

MySQL（四）

1. 查询数据

select 是一个功能非常功能的指令，可以从一张或者多张表中查询出各种结构的数据。

1.1 准备数据

```
#创建商品表:
create table product(
    pid int unsigned not null auto_increment comment '编号',
    pname varchar(50) not null comment '商品名称',
    price decimal(10,2) not null comment '商品价格',
    is_on_sale enum('y','n') not null default 'y' comment '是否上架,y: 是,n:
否',
    primary key (pid)
);
#自动增长列: auto_increment, 要求: 1.必须整形 2.必须主键 primary key

# 插入数据
insert into product(pid,pname,price,is_on_sale)
values(1,'联想',5000,'y'),
(2,'海尔',3000,'y'),
(3,'雷神',5000,'y'),
(4,'JACK JONES',800,'y'),
(5,'真维斯',200,'n'),
(6,'花花公子',440,'n'),
(7,'劲霸',2000,'y'),
(8,'香奈儿',800,'n'),
(9,'相宜本草',200,'y'),
(10,'面霸',5,'y'),
(11,'神州',2000,'y'),
(12,'印度神油',363,'y'),
(13,'高级檀香',45,'y');
```

准备完成后，表格如下：



```
mysql> select * from product;
+-----+-----+-----+-----+
| pid | pname      | price  | is_on_sale |
+-----+-----+-----+-----+
| 1   | 联想       | 5000.00 | y          |
| 2   | 海尔       | 3000.00 | y          |
| 3   | 雷神       | 5000.00 | y          |
| 4   | JACK JONES | 800.00  | y          |
| 5   | 真维斯     | 200.00  | n          |
| 6   | 花花公子   | 440.00  | n          |
| 7   | 劲霸       | 2000.00 | y          |
| 8   | 香奈儿     | 800.00  | n          |
| 9   | 相宜本草   | 200.00  | y          |
| 10  | 面霸       | 5.00    | y          |
| 11  | 神州       | 2000.00 | y          |
| 12  | 印度神油   | 363.00  | y          |
| 13  | 高级檀香   | 45.00   | y          |
+-----+-----+-----+-----+
13 rows in set (0.00 sec)
```

1.2 select 的语法

select 是一个非常复杂的指令，可以查询一张表，也可以同时从多张表中查询数据，我们先看来在查询一张表时的语法：

```
select 字段 [as] 别名 ....
from 表名 [as] 别名
[where 条件
group by 字段名
having 过滤条件
order by 字段名 desc|asc
limit 起始记录,记录数]
```

说明：

1. []中的内容是可以省略。
2. |是或者的意思。
3. 如果同时写了多个指令，那么顺序是固定的 select->from->where->group by->order by->having->limit

2. 基本查询

简单查询	SQL 语句
查询所有的商品	select * from product;
查询商品名和商品价格	select pname,price from product;
表别名, select ... from 表名 [as] 别名	select * from product as p
列别名, select 字段 [as] 别名, ... from 表名	select pname as pn from product
去掉重复值	select distinct price from product
查询结果是表达式: 将所有商品的价格+10 元进行显示	select pname,price+10 from product
使用函数	select version();

2.1 查询所有字段信息

```
select * from 表名; # * 表示所有字段
```

select:选择

from: 从..里

说明: 从表中取出所有的字段。

```
mysql> select * from product;
+-----+-----+-----+-----+
| pid | pname      | price  | is_on_sale |
+-----+-----+-----+-----+
| 1   | 联想       | 5000.00 | y          |
| 2   | 海尔       | 3000.00 | y          |
| 3   | 雷神       | 5000.00 | y          |
| 4   | JACK JONES | 800.00  | y          |
| 5   | 真维斯     | 200.00  | n          |
| 6   | 花花公子   | 440.00  | n          |
| 7   | 劲霸       | 2000.00 | y          |
| 8   | 香奈儿     | 800.00  | n          |
| 9   | 相宜本草   | 200.00  | y          |
| 10  | 面霸       | 5.00    | y          |
| 11  | 神州       | 2000.00 | y          |
| 12  | 印度神油   | 363.00  | y          |
| 13  | 高级檀香   | 45.00   | y          |
+-----+-----+-----+-----+
13 rows in set (0.00 sec)
```

2.2 查询指定的字段

```
select 字段 1, 字段 2, ... from 表名;
```

```
mysql> select pname, price from product;
+-----+-----+
| pname      | price |
+-----+-----+
| 联想       | 5000.00 |
| 海尔       | 3000.00 |
| 雷神       | 5000.00 |
| JACK JONES | 800.00  |
| 真维斯     | 200.00  |
| 花花公子   | 440.00  |
| 劲霸       | 2000.00 |
| 香奈儿     | 800.00  |
| 相宜本草   | 200.00  |
| 面霸       | 5.00    |
| 神州       | 2000.00 |
| 印度神油   | 363.00  |
| 高级檀香   | 45.00   |
+-----+-----+
13 rows in set (0.00 sec)
```

2.3 查询表达式

```
select 表达式 from 表名;
```

说明：表达式可以使用加(+)、减(-)、乘(*)、除(/)、求余(%)算术运算符。

可以对字段中的字段进行运算：



```
mysql> select pname,price,price/2,price%7 from product;
+-----+-----+-----+-----+
| pname      | price  | price/2 | price%7 |
+-----+-----+-----+-----+
| 联想       | 5000.00 | 2500.000000 | 2.00 |
| 海尔       | 3000.00 | 1500.000000 | 4.00 |
| 雷神       | 5000.00 | 2500.000000 | 2.00 |
| JACK JONES | 800.00  | 400.000000  | 2.00 |
| 真维斯     | 200.00  | 100.000000  | 4.00 |
| 花花公子   | 440.00  | 220.000000  | 6.00 |
| 劲霸       | 2000.00 | 1000.000000 | 5.00 |
| 香奈儿     | 800.00  | 400.000000  | 2.00 |
| 相宜本草   | 200.00  | 100.000000  | 4.00 |
| 面霸       | 5.00    | 2.500000    | 5.00 |
| 神州       | 2000.00 | 1000.000000 | 5.00 |
| 印度神油   | 363.00  | 181.500000  | 6.00 |
| 高级檀香   | 45.00   | 22.500000   | 3.00 |
+-----+-----+-----+-----+
13 rows in set (0.00 sec)
```

还可以做单纯的数字运算：

```
mysql> select (17*37*12+20)/3;
+-----+
| (17*37*12+20)/3 |
+-----+
| 2522.6667 |
+-----+
1 row in set (0.00 sec)
```

2.4 别名

编辑 | 讨论 | 分享 | 8029

罗纳尔多[巴西足球运动员]

开放分类：人物 | 足球运动员

罗纳尔多·路易斯·纳扎里奥·达·利马（Ronaldo Luiz Nazario De Lima）1976年9月18日出生在巴西里约热内卢，巴西足球运动员，场上司职前锋。青少年时期成名于克鲁塞罗，1996、1997、2002年三度获得世界足球先生，1998年当选世界杯最佳球员，2002年获得世界杯金靴奖，2010年罗纳尔多获得巴西传奇巨星奖。罗纳尔多的欧洲足球生涯始于埃因霍温，此后先后效力过巴塞罗那、国际米兰、皇家马德里、AC米兰四支豪门球队，共拿到1次西甲冠军奖杯，1次欧洲联盟杯、1次欧洲优胜者杯、1次西班牙国王杯以及2次西班牙超级杯冠军奖杯。2011年2月14日，饱受伤病折磨的罗纳尔多宣布退役，终结了18年的职业生涯。2015年2月，罗纳尔多宣布复出，加盟美国二级联赛的劳德代尔堡前锋足球俱乐部。

编辑摘要

此词条还可添加 信息模块



罗纳尔多[巴西足球运动员]图册

2.4.1 给列加别名

```
select 字段1 [as] 别名1, 字段2 [as] 别名2, ... from 表名;
```

```
mysql> select pname, price, price/2 as banjia, price%7 yu7 from product;
```

pname	price	banjia	yu7
联想	5000.00	2500.000000	2.00
海尔	3000.00	1500.000000	4.00
雷神	5000.00	2500.000000	2.00
JACK JONES	800.00	400.000000	2.00
真维斯	200.00	100.000000	4.00
花花公子	440.00	220.000000	6.00
劲霸	2000.00	1000.000000	5.00
香奈儿	800.00	400.000000	2.00
相宜本草	200.00	100.000000	4.00
面霸	5.00	2.500000	5.00
神州	2000.00	1000.000000	5.00
印度神油	363.00	181.500000	6.00
高级檀香	45.00	22.500000	3.00

```
13 rows in set (0.00 sec)
```

2.4.2 给表加别名

也可以为表起别名，在多表查询时经常使用：

```
select * from 表名 [as] 别名
```



```
mysql> select * from product as p;
+-----+-----+-----+-----+
| pid | pname      | price  | is_on_sale |
+-----+-----+-----+-----+
| 1   | 联想       | 5000.00 | y          |
| 2   | 海尔       | 3000.00 | y          |
| 3   | 雷神       | 5000.00 | y          |
| 4   | JACK JONES | 800.00  | y          |
| 5   | 真维斯     | 200.00  | n          |
| 6   | 花花公子   | 440.00  | n          |
| 7   | 劲霸       | 2000.00 | y          |
| 8   | 香奈儿     | 800.00  | n          |
| 9   | 相宜本草   | 200.00  | y          |
| 10  | 面霸       | 5.00    | y          |
| 11  | 神州       | 2000.00 | y          |
| 12  | 印度神油   | 363.00  | y          |
| 13  | 高级檀香   | 45.00   | y          |
+-----+-----+-----+-----+
13 rows in set (0.00 sec)
```

2.5 distinct 去掉重复值

```
select distinct 字段名 from 表名;
```

```
mysql> select distinct price from product;
+-----+
| price |
+-----+
| 5000.00 |
| 3000.00 |
| 800.00  |
| 200.00  |
| 440.00  |
| 2000.00 |
| 5.00    |
| 363.00  |
| 45.00   |
+-----+
9 rows in set (0.00 sec)
```

注意:

1. distinct 要写在所有字段的前面



2. 只取出来的字段值都相同时才算是重复

```
mysql>
mysql> select distinct price , pname from product;
+-----+-----+
| price | pname |
+-----+-----+
| 5000.00 | 联想 |
| 3000.00 | 海尔 |
| 5000.00 | 雷神 |
| 800.00 | JACK JONES |
| 200.00 | 真维斯 |
| 440.00 | 花花公子 |
| 2000.00 | 劲霸 |
| 800.00 | 香奈儿 |
| 200.00 | 相宜本草 |
| 5.00 | 面霸 |
| 2000.00 | 神州 |
| 363.00 | 印度神油 |
| 45.00 | 高级檀香 |
+-----+-----+
13 rows in set (0.00 sec)
```

2.6 使用函数

语法:

函数名()

例 1、查看当前 MySQL 版本号:

```
mysql>
mysql> select version();
+-----+
| version() |
+-----+
| 5.7.14 |
+-----+
1 row in set (0.00 sec)
```

查看当前数据库名:


```
mysql> select database();
+-----+
| database() |
+-----+
| db1        |
+-----+
1 row in set (0.00 sec)
```

3. 条件查询

查询时设置一个条件，只有满足条件的记录才会被取出来。

语法：

```
select * from 表名 where 条件
```

条件查询时需要用到的运算符：

分类	符号	描述
比较运算符	> < <= >= = <>	大于、小于、小于等于、大于等于、等于、不等于
	BETWEEN ...AND...	显示在某一区间的值(含头含尾)
	IN(set)	显示在 in 列表中的值，例：in(100,200)
	LIKE '张 pattern'	模糊查询，Like 语句中， % 代表零个或多个任意字符， _ 代表一个字符， 例如：first_name like '_a%';
	IS NULL 和 IS NOT NULL	判断是否为空
逻辑运算符	and	多个条件同时成立
	or	多个条件任一成立
	not	不成立，例：where not(salary>100);

3.1 比较运算符

3.1.1 > < <= >= 等于【=】 不等于【!=或者<>】

例 1、取出价格小于 1000 元的商品：

```
mysql> select * from product where price < 1000;
+-----+-----+-----+-----+
| pid | pname      | price | is_on_sale |
+-----+-----+-----+-----+
| 4   | JACK JONES | 800.00 | y          |
| 5   | 真维斯     | 200.00 | n          |
| 6   | 花花公子   | 440.00 | n          |
| 8   | 香奈儿     | 800.00 | n          |
| 9   | 相宜本草   | 200.00 | y          |
| 10  | 面霸       | 5.00   | y          |
| 12  | 印度神油   | 363.00 | y          |
| 13  | 高级檀香   | 45.00  | y          |
+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

例 2、取出商品名称等于“花花公子”的商品：

```
mysql> select * from product where pname='花花公子';
+-----+-----+-----+-----+
| pid | pname      | price | is_on_sale |
+-----+-----+-----+-----+
| 6   | 花花公子   | 440.00 | n          |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

例 3、取出所有上架的商品【提示：is_on_sale='y'】。

3.1.2 between ... and ... : 【在 xxx 和 xxx 之间】 范围查询

例 1、取出价格在 300 到 3000 之间的商品【包含 300 和 3000】：



```
mysql> select * from product where price between 300 and 3000;
+-----+-----+-----+-----+
| pid | pname      | price  | is_on_sale |
+-----+-----+-----+-----+
| 2   | 海尔       | 3000.00 | y          |
| 4   | JACK JONES | 800.00  | y          |
| 6   | 花花公子   | 440.00  | n          |
| 7   | 劲霸       | 2000.00 | y          |
| 8   | 香奈儿     | 800.00  | n          |
| 11  | 神州       | 2000.00 | y          |
| 12  | 印度神油   | 363.00  | y          |
+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

3.1.3 like: 模糊查询

在 like 查询是可以使用两个特殊符号:

_ : 代表 1 个任意字符。

% : 代表 0 个或者多个做任意字符。

例 1.取出商品名称只有两个字, 并且第二个字是霸的商品:

```
mysql> select * from product where pname like '_霸';
+-----+-----+-----+-----+
| pid | pname | price  | is_on_sale |
+-----+-----+-----+-----+
| 7   | 劲霸  | 2000.00 | y          |
| 10  | 面霸  | 5.00    | y          |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

例 2、取出商品名称只有三个字的商品:

```
mysql> select * from product where pname like '___';
+-----+-----+-----+-----+
| pid | pname | price  | is_on_sale |
+-----+-----+-----+-----+
| 5   | 真维斯 | 200.00 | n          |
| 8   | 香奈儿 | 800.00 | n          |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```



例 3、取出所有商品名称中带“神”字的商品：

```
mysql> select * from product where pname like '%神%';
+-----+-----+-----+-----+
| pid | pname   | price  | is_on_sale |
+-----+-----+-----+-----+
| 3   | 雷神    | 5000.00 | y          |
| 11  | 神州    | 2000.00 | y          |
| 12  | 印度神油 | 363.00  | y          |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

例 4、取出所有商品名称以“神”字结尾的商品：

```
mysql> select * from product where pname like '%神';
+-----+-----+-----+-----+
| pid | pname   | price  | is_on_sale |
+-----+-----+-----+-----+
| 3   | 雷神    | 5000.00 | y          |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

3.1.4 in

取出一个集合中给定的商品。

语法：

```
where 字段 in (数据集合)
```

例：取出所有价格等于 300，800，2000，3000 的商品：

```
mysql> select * from product where price in(300,800,2000,3000);
+-----+-----+-----+-----+
| pid | pname      | price  | is_on_sale |
+-----+-----+-----+-----+
| 2   | 海尔       | 3000.00 | y          |
| 4   | JACK JONES | 800.00  | y          |
| 7   | 劲霸       | 2000.00 | y          |
| 8   | 香奈儿     | 800.00  | n          |
| 11  | 神州       | 2000.00 | y          |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

3.1.5 is null

例：取出价格是 null 的商品：【因为表中没有满足条件的记录，所以取出空】

```
mysql> select * from product where price is null;  
Empty set (0.00 sec)
```

3.2 逻辑运算符

3.2.1 and：并且【同个条件同时满足】

例 1、取出价格大于 1000 并且商品名称中带“神”这个字的：

```
mysql> select * from product where price>1000 and pname like '%神%';  
+-----+-----+-----+-----+  
| pid | pname | price  | is_on_sale |  
+-----+-----+-----+-----+  
| 3   | 雷神  | 5000.00 | y          |  
+-----+-----+-----+-----+  
1 row in set (0.00 sec)
```

3.2.2 or：或者【只要有一个条件满足】

例 1：取出价格大于 3000 或者商品名称以“神”结尾的：



```
mysql> select * from product where price > 3000 or pname like '%神';
+-----+-----+-----+-----+
| pid | pname | price  | is_on_sale |
+-----+-----+-----+-----+
| 1   | 联想  | 5000.00 | y          |
| 3   | 雷神  | 5000.00 | y          |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

例 2：取出商品名称以“神”结尾并且价格大于 3000，或者，商品名称由 3 个字组成并且价格小于 2000：

```
mysql> select * from product where (pname like '%神' and price > 3000) or (pname like '___' and price < 2000);
+-----+-----+-----+-----+
| pid | pname | price  | is_on_sale |
+-----+-----+-----+-----+
| 3   | 雷神  | 5000.00 | y          |
| 5   | 真维斯 | 200.00  | n          |
| 8   | 香奈儿 | 800.00  | n          |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

3.2.3 not：非

例 1：取出商品名称不能 null 的：

```
mysql> select * from product where pname is not null;
+-----+-----+-----+-----+
| pid | pname      | price  | is_on_sale |
+-----+-----+-----+-----+
| 1   | 联想      | 5000.00 | y          |
| 2   | 海尔      | 3000.00 | y          |
| 3   | 雷神      | 5000.00 | y          |
| 4   | JACK JONES | 800.00  | y          |
| 5   | 真维斯    | 200.00  | n          |
| 6   | 花花公子  | 440.00  | n          |
| 7   | 劲霸      | 2000.00 | y          |
| 8   | 香奈儿    | 800.00  | n          |
| 9   | 相宜本草  | 200.00  | y          |
| 10  | 面霸      | 5.00    | y          |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

4. 数据排序



4.1 根据一个字段排序

语法：

```
order by 字段名 asc|desc
```

说明：

1. asc：升序。
2. desc：降序。

例 1、取出所有商品并按价格降序排列：

```
mysql> select * from product order by price desc;
+-----+-----+-----+-----+
| pid | pname | price | is_on_sale |
+-----+-----+-----+-----+
| 1 | 联想 | 5000.00 | y |
| 3 | 雷神 | 5000.00 | y |
| 2 | 海尔 | 3000.00 | y |
| 7 | 劲霸 | 2000.00 | y |
| 4 | JACK JONES | 800.00 | y |
| 8 | 香奈儿 | 800.00 | n |
| 6 | 花花公子 | 440.00 | n |
| 5 | 真维斯 | 200.00 | n |
| 9 | 相宜本草 | 200.00 | y |
| 10 | 面霸 | 5.00 | y |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

例 2、取出所有带“神”的商品，并且按价格升序排列

4.2 根据多个字段排序

语法：

```
order by 字段1 asc|desc , 字段2 asc|desc , 字段3 asc|desc....
```

说明：先根据第一个字段排序，当第一个字段的值相同时再根据第二个字段排序，依次类推：

例 1：取出所有的商品以价格降序排列，价格相同的，以 pid 降序排列：

```
mysql> select * from product order by price desc,pid desc;
+-----+-----+-----+-----+
| pid | pname | price | is_on_sale |
+-----+-----+-----+-----+
| 3 | 雷神 | 5000.00 | y |
| 1 | 联想 | 5000.00 | y |
| 2 | 海尔 | 3000.00 | y |
| 7 | 劲霸 | 2000.00 | y |
| 8 | 香奈儿 | 800.00 | n |
| 4 | JACK JONES | 800.00 | y |
| 6 | 花花公子 | 440.00 | n |
| 9 | 相宜本草 | 200.00 | y |
| 5 | 真维斯 | 200.00 | n |
| 10 | 面霸 | 5.00 | y |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```




4.3 随机排序

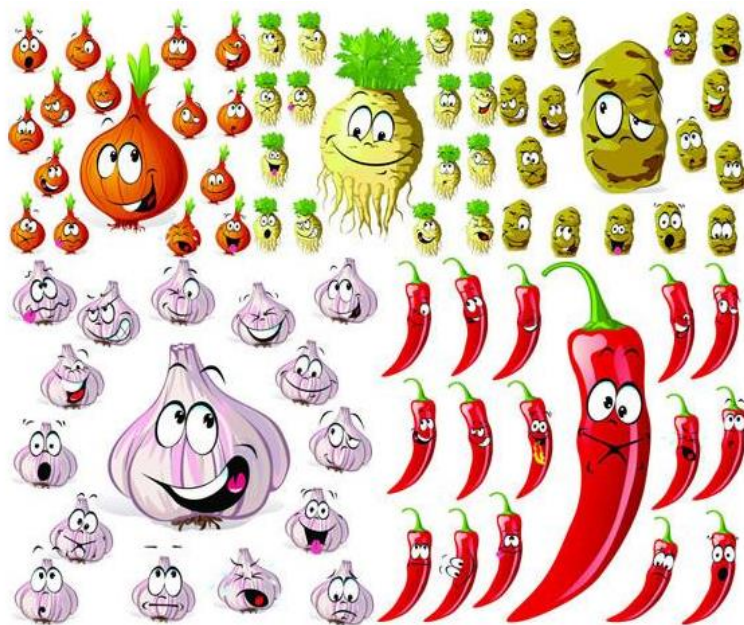
语法:

```
order by rand()
```

rand() : 随机函数

5. 数据分组

分组，根据一个规则把事物归类，相同的在一起。



5.1 分组语法

```
group by 字段名 1, 字段名 2.....
```

5.2 分组的特点

1. 字段值相同的归为一组
2. select 的字段中，不能出现未在分组中的字段

比如：根据 is_on_sale 字段分组，那么 select 只能取这个字段否则出错：

```
mysql> select * from product group by is_on_sale;
ERROR 1055 (42000): Expression #1 of SELECT list is not in GROUP BY clause and contains nonaggregated column 'db1.product.pid' which is not functionally dependent on columns in GROUP BY clause; this is incompatible with sql_mode=only_full_group_by
mysql>
```

只能 select 分组的字段：

```
mysql> select is_on_sale from product group by is_on_sale;
+-----+
| is_on_sale |
+-----+
| y          |
| n          |
+-----+
2 rows in set (0.00 sec)
```

3. 最终结果只保留每一组中的第一条记录

总结：单纯的分组，和去重复实际上是一样的。

5.3 聚合函数

MySQL 中提供了一套聚合函数：

max(字段名)：取表中这个字段的最大值。
min(字段名)：取表中这个字段的最小值。
avg(字段名)：取表中这个字段的平均值。
sum(字段名)：取表中这个字段所有值的和。
count(*或者字段名)：统计记录数。

重点说明：以上函数在不分组时，是对所有数据进行统计，返回一个结果，当分组时，是对每一组进行统计，返回每一组的结果。

例 1、取出表中的最大价格：

```
mysql> select max(price) from product;
+-----+
| max(price) |
+-----+
|      5000.00 |
+-----+
1 row in set (0.00 sec)
```

例 2、取出“上架”商品和“下架”商品分别的最大价格：

```
mysql> select is_on_sale,max(price) from product group by is_on_sale;
+-----+-----+
| is_on_sale | max(price) |
+-----+-----+
| y          |      5000.00 |
| n          |       800.00 |
+-----+-----+
2 rows in set (0.00 sec)
```

例 3、取出表中总的记录数：

```
mysql> select count(*) from product;
+-----+
| count(*) |
+-----+
|         10 |
+-----+
1 row in set (0.00 sec)
```

例 4、取出“上架”商品和“下架”商品分别的记录数：

```
mysql> select count(*),is_on_sale from product group by is_on_sale;
+-----+-----+
| count(*) | is_on_sale |
+-----+-----+
|         7 | y          |
|         3 | n          |
+-----+-----+
2 rows in set (0.00 sec)
```



◇ 扩展: `count(*)`和 `count(字段名)`的区别

`count(*)`: 取表中记录数。

`count(字段名)`: 取表中这个字段值不为 `null` 的记录数。

```
abc
```

```
id    name
```

```
1      abc
```

```
2      null
```

```
3      null
```

```
|
```

```
select count(name) from abc;    --> 1
```

6. having 过滤数据

`having` 和 `where` 很像都可以用来实现数据的过滤，它们的区别是：

1. 位置不同，`where` 写在 `group by` 前面、`having` 写在 `group by` 后面。
2. 执行时机不同，`where` 是在聚合函数之前执行，`having` 是在聚合函数之后执行。



```
mysql> select is_on_sale,max(price) mp from product group by is_on_sale  
-> ;
```

is_on_sale	mp
y	5000.00
n	800.00

2 rows in set (0.00 sec)

使用where时报错，因为执行where时还没有计算出mp呢，因为where是在max聚合函数之前执行的。

```
mysql> select is_on_sale,max(price) mp from product where mp > 2000 group by is_on_sale  
-> ;  
ERROR 1054 (42S22): Unknown column 'mp' in 'where clause'
```

```
mysql>
```

```
mysql>
```

```
mysql> select is_on_sale,max(price) mp from product group by is_on_sale having mp > 2000;  
+-----+-----+  
| is_on_sale | mp      |  
+-----+-----+  
| y          | 5000.00 |  
+-----+-----+  
1 row in set (0.00 sec)
```

having 的应用场景：一般 having 都是和 group by 以及聚合函数一起使用的。

总结：平时都使用 where，但是当需要用聚合函数的结果作为条件时需要使用 having。

7. 限制记录数

可以使用 limit 限制取出记录的数量，limit 要写在 sql 语句的最后。

语法：

```
limit 起始记录,记录数
```

说明：

1. 起始记录是指从第几条记录开始取，第一条记录的下标是 0。
2. 记录数是指从起始记录开始向后依次取的记录数。

比如：limit 0,10 ，意思是：从第 1 条记录开始取，取 10 条。

例 1：取出表中前 3 件商品



```
mysql> select * from product limit 0,3;
+-----+-----+-----+-----+
| pid | pname | price | is_on_sale |
+-----+-----+-----+-----+
| 1 | 联想 | 5000.00 | y |
| 2 | 海尔 | 3000.00 | y |
| 3 | 雷神 | 5000.00 | y |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

注意：当起始值是 0 时，可以省略为 limit 记录数，所以取出前三条记录：

```
mysql> select * from product limit 3;
+-----+-----+-----+-----+
| pid | pname | price | is_on_sale |
+-----+-----+-----+-----+
| 1 | 联想 | 5000.00 | y |
| 2 | 海尔 | 3000.00 | y |
| 3 | 雷神 | 5000.00 | y |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

例 2、从第 3 条记录开始取，取 3 条

```
mysql> select * from product limit 2,3;
+-----+-----+-----+-----+
| pid | pname | price | is_on_sale |
+-----+-----+-----+-----+
| 3 | 雷神 | 5000.00 | y |
| 4 | JACK JONES | 800.00 | y |
| 5 | 真维斯 | 200.00 | n |
+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

8. 今日总结

