**Database**

**数据库（database）种类**：MySQL、SQL Server、Access、Oracle、Sybase、DB2 等等

**SQL 是什么？**

1. SQL，指**结构化查询语言，全称是 Structured Query Language**。

2. SQL 让您可以访问和处理数据库。

3. SQL 是一种 ANSI（美国国家标准化组织）标准的计算机语言。

**SQL 能做什么？**

1. SQL 面向数据库**执行查询**

2. SQL 可从数据库**取回数据 ->查找**

3. SQL 可在数据库中**插入新的记录 ->增加**

4.SQL 可**更新数据库中的数据 ->修改**

5.SQL 可从数据库**删除记录 ->删除**

6.SQL 可**创建新数据库**

7.SQL 可在数据库中**创建新表**

8.SQL 可在数据库中创建存储过程

9.SQL 可在数据库中创建视图

10.SQL 可以设置表、存储过程和视图的权限

**首先明白几个概念**

1. **数据库（database）**：保存有组织的数据的容器（通常是一个文件或一组文件）
2. **表（table）**某种特定类型数据的结构化清单。（存储在表中的数据是同一种类型的数据或清单）
3. **列（column）**表中的一个字段。所有表都是由一个或多个列组成的。
4. **行（row）**表中的一个记录
5. **主键（primary key）**一列（或一组列），其值能够唯一标识表中每一行。（可以创建表时直接设置，也可以创建好再设置）

**注意：**

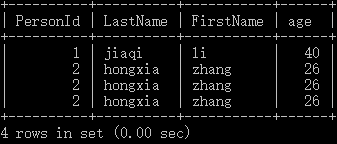
1. 相同数据库中不能两次使用相同的表名，但在不同的数据库中完全可以使用相同的表名
2. 行中应该总是定义主键；若不设置主键，就可以重复添加相同的数据；

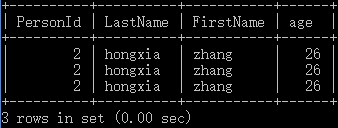
**简单创建个数据库，通顺下过程(SQL不区分大小写)**

1. CREATE DATABASE myDatabase; //创建一个名为 " myDatabase " 的数据库
2. use myDatabase; //选择SQL模式中myDatabase数据库
3. create table Persons( //创建一个表Persons（标注其字段名及类型）
4. PersonId int,
5. LastName varchar(255),
6. FirstName varchar(255),
7. age int
8. );
9. insert into Persons (PersonId,LastName,FirstName,age) values (1, 'jiaqi', 'li', 24);
10. insert into Persons (PersonId,LastName,FirstName,age) values (2, 'hongxia', 'zhang', 26); //插入两条数据values是对应值（增加）
11. select \* from Persons; //从Persons表中取数据\*代表所有（查找）
12. update Persons set age='40',PersonId='1' where FirstName='li';
13. //修改FirstName=‘li‘的行，age改为40，PersonId改为1（修改）
14. select \* from Persons;
15. delete from Persons where age='40'; //删除age=‘40’的行(删除)
16. select \* from Persons;
17. O(∩\_∩)

实际运行一次吧，这样能加深下数据库的作用！

11行运行后的表：****

14行运行后的表：

16行运行后的表：

**主键PRIMARY KEY**

1.创建表时就已经设置主键

CREATE TABLE Persons

(

PersonId int NOT NULL,

LastName varchar(255) NOT NULL,

FirstName varchar(255),

age int,

PRIMARY KEY (PersonId) //设置主键为PersonId

)

2.创建表时未设置主键

CREATE TABLE Persons

(

PersonId int NOT NULL,

LastName varchar(255) NOT NULL,

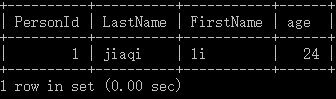
FirstName varchar(255),

age int

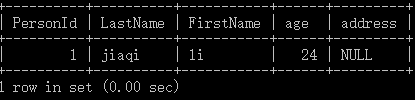
)

ALTER TABLE Persons ADD PRIMARY KEY (PersonId) //设置主键（修改队列）

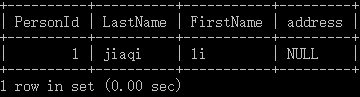
**1.增加列ALTER（字段）** //之前操作（增删改查）其实是对行操作ALTER TABLE 语句用于在已有的表中添加、删除或修改列。

（1）初始表Persons：

（2）ALTER TABLE Persons ADD address varchar(255); //执行后增加字段address



（3）ALTER TABLE Persons DROP COLUMN age; //执行后删除字段address



（4）ALTER TABLE Persons MODIFY COLUMN address varchar(50)；

//执行后修改字段address类型为varchar(50)

**2.删除DROP 语句**

DROP 可以轻松地删除索引、表和数据库。

DROP TABLE Persons; //删除表

TRUNCATE TABLE Persons; //仅仅删除表内数据，不删除表本身

DROP DATABASE myDatabase; //删除数据库

**Important select语句检索**

1. **Select distinct school from persons; // distinct关键字：返回school不同的值**
2. **Select \* from persons; //表中所有数据**
3. **Select school from persons limit 2; //检索返回数据不多于5行**
4. **Select school from persons limit 2,2; //检索返回数据从行2开始，不多于2行**
5. **select persons.school from persons;//使用限定的字段，表名**
6. **select school,city from persons order by city;//按照城市名排序—默认asc**
7. **select school,city from persons order by city desc;//按照城市名排序—降序**
8. **select school,city from persons where school=’shandongdaxue’;//过滤数据**
9. **order by 与 where 同时使用，要把order by 放在where之后！**
10. **过滤数据的组合使用，多条件过滤！**
11. **Select city from persons where city like ‘nan%’; //通配符，找到city字段以nan开头的数据**
12. **多个表检索。。。。。。**

**创建表字段的几种约束：**

**NOT NULL 无值**

**UNIQUE 唯一标示**

**PRIMARY KEY 主键**

**FOREIGN KEY 外键**

**CHECK 用于限制列中的值的范围**

**DEFAULT 默认值**

**每个表可以有多个 UNIQUE 约束，但是每个表只能有一个 PRIMARY KEY 约束**

1. **SQL join 用于根据两个或多个表中的列之间的关系，从这些表中查询数据**
2. **SQL INNER JOIN 关键字，在表中存在至少一个匹配时，INNER JOIN 关键字返回行。与 JOIN 是相同的。**
3. **LEFT JOIN 关键字会从左表 (table\_name1) 那里返回所有的行，即使在右表 (table\_name2) 中没有匹配的行。**
4. **RIGHT JOIN 关键字会右表 (table\_name2) 那里返回所有的行，即使在左表 (table\_name1) 中没有匹配的行。**
5. **FULL JOIN 关键字只要其中某个表存在匹配就会返回行。**
6. **UNION 操作符用于合并两个或多个 SELECT 语句的结果集。UNION 内部的 SELECT 语句必须拥有相同数量的列。列也必须拥有相似的数据类型。同时，每条 SELECT 语句中的列的顺序必须相同。UNION 操作符选取不同的值。如果允许重复的值，请使用 UNION ALL。**
7. **SELECT INTO 语句从一个表中选取数据，然后把数据插入另一个表中。SELECT INTO 语句常用于创建表的备份复件或者用于对记录进行存档。**

**Where是用来约束查找结果的条件**

**like是选取类似的结果，通配符等**