创建数据库:  
 create database 数据库名称; 使用的字符集默认是utf8  
 create database 数据库名称: character set 字符集;  
 查看数据库mysql中所有的数据库  
 show database ;  
 查看某个数据库的定义的信息  
 show create database 数据名称;  
 删除数据库(重点)  
 drop database 数据库名称  
 显示正在使用的数据库  
 select database ();  
 使用数据库,或切换数据库  
 use 数据库名称  
  
 创建数据库的表格  
 create table 表名(  
 字段1(列名) 数据类型(长度) [约束], 约束: 主键约束primary key(飞空,不能重复) auto\_increment(自动增长)  
  
 字段2(列名) 数据类型(长度) [约束], 约束: 非空约束 not null 数据不能为null  
 ...  
 字段n(列名) 数据类型(长度) [约束] 约束: 唯一约束 unique 数据唯一不能重复  
 );  
 [constraint 主键名称] 约束: 默认约束 default  
  
  
 查看数据库中所有的表  
 show tables;  
  
 查看表结构  
 desc 表名;  
  
 删除表  
 drop table 表名;  
  
 修改表结构:  
 ① alter table 表名 add 字段名 数据类型(长度) [约束] 这式添加列  
 ② alter table 表名 modify 字段名 数据类型(长度) [约束] 这式修改列的数据类型[约束]  
 ③ alter table 表名 drop 字段名 ; 这式删除这以列  
 drop index 唯一约束字段名称 这式删除唯一约束  
 drop primary key 这式删除主键约束  
 ④ rename table 旧表名 to 新表名; 这式修改表名字  
 ⑤ alter table 表名 character set 字符集 这式修改表的字符集  
  
  
 修改表数据:  
 ①insert into 表名(字段1,字段2...) values(值1,值2...); 添加数据  
 ②update 表名 set 字段名 = 字段值... where 条件 修改指定的值  
 ③delete from 表名 where 条件; 删除指定的数据  
  
  
 格式二:删除表中的所有数据,但是不会重置主键自增;  
 添加新数据的时候会继续使用之前的主键  
 delete from 表名;  
 格式三:删除表中的所有数据,会重置主键自增;  
 添加新数据的时候会从1开始使用主键  
 truncate table 表名;  
  
  
 查询表中的数据:  
  
 select 字段名|\* from 表名 ; 查询表中的数据  
 select distinct 字段名|\* from 表名 查询表中去除重复的数据  
 select 字段名(int)10000 from 表名  
 select 字段|\* from 表名 where 条件 条件查询  
  
 使用between...and 优化代码  
 1.对时间区间进行查询 '2018-01-01'--'2018-08-15'  
 2.小的值要写在前边,大的值写后边  
 select \* from 表名 where 字段名 between 1000 and 2002; 之间的  
 select \* from 表名 where 字段名 数据 or 数据 或者  
 select \* from 表名 where 字段名 in(数据,数据,数据...) 或者  
  
 模糊运算符  
 like  
 %:多个任意字符  
 \_:一个任意字符  
 select \* from 表名 where 字段名 like '\_\_\_'  
  
 排序查询语句  
 格式:  
 select 字段|\* from 表名 [where 条件]  
 order by 字段名 [ASC]|DESC  
 ASC:升序,不写默认  
 DESC:降序  
 注意:  
 order by语句一定写在sql语句的末尾  
  
 聚合查询语句:对列进行查询,结果会返回一个单一的值,会忽略null值  
  count：统计指定列不为NULL的记录行数；  
 sum：计算指定列的数值和，如果指定列类型不是数值类型，那么计算结果为0；  
 max：计算指定列的最大值，如果指定列是字符串类型，那么使用字符串排序运算；  
 min：计算指定列的最小值，  
 如果指定列是字符串类型，那么使用字符串排序运算；ASKM表的值  
 avg：计算指定列的平均值，如果指定列类型不是数值类型，那么计算结果为0；  
  
 格式:  
 select count(\*|字段),sum(字段),avg(字段),max(字段),min(字段) from 表名 [where 条件];  
 到括号之间不能有空格  
  
  
  
 分组查询语句  
 格式:  
 select 被分组的字段名,查询条件 from 表名 [where 条件]  
 group by 被分组的字段 [having 分组之后条件过滤]  
 order by 排序字段 [ASC]|DESC;  
 注意:  
 被分组的字段,一般都会写在select的后边,方便查看结果  
 where过滤-->分组-->查询-->having对分组后数据过滤-->排序  
  
  
 where:对分组之前的数据过滤  
 having:对分组之后的数据过滤  
  
  
 分页查询  
 格式一:  
 select 字段|\* from 表名 limit m;  
 只查询前m条数据  
 格式二:  
 select 字段|\* from 表名 limit m,n;  
 m:每页开始的行索引,是变化的  
 n:每页的显示数量,是不变的  
 假如每页显示5条数据,查询出第2页的数据  
 SELECT \* FROM product LIMIT 10,5;  
 注意:  
 数据库行的索引从0开始,列的索引从1开始  
  
  
 添加外键约束:  
 作用:保证数据的准确和完整  
 主表有的数据,从表可以有,可以没有  
 主表没有的数据,从表也不能有  
 删除主表的数据,必须保证从表没有使用  
 格式:修改表结构,添加字段  
 alter table 从表名 add [constraint 外键名] foreign key(从表的外键字段) references 主表(主键);  
  
 多对多关系  
 建表原则:创建一张中间表,使用两个主表的主键作为外键  
 主表:商品表products,订单表orders  
 中间表:products\_orders

多表查询:一次性查询两张以上的表  
1.交叉连接查询: 很少用,有错误的数据  
格式: select \* from 表A,表B;  
  
2.内连接查询:使用主键和外键之间的约束作为查询条件  
 a.隐式内连接查询:不使用关键字[inner] join on  
 select \* from 表A,表B where 表A.主键 = 表B.外键;  
 b.显示内连接查询:使用关键字[inner] join on  
 select \* from 表A[inner] join 表B on 表A.主键 = 表B.外键  
  
3.外连接查询:使用主键和外键之间的约束作为查询条件  
 a.左外连接查询: 使用关键字 left [outer] join on  
 select \* from 表A left [outer] join 表B on 表A.主键 = 表B.外键  
 注意:左外键连接以左边的表A为主,左边有的数据,右变没右要用null代替  
 左边没右的数据,右边也不能出现  
 b.右外连接查询: 使用关键字 right [outer] join on  
 和a 方法相反  
  
  
4.子查询: sql语句的嵌套  
  
 a.一条sql语句的查询结果,作为另外一条sql语句的查询条件  
 格式: select \* from 表B where 字段 = (select 字段 from 表A where 条件);  
  
 b.一条sql语句的查询结果,作为另一条sql语句的另一张表  
 格式: select \* from (select \* from 表A where 条件) 表A, 表B where 表A.主键 = 表B.外键;