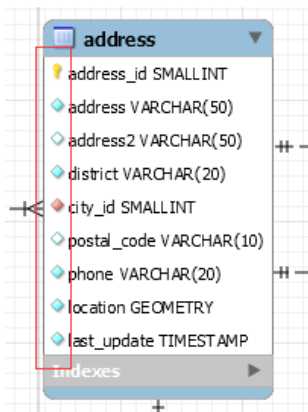


实验一报告

一、回答问题

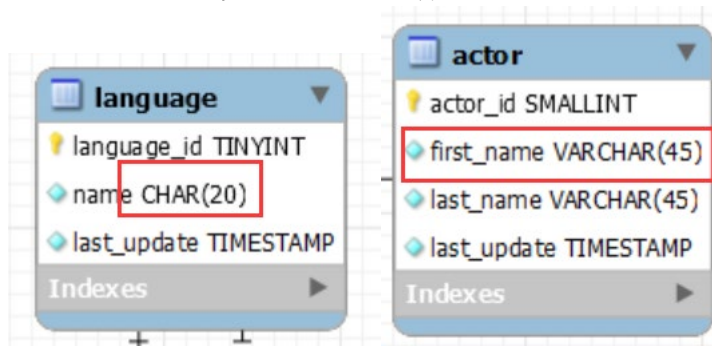
请一边熟悉 sakila 数据库，一边回答以下问题：

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
	主码(Primary key)
	非空(not null)
	可以为空
	外码且非空(Foreign key & not null)

2. char 和 varchar 类型的区别是什么？



char 是长度不可变的字符，空间不足的用空格填充

varchar 是长度可变的字符，通常用一个字节来记录长度

3. 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 共 3 张表，过程如下：

- (1) 在 **film** 表中查询 **film_id**;
 - (2) 再在 **film_actor** 表中通过 **film_id** 查询 **actor_id**;
 - (3) 最后在 **actor** 表中通过 **actor_id** 查询演员姓/名(**first_name** & **last_name**)。
4. 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, 共 4 张表，过程如下：

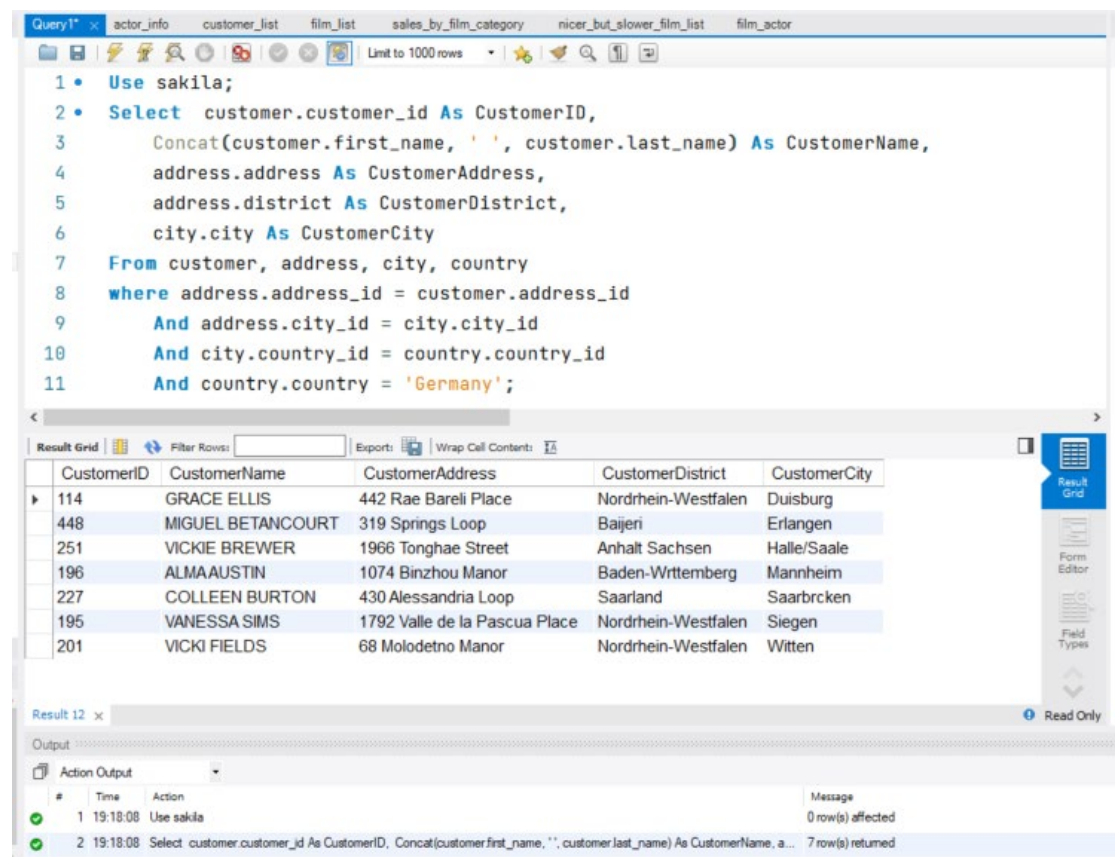
- (1) 在 **customer** 表中通过顾客姓名查询 **customer_id**;
- (2) 在 **rental** 表中通过 **customer_id** 查询 **inventory_id**;
- (3) 再在 **inventory** 表中通过 **inventory_id** 查询 **film_id**;

最后在 **film** 表中通过 **film_id** 查询影片名(title)。

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 country 是 “Germany” 的客户的信息，显示 customer_id、客户姓名、地址、所在区域，所在城市（注意：客户姓名请以 first_name+空格+last_name 的格式，例如：SISSY SOBIESKI）；



```
1 • Use sakila;
2 • Select customer.customer_id As CustomerID,
3       Concat(customer.first_name, ' ', customer.last_name) As CustomerName,
4       address.address As CustomerAddress,
5       address.district As CustomerDistrict,
6       city.city As CustomerCity
7 From customer, address, city, country
8 where address.address_id = customer.address_id
9       And address.city_id = city.city_id
10      And city.country_id = country.country_id
11      And country.country = 'Germany';
```

CustomerID	CustomerName	CustomerAddress	CustomerDistrict	CustomerCity
114	GRACE ELLIS	442 Rae Bareli Place	Nordrhein-Westfalen	Duisburg
448	MIGUEL BETANCOURT	319 Springs Loop	Bajeri	Erlangen
251	VICKIE BREWER	1966 Tonghae Street	Anhalt Sachsen	Halle/Saale
196	ALMAAUSTIN	1074 Binzhou Manor	Baden-Wrttemberg	Mannheim
227	COLLEEN BURTON	430 Alessandria Loop	Saarland	Saarbrcken
195	VANESSA SIMS	1792 Valle de la Pascua Place	Nordrhein-Westfalen	Siegen
201	VICKI FIELDS	68 Molodetno Manor	Nordrhein-Westfalen	Witten

Result 12 x

Output

#	Time	Action	Message
1	19:18:08	Use sakila	0 row(s) affected
2	19:18:08	Select customer.customer_id As CustomerID, Concat(customerfirst_name, ' ', customerlast_name) As CustomerName, a...	7 row(s) returned

2、 列出属于“Music”类型并以“A”开头的电影名；

The screenshot shows a SQL query in the 'Query1*' window. The query is as follows:

```

1 • Use sakila;
2 • Select film.title
3   From film, film_category, category
4   Where film.title Like 'A%'
5         AND film.film_id = film_category.film_id
6         AND film_category.category_id = category.category_id
7         AND category.name = 'Music';

```

The 'Result Grid' shows the following results:

title
ALASKA PHANTOM
ALONE TRIP
AMELIE HELLFIGHTERS

The 'Output' pane shows the execution log:

#	Time	Action	Message
1	19:22:21	Use sakila	0 row(s) affected
2	19:22:21	Select film.title From film, film_category, category Where film.title Like 'A%' AND film.film_id = film_category.film_id AND fi...	3 row(s) returned

3、 找出租 DVD 花费的总费用在 160 至 170 之间的客户，列出他们的 first_name, last_name 和每个人花费的金额；

The screenshot shows a SQL query in the 'Query1*' window. The query is as follows:

```

1 • Use sakila;
2 • Select first_name, last_name, sum(payment.amount)
3   From customer, payment
4   Where customer.customer_id = payment.customer_id
5   group by payment.customer_id
6   Having sum(payment.amount) > 160
7         And sum(payment.amount) <170

```

The 'Result Grid' shows the following results:

first_name	last_name	sum(payment.amount)
DIANE	COLLINS	169.65
TONYA	CHAPMAN	161.68
DAISY	BATES	162.62
LOUIS	LEONE	161.65
MIKE	WAY	166.65
CURTIS	IRBY	167.62
GORDON	ALLARD	160.68
ARNOLD	HAVENS	167.67

The 'Output' pane shows the execution log:

#	Time	Action	Message
1	19:25:12	Use sakila	0 row(s) affected
2	19:25:12	Select first_name, last_name, sum(payment.amount) From customer, payment Where customer.customer_id = payment.cu...	8 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；

Query1* x actor_info customer_list film_list sales_by_film_category nicer_but_slower_film_list film_actor

Limit to 1000 rows

```

1 • Use sakila;
2 • Select film.film_id, film.title, sum(payment.amount)
3   From film, inventory, rental, payment
4   Where film.film_id = inventory.film_id
5         And inventory.inventory_id = rental.inventory_id
6         And rental.rental_id = payment.rental_id
7   Group By film.film_id
8   Order By sum(payment.amount) Desc
9   Limit 1

```

Result Grid

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

Result 16 x

Output

Action Output

#	Time	Action	Message
1	19:31:40	Use sakila	0 row(s) affected
2	19:31:40	Select film.film_id, film.title, sum(payment.amount) From film, inventory, rental, payment Where film.film_id = inventory.film_id...	1 row(s) returned

5、 哪些演员出演的电影超过 40 部？ 请列出演员名、出演的电影数；

Query1* x actor_info customer_list film_list sales_by_film_category nicer_but_slower_film_list film_actor

Limit to 1000 rows

```

1 • Use sakila;
2 • Select Concat(actor.first_name, ' ', actor.last_name) As ActorName,
3         Count(*) As FilmPerformCount
4   From film, film_actor, actor
5   Where film.film_id = film_actor.film_id
6         And film_actor.actor_id = actor.actor_id
7   Group by film_actor.actor_id
8   Having Count(*) > 40

```

Result Grid

ActorName	FilmPerformCount
GINA DEGENERES	42
WALTER TORN	41

Result 10 x

Output

Action Output

#	Time	Action	Message
1	19:10:49	Use sakila	0 row(s) affected
2	19:10:49	Select Concat(actor.first_name, ' ', actor.last_name) As ActorName, Count(*) As FilmPerformCount From film, film_actor, ac...	2 row(s) returned

6、 请找出没有租借过电影《NATURAL STOCK》的顾客姓名；

Query1* x actor_info customer_list film_list sales_by_film_category nicer_but_slower_film_list film_actor inventory rental

Limit to 1000 rows

```

1 • Use sakila;
2 • Select Concat(customer.first_name, ' ', customer.last_name)
3 From customer
4 Where customer.customer_id Not In (
5     Select customer.customer_id
6     From film, inventory, rental, customer
7     Where film.title = 'NATURAL STOCK'
8         And film.film_id = inventory.film_id
9         And inventory.inventory_id = rental.inventory_id
10        And rental.customer_id = customer.customer_id
11 )

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

Concat(customer.first_name, ' ', customer.last_name)
MARY SMITH
PATRICIA JOHNSON
LINDA WILLIAMS
BARBARA JONES
ELIZABETH BROWN
JENNIFER DAVIS
SUSAN WILSON
MARGARET MOORE
DOROTHY TAYLOR
LISA ANDERSON
NANCY THOMAS
KAREN JACKSON
BETTY WHITE

Result 23 x

Output

Action Output

#	Time	Action	Message
1	19:40:26	Use sakila	0 row(s) affected
2	19:40:26	Select Concat(customer.first_name, ' ', customer.last_name) From customer Where customer.customer_id Not In (Select ...	586 row(s) returned

7、 查询既演过《ELEPHANT TROJAN》又演过《DOGMA FAMILY》的演员，列出其姓名；

Query1* x film_list sales_by_film_category nicer_but_slower_film_list film_actor inventory rental actor film

Limit to 1000 rows

```

1 • Use sakila;
2 • Select Concat(actor.first_name, ' ', actor.last_name) As ActorName
3 From film, film_actor, actor
4 Where film.title = 'DOGMA FAMILY'
5     And film.film_id = film_actor.film_id
6     And film_actor.actor_id = actor.actor_id
7     AND actor.actor_id In (
8         Select actor.actor_id
9         From film, film_actor, actor
10        Where film.title = 'ELEPHANT TROJAN'
11            And film.film_id = film_actor.film_id
12            And film_actor.actor_id = actor.actor_id
13 )

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

ActorName
GINA DEGENERES

Result 28 x

Output

Action Output

#	Time	Action	Message
1	19:54:33	Use sakila	0 row(s) affected
2	19:54:33	Select Concat(actor.first_name, ' ', actor.last_name) As ActorName From film, film_actor, actor Where film.title = 'DOGMA F...	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；

The screenshot shows a database query tool interface. The top pane contains a SQL query:

```
1 • Use sakila;
2 • Select category.category_id, category.name, Count(*)
3 From film, film_category, category
4 Where film.film_id = film_category.film_id
5 And film_category.category_id = category.category_id
6 Group By category.category_id
```

The bottom pane displays the results in a table:

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

Below the table, the 'Action Output' pane shows the execution log:

#	Time	Action	Message
1	20:00:19	Use sakila	0 row(s) affected
2	20:00:19	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	16 row(s) returned

9、有哪些影片是 2 个商店都有库存的？显示影片名。

Query1* x film film_category category inventory

```

1 • Use sakila;
2 • Select Distinct film.title
3   From film, inventory
4   Where film.film_id = inventory.film_id
5         And inventory.store_id = 1
6         And film.film_id in (
7             Select Distinct film.film_id
8             From film, inventory
9             Where film.film_id = inventory.film_id
10            And inventory.store_id = 2
11        );

```

Result Grid | Filter Rows: | Exports: | Wrap Cell Content: |

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

Result 46 x

Output

Action Output

#	Time	Action	Message
1	20:07:24	Use sakila	0 row(s) affected
2	20:07:24	Select Distinct film.title From film, inventory Where film.film_id = inventory.film_id And inventory.store_id = 1 And film.film_id in (Select Distinct film.film_id From film, inventory Where film.film_id = inventory.film_id And inventory.store_id = 2)	563 row(s) returned

10、 查询单次租借影片时间最长的 2 位客户，列出其 first_name、last_name 和当次租借时长（单位秒）；

Query1* x film film_category category inventory rental

```

1 • Use sakila;
2 • Select customer.first_name, customer.last_name,
3       (rental.return_date - rental.rental_date) As RentalDurance
4   From customer, rental
5   Where customer.customer_id = rental.customer_id
6   Order By RentalDurance Desc
7   Limit 2

```

Result Grid | Filter Rows: | Exports: | Wrap Cell Content: | Fetch rows: |

first_name	last_name	RentalDurance
STACEY	MONTGOMERY	79818900
KAREN	JACKSON	78819100

Result 55 x

Output

Action Output

#	Time	Action	Message
1	20:17:16	Use sakila	0 row(s) affected
2	20:17:16	Select customer.first_name, customer.last_name, (rental.return_date - rental.rental_date) As RentalDurance From customer, rental Where customer.customer_id = rental.customer_id Order By RentalDurance Desc Limit 2	2 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；

The screenshot shows a database management tool interface. The top pane contains the following SQL script:

```

1 • Use sakila;
2 • Insert Into customer
3   Values(600, 1, 'Flandre', 'Scarlet',
4         'FlandreScarlet@sakilacustomer.org', 1, 1, Now(), Now());
5 • select * from customer
  
```

The bottom pane displays the 'Result Grid' showing the contents of the 'customer' table. The table has columns: customer_id, store_id, first_name, last_name, email, address_id, active, and create_date. The newly inserted row (customer_id 600) is highlighted in blue.

customer_id	store_id	first_name	last_name	email	address_id	active	create_date
593	2	RENE	MCALIST...	RENE.MCALISTER@sakilacustom...	599	1	2006-02-14 22:04:3
594	1	EDUARDO	HIATT	EDUARDO.HIATT@sakilacustome...	600	1	2006-02-14 22:04:3
595	1	TERRENCE	GUNDER...	TERRENCE.GUNDERSON@sakil...	601	1	2006-02-14 22:04:3
596	1	ENRIQUE	FORSYTHE	ENRIQUE.FORSYTHE@sakilacus...	602	1	2006-02-14 22:04:3
597	1	FREDDIE	DUGGAN	FREDDIE.DUGGAN@sakilacusto...	603	1	2006-02-14 22:04:3
598	1	WADE	DELVALLE	WADE.DELVALLE@sakilacustome...	604	1	2006-02-14 22:04:3
599	2	AUSTIN	CINTRON	AUSTIN.CINTRON@sakilacustom...	605	1	2006-02-14 22:04:3
600	1	Flandre	Scarlet	FlandreScarlet@sakilacustomer.org	1	1	2023-09-19 20:29:5

The 'Output' pane shows the execution log:

#	Time	Action	Message
1	20:29:52	Use sakila	0 row(s) affected
2	20:29:52	Insert Into customer Values(600, 1, 'Flandre', 'Scarlet', 'FlandreScarlet@sakilacustomer.org', 1, 1, Now(), Now())	1 row(s) affected
3	20:29:52	select * from customer LIMIT 0, 1000	600 row(s) returned

12、 修改刚才在 customer 表中新增的那条数据；

The screenshot shows the same database management tool interface. The top pane contains the following SQL script:

```

1 • Use sakila;
2 • Update customer
3   Set address_id = 2
4   Where customer_id = 600;
5
6 • Select *
7   From customer
8   Where customer_id = 600;
  
```

The bottom pane displays the 'Result Grid' showing the updated contents of the 'customer' table. The row for customer_id 600 is highlighted in blue, and its address_id has been updated from 1 to 2.

customer_id	store_id	first_name	last_name	email	address_id	active	create_date
600	1	Flandre	Scarlet	FlandreScarlet...	2	1	2023-09-19 20:29:5

The 'Output' pane shows the execution log:

#	Time	Action	Message
1	00:49:15	Use sakila	0 row(s) affected
2	00:49:15	Update customer Set address_id = 2 Where customer_id = 600	1 row(s) affected Rows matched: 1 Changed: 1 Warni
3	00:49:15	Select * From customer Where customer_id = 600	1 row(s) returned

13、删除第 11 步新增的那条数据。

The screenshot shows a SQL Developer window with a query titled 'Query1*' containing the following SQL statements:

```
1 • Use sakila;
2 • Delete From customer
3   Where customer_id = 600;
4
5 • Select *
6   From customer
```

Below the query editor, the 'Result Grid' displays the contents of the 'customer' table. The table has columns: customer_id, store_id, first_name, last_name, email, address_id, active, create_date, and last_update. The data rows are as follows:

customer_id	store_id	first_name	last_name	email	address_id	active	create_date	last_update
595	1	TER...	GUN...	TERRENCE...	601	1	2006-02-14 22:04:37	2006-02-15 04:57:20
596	1	ENRI...	FOR...	ENRIQUE.FO...	602	1	2006-02-14 22:04:37	2006-02-15 04:57:20
597	1	FRE...	DUG...	FREDDIE.DU...	603	1	2006-02-14 22:04:37	2006-02-15 04:57:20
598	1	WADE	DEL...	WADE.DELV...	604	1	2006-02-14 22:04:37	2006-02-15 04:57:20
599	2	AUS...	CINT...	AUSTIN.CINT...	605	1	2006-02-14 22:04:37	2006-02-15 04:57:20

Below the result grid, the 'Output' window shows the 'Action Output' for the executed queries:

#	Time	Action	Message
1	00:55:37	Use sakila	0 row(s) affected
2	00:55:37	Delete From customer Where customer_id = 600	0 row(s) affected
3	00:55:37	Select * From 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，截图并分析原因。

会报错，因为 actor_id 是主键，不能有两个元组的主键相同。

The screenshot shows a SQL Developer window with a query titled 'Query1*' containing the following SQL statements:

```
1 • Use sakila;
2 • Insert Into actor
3   Values (200, 'Flandre', 'Scarlet', Now())
```

Below the query editor, the 'Output' window shows the 'Action Output' for the executed queries:

#	Time	Action	Message
1	00:59:50	Use sakila	0 row(s) affected
2	00:59:50	Insert Into actor Values (200, 'Flandre', 'Scarlet', Now())	Error Code: 1062. Duplicate entry '200' for key 'actor.PRIMARY'

- 2) insert 语句还用了一个函数 NOW(), 是做什么的呢？

这个函数返回一个时间对象，用于标记此条记录增加/修改的时间