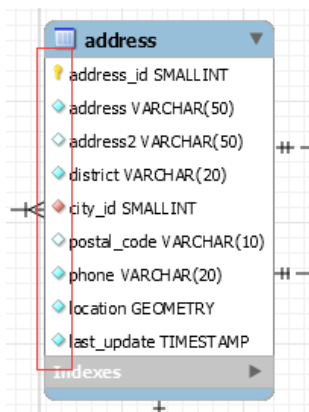


实验一报告

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？



标记	意义
	主键
	不能为空
	可以为空
	外码

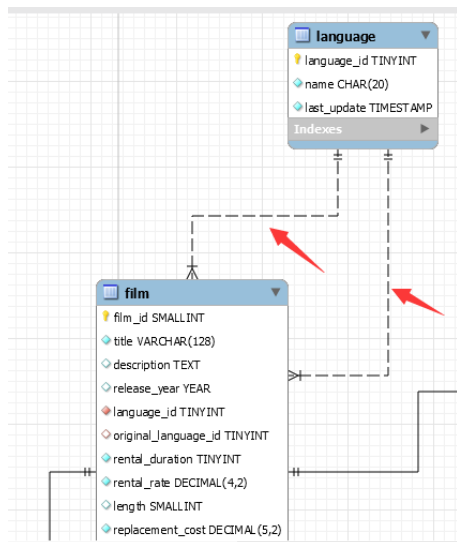
2. 图中哪部分体现影片-演员关系？换句话说，如果要找出演某个影片的演员名字，访问哪几张表可以获得信息？

访问 film, film_actor, actor，通过 film 表的 film_id 可以与 film_actor 表进行连接，通过 film_actor 的 actor_id 与 actor 连接即可。

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, inventory, rental, film 四张表

4. 为什么 film 和 language 表间会有 2 条线？



可以通过 film 表中的 language_id 和 original_language_id 都可以访问 language 表。

二、实验截图

注意截图清晰，截图时需要体现 SQL 语句（可复制粘贴）、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.store_id,address.address,address.location,address.district,city.city,country.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;
```

Result Grid						
	store_id	address	location	district	city	country
▶	1	47 MySakila Drive	BLOB	Alberta	Lethbridge	Canada
	2	28 MySQL Boulevard	BLOB	QLD	Woodridge	Australia

- 2、哪些演员出演过影片《WEST LION》？请列出他的姓名；
- ```
SELECT actor.first_name,actor.last_name
FROM film,film_actor,actor
WHERE film.film_id = film_actor.film_id AND actor.actor_id = film_actor.actor_id and film.title
= 'WEST LION';
```

| first_name | last_name   |
|------------|-------------|
| ZERO       | CAGE        |
| CAMERON    | STREEP      |
| SEAN       | WILLIAMS    |
| CARY       | MCCONAUGHEY |
| MORGAN     | HOPKINS     |
| CUBA       | ALLEN       |
| ALBERT     | JOHANSSON   |
| HUMPHREY   | GARLAND     |
| GREGORY    | GOODING     |

|    |          |                                                                                                                 |                   |
|----|----------|-----------------------------------------------------------------------------------------------------------------|-------------------|
| 23 | 16:26:56 | SELECT actor.first_name,actor.last_name FROM film,film_actor,actor WHERE film.film_id = film_actor.film_id A... | 9 row(s) returned |
|----|----------|-----------------------------------------------------------------------------------------------------------------|-------------------|

- 3、找出租 DVD 花费最高的前 3 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,payment
WHERE customer.customer_id = payment.customer_id
GROUP BY customer.customer_id
ORDER BY SUM(amount) DESC
LIMIT 3;
```

| first_name | last_name | SUM(amount) |
|------------|-----------|-------------|
| KARL       | SEAL      | 221.55      |
| ELEANOR    | HUNT      | 216.54      |
| CLARA      | SHAW      | 195.58      |

3 13:02:36 SELECT first\_name,last\_name,SUM(amount) FROM customer,payment WHERE customer.customer\_id = payme... 3 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,film.title,sum(payment.amount)
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
GROUP BY film.film_id
ORDER BY sum(payment.amount) DESC
LIMIT 1;
```

film_id	title	sum(payment.amount)
879	TELEGRAPH VOYAGE	231.73

25 16:29:52 SELECT film.film_id,film.title,sum(payment.amount) FROM film,inventory,rental,payment WHERE film.film_id = in... 1 row(s) returned

- 5、哪个演员出演的电影超过 40 部？请列出演员 id、演员名、出演的电影数；
- ```
SELECT actor.first_name,actor.last_name,count(film.film_id),actor.actor_id
FROM actor,film_actor,film
WHERE actor.actor_id = film_actor.actor_id AND film_actor.film_id = film.film_id
GROUP BY actor.actor_id HAVING count(film.film_id) > 40;
```

|   | first_name | last_name | count(film.film_id) | actor_id |
|---|------------|-----------|---------------------|----------|
| ▶ | WALTER     | TORN      | 41                  | 102      |
|   | GINA       | DEGENERES | 42                  | 107      |

26 16:30:40 SELECT actor.first\_name,actor.last\_name,count(film.film\_id),actor.actor\_id FROM actor,film\_actor,film WHERE ... 2 row(s) returned

6、 请找出没有租借过电影《WEST LION》的顾客姓名；

```
SELECT customer.first_name,customer.last_name
FROM customer
WHERE customer.customer_id NOT IN
(SELECT c.customer_id FROM customer c,film,rental,inventory
WHERE c.customer_id = rental.customer_id AND rental.inventory_id = inventory.inventory_id
AND inventory.film_id = film.film_id
AND film.title = 'WEST LION'
);
```

| first_name | last_name |
|------------|-----------|
| ▶ MARY     | SMITH     |
| PATRICIA   | JOHNSON   |
| LINDA      | WILLIAMS  |
| BARBARA    | JONES     |
| ELIZABETH  | BROWN     |
| JENNIFER   | DAVIS     |
| MARIA      | MILLER    |
| SUSAN      | WILSON    |
| MARGARET   | MOORE     |
| LISA       | ANDERSON  |
| NANCY      | THOMAS    |
| KADEN      | JACKSON   |

27 16:31:37 SELECT customer.first\_name,customer.last\_name FROM customer WHERE customer.customer\_id NOT IN (\$... 583 row(s) returned

7、 查询演过《FIRE WOLVES》和《JAWBREAKER BROOKLYN》这两部电影的演员，列出其姓名；

```
SELECT actor.first_name,actor.last_name
FROM actor,film,film_actor
WHERE film.film_id = film_actor.film_id AND film_actor.actor_id = actor.actor_id
AND film.title = 'FIRE WOLVES' AND actor.actor_id in (SELECT actor1.actor_id FROM film
film1,actor actor1,film_actor film_actor1
WHERE film1.film_id = film_actor1.film_id AND film_actor1.actor_id = actor1.actor_id AND
film1.title = 'JAWBREAKER BROOKLYN'
);
```

| first_name  | last_name |
|-------------|-----------|
| ▶ CHRISTIAN | AKROYD    |

28 16:32:42 SELECT actor.first\_name,actor.last\_name FROM actor,film,film\_actor WHERE film.film\_id = film\_actor.film\_id A... 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.category_id,count(category.category_id) num,category.name
FROM film,film_category,category
WHERE film.film_id = film_category.film_id AND film_category.category_id =
category.category_id
GROUP BY category.category_id;
```

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

29 16:33:24 SELECT category.category\_id,count(category.category\_id) num,category.name FROM film,film\_category,category 16 row(s) returned

9、有哪些影片是 2 个商店都有库存的？

```
SELECT DISTINCT film1.title
FROM film film1,inventory
WHERE film1.film_id = inventory.film_id AND inventory.store_id=1 AND inventory.film_id in (
SELECT inventory1.film_id
FROM inventory inventory1,film film2
WHERE inventory1.film_id = film2.film_id AND inventory1.store_id=2
);
```

|   | title            |
|---|------------------|
| ▶ | ACADEMY DINOSAUR |
|   | AFFAIR PREJUDICE |
|   | AGENT TRUMAN     |
|   | AIRPLANE SIERRA  |
|   | ALABAMA DEVIL    |
|   | ALADDIN CALENDAR |
|   | ALAMO VIDEOTAPE  |
|   | ALASKA PHANTOM   |
|   | ALIEN CENTER     |
|   | ALLEY EVOLUTION  |
|   | ALONE TRIP       |
|   | ALTER VICTORY    |
|   | AMADEUS HOLY     |

18 17:34:41 SELECT DISTINCT film1.title FROM film film1,inventory WHERE film1.film\_id = inventory.film\_id AND inventory.... 563 row(s) returned

10、 查询单次租借影片最长时间的 3 位客户，列出其 first\_name、last\_name 和当次租借时长；

```
SELECT customer.first_name,customer.last_name,unix_timestamp(rental.return_date) -
unix_timestamp(rental.rental_date) rental_time
FROM customer,rental
WHERE customer.customer_id = rental.customer_id
ORDER BY unix_timestamp(rental.return_date) - unix_timestamp(rental.rental_date) DESC
LIMIT 3;
```

|   | first_name | last_name | rental_time |
|---|------------|-----------|-------------|
| ▶ | MARTIN     | BALES     | 799140      |
|   | ELAINE     | STEVENS   | 799140      |
|   | JACQUELINE | LONG      | 799080      |

32 16:36:06 SELECT customer.first\_name,customer.last\_name,unix\_timestamp(rental.return\_date) - unix\_timestamp(rental.re... 3 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；  
通过查询主键的范围可知 customer\_id 可设为 700 不会冲突，根据参照完整性，其外码  
应选取作为主码所在表的值，store\_id 和 address\_id 可以设置为 1,5

INSERT customer VALUES (700,1,'张','三','abc@163.com',5,1,NOW(),NOW());

33 16:40:25 INSERT customer VALUES (700,1,'张','三','abc@163.com',5,1,NOW(),NOW()) 1 row(s) affected

插入后可以进行一条查询语句

SELECT last\_update FROM customer WHERE customer.first\_name = '张';

|             |                     |                                                                                     |                                                                                     |        |
|-------------|---------------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|--------|
| Result Grid |                     |  |  | Filter |
|             | last_update         |                                                                                     |                                                                                     |        |
| ▶           | 2021-09-15 16:40:25 |                                                                                     |                                                                                     |        |

12、 修改刚才在 customer 表中新增的那条数据；

UPDATE customer SET last\_update=NOW() WHERE customer\_id=700;

35 16:42:06 UPDATE customer SET last\_update=NOW() WHERE customer\_id=700 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0

再次执行 11 所述的查询语句

Result Grid

Filter Rows:



|   |                     |
|---|---------------------|
|   | last_update         |
| ▶ | 2021-09-15 16:42:06 |

13、 删除第 11 步新增的那条数据。

DELETE FROM customer WHERE customer\_id=700;

37 16:46:23 DELETE FROM customer WHERE customer\_id=700 1 row(s) affected

再次执行 11 所述的查询语句

|             |             |                                                                                     |                                                                                     |             |
|-------------|-------------|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|-------------|
| Result Grid |             |  |  | Filter Rows |
|             | last_update |                                                                                     |                                                                                     |             |

### 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。

```
Error Code: 1062. Duplicate entry '201' for key 'actor.PRIMARY'
```

会报错，因为主键不能重复。

- 2) insert 语句还用了一个函数 NOW()，是做什么的呢？  
获取当前系统时间作为插入该数据的更新时间。