# 

# 

数据库实验指导书

实验三 DBMS应用

目录

1. [实验目的 3](#_bookmark0)
2. [实验环境 3](#_bookmark1)
3. [实验步骤 3](#_bookmark2)
   1. [观察并回答问题 3](#_bookmark3)
      1. [关于视图 5](#_bookmark4)
      2. [关于触发器 6](#_bookmark5)
      3. [关于约束 8](#_bookmark6)
      4. [关于存储过程 9](#_bookmark7)
      5. [关于函数 1](#_bookmark8)2
   2. [创建新用户并分配权限 1](#_bookmark9)3
   3. [设计并实现 1](#_bookmark10)5

[3.4 思考题 17](#_bookmark11)

[附录 1 GRANT 命令 18](#_bookmark12)

# 实验目的

1、 理解视图、触发器、约束、存储过程和函数的基本概念，掌握它们的用法；

2、 能结合实例设计合理的视图、触发器和存储过程；

3、 结合实验加深对数据库完整性和安全性的理解。

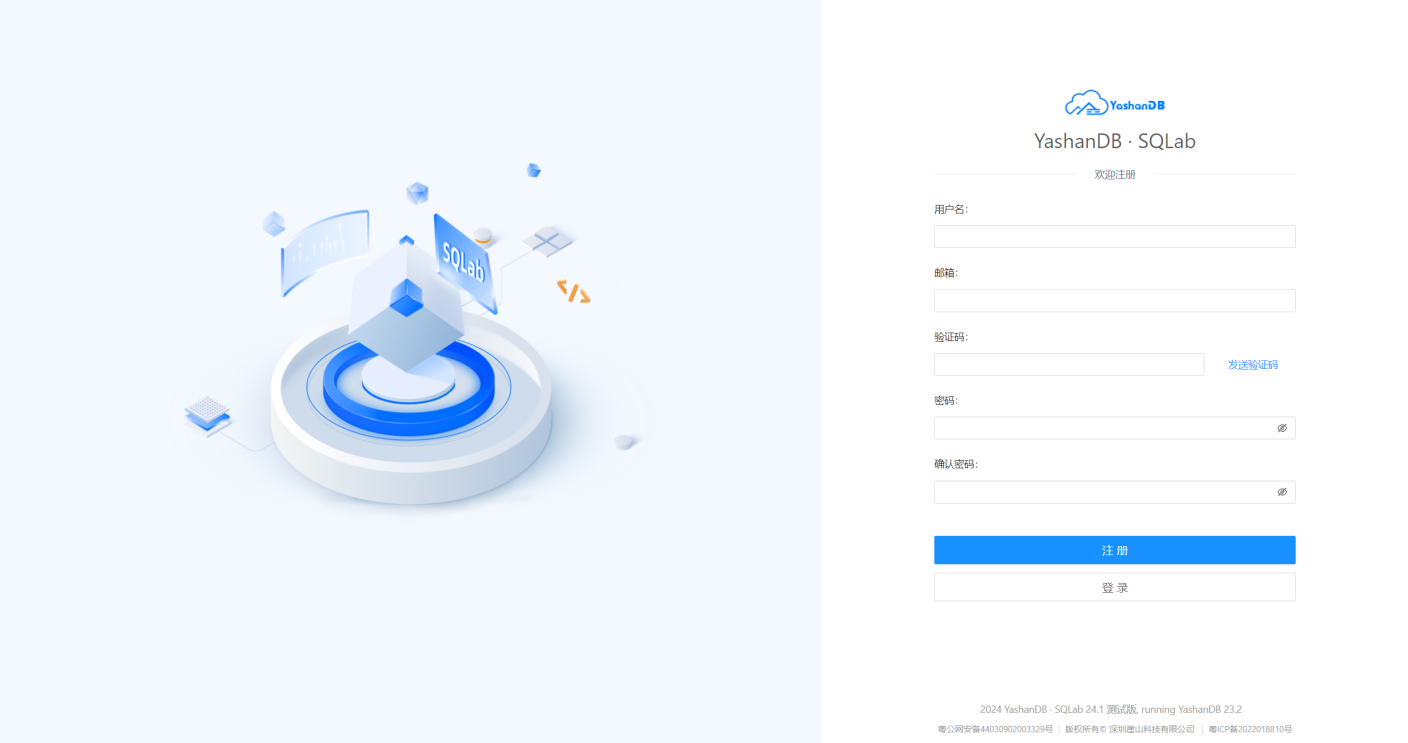
# 实验环境

Windows 10 操作系统、SQLab平台。

## 实验步骤

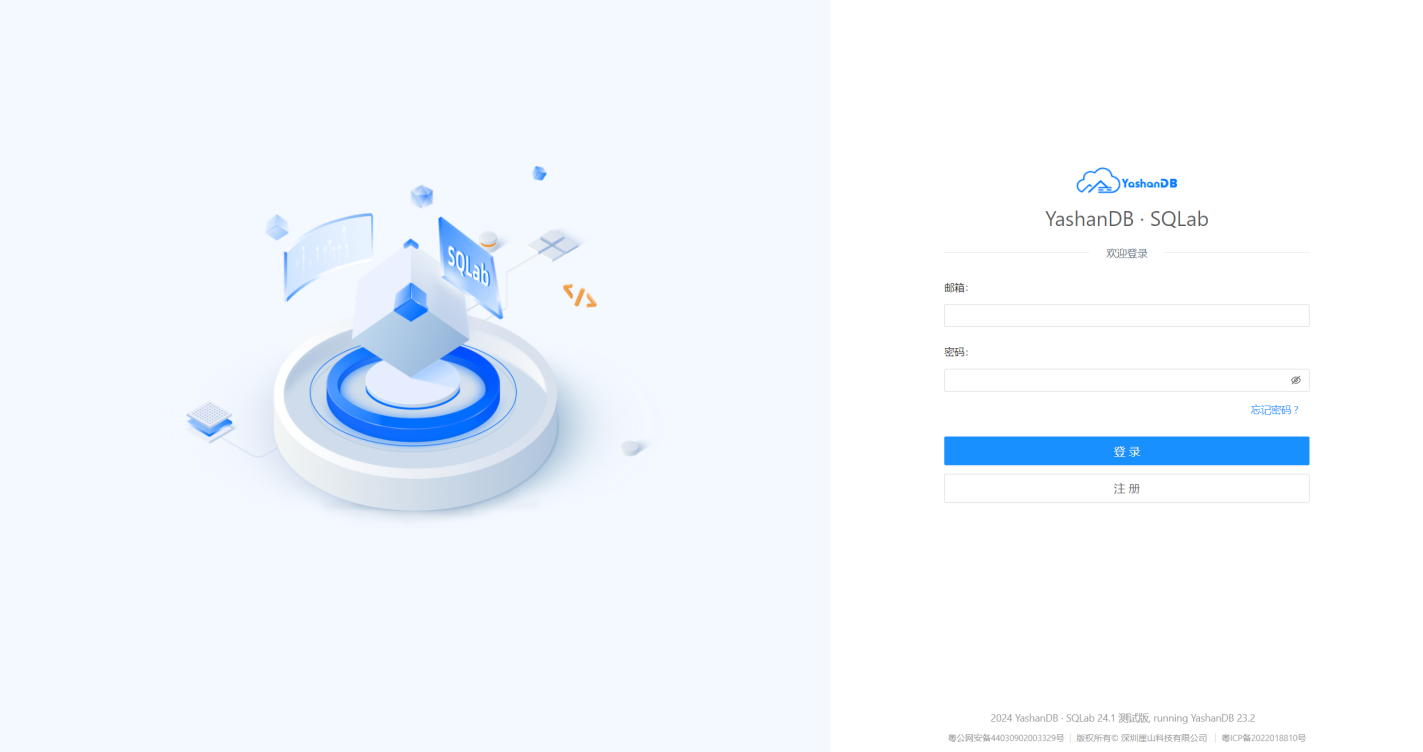
**用户注册（如已注册，此步骤可忽略）**

1. 访问SQLab地址（https://sqlab.yashandb.com）
2. 点击注册按钮，跳转到用户注册页面
3. 填写用户信息，点击发送验证码，然后查看邮箱中的验证码并填写
4. 点击注册按钮，完成注册，并成功登陆SQLab



**用户登录**

1. 访问SQLab地址（https://sqlab.yashandb.com）
2. 填写已注册的邮箱和密码
3. 点击登录按钮，成功登陆SQLab



**使用手册**

1. 点击右上角如下图标，查看使用手册，可以查看完整的平台功能如何使用。



**查看用户及对象**

依次点击“崖山实验室”—>“我的对象”—>“SAKILA”，即可查看到sakila用户下所有的对象。



### 关于视图

1、 回答问题：

1. 分析以下 3 个视图，回答以下问题：

SELECT DBMS\_METADATA.GET\_DDL('VIEW', 'ACTOR\_INFO','SAKILA') DDL FROM DUAL;

SELECT DBMS\_METADATA.GET\_DDL('VIEW', 'FILM\_LIST','SAKILA') DDL FROM DUAL;

SELECT DBMS\_METADATA.GET\_DDL('VIEW', 'SALES\_BY\_FILM\_CATEGORY','SAKILA') DDL FROM DUAL;

|  |  |  |
| --- | --- | --- |
| 视图名 | 关联表 | 作用 |
| actor\_info | actor  film\_actor  film  category  film\_category |  |
| film\_list | film  category  film\_category  language  inventory  rental  payment |  |
| sales\_by\_film\_category | category  film\_category  film  inventory  rental  payment |  |

1. 创建视图分别执行以下 3 句 SQL 语句：

创建视图：

CREATE OR REPLACE VIEW sakila.customer\_simple\_view AS

SELECT customer\_id,

first\_name,

last\_name,

email,

address\_id,

active,

create\_date,

last\_update

FROM customer;

执行SQL语句：

update sakila.customer\_simple\_view set email = 'newemail@example.com' where customer\_id = 1;

update sakila.staff\_list set "zip code" = '518055' where ID = '1';

update sakila.film\_list set price = 1.99 where FID = '1';

截图执行结果，并分析一下视图在什么情况下可以进行 update 操作，什么

情况下不能？

1. 执行以下命令查询 customer\_simple\_view 视图数据是否更新：

select customer\_id,first\_name,last\_name,email from sakila.customer\_simple\_view where customer\_id=1;

### 关于触发器

1、 回答问题：

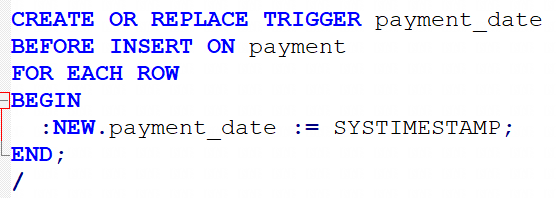
1. 触发器rental\_date建在哪个表上？这个触发器实现什么功能？在这个表上新增一条数据，验证一下触发器是否生效。（截图语句和执行结果）

|  |
| --- |
|  |

1. 触发器 del\_film 建在哪个表上？这个触发器实现什么功能？在这个表上删除一条记录，验证一下触发器是否生效。（截图语句和执行结果）

|  |
| --- |
|  |

1. 现在有这样一个建立触发器的语句：

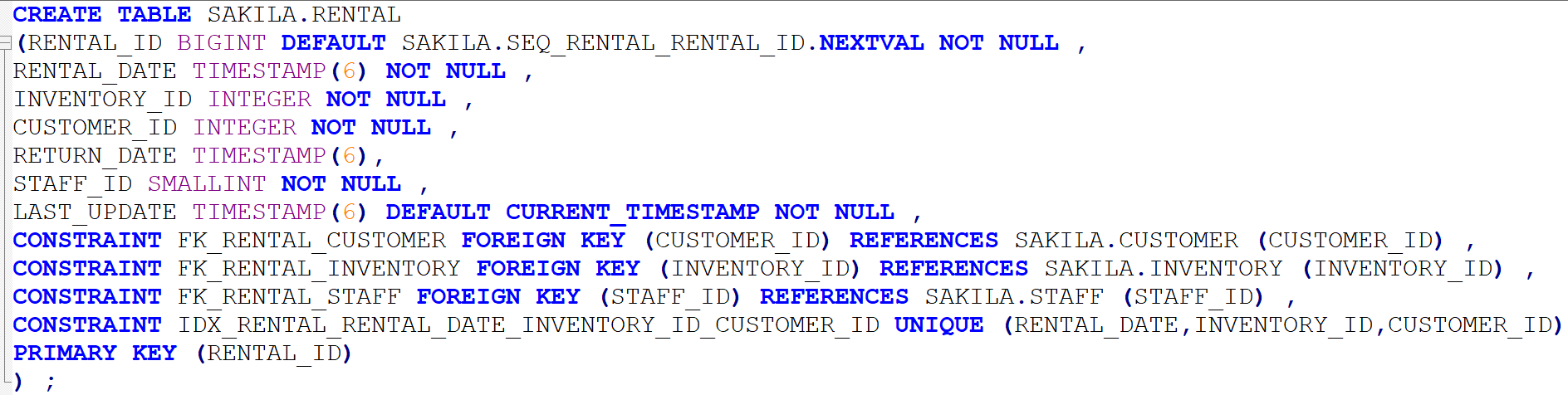


此时需要迁移payment表历史记录，请同学们思考，这个触发器是否会生效？如果生效的话，有什么办法可以避免迁移记录的payment\_date被修改？

|  |
| --- |
|  |

### 关于约束

现有rental表的建表语句如下：



回答问题：

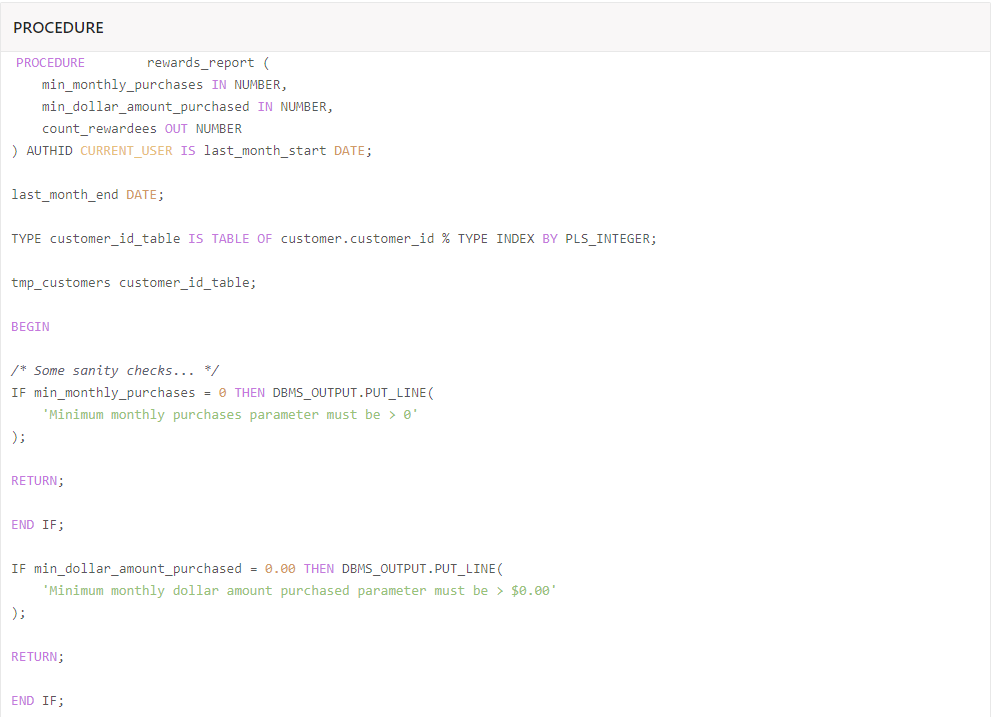
1. rental表上建了哪几种约束？这些约束分别实现什么功能？

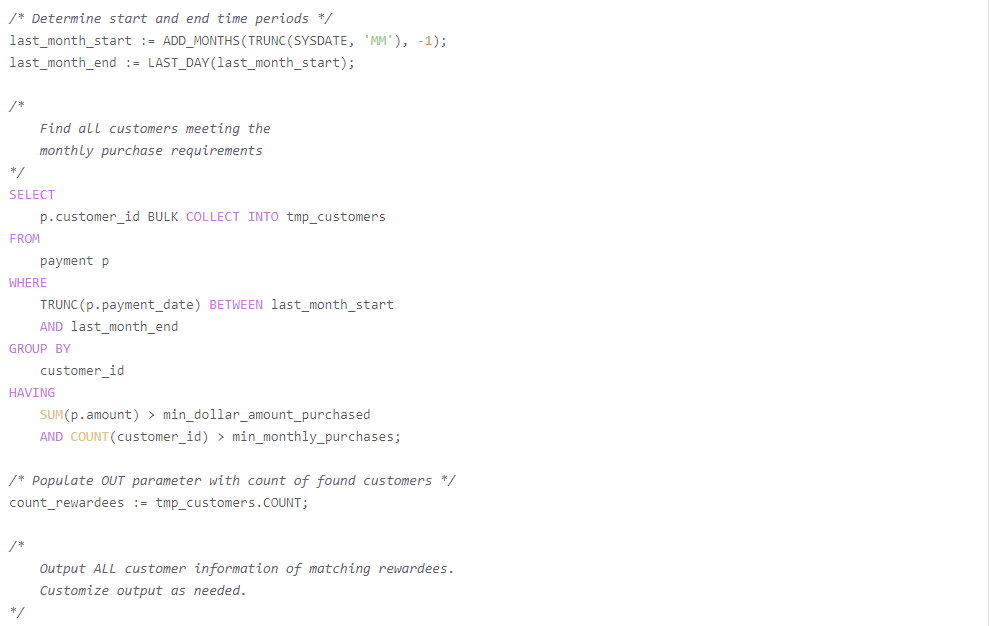
|  |  |
| --- | --- |
| 约束类型 | 功能 |
|  |  |
|  |  |
|  |  |
|  |  |

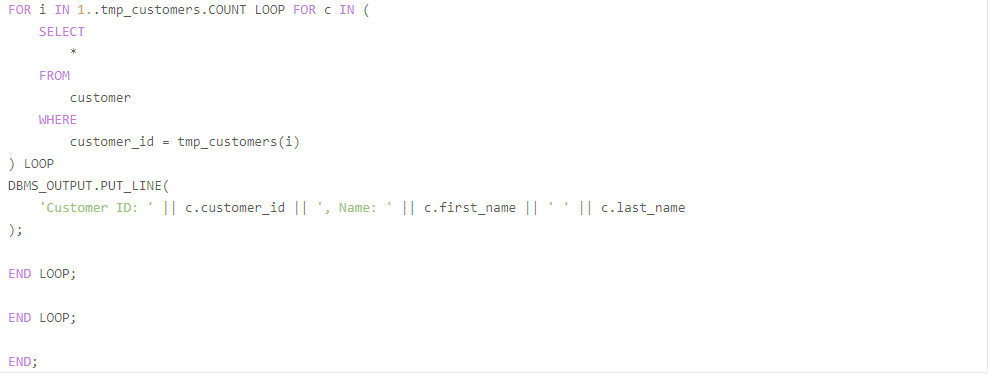
1. 图中的 ON DELETE RESTRICT 和 ON UPDATE CASCADE 是什么意思？

### 关于存储过程

观察 SAKILA用户里面的 rewards\_report 存储过程：





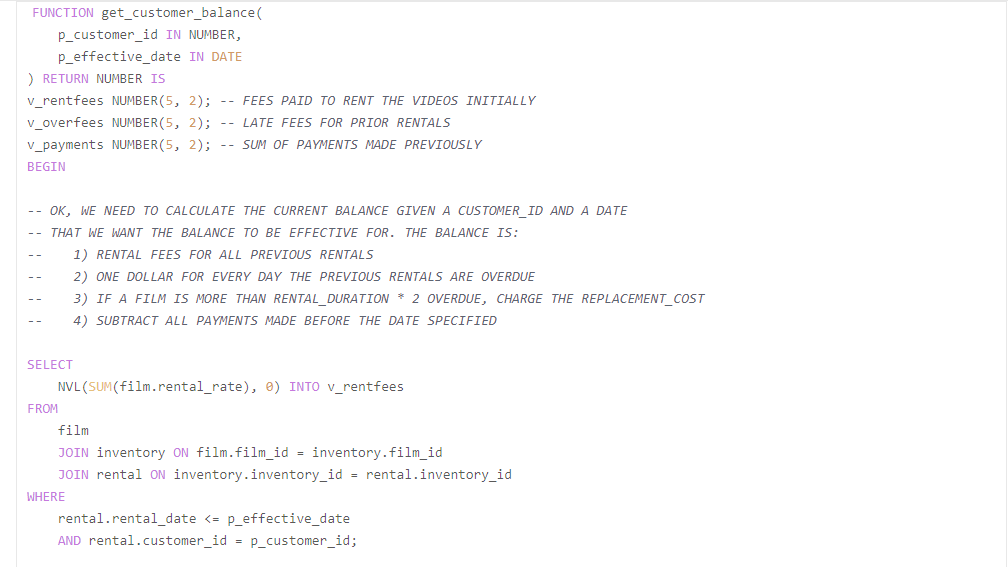


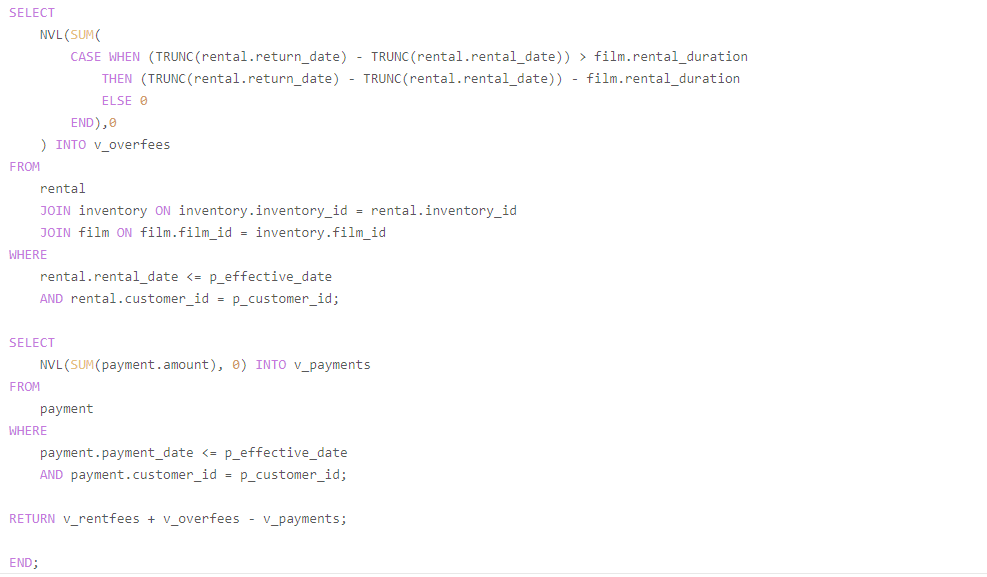
回答问题：

1. 这个存储过程实现了什么功能？输出参数 count\_rewardees 是什么？
2. last\_month\_start := ADD\_MONTHS(TRUNC(SYSDATE, 'MM'), -1);具体是怎么获取到上个月第一天的日期的？请展开说明。

### 关于函数

观察 SAKILA用户里面的 get\_customer\_balance 函数：





回答问题：

1. 这个函数实现了什么功能？返回值是什么？
2. 这个函数体中用到了 3 个函数，是哪几个函数？这 3 个函数的作用分别是？

|  |  |
| --- | --- |
| 函数 | 作用 |
|  |  |
|  |  |
|  |  |

## 创建新用户并分配权限

（请把过程截图填写到实验报告中）

1、 创建新用户 (用户名SAKILA\_TESE，密码 123456)

2、 为新用户SAKILA\_TEST赋予访问 sakila 的权限

更多 grant 的用法参考附录 1。

## 设计并实现

根据应用场景，为 Sakila 数据库合理地设计并实现：

1. 设计 1 个视图，通过关联customer和address 2 个表，可以查看客户及其地址信息的列表，方便用户查看地址信息；

|  |
| --- |
|  |

1. 设计 1 个触发器，融合ins\_film、upd\_film和del\_film的功能，需要在报告里体现触发器生效；

|  |
| --- |
|  |

1. 设计 1 个存储过程，用于获取特定客户在指定时间段内的租赁历史记录。它将返回客户的租赁信息，包括租赁日期、归还日期、租赁的电影标题以及租赁费用。须在报告里调用，并展示结果。

|  |
| --- |
|  |

注意：请将创建语句、执行结果截图记录到实验报告里。

## 思考题

（这部分不是必做题，供有兴趣的同学思考）

在MySQL开发规范中，通常要求为每张表创建一个主键，但是在YashanDB中，表可以没有主键。请分析一下原因。主键是否没有存在的必要？

附录 1 GRANT 命令

一、**grant** 普通数据用户，查询、插入、更新、删除 数据库中某个表数据的权利。

或者，用一条YASHANDB命令来替代：

grant select on testdb.test01 to common\_user;

grant insert on testdb.test01 to common\_user;

grant update on testdb.test01 to common\_user;

grant delete on testdb.test01 to common\_user;

二、**grant** 数据库开发人员，创建表、索引、视图、存储过程、函数。。。等权限。

grant all privileges on testdb.test01 to common\_user;

grant 创建、修改、删除YashanDB数据表结构权限。

grant create table to developer;

grant alter on testdb.test01 to developer;

grant drop any table to developer;

grant 操作 YashanDB 外键权限。

grant references on testdb.test01 to developer;

grant 操作 YashanDB 临时表权限。

grant create table to developer;

grant 操作 YashanDB 索引权限。

grant index on testdb.test01 to developer;

grant 操作 YashanDB 视图、查看视图源代码 权限。

grant create view to developer;

grant select any table to developer;--通过select text from all\_views;查看视图源代码

grant 操作 YashanDB 存储过程、函数 权限。

grant create any procedure to developer; -- now, can show proced ure status

grant alter any procedure to developer;-- now, you can drop a procedure

grant EXECUTE ANY PROCEDURE to developer;

三、**grant** 普通 **DBA** 管理拥有对表的所有对象特权。

grant all privileges on testdb.test01 to common\_dba;

四、**grant** 高级 **DBA** 管理 **YashanDB** 具备所有的系统特权。

grant all privileges to high\_dba;

五、**YashanDB grant** 权限，分别可以作用在多个层次上。

1. grant 作用在系统特权上：

grant select any table to high\_dba; -- high\_dba 具备对数据库中（sys schema除外）任意表、视图、动态视图、物化视图发起查询的权限；动态视图暂不受限

grant all privileges to high\_dba; -- high\_dba具备所有的系统特权

1. grant 作用在角色上：

grant connect to high\_dba; -- CONNECT角色具有CREATE SESSION权限，通过赋予CONNECT角色使用户能够登录会话。

1. grant 作用在对象上：

grant select on testdb.test01 to high\_dba; -- high\_dba 可以查询 test01 中的所有数据。

这里在给一个用户授权多张表时，可以多次执行以上语句。例如：

grant select on testdb.test02 to high\_dba;

grant update on testdb.test03 to high\_dba;

1. grant 作用在表中的索引上：

grant index on testdb.test01 to high\_dba;

1. grant 作用在存储过程上：

grant create any procedure to high\_dba;

六、查看**YashanDB**用户权限

查看用户系统权限：

select \* from dba\_sys\_privs where grantee='common\_user';

查看用户对象权限：

select \* from dba\_tab\_privs where grantee='common\_user';

查看用户拥有的角色：

select \* from dba\_role\_privs where grantee='common\_user';

select \* from dba\_tab\_privs where grantee='common\_user';

select \* from dba\_tab\_privs where grantee='common\_user';

七、撤销已经赋予给 **YashanDB** 用户权限的权限。

revoke 跟 grant 的语法差不多，只需要把关键字 to 换成 from 即可：

grant all privileges to high\_dba;

revoke all privileges from high\_dba;

八、**YashanDB grant**、**revoke** 用户权限注意事项

1.grant, revoke 用户权限后，该用户权限立即生效。

2.如果想让授权的用户，也可以将这些对象或者系统权限 grant 给其他用户，需要选项 **grant option** 和admin option

grant select on testdb.test01 to common\_user with grant option;

grant create table to common\_user with admin option;

这个特性一般用不到。实际中，数据库权限最好由 DBA 来统一管理。

（参考链接 [GRANT | YashanDB Doc](https://doc.yashandb.com/yashandb/23.2/zh/%E5%BC%80%E5%8F%91%E6%89%8B%E5%86%8C/SQL%E5%8F%82%E8%80%83%E6%89%8B%E5%86%8C/SQL%E8%AF%AD%E5%8F%A5/GRANT.html)）