实验五 单表操作——商品评论表的操作以及排序分组实验

1. 实验目的  
   1、掌握单表操作的数据操作等相关命令  
   2、掌握select命令中分组以及聚合函数的使用

3、熟悉运算符的使用，几个特殊的运算符（between...and...，模糊查询like等）重点掌握。  
二、实验内容

1、根据教材P150-152，完成商品表和商品评论表的各种需求的查询操作。

1. 查询商品id等于8且有效的评论内容
2. 查询每个用户评论的商品数量。
3. 查询最新发布的5条有效商品评论信息
4. 查询评论过两种以上不同商品的用户id及对应商品的id。
5. 结合sh\_goods和sh\_goods\_comment表，查询没有任何评论信息的商品id和name。
6. 结合sh\_goods和sh\_goods\_comment表，查询商品评分为5星的商品评论信息。
7. 依据sh\_goods的结构与数据，在mydb数据库中创建一张tm\_goods表，并将价格在20到50之间的商品价格减5元，库存量再新增300件。
8. 在sh\_goods表中查询评分小于4的商品的不同分类id。
9. 头歌：https://www.educoder.net/tasks/fmqzit2ba6uj，掌握如何排序及分组数据。

三、实验步骤和过程记录

步骤1：选定要操作的数据库，并查询商品id等于8且有效的评论。其代码和结果如下图所示。

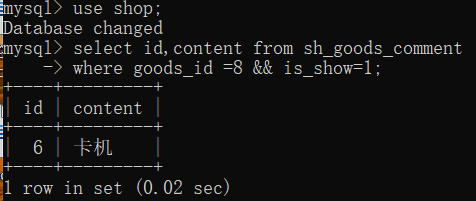


图1 选数据库和修改属性值

步骤2：选取数据库和修改属性之后，紧接着就是查询每个用户评论的商品数量。其代码和结果如下图2所示。

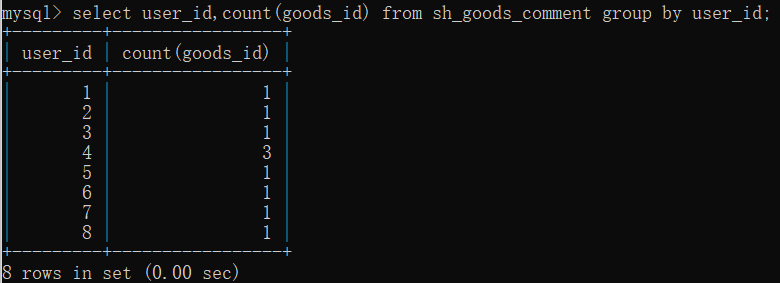


图2 查询每个用户评论的商品数量

步骤3：在进行查询每个用户评论的商品数量操作之后，紧接着的就是查询最新发布的5条有效商品评论信息。其操作的代码和结果如下图3所示。

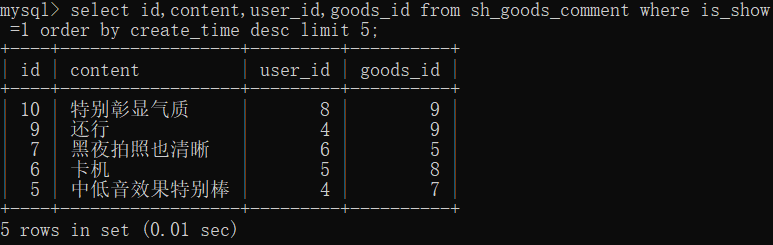


图3 查询最新发布的5条有效商品评论信息

步骤4：在进行查询最新发布的5条有效商品评论信息操作之后，紧接着的就是查询评论过两种以上不同商品的用户id对应的商品id。其操作的代码和结果如下图4所示。

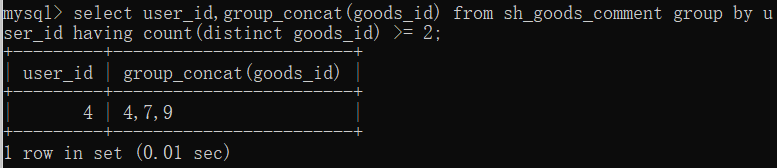


图4 查询评论过两种以上不同商品的用户id对应的商品id

步骤5：在进行查询评论过两种以上不同商品的用户id对应的商品id操作之后，紧接着的就是结合sh\_goods\_comments和sh\_goods表，查询没有任何评论信息的商品id和name。其操作的代码和结果如下图5所示。

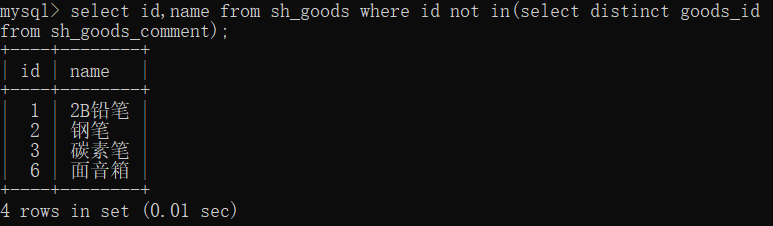


图5 结合sh\_goods\_comments和sh\_goods表查询没有任何评论信息的商品id和name

步骤6： 在进行结合sh\_goods\_comments和sh\_goods表，查询没有任何评论信息的商品id和name的操作之后，紧接着的就是结合sh\_goods\_comments和sh\_goods表查询商品评分为5星的商品评论信息。其操作的代码和结果如下图6所示。

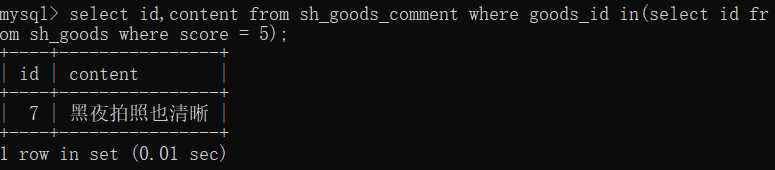


图6 结合sh\_goods\_comments和sh\_goods表查询商品评分为5星的商品评论信息

步骤7：在mydb数据库中创建一张tm\_goods表，并将价格在20到50之间的商品价格减5元，库存量再新增300件，其操作的代码和结果如下图7所示。

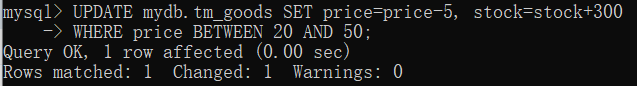


图7 修改价格在20到50之间的商品的价格和库存操作图

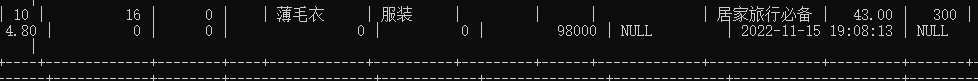


图8 修改价格在20到50之间的商品的价格和库存结果图

步骤8：在sh\_goods表中查询评分小于4的商品的不同分类id的操作代码及其结果如下图9所示。

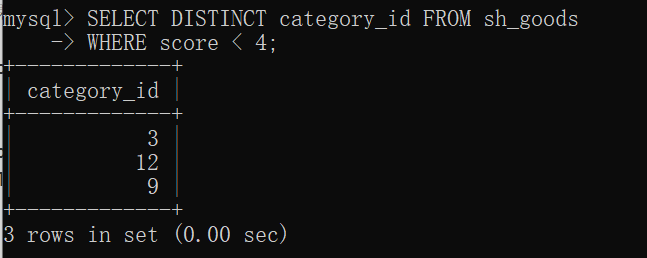


图9 在sh\_goods表中查询评分小于4的商品的不同分类id

1. 问题思考与总结  
    1、聚合函数的使用有什么要求
2. 聚合函数可以应用于查询语句的SELECT中，或者HAVING子句中，并结合group by使用
3. 不可用于WHERE语句中，因为WHERE是对逐条的行记录进行筛选。having是一个过滤声明，通常在查询书里末端，主要用来对经过前面各种约束后查询到的数据结果进行过滤。注意：having只能放在group by之后并结合group by使用，没有group by 不能使用having。
4. where是一个约束声明，在较早的位置执行，先执行where的约束条件我，再返回结果。注意：where子句不能使用聚合函数。
5. 分组应用总结

（一）GROUP BY子句可以包含任意数目的列。使得对分组进行嵌套，为数据分组提供更细致的控制。（二）若在GROUP BY中嵌套分组，数据将在最后规定的分组上进行汇总。在建立分组时，指定的所有列都一起计算。（三）GROUP BY子句中列出的每个列都必须是检索列或有效的表达式，若在select中使用表达式，则必须在GROUP BY中指定相同的表达式。（四）除聚集计算语句外，select语句中的每个列都必须在GROUP BY子句中给出。（五）若分组列中有null值，则null将作为一个分组返回。若列中有多行null，它们将分为一组。（六）GROUP BY出现在where后，order by前。

1. 运算符的使用总结

表1 SQL 逻辑运算符表

|  |  |
| --- | --- |
| 运算符 | 描述 |
| = | 检查两个操作数的值是否相等，如果是，则条件为真(true)。 |
| != | 检查两个操作数的值是否相等，如果值不相等则条件为真(true)。 |
| <> | 检查两个操作数的值是否相等，如果值不相等则条件为真(true)。 |
| > | 检查左操作数的值是否大于右操作数的值，如果是，则条件为真(true)。 |
| < | 检查左操作数的值是否小于右操作数的值，如果是，则条件为真(true)。 |
| >= | 检查左操作数的值是否大于或等于右操作数的值，如果是，则条件为真(true)。 |
| <= | 检查左操作数的值是否小于或等于右操作数的值，如果是，则条件为真(true)。 |
| !< | 检查左操作数的值是否不小于右操作数的值，如果是，则条件变为真(true)。 |
| !> | 检查左操作数的值是否不大于右操作数的值，如果是，则条件变为真(true)。 |

表2 SQL 比较运算符表

|  |  |
| --- | --- |
| 运算符 | 描述 |
| ALL | ALL运算符用于将值与另一个值集中的所有值进行比较。 |
| AND | AND运算符允许在SQL语句的WHERE子句中指定多个条件。 |
| ANY | ANY运算符用于根据条件将值与列表中的任何适用值进行比较。 |
| BETWEEN | BETWEEN运算符用于搜索在给定最小值和最大值内的值。 |
| EXISTS | EXISTS运算符用于搜索指定表中是否存在满足特定条件的行。 |
| [IN](https://www.w3cschool.cn/sql/r9qdifov.html) | IN运算符用于将值与已指定的文字值列表进行比较。 |
| LIKE | LIKE运算符用于使用通配符运算符将值与类似值进行比较。 |
| NOT | NOT运算符反转使用它的逻辑运算符的含义。 例如：NOT EXISTS, NOT BETWEEN, NOT IN等等，这是一个否定运算符。 |
| OR | OR运算符用于组合SQL语句的WHERE子句中的多个条件。 |
| IS NULL | IS NULL运算符用于将值与NULL值进行比较。 |
| UNIQUE | UNIQUE运算符搜索指定表的每一行的唯一性(无重复项)。 |

表3 SQL 算术运算符表

|  |  |
| --- | --- |
| 运算符 | 描述 |
| + | 加法，执行加法运算。 |
| - | 减法，执行减法运算。 |
| \* | 乘法，执行乘法运算。 |
| / | 用左操作数除以右操作数。 |
| % | 用左操作数除以右操作数并返回余数。 |

1. 本章重要命令的总结。

**数据库的创建**：

create database 数据库名称;create database 数据库名称 character set 字符集;create database 数据库名称 character set 字符集 collate 比较规则(utf8\_general\_ci);

**数据库的修改：**alter database 数据库名称 character set 字符集;

数据库的删除:

drop database 数据库名称;**数据库的查看：**

show databases;//展示所有数据库

show create database 数据库名称;//查看数据库的字符集（查看建立数据库的过程）

**表的创建：**

create table 表的名称 (

列名1（通常为id） 数据类型（一般为int) 约束1 约束2 …（primary key auto\_increment）,

列名2 数据类型 约束1 约束2 …,

…………

列名n 数据类型 约束1 约束2 …，

);

**表的修改：**

alter table 表的名称 add 列名 数据类型 约束1 约束2……;//在表中新加一列。

alter table 表的名称 modify 列名 新数据类型 约束1 约束2……;//修改列的数据类型

alter table 表的名称 drop 列名;//删除某列

alter table 表的名称 character set 字符集; //修改表的字符集

alter table 表的名称 change 列名 新列名 新数据类型 约束1 约束2……；//修改列名（重新定义）

rename table 表的名称 to 新的表名;

**表的删除：**

drop table 表的名称;

**表的查看：**

show tables;//查看数据库内的所有表

show create table 表的名称;//查看建立表时的所用语句

desc 表的名称;//查看表的结构

**插入语句：**

insert into 表名 values(对应插入数据1,对应插入数据2,对应插入数据3,……);

**更新语句：**

update 表名 set 列名=具体数据;//不使用where默认进行全体操作，对应列全部修改为具体数据。

update 表名 set 列名=具体数据 where 定位语句;（例如id=1)//修改对应定位的数据；

update 表名 set 列名1=具体数据1,列名2=具体数据2…… where 定位语句;（例如id=1)//修改多处

update 表名 set 列名=列名运算结果式（结果是具体数据） where 定位语句;（例如id=1)//修改对应定位的数据；

**删除语句：**

delete from 表名;//不加定位，全部删除！

delete from 表名 where 定位语句;//删除定位项

查询语句：

select \* from 表名;//全体查询，显示整张表的数据。

select 列名1,列名2,…… from 表名;//全体查询，显示需要的对应项目。

select \*,列名运算式 as 新名称（列名计算式） from 表名;//as可以不写，显示全体加上新添加的项目

select \* from 表名 where 定位语句;//加入定位语句，显示满足定位条件的所有项目

select \* from 表名 where 定位语句;

**使用order by 排序：**

select 函数名(对应参数) from 表名

5.遇到的问题

**问题1：**按照课本的代码写上去报错，内容是No database selected，这是因为课本的数据库操作是和上一章连在一起的，因此不需要再次选中shop数据库，重新登录数据库之后需要选中数据。因此加上一句use shop;命令，shop数据库被选中。截图如下图所示。

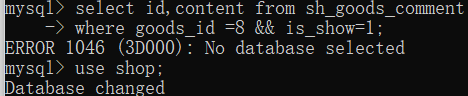


图 问题1报错和解决方式