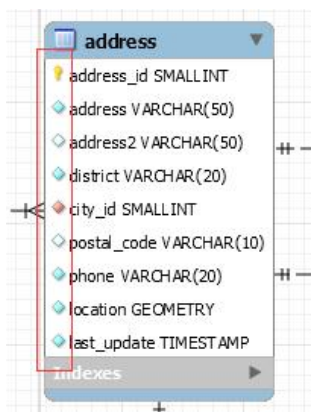


实验一报告

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
	表示该字段是主键
	表示该字段不能为空
	表示该字段可以为空
	表示该字段不能为空且为外键

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？

film_actor

film_actor 表、actor 表和 film 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？

customer 表、rental 表、inventory 表、film 表

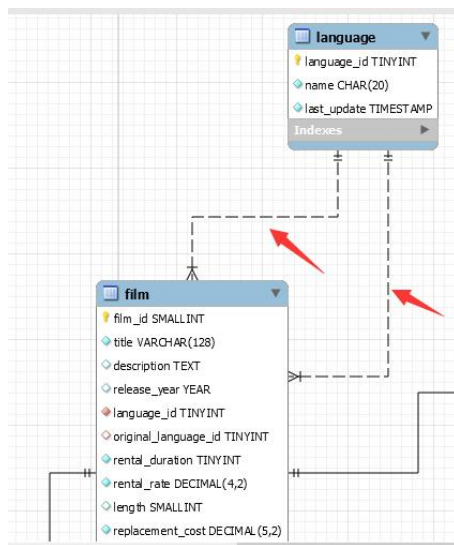
4. film 和 language 表间的 2 条虚线表示什么意思？

虚线表示 non-identify relationship 表示的是子表格和母表格之间没有很强的联系。子表格里的每一行可能和母表格里的某一行相关。

即 language_id 作为 language 表的主键是 film 表的外键

1：n 表示一个 language_language_id 可以对应多个 film_language_id 和

film_original_id



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；
use sakila;

```
select store_id,store.address_id,city.city_id,country.country_id 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;
```

```
1 • use sakila;
2 • select store_id,store.address_id,city.city_id,country.country_id from store,address,city,country
3 where store.address_id = address.address_id and address.city_id = city.city_id and city.country_id = country.country_id;
```

store_id	address_id	city_id	country_id
1	1	300	20
2	2	576	8

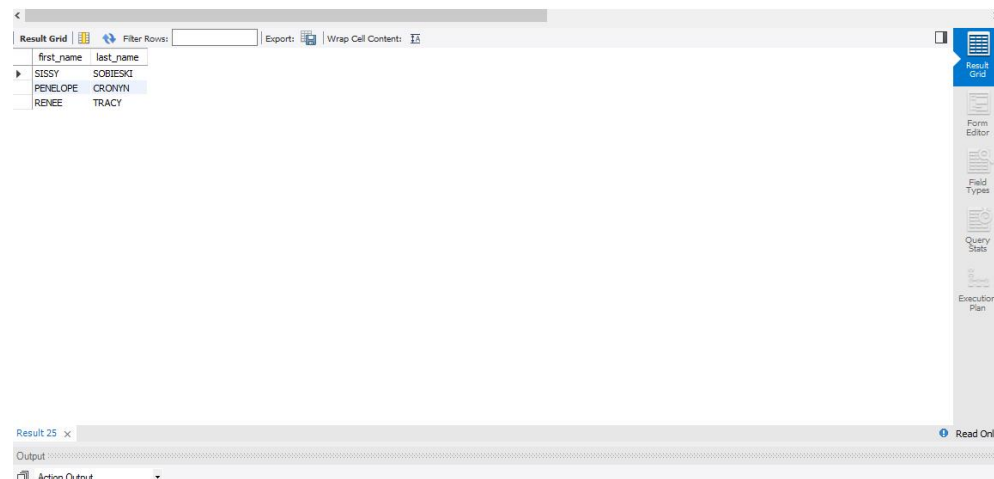
#	Time	Action	Message
180	16:08:43	select store_id,store.address_id,city.city_id,country.country_id from store where store.address_id = address.address_id and address.city_id = city.city_id...	Error Code: 1054. Unknown column 'city.city_id' in 'field list'
181	16:08:45	use sakila	0 row(s) affected
182	16:08:45	select store_id,store.address_id,city.city_id,country.country_id from store where store.address_id = address.address_id and address.city_id = city.city_id...	Error Code: 1054. Unknown column 'city.city_id' in 'field list'
183	16:08:47	use sakila	0 row(s) affected
184	16:08:47	select store_id,store.address_id,city.city_id,country.country_id from store,address,city,country where store.address_id = address.address_id and 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;

use sakila;

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

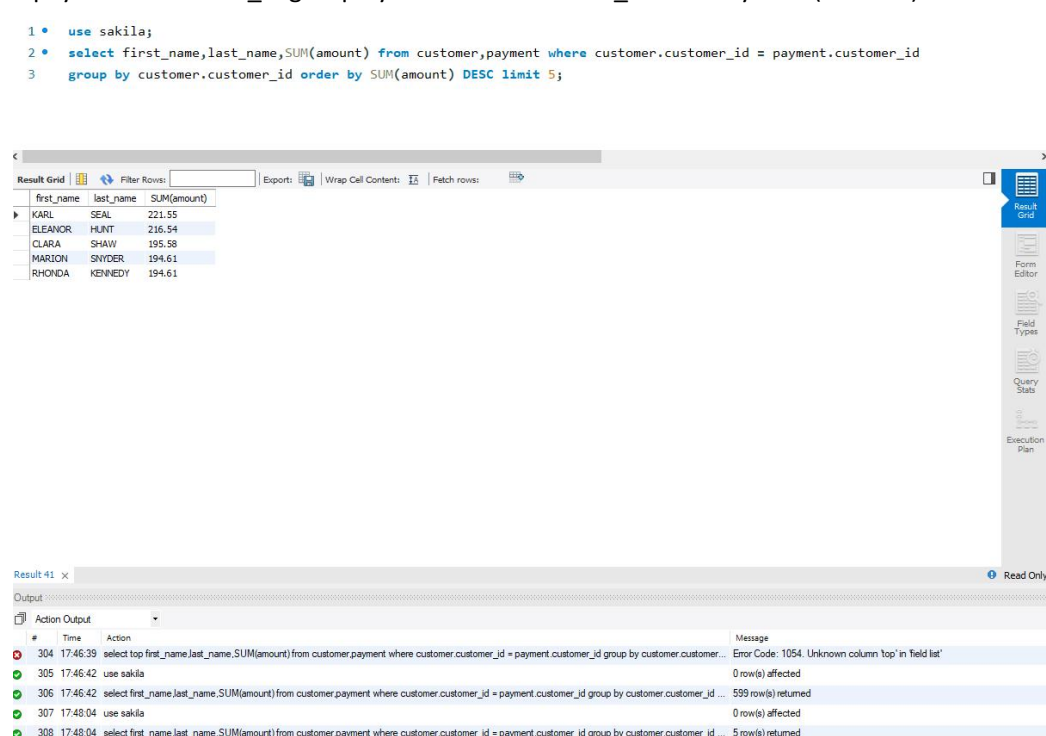
```
1 • use sakila;
2 • select first_name,last_name from actor,film,film_actor where film.film_id = film_actor.film_id and
3   actor.actor_id = film_actor.actor_id and film.title = 'ROCKY WAR';
```



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；

use sakila;

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 5;



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；

use sakila;

```
select film.film_id,film.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;
```

```
1 • use sakila;
2 • select film.film_id,film.title,SUM(amount) from payment,rental,inventory,film where payment.rental_id = rental.rental_id
3 and rental.inventory_id = inventory.inventory_id and inventory.film_id = film.film_id
4 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 49 x

Output

Action Output

#	Time	Action	Message
324	18:16:51	select film.film_id,title,SUM(amount) from payment,rental,inventory,film where payment.rental_id = rental.rental_id and rental.inventory_id = inventory.i...	958 row(s) returned
325	18:16:55	use sakila	0 row(s) affected
326	18:16:55	select top film.film_id,film.title,SUM(amount) from payment,rental,inventory,film where payment.rental_id = rental.rental_id and rental.inventory_id = invent...	Error Code: 1064. You have an error in your SQL syntax; check the
327	18:17:03	use sakila	0 row(s) affected
328	18:17:03	select film.film_id,film.title,SUM(amount) from payment,rental,inventory,film where payment.rental_id = rental.rental_id and rental.inventory_id = inventory.i...	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；

use sakila;

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

Result Grid

actor_id	first_name	last_name
23	SANDRA	KILMER
81	SCARLETT	DAMON
102	WALTER	TORN
107	GINA	DEGENERES
181	MATTHEW	CARREY
198	MARY	KETTEL

Result 51 x

Output

Action Output

#	Time	Action	Message
332	18:22:14	select actor.actor_id,first_name,last_name from film,film_actor,actor where film.film_id = film_actor.film_id and actor.actor_id = film_actor.actor_id and co...	Error Code: 1111. Invalid use of group function
333	18:22:28	use sakila	0 row(s) affected
334	18:22:28	select actor.actor_id,first_name,last_name from film,film_actor,actor where film.film_id = film_actor.film_id and actor.actor_id = film_actor.actor_id group b...	200 row(s) returned
335	18:23:07	use sakila	0 row(s) affected
336	18:23:07	select actor.actor_id,first_name,last_name from film,film_actor,actor where film.film_id = film_actor.film_id and actor.actor_id = film_actor.actor_id group b...	6 row(s) returned

6、 请找出没有租借过电影《TELEGRAPH VOYAGE》的顾客姓名；

use sakila;

```
select first_name,last_name from customer where not exists(select * from 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 = "TELEGRAPH VOYAGE");
```

The screenshot shows a database client interface with a SQL editor at the top and a results grid below. The SQL query is as follows:

```
1 • use sakila;
2 • select first_name,last_name from customer where not exists(select * from film,inventory,rental
3   where film.film_id = inventory.film_id and inventory.inventory_id = rental.inventory_id
4   and rental.customer_id = customer.customer_id and film.title = "TELEGRAPH VOYAGE");
```

The results grid displays a list of customer names. The first column is 'first_name' and the second column is 'last_name'. The results are:

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
CAROL	GARCIA
RUTH	MARTINEZ
SHARON	ROBINSON
MICHELLE	CLARK
LAURA	RODRIGUEZ
SARAH	LEWIS
KIMBERLY	LEE
DEBORAH	WALKER
JESSICA	HALL

The output pane at the bottom shows the execution log with the following messages:

```
# Time Action Message
350 18:36:49 select first_name,last_name from customer where not exists(select * from 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 = "TELEGRAPH VOYAGE"); 572 row(s) returned
351 18:37:32 use sakila 0 row(s) affected
352 18:37:32 select first_name,last_name from customer where not exists(select * from 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 = "TELEGRAPH VOYAGE"); 572 row(s) returned
353 18:37:41 use sakila 0 row(s) affected
354 18:37:41 select first_name,last_name from customer where not exists(select * from 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 = "TELEGRAPH VOYAGE"); 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；

use sakila;

```
select first_name,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 = "ELEPHANT TROJAN" or film.title =
"SPLASH GUMP") group by actor.actor_id having count(*) = 2;
```

The screenshot shows a database client interface with a SQL editor at the top and a results grid below. The SQL query is as follows:

```
1 • use sakila;
2 • select first_name,last_name from film,film_actor,actor where film.film_id = film_actor.film_id and actor.actor_id = film_actor.actor_id
3   and (film.title = "ELEPHANT TROJAN" or film.title = "SPLASH GUMP") group by actor.actor_id having count(*) = 2;
```

The results grid displays a list of actor names. The first column is 'first_name' and the second column is 'last_name'. The results are:

first_name	last_name
PELOLOPE	GUINNESS
CAMERON	STREEP

The output pane at the bottom shows the execution log with the following messages:

```
# Time Action Message
382 18:56:24 select a1.first_name,a1.last_name from film as f1,film_actor as fa1,actor as a1 where f1.film_id = fa1.film_id and a1.actor_id = fa1.actor_id and f1.title = "ELEPHANT TROJAN" or f1.title = "SPLASH GUMP" group by a1.actor_id having count(*) = 2; Error Code: 1241. Operand should contain 1 column(s)
383 19:03:23 use sakila 0 row(s) affected
384 19:03:23 select a1.first_name,a1.last_name from film as f1,film_actor as fa1,actor as a1 where f1.film_id = fa1.film_id and a1.actor_id = fa1.actor_id and f1.title = "ELEPHANT TROJAN" or f1.title = "SPLASH GUMP" group by a1.actor_id having count(*) = 2; 0 row(s) returned
385 19:04:03 use sakila 0 row(s) affected
386 19:04:03 select a1.first_name,a1.last_name from film as f1,film_actor as fa1,actor as a1 where f1.film_id = fa1.film_id and a1.actor_id = fa1.actor_id and f1.title = "ELEPHANT TROJAN" or f1.title = "SPLASH GUMP" group by a1.actor_id having count(*) = 2; 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；

use sakila;

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

The screenshot shows a database query tool interface. The top panel displays the SQL query: `1 • use sakila;`, `2 • select category.category_id,category.name,count(*) from film,film_category,category where film.film_id = film_category.film_id`, and `3 and category.category_id = film_category.category_id group by category.category_id;`. The middle panel shows the 'Result Grid' with 16 rows of data. The bottom panel shows the 'Action Output' log with 6 steps, including the execution of the query and the return of 16 rows.

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	Gaines	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？

use sakila;

```
select film.film_id,film.title from film,inventory,store where film.film_id = inventory.film_id and
store.store_id = inventory.store_id group by film.film_id having count(*) = 2;
```

The screenshot shows a database query tool interface. The top panel displays the SQL query: `1 • use sakila;`, `2 • select film.film_id,film.title from film,inventory,store where film.film_id = inventory.film_id and store.store_id = inventory.store_id`, and `3 group by film.film_id having count(*) = 2;`. The middle panel shows the 'Result Grid' with 63 rows of data. The bottom panel shows the 'Action Output' log with 5 steps, including the execution of the query and the return of 133 rows.

film_id	title
29	ANITRUST TOMATOES
30	ANYTHING SAVANNAH
53	BANG KIWAI
82	BLOOD ARGONAUTS
94	BRAVEHEART HUMAN
106	BULWORTH COMMANDMENTS
161	CLUE GRAIL
168	COMANCHEROS ENEMY
178	CONNECTION MICROCOSMOS
182	CONTROL ANTHEM
185	COWBOY DOOM
196	CRUELTY UNFORGIVEN
197	CRUSADE HONEY
211	DARLING BREAKING
224	DESPERATE TRAINSPOTTING
230	DIARY PANIC
237	DIVORCE SHINING
238	DOCTOR GRAIL
259	DUCK RACER
261	DUFFEL APOCALYPSE
278	ELF MURDER
279	ELIZABETH SHANE
289	EVE RESURRECTION
310	FEVER EMPIRE
114	REDEMPTION IN FRODO BAGGINS

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；

use sakila;

```
select customer.customer_id,(return_date - rental_date) from customer,rental where
rental.customer_id = customer.customer_id group by rental.rental_id order by (return_date -
rental_date) DESC limit 6;
```

1 • use sakila;

2 • select customer.customer_id,(return_date - rental_date) from customer,rental where rental.customer_id = customer.customer_id

3 group by rental.rental_id order by (return_date - rental_date) DESC limit 6;

customer_id	(return_date - rental_date)
211	798.18900
13	788.19100
196	788.18300
327	788.17200
307	788.16900
187	788.16900

Result 65 x

Output

Action Output

#	Time	Action	Message
408	19:19:58	select customer.customer_id from customer,rental where rental.customer_id = customer.customer_id group by customer.customer_id order by (return_da...	Error Code: 1055. Expression #1 of ORDER BY clause is not in GROUP BY clause and contains nonaggregated col...
409	19:20:53	use sakila	0 row(s) affected
410	19:20:53	select customer.customer_id from customer,rental where rental.customer_id = customer.customer_id group by rental.rental_id order by (return_date - ren...	1000 row(s) returned
411	19:21:08	use sakila	0 row(s) affected
412	19:21:08	select customer.customer_id,(return_date - rental_date) from customer,rental where rental.customer_id = customer.customer_id group by rental.rental_id...	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；

use sakila;

insert into customer

values (600,2,"haoyuan","deng","1595466987@qq.com",599,1,NOW(),null);

其中，store_id 和 address_id 是外键，需要依赖于 store 表已有的 store_id 和 address 表已有的 address_id，customer_id 是主键，不能和表中其他数据中的 customer_id 冲突

1 • use sakila;

2 • insert into customer

3 values (600,2,"haoyuan","deng","1595466987@qq.com",599,1,NOW(),null);

Output

Action Output

#	Time	Action	Message
425	19:32:15	insert into customer(customer_id,store_id,first_name,last_name,email,address_id,active,create_date,last_update) values (600,2,'haoyuan','deng','15...	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('sakila','customer', CONSTRAINT
426	19:34:11	use sakila	0 row(s) affected
427	19:34:11	insert into customer(customer_id,store_id,first_name,last_name,email,address_id,active,create_date,last_update) values (600,2,'haoyuan','deng','159...	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('sakila','customer', CONSTRAINT
428	19:35:33	use sakila	0 row(s) affected
429	19:35:33	insert into customer values (600,2,'haoyuan','deng','1595466987@qq.com',599,1,NOW(),null)	1 row(s) affected

12、 修改刚才在 customer 表中新增的那条数据；

use sakila;

update customer set store_id = 1 where customer_id = 600 and first_name = "haoyuan" and last_name = "deng";

```
1 • use sakila;
2 • update customer set store_id = 1 where customer_id = 600 and first_name = "haoyuan" and last_name = "deng";
```

#	Time	Action	Message
427	19:34:11	insert into customer(customer_id,store_id,first_name,last_name,email,address_id,active,create_date,last_update) values (600,2,"haoyuan","deng","159...	Error Code: 1452. Cannot add or update a child row: a foreign key constraint fails ('sakila`.`customer`, CONSTRAINT...
428	19:35:33	use sakila	0 row(s) affected
429	19:35:33	insert into customer values (600,2,"haoyuan","deng","1595466387@qq.com",559,1,NOW(),null)	1 row(s) affected
430	19:41:49	use sakila	0 row(s) affected
431	19:41:49	update customer set store_id = 1 where customer_id = 600 and first_name = "haoyuan" and last_name = "deng"	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0

13、 删除第 11 步新增的那条数据。

use sakila;

delete from customer where customer_id = 600 and first_name = "haoyuan" and last_name = "deng";

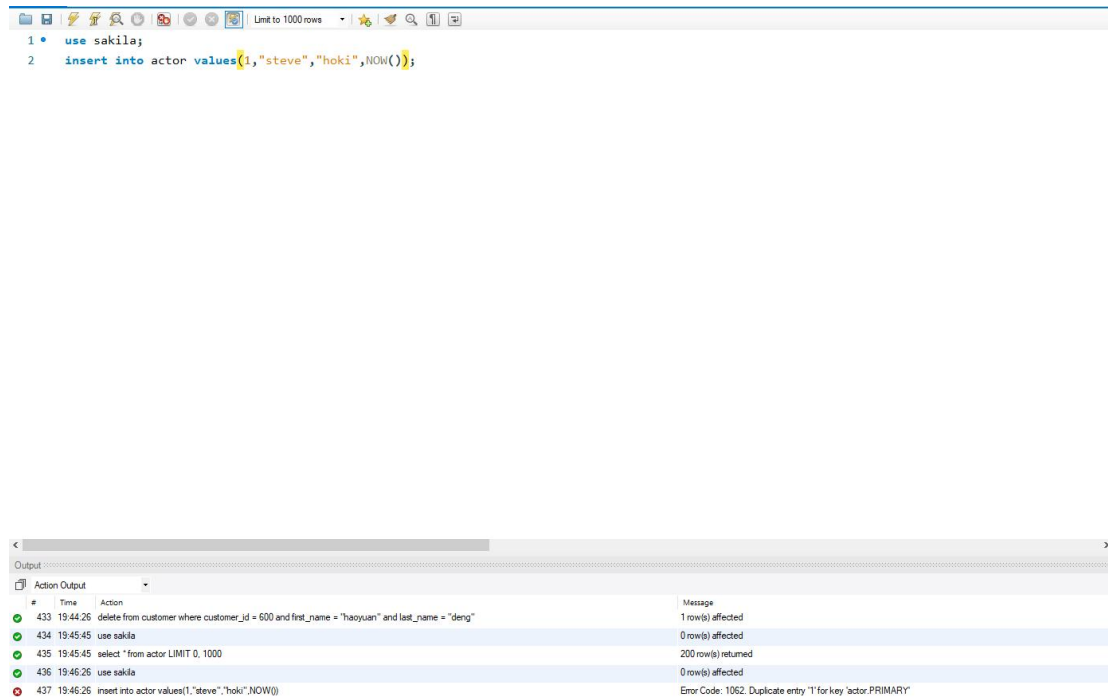
```
1 • use sakila;
2 • delete from customer where customer_id = 600 and first_name = "haoyuan" and last_name = "deng";
```

#	Time	Action	Message
429	19:35:33	insert into customer values (600,2,"haoyuan","deng","1595466387@qq.com",559,1,NOW(),null)	1 row(s) affected
430	19:41:49	use sakila	0 row(s) affected
431	19:41:49	update customer set store_id = 1 where customer_id = 600 and first_name = "haoyuan" and last_name = "deng"	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0
432	19:44:26	use sakila	0 row(s) affected
433	19:44:26	delete from customer where customer_id = 600 and first_name = "haoyuan" and last_name = "deng"	1 row(s) affected

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。

会产生代码为 1062 的错误，表示插入的数据主键和已有数据的主键重复，因为主键是用于区分每条数据和作为每条数据唯一性的判断依据，因此主键不能重复，主键重复的数据插入将会给数据库带来严重的后果，因此不允许插入



```
1 • use sakila;
2 insert into actor values(1,"steve","hoki",NOW());
```

Output

#	Time	Action	Message
433	19:44:26	delete from customer where customer_id = 600 and first_name = "haoyuan" and last_name = "deng"	1 row(s) affected
434	19:45:45	use sakila	0 row(s) affected
435	19:45:45	select * from actor LIMIT 0, 1000	200 row(s) returned
436	19:46:26	use sakila	0 row(s) affected
437	19:46:26	insert into actor values(1,"steve","hoki",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？

函数 NOW() 是 DATETIME 或 TIMESTAMP 列的默认值。当 INSERT 语句中省略日期或时间值时，MySQL 会将当前日期和时间插入到默认值为 NOW () 的列中。