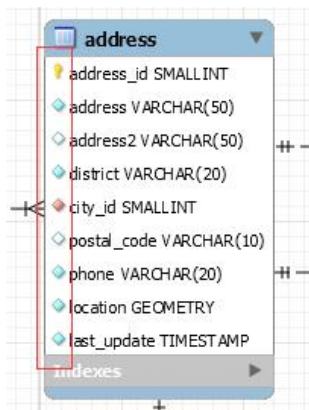


实验一报告

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
💎	表示不能取 null(空值)
◇	没有约束条件
🔴	无符号且不能为空，外键

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？

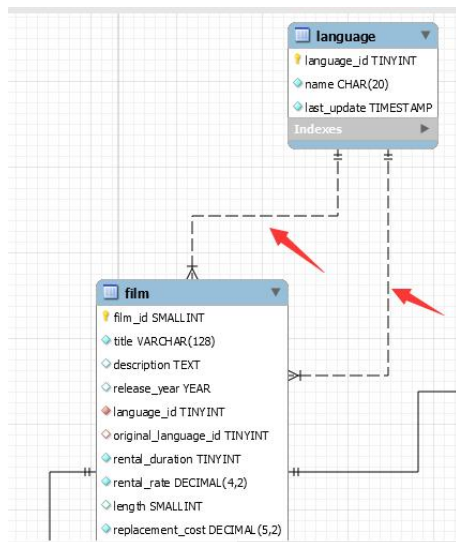
答：film,actor,film_actor;即要找出演某个影片的演员名字，需要访问这 3 张表。

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？

答：需要访问 customer,rental,inventory,film 这 4 张表。

- 4.film 和 language 表间的 2 条虚线表示什么意思？

答：表示 Non-identifying relationship;language 表的主键 language_id 是 film 表的 original_language_id 和 language_id 属性的一部分，且是 1 对多的关系，同时存在没有语言信息的 film。



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)

- 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；

```
select store_id,address,district,city,country
from
    store,address,city,country
where
    store.address_id = address.address_id and
    address.city_id = city.city_id and
    city.country_id = country.country_id;
```

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

Result 5 x Read Only Context H

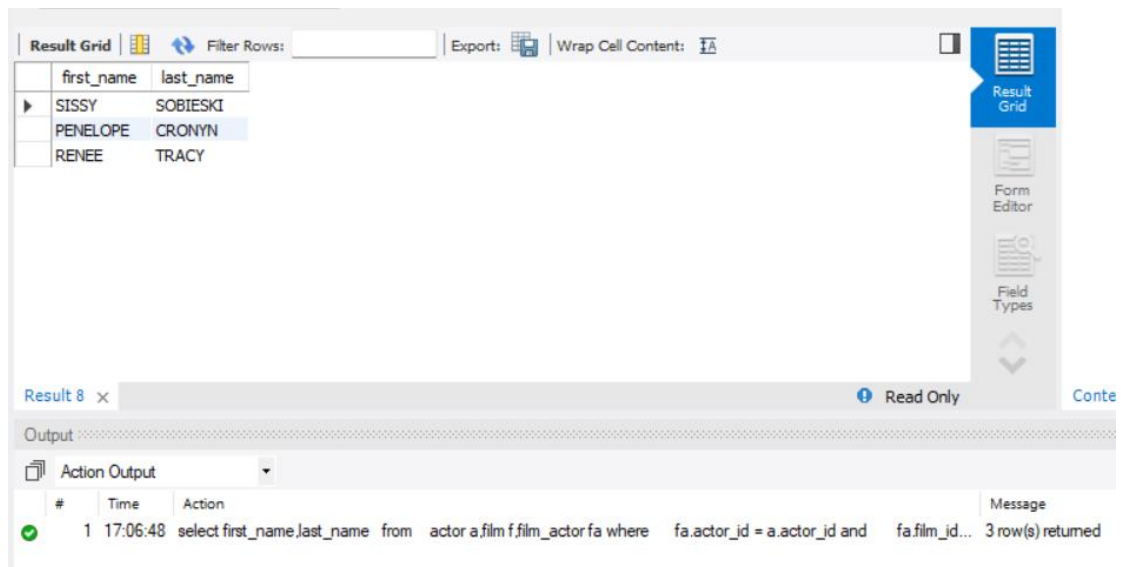
Output:

Action Output

#	Time	Action	Message
1	16:59:35	select store_id,address,district,city,country from store,address,city,country where store.address_id = address....	2 row(s) returned

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

```
select first_name,last_name
from
    actor a,film f,film_actor fa
where
    fa.actor_id = a.actor_id and
    fa.film_id = f.film_id and
    f.title = 'ROCKY WAR';
```



The screenshot shows a database query result grid with the following data:

	first_name	last_name
▶	SISSY	SOBIESKI
	PENELOPE	CRONYN
	RENEE	TRACY

Below the grid, the 'Output' section shows the executed SQL query and its results:

```
select first_name,last_name from actor a,film f,film_actor fa where fa.actor_id = a.actor_id and fa.film_id = f.film_id and f.title = 'ROCKY WAR';
```

Message: 3 row(s) returned

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；

```
select first_name,last_name,sum(amount)
from
    customer cus,payment pay
where
    cus.customer_id = pay.customer_id
group by first_name,last_name
order by sum(amount) desc
limit 5;
```

Result Grid			
	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

Result 10 x Read Only Context H

Output

Action Output

#	Time	Action	Message
1	17:19:57	select first_name,last_name,sum(amount) from customer cus,payment pay where cus.customer_id = pay.cust...	5 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.film_id, title, sum(amount)
FROM
film ,payment pay,rental ren,inventory inv
where
    film.film_id = inv.film_id and
    inv.inventory_id = ren.inventory_id and
    pay.rental_id = ren.rental_id
group by film_id
order by sum(amount) desc
limit 1;
```

Result Grid			
	film_id	title	sum(amount)
▶	879	TELEGRAPH VOYAGE	231.73

Result 14 x Read Only Context Help S

Output

Action Output

#	Time	Action	Message
1	17:32:19	SELECT film.film_id, title, sum(amount) FROM film ,payment pay,rental ren,inventory inv where film.film_id = inv.f...	1 row(s) returned

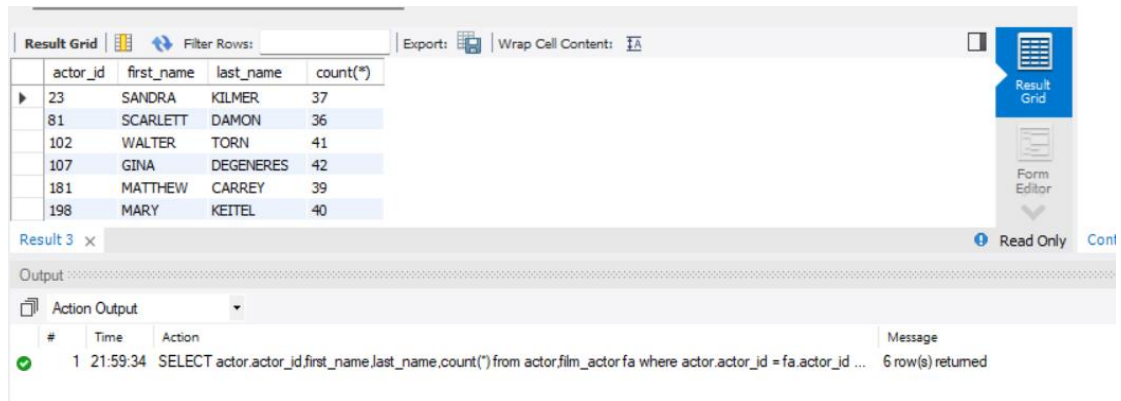
5、 哪个演员出演的电影超过 35 部？请列出演员 id、演员名、出演的电影数；

```
SELECT actor.actor_id,first_name,last_name,count(*)
from
    actor,film_actor fa
where
```

```

actor.actor_id = fa.actor_id
group by actor_id having count(*) >35;

```



actor_id	first_name	last_name	count(*)
23	SANDRA	KILMER	37
81	SCARLETT	DAMON	36
102	WALTER	TORN	41
107	GINA	DEGENERES	42
181	MATTHEW	CARREY	39
198	MARY	KEITEL	40

Result 3 x

Output

Action Output

#	Time	Action	Message
1	21:59:34	SELECT actor.actor_id,first_name,last_name,count(*) from actor,film_actor fa where actor.actor_id = fa.actor_id ...	6 row(s) returned

6、请找出没有租借过电影《TELEGRAPH VOYAGE》的顾客姓名；

```

SELECT first_name,last_name
from customer cus
where not exists
(
    select * from customer cus1,rental ren,inventory inv,film
    where cus1.customer_id = ren.customer_id and
        ren.inventory_id = inv.inventory_id and
        inv.film_id = film.film_id and
        cus1.customer_id = cus.customer_id and
        film.title = 'TELEGRAPH VOYAGE'
);

```



first_name	last_name
MARY	SMITH
LINDA	WILLIAMS
BARBARA	JONES
ELIZABETH	BROWN
JENNIFER	DAVIS

customer 3 x

Output

Action Output

#	Time	Action	Message
1	21:36:25	SELECT first_name,last_name from customer cus where not exists (select * from customer cus1,rental ren...	572 row(s) returned

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

```

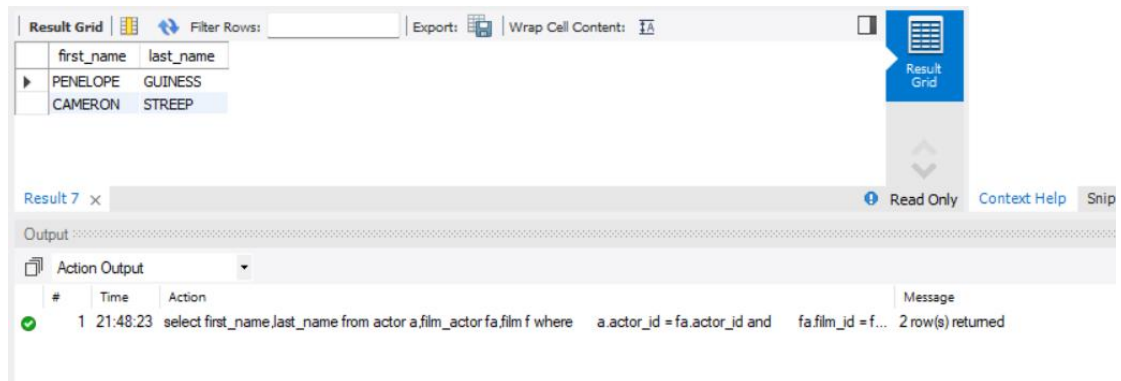
select first_name,last_name
from actor a,film_actor fa,film f
where
    a.actor_id = fa.actor_id and
    fa.film_id = f.film_id and
    f.title = 'ELEPHANT TROJAN' and

```

```

        a.actor_id in
    (
        select a1.actor_id
        from actor a1,film_actor fa1,film f1
        where
            a1.actor_id = fa1.actor_id and
            fa1.film_id = f1.film_id and
            f1.title = 'SPLASH GUMP'
    );

```



first_name	last_name
PENELOPE	GUINNESS
CAMERON	STREEP

Result 7 x

Output

Action Output

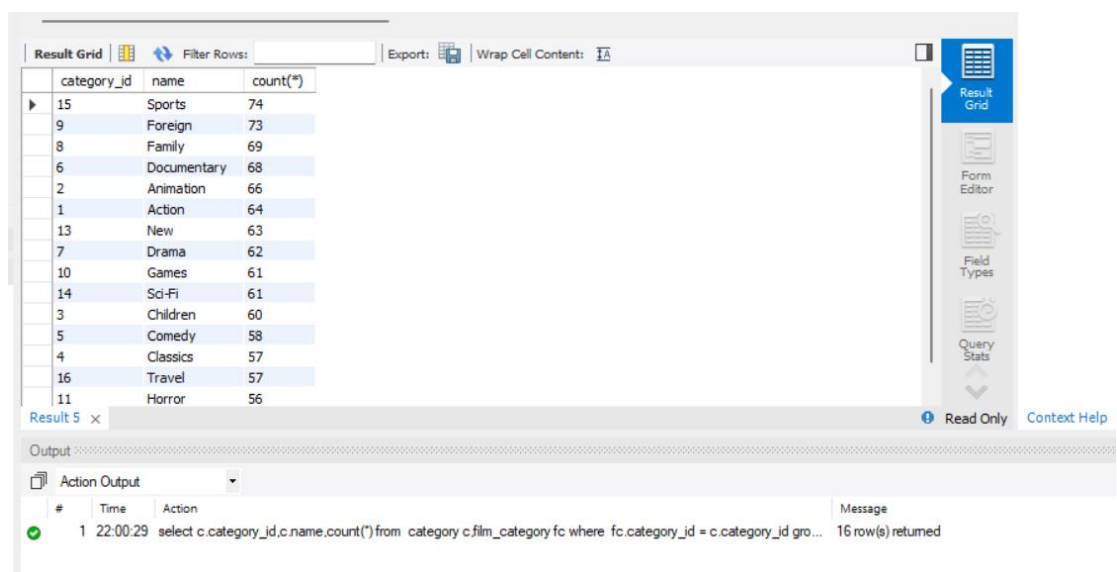
#	Time	Action	Message
1	21:48:23	select first_name,last_name from actor a,film_actor fa,film f where a.actor_id = fa.actor_id and fa.film_id = f...	2 row(s) returned

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 c.category_id,c.name,count(*)
from category c,film_category fc
where fc.category_id = c.category_id
group by c.category_id
order by count(*) desc;

```



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

Result 5 x

Output

Action Output

#	Time	Action	Message
1	22:00:29	select c.category_id,c.name,count(*) from category c,film_category fc where fc.category_id = c.category_id gro...	16 row(s) returned

9、有哪些影片是 2 个商店都有库存的？

```

select distinct title
from film,inventory inv1,inventory inv2
where
    film.film_id = inv1.film_id and
    inv2.film_id = film.film_id and
    inv1.store_id != inv2.store_id;

```

Result 12

title
ACADEMY DINOSAUR
AFFAIR PREJUDICE
AGENT TRUMAN
AIRPLANE SIERRA
ALABAMA DEVIL
ALADDIN CALENDAR

Output

#	Time	Action	Message
1	22:05:35	select distinct title from film,inventory inv1,inventory inv2 where film.film_id = inv1.film_id and inv2.film_id = fi...	563 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；

```

select first_name,last_name,timediff(return_date,rental_date)
from customer cus,rental ren
where
    cus.customer_id = ren.customer_id
order by timediff(return_date,rental_date) desc
limit 6;

```

Result 23

first_name	last_name	timediff(return_date,rental_date)
MARTIN	BALES	221:59:00
ELAINE	STEVENS	221:59:00
JACQUELINE	LONG	221:58:00
JAMES	GANNON	221:58:00
VERA	MCCOY	221:58:00
PEARL	GARZA	221:58:00

Output

#	Time	Action	Message
1	22:59:47	select first_name,last_name,timediff(retim_date,rental_date) from customer cus,rental ren where cus.custome...	6 row(s) returned

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','Yang','hang','2094684843@qq.com','300','1',now(),now());
select * from customer
order by customer_id desc;

```

Result Grid							
Filter Rows:							
	customer_id	store_id	first_name	last_name	email	address_id	active
▶	600	1	Yang	hang	2094684843@qq.com	300	1
	599	2	AUSTIN	CINTRON	AUSTIN.CINTRON@sakilacustomer.org	605	1
	598	1	WADE	DELVALLE	WADE.DELVALLE@sakilacustomer.org	604	1
	597	1	FREDDIE	DUGGAN	FREDDIE.DUGGAN@sakilacustomer.org	603	1
	596	1	ENRIQUE	FORSYTHE	ENRIQUE.FORSYTHE@sakilacustomer.org	602	1

Output							
Action Output							
#	Time	Action					Message
✓	1 23:12:00	insert into customer values('600','1','Yang','hang','2094684843@qq.com','300','1',now(),now())					1 row(s) affected
✓	2 23:12:00	select * from customer order by customer_id desc LIMIT 0, 1000					600 row(s) returned

12、 修改刚才在 customer 表中新增的那条数据；

UPDATE customer

SET last_name = '行',first_name = '杨'

Where customer_id = '600';

select * from customer

where customer_id = '600';

Result Grid							
Filter Rows:							
	customer_id	store_id	first_name	last_name	email	address_id	active
▶	600	1	杨	行	2094684843@qq.com	300	1
▶▶	NOLE	NOLE	NOLE	NOLE	NOLE	NOLE	NOLE

Output							
Action Output							
#	Time	Action					Message
✓	1 23:14:54	UPDATE customer SET last_name = '行',first_name = '杨' Where customer_id = '600'					0 row(s) affected Rows matched: 1 Changed: 0 Warnings: 0
✓	2 23:14:54	select * from customer where customer_id = '600' LIMIT 0, 1000					1 row(s) returned

13、 删除第 11 步新增的那条数据。

delete from customer

where customer.customer_id = '600';

select * from customer

order by customer_id desc;

Result Grid							
Filter Rows:							
	customer_id	store_id	first_name	last_name	email	address_id	active
▶	599	2	AUSTIN	CINTRON	AUSTIN.CINTRON@sakilacustomer.org	605	1
	598	1	WADE	DELVALLE	WADE.DELVALLE@sakilacustomer.org	604	1
	597	1	FREDDIE	DUGGAN	FREDDIE.DUGGAN@sakilacustomer.org	603	1
	596	1	ENRIQUE	FORSYTHE	ENRIQUE.FORSYTHE@sakilacustomer.org	602	1
	595	1	TERRENCE	GUNDERSON	TERRENCE.GUNDERSON@sakilacustomer.org	601	1

customer 27 x

Apply Context Help Snippets

Output

Action Output

#	Time	Action	Message
1	23:09:30	delete from customer where customer.customer_id = '600'	1 row(s) affected
2	23:09:30	select * from customer order by customer_id desc LIMIT 0, 1000	599 row(s) returned

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。

INSERT INTO actor

VALUES('1', 'GRACE', 'WHITE', NOW());

Output			
Action Output			
#	Time	Action	Message
1	23:16:59	INSERT INTO actor VALUES('1', 'GRACE', 'WHITE', NOW())	Error Code: 1062. Duplicate entry '1' for key 'actor.PRIMARY'

答：会发生报错，因为 actor_id 是 actor 表的主键，是区分 actor 表内信息的标志，不允许发生重复。

- 2) insert 语句还用了一个函数 NOW()，是做什么的呢？

答：用来记录当前的时间，用于保存维护时间信息如 create_date, last_update。