

实验七 用户权限、视图及事务实验

一、实验目的

- 1、掌握创建用户、用户权限的授权和回收权限等相关命令的使用
- 2、掌握视图的基本操作。
- 3、掌握事务的隔离级别的设置。

二、实验内容

1、在 shop 数据库中创建两个视图。

(1) 创建用户 `shop_view_test`;

(2) 创建视图: `sh_view_attr`: 用于根据商品分类的 `id` 查所有属性信息; 要求: 指定视图用户为 `DEFINER`。

(3) 查询视图中所有的记录, 以及原表中有而视图中没有的字段信息。增加截图分析结果。

(4) 显示的记录条数限制 `limit n`, 测试 `limit` 自句在原表中使用 `select` 语句和视图的 `select` 语句中使用的差异。

(5) 创建视图: `sh_view_goods_attr`: 用于根据商品 `id` 查找所有属性信息。要求: 指定给视图分别指定控制视图安全 `SQL SECURITY` 的方式分别为: `DEFINER`, 和 `INVOKER`。

(6) 为视图: `sh_view_goods_attr` 创建用户, 并指定用户的权限为 `select`、`update` 和 `delete` 权限。在该权限下, 执行以上三个命令, 给视图表做相应的操作, 并能看到效果, 截图分析。

(7) 回收以上用户 `update` 和 `delete` 的权限。再执行 `update` 和 `delete` 操作, 看是否能执行, 截图分析。

2、利用十五实现再用户下订单后, 订单商品表 `sh_order_goods` 中对应订单插入的商品数量大于实际商品库存量时, 取消 `sh_order_goods` 表中数据的添加。

(1) 事务操作前, 查看 `sh_order_goods` 订单商品表

(2) 开启事务。

(3) 在事务中执行对应操作。

(4) 回滚事务。

(5) 查看 `sh_order_goods` 订单商品表。

3、利用事务在用户下订单前, 检测当前用户是否已被激活, 若未激活, 则需要激活此

用户后，才能再次下订单。

三、实验步骤和过程记录

步骤 1: 创建用户 shop_view_test 的 SQL 语言以及操作成功的截图如下图 1 所示。

```
mysql> CREATE USER 'shop_view_test'@'localhost' IDENTIFIED BY '1234';
Query OK, 0 rows affected (0.00 sec)
```

图 1 创建用户 shop_view_test

步骤 2: 创建视图: sh_view_attr 的 SQL 语言以及操作成功的截图如下图 2 所示。

```
mysql> use shop;
Database changed
mysql> Create View sh_view_attr As
-> Select a.sort sort1,a.name attr1,b.sort sort2,b.name attr2,
-> b.category_id From sh_goods_attr a Join sh_goods_attr b On a.id=b.parent_id
-> Where b.parent_id!=0 Order By a.sort ASC,b.sort ASC;
Query OK, 0 rows affected (0.01 sec)
```

图 2 创建 sh_view_attr 视图

步骤 3: 在 sh_view_attr 中查询 category_id 为 6 的 SQL 语言以及操作成功的截图如下图 3 所示。

```
mysql> Select * From sh_view_attr Where category_id=6;
```

sort1	attr1	sort2	attr2	category_id
0	基本信息	0	机身颜色	6
0	基本信息	1	输入方式	6
0	基本信息	2	操作系统	6
1	屏幕	0	屏幕尺寸	6
1	屏幕	1	屏幕材质	6
1	屏幕	2	分辨率	6
2	摄像头	0	前置摄像头	6
2	摄像头	1	后置摄像头	6
3	电池信息	0	电池容量	6
3	电池信息	1	是否可拆卸	6

10 rows in set (0.00 sec)

图 3 查询 category_id=6 的结果

步骤 4: 查询 sh_goods_attr 表的 SQL 语言以及操作成功的截图如下图 4 所示。

```
mysql> Select * From sh_goods_attr;
```

id	parent_id	category_id	name	sort
1	0	6	基本信息	0
2	1	6	机身颜色	0
3	1	6	输入方式	1
4	1	6	操作系统	2
5	0	6	屏幕	1
6	5	6	屏幕尺寸	0
7	5	6	屏幕材质	1
8	5	6	分辨率	2
9	0	6	摄像头	2
10	9	6	前置摄像头	0
11	9	6	后置摄像头	1
12	0	6	电池信息	3
13	12	6	电池容量	0
14	12	6	是否可拆卸	1

14 rows in set (0.00 sec)

图 4 查询 sh_goods_attr 表

步骤 5: 显示的记录条数限制 limit n, 测试 limit 自句在原表中使用 select 语句和视图的 select 语句中使用的差异的 SQL 语言以及操作成功的截图如下图 5 所示。

```
mysql> Select * From sh_view_attr Where category_id=6 limit 3;
```

sort1	attr1	sort2	attr2	category_id
0	基本信息	0	机身颜色	6
0	基本信息	1	输入方式	6
0	基本信息	2	操作系统	6

3 rows in set (0.00 sec)

图 5 显示的记录条数限制

步骤 6: 创建视图 sh_view_goods_attr 用于根据商品 id 查找所有属性信息的 SQL 语言以及操作成功的截图如下图 6 示。

```
mysql> Create View sh_view_goods_attr AS
-> Select b.name attr1, a.name attr2, c.attr_value, c.goods_id
-> From sh_goods_attr a Join sh_goods_attr b On a.parent_id=b.id
-> Join sh_goods_attr_value c
-> On a.id=c.attr_id Where a.parent_id !=0 Order By b.sort ASC, a.sort ASC;
Query OK, 0 rows affected (0.01 sec)
```

图 6 创建视图 sh_view_goods_attr

步骤 7: 为视图: sh_view_goods_attr 创建用户, 并指定用户的权限为 select、update 和 delete 权限的 SQL 语言以及操作成功的截图如下图 7 示。

```
mysql> select * from sh_view_goods_attr ;
```

attr1	attr2	attr_value	goods_id
基本信息	机身颜色	黑色	5
基本信息	输入方式	触摸屏	5
基本信息	操作系统	Android	5
屏幕	屏幕尺寸	5.5寸	5
屏幕	屏幕材质	IPS	5
屏幕	分辨率	1920*1080	5
摄像头	前置摄像头	1600万	5
摄像头	后置摄像头	800万	5
电池信息	电池容量	3500mAh	5
电池信息	是否可拆卸	否	5

10 rows in set (0.00 sec)

图 7 为视图添加用户

步骤 8: 回收以上用户 update 和 delete 的权限。再执行 update 和 delete 操作的 SQL 语言以及操作成功的截图如下图 8 所示。

```
select *from sh_view_goods_attr;
ERROR 1142 (42000): INSERT command denied to user
'sh_view_goods_attr'@'localhost' for table 'sh_view_goods_attr'
```

图 8 回收用户权限

步骤 9: 事务操作前, 查看 sh_order_goods 订单商品表的 SQL 语言以及操作成功的截图如下图 9 示。

```
mysql> select id, order_id, goods_id, goods_num from sh_order_goods;
Empty set (0.00 sec)
```

图 9 查看 sh_order_goods 订单

步骤 10: 开启事务的 SQL 语言以及操作成功的截图如下图 10 所示。

```
mysql> start transaction;
Query OK, 0 rows affected (0.00 sec)
```

图 10 开启事务

步骤 11: 在事务中执行插入的 SQL 语言以及操作成功的截图如下图 11 所示。

```
mysql> insert Into sh_order_goods
-> (order_id,goods_id,goods_num) values(1,9,6);
Query OK, 1 row affected (0.00 sec)

mysql> select id,order_id,goods_id,goods_num from sh_order_goods;
+----+-----+-----+-----+
| id | order_id | goods_id | goods_num |
+----+-----+-----+-----+
| 2 | 1 | 9 | 6 |
+----+-----+-----+-----+
1 row in set (0.00 sec)
```

图 11 事务表中实现插入

步骤 12: 回滚事务的 SQL 语言以及操作成功的截图如下图 12 所示。

```
mysql> rollback;
Query OK, 0 rows affected (0.00 sec)
```

图 12 回滚操作

步骤 13: 查看 sh_order_goods 订单商品表的 SQL 语言以及操作成功的截图如下图 13 所示。

```
mysql> select id,order_id,goods_id,goods_num
-> from sh_order_goods;
Empty set (0.00 sec)
```

图 13 查看 sh_order_goods 订单

四、问题思考与总结

问题 1: 在创建视图 sh_view_test 时, 没有选择数据库, 结果提示错误 No database selected。

报错截图如下图 14 所示。

```
mysql> Create View sh_view_attr As
-> Select a.sort sort1,a.name attr1,b.sort sort2,b.name attr2,
-> b.category_id From sh_goods_attr a Join sh_goods_attr b On a.id=b.parent_id
-> Where b.parent_id!=0 Order By a.sort ASC,b.sort ASC;
ERROR 1046 (3D000): No database selected
```

图 14 问题 1 报错

解决 1: 报错的原因是没有选择数据库, 因此给该操作选择一个数据库就能够保证程序运行正确。解决报错的 SQL 语句以及正确运行的结果如下图 15 所示。

```
mysql> use shop;  
Database changed  
mysql> Create View sh_view_attr As  
-> Select a.sort sort1,a.name attr1,b.sort sort2,b.name attr2,  
-> b.category_id From sh_goods_attr a Join sh_goods_attr b On a.id=b.parent_id  
-> Where b.parent_id!=0 Order By a.sort ASC,b.sort ASC;  
Query OK, 0 rows affected (0.01 sec)
```

图 15 问题 1 解决截图

根据以上实验结果, 分析本次实验。

1、视图的含义

视图是一种虚拟的表, 是从数据库中的一个表或多个表中导出来的表, 视图还可以从已经存在的视图的基础上定义。数据库只存放了视图的定义, 而并没有存放视图中的数据。这些数据存放在原来的表中。使用视图查询数据时, 数据库系统会从原来的表中取出对应的数据。因此, 视图中的数据依赖于原来的表中的数据。一旦表中的数据发生改变, 显示在视图中的数据也会发生改变。

2、视图的作用

视图是在原有的表或者视图的基础上重新定义的虚拟表, 这可以从原有的表上选取对用户有用的信息。视图起着类似于筛选的作用。

2.1 使用操作简单化, 视图需要达到的目的就是所见即所需。

2.2 增加数据的安全性, 通过视图, 用户只能查询和修改指定的数据。

2.3 提高表的逻辑独立性, 视图可以屏蔽原有表结构变化带来的影响

3、视图的创建

创建视图是指已存在的数据库上建立视图。视图可以建立在一张表或多张表中。

```
CREATE [ALGORITHM = {UNDEFINED | MERGE | TEMPTABLE}]
```

```
VIEW 视图名 [(属性清单)]
```

```
AS SELECT 语句
```

```
[WITH [CASCADED | LOCAL] CHECK OPTION];
```

4、视图的查看

查看视图是指查看数据库中已存在的视图的定义。查看视图必须要有的 SHOW VIEW 的权限, MySQL 数据库下的 user 表中保存着这个信息。查看视图的方法包括 DESCRIBE 语句、SHOW TABLE STATUS 语句、SHOW CREATE VIEW 语句和查询 information_schema 数据库下的 views 表等。