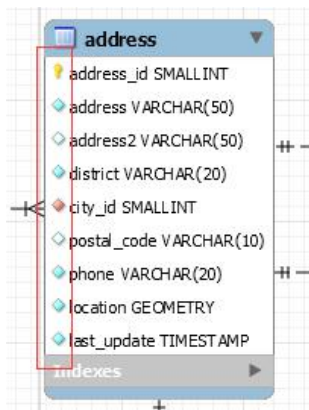


实验一报告

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
💎	非空
◇	默认为空
💎	非空整数

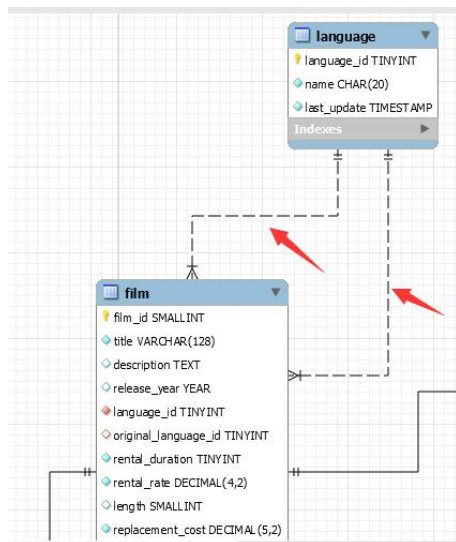
2. 图中哪部分体现影片-演员关系？换句话说，如果要找出演某个影片的演员名字，访问哪几张表可以获得信息？

答：访问 film、film_actor、actor 可以获得影片-演员关系。

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：访问 customer、rental、inventory、film

4. film 和 language 表间的 2 条虚线表示什么意思？



答：表示 film 表中的 language_id 和 original_language_id 是指向 language 表的外键。

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；
sql 语句：

```
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;
```

执行结果：

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

output 窗口：

165	17:09:00	SELECT first_name,...	0 row(s) returned
166	19:11:09	select store_id, add...	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;

sql 语句：

```
SELECT first_name, last_name
```

```

from film, film_actor, actor
where film.film_id = film_actor.film_id and film_actor.actor_id = actor.actor_id and film.title =
'ROCKY WAR';
执行结果：

```

	first_name	last_name
▶	SISSY	SOBIESKI
	PENELOPE	CRONYN
	RENEE	TRACY

output 窗口：

✓	165	17:39:55	SELECT first_name,...	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；



sql 语句：

```



select first_name, last_name, sum(amount)
from payment, rental, customer
where payment.rental_id = rental.rental_id and rental.customer_id = customer.customer_id
group by customer.customer_id
order by sum(amount) desc
limit 5;
执行结果：

```

Result Grid



Filter Rows:

Export:

	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

output 窗口：

✓	187	20:13:53	select first_name, la...	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；

sql 语句：

```
select film.film_id, title, sum(amount)
from payment, rental, inventory, film
where payment.rental_id = rental.rental_id and rental.inventory_id = inventory.inventory_id and
inventory.film_id = film.film_id
group by film.film_id
order by sum(amount) desc
limit 1;
```

执行结果：

Result Grid	Filter Rows:	Export:	Wrap C
film_id	title	sum(amount)	
879	TELEGRAPH VOYAGE	231.73	

output 窗口：

#	Time	Action	Message
1	20:31:15	select film.film_id, title, sum(amount) from payment, rent...	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；

sql 语句：


```
select actor.actor_id, first_name, last_name, count(*) film_num
from film, film_actor, actor
where film.film_id = film_actor.film_id and film_actor.actor_id = actor.actor_id
group by actor.actor_id having count(*) > 35;
```

执行结果：

Result Grid

 Filter Rows:

Export:



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

output 窗口：

Output

📄 Action Output ▾

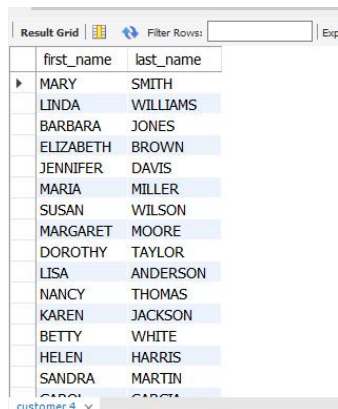
#	Time	Action	Message
✓ 1	20:40:40	select actor.actor_id, first_name, last_name, count(*) fil...	6 row(s) returned

6、 请找出没有租借过电影《TELEGRAPH VOYAGE》的顾客姓名；

sql 语句：

```
select first_name, last_name
from customer
where customer_id not in
    (select customer.customer_id
     from customer, rental, inventory, film
     where customer.customer_id = rental.customer_id and rental.inventory_id =
inventory.inventory_id and
        inventory.film_id = film.film_id and title = 'TELEGRAPH VOYAGE');
```

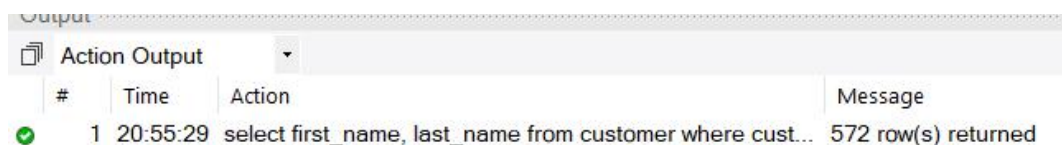
执行结果：



The screenshot shows a 'Result Grid' window with a table containing two columns: 'first_name' and 'last_name'. The table lists 20 customers, starting with MARY SMITH and ending with SANDRA MARTIN. The first row is expanded, showing the details for MARY SMITH.

first_name	last_name
MARY	SMITH
LINDA	WILLIAMS
BARBARA	JONES
ELIZABETH	BROWN
JENNIFER	DAVIS
MARIA	MILLER
SUSAN	WILSON
MARGARET	MOORE
DOROTHY	TAYLOR
LISA	ANDERSON
NANCY	THOMAS
KAREN	JACKSON
BETTY	WHITE
HELEN	HARRIS
SANDRA	MARTIN

output 窗口：



The screenshot shows an 'Output' window with a tab labeled 'Action Output'. It displays a single row of execution information: a green checkmark, the number 1, the time 20:55:29, the SQL query text, and a message stating '572 row(s) returned'.

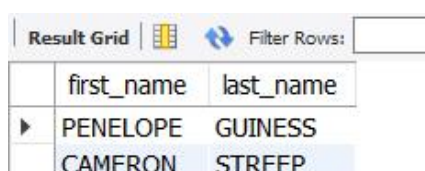
#	Time	Action	Message
1	20:55:29	select first_name, last_name from customer where cust...	572 row(s) returned

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

sql 语句：

```
select first_name, last_name
from actor, film_actor, film f1
where actor.actor_id = film_actor.actor_id and film_actor.film_id = f1.film_id
and f1.title = 'ELEPHANT TROJAN' and actor.actor_id in
    (select actor.actor_id
     from actor, film_actor, film f2
     where actor.actor_id = film_actor.actor_id and film_actor.film_id = f2.film_id
     and f2.title = 'SPLASH GUMP');
```

执行结果：



The screenshot shows a 'Result Grid' window with a table containing two columns: 'first_name' and 'last_name'. The table lists two actors: PENELOPE GUINNESS and CAMERON STREEP. The first row is expanded, showing the details for PENELOPE GUINNESS.

first_name	last_name
PENELOPE	GUINNESS
CAMERON	STREEP

output 窗口:

Output			
Action Output			
#	Time	Action	Message
1	21:07:55	select first_name, last_name from actor, film_actor, film ...	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；

sql 语句:

```
select category.category_id, category.name, count(*) total
from category, film_category
where category.category_id = film_category.category_id
group by category.category_id;
```

执行结果:

Result Grid			
Filter Rows:			
	category_id	name	total
▶	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

output 窗口:

Output			
Action Output			
#	Time	Action	Message
1	21:15:31	select category.category_id, category.name, count(*) to...	16 row(s) returned

9、有哪些影片是 2 个商店都有库存的？

sql 语句:

```
select distinct title
from inventory i1, inventory i2, film
where i1.film_id = i2.film_id and i1.film_id = film.film_id and i1.store_id = 1 and i2.store_id = 2
```

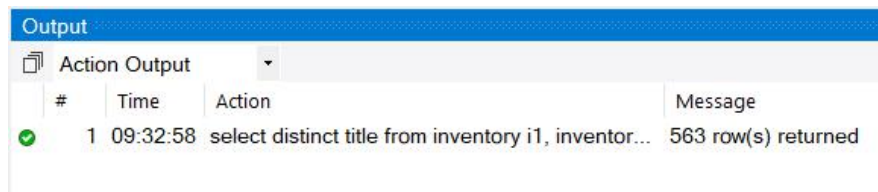
执行结果:



Result Grid | Filter Rows:

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...
ANGELS LIFE

output 窗口:



Output

Action Output

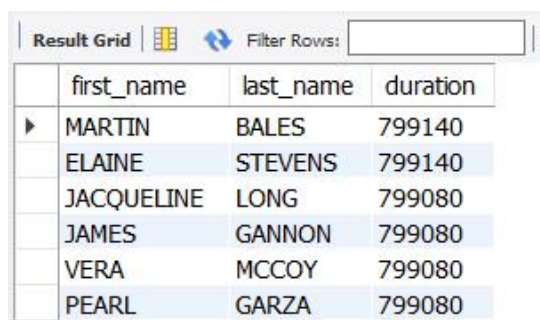
#	Time	Action	Message
1	09:32:58	select distinct title from inventory i1, inventor...	563 row(s) returned

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;

sql 语句:

```
select first_name, last_name, (UNIX_TIMESTAMP(return_date) - UNIX_TIMESTAMP(rental_date))  
duration  
from rental, customer  
where rental.customer_id = customer.customer_id  
order by UNIX_TIMESTAMP(return_date) - UNIX_TIMESTAMP(rental_date) desc  
limit 6
```

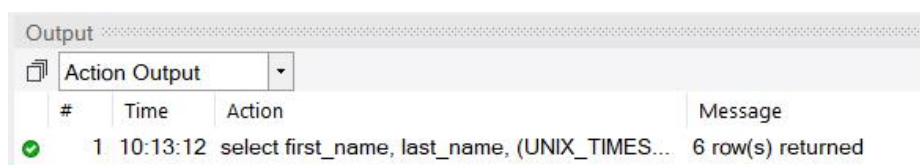
执行结果:



Result Grid | Filter Rows:

first_name	last_name	duration
MARTIN	BALES	799140
ELAINE	STEVENS	799140
JACQUELINE	LONG	799080
JAMES	GANNON	799080
VERA	MCCOY	799080
PEARL	GARZA	799080

output 窗口:



Output

Action Output

#	Time	Action	Message
1	10:13:12	select first_name, last_name, (UNIX_TIMES...	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；

sql 语句：

```
INSERT INTO customer
```

```
VALUES (600, 1, 'Allen', 'Iverson', 'Allen.Iverson@sakilacustomer.org', 1, 1, now(), '2022-11-15 04:57:20');
```

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
591	1	KENT	ARSENAU...	KENT.ARSENAULT@sakilacustomer.org	597	1	2006-02-14 22:04:37	2006-02-15 04:57:20
592	1	TERRANCE	ROUSH	TERRANCE.ROUSH@sakilacustomer....	598	0	2006-02-14 22:04:37	2006-02-15 04:57:20
593	2	RENE	MCALISTER	RENE.MCALISTER@sakilacustomer.org	599	1	2006-02-14 22:04:37	2006-02-15 04:57:20
594	1	EDUARDO	HIATT	EDUARDO.HIATT@sakilacustomer.org	600	1	2006-02-14 22:04:37	2006-02-15 04:57:20
595	1	TERRENCE	GUNDER...	TERRENCE.GUNDERSON@sakilacust...	601	1	2006-02-14 22:04:37	2006-02-15 04:57:20
596	1	ENRIQUE	FORSYTHE	ENRIQUE.FORSYTHE@sakilacustome...	602	1	2006-02-14 22:04:37	2006-02-15 04:57:20
597	1	FREDDIE	DUGGAN	FREDDIE.DUGGAN@sakilacustomer.org	603	1	2006-02-14 22:04:37	2006-02-15 04:57:20
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
600	1	Allen	Iverson	Allen.Iverson@sakilacustomer.org	1	1	2022-11-29 10:33:20	2022-11-15 04:57:20

output 窗口：

Output			
Action Output			
#	Time	Action	Message
1	10:33:20	INSERT INTO customer VALUES (600,...	1 row(s) affected

12、 修改刚才在 customer 表中新增的那条数据；

sql 语句：

```
update customer
```

```
set last_update = now()
```

```
where customer_id = 600;
```

执行结果：

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
591	1	KENT	ARSENAU...	KENT.ARSENAULT@sakilacustomer.org	597	1	2006-02-14 22:04:37	2006-02-15 04:57:20
592	1	TERRANCE	ROUSH	TERRANCE.ROUSH@sakilacustomer....	598	0	2006-02-14 22:04:37	2006-02-15 04:57:20
593	2	RENE	MCALISTER	RENE.MCALISTER@sakilacustomer.org	599	1	2006-02-14 22:04:37	2006-02-15 04:57:20
594	1	EDUARDO	HIATT	EDUARDO.HIATT@sakilacustomer.org	600	1	2006-02-14 22:04:37	2006-02-15 04:57:20
595	1	TERRENCE	GUNDER...	TERRENCE.GUNDERSON@sakilacust...	601	1	2006-02-14 22:04:37	2006-02-15 04:57:20
596	1	ENRIQUE	FORSYTHE	ENRIQUE.FORSYTHE@sakilacustome...	602	1	2006-02-14 22:04:37	2006-02-15 04:57:20
597	1	FREDDIE	DUGGAN	FREDDIE.DUGGAN@sakilacustomer.org	603	1	2006-02-14 22:04:37	2006-02-15 04:57:20
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
600	1	Allen	Iverson	Allen.Iverson@sakilacustomer.org	1	1	2022-11-29 10:33:20	2022-11-29 10:46:56

output 窗口：

Output			
Action Output			
#	Time	Action	Message
1	10:46:56	update customer set last_update = now...	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0

13、 删除第 11 步新增的那条数据。

sql 语句:

```
delete from customer
```

```
where customer_id = 600;
```

执行结果:

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
590	2	SETH	HANNON	SETH.HANNON@sakilacustomer.org	596	1	2006-02-14 22:04:37	2006-02-15 04:57:20
591	1	KENT	ARSENAU...	KENT.ARSENAULT@sakilacustomer.org	597	1	2006-02-14 22:04:37	2006-02-15 04:57:20
592	1	TERRANCE	ROUSH	TERRANCE.ROUSH@sakilacustomer....	598	0	2006-02-14 22:04:37	2006-02-15 04:57:20
593	2	RENE	MCALISTER	RENE.MCALISTER@sakilacustomer.org	599	1	2006-02-14 22:04:37	2006-02-15 04:57:20
594	1	EDUARDO	HIATT	EDUARDO.HIATT@sakilacustomer.org	600	1	2006-02-14 22:04:37	2006-02-15 04:57:20
595	1	TERRENCE	GUNDER...	TERRENCE.GUNDERSON@sakilacust...	601	1	2006-02-14 22:04:37	2006-02-15 04:57:20
596	1	ENRIQUE	FORSYTHE	ENRIQUE.FORSYTHE@sakilacustome...	602	1	2006-02-14 22:04:37	2006-02-15 04:57:20
597	1	FREDDIE	DUGGAN	FREDDIE.DUGGAN@sakilacustomer.org	603	1	2006-02-14 22:04:37	2006-02-15 04:57:20
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

output 窗口:

#	Time	Action	Message
1	10:53:41	delete from customer where customer_i...	1 row(s) affected
2	10:53:45	SELECT * FROM sakila.customer LIMIT ...	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。

#	Time	Action	Message
1	11:30:45	INSERT INTO actor VALUES (1, 'Allen', '...	Error Code: 1062. Duplicate entry '1' for key 'actor.PRIMARY'

原因: actor_id 为 actor 表的主码, 其值唯一标识一个元组, 不能有重复。

- 2) insert 语句还用了一个函数 NOW(), 是做什么的呢?

答: NOW()用于返回当前的日期和时间。