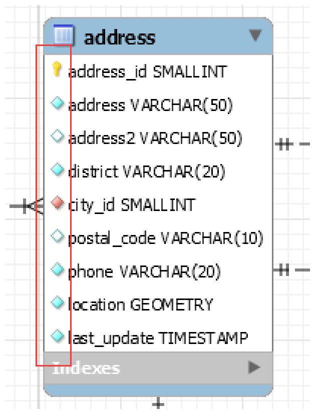
实验一报告

一、 回答问题

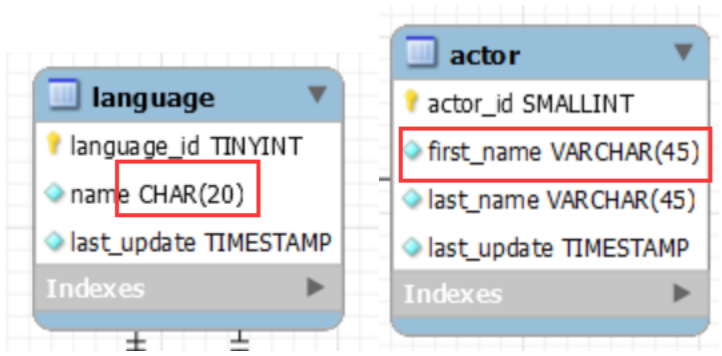
请一边熟悉 sakila 数据库，一边回答以下问题：

1. sakila.mwb 模型中，表结构里每个字段前面的小标记分别表示什么意思？（观察字段的属性）



标记	意义
🔑	主键字段
🔷	普通字段
◇	可以为空的字段
♦	外键字段

2. char 和 varchar 类型的区别是什么？



char 是固定长度的变量，varchar 是可变长度的变量。

3. 图中哪部分体现影片-演员关系？换句话说，如果要找出演某个影片的演员名

字，访问哪几张表可以获得信息？

film，actor，film_actor

4. 如果已知某个顾客姓名，要找到他租借的所有影片名，需要访问哪几张表？

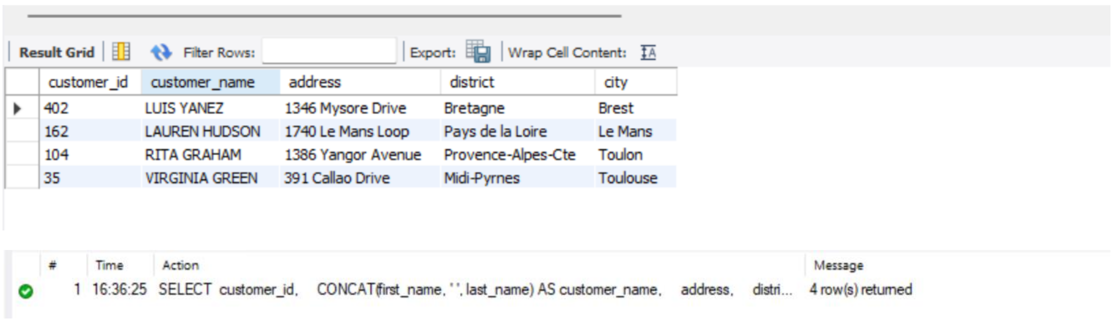
customer，rental，inventory，film

二、 实验截图

（请注意粘贴文本格式的 SQL 语句，截图执行结果和 Output 窗口）

1、 请列出所有 country 是“France”的客户的信息，显示 customer_id、客户姓名、地址、所在区域，所在城市（注意：客户姓名请以 first_name+空格+last_name 的格式，例如：SISSY SOBIESKI）;

```
SELECT
    customer_id,
    CONCAT(first_name, ' ', last_name) AS customer_name,
    address,
    district,
    city
FROM
    customer
    JOIN address USING (address_id)
    JOIN city USING (city_id)
    JOIN country USING (country_id)
WHERE
    country = "France";
```

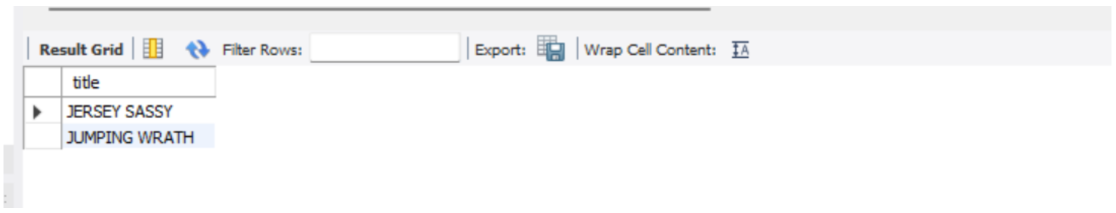


customer_id	customer_name	address	district	city
402	LUIS YANEZ	1346 Mysore Drive	Bretagne	Brest
162	LAUREN HUDSON	1740 Le Mans Loop	Pays de la Loire	Le Mans
104	RITA GRAHAM	1386 Yangor Avenue	Provence-Alpes-Cte	Toulon
35	VIRGINIA GREEN	391 Callao Drive	Midi-Pyrnes	Toulouse

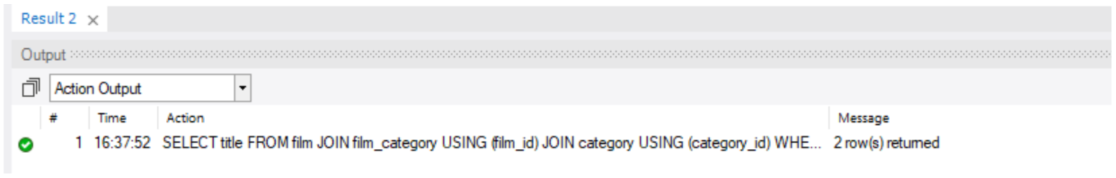
```
#   Time      Action                                                                          Message
1   16:36:25  SELECT  customer_id,   CONCAT(first_name, ' ', last_name) AS customer_name,   address,   distri...   4 row(s) returned
```

2、　列出属于“Children”类型并以“J”开头的电影名；

```
SELECT
    title
FROM
    film
    JOIN film_category USING (film_id)
    JOIN category USING (category_id)
WHERE
    name = 'Children' AND LEFT(title, 1) = 'J';
```



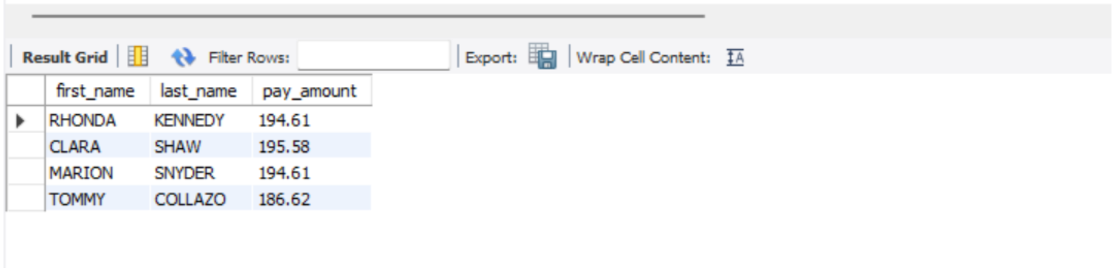
title
JERSEY SASSY
JUMPING WRATH



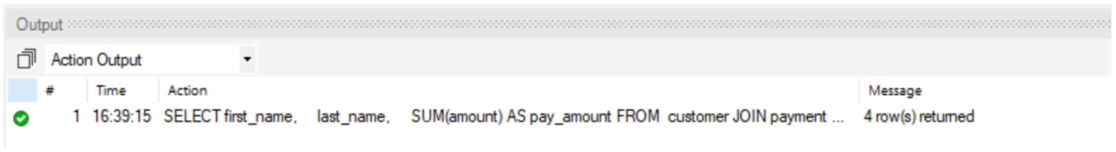
```
#  Time      Action                                                                          Message
1  16:37:52  SELECT title FROM film JOIN film_category USING (film_id) JOIN category USING (category_id) WHE...  2 row(s) returned
```

3、　找出费用在 180 至 200 之间的客户，列出他们的 first_name, last_name 和每个人花费的金额；

```
SELECT
    first_name,
    last_name,
    SUM(amount) AS pay_amount
FROM
    customer
    JOIN payment USING (customer_id)
GROUP BY
    customer_id
HAVING
    pay_amount > 180 AND pay_amount < 200;
```



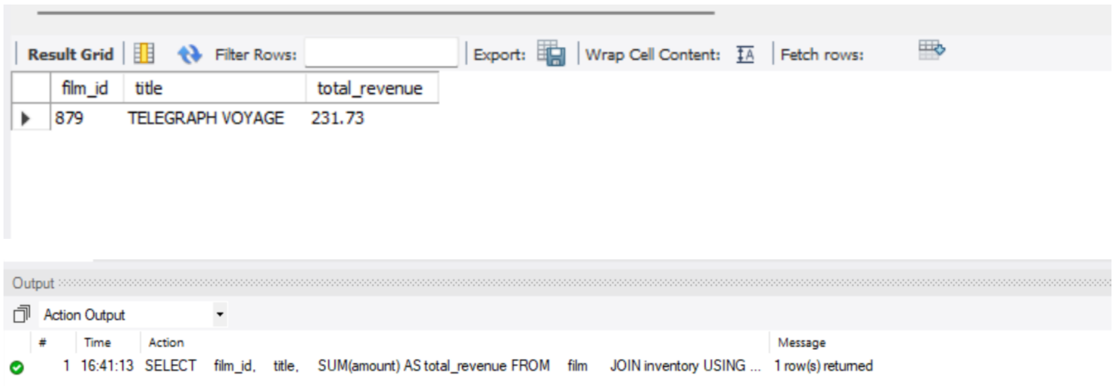
first_name	last_name	pay_amount
RHONDA	KENNEDY	194.61
CLARA	SHAW	195.58
MARION	SNYDER	194.61
TOMMY	COLLAZO	186.62



```
#  Time      Action                                                                      Message
1  16:39:15  SELECT first_name,   last_name,   SUM(amount) AS pay_amount FROM  customer JOIN payment ...  4 row(s) returned
```

4、　哪个影片获得了总体最高的租金？请列出影片 id、影片名、总租金；

```
SELECT
    film_id,
    title,
    SUM(amount) AS total_revenue
FROM
    film
    JOIN inventory USING (film_id)
    JOIN rental USING (inventory_id)
    JOIN payment USING (rental_id)
GROUP BY
    film_id
ORDER BY
    total_revenue DESC
LIMIT 1;
```

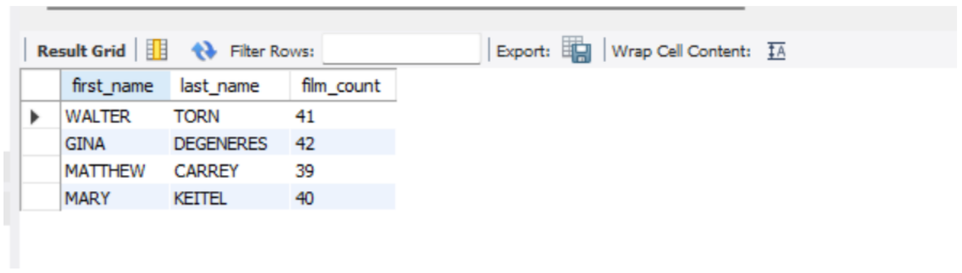


film_id	title	total_revenue
879	TELEGRAPH VOYAGE	231.73

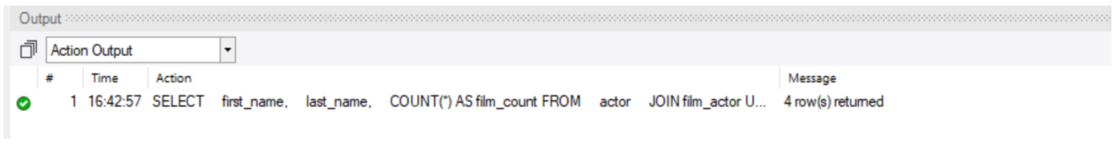
#	Time	Action	Message
1	16:41:13	SELECT film_id, title, SUM(amount) AS total_revenue FROM film JOIN inventory USING ...	1 row(s) returned

5、　哪些演员出演的电影超过 38 部？ 请列出演员名、出演的电影数；

```
SELECT
    first_name,
    last_name,
    COUNT(*) AS film_count
FROM
    actor
    JOIN film_actor USING (actor_id)
GROUP BY
    actor_id
HAVING
    COUNT(*) > 38;
```



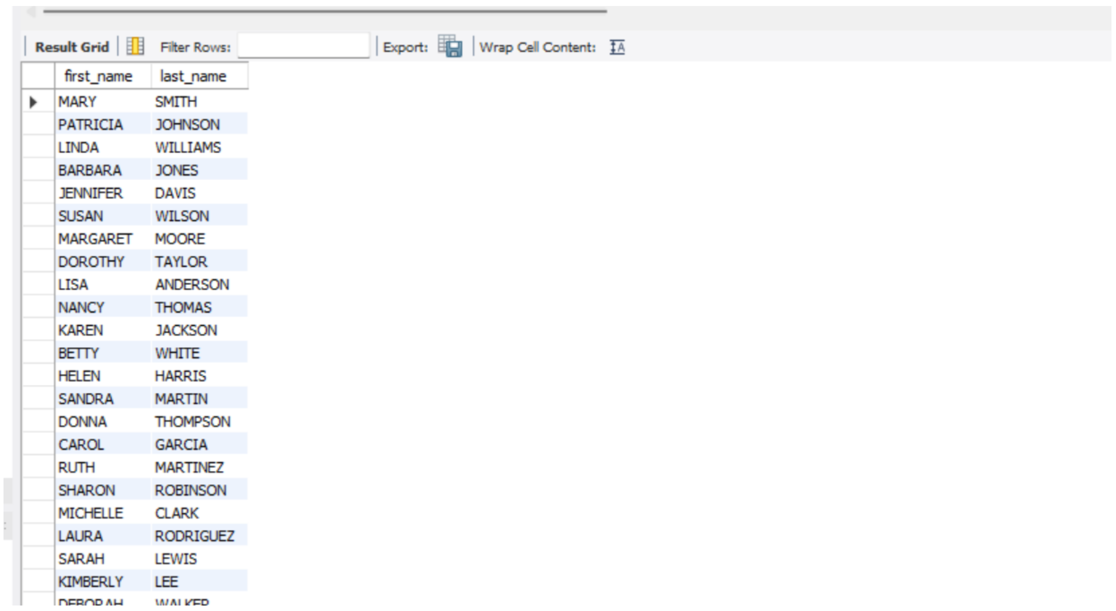
first_name	last_name	film_count
WALTER	TORN	41
GINA	DEGENERES	42
MATTHEW	CARREY	39
MARY	KEITEL	40



```
Output
Action Output
#   Time      Action                                                                                   Message
1   16:42:57  SELECT  first_name,  last_name,  COUNT(*) AS film_count FROM  actor  JOIN film_actor U...  4 row(s) returned
```

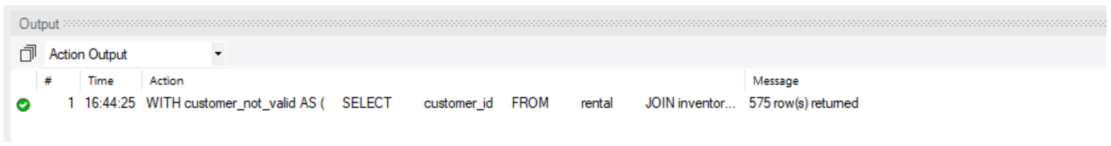
6、　请找出没有租借过电影《NATURAL STOCK》的顾客姓名；

```sql
WITH customer_not_valid AS (
    SELECT
        customer_id
    FROM
        rental
        JOIN inventory USING (inventory_id)
        JOIN film fi USING (film_id)
    WHERE
        fi.title = 'NATURAL STOCK'
)
SELECT
    first_name,
    last_name
FROM
    customer
WHERE
    customer_id NOT IN (
        SELECT customer_id
        FROM customer_not_valid
    );
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

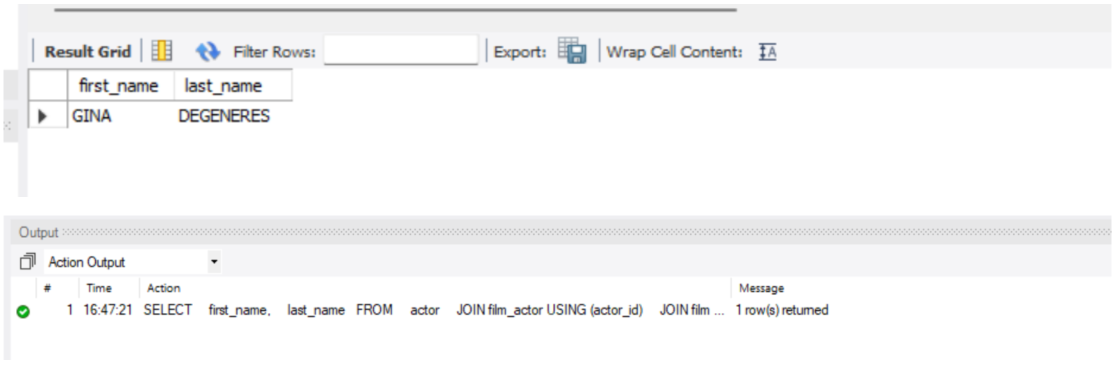
first_name	last_name
MARY	SMITH
PATRICIA	JOHNSON
LINDA	WILLIAMS
BARBARA	JONES
JENNIFER	DAVIS
SUSAN	WILSON
MARGARET	MOORE
DOROTHY	TAYLOR
LISA	ANDERSON
NANCY	THOMAS
KAREN	JACKSON
BETTY	WHITE
HELEN	HARRIS
SANDRA	MARTIN
DONNA	THOMPSON
CAROL	GARCIA
RUTH	MARTINEZ
SHARON	ROBINSON
MICHELLE	CLARK
LAURA	RODRIGUEZ
SARAH	LEWIS
KIMBERLY	LEE
DEBORAH	WALKER



```
Output
Action Output
#   Time      Action                                                                          Message
1   16:44:25  WITH customer_not_valid AS (  SELECT  customer_id  FROM  rental  JOIN inventor...  575 row(s) returned
```

7、　查询既演过《ELEPHANT TROJAN》又演过《DOGMA FAMILY》的演员，列出其姓名；

```
SELECT
    first_name,
    last_name
FROM
    actor
    JOIN film_actor USING (actor_id)
    JOIN film USING (film_id)
WHERE
    title IN ('ELEPHANT TROJAN', 'DOGMA FAMILY')
GROUP BY
    actor_id
HAVING
    COUNT(*) = 2;
```

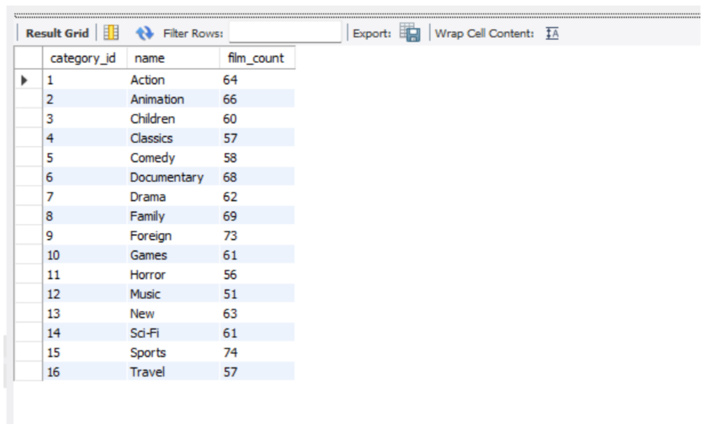


first_name	last_name
GINA	DEGENERES

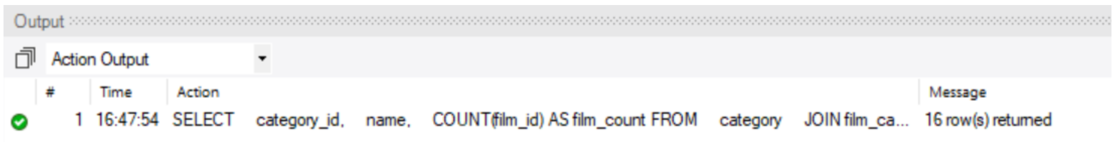
```
#   Time      Action                                                                 Message
1   16:47:21  SELECT first_name, last_name FROM actor JOIN film_actor USING (actor_id) JOIN film ...   1 row(s) returned
```

8、　统计每种类型的影片数，显示类型编号、类型名称、该类型影片数；

```
SELECT
    category_id,
    name,
    COUNT(film_id) AS film_count
FROM
    category
    JOIN film_category USING (category_id)
    JOIN film USING (film_id)
GROUP BY
    category_id;
```



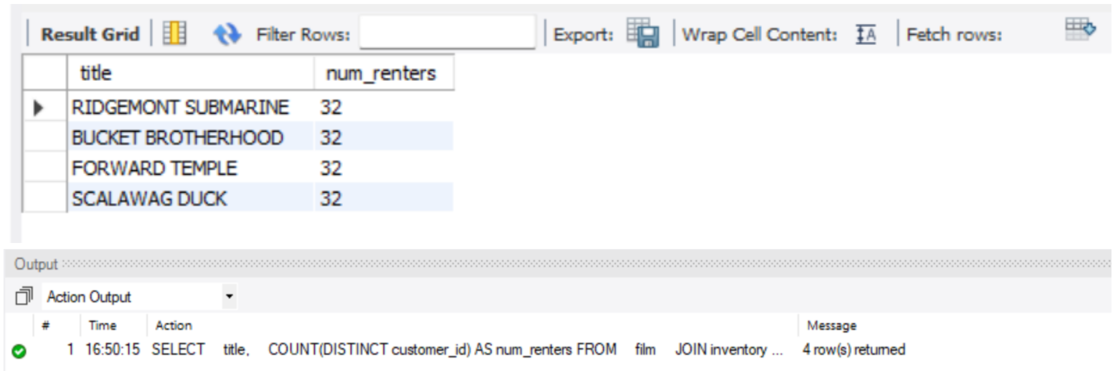
category_id	name	film_count
1	Action	64
2	Animation	66
3	Children	60
4	Classics	57
5	Comedy	58
6	Documentary	68
7	Drama	62
8	Family	69
9	Foreign	73
10	Games	61
11	Horror	56
12	Music	51
13	New	63
14	Sci-Fi	61
15	Sports	74
16	Travel	57



```
Output
Action Output
#  Time      Action                                                                                  Message
1  16:47:54  SELECT   category_id,   name,   COUNT(film_id) AS film_count FROM   category   JOIN film_ca...  16 row(s) returned
```

9、　找出最热门的（被最多不同人租借过）影片名，并显示租借人数；

```
SELECT
    title,
    COUNT(DISTINCT customer_id) AS num_renters
FROM
    film
    JOIN inventory USING (film_id)
    JOIN rental USING (inventory_id)
GROUP BY
    film_id
ORDER BY
    COUNT(DISTINCT customer_id) DESC
LIMIT 4;
```

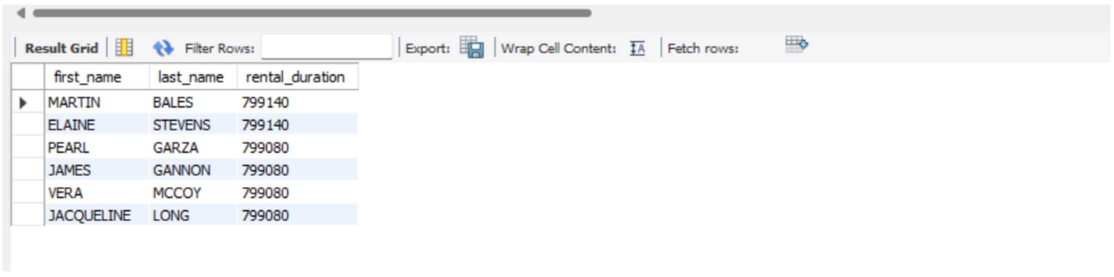


title	num_renters
RIDGEMONT SUBMARINE	32
BUCKET BROTHERHOOD	32
FORWARD TEMPLE	32
SCALAWAG DUCK	32

```
Output
Action Output
#  Time      Action                                                                                  Message
1  16:50:15  SELECT   title,   COUNT(DISTINCT customer_id) AS num_renters FROM   film   JOIN inventory ...  4 row(s) returned
```

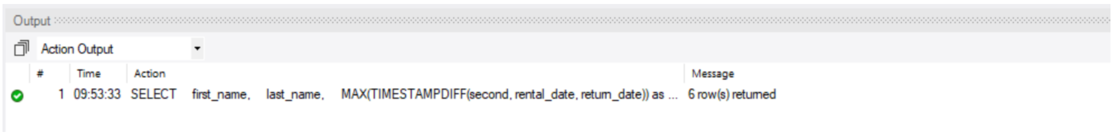
10、　查询单次租借影片时间最长的 6 位客户，列出其 first_name、last_name 和当次租借时长（单位秒）；

```
SELECT
    first_name,
    last_name,
    MAX(TIMESTAMPDIFF(second, rental_date, return_date)) as
rental_duration
FROM
    customer
    JOIN rental USING (customer_id)
GROUP BY
    customer_id
ORDER BY
    rental_duration DESC
LIMIT 6;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

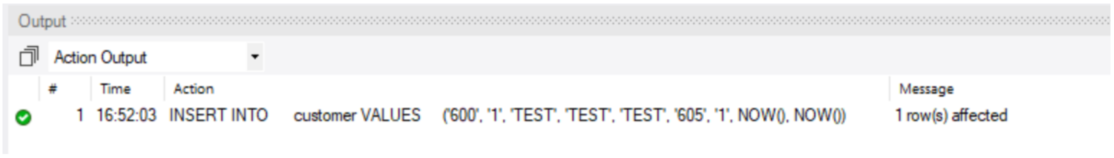
first_name	last_name	rental_duration
MARTIN	BALES	799140
ELAINE	STEVENS	799140
PEARL	GARZA	799080
JAMES	GANNON	799080
VERA	MCCOY	799080
JACQUELINE	LONG	799080



```
Output
Action Output
  #  Time      Action                                                                           Message
  1  09:53:33  SELECT  first_name,  last_name,  MAX(TIMESTAMPDIFF(second, rental_date, retum_date)) as ...  6 row(s) retumed
```

11、　在 customer 表中新增一条数据，注意 customer 表与其他表的关系；

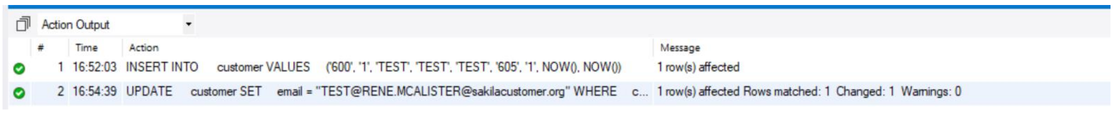
```
INSERT INTO
    customer
VALUES
    ('600', '1', 'TEST', 'TEST', 'TEST', '605', '1', NOW(), NOW())
```



```
Output
Action Output
  #  Time      Action                                                                            Message
  1  16:52:03  INSERT INTO   customer VALUES   ('600', '1', 'TEST', 'TEST', 'TEST', '605', '1', NOW(), NOW())   1 row(s) affected
```

12、　修改刚才在 customer 表中新增的那条数据；

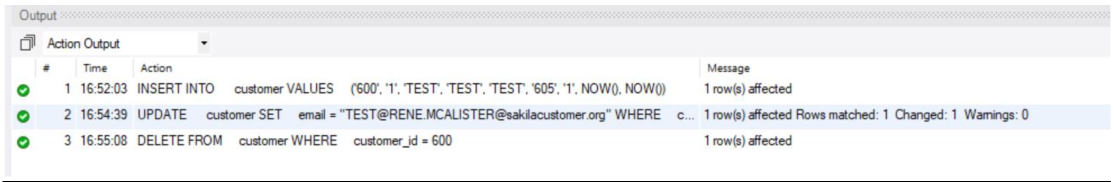
```
UPDATE
    customer
SET
    email = "TEST@RENE.MCALISTER@sakilacustomer.org"
WHERE
    customer_id = 600
```



```
Action Output
  #  Time      Action                                                                                   Message
  1  16:52:03  INSERT INTO   customer VALUES   ('600', '1', 'TEST', 'TEST', 'TEST', '605', '1', NOW(), NOW())   1 row(s) affected
  2  16:54:39  UPDATE   customer SET   email = "TEST@RENE.MCALISTER@sakilacustomer.org" WHERE   c...   1 row(s) affected Rows matched: 1  Changed: 1  Wamings: 0
```

13、　删除第 11 步新增的那条数据。

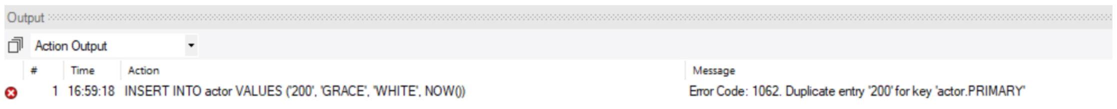
```
DELETE FROM
    customer
WHERE
    customer_id = 600
```



```
Output
Action Output
  #  Time      Action                                                                                   Message
  1  16:52:03  INSERT INTO   customer VALUES   ('600', '1', 'TEST', 'TEST', 'TEST', '605', '1', NOW(), NOW())   1 row(s) affected
  2  16:54:39  UPDATE   customer SET   email = "TEST@RENE.MCALISTER@sakilacustomer.org" WHERE   c...   1 row(s) affected Rows matched: 1  Changed: 1  Wamings: 0
  3  16:55:08  DELETE FROM   customer WHERE   customer_id = 600                                      1 row(s) affected
```

三、 思考题

1) 如果 insert 一条数据到 actor 表，但 actor_id 和已有数据重复，会发生什么？同学们请自己尝试一下，截图并分析原因。



```
Output
Action Output
#  Time      Action                                                  Message
1  16:59:18  INSERT INTO actor VALUES ('200', 'GRACE', 'WHITE', NOW())  Error Code: 1062. Duplicate entry '200' for key 'actor.PRIMARY'
```

会报错，提示主键冲突。

2) insert 语句还用了一个函数 NOW()，是做什么的呢？
获取当前时间戳作为值填入那一列。