# SQL 学习笔记（Structured Query Language）

## 一、基础概念 Basics

SQL (Structured Query Language) 是用于管理和操作数据库的标准语言。  
  
Database（数据库）是有组织的数据集合，通过 DBMS（Database Management System，数据库管理系统）进行管理与交互。  
  
常见数据库管理系统（DBMS）：MySQL, SQLite, PostgreSQL, Oracle Database。

## 二、MySQL 与 SQL 的区别

SQL 是标准语言，用于操作数据库。  
MySQL 是一个数据库管理系统，基于 SQL 语言实现。

## 三、数据库操作 Database Commands

1️⃣ 查看所有数据库: SHOW DATABASES;  
2️⃣ 清屏: Ctrl + C 后输入 system cls  
3️⃣ 创建数据库: CREATE DATABASE <database\_name>;  
 示例: CREATE DATABASE soap\_store;  
4️⃣ 删除数据库: DROP DATABASE <database\_name>;  
5️⃣ 选择使用数据库: USE <database\_name>;  
6️⃣ 查看当前使用的数据库: SELECT DATABASE();

## 四、数据类型 Data Types

文档参考: https://dev.mysql.com/doc/refman/8.4/en/string-types.html

## 五、创建与查看表格 Tables

创建表格:  
CREATE TABLE cats (name VARCHAR(50), age INT);  
CREATE TABLE dogs (name VARCHAR(50), breed VARCHAR(50), age INT);  
  
查看所有表: SHOW TABLES;  
查看表格结构: SHOW COLUMNS FROM cats; 或 DESC cats;  
删除表格: DROP TABLE <table\_name>;

## 六、插入与查看数据 Insert & Select

插入数据:  
INSERT INTO cats (name, age) VALUES ('Blue Steele', 5);  
INSERT INTO cats (name, age) VALUES ('Jenkins', 7);  
  
交换顺序插入:  
INSERT INTO cats (age, name) VALUES (2, 'Beth');  
  
一次插入多行:  
INSERT INTO cats (name, age) VALUES ('Meatball', 5), ('Turkey', 1), ('Potato Face', 15);  
  
查看数据:  
SELECT \* FROM cats;

## 七、NOT NULL 与 DEFAULT

null 不能为空 default 默认值

如果代码里写了not null -> 强制你必须提供东西，那么就不能在这个column里面填null了。比如手机号码。但是如果我没specify not null 只设定了 default，那么就可以写null

NOT NULL: 不允许为空。<如果空了就不允许录入，报错>  
CREATE TABLE cats2 (name VARCHAR(100) NOT NULL, age INT NOT NULL);  
  
DEFAULT 只设定默认值 <空缺将自动变为Deafault的数字，可以主动输入NULL,NULL的值>  
CREATE TABLE cats3 (name VARCHAR(20) DEFAULT 'no name provided', age INT DEFAULT 99);

结合使用 NOT NULL 且设定default 默认值:<可以空值，但是输入NULL就报错>  
CREATE TABLE cats4 (name VARCHAR(20) NOT NULL DEFAULT 'unnamed', age INT NOT NULL DEFAULT 99);

## 八、主键 PRIMARY KEY

主键（Primary Key） 是表中的一个或多个字段，用来唯一标识表中的每一行数据。它保证每条记录都是独一无二的，并且不能为空（NOT NULL）。

Each table can have only **one primary key**. The primary key value must be **unique** (no duplicates). The primary key value **cannot be NULL.**

写法一:  
CREATE TABLE unique\_cats (cat\_id INT PRIMARY KEY, name VARCHAR(100) NOT NULL, age INT NOT NULL);  
  
写法二:  
CREATE TABLE unique\_cats2 (cat\_id INT, name VARCHAR(100) NOT NULL, age INT NOT NULL, PRIMARY KEY (cat\_id));

## 九、自增 AUTO\_INCREMENT 每列自动+1

1CREATE TABLE unique\_cats3 (cat\_id INT AUTO\_INCREMENT, name VARCHAR(100) NOT NULL, age INT NOT NULL, PRIMARY KEY (cat\_id));

2CREATE TABLE cats5 (cat\_id INT AUTO\_INCREMENT PRIMARY KEY , name VARCHAR(100) NOT NULL );

## 十、额外技巧 Tips

注释: -- 或 Ctrl + / 自动加注释  
字符串转义: 用 \ 来逃离引号，如 'It\'s a cat'

## 十一、CRUD 操作（Create / Read / Update / Delete）

### 概述 C - Create -> INSERT （新增数据） R - Read -> SELECT （读取数据） U - Update -> UPDATE （更新数据） D - Delete -> DELETE （删除数据）

### Create（创建）➡ INSERT

插入数据:  
INSERT INTO cats (name, age)  
VALUES ('Blue Steele', 5);

### Read（读取）➡ SELECT

查看所有列:  
SELECT \* FROM cats;  
只查看指定列:  
SELECT name FROM cats;  
SELECT name, breed FROM cats;

### WHERE 子句（条件筛选）

WHERE 可用于 SELECT / UPDATE / DELETE 等命令，用来设定条件:  
SELECT \* FROM cats WHERE age = 4;  
SELECT \* FROM cats WHERE name = 'Egg';

### 别名（Aliases）

在结果中临时重命名列:  
SELECT cat\_id AS id, name FROM cats;

### Update（更新）➡ UPDATE

更新单条或多条数据:  
UPDATE cats SET breed='Shorthair' WHERE breed='Tabby';  
UPDATE cats SET age=14 WHERE name='Misty';  
  
💡 提示: 在执行 UPDATE 前先用 SELECT 检查，避免误改。

### Delete（删除）➡ DELETE

删除指定条件的数据:  
DELETE FROM cats WHERE name='Egg';  
删除所有行（保留表结构）:  
DELETE FROM cats;  
其他示例:  
DELETE FROM cats WHERE age=4;  
DELETE FROM cats WHERE cat\_id=age;

## 十二、String Functions（字符串函数）

### 1️⃣ How to Load SQL Files

In MySQL Workbench:  
1. Select the target database.  
2. Run: SOURCE /path/to/book\_data.sql;  
  
In Terminal:  
1. Login: mysql -u root -p  
2. USE database\_name;  
3. SOURCE /path/to/book\_data.sql;  
  
在 Workbench 选择数据库后执行 SOURCE 命令；Terminal 中先登录再执行文件。

### 2️⃣ CONCAT() and CONCAT\_WS()

CONCAT() joins multiple strings together:  
SELECT CONCAT(author\_fname, ' ', author\_lname) AS author\_name FROM books;  
  
CONCAT\_WS() joins strings with a separator:  
SELECT CONCAT\_WS('-', title, author\_fname, author\_lname) FROM books;  
  
CONCAT 拼接多个字符串；CONCAT\_WS 在字符串间自动添加分隔符。

### 3️⃣ SUBSTRING() / SUBSTR()

Extract part of a string:  
SELECT SUBSTRING('Hello World', 1, 4);  
SELECT SUBSTRING('Hello World', 7);  
SELECT SUBSTRING('Hello World', -3);  
SELECT SUBSTRING(title, 1, 10) AS 'short title' FROM books;  
  
SUBSTR() is alias of SUBSTRING():  
SELECT SUBSTR(author\_lname, 1, 1) FROM books;  
  
Combine with CONCAT:  
SELECT CONCAT(SUBSTRING(title, 1, 10), '...') AS 'short title' FROM books;  
SELECT CONCAT(SUBSTR(author\_fname,1,1), '.', SUBSTR(author\_lname,1,1), '.') FROM books;  
  
SUBSTRING 或 SUBSTR 用于截取子字符串；可结合 CONCAT 使用。

### 4️⃣ REPLACE()

Replace part of a string:  
SELECT REPLACE('Hello World', 'Hell', '%$#@');  
SELECT REPLACE('Hello World', 'l', '7');  
SELECT REPLACE(title, ' ', '-') FROM books;  
  
REPLACE 将字符串中指定内容替换为新的文本。

### 5️⃣ REVERSE()

Reverse the order of characters:  
SELECT REVERSE('abc');  
SELECT REVERSE('Hello World');  
SELECT CONCAT(author\_fname, REVERSE(author\_fname)) FROM books;  
  
REVERSE 反转字符串字符顺序。

### 6️⃣ CHAR\_LENGTH() vs LENGTH()

CHAR\_LENGTH() returns number of characters; LENGTH() returns bytes.  
SELECT LENGTH('海豚'), CHAR\_LENGTH('海豚');  
SELECT author\_lname, CHAR\_LENGTH(author\_lname) AS length FROM books;  
  
CHAR\_LENGTH 计算字符数量，LENGTH 计算字节数。

### 7️⃣ UPPER() / LOWER()

Convert to uppercase or lowercase:  
SELECT UPPER('Hello World');  
SELECT LOWER('Hello World');  
SELECT CONCAT('My favorite book is ', UPPER(title)) FROM books;  
  
UPPER 转大写；LOWER 转小写。

### 8️⃣ INSERT()

Insert or replace part of string:  
SELECT INSERT('Quadratic', 3, 4, 'What');  
SELECT INSERT('Quadratic', 3, 100, 'What');  
  
INSERT(str,pos,len,newstr) 从 pos 开始替换 len 个字符为 newstr。

### 9️⃣ LEFT() and RIGHT()

Get leftmost or rightmost characters:  
SELECT LEFT('foobarbar', 5);  
SELECT RIGHT('omghahalol!', 4);  
  
LEFT 从左取字符；RIGHT 从右取字符。

### 🔟 REPEAT()

Repeat a string:  
SELECT REPEAT('ha', 4);  
  
REPEAT 将字符串重复指定次数。

### 11️⃣ TRIM()

Remove spaces or specific characters:  
SELECT TRIM(' bar ');  
SELECT TRIM(LEADING 'x' FROM 'xxxbarxxx');  
SELECT TRIM(BOTH 'x' FROM 'xxxbarxxx');  
SELECT TRIM(TRAILING 'xyz' FROM 'barxxyz');  
  
TRIM 删除两端空格或指定字符；LEADING 为开头，TRAILING 为结尾，BOTH 为两端。

### 12️⃣ SQL Formatting Tools

Workbench: use brush icon to format.  
Shortcut: Ctrl + B / Cmd + B  
Online Tool: https://codebeautify.org/sqlformatter  
  
可使用 Workbench 或在线工具格式化 SQL 代码。