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,一条记录包括身份正号,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及特定列的记录

select * from 表名

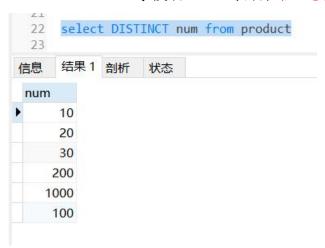
select 列名,列名,列名... from 表名

18 19	8	erec	t * fr	om pr	oduc c	1.	
息	4	吉果 1	剖析	状态			
pid		pnam	ne	pri	ce	num	
	1	苹果申	主脑		18000		10
	2	华为5	G手机		30000		20
	3	小米引	E机		1800		30
	4	iPhor	nex		8000		10
	5	iPhor	ne7		6000	2	00
	6	iPhor	ne6s		4000	10	000
	7	iPhor	ne6		3500	1	00
	8	iPhor	e5s		3000	1	00

20 21	selec	t pid,	pname,	num fro	om produ
信息	结果 1	剖析	状态		
pid	pnam	ie	nun	1	
	1 苹果印	酗		10	
	2 华为5	G手机		20	
	3 小米引	F机		30	
	4 iPhor	nex		10	
	5 iPhor	ne7		200	
	6 iPhor	ne6s		1000	
	7 iPhor	ne6		100	
	8 iPhor	ne5s		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



3、别名查询

select 列名 as 别名,列名 from 表 列别名 as 可以不写

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

om p	10, num fr	oname,price/	pid,	select	26		状态	剖析	结果1	息
		状态	剖析	结果 1	言息		1.32			
	num	price/10	e	pnam	pid	num	(1 sucottores		pnam	pid
	10	1800		1 苹果电	1.	10	18000	围陷	1 苹果电	
	20	3000	The same	2 华为50		20	30000	G手机	2 华为5	
						30	1800	手机	3 小米手	
	30	180	(A) (B)	3 小米手		10	8000	av.	4 iPhon	
	10	800	ex	4 iPhon		957630700	V1.020/7000.00			
	200	600	e7	5 iPhone		200	6000	ne7	5 iPhon	
	1000	400	e6s	6 iPhon		1000	4000	ne6s	5 iPhon	
	100	350	e6	7 iPhon		100	3500	ne6	7 iPhon	
	100	300	e5s	8 iPhone		100	3000	ne5s	B iPhon	

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: >,



(2)、小于: <,



(3) 大于等于: >=,



(4) 小于等于: <=,



(5) 等于: = 不能用 null 判断,



(6)、不等于:!=或⇔,



(7) 安全等于: <=> 可以用 null 值判断



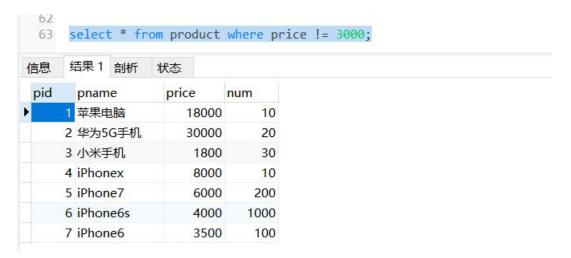
- 2、逻辑运算符(建议用单词,可读性来说)
- (1)、and 与



(2) 、or 或

```
59
     -- 价格 大于3000 或者 数量 大于30
     select * from product;
 60
     select *from product where price > 3000 or num > 30;
     结果1 剖析
信息
                 状态
pid
                   price
      pname
                           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 (! =)



3、范围

(1)、区间范围

between...and....

5 iPhone7

6 iPhone6s

7 iPhone6

8 iPhone5s

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

200

1000

100

100

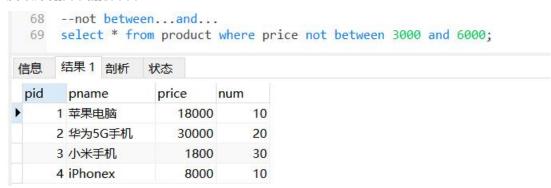
6000

4000

3500

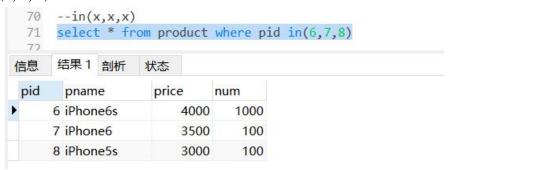
3000

not between...and....

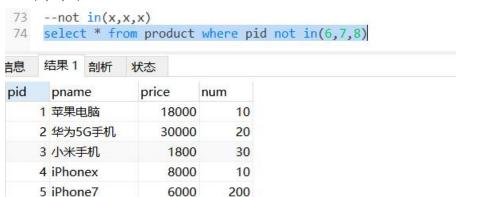


(2)、集合范围

In(x, x, x)

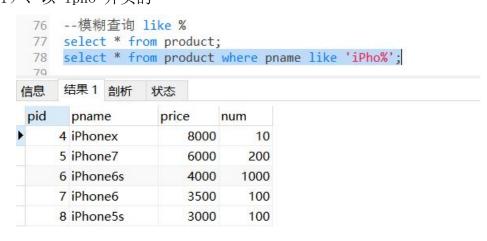


Not in(x, x, x)

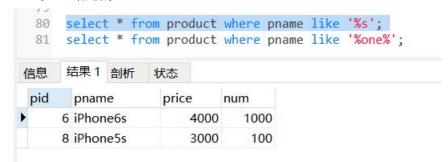


4、模糊查询

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



(2)、以 s 结尾的



(3)、one 在中间的



5、特殊的 null 值处理





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];



(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

信息	结果 1	剖析	状态	
pid	pnam	ne	price	num
>	3 小米引	戶机	18	00 30
	8 iPhor	ne5s	30	00 100
	7 iPhor	ne6	35	00 100
	6 iPhor	ne6s	40	00 1000
	5 iPhor	ne7	60	200
	4 iPhor	nex	80	00 10
	1 苹果申	刨脑	180	00 10
	2 华为5	G手机	300	00 (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 * trom product;
94 select max(num) from product;
            结果1 剖析 状态
      信息
       max(num)
                1000
```

(2) 、min() 求最小值

```
96
     select min(num) from product;
 97
 92
    结果 1 剖析
信息
                状态
min(num)
        10
```

(3)、avg() 求这一列的平均值

```
97
     select avg(price) from product;
 92
    结果 1 剖析 状态
信息
avg(price)
     9287.5
```

(4)、count() 统计这一列有多少条数据

```
98
     select count(price) from product;
     select count(*) from product;
100
101
信息
     结果 1 剖析 状态
count(price)
            8
```

(5)、sum() 求和

```
101
     select sum(price) from product;
102
     结果1 剖析
                 状态
信息
sum(price)
       74300
```

7、分页查询

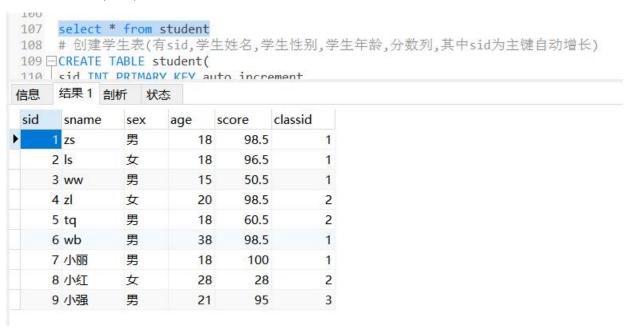
a 表示的是跳过的数据条数,b 表示的是要查询的数据条数



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。分组的目的就是为了统计,一般分组会跟聚合函数一起使用。

学生表









分组后筛选 having



