

## SQL

注：由于一些历史原因，SQL可以按字母读，亦可以读作 sequal.

### Database

database 是一个存储数据的程序。它（们）擅长数据的存储、检索等基本功能

database 有很多种类。此处我们讨论 rational database，一种具有行和列（长得像 excel）的数据表  
几乎每张表都有 id 列。我们可以通过 id 的映射关系将两张表建立联系。

### SQL

SQL几乎是与数据库程序交流的最流行的方式。

以下是一些 SQL 中常见的命令。

1) CREATE TABLE <name> (<column name> <data type>, ...)

例如 CREATE TABLE user (id INTEGER PRIMARY KEY, name TEXT)

主键：表示该列是标示某行的主要方式。

2) INSERT INTO <name> VALUES (\*datas)

\*datas 的项数和顺序必须与列匹配，否则可能会报错。

当然也可以在 VALUES 前加上诸如 (id, name) 之类的列表，其余部分则填充默认值。

3) SELECT

I. SELECT \* FROM <name> ORDER BY <column> (DESC)

将数据按 column 列的值的大小排序。

II. SELECT <column>, ...

指定选择输出的列，若为 \* 则输出所有列。

III. SELECT ... WHERE <cond1> (AND <cond2> ...) ...

为选择指定条件。

IV. SELECT Aggregate functions(<column>)

包括 MIN, MAX, COUNT 等 Excel 常见函数，相当于另一种筛选的方式。

（得到符合条件 / 计算完毕的那一行）

V. SELECT ... GROUP BY <column>

按 column 的值进行分类。

上述操作的语法顺序是 SELECT → FROM → WHERE → GROUP BY → ORDER BY → LIMIT

## VI. JOIN

SELECT <column> FROM <table1> JOIN <table 2> ON <cond>

将 tb1 中 满足 <cond> 条件 的 数据 合并 至 tb2 中, 得到 合并 后 的 db (non-destructive)  
此处的 <column> 可来自 不同 的 tables.

将 JOIN 改为 LEFT OUTER JOIN 可使 db1 中 所有 项目 出现在 结果 中

即使 它们 可能 不 满足 <cond>, 空缺 的 数据 一律 填充 为 NULL.

4) UPDATE <name> SET <column> = ... WHERE id = ...

更新 表 的 内容.

亦可 替换 为 其他 筛选 条件.

5) DELETE FROM <name> WHERE <cond>

删除 符合 条件 的 数据. 最好 根据 主键 以 免 误伤.