实验二报告

一、 观察并回答问题

1. 观察 sakila.mwb 并回答问题

（1） 图中共有几个 View?

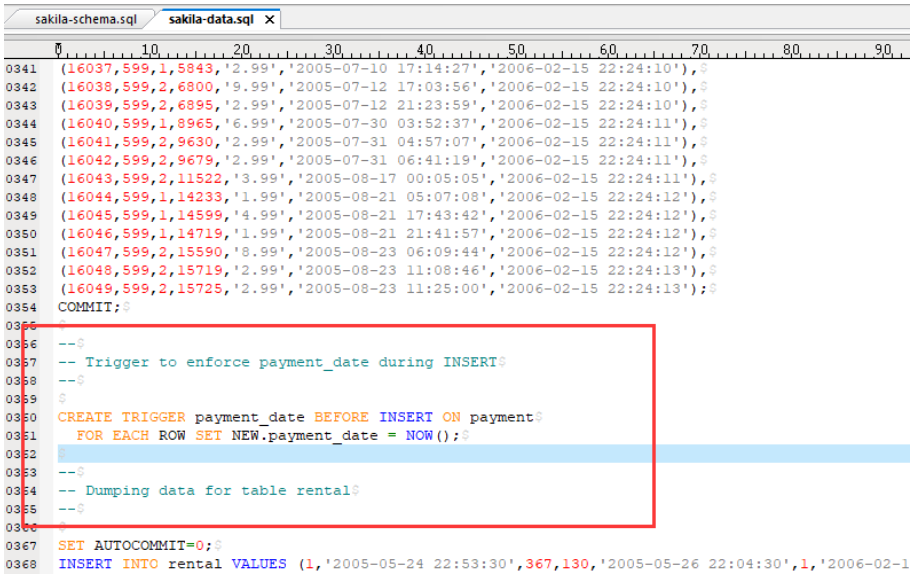
图中共有 7 个视图

（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, film_category, film_actor	显示演员及其参演电影信息
film_list	film, film_category, film_actor,, film_category, film_actor, actor, category	显示每一部电影的片名、描述、分类、评分、演员等具体信息
sales_by_store	payment, rental, inventory, store, address, city, country, staff	显示商店名，位置，经理和销售额

2. 观察 sakila-schema.sql 和 sakila-data.sql 并回答问题

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的语句：



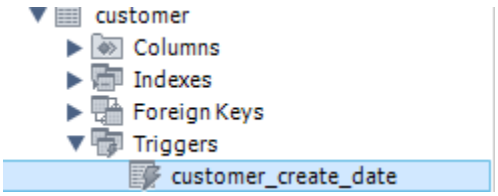
请同学们思考，这个触发器是否可以移到 sakila-schema.sql 里去执行？为什么？

不行，因为 sakila-schema.sql 只负责建立表的结构，数据的导入与它无关，如果触发器移到 sakila-schema.sql 里面，触发器不会触发

3. 观察数据库的触发器 customer_create_date 并回答问题

（1） customer_create_date 触发器建在哪个表上？

在 customer 这个表上



（2） 这个触发器实现什么功能？

```
1  CREATE
2  DEFINER=`paul`@`localhost`
3  TRIGGER `sakila`.`customer_create_date`
4  BEFORE INSERT ON `sakila`.`customer`
5  FOR EACH ROW
6  SET NEW.create_date = NOW()
```

这个触发器的功能是在插入一条记录时，将创建的日期自动置为当前执行操作时的日期。

（3） 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）

```
insert into customer
values(601,1,'Alisan','Lisa','Alisan.Lisa@sakilacustomer.org',71,1,null,null);
```

601	1	Alisan	Lisa	Alisan.Lisa@sakilacustomer.org	71	1	2021-09-19 19:52:01	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

二、 设计并实现

根据应用场景，为 Sakila 数据库合理地设计并实现：

（注意：请将创建语句、调用结果截图记录到实验报告里）

1. 设计 1 个视图，至少关联 3 个表；

视图功能：显示每个客户每次的租借信息，包括用户名，电影片名，租借时间，归还时间和借还间隔。

创建语句：

```
CREATE
    ALGORITHM=UNDEFINED
    DEFINER=`root`@`localhost`
    SQL SECURITY DEFINER
    VIEW `customer_rental` As
    select
        `c`.`customer_id` As `ID`,
        concat(`c`.`first_name`,_utf8mb4' ',`c`.`last_name`) AS `name`,
```

```
    `film`.`title` As `title`,
    `r`.`rental_date` As `租借日期`,
    `r`.`return_date` As `归还日期`,
    (timestampdiff(day,`r`.`rental_date`,`r`.`return_date`)) As `租借时间`
from
    ((`customer` `c`
    join `rental` `r` on((`c`.`customer_id` = `r`.`customer_id` ))
    join `inventory` on ((`r`.`inventory_id` = `inventory`.`inventory_id` ))
    join film on ((`film`.`film_id` = `inventory`.`film_id`)) ))
```

视图：

ID	name	title	租借日期	归还日期	租借时间
1	MARY SMITH	TALENTED HOMICIDE	2005-05-28 10:35:23	2005-06-03 06:32:23	5
1	MARY SMITH	UNFORGIVEN ZOOLANDER	2005-08-21 23:33:57	2005-08-23 01:30:57	1
1	MARY SMITH	USUAL UNTOUCHABLES	2005-07-28 16:18:23	2005-07-30 17:56:23	2
1	MARY SMITH	WOMEN DORADO	2005-07-28 19:20:07	2005-07-29 22:54:07	1
1	MARY SMITH	YOUTH KICK	2005-06-21 06:24:45	2005-06-28 03:28:45	6
1	MARY SMITH	PATIENT SISTER	2005-08-19 09:55:16	2005-08-20 14:44:16	1
1	MARY SMITH	PATIENT SISTER	2005-05-25 11:30:37	2005-06-03 12:00:37	9
1	MARY SMITH	RACER EGG	2005-08-02 15:36:52	2005-08-10 16:40:52	8
1	MARY SMITH	SATURDAY LAMBS	2005-07-08 07:33:56	2005-07-12 13:25:56	4
1	MARY SMITH	SAVANNAH TOWN	2005-06-18 13:33:59	2005-06-19 17:40:59	1
1	MARY SMITH	SNATCH SLIPPER	2005-07-09 13:24:07	2005-07-14 14:01:07	5
1	MARY SMITH	ADAPTATION HOLES	2005-08-01 08:51:04	2005-08-10 12:12:04	9
1	MARY SMITH	AMISTAD MIDSUMMER	2005-07-29 03:58:49	2005-08-01 05:16:49	3
1	MARY SMITH	ATTACKS HATE	2005-06-18 08:41:48	2005-06-22 03:36:48	3
1	MARY SMITH	BIKINI BORROWERS	2005-08-22 20:03:46	2005-08-30 01:51:46	7

2. 设计 1 个触发器，需要在报告里体现触发器生效。

新建一张表 customer_history，用来存储从 customer 中删掉的用户信息，触发器 customer_delete 在 customer 表上。当 customer 表中删除一条记录时，触发器自动向 customer_history 中插入相同的记录，删除的时间为操作执行时间。

customer_history 表创建语句：

```
CREATE TABLE `customer_history` (
    `customer_id` smallint unsigned NOT NULL ,
    `store_id` tinyint unsigned NOT NULL,
    `first_name` varchar(45) NOT NULL,
    `last_name` varchar(45) NOT NULL,
    `email` varchar(50) DEFAULT NULL,
    `address_id` smallint unsigned NOT NULL,
    `active` tinyint(1) NOT NULL DEFAULT '1',
    `delete_date` datetime NOT NULL,
    `last_update` timestamp NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP,
    PRIMARY KEY (`customer_id`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8mb4;
```

customer_delete 触发器创建语句：

```
DELIMITER ;;
CREATE TRIGGER
`delete_customer` AFTER DELETE ON `customer` FOR EACH ROW BEGIN
    INSERT INTO customer_history(customer_id, store_id, first_name, last_name, email, address_id,active,delete_date,
        last_update)
        VALUES(old.customer_id, old.store_id, old.first_name, old.last_name, old.email, old.address_id,old.active,now(),
        old.last_update);
END ;;
DELIMITER ;
```

用实验 1 中的第 11 问和第 13 问：

向 customer 中插入第 600 号客户

```
insert into address
values(606,'1329 Fukuyama Street',null,'Heilongjiang',537,27107,288241215399,st_geomfromtext('POINT (128.044975346.9804391)'),now());
insert into customer
values(600,1,"Luis","Sword",'Sword.Luis@sakilacustomer.org',606,1,now(),now());
```

然后删除此用户

```
delete from customer
where customer_id = 600;
delete from address
where address_id = 606;
```

customer_history 中自动插入了此用户信息

```
SELECT * FROM sakila.customer_history;
```

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

customer_id	store_id	first_name	last_name	email	address_id	active	delete_date	last_update
600	1	Luis	Sword	Sword.Luis@sakilacustomer.org	606	1	2021-09-19 20:...	2021-09-19 20:03:17
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

三、 思考题

（这部分不是必做题，供有兴趣的同学思考）

如果可以给这个数据库做修改，你想修改哪个地方？指出并说明原因。

MySQL 新手，还不了解数据库，用着挺顺手，功能还没探索完全，是在没有资格谈论这个数据库有什么不足。