

一、主键与外键的关系：

关系型数据库中的一条记录中有若干个属性，若其中某一个属性组(注意是组)能唯一标识一条记录，该属性组就可以成为一个主键。但外键不是唯一的。

比如：

- 1、学生表(学号, 姓名, 性别, 班级)；其中每个学生的学号是唯一的，学号就是一个主键。
- 2、课程表(课程编号, 课程名, 学分)；其中课程编号是唯一的, 课程编号就是一个主键。
- 3、成绩表(学号, 课程号, 成绩)；成绩表中单一一个属性无法唯一标识一条记录，学号和课程号的组合才可以唯一标识一条记录，所以学号和课程号的属性组是一个主键。

注：成绩表中的学号不是成绩表的主键，但它和学生表中的学号相对应，并且学生表中的学号是学生表的主键，则称成绩表中的学号是学生表的外键。同理成绩表中的课程号是课程表的外键。

定义主键和外键主要是为了维护关系数据库的完整性，总结一下：

- 1、主键是能确定一条记录的唯一标识，比如，一条记录包括身份证号，姓名，年龄。身份证号是唯一能确定你这个人的，其他都可能有重复，所以身份证号是主键。
- 2、外键用于与另一张表的关联，是能确定另一张表记录的字段，用于保持数据的一致性。比如，A 表中的一个字段，是 B 表的主键，那他就可以是 A 表的外键。

主键在物理层面上只有两个用途：

- 1、惟一地标识一行。
- 2、作为一个可以被外键有效引用的对象。

二、外键约束

1、概念：在正常项目中很多时候我们必须要进行拆表，将数据分别存放在多张表中，以减少冗余数据。但是拆分出来的表与表之间是有着关联关系的，我们必须得通过一种约束来约定表与表之间的关系，这种约束就是外键约束。

2、作用：外键约束是保证一个或两个表之间的参照完整性, 外键是构建于一个表的两个字段或是两个表的两个字段之间的参照关系。

3、总结来说就是 a 表的主键，在 b 表里面也存在，并且要保证必须一致，这就是外键约束。

单表查询

1、查询某张表所有内容 & 特定列的记录

select * from 表名

```
17
18 select * from product;
19
```

信息	结果 1	剖析	状态
pid	pname	price	num
1	苹果电脑	18000	10
2	华为5G手机	30000	20
3	小米手机	1800	30
4	iPhoneX	8000	10
5	iPhone7	6000	200
6	iPhone6s	4000	1000
7	iPhone6	3500	100
8	iPhone5s	3000	100

select 列名, 列名, 列名... from 表名

```
17
20 select pid,pname,num from product
21
```

信息	结果 1	剖析	状态
pid	pname	num	
1	苹果电脑	10	
2	华为5G手机	20	
3	小米手机	30	
4	iPhoneX	10	
5	iPhone7	200	
6	iPhone6s	1000	
7	iPhone6	100	
8	iPhone5s	100	

2、去重查询 distinct

select distinct 字段名 from 表名; 注：要数据一模一样才能去重

```
21
22 select DISTINCT num from product
23
```

信息	结果 1	剖析	状态
num			
10			
20			
30			
200			
1000			
100			

3、别名查询

select 列名 as 别名 , 列名 from 表

列别名 as 可以不写

23	
24	<code>select pid as '序号',pname as '名字',price as '价格',num as '数量' from product</code>

信息	结果 1	剖析	状态
序号	名字	价格	数量
1	苹果电脑	18000	10
2	华为5G手机	30000	20
3	小米手机	1800	30
4	iphonex	8000	10
5	iPhone7	6000	200
6	iPhone6s	4000	1000
7	iPhone6	3500	100
8	iPhone5s	3000	100

4、运算查询（+，-，*，/，%等），不会影响原表中的结果

Price/10 前后的对比

信息	结果 1	剖析	状态
pid	pname	price	num
1	苹果电脑	18000	10
2	华为5G手机	30000	20
3	小米手机	1800	30
4	iphonex	8000	10
5	iPhone7	6000	200
6	iPhone6s	4000	1000
7	iPhone6	3500	100
8	iPhone5s	3000	100

25	
26	<code>select pid,pname,price/10,num from product</code>

信息	结果 1	剖析	状态
pid	pname	price/10	num
1	苹果电脑	1800	10
2	华为5G手机	3000	20
3	小米手机	180	30
4	iphonex	800	10
5	iPhone7	600	200
6	iPhone6s	400	1000
7	iPhone6	350	100
8	iPhone5s	300	100

运算查询字段，字段之间是可以的；字符串等类型可以做运算查询，但结果没有意义

5、条件查询

1、比较运算符

(1)、大于：>，

34	<code># 大于：></code>
35	<code>select * from product where price > 3000;</code>
36	

信息	结果 1	剖析	状态
pid	pname	price	num
1	苹果电脑	18000	10
2	华为5G手机	30000	20
4	iphonex	8000	10
5	iPhone7	6000	200
6	iPhone6s	4000	1000
7	iPhone6	3500	100

(2)、小于：<，

```

37 # 小于: <
38 select * from product where price < 3000;
39

```

信息	结果 1	剖析	状态
pid	pname	price	num
3	小米手机	1800	30

(3) 大于等于: >=,

```

39
40 # 大于等于: >=
41 select * from product where price >= 3000;
42

```

信息	结果 1	剖析	状态
pid	pname	price	num
1	苹果电脑	18000	10
2	华为5G手机	30000	20
4	iPhoneX	8000	10
5	iPhone7	6000	200
6	iPhone6s	4000	1000
7	iPhone6	3500	100
8	iPhone5s	3000	100

(4) 小于等于: <=,

```

42
43 # 小于等于: <=
44 select * from product where price <= 3000;
45

```

信息	结果 1	剖析	状态
pid	pname	price	num
3	小米手机	1800	30
8	iPhone5s	3000	100

(5) 等于: = 不能用 null 判断,

```

45
46 # 等于: = 不能用于null判断
47 select * from product where pid = 1;
48

```

信息	结果 1	剖析	状态
pid	pname	price	num
1	苹果电脑	18000	10

(6) 、不等于: !=或<>,

```

48
49 # 不等于: != 或 <>
50 select * from product where pid <> 1;
51

```

信息	结果 1	结果 2	剖析	状态
	pid	pname	price	num
▶	2	华为5G手机	30000	20
	3	小米手机	1800	30
	4	iPhoneX	8000	10
	5	iPhone7	6000	200
	6	iPhone6s	4000	1000
	7	iPhone6	3500	100
	8	iPhone5s	3000	100

(7) 安全等于: <=> 可以用 null 值判断

```
51
52 # 安全等于: <=> 可以用于null值判断
53 select * from product where pid <=> 1;
```

信息	结果 1	剖析	状态	
	pid	pname	price	num
▶	1	苹果电脑	18000	10

2、逻辑运算符（建议用单词，可读性来说）

(1)、and 与

```
56 -- 价格 大于3000 并且 数量 大于20
57 select *from product where price > 3000 and num > 20;
58
```

信息	结果 1	剖析	状态
----	------	----	----

pid	pname	price	num
5	iPhone7	6000	200
6	iPhone6s	4000	1000
7	iPhone6	3500	100

(2)、or 或

```

59 -- 价格 大于3000 或者 数量 大于30
60 select * from product;
61 select * from product where price > 3000 or num > 30;

```

信息	结果 1	剖析	状态
pid	pname	price	num
1	苹果电脑	18000	10
2	华为5G手机	30000	20
4	iPhonex	8000	10
5	iPhone7	6000	200
6	iPhone6s	4000	1000
7	iPhone6	3500	100
8	iPhone5s	3000	100

(3)、not 非 (! =)

```

62
63 select * from product where price != 3000;

```

信息	结果 1	剖析	状态
pid	pname	price	num
1	苹果电脑	18000	10
2	华为5G手机	30000	20
3	小米手机	1800	30
4	iPhonex	8000	10
5	iPhone7	6000	200
6	iPhone6s	4000	1000
7	iPhone6	3500	100

3、范围

(1)、区间范围

between...and....

```

66 --between...and....
67 select * from product where price between 3000 and 6000;

```

信息	结果 1	剖析	状态
pid	pname	price	num
5	iPhone7	6000	200
6	iPhone6s	4000	1000
7	iPhone6	3500	100
8	iPhone5s	3000	100

not between...and....

```
68 --not between...and...
69 select * from product where price not between 3000 and 6000;
```

信息	结果 1	剖析	状态
pid	pname	price	num
1	苹果电脑	18000	10
2	华为5G手机	30000	20
3	小米手机	1800	30
4	iPhonex	8000	10

(2)、集合范围

In(x, x, x)

```
70 --in(x,x,x)
71 select * from product where pid in(6,7,8)
72
```

信息	结果 1	剖析	状态
pid	pname	price	num
6	iPhone6s	4000	1000
7	iPhone6	3500	100
8	iPhone5s	3000	100

Not in(x, x, x)

```
73 --not in(x,x,x)
74 select * from product where pid not in(6,7,8)
```

信息	结果 1	剖析	状态
pid	pname	price	num
1	苹果电脑	18000	10
2	华为5G手机	30000	20
3	小米手机	1800	30
4	iPhonex	8000	10
5	iPhone7	6000	200

4、模糊查询

like 'xxx' 模糊查询是处理字符串的时候进行部分匹配 如果想要表示 0~n 个字符, 用%

(1)、以 ipho 开头的

```
76 --模糊查询 like %
77 select * from product;
78 select * from product where pname like 'iPho%';
79
```

信息	结果 1	剖析	状态
pid	pname	price	num
4	iPhonex	8000	10
5	iPhone7	6000	200
6	iPhone6s	4000	1000
7	iPhone6	3500	100
8	iPhone5s	3000	100

(2)、以 s 结尾的

```
80 select * from product where pname like '%s';
81 select * from product where pname like '%one%';
```

信息	结果 1	剖析	状态
pid	pname	price	num
6	iPhone6s	4000	1000
8	iPhone5s	3000	100

(3)、one 在中间的

```
80 select * from product where pname like '%s';
81 select * from product where pname like '%one%';
```

信息	结果 1	剖析	状态
pid	pname	price	num
4	iPhoneX	8000	10
5	iPhone7	6000	200
6	iPhone6s	4000	1000
7	iPhone6	3500	100
8	iPhone5s	3000	100

5、特殊的 null 值处理

```
81
82 select * from product WHERE num is null
83 select * from product where num <=> null
84
```

信息	结果 1	剖析	状态
pid	pname	price	num
2	华为5G手机	30000	(Null)

```
84
85 select * from product where num is not null
```

信息	结果 1	剖析	状态
pid	pname	price	num
1	苹果电脑	18000	10
3	小米手机	1800	30
4	iPhoneX	8000	10
5	iPhone7	6000	200
6	iPhone6s	4000	1000
7	iPhone6	3500	100
8	iPhone5s	3000	100

6、排序查询

(1)、单例排序 ASC: 升序, 默认值; DESC: 降序
SELECT 字段名 FROM 表名 [WHERE 条件] ORDER BY 字段名 [ASC|DESC];

```
87 select * from product;
88 select * from product order by pid desc
89
```

信息	结果 1	剖析	状态
pid	pname	price	num
8	iPhone5s	3000	100
7	iPhone6	3500	100
6	iPhone6s	4000	1000
5	iPhone7	6000	200
4	IPhonex	8000	10
3	小米手机	1800	30
2	华为5G手机	30000	(Null)
1	苹果电脑	18000	10

```
89
90 select * from product order by num asc
```

信息	结果 1	剖析	状态
pid	pname	price	num
2	华为5G手机	30000	(Null)
1	苹果电脑	18000	10
4	IPhonex	8000	10
3	小米手机	1800	30
7	iPhone6	3500	100
8	iPhone5s	3000	100
5	iPhone7	6000	200
6	iPhone6s	4000	1000

(2)、组合排序

同时对多个字段进行排序, 如果第 1 个字段相等, 则按第 2 个字段排序, 依次类推

```
90
91 select * from product order by price asc,num desc
```

信息	结果 1	剖析	状态
pid	pname	price	num
3	小米手机	1800	30
8	iPhone5s	3000	100
7	iPhone6	3500	100
6	iPhone6s	4000	1000
5	iPhone7	6000	200
4	IPhonex	8000	10
1	苹果电脑	18000	10
2	华为5G手机	30000	(Null)

7、聚合函数

聚合函数通常会和分组查询一起使用，用于统计每组的数据

(1)、max() 求最大值

```
93 select * from product;
94 select max(num) from product;
```

信息	结果 1	剖析	状态
max(num)			
▶	1000		

(2)、min() 求最小值

```
96 select min(num) from product;
```

信息	结果 1	剖析	状态
min(num)			
▶	10		

(3)、avg() 求这一列的平均值

```
97 select avg(price) from product;
```

信息	结果 1	剖析	状态
avg(price)			
▶	9287.5		

(4)、count() 统计这一列有多少条数据

```
98
99 select count(price) from product;
100 select count(*) from product;
```

信息	结果 1	剖析	状态
count(price)			
▶	8		

(5)、sum() 求和

```
101
102 select sum(price) from product;
```

信息	结果 1	剖析	状态
sum(price)			
▶	74300		

7、分页查询

a 表示的是跳过的数据条数, b 表示的是要查询的数据条数

```

101
102 select * from product limit 2,4;

```

信息	结果 1	剖析	状态
pid	pname	price	num
3	小米手机	1800	30
4	iPhoneX	8000	10
5	iPhone7	6000	200
6	iPhone6s	4000	1000

8、分组查询

GROUP BY 将分组字段结果中相同内容作为一组，并且返回每组的第一条数据，所以单独分组没什么用处。分组的目的是为了统计，一般分组会跟聚合函数一起使用。

学生表

100

107 select * from student

108 # 创建学生表(有sid,学生姓名,学生性别,学生年龄,分数列,其中sid为主键自动增长)

109 CREATE TABLE student(

110 sid INT PRIMARY KEY auto increment

信息

结果 1

剖析

状态

sid	sname	sex	age	score	classid
1	zs	男	18	98.5	1
2	ls	女	18	96.5	1
3	ww	男	15	50.5	1
4	zl	女	20	98.5	2
5	tq	男	18	60.5	2
6	wb	男	38	98.5	1
7	小丽	男	18	100	1
8	小红	女	28	28	2
9	小强	男	21	95	3

```

120
129 select * from student
130
131 -- 根据性别分组 统计每一组有多少人
132 select sex, count(sex) from student group by sex;
133

```

信息	结果 1	剖析	状态
sex	count(sex)		
女	3		
男	6		

```

133
134 -- 根据性别分组，统计每组学生的平均分
135 select sex,avg(score) from student group by sex;
136

```

信息	结果 1	剖析	状态
	sex	avg(score)	
▶女	74.33333333333333		
男	83.83333333333333		

```

136
137 -- 根据性别分组，统计每组学生的总分
138 select sex,sum(score) from student group by sex;
139

```

信息	结果 1	剖析	状态
	sex	sum(score)	
▶女	223		
男	503		

分组后筛选 having

```

137
140 -- 分组后，也可以筛选内容 having
141 -- 根据性别分组 统计每一组有多少人 并且 筛选出 每组大于5的数据
142 select sex, count(sex) from student group by sex having count(sex) > 5;
143

```

信息	结果 1	剖析	状态
	sex	count(sex)	
▶男	6		

```

143
144 -- 根据性别分组 年龄大于18，并且组内人数大于4
145 select sex,count(sid) from student where age>18 group by sex having count(sid) > 1;

```

信息	结果 1	剖析	状态
	sex	count(sid)	
▶女	2		
男	2		