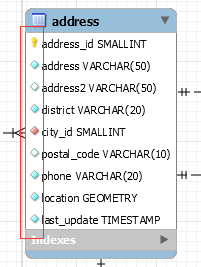
实验一报告

# 回答问题

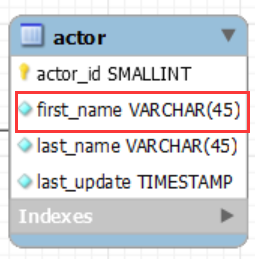
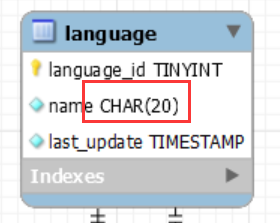
请一边熟悉sakila数据库，一边回答以下问题：

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

# 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

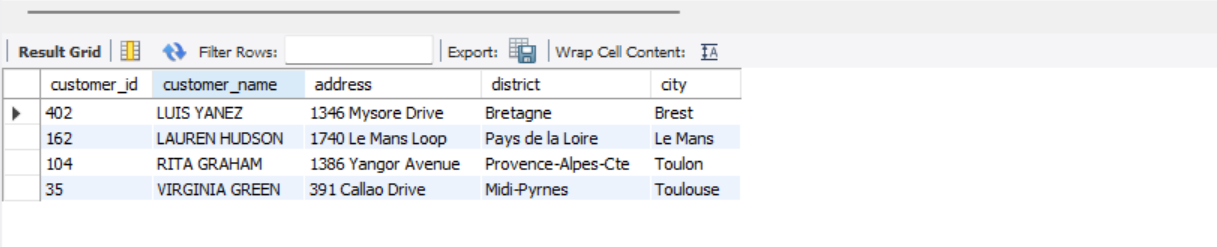
    JOIN address USING (address\_id)

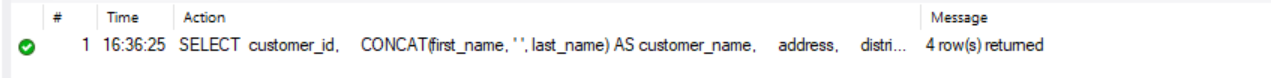
    JOIN city USING (city\_id)

    JOIN country USING (country\_id)

WHERE

    country = "France";





1. 列出属于“Children”类型并以“J”开头的电影名；

SELECT

    title

FROM

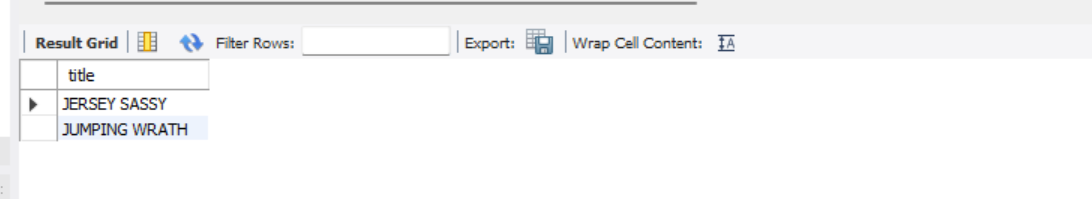
    film

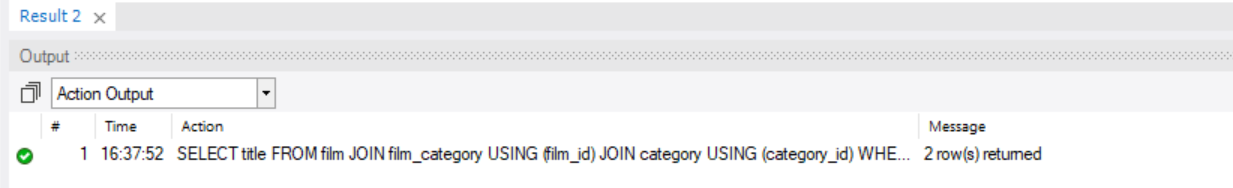
    JOIN film\_category USING (film\_id)

    JOIN category USING (category\_id)

WHERE

    name = 'Children' AND LEFT(title, 1) = 'J';





1. 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

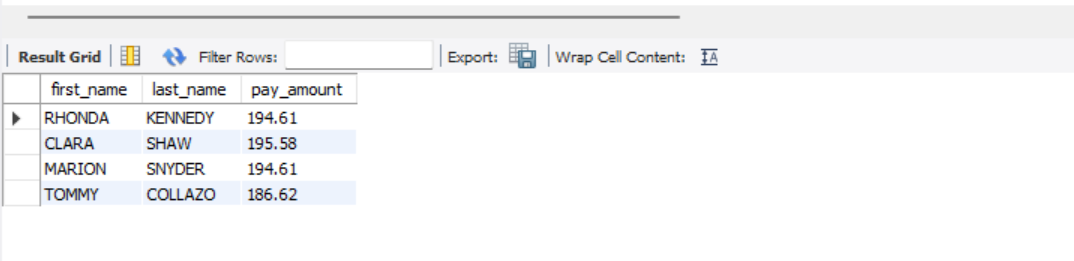
    JOIN payment USING (customer\_id)

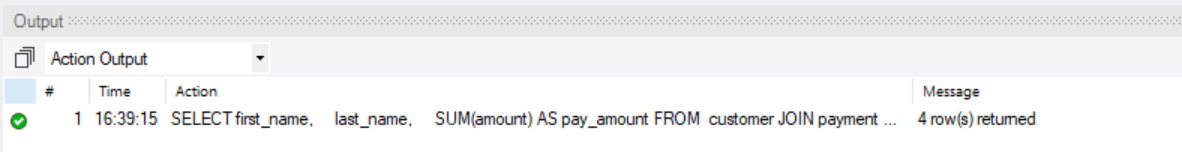
GROUP BY

    customer\_id

HAVING

    pay\_amount > 180 AND pay\_amount < 200;





1. 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)

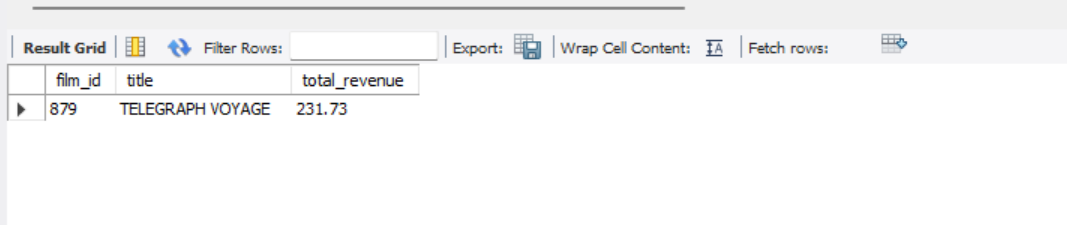
GROUP BY

    film\_id

ORDER BY

    total\_revenue DESC

LIMIT 1;





1. 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

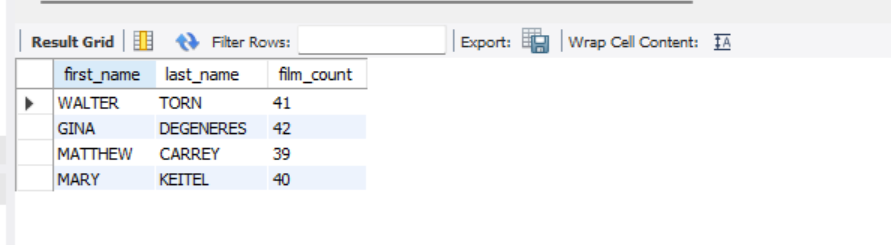
    JOIN film\_actor USING (actor\_id)

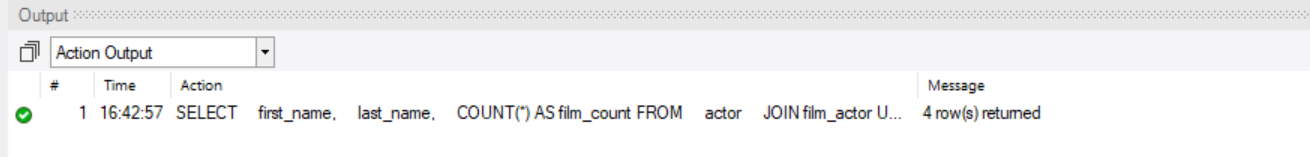
GROUP BY

    actor\_id

HAVING

COUNT(\*) > 38;





1. 请找出没有租借过电影《NATURAL STOCK》的顾客姓名；

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

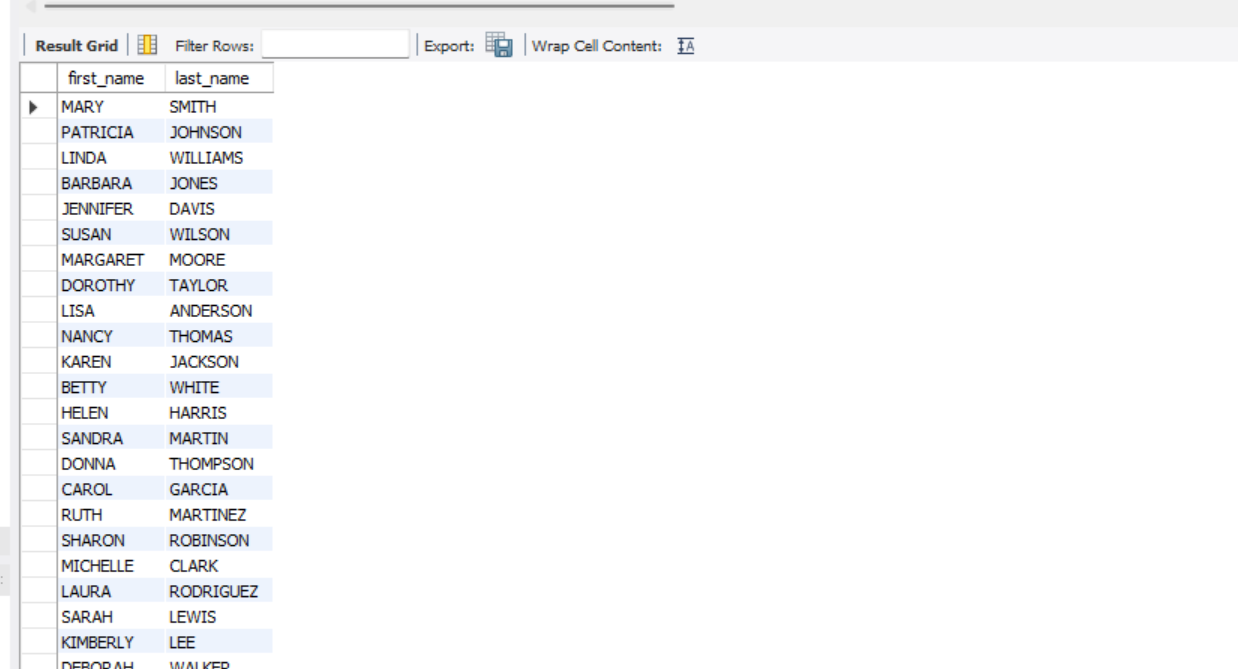
WHERE

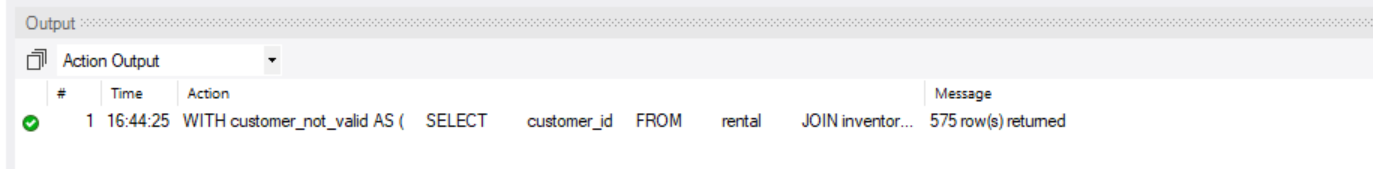
    customer\_id NOT IN (

        SELECT customer\_id

        FROM customer\_not\_valid

);





1. 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

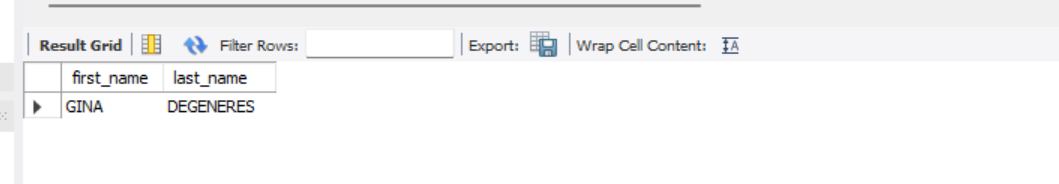
    title IN ('ELEPHANT TROJAN', 'DOGMA FAMILY')

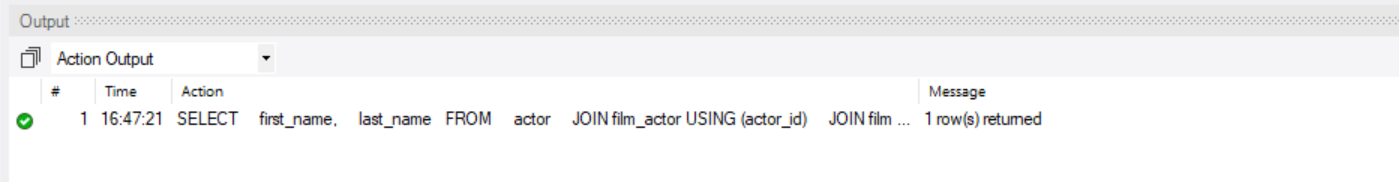
GROUP BY

    actor\_id

HAVING

    COUNT(\*) = 2;





1. 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

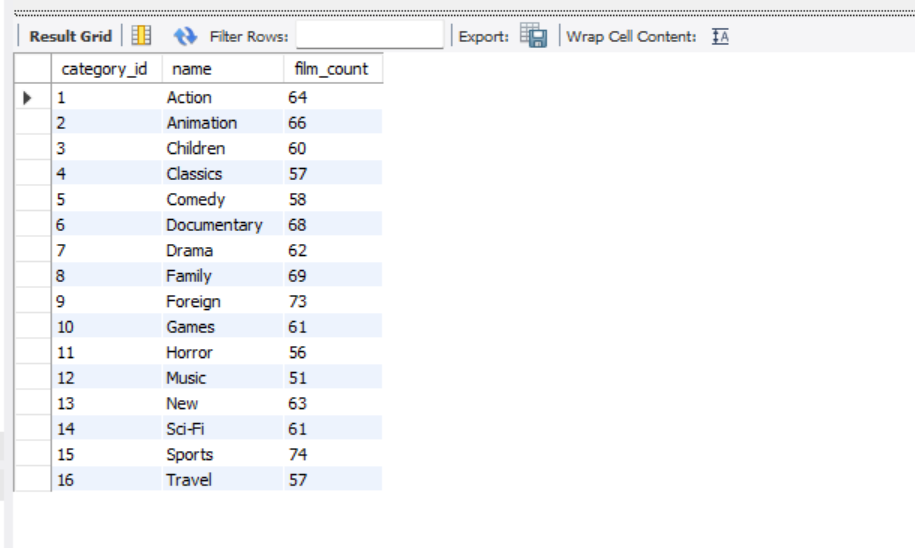
    category

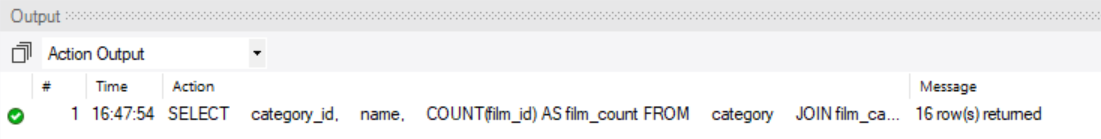
    JOIN film\_category USING (category\_id)

    JOIN film USING (film\_id)

GROUP BY

    category\_id;





1. 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)

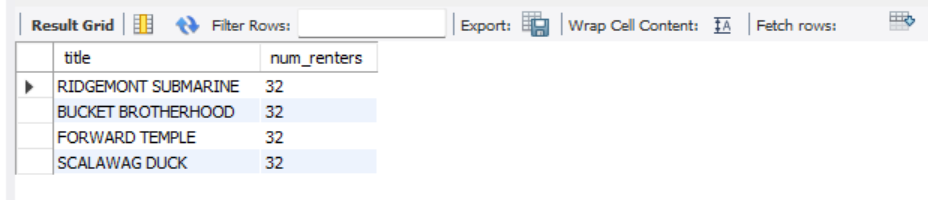
GROUP BY

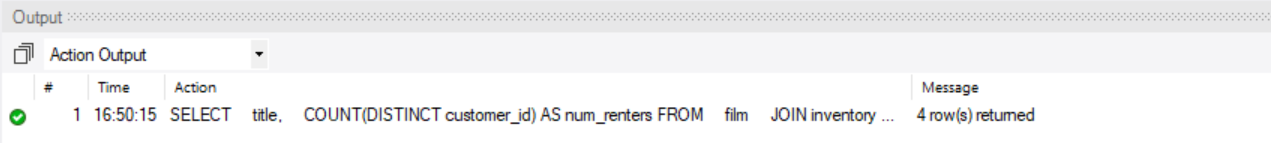
    film\_id

ORDER BY

    COUNT(DISTINCT customer\_id) DESC

LIMIT 4;





1. 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)

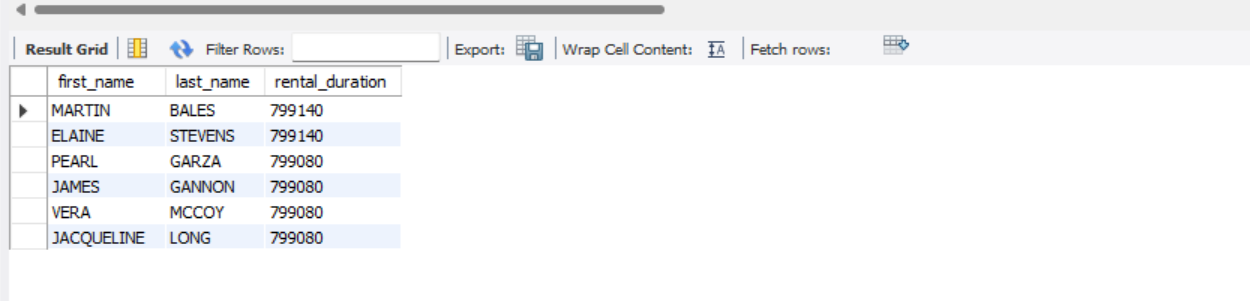
GROUP BY

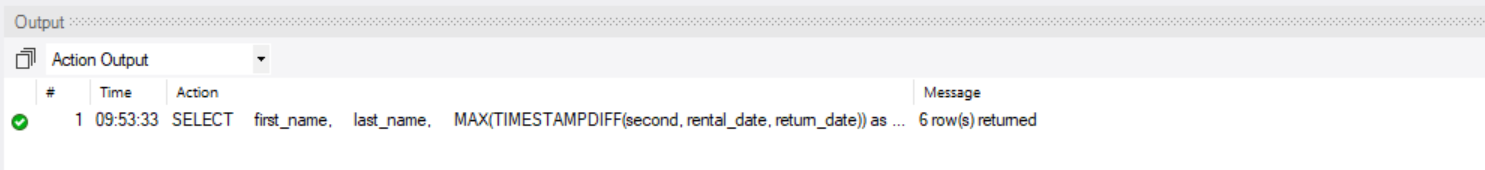
    customer\_id

ORDER BY

    rental\_duration DESC

LIMIT 6;





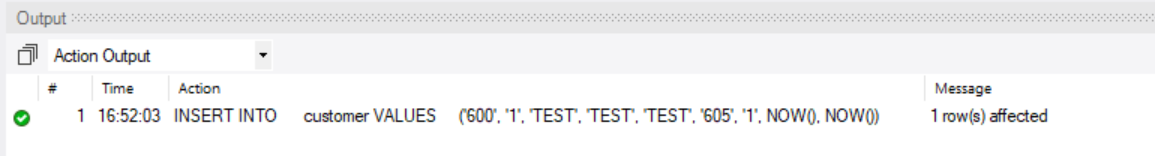
1. 在customer表中新增一条数据，注意customer表与其他表的关系；

INSERT INTO

customer

VALUES

    ('600', '1', 'TEST', 'TEST', 'TEST', '605', '1', NOW(), NOW())



1. 修改刚才在customer表中新增的那条数据；

UPDATE

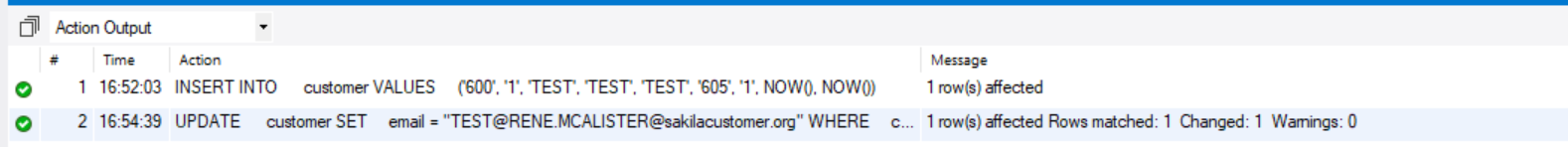
    customer

SET

    email = "TEST@RENE.MCALISTER@sakilacustomer.org"

WHERE

    customer\_id = 600



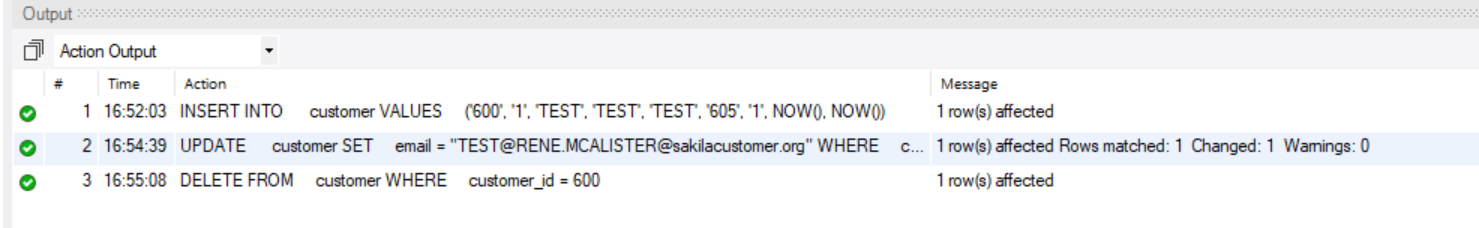
1. 删除第11步新增的那条数据。

DELETE FROM

    customer

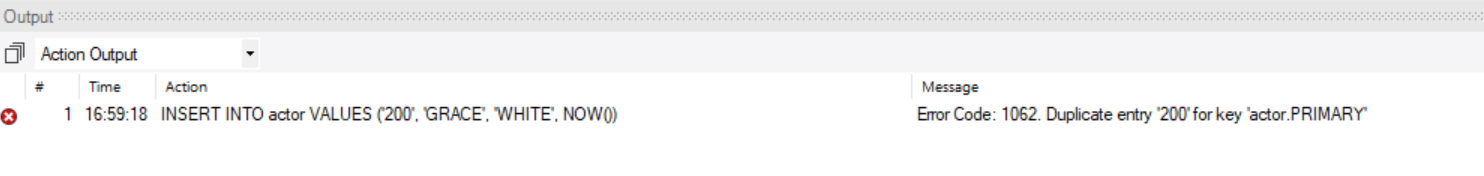
WHERE

    customer\_id = 600

****

# 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。



会报错，提示主键冲突。

1. 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。