

实验二报告

一、 观察并回答问题

1. 关于视图

(1) sakila.mwb 模型图中共有几个 View?

总共有 7 个。

(2) 分析以下 3 个视图，回答以下问题：

视图名	关联表	作用
actor_info	Film, actor, category, film_actor, film_category	列出演员出演的电影，按照类型分类。
film_list	Film, category, film_category, actor, film_actor	列出电影和出演该电影的演员，演员用逗号分离开。
sales_by_film_category	Category, payment, rental, inventory, film, film_category, category	列出所有分类之下的租售的金额。

(3) 分别执行以下 2 句 SQL 语句：

```
update staff_list set `zip code` = '518055' where ID = '1';
```

```
update film_list set price = 1.99 where FID = '1';
```

截图执行结果，并分析一下视图在什么情况下可以进行 update 操作，什么情况下不能？

```
mysql> update staff_list set `zip code` = '518055' where ID = '1';
Query OK, 1 row affected (0.05 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> update film_list set price = 1.99 where FID = '1';
ERROR 1288 (HY000): The target table film_list of the UPDATE is not updatable
mysql> 
```

- (4) 执行以下命令查询 sakila 数据库中的视图是否可更新，截图执行结果：

```
SELECT table_name, is_updatable FROM information_schema.views  
WHERE table_schema = 'sakila';
```

```
mysql> SELECT table_name, is_updatable FROM informat  
ion_schema.views  
→ WHERE table_schema = 'sakila';  
  
+-----+-----+  
| TABLE_NAME | IS_UPDATABLE |  
+-----+-----+  
| actor_info | NO |  
| customer_list | YES |  
| film_list | NO |  
| nicer_but_slower_film_list | NO |  
| sales_by_film_category | NO |  
| sales_by_store | NO |  
| staff_list | YES |  
+-----+-----+  
7 rows in set (0.01 sec)  
  
mysql> 
```

2. 关于触发器

- (1) 触发器 customer_create_date 建在哪个表上？这个触发器实现什么功能？在这个表上新增一条数据，验证一下触发器是否生效。（截图语句和执行结果）

在 customer 表上，其能够在插入 customer 的时候自动生成创建日期。



Administrator SQL File 3*

Limit to 1000 rows

```
1 • select customer_id, first_name, create_date from customer;
```

Result Grid Filter Rows: Edit: Export/Import:

	customer_id	first_name	create_date
	593	RENE	2006-02-14 22:04:37
	594	EDUARDO	2006-02-14 22:04:37
	595	TERRENCE	2006-02-14 22:04:37
	596	ENRIQUE	2006-02-14 22:04:37
	597	FREDDIE	2006-02-14 22:04:37
	598	WADE	2006-02-14 22:04:37
	599	AUSTIN	2006-02-14 22:04:37
	600	walter	2023-09-12 18:00:23
	602	Alice	2023-09-21 22:34:29
*	NULL	NULL	NULL

- (2) 触发器 upd_film 建在哪个表上？这个触发器实现什么功能？在这个表上修改一条数据的 description 字段，验证一下触发器是否生效。（截图语句和执行结果）

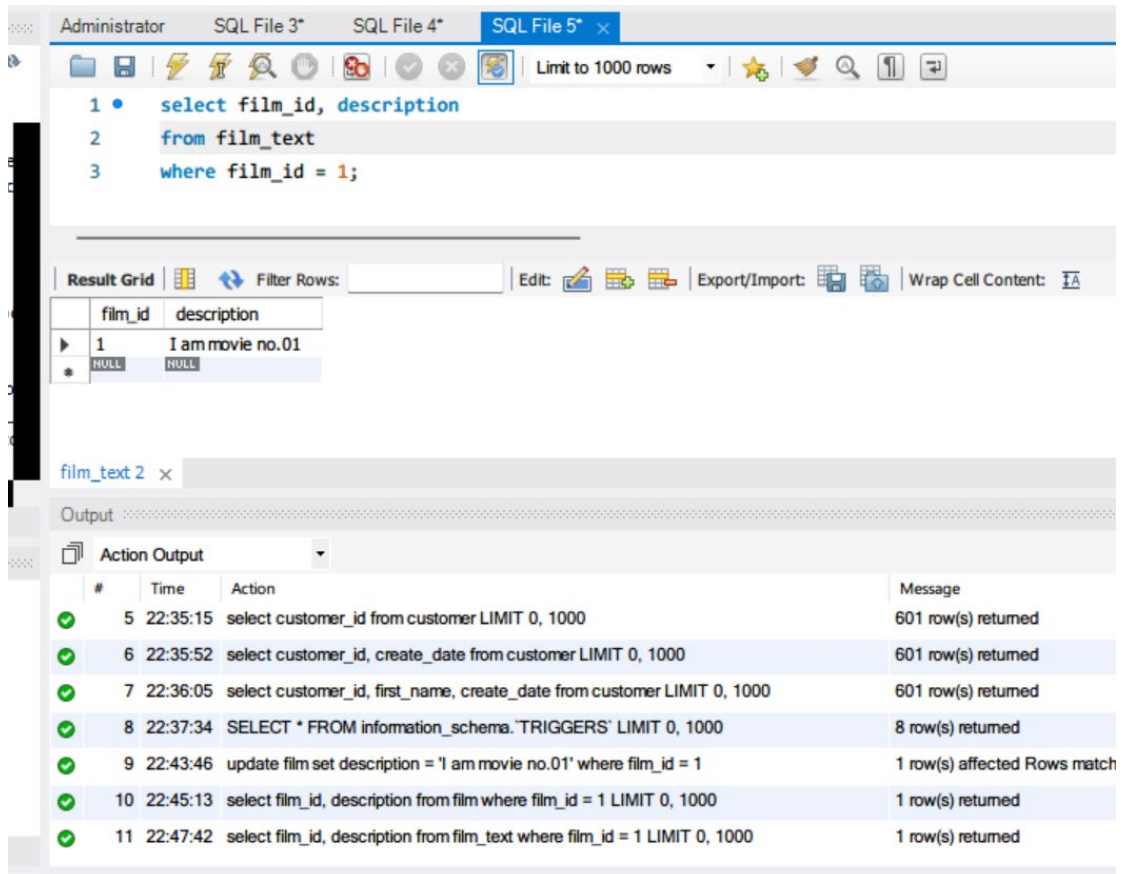
Upd_film 建在 film 表上。其能够在更新 film 的信息的时候，将 film 的更新同步到 film_text 表之中。

Administrator SQL File 3* SQL File 4* SQL File 5*

Limit to 1000 rows

```
1 update film
2 set description = 'I am movie no.01'
3 where film_id = 1;
```

可以看出 film_text 已经被更新了。



- (3) 我们可以看到 sakila-schema.sql 里的语句是用于创建数据库的结构,包括表、视图、触发器等,而 sakila-data.sql 主要是用于往表写入数据。但 sakila-data.sql 里有这样一个建立触发器 payment_date 的语句,这个触发器是否可以移到 sakila-schema.sql 里去执行?为什么?

不能。因为这个触发器能够在插入 payment 的时候自动更新 payment 日期。但是在导入 data 的时候,就要插入 payment,若是在导入之前就定义了触发器,则所有的 payment 日期都变为导入的日期。

3. 关于约束

观察 sakila-schema.sql 里面的 create table store 一段:

```
335 • CREATE TABLE store (
336   store_id TINYINT UNSIGNED NOT NULL AUTO_INCREMENT,
337   manager_staff_id TINYINT UNSIGNED NOT NULL,
338   address_id SMALLINT UNSIGNED NOT NULL,
339   last_update TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,
340   PRIMARY KEY (store_id),
341   UNIQUE KEY idx_unique_manager (manager_staff_id),
342   KEY idx_fk_address_id (address_id),
343   CONSTRAINT fk_store_staff FOREIGN KEY (manager_staff_id) REFERENCES staff (staff_id) ON DELETE RESTRICT ON UPDATE CASCADE,
344   CONSTRAINT fk_store_address FOREIGN KEY (address_id) REFERENCES address (address_id) ON DELETE RESTRICT ON UPDATE CASCADE
345 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
346
```

- (1) store 表上建了哪几种约束? 这些约束分别实现什么功能? (至少写 3 个)

约束类型	功能
Primary key	说明 store_id 是主键，这要求其值是唯一的，并且不能为 null。
Unique key	要求键的值是唯一的。
Default on	对于 last_update 来说，其在 on 后面的语句，也就是 update 的时候会更新为默认值，其值为 current_timestamp，也就是当前时间。
Foreign key	说明 manager_staff_id 是来自 staff 表的外键。

(2) 图中第 343 行的 ON DELETE RESTRICT 和 ON UPDATE CASCADE 是什么意思？
ON UPDATE CASCADE 的意思是，当这个外键在其来源的表中被更新的时候，这个表内的外键也会被更新。

ON DELETE RESTRICT 的意思是，当这个外键在来源的表中被删除的时候，这个删除操作将不被允许。

4. 关于存储过程

观察 sakila-schema.sql 里面的 rewards_report 存储过程，结合官方文档分析：

(1) 这个存储过程 rewards_report 实现了什么功能？输出参数 count_rewardees 是什么？

Rewards_report 过程可以计算上个月收入和消费记录达到一定条件的消费者的数量。
一个参数是消费者上个月的满足条件的最低收入，另一个满足条件的最低购买次数。
Count_rewardees 就是计算出的消费者的数量。

(2) 图中第 483 行的 NOT DETERMINISTIC 和第 485 行的 SQL SECURITY DEFINER 分别是什么含义？
若是对于同样的输入参数，过程的返回值都是一致的，则称这个过程是 deterministic 的，否则为 not deterministic。

SQL security definer 声明了这个过程的调用者的权限为 definer，只有 definer 账户才能够调用这个过程。

5. 关于函数

观察 sakila-schema.sql 里面的 get_customer_balance 函数，结合官方文档分析：

(1) 这个函数 get_customer_balance 实现了什么功能？返回值是什么？

该函数返回客户账户的欠款金额。返回值是 **decimal(5,2)**也就是欠款金额。

- (2) 这个函数体中用到了 3 个 Mysql 内置函数，是哪几个函数？这 3 个函数的作用分别是什么？

函数	作用
IFNULL	若 expr1 非 NULL，则值为 expr1，否则为 expr2
SUM	求和
TO_DAYS	给定日期，返回从第零年开始计数的日期。

二、 创建新用户并分配权限

(截图语句和执行结果)

- (1) 执行命令新建 sakila_test 用户（密码 123456）；

```
mysql> create user 'sakila_another_test'@'localhost' identified by '123456';  
Query OK, 0 rows affected (0.02 sec)
```

- (2) 执行命令查看当前已有用户；

```
mysql> select user from mysql.user;  
+-----+  
| user                                     |  
+-----+  
| mysql.infoschema                       |  
| mysql.session                         |  
| mysql.sys                             |  
| root                                   |  
| sakila_another_test                   |  
| sakila_test                           |  
+-----+  
6 rows in set (0.00 sec)
```

- (3) 执行命令把 sakila 数据库的访问权限赋予 sakila_test 用户；

```
mysql> show grants for 'sakila_another_test'@'localhost';
+-----+
| Grants for sakila_another_test@localhost |
+-----+
| GRANT USAGE ON *.* TO `sakila_another_test`@`localhost` |
+-----+
1 row in set (0.00 sec)
```

- (4) 切换到 sakila_test 用户，执行 `select * from film` 操作。

The screenshot shows a SQL client window with a query editor and a result grid. The query editor contains the following SQL statements:

```
1 • use sakila;
2 • select * from film;
```

The result grid displays the following data:

film_id	title	description	release_year	language_id	original_language_id	rental_duration
1	ACADEMY DINOSAUR	I am movie no.01	2006	1	HULL	6
2	ACE GOLDFINGER	A Astounding Epistle of a Database Administrat...	2006	1	HULL	3
3	ADAPTATION HOLES	A Astounding Reflection of ; A Astounding Epistle of a Database Administrator And a Explorer who must Find a Car in Ancient C				
4	AFFAIR PREJUDICE	A Fanciful Documentary of a Frisbee And a Lum...	2006	1	HULL	5
5	AFRICAN EGG	A Fast-Paced Documentary of a Pastry Chef An...	2006	1	HULL	6
6	AGENT TRUMAN	A Intrepid Panorama of a Robot And a Boy who...	2006	1	HULL	3

The output pane shows the following actions and messages:

#	Time	Action	Message
17	09:31:50	use sakila	Error Code: 1044. Access denied for user 'sakila_another_test'@'localhost' to da
18	09:31:50	use sakila	Error Code: 1044. Access denied for user 'sakila_another_test'@'localhost' to da
19	09:31:50	use sakila	Error Code: 1044. Access denied for user 'sakila_another_test'@'localhost' to da
20	09:31:51	use sakila	Error Code: 1044. Access denied for user 'sakila_another_test'@'localhost' to da
21	09:32:02	use sakila	0 row(s) affected
22	09:32:02	select * from film LIMIT 0, 1000	1000 row(s) returned

三、设计并实现

根据应用场景，为 Sakila 数据库合理地设计并实现：

（截图语句和执行结果）

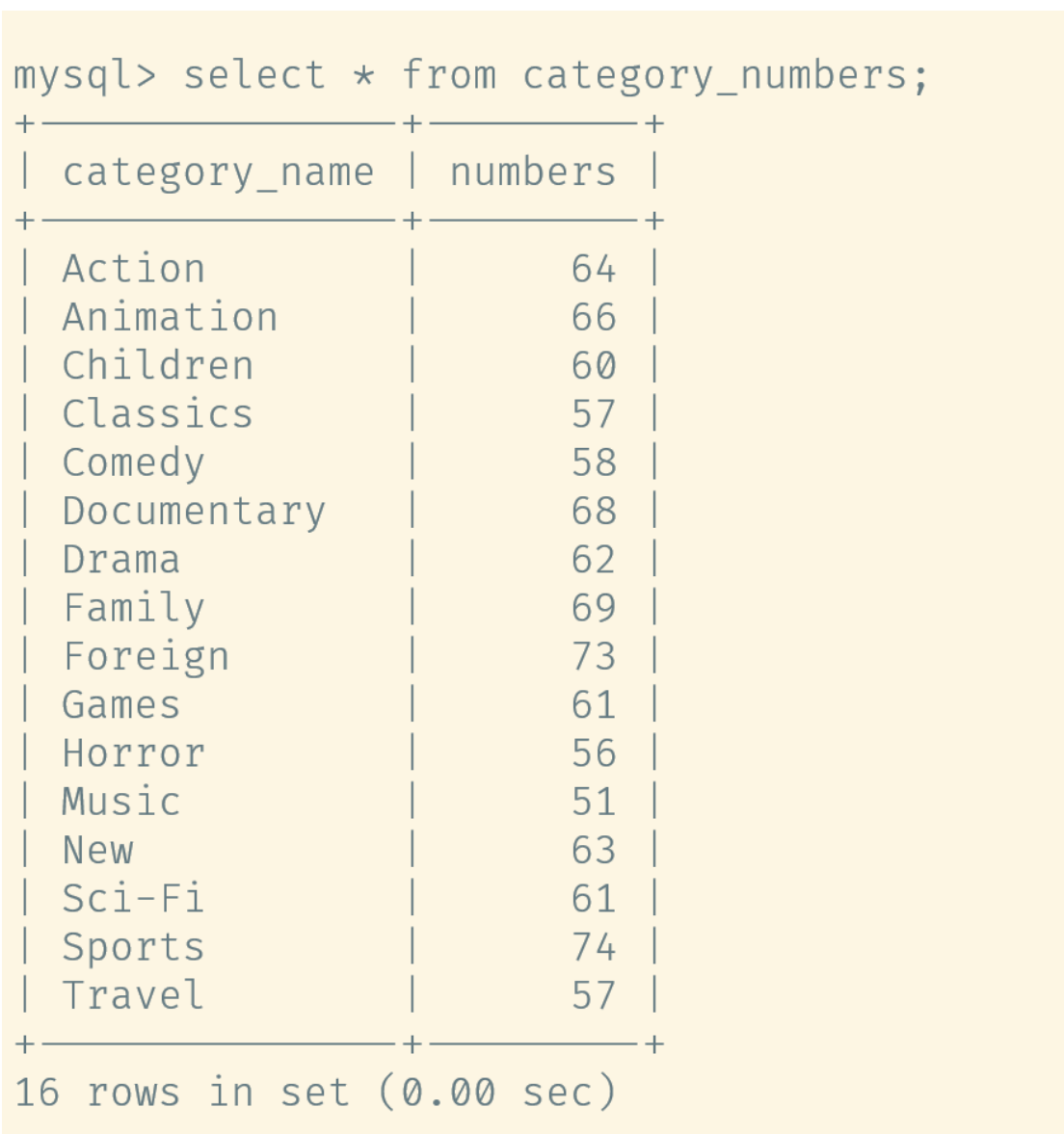
- 设计 1 个视图，至少关联 2 个表；

(1) 执行新建视图的语句，并截图 SQL 和执行结果：



```
use sakila;
drop view if exists category_numbers;
create view category_numbers
as
select c.name as category_name, count(*) as numbers
from category as c, film_category as fc
where c.category_id = fc.category_id
group by c.name;
```

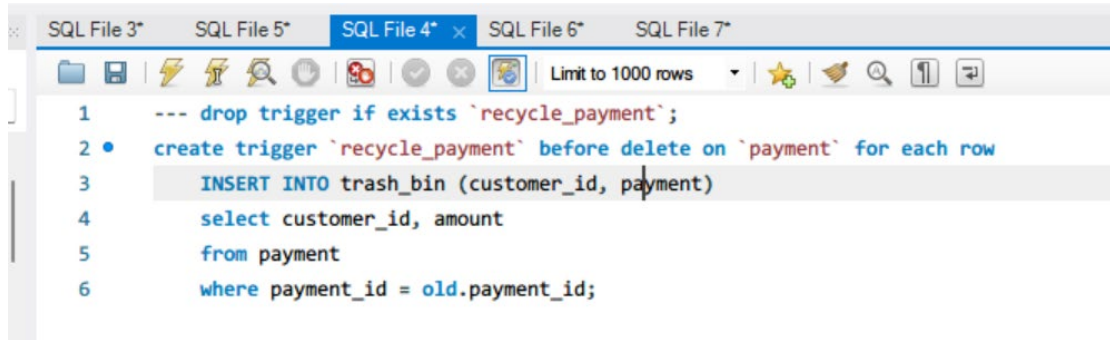
(2) 执行 `select * from [视图名]`, 截图执行结果:



```
mysql> select * from category_numbers;
+-----+-----+
| category_name | numbers |
+-----+-----+
| Action        | 64      |
| Animation     | 66      |
| Children      | 60      |
| Classics      | 57      |
| Comedy        | 58      |
| Documentary   | 68      |
| Drama         | 62      |
| Family        | 69      |
| Foreign       | 73      |
| Games         | 61      |
| Horror        | 56      |
| Music         | 51      |
| New           | 63      |
| Sci-Fi        | 61      |
| Sports        | 74      |
| Travel        | 57      |
+-----+-----+
16 rows in set (0.00 sec)
```

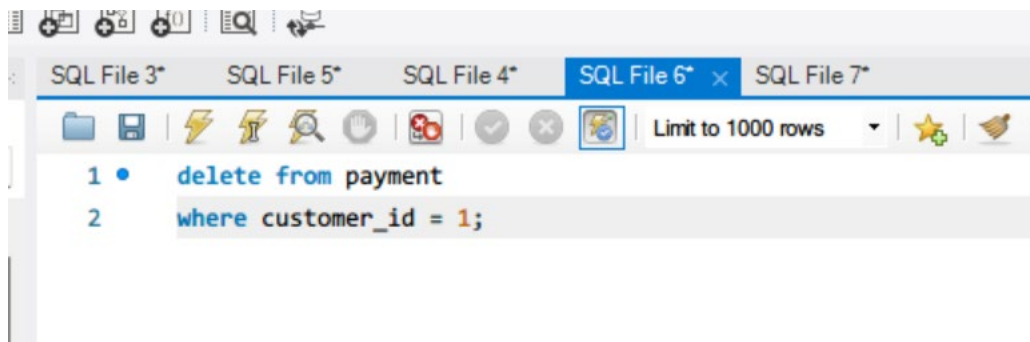
2. 设计 1 个触发器, 需要体现触发器生效。

(1) 执行新建触发器的语句, 并截图 SQL 和执行结果:

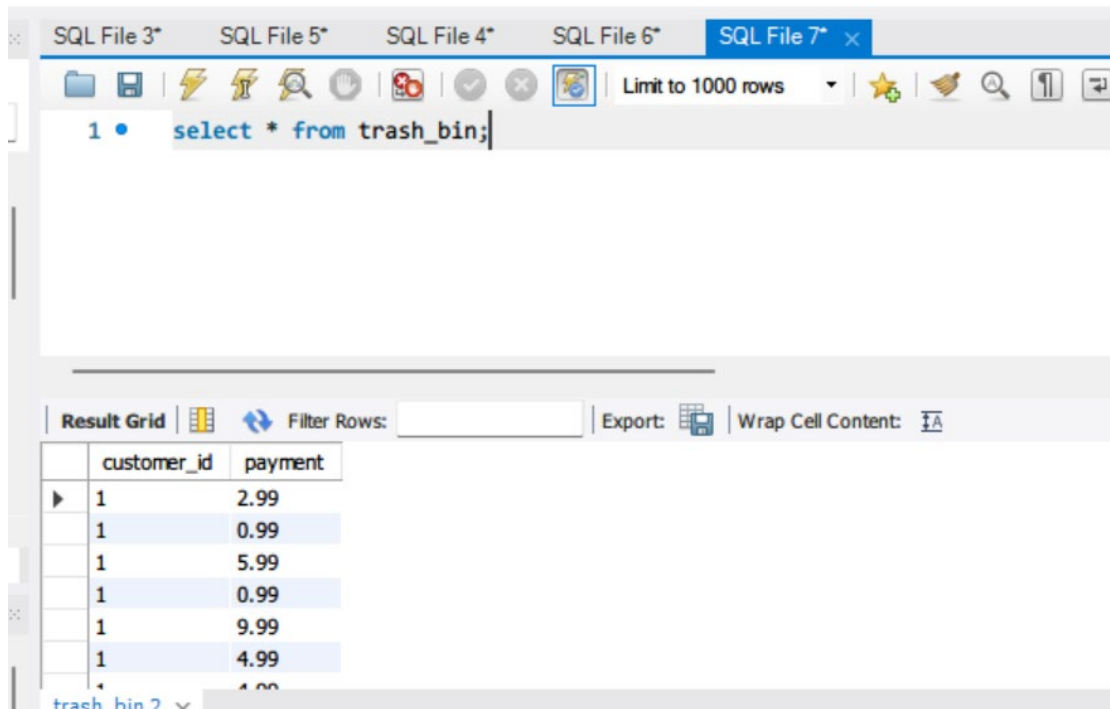


```
1 --- drop trigger if exists `recycle_payment`;  
2 • create trigger `recycle_payment` before delete on `payment` for each row  
3   INSERT INTO trash_bin (customer_id, payment)  
4   select customer_id, amount  
5   from payment  
6   where payment_id = old.payment_id;
```

(2) 验证触发器是否生效，截图验证过程：



```
1 • delete from payment  
2   where customer_id = 1;
```



```
1 • select * from trash_bin;
```

	customer_id	payment
▶	1	2.99
	1	0.99
	1	5.99
	1	0.99
	1	9.99
	1	4.99
	1	4.99

3. 设计 1 个存储过程，需要调用该存储过程。

(1) 执行新建存储过程的语句，并截图 SQL 和执行结果：

The screenshot displays a SQL IDE interface with four tabs: 'SQL File 3*', 'SQL File 5*', 'SQL File 8*', and 'SQL File 5*'. The active tab shows a SQL script for creating a stored procedure named 'total_tips'. The script includes a delimiter, a CREATE PROCEDURE statement with parameters 'p_customer_id int' and 'p_tips float', a 'reads sql data' comment, a BEGIN block containing a SELECT query, and an END statement. The output pane below shows a sequence of actions: dropping the procedure, creating it, calling it with arguments (259, @total_tips), selecting the result with a LIMIT, dropping it again, and finally creating it once more.

```
1 delimiter //
```

```
2 create procedure total_tips (
```

```
3     in p_customer_id int
```

```
4     , out p_tips float
```

```
5 )
```

```
6 reads sql data
```

```
7 begin
```

```
8     select sum(amount)
```

```
9     from payment
```

```
10    where customer_id = p_customer_id
```

```
11    and rental_id is null
```

```
12    group by customer_id
```

```
13    into p_tips;
```

```
14 end //
```

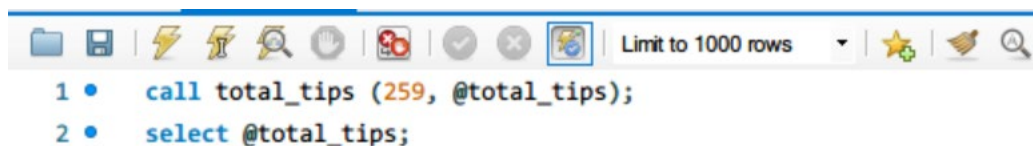
```
15 delimiter :
```

Output

Action Output

#	Time	Action
21	11:59:50	drop procedure total_tips
22	11:59:52	create procedure total_tips (in p_customer_id int , out p_tips float) reads sql data ...
23	11:59:55	call total_tips (259, @total_tips)
24	11:59:55	select @total_tips LIMIT 0, 1000
25	12:00:11	drop procedure total_tips
26	12:00:13	create procedure total_tips (in p_customer_id int , out p_tips float) reads sql data ...

(2) 调用该存储过程，截图调用结果：



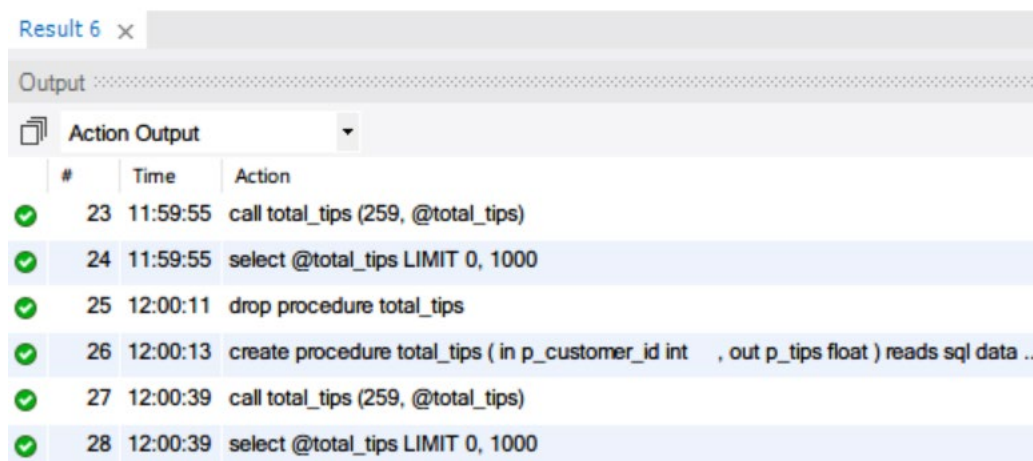
The screenshot shows a toolbar with various icons for file operations, execution, and search. Below the toolbar, the following SQL queries are entered:

```
1 • call total_tips (259, @total_tips);
2 • select @total_tips;
```



The 'Result Grid' displays the output of the SQL query. It has a toolbar with 'Filter Rows', 'Export', and 'Wrap Cell Content' options. The table contains one column named '@total_tips' and one row with the value '1.9900000095367432'.

@total_tips
1.9900000095367432



The 'Action Output' window shows a log of database actions. Each entry includes a status icon, a sequence number, a timestamp, and the action performed.

#	Time	Action
✓ 23	11:59:55	call total_tips (259, @total_tips)
✓ 24	11:59:55	select @total_tips LIMIT 0, 1000
✓ 25	12:00:11	drop procedure total_tips
✓ 26	12:00:13	create procedure total_tips (in p_customer_id int , out p_tips float) reads sql data ...
✓ 27	12:00:39	call total_tips (259, @total_tips)
✓ 28	12:00:39	select @total_tips LIMIT 0, 1000

四、思考题

(这部分不是必做题，供有兴趣的同学思考)

在阿里开发规范里有一条“【强制】不得使用外键与级联，一切外键概念必须在应用层解决。”请分析一下原因。你认为外键是否没有存在的必要？