**sql语言** Structured Query Language, 结构化查询语言，对数据库进行操作

数据库是文件系统，使用标准sql对数据库进行操作

标准sql:在mysql里面使用语句，在oracle、db2都可以使用这个语句

sql是非过程性语言，不需要依赖于其他的条件就可以运行

sql的分类

1.DDL (数据定义语言) 创建数据库，创建数据库表 常用create

2.DML (数据操纵语言) 对表中的记录进行增加 修改 删除操作 常用insert update delete

3.DCL (数据控制语言) 数据库的编程的语言

4.DQL (数据查询语言) 对表中的进行查询的操作 常用select

**使用sql对数据库操作**

1.**连接数据库** **mysql -u root -p**

2.**创建数据库** **create database** 数据库的名称**;**

3.**查看所有的数据库** **show databases;**

4.**删除数据库** **drop database** 要删除的数据库的名称**;**

5.**切换数据库** **use** 数据库名称;

**使用sql对数据库表操作** 如果想要创建一个数据库表，这个表要在一个数据库里面，所以需要切换到数据库

**1.创建数据库表**

**create table 表名称 (**

**字段 类型,**

**字段 类型**

**);**

**2.查看表结构 desc** 表名称**;**

**3.删除表 drop table** 要删除表的名称**;**

**4.查看当前的数据库里面有哪些表 show tables;**

**5.增加表中的字段 alter table 表名称 add 字段 类型;**

**mysql的数据类型**

**字符串型:** **VARCHAR、CHAR** 当创建表时候，使用字符串类型，需要指定数据的长度 name varchar(40)，

varchar和char的**区别**:

varchar的长度是可变的， 比如 name varchar(5),存值 a,会直接把a存进去

char的长度是固定的， 比如 name char(5),存值 b，把b存进去,后面会加空格补齐长度

**大数据类型:** BLOB、TEXT (可以存储文件，一般开发，不会直接把文件存到数据库里面，存文件的路径)

**数值型:TINYINT**(byte) 、**SMALLINT**(short)、**INT**(int)、**BIGINT**(long)、**FLOAT**(float)、**DOUBLE**(double)

**逻辑性:BIT**(boolean)

**日期型:DATE:**表示日期 **TIME**:表示时间

**日期和时间:DATETIME**:手动添加到数据表里面 **TIMESTAMP**:自动添加到表里面

**mysql的约束**

**1.非空约束** **not null** 数据不能为空 username varchar(40) not null,

**2.唯一性约束** **unique** 表中的记录不能重复的

**3.主键约束** **primary key** 表示非空并且唯一性 id int primary key

可以使用自动增长 **auto\_increment**  id int primary key atuo\_increment 之后添加的时候添加为null 会自动增长

**使用sql对表中的记录进行操作**（增删改查操作）

**1.向表里面添加记录** **insert into** 要添加的表名称 **values(**要添加的值**);**

注意:当添加的字段的数据类型是int类型，直接写值,添加的字段的类型是varchar类型和日期类型，使用单引号把值包起来

**2.修改表里面的记录** **update** 表名称 **set** 要修改的字段的名称1=修改的值1,要修改的字段的名称2=修改的值2 **where** 条件

**3.删除表里面的记录** **delete from** 表名称 **where** 条件如果不添加where条件，会把表里面的所有的记录都删除

**4.查询表中的记录** **select** 要查询的字段的名称 或者 **\***(代表查询所有) **from** 表名称 **where** 条件

查询表里面的所有的数据 select \* from 表名称;

**where子句**

**1.运算符 < > >= <=** select \* from user where chinese > 60;

**2.in：在范围内** select \* from user where english in (80,90);

**3.and：**在where里面如果有多个条件，表示多个条件同时满足 select \* from user where chinese=100 and english=30;

**4.得到区间范围的值** 1.使用**>=和and**: 2.使用**between and**

查询user表里面语文成绩在70-100之间的值

select \* from user where chinese >=70 and chinese <=100;

select \* from user where chinese between 70 and 100;

**5.like：**模糊查询 比如:查询user表里面username包含a的人员信息:select \* from user where username like '%a%';

**别名** 在要使用别名的字段后面 **as** 别名 比如:select username as u1,chinese as c1 from user;

**查看当前的运行的数据库 select database();**

**去除表里面重复记录** **select distinct \* from** 表名**;**

**对表中查询的记录排序**  **order by**(要写在select语句的最后)

**1.升序** **order by** 要排序字段 **asc** （asc可以省略，默认的情况下就是升序）

**2.降序** **order by** 要排序字段 **desc**

**聚集函数** 使用提供一些函数，直接实现某些功能

**1.count()函数** **根据查询的结果，统计记录数**  **select count(\*) from ...where....**

**2.sum()函数** **求和**  **select sum(**要进行求和字段**) from ...where...**.

**3.avg()函数** **计算的平均数** **select avg(**要计算平均数的字段名称**) from ...**

**4.max()函数** **计算最大值**  **select max(**字段**) from...**

**5.min()函数** **计算最小值** **select min(**字段**) from...**

**分组操作 group by** 根据分组的字段

比如select name,sum(price) from orders group by name;

在分组的基础之上再进行条件的判断, **where里面不能写聚集函数,需要用having，**在having后面可以写聚集函数

比如:对商品进行分类，得到每类商品的总价格大于66的商品

select name,sum(price) from orders **group by name having sum(price)>66;**

**mysql的关键字limit** 查询表中的某几条记录

注意:limit关键字不是标准sql的关键字，只能在mysql数据库里面使用，实现分页的功能

在oracle里面特有关键字 rownum, 在sqlserver里面特有关键字 top

使用limit查询前几条记录 **select \* from** 表名称 **limit** 前几条**;**

使用limit查询第几条到第几条记录 **select \* from** 表名称 **limit** 开始的记录数的位置,向后多少条; **(记录数从0开始)**

**mysql数据库的备份: mysqldump数据导出命令**

**数据导出:**使用命令: **mysqldump –u root -p 数据库名称 > 路径:备份文件名称.sql**

比如 mysqldump –u root –p user > d:\user.sql

**在数据导出的时候必须使用 mysqldump命令(如果用可视化工具导出,兼容性可能不好)**

**数据导入: source 路径:\备份的文件名称.sql** source d:\user.sql

**在数据导入的时候需要进入到数据库里面**