聚合函数

- count() 根据某个字段来统计总记录数

```
select count(id) from student;
```

- sum()计算某个字段值的总和

```
select sum(score) from student;
```

- avg()计算某个字段值的总和的平均值

```
select avg(score) from student;
```

- max()计算某个字段值的最大值

```
select max(score) from student;
```

- min()计算某个字段值的最小值

```
select min(score) from student;
```

分组：按照某个字段对查询结果进行分组。

```
select name,count(id) from student group by name;
```

```
select name,count(id) from student group by name having count(id) > 1;
```

```
select name,count(id) from student group by name order by count(id)
asc/desc;
```

MySQL运算符：

- 算术运算符：
  - 执行运算符：加减乘除

    ```
    select score+10 from student;
    ```

  - 比较运算符：大于，大于等于，等于，小于，小于等于，不等于，结果0/1，0表示false，1表示true

```
select score = 100 from student;
```

- 逻辑运算符：与，或，非，结果0/1，0表示false，1表示true

```
select !score > 60 from student;
```

```
select score > 60 && age < 20 from student;
```

```
select score > 60 || age < 20 from student;
```

- 特殊运算符
  - is null 判断值是否为空，结果0/1，0表示false，1表示true

```
select score is null from student;
```

  - between and 判断值是否在某个区间之内，包含两个边界值。

```
select age between 18 and 20 from student;
```

```
select age >= 18 && age <= 20 from student;
```

```
select age >= 18 and age <= 20 from student;
```

  - in 判断值是否在某个确定的集合内

```
select score,name,age from student where age in (18,19,20);
```

```
select score,name,age from student where age = 18 || age = 19 ||
age = 20;
```

```
select score,name,age from student where age = 18 or age = 19 or
age = 20;
```

  - like 模糊查询

    以"张"开头

```
select * from student where name like "张%";
```

```
select * from student where name like "%张";
```

```
select * from student where name like "%张%";
```

```
select * from student where name like "__";
```

```
select * from student where name like "张__";
```

```
select * from student where name like "_张_";
```

```
select * from student where name like "__张";
```

表设计

主键：用来标识一个字段，一旦某个字段被设置为主键，该字段的值就是每一行记录的唯一标识。

每个人的身份证号都是唯一的，可以通过身份证号来确定一个人，不会有重复，数据库中区分不同的数据，通过主键来区分，默认情况下，每张表都应该有一个主键。一张表只能有一个主键，所谓的一张表多个主键，指的是联合主键，用多个字段一起作为一张表的注解。

```
create table student(
  id int primary key auto_increment,
  name varchar(11),
  score double,
  age int
);
```

主键生成策略：代理主键，主键所对应的字段与业务无关，仅仅是用来标识一行数据。一般定义为int类型，一方面是因为int类型存储空间较小，另外一方面是因为int类型可以设置自增。

外键：用来标识一个字段，一旦一张数据表中添加了外键，就意味着这张表要被另外一张表所约束，被约束的表（外键所在的表）叫做从表，约束别人的表就叫做主表，通过建立从表的外键与主表的主键的约束关系来形成两张表之间的约束关系。

```
alter table student add foreign key(cid) references class(id);
```

```
create table student(
  id int primary key auto_increment,
  name varchar(11),
  score double,
  age int,
  cid int,
  foreign key (cid) references class(id)
);
```

```
public class Student{
    private int id;
    private String name;
    private double score;
    private int age;
    private Classes classes;
}
```

```
public class Classes{
    private int id;
    private String name;
    private List<Student> students;
}
```

```
Student student = new Student(1,"张三");
Student student2 = new Student(2,"李四");
Student student3 = new Student(3,"王五");
Classes classes = new Classes(1,"Java1班");
Classes classes2 = new Classes(2,"Java2班");
student.setClasses(classes);
student2.setClasses(classes);
student3.setClasses(classes2);

List<Student> list = new ArrayList<Student>();
list.add(student);
list.add(student2);
classes.setStudents(list);
List<Student> list2 = new ArrayList<Student>();
list2.add(student3);
classes2.setStudents(list2);
```