SQL是设计用于在关系数据库管理系统中管理数据的标准语言。

书写格式：

SQL关键字不区分大小写，SELECT与select表示相同。但是，数据库名称和表名称可能区分大小写，具体取决于操作系统。Unix或Linux平台区分大小写，而Windows平台则不区分大小写。

注释：

单行注释：使用两个连续的连字符（--）作为开始行。

多行注释：在注释前面加一个斜杠，然后加上一个星号（/*），在注释后面加一个星号，然后再加上一个斜杠（*/）。

在MySQL中创建数据库

使用命令行工具在MySQL中创建数据库。

步骤1：调用MySQL命令行工具

要调用MySQL命令行，必须先登录MySQL服务器。要以root用户身份登录，在终端中键入以下命令，然后按Enter。

```
shell>  mysql -u root -p
```

步骤2：建立MySQL数据库

现在，执行以下命令以创建名为demo的数据库。

```
mysql>  CREATE DATABASE demo;
```

如果创建一个已经存在的数据库，则会收到一条错误消息。为了避免这种情况，可以在MySQL中使用可选子句IF NOT EXISTS，如下所示：

```
mysql> CREATE DATABASE IF NOT EXISTS demo;
```

步骤3：选择数据库

键入以下命令，然后按Enter。将看到输出“数据库已更改”。现在，演示数据库被选作所有将来操作的默认数据库。

```
mysql> USE demo;
```

语法

创建表的基本语法可以通过以下方式给出：

```
CREATE TABLE table_name (column1_name data_type constraints,column2_name data_type constraints,....

);
-- MySQL数据库的语法

CREATE TABLE persons (
    id INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(50) NOT NULL,
    birth_date DATE,
    phone VARCHAR(15) NOT NULL UNIQUE

);

-- SQL Server数据库的语法

CREATE TABLE persons (
    id INT NOT NULL PRIMARY KEY IDENTITY(1,1),
    name VARCHAR(50) NOT NULL,
    birth_date DATE,
    phone VARCHAR(15) NOT NULL UNIQUE
```

MySQL支持的最常用的数据类型

数据类型	描述
INT	存储介于-2147483648至2147483647之间的数值
DECIMAL	以精确的精度存储十进制值。
CHAR	存储最大长度为255个字符的定长字符串。
VARCHAR	存储可变长度的字符串，最大长度为65,535个字符。
TEXT	存储最大大小为65,535个字符的字符串。
DATE	以YYYY-MM-DD格式存储日期值。
DATETIME	以YYYY-MM-DD HH：MM：SS格式存储组合的日期/时间值。
TIMESTAMP	存储时间戳值。TIMESTAMP 值存储为自Unix纪元（'1970-01-01 00:00:01'UTC）以来的秒数。

SQL 约束

是对表的一个或多个列的限制，以限制可以存储在该列中的值的类型。约束提供了一种标准机制来维护数据库表内数据的准确性和完整性。
SQL中有几种不同类型的约束，包括：

- NOT NULL:限制指定列不接受NULL值。
- PRIMARY KEY：具有唯一标识表中的行值的列的列或集。
- UNIQUE：限制一个或多个列在表中包含唯一值。

- DEFAULT：指定列的默认值。
- FOREIGN KEY：外键(FK)是一列或列的组合，用于在两个表中的数据之间建立和加强关系。
- CHECK：用于限制可以放置在列中的值。

插入数据

```
INSERT INTO table_name (column1,column2,...) VALUES (value1,value2,...);
```

选择数据

```
SELECT column1_name, column2_name, columnN_name FROM table_name;
```

根据条件选择记录

提取满足指定条件的那些记录。

```
SELECT column_list FROM table_name WHERE condition;
```

AND与OR运算符

AND运算符将两个条件组合在一起，并且仅当两个条件的结果都为TRUE时才返回TRUE。
OR运算符将两个条件组合在一起的逻辑运算符，但是当两个条件中的任何一个为TRUE时，它将返回TRUE。

```
SELECT * FROM employees WHERE salary > 5000 AND (dept_id = 1 OR dept_id = 5);
```

IN运算符

用于检查一组值中是否存在特定值。

```
SELECT column_list FROM table_name WHERE column_name IN (value1, value1,...);
```

BETWEEN 运算符

指定要测试的范围。

```
SELECT * FROM employees WHERE salary BETWEEN 7000 AND 9000;
```

对结果集排序

用于按升序或降序对查询返回的数据进行排序。

```
SELECT column_list FROM table_name ORDER BY column_name ASC|DESC;
```

OP语法

TOP子句用于限制返回的行数。

```
SELECT TOP number|percent column_list FROM table_name;
```

MySQL LIMIT语法

MySQL的LIMIT子句与SQL TOP子句具有相同的作用。当指定了两个参数时，第一个参数指定要返回的第一行的偏移量，即起点，而第二个参数指定要返回的最大行数。初始行的偏移量是0（不是1）。其基本语法为：

```
SELECT column_list FROM table_name LIMIT number;
```

DISTINCT 子句

用于从结果集中删除重复的行：

```
SELECT DISTINCT column_list FROM table_name;
```

更新数据

UPDATE语句用于更新表中的现有数据。

```
UPDATE table_name SET column1_name = value1, column2_name = value2,...WHERE condition;
```

删除数据

DELETE语句用于从表中删除一个或多个行。

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

删除所有数据

如果未在WHERE语句中指定子句，则DELETE表中的所有行都将被删除。但是，目标表本身不会被删除，这意味着表结构，属性和索引将保持不变，但是表中的数据会被清空。
以下语句将删除人员（persons）表中的所有记录：

```
DELETE FROM persons;
```

清空表

TRUNCATE TABLE语句从表中删除所有行，但表结构及其列，约束，索引等保持不变。要删除表及其数据，可以使用该DROP TABLE语句。

```
TRUNCATE TABLE table_name;
```

数据库中删除表

DROP TABLE删除一个或多个表。

```
DROP TABLE table1_name, table2_name, ...;
```

删除数据库

从数据库服务器中永久删除演示数据库。

```
DROP DATABASE demo;
```