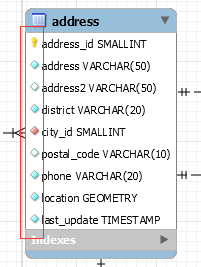
实验一报告

# 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 |
|  | 该字段为该表的主键。 |
|  | 该字段不允许为空值，且不是主键和外键。 |
|  | 该字段允许为空值。 |
|  | 该字段为该表的外键，且不允许空值。 |

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？

访问 film、actor 和 film\_actor 即可获得所需信息。总体分为三个步骤：

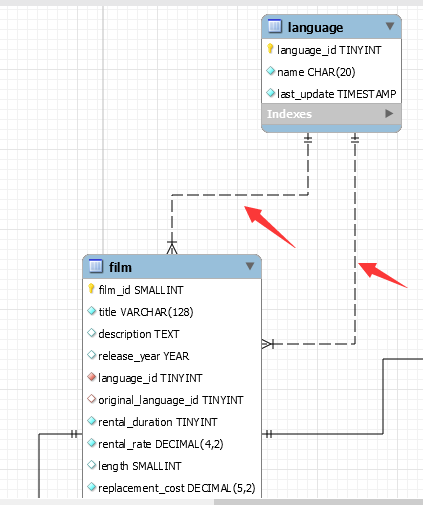
1. 在 film 表中通过影片名 title 获得影片编号 film\_id.
2. 在 film\_actor 表中通过影片编号 film\_id 获得出演该影片的所有演员编号 actor\_id.
3. 在 actor 表中通过演员编号 actor\_id 获得演员名字 first\_name、last\_name.

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？

访问 customer、rental、inventory 和 film 即可获得所需信息。总体分为四个步骤：

1. 在 customer 表中通过顾客姓名 first\_name 和 last\_name 获得顾客编号 customer\_id.
2. 在 rental 表中通过顾客编号 customer\_id 获得该顾客租借的所有 DVD 编号 inventory\_id.
3. 在 inventory 表中通过 DVD 编号 inventory\_id 获得影片编号 film\_id.
4. 在 film 表中通过影片编号 film\_id 获得影片名 title.

4. 为什么film和language表间会有2条线？



表示 film 表中的两个外键 language\_id 与 original\_language\_id 均对应 language 表中的主键。

# 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, country, city

WHERE store.address\_id = address.address\_id

AND address.city\_id = city.city\_id

AND city.country\_id = country.country\_id;

屏幕的截图

描述已自动生成结果截图：

1. 哪些演员出演过影片《WEST LION》？请列出他的姓名；

SQL 语句：

SELECT CONCAT(first\_name, ' ', last\_name) as name

FROM actor, film, film\_actor

WHERE film.title = 'WEST LION'

AND actor.actor\_id = film\_actor.actor\_id

AND film.film\_id = film\_actor.film\_id;

屏幕的截图

描述已自动生成结果截图：

1. 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；

SQL 语句：

SELECT first\_name, last\_name, SUM(amount) as summation

FROM customer, payment

WHERE payment.customer\_id = customer.customer\_id

GROUP BY first\_name, last\_name

ORDER BY summation DESC

LIMIT 3;

截图里有图片

描述已自动生成结果截图：

1. 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) as summation

FROM rental, payment, inventory, film

WHERE film.film\_id = inventory.film\_id

AND payment.rental\_id = rental.rental\_id

AND rental.inventory\_id = inventory.inventory\_id

GROUP BY title, film.film\_id

ORDER BY summation DESC

LIMIT 1;

结果截图：

银色的手机截图

描述已自动生成

1. 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；

SQL 语句：

SELECT actor.actor\_id, CONCAT(first\_name, ' ', last\_name) as name, COUNT(film\_actor.film\_id) as summation

FROM actor, film\_actor

WHERE actor.actor\_id = film\_actor.actor\_id

GROUP BY actor.actor\_id, name

HAVING summation > 40;

结果截图：

截图里有图片

描述已自动生成

1. 请找出没有租借过电影《WEST LION》的顾客姓名；

SQL 语句：

SELECT DISTINCT CONCAT(first\_name, ' ', last\_name) as name

FROM rental, customer

WHERE rental.customer\_id = customer.customer\_id

AND rental.customer\_id NOT IN (

SELECT rental.customer\_id

FROM rental, inventory, film

WHERE film.title = 'WEST LION'

AND film.film\_id = inventory.film\_id

AND inventory.inventory\_id = rental.inventory\_id

);

电脑萤幕截图

描述已自动生成结果截图：

1. 查询演过《FIRE WOLVES》和《JAWBREAKER BROOKLYN》这两部电影的演员，列出其姓名；

SQL 语句：

SELECT CONCAT(first\_name, ' ', last\_name) as name

FROM actor

WHERE actor\_id = (

SELECT FA1.actor\_id

FROM film\_actor FA1, film\_actor FA2

WHERE FA1.actor\_id = FA2.actor\_id

AND FA1.film\_id = (

SELECT film\_id FROM film WHERE title = 'FIRE WOLVES'

)

AND FA2.film\_id = (

SELECT film\_id FROM film WHERE title = 'JAWBREAKER BROOKLYN'

)

);

结果截图：

截图里有图片

描述已自动生成

1. 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(category.category\_id) as amount

FROM film\_category, category

WHERE category.category\_id = film\_category.category\_id

GROUP BY category.category\_id, category.name;

图片包含 图形用户界面

描述已自动生成结果截图：

1. 有哪些影片是2个商店都有库存的？

SQL 语句：

SELECT DISTINCT title

FROM film, inventory I1, inventory I2

WHERE I1.film\_id = I2.film\_id

AND I1.store\_id <> I2.store\_id

AND I1.film\_id = film.film\_id;

电脑萤幕截图

描述已自动生成结果截图：

1. 查询单次租借影片时间最长的3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, TIMESTAMPDIFF(second , rental\_date, return\_date) as rental\_length

FROM customer, rental

WHERE customer.customer\_id = rental.customer\_id

ORDER BY rental\_length DESC

LIMIT 3;

屏幕的截图

描述已自动生成结果截图：

1. 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 sakila.customer (store\_id, first\_name, last\_name, email, address\_id, create\_date)

VALUES (1, 'PINSKY', 'ROBIN', 'xu568059888@gmail.com', 6, NOW());

图形用户界面, 文本, 应用程序

描述已自动生成结果截图：

电脑屏幕的截图

描述已自动生成customer表新增数据：

1. 修改刚才在customer表中新增的那条数据；

SQL 语句：

UPDATE customer

SET active = 0

WHERE customer\_id = 609;

图形用户界面, 应用程序

描述已自动生成结果截图：

截图里有图片

描述已自动生成图形用户界面, 文本, 应用程序

描述已自动生成customer表修改数据：

1. 删除第11步新增的那条数据。

SQL 语句：

UPDATE customer

SET active = 0

WHERE customer\_id = 609;

图形用户界面, 应用程序

描述已自动生成结果截图：

**图形用户界面, 文本

描述已自动生成**customer表删除数据：

# 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。

考虑执行如下语句：

INSERT INTO actor (actor\_id, first\_name, last\_name, last\_update)

VALUES (1, 'PINSKY', 'ROBIN', '2021-09-21 16:10:17')

执行结果如下:

[2021-09-21 16:10:21] [23000][1062] Duplicate entry '1' for key 'actor.PRIMARY'

[2021-09-21 16:10:21] [23000][1062] Duplicate entry '1' for key 'actor.PRIMARY'

表示插入的数据的字段与现有某行数据的字段重复，插入失败。

1. insert语句还用了一个函数NOW()，是做什么的呢？

NOW()函数以“YYYY-MM-DD HH:mm:SS” 的格式返回该条指令执行时的时间。