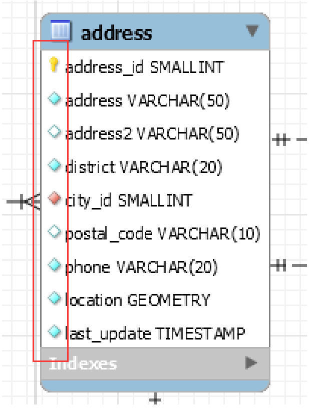
实验一报告

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）



address

address_id SMALLINT

address VARCHAR(50)

address2 VARCHAR(50)

district VARCHAR(20)

city_id SMALLINT

postal_code VARCHAR(10)

phone VARCHAR(20)

location GEOMETRY

last_update TIMESTAMP

Indexes

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？

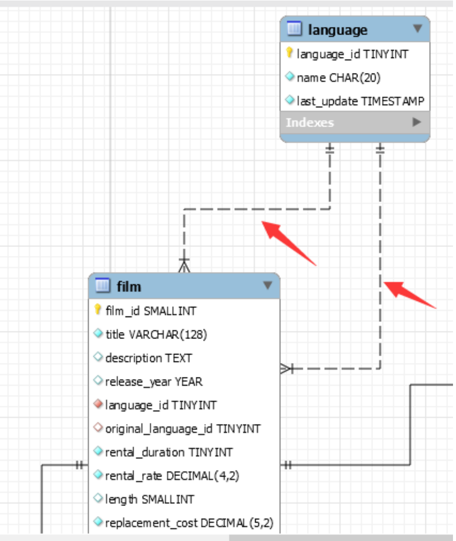
film-film_actor-actor

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



二、 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）

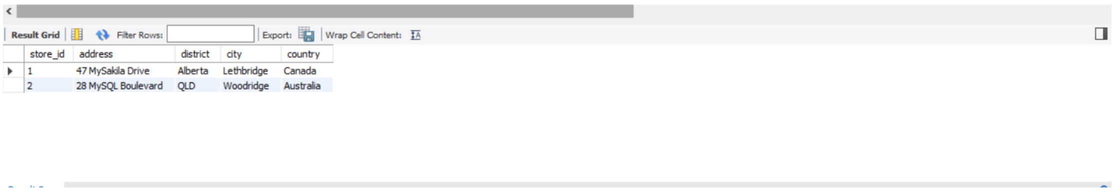
脚本代码：

```
=================================================================

# 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
```

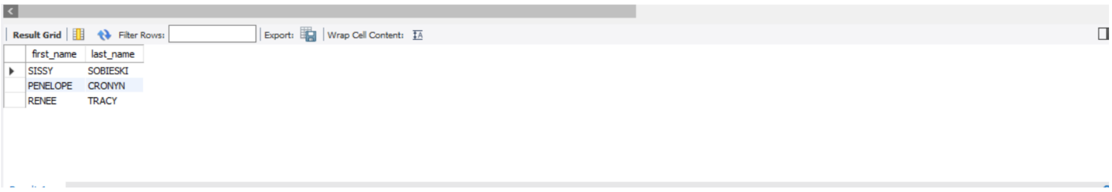
```
inventory, film, store where film.film_id=inventory.film_id and
inventory.store_id=store.store_id) as film_store where exists(select store_id from inventory
where filmid=film_id and storeid<>store_id);
# 10   查询单次租借影片时间最长的 6 位客户，列出其 first_name、last_name 和当
次租借时长；
#select fstname, lstname, deltdate 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；



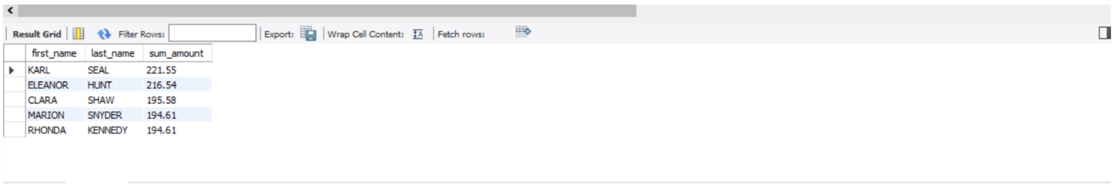
store_id	address	district	city	country
1	47 MySakila Drive	Alberta	Lethbridge	Canada
2	28 MySQL Boulevard	QLD	Woodridge	Australia

2、 哪些演员出演过影片《ROCKY WAR》？请列出他的 first_name, last_name；



first_name	last_name
SISSY	SOBIESKI
PENELOPE	CRONYN
RENEE	TRACY

3、　　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；

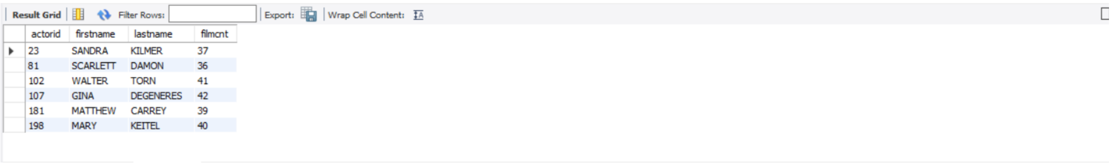


first_name	last_name	sum_amount
KARL	SEAL	221.55
ELEANOR	HUNT	216.54
CLARA	SHAW	195.58
MARION	SNYDER	194.61
RHONDA	KENNEDY	194.61

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；

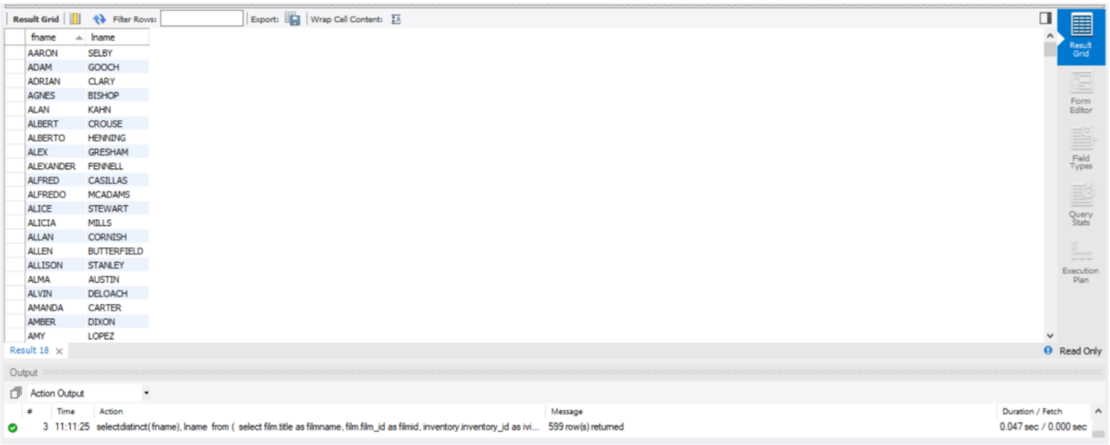
```
# filmid, filmname, sum_rentage
'879', 'TELEGRAPH VOYAGE', '231.73'
```

5、　　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；



actorid	firstname	lastname	filmcnt
23	SANDRA	KILMER	37
81	SCARLETT	DAMON	36
102	WALTER	TORN	41
107	GINA	DEGENERES	42
181	MATTHEW	CARREY	39
198	MARY	KEITEL	40

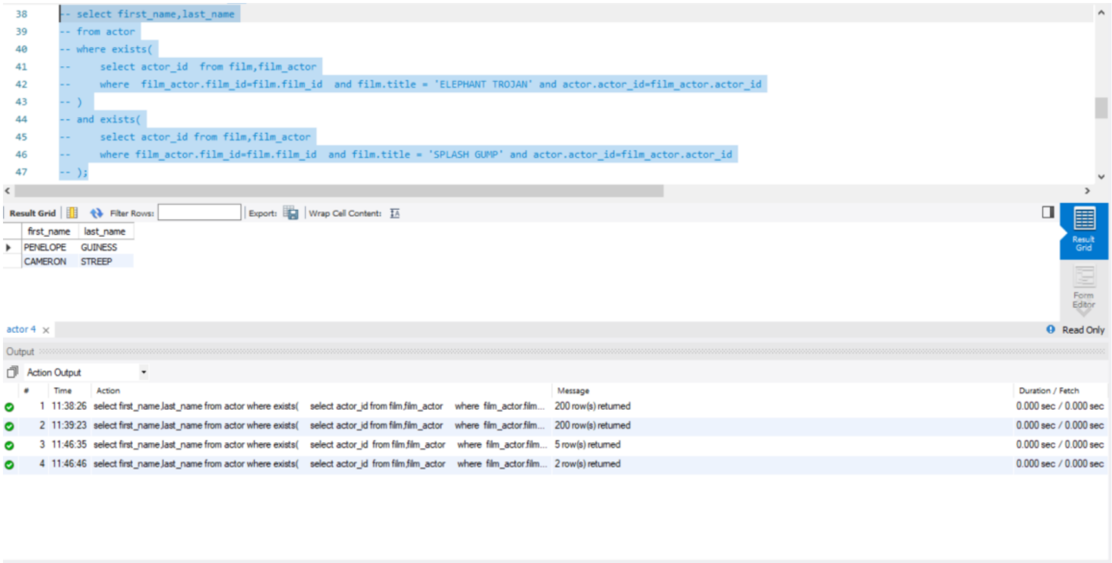
6、　　请找出没有租借过电影《TELEGRAPH VOYAGE》的顾客姓名；



fname	lname
AARON	SELBY
ADAM	GOOCH
ADRIAN	CLARY
AGNES	BISHOP
ALAN	KAHN
ALBERT	CROUSE
ALBERTO	HENNING
ALEX	GRESHAM
ALEXANDER	FENNELL
ALFRED	CASILLAS
ALFREDO	MCADAMS
ALICE	STEWART
ALICIA	MILLS
ALLAN	CORNISH
ALLEN	BUTTERFIELD
ALLISON	STANLEY
ALMA	AUSTIN
ALVIN	DELOACH
AMANDA	CARTER
AMBER	DIXON
AMY	LOPEZ

```
#   Time      Action                                                                                              Message               Duration / Fetch
3   11:11:25  selectdistinct(fname), lname  from (  select film.title as filmname, film.film_id as filmid, inventory.inventory_id as ivi...  599 row(s) returned  0.047 sec / 0.000 sec
```

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

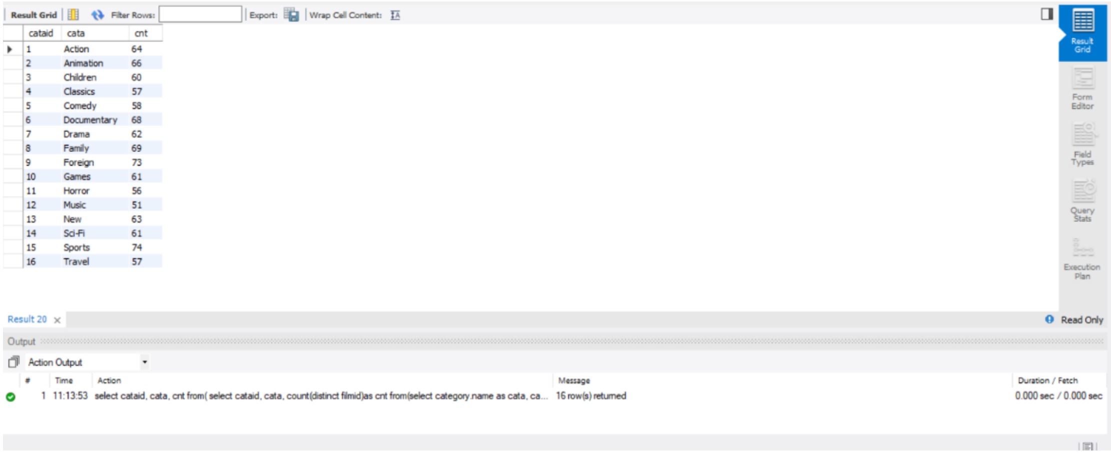


```
-- select first_name,last_name
-- from actor
-- where exists(
--     select actor_id  from film,film_actor
--     where  film_actor.film_id=film.film_id  and film.title = 'ELEPHANT TROJAN' and actor.actor_id=film_actor.actor_id
-- )
-- and exists(
--     select actor_id from film,film_actor
--     where film_actor.film_id=film.film_id  and film.title = 'SPLASH GUMP' and actor.actor_id=film_actor.actor_id
-- );
```

first_name	last_name
PENELOPE	GUINESS
CAMERON	STREEP

```
#  Time      Action                                                                                                   Message              Duration / Fetch
1  11:38:26  select first_name,last_name from actor where exists(   select actor_id from film,film_actor   where  film_actor.film...  200 row(s) returned  0.000 sec / 0.000 sec
2  11:39:23  select first_name,last_name from actor where exists(   select actor_id from film,film_actor   where  film_actor.film...  200 row(s) returned  0.000 sec / 0.000 sec
3  11:46:35  select first_name,last_name from actor where exists(   select actor_id  from film,film_actor   where  film_actor.film...  5 row(s) returned    0.000 sec / 0.000 sec
4  11:46:46  select first_name,last_name from actor where exists(   select actor_id  from film,film_actor   where  film_actor.film...  2 row(s) returned    0.000 sec / 0.000 sec
```

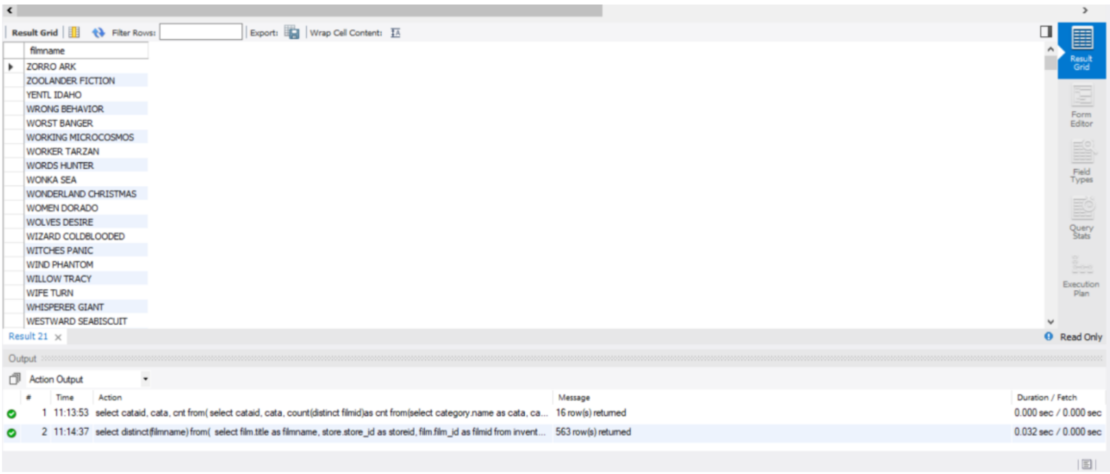
8、　　统计每种类型的影片数，显示类型编号、类型名称、该类型影片数；



cataid	cata	cnt
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

```
#  Time      Action                                                                                                   Message             Duration / Fetch
1  11:13:53  select cataid, cata, cnt from( select cataid, cata, count(distinct filmid)as cnt from(select category.name as cata, ca...  16 row(s) returned  0.000 sec / 0.000 sec
```

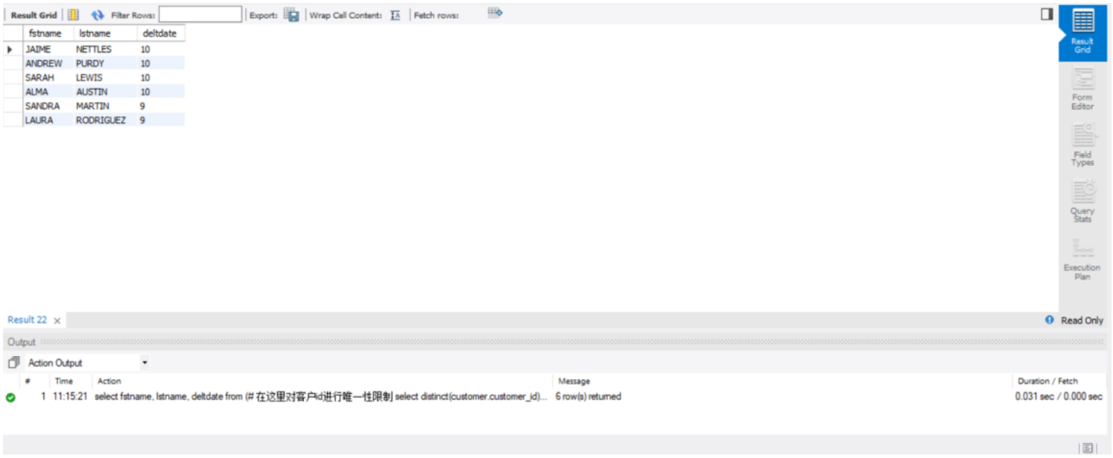
9、　　有哪些影片是 2 个商店都有库存的？



filmname
ZORRO ARK
ZOOLANDER FICTION
YENTL IDAHO
WRONG BEHAVIOR
WORST BANGER
WORKING MICROCOSMOS
WORKER TARZAN
WORDS HUNTER
WONKA SEA
WONDERLAND CHRISTMAS
WOMEN DORADO
WOLVES DESIRE
WIZARD COLDBLOODED
WITCHES PANIC
WIND PHANTOM
WILLOW TRACY
WIFE TURN
WHISPERER GIANT
WESTWARD SEABISCUIT

```
#  Time      Action                                                                                                      Message              Duration / Fetch
1  11:13:53  select cataid, cata, cnt from( select cataid, cata, count(distinct filmid)as cnt from(select category.name as cata, ca...  16 row(s) returned   0.000 sec / 0.000 sec
2  11:14:37  select distinct(filmname) from(  select film.title as filmname, store.store_id as storeid, film.film_id as filmid from invent...  563 row(s) returned  0.032 sec / 0.000 sec
```

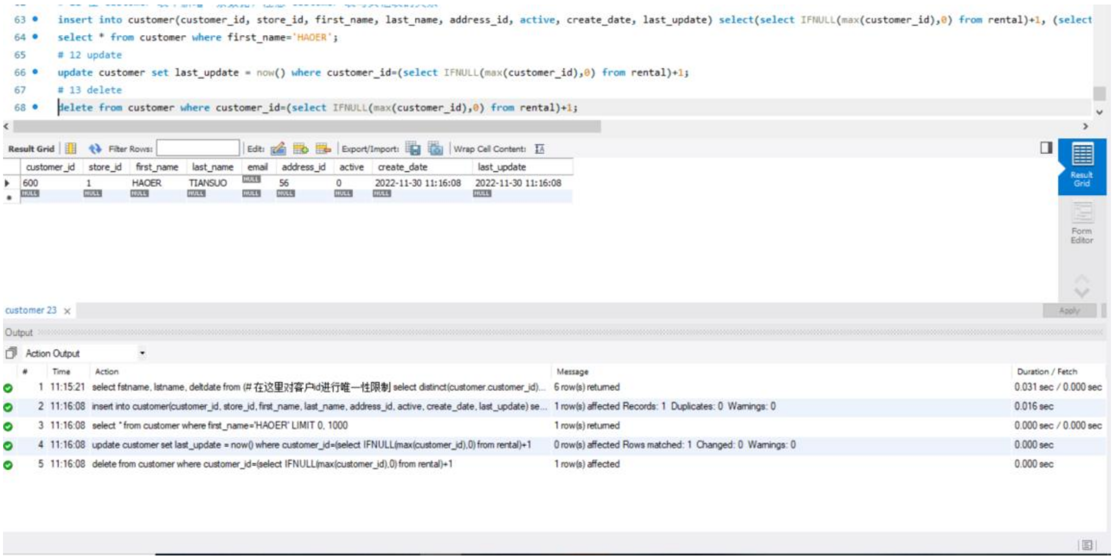
10、　查询单次租借影片时间最长的 6 位客户，列出其 first_name、last_name 和当次租借时长；



fstname	lstname	deltdate
JAIME	NETTLES	10
ANDREW	PURDY	10
SARAH	LEWIS	10
ALMA	AUSTIN	10
SANDRA	MARTIN	9
LAURA	RODRIGUEZ	9

```
#   Time      Action                                                                                        Message              Duration / Fetch
1   11:15:21  select fstname, lstname, deltdate from (# 在这里对客户id进行唯一性限制 select distinct(customer.customer_id)...  6 row(s) returned    0.031 sec / 0.000 sec
```

11、　12、13 在 customer 表中新增一条数据，注意 customer 表与其他表的关系；



```sql
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
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;
```

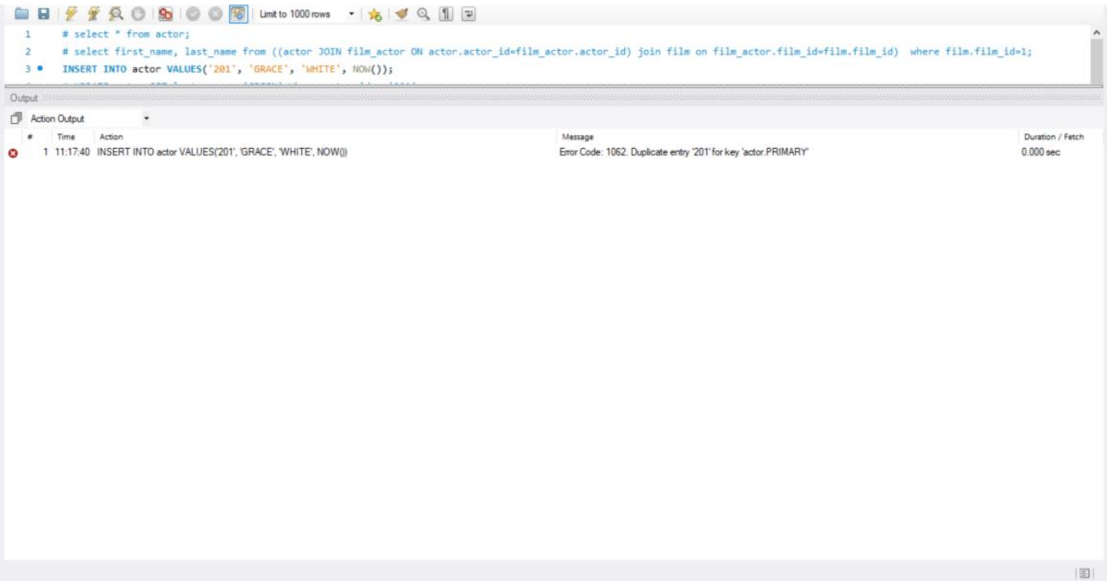
customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
600	1	HAOER	TIANSUO	NULL	56	0	2022-11-30 11:16:08	2022-11-30 11:16:08
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
#   Time      Action                                                                                                           Message                                                Duration / Fetch
1   11:15:21  select fstname, lstname, deltdate from (# 在这里对客户id进行唯一性限制 select distinct(customer.customer_id)...     6 row(s) returned                                      0.031 sec / 0.000 sec
2   11:16:08  insert into customer(customer_id, store_id, first_name, last_name, address_id, active, create_date, last_update) se...  1 row(s) affected Records: 1  Duplicates: 0  Warnings: 0  0.016 sec
3   11:16:08  select * from customer where first_name='HAOER' LIMIT 0, 1000                                                   1 row(s) returned                                      0.000 sec / 0.000 sec
4   11:16:08  update customer set last_update = now() where customer_id=(select IFNULL(max(customer_id),0) from rental)+1     0 row(s) affected Rows matched: 1  Changed: 0  Warnings: 0  0.000 sec
5   11:16:08  delete from customer where customer_id=(select IFNULL(max(customer_id),0) from rental)+1                        1 row(s) affected                                      0.000 sec
```

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。

插入失败，因为 actor_id 是唯一属性



```
# select * from actor;
# 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.film_id=1;
INSERT INTO actor VALUES('201', 'GRACE', 'WHITE', NOW());
```

```
#  Time      Action                                             Message                                                          Duration / Fetch
1  11:17:40  INSERT INTO actor VALUES('201', 'GRACE', 'WHITE', NOW())   Error Code: 1062. Duplicate entry '201' for key 'actor.PRIMARY'   0.000 sec
```

2) 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;