

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?

7

(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	列出所有演员及他们出演的电影
film_list	film, category, film_category, actor, film_actor	列出所有电影及参加演出的演员
sales_by_store	city, country, payment, rental, inventory, store, address, staff	提供所有商店的销售情况，以及商店的位置和经理名字

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：

```

sakila-schema.sql sakila-data.sql X
0 10 20 30 40 50 60 70 80 90
0341 (16037,599,1,5843,'2.99','2005-07-10 17:14:27','2006-02-15 22:24:10'),
0342 (16038,599,2,6800,'9.99','2005-07-12 17:03:56','2006-02-15 22:24:10'),
0343 (16039,599,2,6895,'2.99','2005-07-12 21:23:59','2006-02-15 22:24:10'),
0344 (16040,599,1,8965,'6.99','2005-07-30 03:52:37','2006-02-15 22:24:11'),
0345 (16041,599,2,9630,'2.99','2005-07-31 04:57:07','2006-02-15 22:24:11'),
0346 (16042,599,2,9679,'2.99','2005-07-31 06:41:19','2006-02-15 22:24:11'),
0347 (16043,599,2,11522,'3.99','2005-08-17 00:05:05','2006-02-15 22:24:11'),
0348 (16044,599,1,14233,'1.99','2005-08-21 05:07:08','2006-02-15 22:24:12'),
0349 (16045,599,1,14599,'4.99','2005-08-21 17:43:42','2006-02-15 22:24:12'),
0350 (16046,599,1,14719,'1.99','2005-08-21 21:41:57','2006-02-15 22:24:12'),
0351 (16047,599,2,15590,'8.99','2005-08-23 06:09:44','2006-02-15 22:24:12'),
0352 (16048,599,2,15719,'2.99','2005-08-23 11:08:46','2006-02-15 22:24:13'),
0353 (16049,599,2,15725,'2.99','2005-08-23 11:25:00','2006-02-15 22:24:13');
0354 COMMIT;
0355
0356 --
0357 -- Trigger to enforce payment_date during INSERT
0358 --
0359
0360 CREATE TRIGGER payment_date BEFORE INSERT ON payment
0361 FOR EACH ROW SET NEW.payment_date = NOW();
0362
0363 --
0364 -- Dumping data for table rental
0365 --
0366
0367 SET AUTOCOMMIT=0;
0368 INSERT INTO rental VALUES (1,'2005-05-24 22:53:30',367,130,'2005-05-26 22:04:30',1,'2006-02-1

```

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？

该语句设定了一个触发器用于在每次插入 payment 表之前设定 date 为 Now()，但在这条语句之前首先需要插入 payment 表中许多初始数据，脚本希望这部分数据的 date 属性不被触发器影响，所以将该语句置于 sakila-schema.sql 中执行，将其放置在插入初始数据之后，从而不影响初始数据。

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？
Customer
- (2) 这个触发器实现什么功能？
在每次插入 Customer 表之前设定 date 属性为 Now()
- (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）

```

1 • insert into customer(customer_id, store_id, first_name, last_name, email, address_id, active)
2   values (666, 1, 'Yikai', 'Li', 'xxxx@example.com', 3, 1);
3 • select * from customer order by last_update desc;

```

	customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
▶	666	1	Yikai	Li	xxxx@example.com	3	1	2021-09-18 11:06:19	2021-09-18 11:06:19
	1	1	MARY	SMITH	MARY.SMITH@sakilacustomer.org	5	1	2006-02-14 22:04:36	2006-02-15 04:57:20
	2	1	PATRICIA	JOHNSON	PATRICIA.JOHNSON@sakilacustomer.org	6	1	2006-02-14 22:04:36	2006-02-15 04:57:20
	3	1	LINDA	WILLIAMS	LINDA.WILLIAMS@sakilacustomer.org	7	1	2006-02-14 22:04:36	2006-02-15 04:57:20
	4	2	BARBARA	JONES	BARBARA.JONES@sakilacustomer.org	8	1	2006-02-14 22:04:36	2006-02-15 04:57:20

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；
视图创建代码如下：

```

create VIEW rental_list
as
select rental.rental_id as id, film.title as title, CONCAT(staff.first_name, _utf8mb4' ',
staff.last_name) as staff_name,
      CONCAT(customer.first_name, _utf8mb4' ', customer.last_name) as customer_name,
payment.amount as rent, rental.rental_date as rental_date,
      rental.return_date as return_date
from rental join inventory on rental.inventory_id=inventory.inventory_id join customer on
rental.customer_id=customer.customer_id join film
      on inventory.film_id=film.film_id join payment on rental.rental_id=payment.rental_id join
staff on rental.staff_id=staff.staff_id
order by rental_date;

```

上述语句创建一个 **rental_list** 视图将每次租借的租借号，电影名，负责的员工名，租借人名，租金，租出时间和归还时间链接成一个视图。

创建视图完成后查询结果如下：

	id	title	staff_name	customer_name	rent	rental_date	return_date
▶	1	BLANKET BEVERLY	Mike Hillyer	CHARLOTTE HUNTER	2.99	2005-05-24 22:53:30	2005-05-26 22:04:30
	2	FREAKY POCUS	Mike Hillyer	TOMMY COLLAZO	2.99	2005-05-24 22:54:33	2005-05-28 19:40:33
	3	GRADUATE LORD	Mike Hillyer	MANUEL MURRELL	3.99	2005-05-24 23:03:39	2005-06-01 22:12:39
	4	LOVE SUICIDES	Jon Stephens	ANDREW PURDY	4.99	2005-05-24 23:04:41	2005-06-03 01:43:41
	5	IDOLS SNATCHERS	Mike Hillyer	DELORES HANSEN	6.99	2005-05-24 23:05:21	2005-06-02 04:33:21

9 • select * from rental_list

- 设计 1 个触发器，需要在报告里体现触发器生效。

触发器设置代码如下：

```

DELIMITER ;;
CREATE DEFINER = CURRENT_USER TRIGGER `sakila`.`customer_AFTER_UPDATE` AFTER
UPDATE ON `customer` FOR EACH ROW
BEGIN
    if(old.customer_id != new.customer_id)
    then update rental set rental.customer_id=new.customer_id
    where rental.customer_id=old.customer_id;
    end if;
END;;

```

触发器功能为在 **customer** 表中 **customer_id** 发生更新后，相应地更新 **rental** 表中的 **customer_id**。

执行下述代码进行验证

```

select * from rental where customer_id=10;
update customer set customer.customer_id=999
where customer.customer_id=10;
select * from rental where customer_id=10;
select * from rental where customer_id=999;

```

下图为更新 customer 表之前在 rental 表中查询 customer_id=10 得到的结果

	rental_id	rental_date	inventory_id	customer_id	return_date	staff_id	last_update
▶	1140	2005-05-31 19:36:30	3731	10	2005-06-07 18:33:30	2	2006-02-15 21:30:53
	1801	2005-06-16 20:21:53	1015	10	2005-06-18 23:18:53	1	2006-02-15 21:30:53
	1995	2005-06-17 11:11:14	1866	10	2005-06-26 16:37:14	1	2006-02-15 21:30:53
	2222	2005-06-18 03:26:23	3834	10	2005-06-26 08:50:23	2	2006-02-15 21:30:53
	2814	2005-06-19 20:01:59	1783	10	2005-06-26 01:28:59	2	2006-02-15 21:30:53

25 row(s) returned

下图为更新 customer 表之后查询 customer_id=10 得到的结果，此时因为 customer 表中 customer_id 的更新导致 rental 表中 customer_id=10 的记录也被更新。

0 row(s) returned

下图为查询 rental 表 customer_id=999 得到的结果

rental_id	rental_date	inventory_id	customer_id	return_date	staff_id	last_update
1140	2005-05-31 19:36:30	3731	999	2005-06-07 18:33:30	2	2006-02-15 21:30:53
1801	2005-06-16 20:21:53	1015	999	2005-06-18 23:18:53	1	2006-02-15 21:30:53
1995	2005-06-17 11:11:14	1866	999	2005-06-26 16:37:14	1	2006-02-15 21:30:53
2222	2005-06-18 03:26:23	3834	999	2005-06-26 08:50:23	2	2006-02-15 21:30:53
2814	2005-06-19 20:01:59	1783	999	2005-06-26 01:28:59	2	2006-02-15 21:30:53

25 row(s) returned

可以看到此时查询得到的结果和之前查询 customer_id=10 的结果相同。

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。