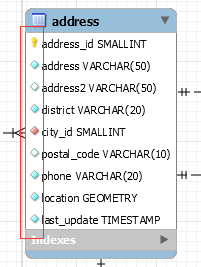
实验一报告

# 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 |
|  | 不可为空的属性 |
|  | 一般属性 |
|  | 唯一属性 |

2. 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？

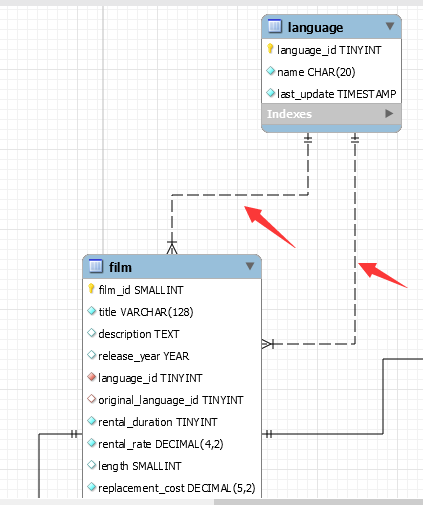
film-film\_actor-actor

3. 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

4. film和language表间的2条虚线表示什么意思？

多对1的非识别关系，或者说依赖， 表示电影可能是译制片，有源语言和现在版本的语言



# 实验截图

*（注意截图清晰，截图时需要体现SQL语句、执行结果、Output窗口）*

*脚本代码：*

*===================================================================*

*# 1*

*# select store\_id, address, district, city, country from (((store join address on store.address\_id=address.address\_id) join city on address.city\_id=city.city\_id) join country on city.country\_id=country.country\_id);*

*# 2*

*# select first\_name, last\_name from ((actor JOIN film\_actor ON actor.actor\_id=film\_actor.actor\_id) join film on film\_actor.film\_id=film.film\_id) where film.title='ROCKY WAR';*

*# Select customer\_id, sum(amount) as amount\_sum From payment Group by customer\_id order by amount\_sum desc;*

*# 3*

*# SELECT first\_name, last\_name, sum(amount)as sum\_amount from (customer join payment on customer.customer\_id=payment.customer\_id) group by customer.customer\_id order by sum\_amount desc limit 5;*

*# 4*

*# select filmid, filmname, sum(rentage) as sum\_rentage*

*# from (*

*# select film.film\_id as filmid, inventory.inventory\_id as inventoryid, film.title as filmname,rental.rental\_id as rentalid, payment.amount as rentage*

*# from film, inventory, rental, payment where film.film\_id=inventory.film\_id and inventory.inventory\_id=rental.inventory\_id and rental.rental\_id=payment.rental\_id)as kk*

*# group by filmid order by sum\_rentage desc limit 1;*

*# 5*

*# select actorid, firstname, lastname, filmcnt*

*# from(*

*# select actorid, firstname, lastname, count(distinct filmid) as filmcnt*

*# from (select actor.actor\_id as actorid, actor.first\_name as firstname, actor.last\_name as lastname, film\_actor.film\_id as filmid*

*# from actor, film\_actor*

*# where actor.actor\_id=film\_actor.actor\_id) as kk*

*# group by actorid) as kkk*

*# where filmcnt>'35';*

*# 6 film inventory rental customer*

*# select distinct( fname), lname*

*# from (*

*# select film.title as filmname, film.film\_id as filmid, inventory.inventory\_id as ivid, rental.customer\_id as cid, customer.first\_name as fname, customer.last\_name as lname*

*# from film, inventory, rental, customer*

*# where film.film\_id=inventory.film\_id and inventory.inventory\_id=rental.inventory\_id and rental.customer\_id=customer.customer\_id) as kk*

*# where filmname<>'TELEGRAPH VOYAGE';*

*# 7*

*-- select first\_name,last\_name*

*-- from actor*

*-- where exists(*

*-- select actor\_id from film,film\_actor*

*-- where film\_actor.film\_id=film.film\_id and film.title = 'ELEPHANT TROJAN' and actor.actor\_id=film\_actor.actor\_id*

*-- )*

*-- and exists(*

*-- select actor\_id from film,film\_actor*

*-- where film\_actor.film\_id=film.film\_id and film.title = 'SPLASH GUMP' and actor.actor\_id=film\_actor.actor\_id*

*-- );*

*# 8*

*# select cataid, cata, cnt from(*

*# select cataid, cata, count(distinct filmid)as cnt from(select category.name as cata, category.category\_id as cataid, film\_category.film\_id as filmid from category, film\_category where category.category\_id=film\_category.category\_id )as newcate group by cataid) as catecount;*

*# 9 有哪些影片是 2 个商店都有库存的？ inventory, film, store*

*# select distinct(filmname) from(*

*# select film.title as filmname, store.store\_id as storeid, film.film\_id as filmid from inventory, film, store where film.film\_id=inventory.film\_id and inventory.store\_id=store.store\_id) as film\_store where exists(select store\_id from inventory where filmid=film\_id and storeid<>store\_id);*

*# 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；*

*#select fstname, lstname, deltdate from (# 在这里对客户id进行唯一性限制*

*# select distinct(customer.customer\_id) as customerid, customer.first\_name as fstname, customer.last\_name as lstname, datediff(rental.return\_date, rental.rental\_date) as deltdate*

*# from customer, rental*

*# where customer.customer\_id=rental.customer\_id) as newrental*

*# group by customerid*

*# order by deltdate*

*# desc limit 6;*

*# 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(customer\_id, store\_id, first\_name, last\_name, address\_id, active, create\_date, last\_update) select(select IFNULL(max(customer\_id),0) from rental)+1, (select IFNULL(store\_id,0) from store limit 1) ,'HAOER','TIANSUO',(select IFNULL(address\_id,0) from address limit 1), 0, '2006-02-15 21:30:53',now();*

*#select \* from customer where first\_name='HAOER';*

*# 12 update*

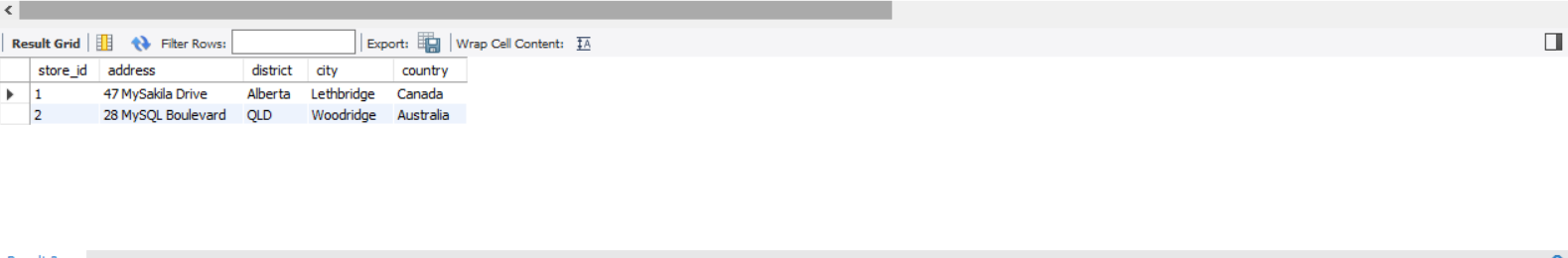
*# update customer set last\_update = now() where customer\_id=(select IFNULL(max(customer\_id),0) from rental)+1;*

*# 13 delete*

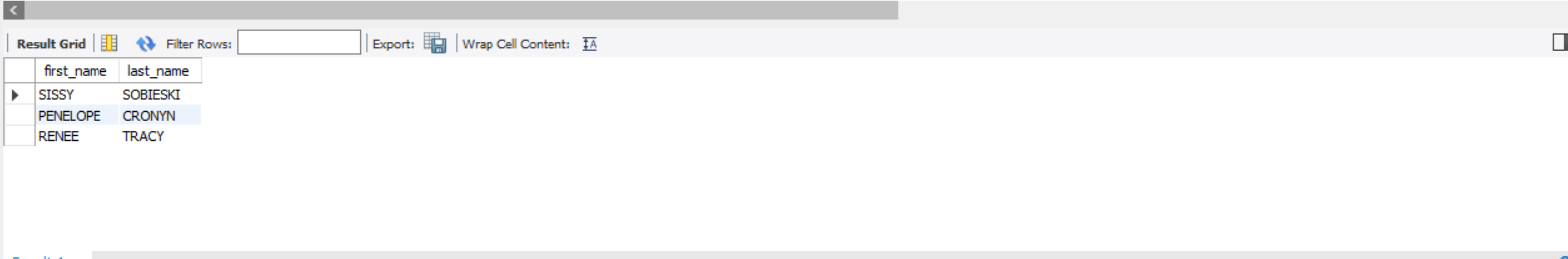
*# delete from customer where customer\_id=(select IFNULL(max(customer\_id),0) from rental)+1;*

*====================================================================*

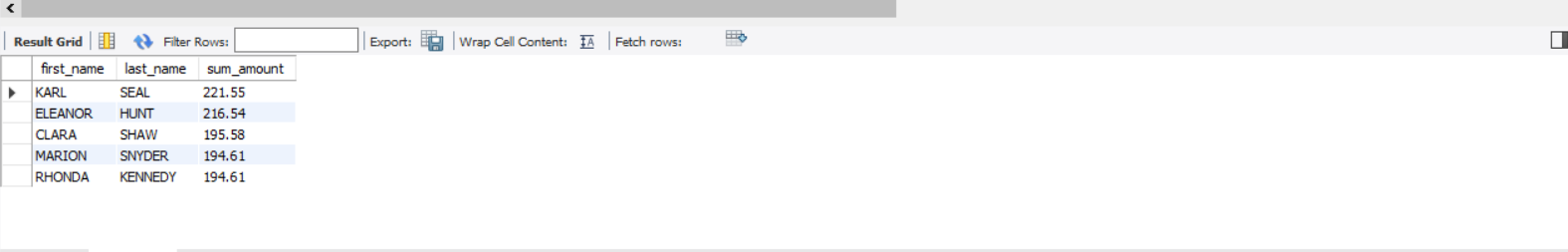
1. 请列出所有商店的详细地址，显示商店id，商店地址，所在区域，所在城市，所在国家；

**

1. 哪些演员出演过影片《ROCKY WAR》？请列出他的first\_name, last\_name；



1. 找出租DVD花费最高的前5名，请列出他们的first\_name, last\_name和每个人花费的金额；

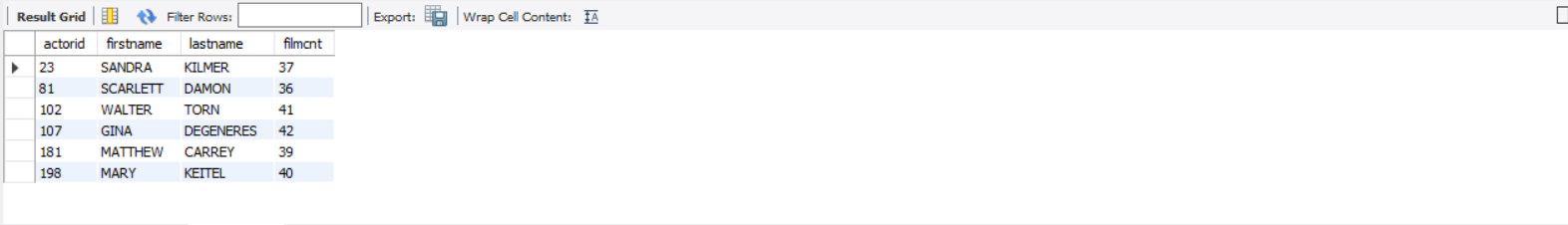


1. 哪个影片获得了总体最高的租金？请列出影片id、影片名、总租金；

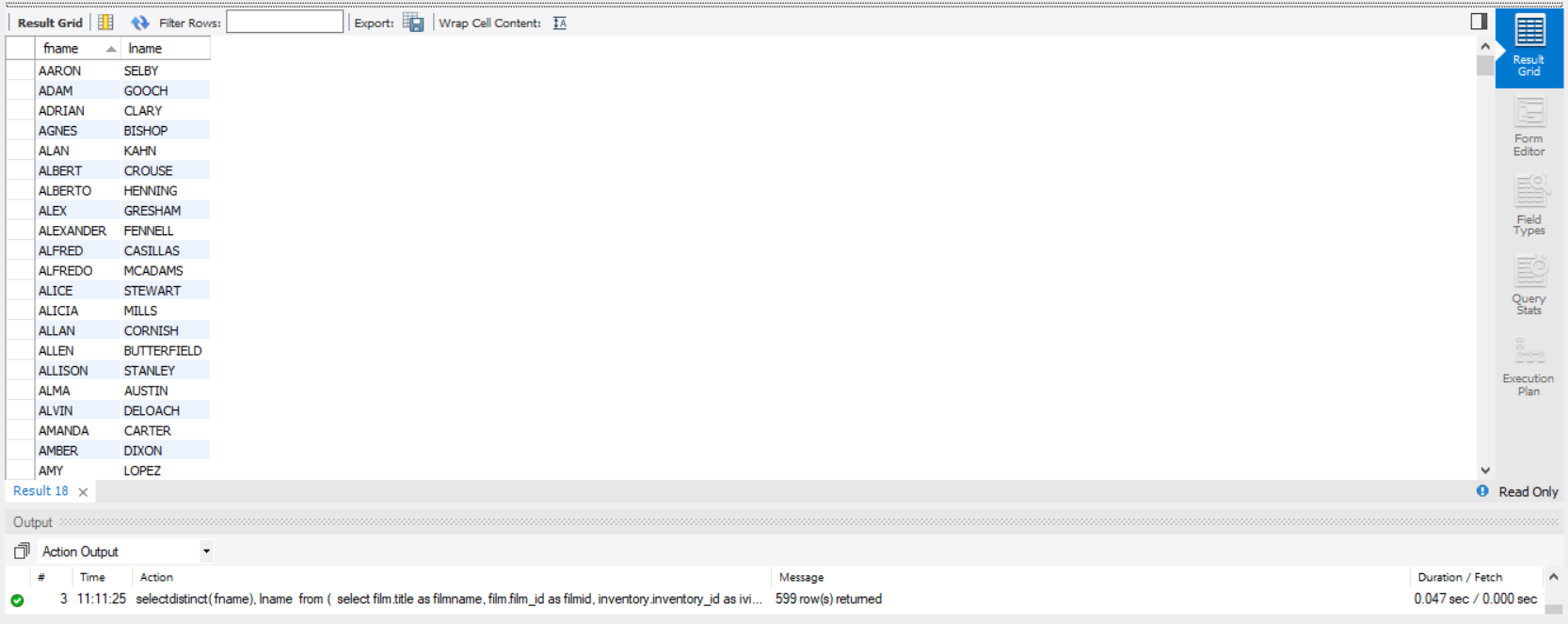
# filmid, filmname, sum\_rentage

'879', 'TELEGRAPH VOYAGE', '231.73'

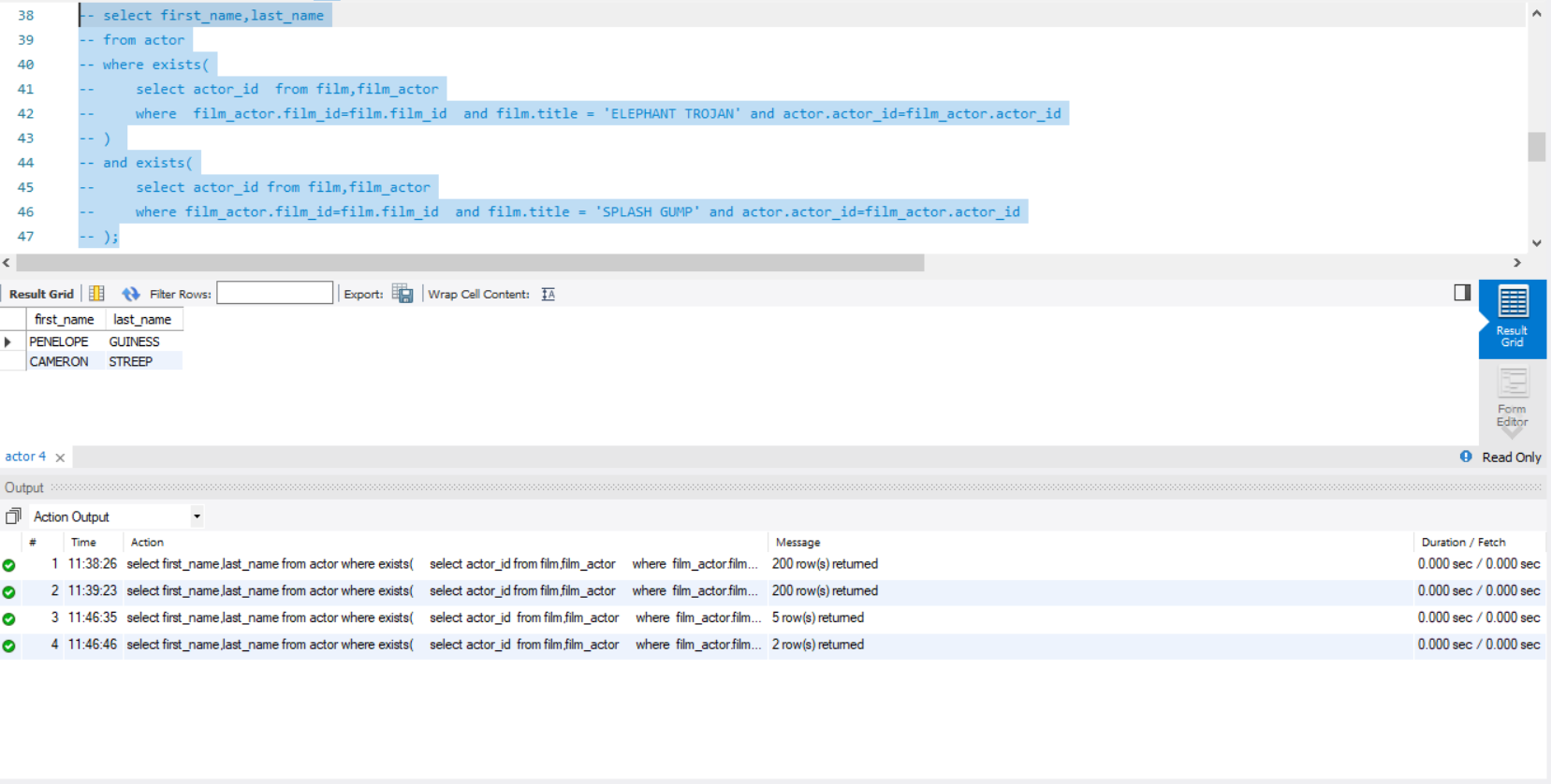
1. 哪个演员出演的电影超过35部？ 请列出演员id、演员名、出演的电影数；



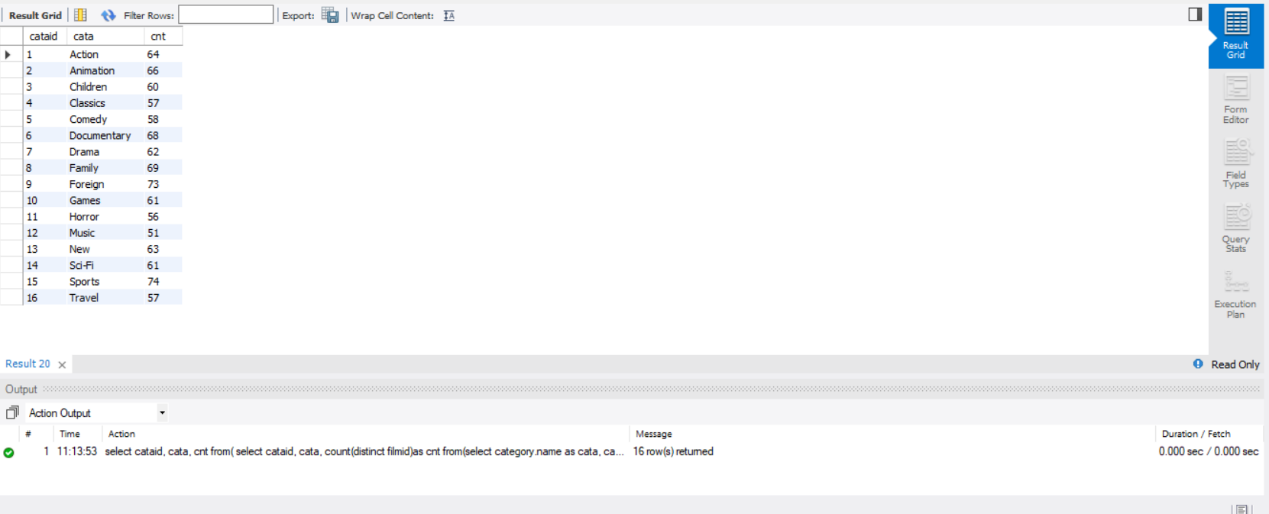
1. 请找出没有租借过电影《TELEGRAPH VOYAGE》的顾客姓名；



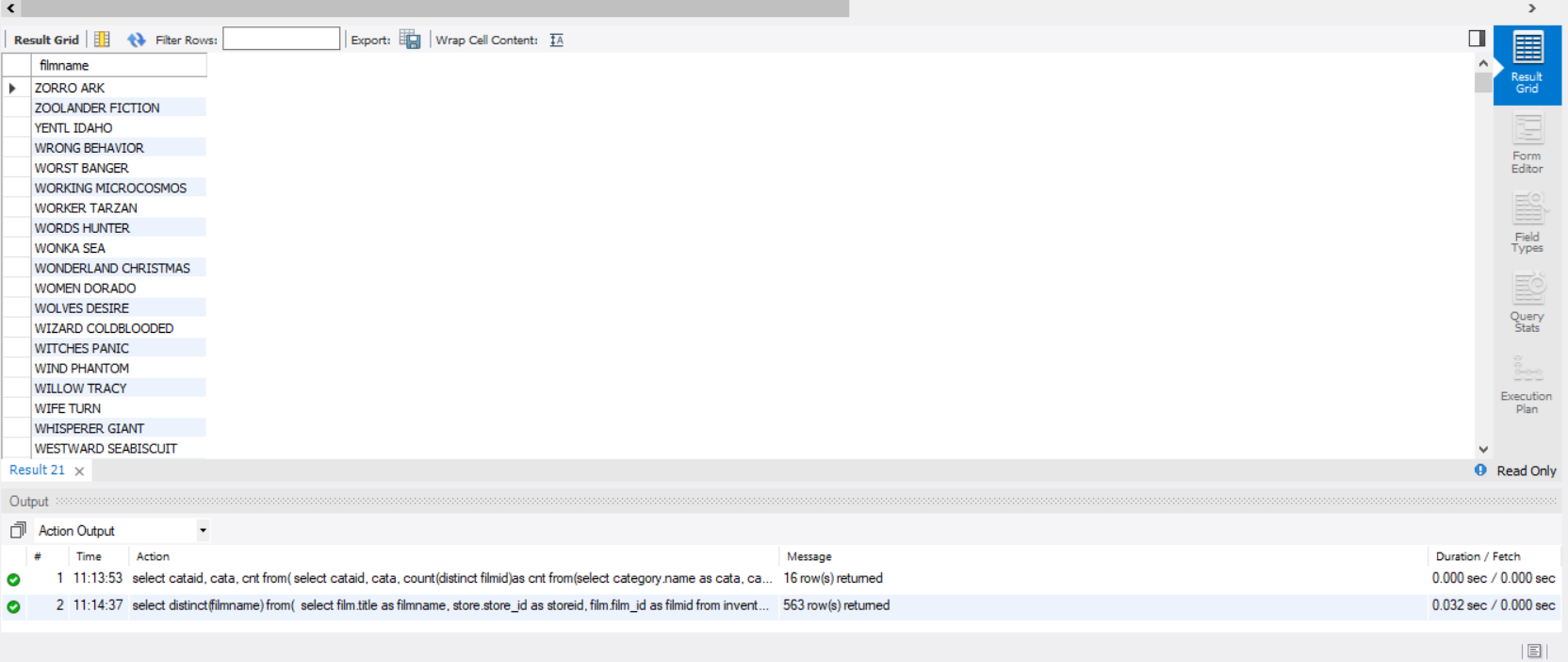
1. 查询演过《ELEPHANT TROJAN》和《SPLASH GUMP》这两部电影的演员，列出其姓名；



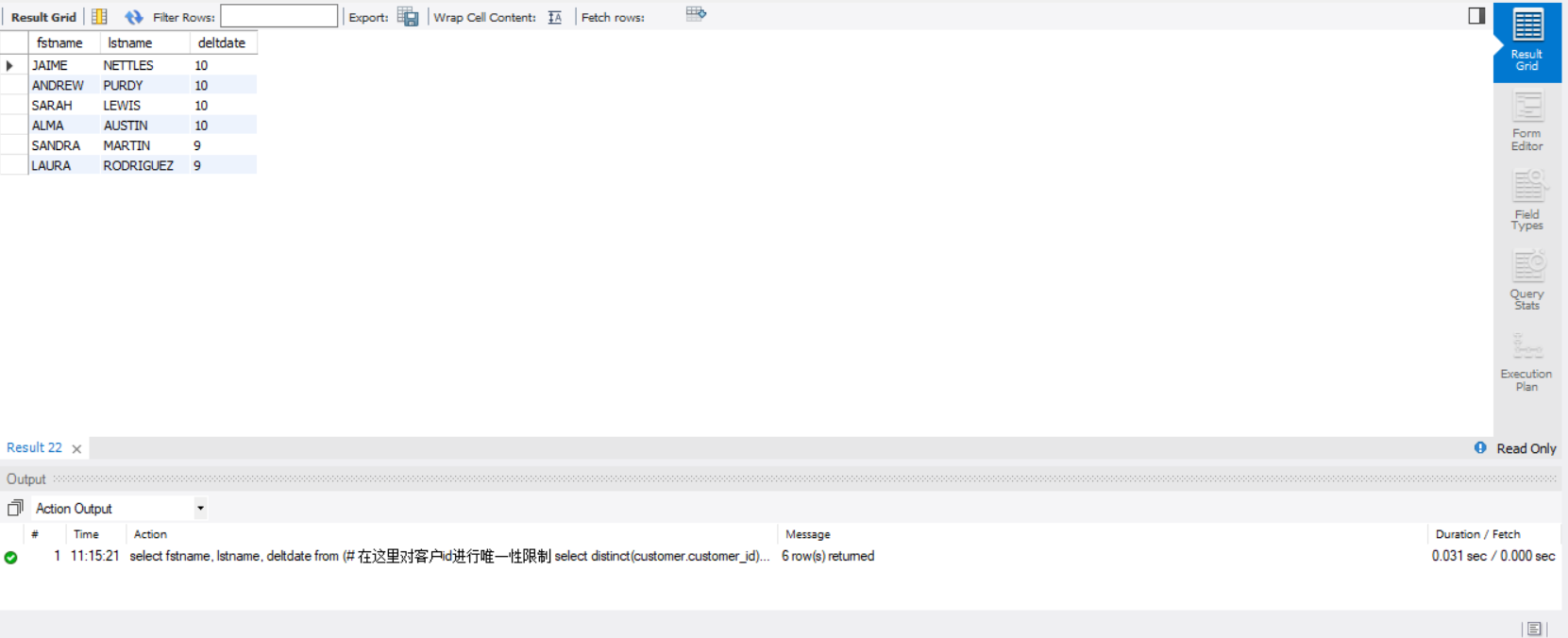
1. 统计每种类型的影片数，显示类型编号、类型名称、该类型影片数；



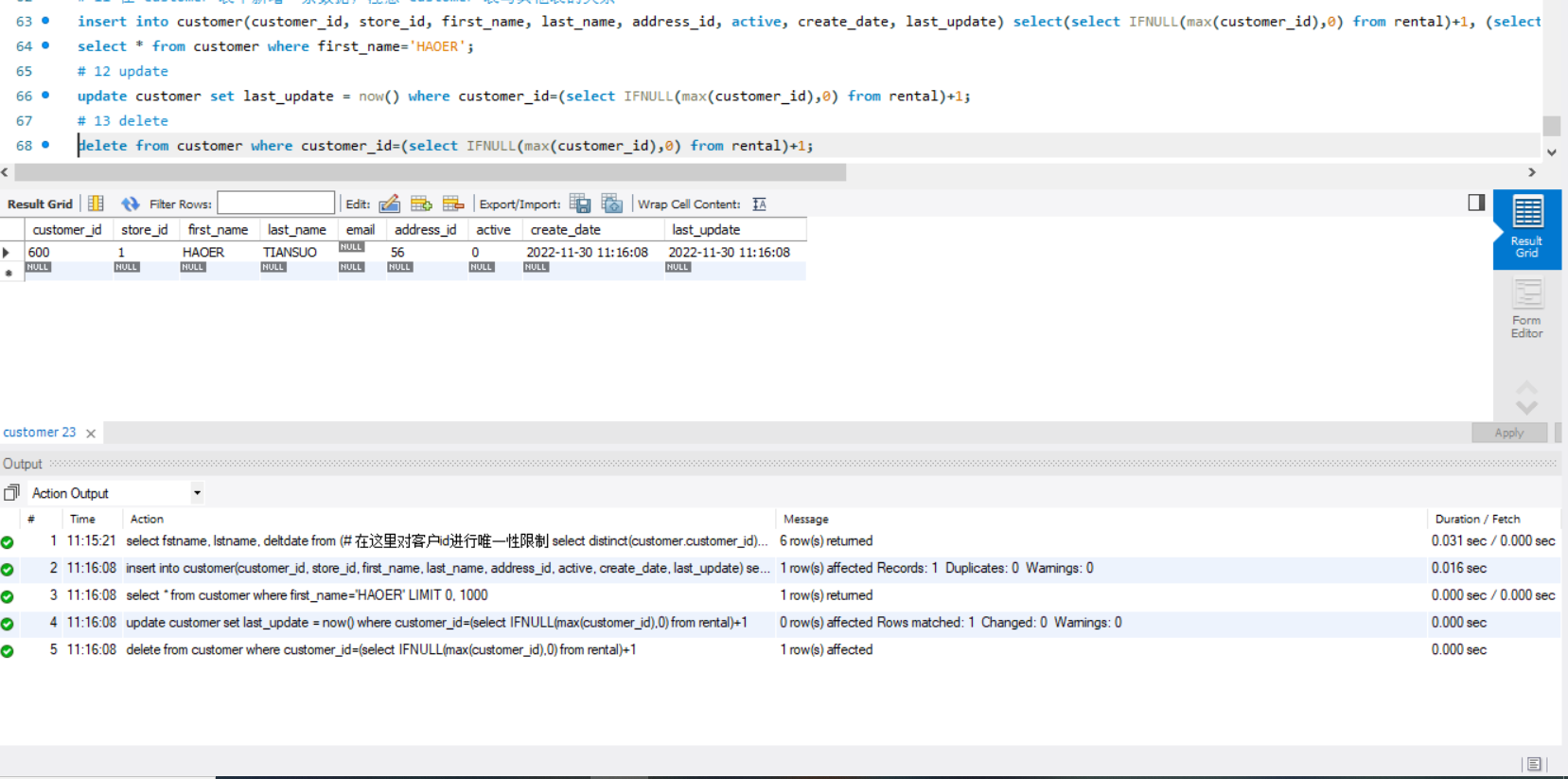
1. 有哪些影片是2个商店都有库存的？



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；



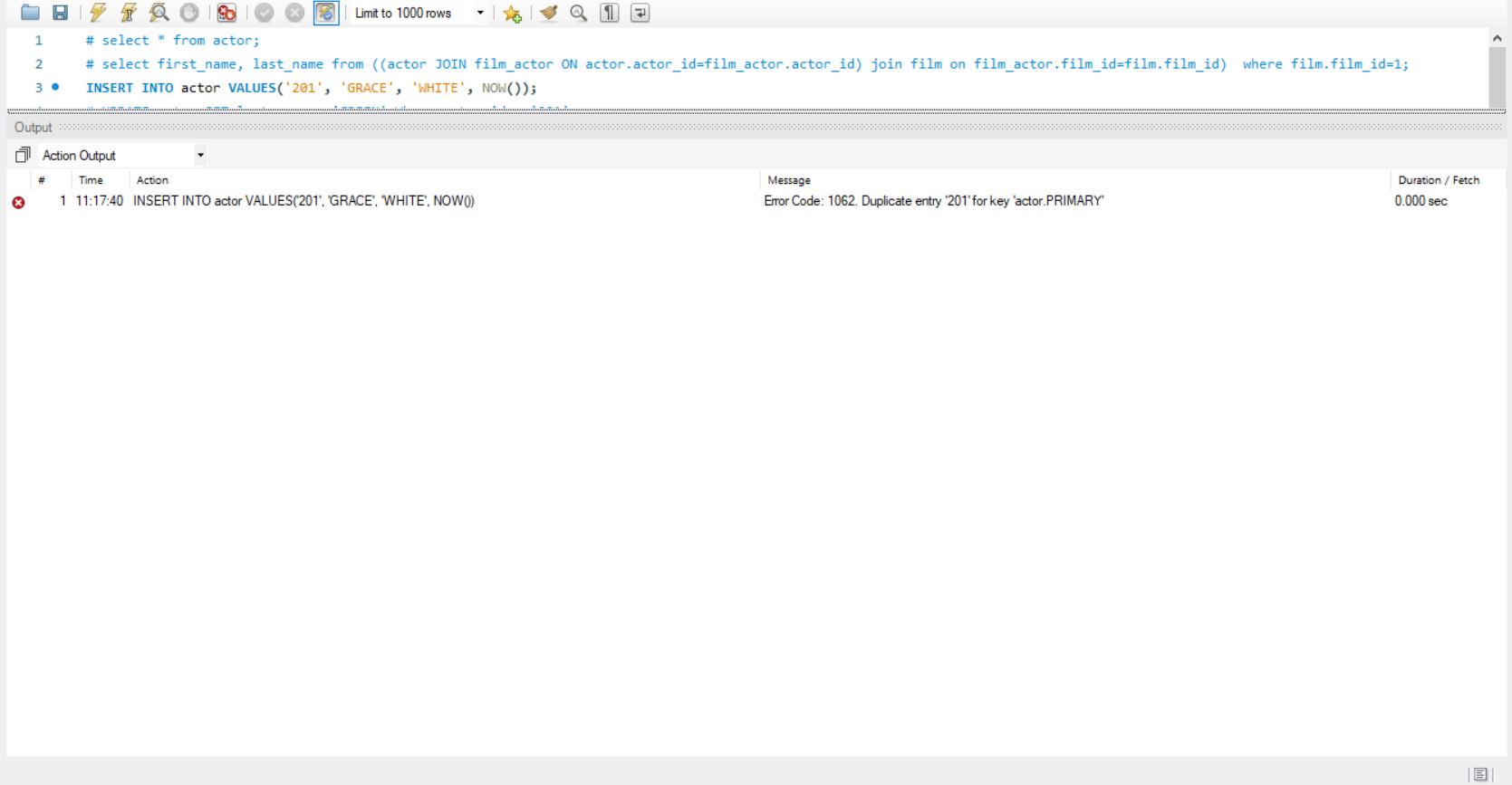
1. 12、13在customer表中新增一条数据，注意customer表与其他表的关系；



# 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。

插入失败，因为actor\_id是唯一属性



1. insert语句还用了一个函数NOW()，是做什么的呢？

获取当前计算机的时间，用作create\_date;