -- Disallow NULL
DEFAULT    -- Set default value
UNIQUE     -- Ensure unique values
CHECK (condition) -- Ensure condition is true
PRIMARY KEY(col1)  -- Unique + Not Null
FOREIGN KEY (col) REFERENCES other_table(col)
```

Modifying Data

```
UPDATE table_name
SET col1 = new_value, col2 = new_value
WHERE condition;
```

Deleting Data

```
DELETE FROM table_name WHERE condition;
```

Ordering Results

```
SELECT * FROM table_name ORDER BY col ASC;
SELECT * FROM table_name ORDER BY col DESC;
SELECT * FROM table_name ORDER BY col1 ASC, col2 DESC;
```

Grouping Data

```
SELECT col, COUNT(*) FROM table_name GROUP BY col;
SELECT col, AVG(salary) FROM table_name GROUP BY col HAVING AVG(salary)>50000;
```

Altering Table Structure

```
ALTER TABLE table_name ADD new_column datatype;
ALTER TABLE table_name MODIFY column_name datatype;
ALTER TABLE table_name CHANGE old_name new_name datatype;
```

Dropping Table

```
DROP TABLE table_name;
```

MySQL Functions

String Functions

```
SELECT CHAR(72,97,114,114,121);
SELECT CONCAT('Harry','Bhai');
SELECT LOWER('Harry');
SELECT UPPER('CodeWithHarry');
SELECT SUBSTRING('HelloWorld', 1, 5);
SELECT TRIM('   Harry   ');
SELECT INSTR('CodeWithHarry','Harry');
SELECT LENGTH('Harry');
```

Numeric Functions

```
SELECT MOD(11,4);
SELECT POWER(2,3);
SELECT ROUND(15.193,1);
SELECT SQRT(144);
SELECT TRUNCATE(15.75,1);
```

Date/Time Functions

```
SELECT CURDATE();
SELECT NOW();
SELECT DATE('2021-12-10 12:00:00');
SELECT YEAR(NOW());
SELECT MONTH(NOW());
SELECT DAY(NOW());
SELECT SYSDATE();
```

Aggregate Functions

```
SELECT AVG(col) FROM table_name;
SELECT COUNT(*) FROM table_name;
SELECT MAX(col) FROM table_name;
SELECT MIN(col) FROM table_name;
SELECT SUM(col) FROM table_name;
```

Joins

```
-- Inner Join
SELECT t1.col, t2.col
FROM table1 t1
INNER JOIN table2 t2 ON t1.id = t2.id;

-- Left Join
SELECT ...
FROM table1
LEFT JOIN table2 ON table1.id = table2.id;

-- Right Join
SELECT ...
FROM table1
RIGHT JOIN table2 ON table1.id = table2.id;

-- Full Join (MySQL Workaround)
SELECT ...
FROM table1
LEFT JOIN table2 ON table1.id=table2.id
UNION
SELECT ...
FROM table1
RIGHT JOIN table2 ON table1.id=table2.id;

-- Self Join
SELECT a.col, b.col
FROM table a, table b
WHERE a.id < b.id;
```

Indexes (Performance)

```
CREATE INDEX idx_name ON table_name(column_name);
DROP INDEX idx_name ON table_name;
SHOW INDEX FROM table_name;
```

Views (Virtual Tables)

```
CREATE VIEW view_name AS SELECT col1, col2 FROM table_name WHERE condition;
DROP VIEW view_name;
```

Transactions (Atomic Operations)

```
START TRANSACTION;
UPDATE accounts SET balance = balance - 500 WHERE id = 1;
UPDATE accounts SET balance = balance + 500 WHERE id = 2;
COMMIT; -- Save changes
ROLLBACK; -- Undo changes
```

User Management & Privileges

```
CREATE USER 'username'@'localhost' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON dbname.* TO 'username'@'localhost';
FLUSH PRIVILEGES;
```
