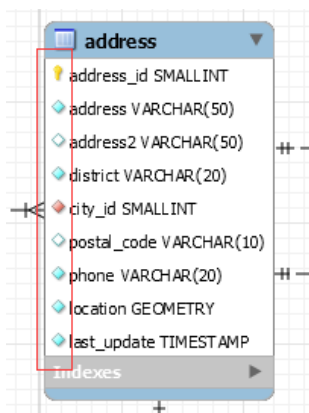


实验一报告

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
◆	非主属性(非空)
◇	默认空值
◆	外键(非空)

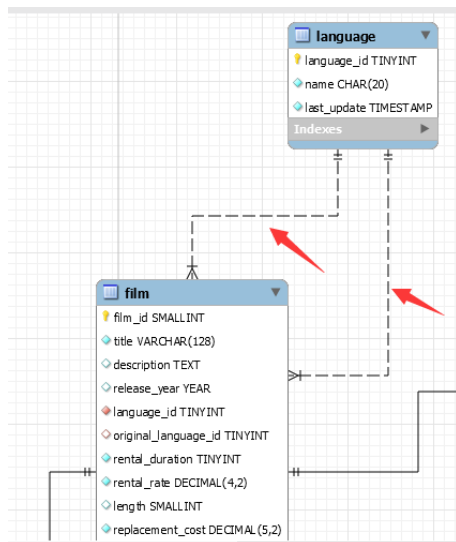
2. 图中哪部分体现影片-演员关系？换句话说，如果要找出演某个影片的演员名字，访问哪几张表可以获得信息？

需要访问 `film`, `film_actor`, `actor` 这三张表

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？

需要访问 `customer`, `rental`, `inventory`, `film` 这四张表

4. 为什么 `film` 和 `language` 表间会有 2 条线？



表示这两张表可以通过外键 language_id, original_language 这相连

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 address,city,country,store
where store.address_id = address.address_id and address.city_id=city.city_id and
city.country_id=country.country_id
```

49 19:14:26 select store_id,address,district,city,country from address,city,country,store where store.address_id = address.address_id and address.city_id=c... 2 row(s) returned 0.016 sec / 0.000 sec
50 19:14:59 SELECT * FROM sakila.store 2 row(s) returned 0.016 sec / 0.000 sec

	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《WEST LION》？请列出他的姓名；

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

51 19:15:57 select first_name, last_name from actor, film, film_actor where actor.actor_id = film_actor.actor_id and film.film_id = film_actor.film_id and film.tit... 9 row(s) returned 0.016 sec / 0.000 sec

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

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 payment, customer
where payment.customer_id = customer.customer_id
group by customer.customer_id
order by sum(amount) DESC limit 3
```

52 19:16:41 select first_name, last_name, sum(amount) from payment, customer where payment.customer_id = customer.customer_id group by customer.c... 3 row(s) returned 0.046 sec / 0.000 sec

	first_name	last_name	sum(amount)
▶	KARL	SEAL	221.55
	ELEANOR	HUNT	216.54
	CLARA	SHAW	195.58

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
```

53 19:17:11 select film.film_id, title, sum(amount) from film, inventory, rental, payment where film.film_id = inventory.film_id and inventory.inventory_id = rental... 1 row(s) returned 0.078 sec / 0.000 sec

	film_id	title	sum(amount)
▶	879	TELEGRAPH VOYAGE	231.73

TELEGRAPH VOYAGE

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, first_name, last_name, count(film.film_id)
```

```

from film_actor, film, actor
where film_actor.film_id = film.film_id and film_actor.actor_id = actor.actor_id
group by film_actor.actor_id
having count(film.film_id) >= 40

```

54 19:17:46 select actor.actor_id, first_name, last_name, count(film.film_id) from film_actor, film, actor where film_actor.film_id = film.film_id and film_actor.actor_id = actor.actor_id; 3 row(s) returned 0.032 sec / 0.000 sec

	actor_id	first_name	last_name	count(film.film_id)
▶	102	WALTER	TORN	41
	107	GINA	DEGENERES	42
	198	MARY	KEITEL	40

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 distinct customer.customer_id
    from customer, film, inventory, rental
    where film.film_id = inventory.film_id and inventory.inventory_id = rental.inventory_id
    and rental.customer_id = customer.customer_id and film.title = "WEST LION")

```

54 19:17:46 select actor.actor_id, first_name, last_name, count(film.film_id) from film_actor, film, actor where film_actor.film_id = film.film_id and film_actor.actor_id = actor.actor_id; 3 row(s) returned 0.032 sec / 0.000 sec

Result Grid | Filter Rows:

	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

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 film f1, film f2, film_actor a1, film_actor a2, actor
where f1.film_id = a1.film_id and f2.film_id = a2.film_id and a1.actor_id = a2.actor_id and
actor.actor_id = a1.actor_id

```

and f1.title = "FIRE WOLVES" and f2.title = "JAWBREAKER BROOKLYN"

56 19:19:00 select actor.first_name, actor.last_name from film f1, film f2, film_actor a1, film_actor a2, actor where f1.film_id = a1.film_id and f2.film_id = a2.film_id and a1.actor_id = a2.actor_id and f1.title = "FIRE WOLVES" and f2.title = "JAWBREAKER BROOKLYN" 1 row(s) returned 0.000 sec / 0.000 sec

	first_name	last_name
▶	CHRISTIAN	AKROYD

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(film.film_id)
```

```
from category, film_category, film
```

```
where category.category_id = film_category.category_id and film.film_id = film_category.film_id
```

```
group by category.category_id
```

```
order by category.category_id
```

59 19:20:41 select category.category_id, category.name, count(film.film_id) from category, film_category, film where category.category_id = film_category.category_id 16 row(s) returned

	category_id	name	count(film.film_id)
▶	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

9.有哪些影片是 2 个商店都有库存的？

```
select distinct film.title
```

```
from film, inventory i1, inventory i2, store s1, store s2
```

```
where film.film_id = i1.film_id and film.film_id = i2.film_id and s1.store_id = i1.store_id
```

```
and s2.store_id = i2.store_id and s1.store_id <> s2.store_id
```

60 19:21:18 select distinct film.title from film, inventory i1, inventory i2, store s1, store s2 where film.film_id = i1.film_id and film.film_id = i2.film_id and s1.store_id = i1.store_id and s2.store_id = i2.store_id and s1.store_id <> s2.store_id 563 row(s) returned

	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
	AMERICAN CIRCUS
	AMISTAD MIDSUMMER
	ANACONDA CONFES...

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(microsecond, rental.rental_date, rental.return_date)
```

```
from rental, customer
```

```
where rental.customer_id = customer.customer_id
```

```
order by(timestampdiff(microsecond, rental.rental_date, rental.return_date)) DESC
```

```
limit 3
```

61 19:23:06 select first_name, last_name, timestampdiff(microsecond, rental.rental_date, rental.return_date) from rental, customer where rental.customer_id = ... 3 row(s) returned

	first_name	last_name	timestampdiff(microsecond, rental.rental_date, rental.return_date)
▶	MARTIN	BALES	799140000000
	ELAINE	STEVENS	799140000000
	JACQUELINE	LONG	799080000000

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

```
insert into address
```

```
values(606, '1329
```

Fukuyama

```
Street', null, 'Heilongjiang', 537, 27107, 288241215399, st_geomfromtext('POINT (128.0449753 46.9804391)'), now());
```

```
insert into customer
```

```
values(600, 1, "Luis", "Sword", 'Sword.Luis@sakilacustomer.org', 606, 1, now(), now());
```

63 19:23:37 insert into customer values(600, 1, "Luis", "Sword", 'Sword.Luis@sakilacustomer.org', 606, 1, now(), now())

1 row(s) affected

605	1325	Fukuyama Street		Heilongjiang	537	27107	288241215394	BLOB	2014-09-25 22:30:44
606	1329	Fukuyama Street	NULL	Heilongjiang	537	27107	288241215399	BLOB	2021-09-19 19:23:37

599	2	AUSTIN	CINTRON	AUSTIN.CINTRON@sakilacustomer.org	605	1	2006-02-14 22:04:37	2006-02-15 04:57:20
600	1	Luis	Sword	Sword.Luis@sakilacustomer.org	606	1	2021-09-19 19:23:37	2021-09-19 19:23:37

12. 修改刚才在 customer 表中新增的那条数据;

update customer

set email = 'Luis.Sword@sakilacustomer.org'

where customer_id = 600

67	19:25:20	update customer set email = 'Luis.Sword@sakilacustomer.org' where customer_id = 600					1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	
599	2	AUSTIN	CINTRON	AUSTIN.CINTRON@sakilacustomer.org	605	1	2006-02-14 22:04:37	2006-02-15 04:57:20
600	1	Luis	Sword	Luis.Sword@sakilacustomer.org	606	1	2021-09-19 19:23:37	2021-09-19 19:25:20

13.删除第 11 步新增的那条数据。

delete from customer

where customer_id = 600;

delete from address

where address_id = 606;

✓	68	19:25:53	delete from customer where customer_id = 600	1 row(s) affected
✓	69	19:25:53	delete from address where customer_id = 600	1 row(s) affected

604	1331 Usak Boulevard	Vaud	296	61960	145308717464	BL00	2014-09-25 22:33:14
605	1325 Fukuyama Street	Heilongjiang	537	27107	288241215394	BL00	2014-09-25 22:30:44
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

598	1	WADE	DELVALLE	WADE.DELVALLE@sakilacustomer.org	604	1	2006-02-14 22:04:37	2006-02-15 04:57:20
599	2	AUSTIN	CINTRON	AUSTIN.CINTRON@sakilacustomer.org	605	1	2006-02-14 22:04:37	2006-02-15 04:57:20
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	

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。

会显示 duplicate entry, 插入错误, 因为 actor_id 是主键, 不能有重复值

insert into actor

values(1,'PENELOPE','GUINNESS',now());

72	19:29:02	insert into actor values(1, 'PENELOPE', 'GUINNESS', now())	Error Code: 1062. Duplicate entry '1' for key 'actor.PRIMARY'
----	----------	--	---

2) insert 语句还用了一个函数 NOW(), 是做什么的呢?

NOW() 函数获取当前的时间以 datetime 格式输出