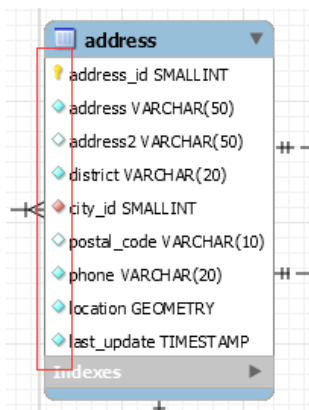


实验一报告

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
	非空
	可以为空
	非空外键

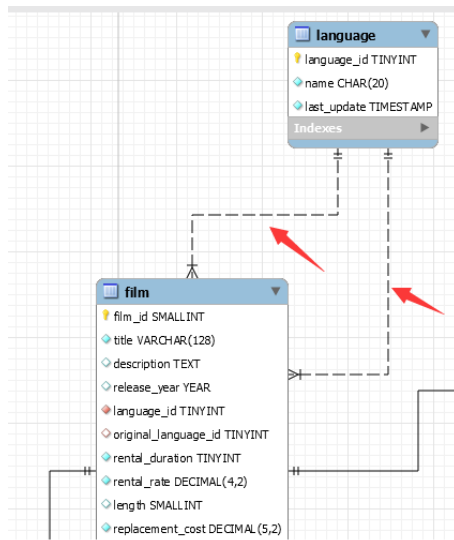
2. 图中哪部分体现影片-演员关系？换句话说，如果要找出演某个影片的演员名字，访问哪几张表可以获得信息？

三张表 film, film_actor 和 actor 体现了影片-演员关系。通过 film 表得到一部电影唯一对应的 film_id，这是 film_actor 的外码，通过 film_id 相同的 film_actor 中的元组，可以找到出演这部电影的演员的 actor_id。actor_id 是 actor 表的主键，和演员一一对应。这样就可以通过 film_id 找到全部出演这个电影的演员姓名。

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？需要访问四张表 customer, rental, inventory, film。每位 customer 有唯一对应的 customer_id，同时是 rental 表的外键，记录了这位顾客所有租借记录。通过 rental

查找出对应记录的 inventory_id，这是 inventory 表的主键。每个 inventory_id 对应一个 film_id，根据 film_id 在 film 表中就能找到对应的电影名字。

4. 为什么 film 和 language 表间会有 2 条线？



film 表中 language_id 和 original_language_id 都是 film 表的外键，与 language 表的主键 language_id 相对应，所以有两条线与 language 相连。

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_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
```

Result Grid					
Filter Rows: <input type="text"/>					
Export:					
Wrap Cell Content:					
store_id	address	district	city	country	
1	47 MySakila Drive	Alberta	Lethbridge	Canada	
2	28 MySQL Boulevard	QLD	Woodridge	Australia	

Result 2 x

Output

Action Output

#	Time	Action	Message
1	21:18:57	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...	2 row(s) returned

2、 哪些演员出演过影片《WEST LION》？ 请列出他的姓名；

```

select
    first_name,
    last_name
from
    film,
    film_actor,
    actor
where
    film.title = 'WEST LION'
    and film.film_id = film_actor.film_id
    and film_actor.actor_id = actor.actor_id

```

Result Grid					
Filter Rows: <input type="text"/>					
Export:					
Wrap Cell Content:					
first_name	last_name				
ZERO	CAGE				
CAMERON	STREEP				
SEAN	WILLIAMS				
CARY	MCCONAUGHEY				
MORGAN	HOPKINS				
CUBA	ALLEN				
ALBERT	JOHANSSON				
HUMPHREY	GARLAND				
GREGORY	GOODING				

Result 1 x

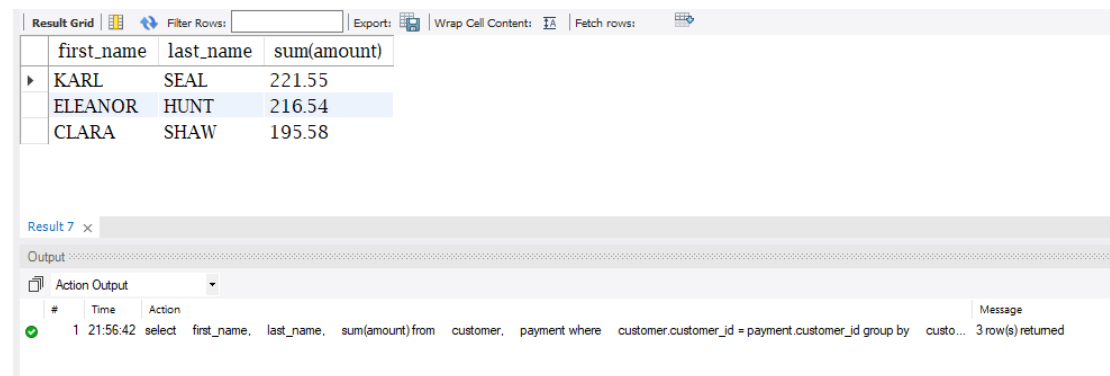
Output

Action Output

#	Time	Action	Message
1	21:18:57	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...	2 row(s) returned
2	21:21:00	select first_name, last_name from film, film_actor, actor where film.title = 'WEST LION' and film.film_id = film_actor.film_id ...	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

Result 7 x

Output

Action Output

#	Time	Action	Message
1	21:56:42	select first_name, last_name, sum(amount) from customer, payment where customer.customer_id = payment.customer_id group by custo...	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,
    title,
    sum(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(amount) desc
limit
    1
```

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

Result 4 ×			
Output			
Action Output			
#	Time	Action	Message
✓	1 22:11:00	select film.film_id, title, sum(amount) from film, inventory, rental, payment where film.film_id = inventory.film_id and inventory.inventor...	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.actor_id,
    actor.first_name,
    actor.last_name,
    count(*)
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(*) > 40

```

Result Grid				
	actor_id	first_name	last_name	count(*)
▶	102	WALTER	TORN	41
	107	GINA	DEGENERES	42

Result 2 ×				
Output				
Action Output				
#	Time	Action	Message	
✓	1 22:21:15	select actor.actor_id, actor.first_name, actor.last_name, count(*) from actor, film_actor, film where actor.actor_id = film_actor.actor_id ...	2 row(s) returned	

6、 请找出没有租借过电影《WEST LION》的顾客姓名；

```
select
    first_name,
    last_name
from
    customer
where
    customer_id not in(
        select
            customer_id
        from
            film,
            inventory,
            rental
        where
            film.film_id = inventory.film_id
            and inventory.inventory_id = rental.inventory_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
KAREN	JACKSON
BETTY	WHITE
HELEN	HARRIS
SANDRA	MARTIN
DONNA	THOMPSON
CAROL	GARCIA
RUTH	MARTINEZ
MICHELLE	CLARK
LAURA	RODRIGUEZ

customer 5 x

Output

Action Output

#	Time	Action	Message
1	22:29:45	select first_name, last_name from customer where customer_id not in(select customer_id from film, inventory, ...	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 distinct
    first_name,
    last_name
from
    actor a,
    film_actor fa1,
    film_actor fa2,
    film f1,
    film f2
where
    a.actor_id = fa1.actor_id
    and a.actor_id = fa2.actor_id
    and fa1.film_id = f1.film_id
    and fa2.film_id = f2.film_id
    and f1.title = 'FIRE WOLVES'
    and f2.title = 'JAWBREAKER BROOKLYN'
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	first_name	last_name
▶	CHRISTIAN	AKROYD

Result 5

Output

Action Output

#	Time	Action	Message
1	22:49:24	select distinct first_name, last_name from actor a, film_actor fa1, film_actor fa2, film f1, film f2 where a.actor_id = fa1.actor_id and 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,
    category.name,
    count(*)
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:

Export:

Wrap Cell Content:

	category_id	name	count(*)
▶	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

Result 1

Output

Action Output

#	Time	Action	Message
✓ 1	22:49:24	select distinct first_name, last_name from actor a, film_actor fa1, film_actor fa2, film f1, film f2 where a.actor_id = fa1.actor_id and a...	1 row(s) returned
✓ 2	22:52:52	select category.category_id, category.name, count(*) from film, film_category, category where film.film_id = film_category.film_id and film_cat...	16 row(s) returned

9、 有哪些影片是 2 个商店都有库存的？

```
select distinct
    film.film_id,
    film.title
from
    film,
    inventory i1,
    inventory i2
where
    film.film_id = i1.film_id
    and film.film_id = i2.film_id
    and i1.store_id < i2.store_id
```

Result Grid		Filter Rows:	Export:	Wrap Cell Contents:
film_id	title			
1	ACADEMY DINOSAUR			
4	AFFAIR PREJUDICE			
6	AGENT TRUMAN			
7	AIRPLANE SIERRA			
9	ALABAMA DEVIL			
10	ALADDIN CALENDAR			
11	ALAMO VIDEOTAPE			
12	ALASKA PHANTOM			
15	ALIEN CENTER			
16	ALLEY EVOLUTION			
17	ALONE TRIP			
18	ALTER VICTORY			
19	AMADEUS HOLY			
21	AMERICAN CIRCUS			
22	AMISTAD MIDSUMMER			
23	ANACONDA CONFES...			
25	ANGELS LIFE			
26	ANNIE IDENTITY			
31	APACHE DIVINE			
35	ARACHNOPHOBIA R...			
37	ARIZONA BANG			
39	ARMAGEDDON LOST			
43	ATLANTIS CAUSE			
44	ATTACKS HATE			

Result 3 x

Output

Action Output

#	Time	Action	Message
1	13:25:22	select distinct film.film_id, film.title from film, inventory i1, inventory i2 where film.film_id = i1.film_id and film.film_id = i2.film_id ...	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
    first_name,
    last_name,
    timestampdiff(second, rental_date, return_date) rental_time_second
from
    customer,
    rental
where
    customer.customer_id = rental.customer_id
order by
    rental_time_second desc
limit
    3
```


Result Grid			
	Filter Rows:	Exports	Wrap Cell Content: Fetch rows:
first_name	last_name	rental_time_second	
MARTIN	BALES	799140	
ELAINE	STEVENS	799140	
JACQUELINE	LONG	799080	

Result 13			
Output			
Action Output			
#	Time	Action	Message
1	11:14:48	select first_name, last_name, timestampdiff(second, rental_date, return_date) rental_time from customer, rental where customer...	3 row(s) returned
2	11:15:45	select first_name, last_name, timestampdiff(second, rental_date, return_date) rental_time_second from customer, rental where ...	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；

```
insert into
  customer
value(
  600,
  1,
  'AILAN',
  'XIER',
  'AILANXIER@123.com',
  1,
  1,
  now(),
  now()
)
```

Result Grid									
Filter Rows:		Edits:		Export/Imports:		Wrap Cell Content:			
customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update	
595	1	TERRENCE	GUNDERS...	TERRENCE.GUNDERSON@sak...	601	1	2006-02-14 22:04:37	2006-02-15 04:57:20	
596	1	ENRIQUE	FORSYTHE	ENRIQUE.FORSYTHE@sakilac...	602	1	2006-02-14 22:04:37	2006-02-15 04:57:20	
597	1	FREDDIE	DUGGAN	FREDDIE.DUGGAN@sakilacus...	603	1	2006-02-14 22:04:37	2006-02-15 04:57:20	
598	1	WADE	DELVALLE	WADE.DELVALLE@sakilacust...	604	1	2006-02-14 22:04:37	2006-02-15 04:57:20	
599	2	AUSTIN	CINTRON	AUSTIN.CINTRON@sakilacust...	605	1	2006-02-14 22:04:37	2006-02-15 04:57:20	
600	1	AILAN	XIER	AILANXIER@123.com	1	1	2021-09-17 14:40:38	2021-09-17 14:40:38	
* <div>filter</div> <div>filter</div> <div>filter</div> <div>filter</div> <div>filter</div> <div>filter</div> <div>filter</div> <div>filter</div>									
customer 3 x									
Output									
Action Output									
#	Time	Action					Message		
✓ 1	14:40:38	insert into customer value(600,1,'AILAN','XIER','AILANXIER@123.com',1,1,now(),now())					1 row(s) affected		
✓ 2	14:40:51	SELECT * FROM sakila.customer 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
  email = 'AIMOXIER@321.com',
  last_update = now()
where
  customer_id = 600
```

Result Grid										Filter Rows:		Edits		Export/Import:		Wrap Cell Content:		15	
	customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update										
	597	1	FREDDIE	DUGGAN	FREDDIE.DUGGAN@sakilacus...	603	1	2006-02-14 22:04:37	2006-02-15 04:57:20										
	598	1	WADE	DELVALLE	WADE.DELVALLE@sakilacust...	604	1	2006-02-14 22:04:37	2006-02-15 04:57:20										
	599	2	AUSTIN	CINTRON	AUSTIN.CINTRON@sakilacust...	605	1	2006-02-14 22:04:37	2006-02-15 04:57:20										
	600	1	AILAN	XIER	AIMOXIER@321.com	1	1	2021-09-17 14:40:38	2021-09-17 14:47:29										
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL										
customer 5																			
Output																			
Action Output																			
	#	Time	Action		Message														
	1	14:47:29	update	customer set email = 'AIMOXIER@321.com', last_update = now() where customer_id = 600	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0														
	2	14:47:33	SELECT *	FROM sakila.customer LIMIT 0, 1000	600 row(s) returned														

13、 删除第 11 步新增的那条数据。

```
delete from
  customer
where
  customer_id = 600
```

Result Grid										Edit: Filter Rows: <input type="text"/>		Export/Import: Wrap Cell Content:	
	customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update				
	596	1	ENRIQUE	FORSYTHE	ENRIQUE.FORSYTHE@sakilac...	602	1	2006-02-14 22:04:37	2006-02-15 04:57:20				
	597	1	FREDDIE	DUGGAN	FREDDIE.DUGGAN@sakilacus...	603	1	2006-02-14 22:04:37	2006-02-15 04:57:20				
	598	1	WADE	DELVALLE	WADE.DELVALLE@sakilacust...	604	1	2006-02-14 22:04:37	2006-02-15 04:57:20				
	599	2	AUSTIN	CINTRON	AUSTIN.CINTRON@sakilacust...	605	1	2006-02-14 22:04:37	2006-02-15 04:57:20				
*													
customer 6 x													
Output													
Action Output													
	#	Time	Action					Message					
✓	1	14:47:29	update	customer set	email = 'AIMOXIER@321.com',	last_update = now()	where	customer_id = 600	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0				
✓	2	14:47:33	SELECT *	FROM sakila.customer LIMIT 0, 1000							600 row(s) returned		
✓	3	14:48:18	delete from	customer where	customer_id = 600						1 row(s) affected		
✓	4	14:48:29	SELECT *	FROM sakila.customer 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。

输入更新代码：

```
insert into
  actor
value(
  1,
  'AILAN',
  'XIER',
  now()
)
```

错误提示：

Output									
Action Output									
#	Time	Action					Message		
1	14:53:35	insert into	actor value(1,	'AILAN',	'XIER',	now()	Error Code: 1062. Duplicate entry '1' for key 'PRIMARY'	

显示因为重复主键属性二无法插入该新元组。因为在 actor 表中 actor_id 是主键，已存在主键为 1 的元组，现在再插入主键为 1 的新元组，会破坏主键的唯一性索引属性，操作不合法。

2) insert 语句还用了一个函数 NOW(), 是做什么的呢？

返回当前日期和时间值，格式为「YYYY-MM-DD HH: MM: SS」或「YYYYMMDDHHMMSS」，具体格式根据函数用在字符串或数字语境中而定。