

第五章 第五次作业

1. 一、数据库介绍

Sakila 是一个由 MySQL 官方提供的示例数据库，主要设计用于电影租赁商店的业务场景。

Sakila 数据库中，每个表的具体字段请参考官方文档中 5.1 Tables 中的解释：
<https://downloads.mysql.com/docs/sakila-en.a4.pdf>

二、数据导入

1. 在 Navicat 中创建一个新的数据库，命名为 sakila。

2. 导入表结构与数据：

方法一：右键新创建好的 sakila 数据库，选择“运行 SQL 文件”：

在文件一栏中，先选择 sakila-schema.sql 文件进行导入，导入成功后关闭“运行 SQL 文件”的界面；

同样的方法，再选择 sakila-data.sql 文件进行导入。

刷新数据库，检查表与表中的数据是否成功导入。

方法二：右键新创建好的 sakila 数据库，选择“命令列界面”。

执行如下两行命令：

```
“source /Downloads/sakila-db/sakila-schema.sql;”
```

```
“source /Downloads/sakila-db/sakila-data.sql;”
```

刷新数据库，检查表与表中的数据是否成功导入。

三、写出下面的查询语句，并保存到“实验五.docx”文件中。要求文件中除了提交 SQL 语句，还需有 SQL 语句对应的执行结果。

2. 请为 'rental' 表中的 'inventory_id' 添加一个外键约束，确保其引用 'inventory' 表中的 'inventory_id'，并且当 'inventory' 表中的记录被删除时，相应的 'rental' 记录也被删除。

```
alter table rental add foreign key (inventory_id) references inventory  
↪ (inventory_id) on delete cascade;
```

417 ms 中有 16,044 行受到影响

3. 在 'customer' 表中添加一个检查约束，确保 'active' 列的值只能是 0 或 1。

```
alter table customer add check ( active in (0, 1) );
```

102 ms 中有 599 行受到影响

4. 创建一个名为 'sakila_user' 的数据库用户，并授予其对 'film' 表的 SELECT 和 UPDATE 权限。

```
create user sakila_user identified by random password;  
grant select, update on film to sakila_user;
```

	user	host	generated password
1	sakila_user	%	""

在 49 ms (execution: 36 ms, fetching: 13 ms) 内检索到从 1 开始的 1 行
在 33 ms 内完成

5. 从 'sakila_user' 用户收回对 'film' 表的 UPDATE 权限。

```
revoke update on film from sakila_user;
```

在 39 ms 内完成

6. 查询每个影片的租赁次数（包括影片名称），并按租赁次数降序排列。

```
select film.title, count(*) as 租赁次数  
from film  
      join inventory using (film_id)  
      join rental using (inventory_id)  
group by film.title  
order by 租赁次数 desc;
```

	title	租赁次数
1	BUCKET BROTHERHOOD	34
2	ROCKETEER MOTHER	33
3	FORWARD TEMPLE	32
4	GRIT CLOCKWORK	32
5	JUGGLER HARDLY	32
...
954	SEVEN SWARM	5
955	TRAFFIC HOBBIT	5
956	HARDLY ROBBERS	4
957	MIXED DOORS	4
958	TRAIN BUNCH	4

7. 查询每个影片类别的租赁次数，并找出租赁次数最多的类别。

```
select category.name, count(*) as 租赁次数
from category
    join film_category using (category_id)
    join inventory using (film_id)
    join rental using (inventory_id)
group by category.name
order by 租赁次数 desc;
```

	name	租赁次数
1	Sports	1179
2	Animation	1166
3	Action	1112
4	Sci-Fi	1101
5	Family	1096
6	Drama	1060
7	Documentary	1050
8	Foreign	1033
9	Games	969
10	Children	945
11	Comedy	941
12	New	940
13	Classics	939
14	Horror	846
15	Travel	837
16	Music	830

可以看到租赁次数最多的类别是 Sports。

8. 查询每位客户的总支出金额（租赁费用），并按总支出金额降序排列。

```
select concat(customer.first_name, " ", customer.last_name) 客户名称,
       sum(amount) 总支出金额
from customer
       join payment using (customer_id)
group by customer_id
order by 总支出金额 desc;
```

	客户名称	总支出金额
1	KARL SEAL	221.55
2	ELEANOR HUNT	216.54
3	CLARA SHAW	195.58
4	RHONDA KENNEDY	194.61
5	MARION SNYDER	194.61
...
595	ANNIE RUSSELL	58.82
596	JOHNNY TURPIN	57.81
597	BRIAN WYMAN	52.88
598	LEONA OBRIEN	50.86
599	CAROLINE BOWMAN	50.85

9. 查询出超过 6 个月没有被租赁的影片名称。

这些影片的租赁日期都是 2005 年的，如果说是距离现在的话那都是超过 6 个月的了。但是按照题意那应该是这样写，所以筛选出了全部的电影名称。

```
select film.title
from film
where not exists(select *
                 from rental
                 right join inventory using (inventory_id)
                 where film.film_id = inventory.film_id
                 and rental_date >= date_sub(current_date, interval 6
                 ↪ month));
```

	title
1	ACADEMY DINOSAUR
2	ACE GOLDFINGER
3	ADAPTATION HOLES
4	AFFAIR PREJUDICE
5	AFRICAN EGG
...	...
996	YOUNG LANGUAGE
997	YOUTH KICK
998	ZHIVAGO CORE
999	ZOOLANDER FICTION
1000	ZORRO ARK