

**期末项目设计报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 题 目 | 基于Oracle的日用品商城的数据库设计 | | |
| 课程 | Oracle数据库应用 | | |
| 学 院 | 信息科学与工程学院 | | |
| 专 业 | 软件工程 | 年级 | 2018级 |
| 学生姓名 | 马巧 | 学号 | 201810601109 |
| 指导教师 | 赵卫东 | 职称 | 副教授 |

|  |  |  |  |
| --- | --- | --- | --- |
| **评分项** | **评分标准** | **满分** | **得分** |
| 文档整体 | 文档内容详实、规范，美观大方 | 10 |  |
| 表设计 | 表，表空间设计合理，数据合理 | 20 |  |
| 用户管理 | 权限及用户分配方案设计正确 | 20 |  |
| PL/SQL设计 | 存储过程和函数设计正确 | 30 |  |
| 备份方案 | 备份方案设计正确 | 20 |  |
| **得分合计** | | |  |

2021 年 6 月 1 日

[日用品商城 3](#_Toc74505801)

[系统总体结构图 3](#_Toc74505802)

[日用品商城ER模型 3](#_Toc74505803)

[实体模型 3](#_Toc74505804)

[实体联系模型 5](#_Toc74505805)

[日用品商城整体ER图 6](#_Toc74505806)

[日用品商城数据表的设计 7](#_Toc74505807)

[创建数据库 9](#_Toc74505808)

[配置插接式数据库PDB 9](#_Toc74505809)

[通过Oracle SQL Developer连接数据库 9](#_Toc74505810)

[用户创建与空间分配 10](#_Toc74505811)

[创建表，约束和索引 12](#_Toc74505812)

[创建触发器、序列和视图 18](#_Toc74505813)

[创建程序包、函数和过程 19](#_Toc74505814)

[创建程序包 24](#_Toc74505815)

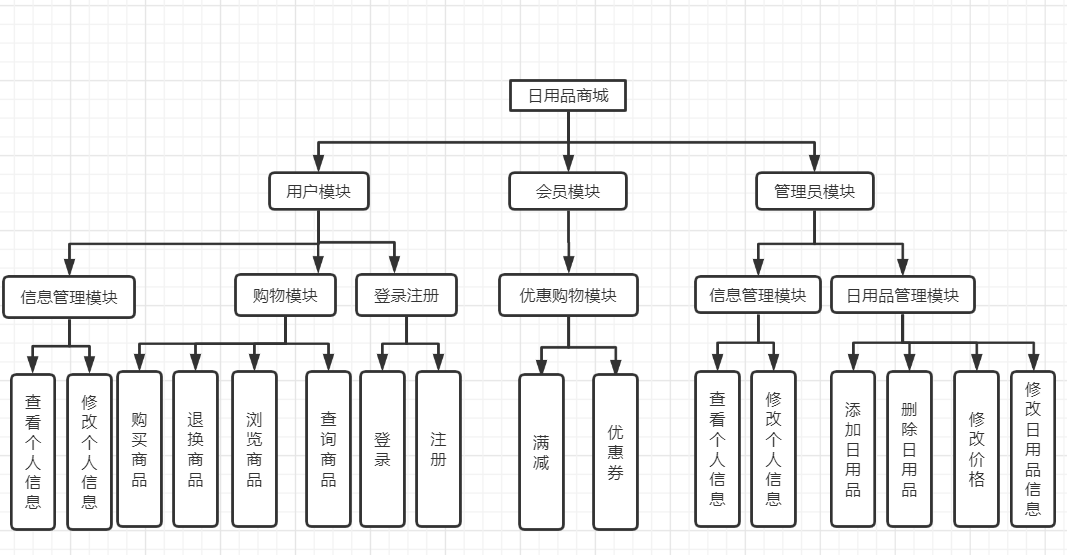
[数据库备份 27](#_Toc74505816)

[参考文献 28](#_Toc74505817)

# 日用品商城

本系统是基于微信小程序实现的一个日用品商城系统，主要针对成都大学学生开放的在线购买日用品的优惠购物平台，只要在线购买后会送至寝室或校园内其他区域。如果同学们去附近的超市购买日用品需要走路，而且也不一定比我们这个渠道便宜。因为我们这个平台直接从厂家处拿货，在价格上是很有优势的。

## 系统总体结构图



## 日用品商城ER模型

### 实体模型

根据应用场景分析，共有5个原始的实体(Entity)，它们是管理员、用户、会员、订单和日用品。

管理员(admin)包括管理员id（admin\_id）、联系电话（tel）、管理员名（admin\_username）和管理员密码(password)，如图1-1。

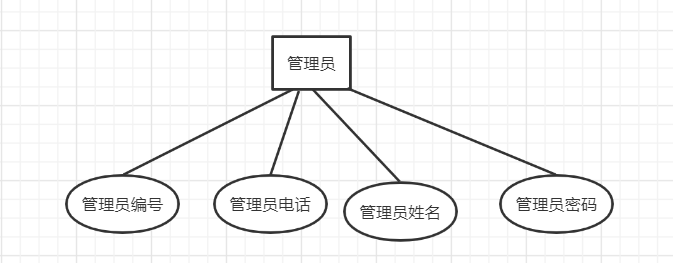


图 1-1 管理员（admin）实体

日用品（goods）包括日用品id（goods \_id）、日用品图片（image）、日用品名称（name）、日用品简介（Introduction）、日用品价格（price）。管理员可对日用品的种类和数量进行增删查改。日用品实体，如图1-2。

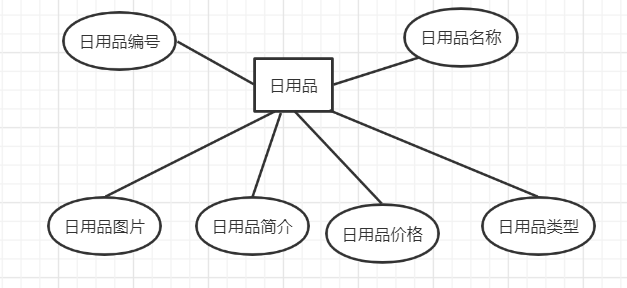


图 1-2 日用品（goods）实体

用户（user）包括用户id（user\_id）、用户名（username）、联系电话（tel）、用户地址（address）。在每次用户下单后，订单上面需要根据用户id或是用户名对该订单的详细记录写进数据库。用户实体，如图1-3。

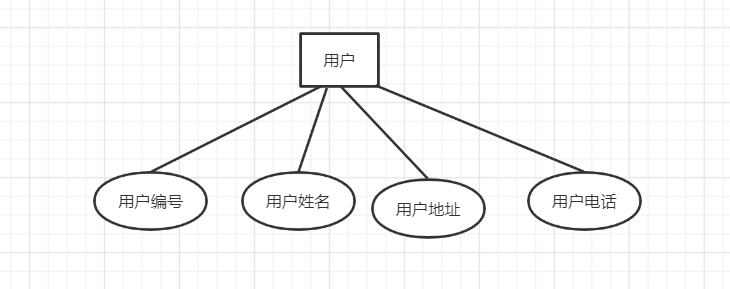


图 1-3 用户（user）实体

会员（member）包括会员id（member\_id）、用户id（user\_id）、会员购买时间（member\_start）、会员截止时间（member\_end）、购买时长（member\_time）。可以根据会员的结束时间来提醒用户是否续费，根据购买时长来计算会员的截止时间。会员实体，如图1-4。

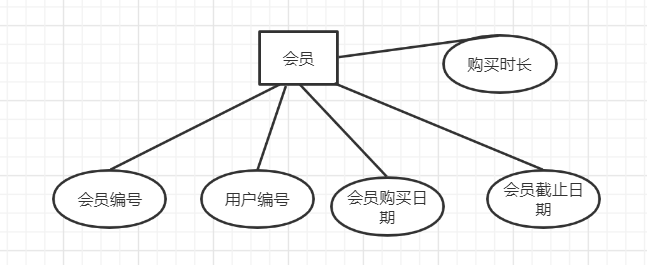


图 1-4 会员（member）实体

订单（order）包括订单编号（order\_id）、用户id（user\_id）、日用品id（goods \_id）、日用品数量（goods\_nums）、用户地址（address）、下单时间（order\_time）、订单总金额（total\_price）。订单实体，如图1-5。

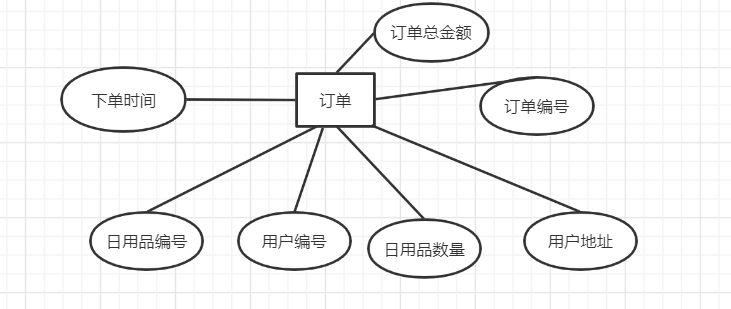


图 1-5 订单（order）实体

### 实体联系模型

管理员和日用品的关系是一对多的关系，在这个商城系统，只有一个管理员账号，每个员工可以登录这个管理员账号对日用品信息进行增删查改。

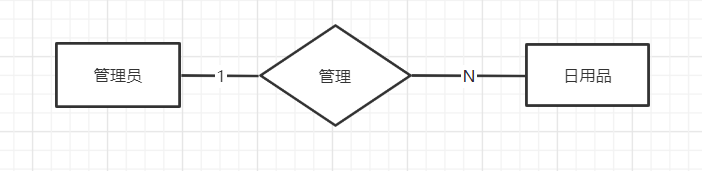


图 1-6 管理员与日用品的关系简图

用户和日用品的关系是多对多，一个用户可以购买多个日用品，同一种日用品可以被多个用户购买。

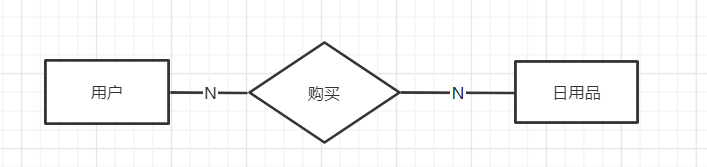


图 1-7 用户和日用品的关系简图

用户和会员的关系是多对一，一个用户只能购买一个会员。

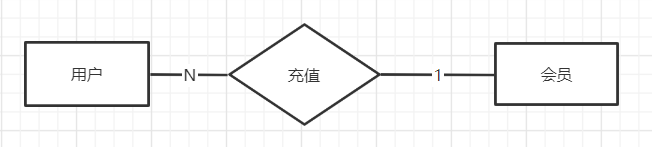


图 1-8 用户和会员的关系简图

用户和订单的关系是多对多，每个人的下单没有限制，一个人可以下多个订单。因为一个用户的下单地址可能不一样，所以在这里每一个订单都必须绑定自己的订单地址。

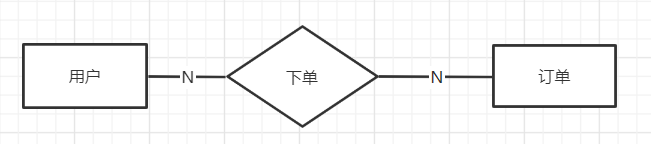


图 1-9 用户和订单的关系简图

## 日用品商城整体ER图

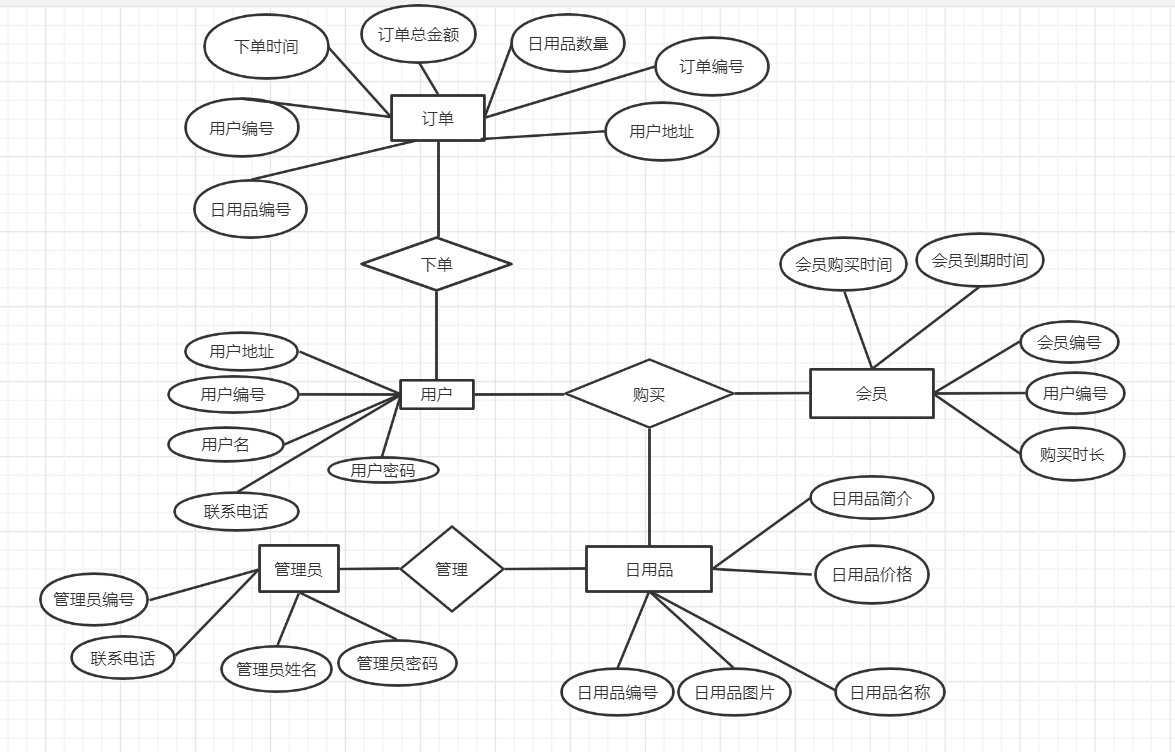


图 1-10 日用品商城ER图

## 日用品商城数据表的设计

ER模型建立好以后，我们就可以设计Oracle的关系表了。在独立实体中找出主要属性设置为主键，比如在订单表中，订单编号（order\_id）是主键。由关系派生出的实体要加入外键关系，如订单中要加入外键用户id（user\_id）属性。

管理员（admin）表包括管理员id（admin\_id）、联系电话(tel)、管理员名（admin\_name）、管理员密码（password）,见表2-1。

表 2-1 2 日用品商城管理员ADMIN

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **数据类型** | **可以为空** | **注释** | **备注** |
| admin\_id | varchar | no | 管理员ID | 主键 |
| tel | varchar | no | 管理员联系方式 |  |
| admin\_name | varchar | no | 管理员名 |  |
| password | varchar | no | 管理员密码 |  |

用户表USERS包括用户ID（user\_id）、用户的联系方式（tel）、用户名（username）、用户的联系地址（address），其中用户的联系地址可以有多个地址，用逗号（，）隔开，见表2-2。

表 2-2 日用品商城用户USERS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **数据类型** | **可以为空** | **注释** | **备注** |
| user\_id | varchar | no | 用户ID | 主键 |
| tel | varchar | no | 用户联系方式 |  |
| username | varchar | no | 用户名 |  |
| address | varchar | yes | 用户地址 | 可以有多个地址 |

日用品表goods包括日用品ID（goods\_id）、日用品图片（goods\_image）、日用品名称（goods\_name）、日用品简介（introduce）、日用品价格（price），见表2-3。

表 2-3 日用品商城日用品GOODS

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **数据类型** | **可以为空** | **注释** | **备注** |
| goods\_id | varchar | no | 日用品ID | 主键 |
| goods\_image | varchar | no | 日用品图片 |  |
| goods\_name | varchar | no | 日用品名称 |  |
| introduce | varchar | yes | 日用品简介 | 可以为空 |
| price | float | no | 日用品价格 |  |
| type | varchar | no | 日用品的类型 | 种类 |

会员表MEMBER包括会员ID（member\_id）、会员购买时间（member\_start）、会员截止时间（member\_end）、购买时长（member\_time）、用户ID（user\_id），见表1-4。

表 2-4 日用品商城会员MEMBER

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **数据类型** | **可以为空** | **注释** | **备注** |
| member\_id | varchar | no | 会员ID | 主键 |
| member\_start | date | no | 会员开始时间 |  |
| member\_end | date | no | 会员结束时间 |  |
| member\_time | varchar | no | 会员总时间 |  |
| user\_id | varchar | no | 用户id | 外键 |

订单（order）包括订单编号（order\_id）、用户id（user\_id）、商品id(goods\_id)、商品数量(goods\_nums)、用户地址（address）、下单时间（order\_time）、订单总金额（total\_price）。订单实体，如表2-5。

表 2-5 日用品商城订单ORDER

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **数据类型** | **可以为空** | **注释** | **备注** |
| order\_id | varchar | no | 订单ID | 主键 |
| goods\_id | varchar | no | 商品ID | 外键 |
| user\_id | varchar | no | 用户ID | 外键 |
| username | varchar | no | 用户名 |  |
| address | varchar | no | 订单地址 |  |
| order\_time | date | no | 下单时间 |  |
| total\_price | float | no | 订单总金额 |  |
| goods\_nums | Int | no | 商品数量 |  |

## 创建数据库

### 配置插接式数据库PDB

配置插接式数据库PDB可以通过SQL语句创建插接式数据库，在创建时可以选择创建一个新的空的PDB，不包含任何数据；也可以选择一个现有的PDB克隆成另一个PDB，所谓克隆，是指将原有的PDB的用户信息和数据信息一起复制到新的PDB中，这也是Oracle 12C的重要特性。

### 通过Oracle SQL Developer连接数据库

Oracle SQL Developer是一个免费的GUI图形界面的管理和开发工具，可以提高工作效率并简化数据库开发任务。SQL Developer可以在没有安装数据库的客户端上运行，支持Windows、Linux、Mac系统。 可以在Oracle官网上下载，直接使用，无需安装。Oracle SQL Developer是基于Java的应用程序，如果客户端没有安装Java，就需要下载自带有Java的SQL Developer，如果客户端已经安装了Java，就可以下载不带Java版本。

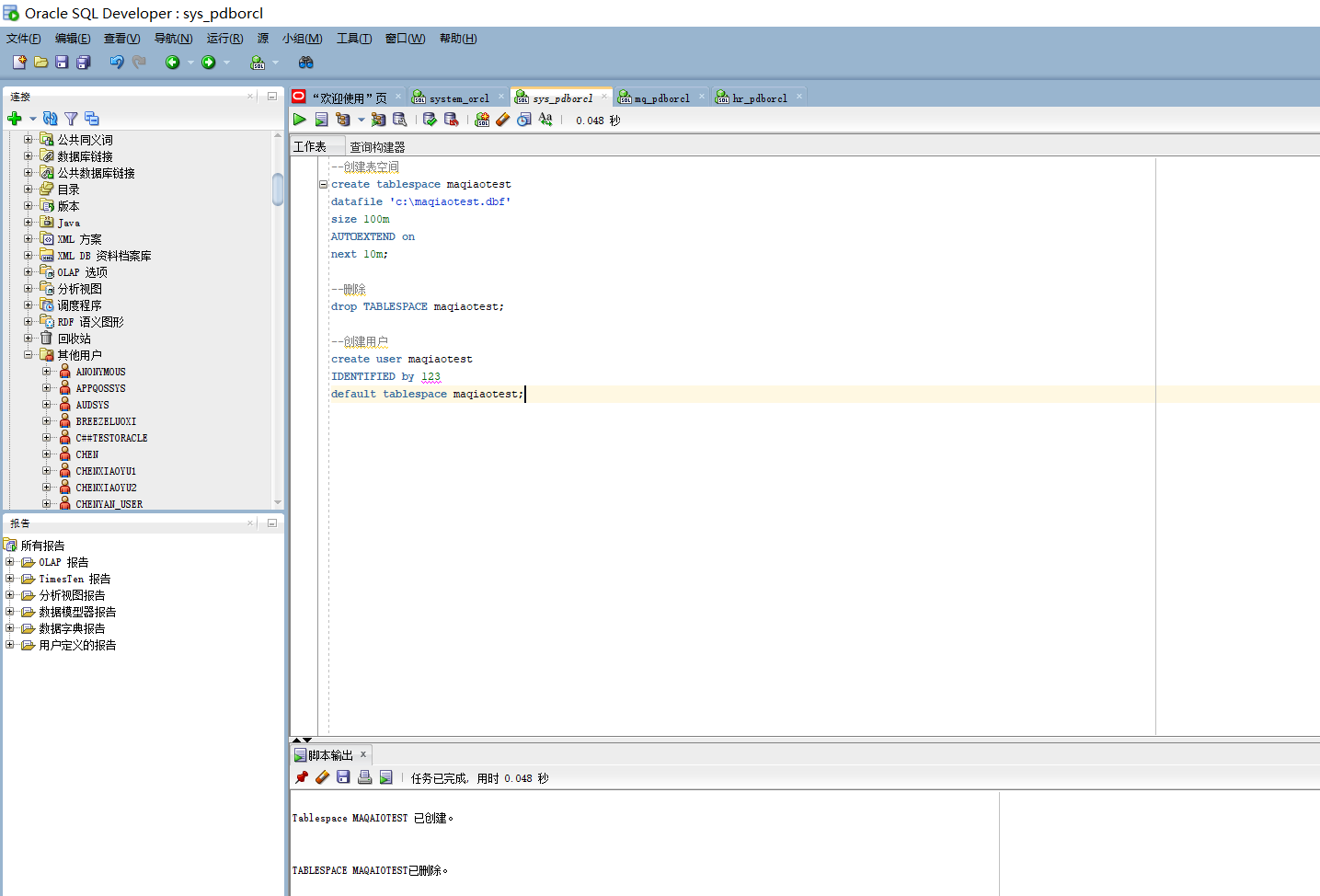


图 2-1 日用品商城通过SQL Developer连接Oracle

## 用户创建与空间分配

表的结构设计之后，还需要考虑用户和空间的分配问题。我们需要为系统创建一个新表空间,新建一个用户（MAQIAO）。另外还需要为日用品商城系统MAQIAO用户存储订单记录。

下面是SYSTEM用户创建的一个表空间MAQIAO的命令，注意给MAQIAO表空间分配了两个数据文件：pdbtest\_MAQIAO1.dbf和pdbtest\_MAQIAO2.dbf，这两个数据文件初始大小都是100M，所以表空间的初始大小是200M。

|  |
| --- |
| create tablespace MAQIAO  datafile  '/home/oracle/app/oracle/oradata/orcl/pdborcl/pdbtest\_MAQIAO1.dbf'  size 100M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED,  '/home/oracle/app/oracle/oradata/orcl/pdborcl/pdbtest\_MAQIAO2.dbf'  size 100M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED  EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO; |

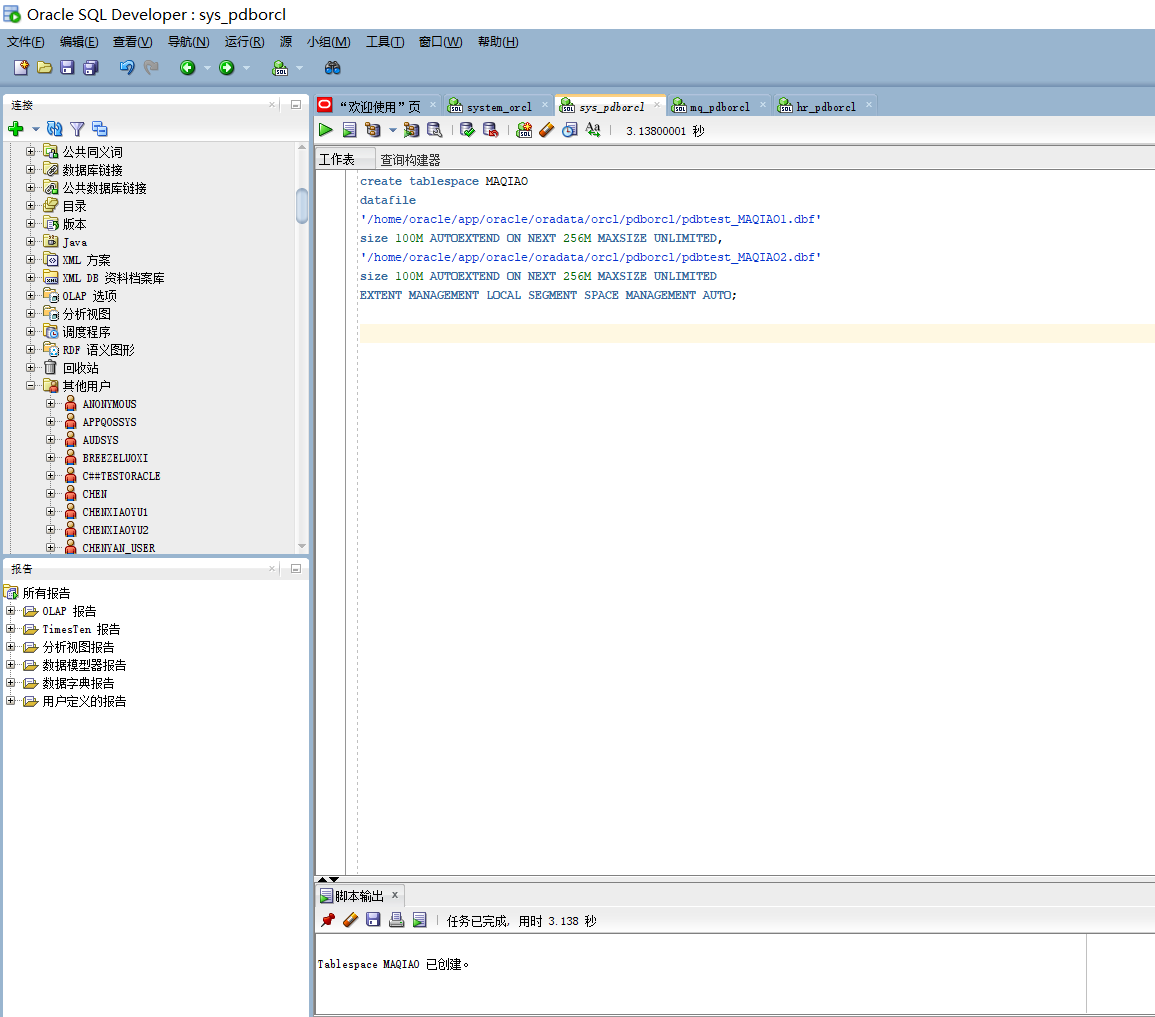


图 3-1 日用品商城创建表空间MAQIAO

表空间MAQIAO创建完成之后，就可以创建用户了。我们可以为本系统设计的用户名称是MAQIAO。用户创建之后，给用户MAQIAO\_U分配表空间MAQIAO的使用配额，再分配角色CONNECT和RESOURCE，以便该用户可以连接数据库，可以创建资源（表，过程，序列等资源对象），最后再分配一个系统权限：”CREATE VIEW”，以便该用户可以创建视图。

|  |
| --- |
| --创建用户  create user MAQIAO\_U  IDENTIFIED by 123  default tablespace MAQIAO;  --为用户授权  -- ROLES  GRANT "CONNECT" TO MAQIAO\_U WITH ADMIN OPTION;  GRANT "RESOURCE" TO MAQIAO\_U WITH ADMIN OPTION;  ALTER USER MAQIAO\_U DEFAULT ROLE "CONNECT","RESOURCE";  -- SYSTEM PRIVILEGES  GRANT CREATE VIEW TO MAQIAO\_U WITH ADMIN OPTION;  --切换到自己的用户  --给用户MAQIAO修改用户表空间和临时表空间  alter user MAQIAO DEFAULT TABLESPACE "USERS" TEMPORARY TABLESPACE "TEMP";  -- QUOTAS  ALTER USER MAQIAO QUOTA UNLIMITED ON USERS;  ALTER USER MAQIAO QUOTA UNLIMITED ON MAQIAO; |

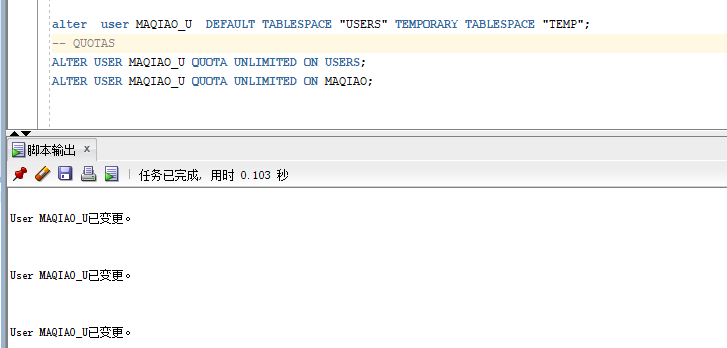


图 3-2 日用品商城创建用户以及分配表空间

到现在，我们有了两个表空间USERS（原有的）和MAQIAO（新建的）。这里我们假定店主的销售订单非常多，订单表的记录非常大，可能上千万条记录，所以我们把ORDERS表设计为基于REDER\_DATA的分区表，以加快局部时间范围的查询速度。存储空间的分如表4-1。

表 4-1 日用品商城存储空间分配

|  |  |  |
| --- | --- | --- |
| **表** | **表空间USERS** | **表空间MAQIAO** |
| ORDERS | 存储全部数据 |  |
| USERS | 存储全部数据 |  |
| ADMIN | 存储全部数据 |  |
| MEMBER | 存储2021年前（不含）的数据 | 存储2021年前（不含）的数据 |
| GOODS | 存储全部数据 |  |

## 创建表，约束和索引

用户和空间分配完成后，可以创建表，约束和索引了。创建表的命令是 CREATE TABLE，因为 ORDERS 表按分区存储，分区类型选择“RANG”范围分区，创建 ORDERS 表和其他表的语句是：

|  |
| --- |
| CREATE TABLE GOODS(  goods\_id VARCHAR2(10) PRIMARY KEY NOT NULL,  goods\_image VARCHAR2(14) NOT NULL,  goods\_name VARCHAR2(13) NOT NULL,  introduce VARCHAR2(14) NOT NULL,  price FLOAT NOT NULL,  type VARCHAR2(14) NOT NULL  );  CREATE TABLE USERS  (  user\_id VARCHAR2(10) PRIMARY KEY NOT NULL,  tel VARCHAR2(10) NOT NULL,  username VARCHAR2(10) NOT NULL,  address VARCHAR2(10)  );  CREATE TABLE ADMIN  (  admin\_id VARCHAR2(10) PRIMARY KEY NOT NULL,  tel VARCHAR2(10) NOT NULL,  admin\_name VARCHAR2(10) NOT NULL,  password VARCHAR2(10) NOT NULL  );  CREATE TABLE MEMBER  (  member\_id VARCHAR2(10) PRIMARY KEY NOT NULL,  member\_start DATE NOT NULL,  member\_end DATE NOT NULL ,  member\_time VARCHAR2(10) NOT NULL,  user\_id VARCHAR2(10) NOT NULL,  constraint fk\_member\_user\_id foreign key (user\_id) references USERS(user\_id)  );  create table ORDERS  (  order\_id VARCHAR2(10) PRIMARY KEY NOT NULL,  goods\_nums NUMBER(2) NOT NULL,  address VARCHAR2(10) NOT NULL,  order\_time DATE NOT NULL,  total\_price FLOAT NOT NULL,  goods\_id VARCHAR2(10) NOT NULL,  user\_id VARCHAR2(10) NOT NULL  ); |

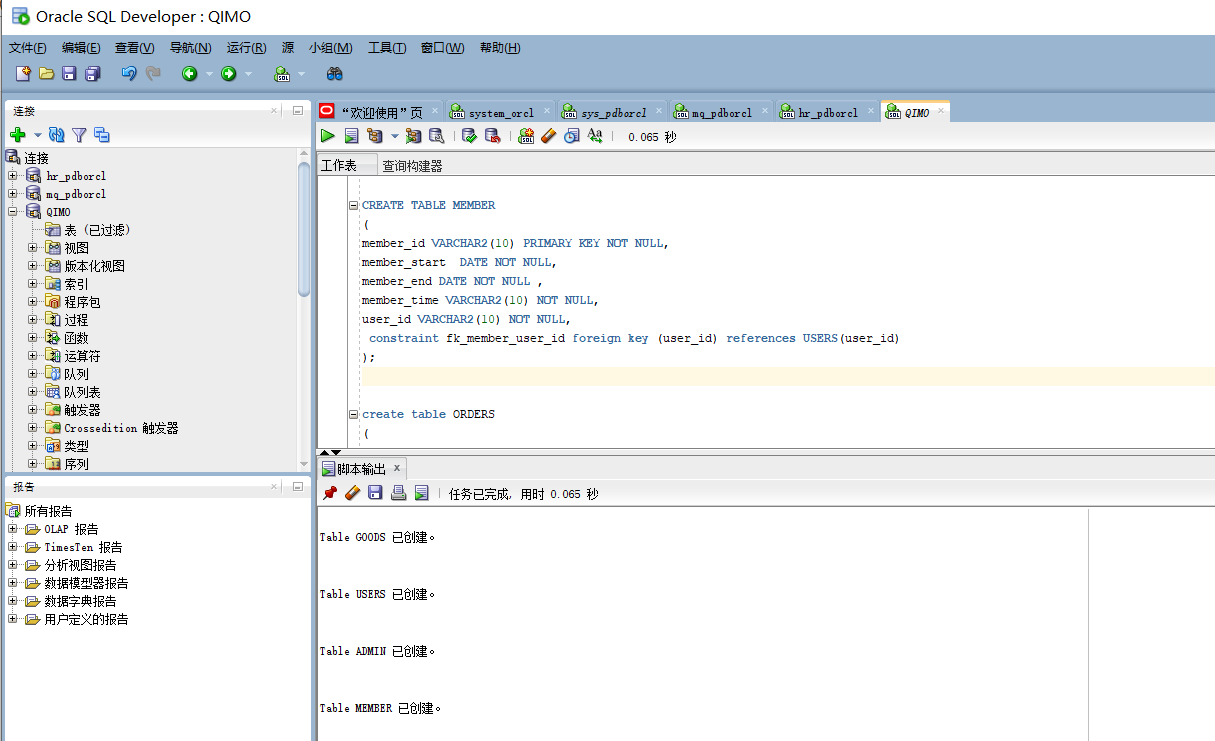


图 5-1 日用品商城创建表及其约束

在这个命令中，核心的语句是“PARTITION BY RANGE (order\_time)”，表示按订单时间 ORDER\_TIME 的范围进行分区。分区 PARTITION\_BEFORE\_2020存储订单时间小于2020年 的订单记录，这个分区存储在表空间 USERS 中 ,而分区PARTITION\_BEFORE\_2021 存储的是订单时间小于 2021 年的订单记录，由于有分区PARTITION\_BEFORE\_2020 的存在，PARTITION\_BEFORE\_2021 分区实际只存储 2020年一年的记录。创建 ORDERS表的部分语句如下：

|  |
| --- |
| create table ORDERS  (  order\_id VARCHAR2(10) PRIMARY KEY NOT NULL,  goods\_nums NUMBER(2) NOT NULL,  address VARCHAR2(10) NOT NULL,  order\_time DATE NOT NULL,  total\_price FLOAT NOT NULL,  goods\_id VARCHAR2(10) NOT NULL,  user\_id VARCHAR2(10) NOT NULL  )  TABLESPACE USERS  PCTFREE 10 INITRANS 1  STORAGE ( BUFFER\_POOL DEFAULT )  NOCOMPRESS NOPARALLEL  PARTITION BY RANGE (order\_time)  (  PARTITION PARTITION\_BEFORE\_2020 VALUES LESS THAN (  TO\_DATE(' 2020-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',  'NLS\_CALENDAR=GREGORIAN'))  NOLOGGING  TABLESPACE USERS  PCTFREE 10  INITRANS 1  STORAGE  (  INITIAL 8388608  NEXT 1048576  MINEXTENTS 1  MAXEXTENTS UNLIMITED  BUFFER\_POOL DEFAULT  )  NOCOMPRESS NO INMEMORY  , PARTITION PARTITION\_BEFORE\_2021 VALUES LESS THAN (  TO\_DATE(' 2021-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',  'NLS\_CALENDAR=GREGORIAN'))  NOLOGGING  );  --添加约束  alter table ORDERS add constraint fk\_user\_good foreign key (goods\_id) references GOODS(goods\_id) ;  alter table ORDERS add constraint fk\_user foreign key (user\_id) references USERS(user\_id) ; |

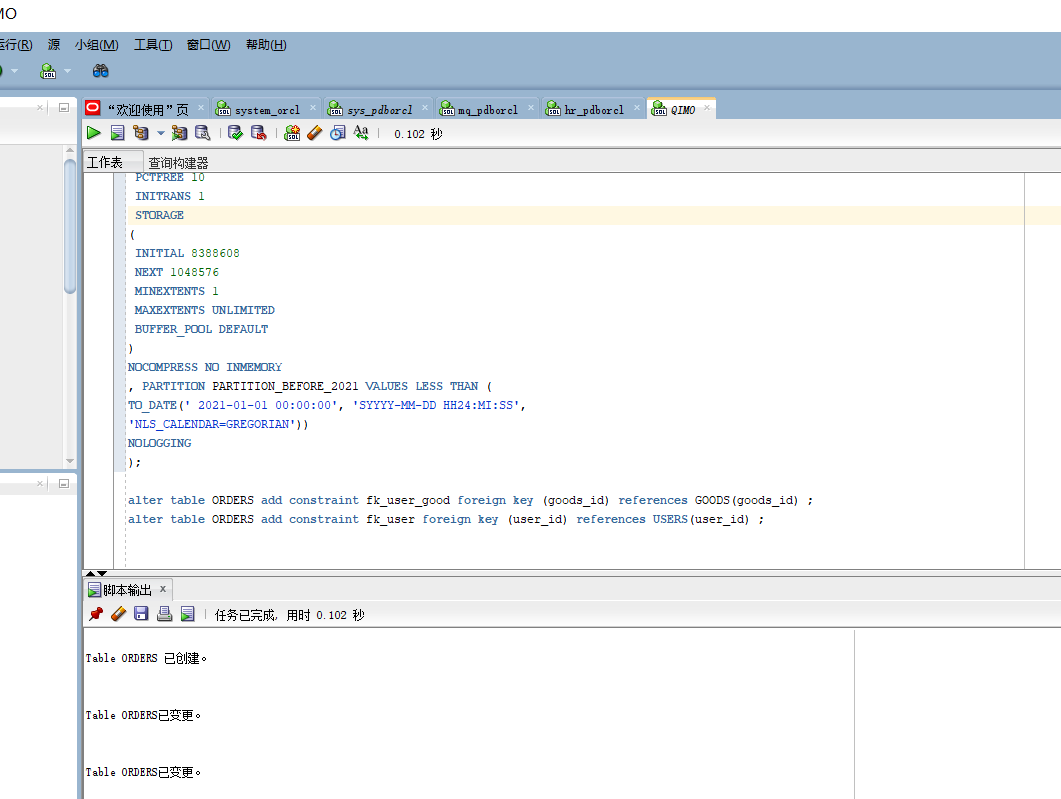


图 5-2 创建order表

给orders要引用的外键的基础表加数据

|  |
| --- |
| insert into USERS values('1','0987690','tom','cdu-10she');  insert into USERS values('2','0187190','lili','cdu-8she');  insert into GOODS values('1','1.jpg','Wash liquid','wash cloth',16.0,'wash');  insert into GOODS values('2','2.jpg','Hair shampoo','wash hair',23.0,'wash'); |

向订单表中加入50000条数据

|  |
| --- |
| declare  dt date;  BEGIN  for i IN 1..100 LOOP  INSERT INTO orders (order\_id,goods\_nums,address,order\_time,total\_price,goods\_id,user\_id)  VALUES ('001'+i,2,'cdu-15she',to\_date('2016-3-2','yyyy-mm-dd'),100,'1','1');  END LOOP;  END; |

查询插入的数据

|  |
| --- |
| select \* from orders;  select count(order\_id) from orders; |

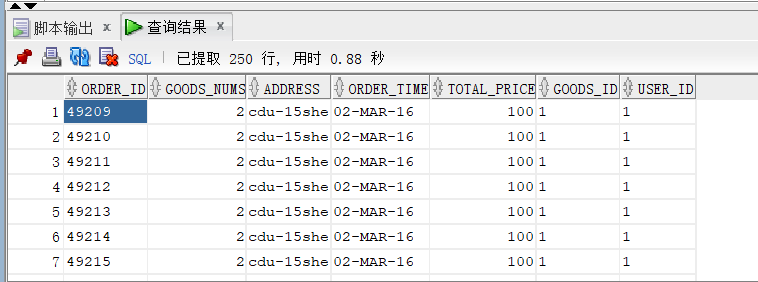


图 5-3 查询



图 5-4 向order表中插入50000条数据

因为订单表的数据很多，我们在这里可以为订单表建立索引。

为user\_id建立索引

|  |
| --- |
| create index info\_user\_id on orders (user\_id); |

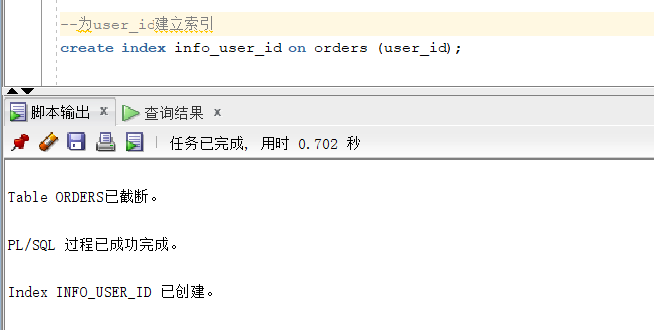


图 5-5 日用品商城创建索引

## 创建触发器、序列和视图

主要用于级联更新，如更新users表中的users\_id时，member表的user\_id也更新。

|  |
| --- |
| CREATE OR REPLACE TRIGGER cascade\_trigger  AFTER UPDATE OF user\_id ON users  FOR EACH ROW  BEGIN  UPDATE MEMBER SET user\_id=:new.user\_id WHERE user\_id=:old.user\_id;  END; |

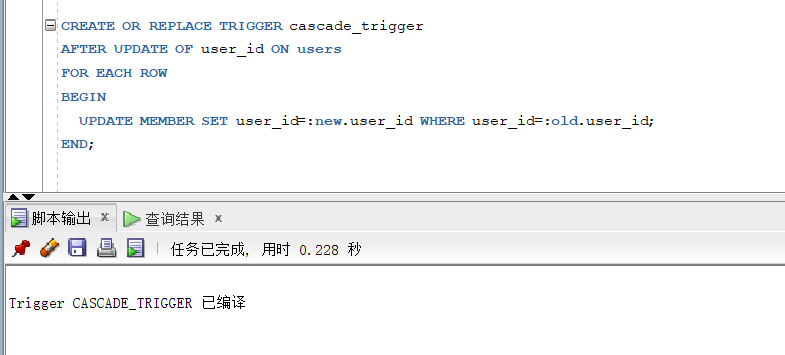


图 6-1 日用品商城创建级联更新用户id的触发器

创建序列：为orders表创建自增序列。

|  |
| --- |
| CREATE SEQUENCE orders\_sequence  INCREMENT BY 1  START WITH 1  NOMAXVALUE  NOCYCLE  CACHE 10; |

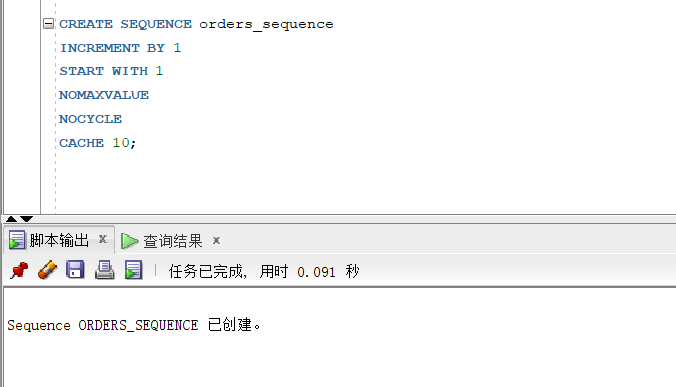


图 6-2 日用品商城创建序列

## 创建程序包、函数和过程

71创建一个存储过程，修改admin表中的admin\_name字段。

|  |
| --- |
| create or replace procedure update\_admin  (  v\_admin\_id varchar2,  v\_admin\_name varchar2  ) is  begin  update ADMIN set admin\_name=v\_admin\_name where admin\_id=v\_admin\_id;  end update\_admin; |

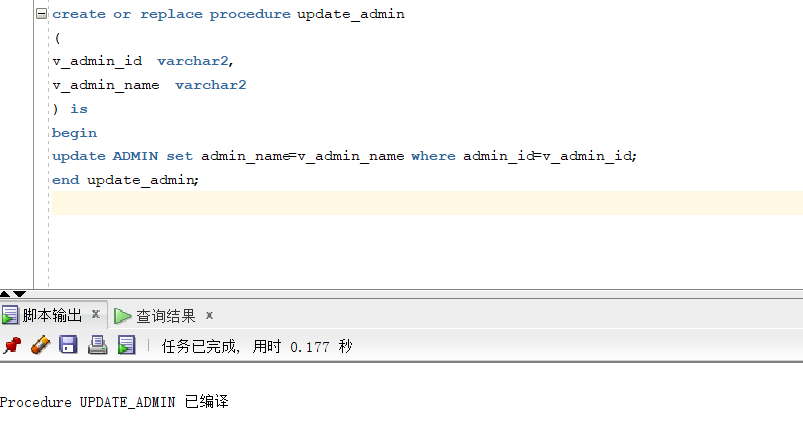


图 7-1 日用品商城创建更新管理员密码的存储过程

测试存储过程：

|  |
| --- |
| INSERT INTO ADMIN VALUES ('1','10087','zhangsan','123');  exec update\_admin('1','test');  select \* from ADMIN; |

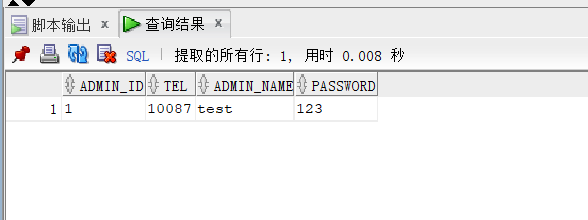


图 7-2 日用品商城测试更新管理员密码的存储过程

创建一个存储过程，修改admin表中的tel字段。

|  |
| --- |
| create or replace procedure update\_adminTel  (  v\_admin\_id varchar2,  v\_admin\_tel varchar2  ) is  begin  update admin set tel=v\_admin\_tel where admin\_id=v\_admin\_id;  end update\_adminTel; |

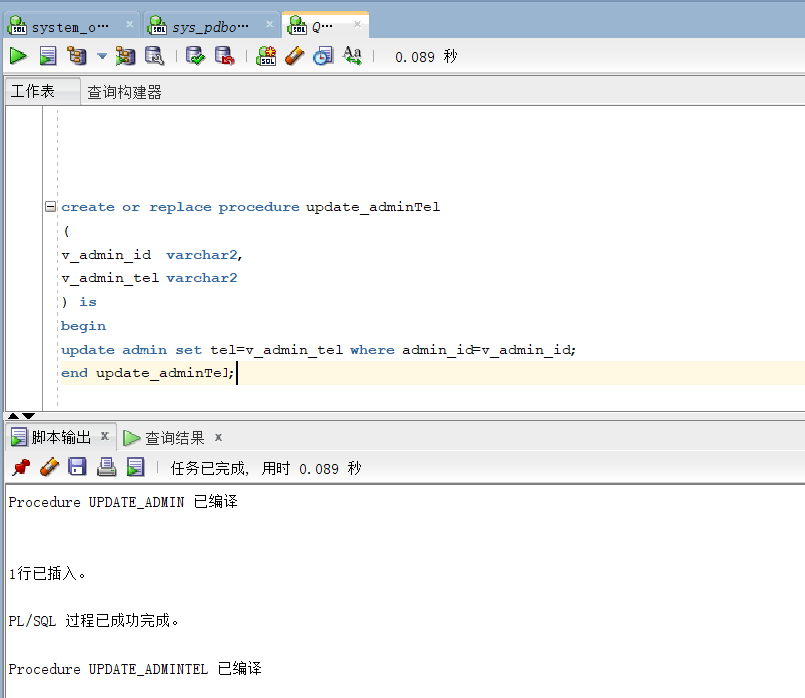


图 7-3 日用品商城创建更新管理员联系电话的存储过程

测试存储过程：

|  |
| --- |
| select \* from admin;  exec update\_adminTel('001','123456');  select \* from admin; |

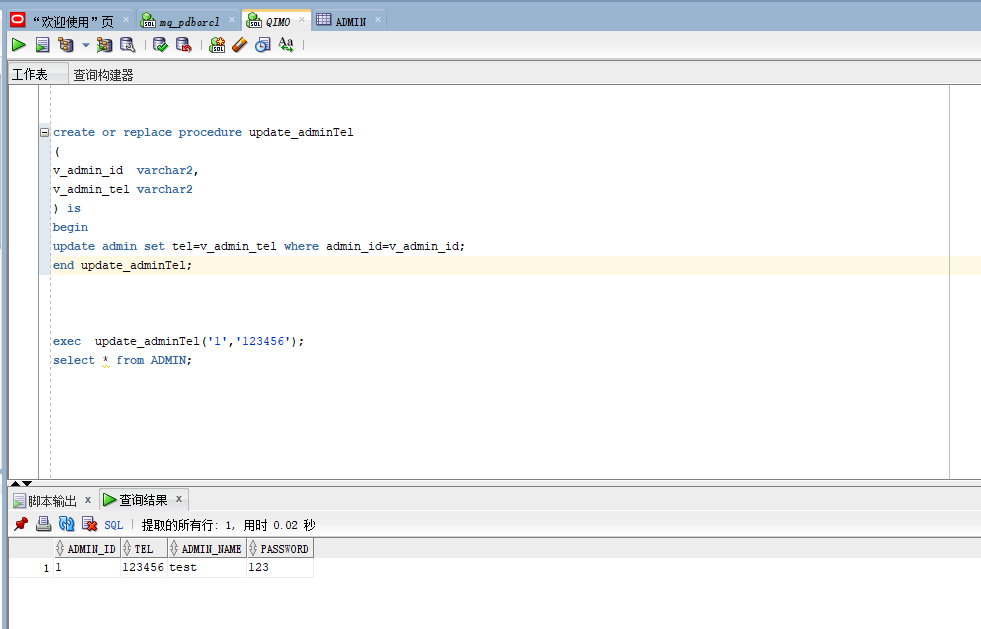


图 7-4 日用品商城测试更新管理员联系电话的存储过程

创建一个存储过程，查看admin表中根据admin\_id查询出admin的name

|  |
| --- |
| create or replace procedure admin\_out\_name  (  v\_admin\_id in VARCHAR2,  v\_name out VARCHAR2  ) is  vname VARCHAR2(10) ;  begin  select admin\_name into vname from admin where admin\_id=v\_admin\_id;  v\_name:=vname;  end; |

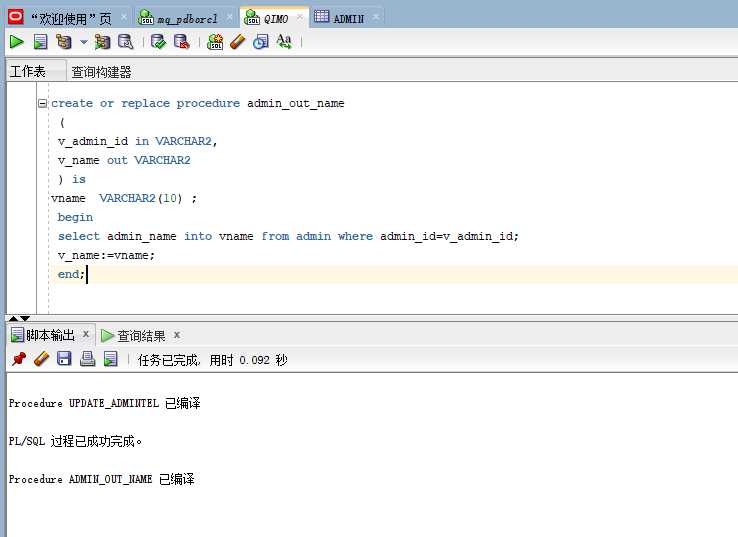


图 7-5 日用品商城创建查看管理员name的存储过程

测试存储过程：

|  |
| --- |
| var vname VARCHAR2  exec admin\_out\_name('1',:vname); |

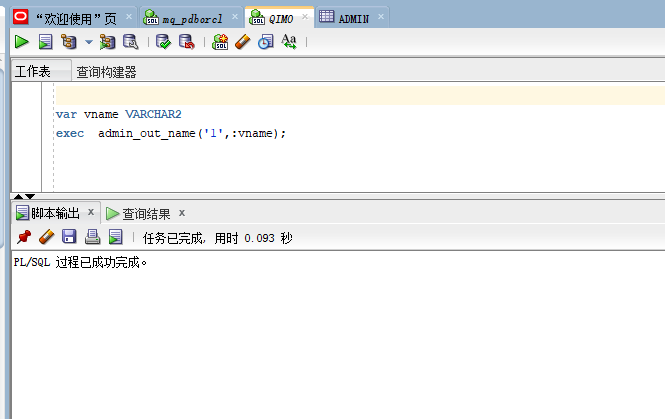


图 7-6 日用品商城测试查看管理员name的存储过程

## 创建程序包

包用于组合逻辑相关的过程和函数，它由包规范和包体两个部分组成。包规范用于定义公用的常量、变量、过程和函数，创建包规范可以使用CREATE PACKAGE命令，创建包体可以使用CREATE PACKAGE BODY。本系统设计了一些函数及过程，共用的一些程序包放在程序包MyPack中。其他的程序包放在自己的包中。

|  |
| --- |
| create package pkg\_admin is  procedure admin\_update\_tel(v\_admin\_id varchar2,v\_admin\_tel varchar2);  function admin\_get\_name(v\_admin\_id varchar2) return varchar2;  end; |

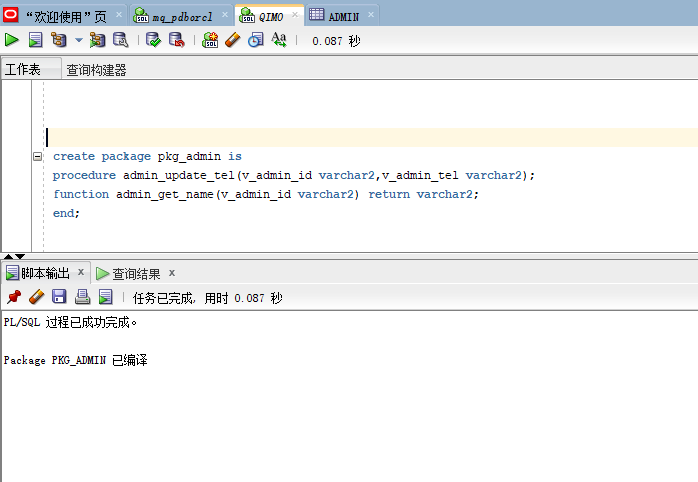


图 7-7 日用品商城创建关于admin表的程序包

创建关于admin表程序包admin\_pkg的包体：

|  |
| --- |
| create or replace package body pkg\_admin  is  procedure admin\_update\_Tel  (  v\_admin\_id varchar2,  v\_admin\_tel varchar2  ) is  begin  update admin set tel=v\_admin\_tel where admin\_id=v\_admin\_id;  end;    function admin\_get\_name  (  v\_admin\_id varchar2  )  return varchar2 is  vname varchar2(10);  begin  select admin\_name into vname from admin where admin\_id= v\_admin\_id;  return vname;  end;  end; |

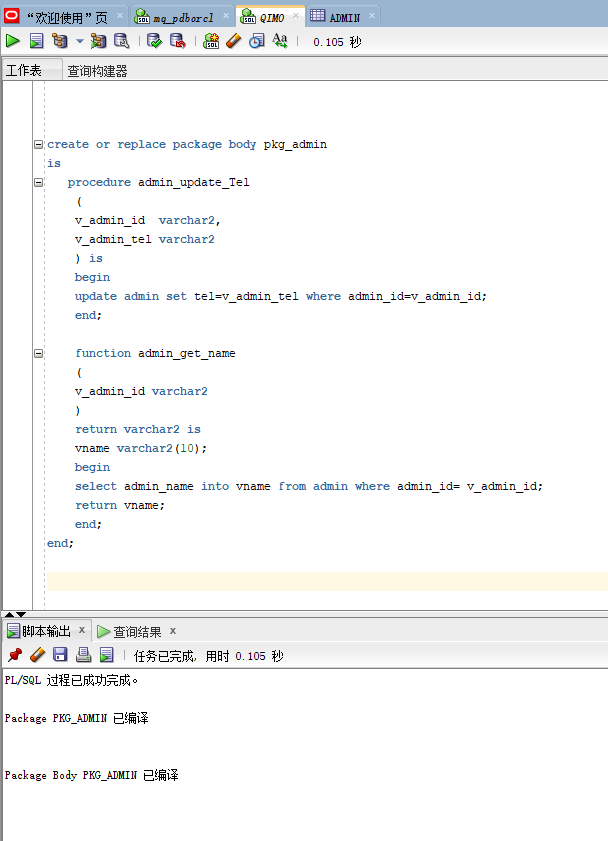


图 7-8 日用品商城创建关于admin表的程序包的包体

要使用包，必须先创建包规范，然后在创建包体当要调用包的过程和函数时，在过程和函数的名称前加上包名作为前缀（包名子程序名称），而如果要访问其他方案的包时需要在包的名称前加上方案的名称（方案名称包名子程序名称）

测试admin\_pkg的程序包：

|  |
| --- |
| select pkg\_admin.admin\_get\_name('1') from dual; |

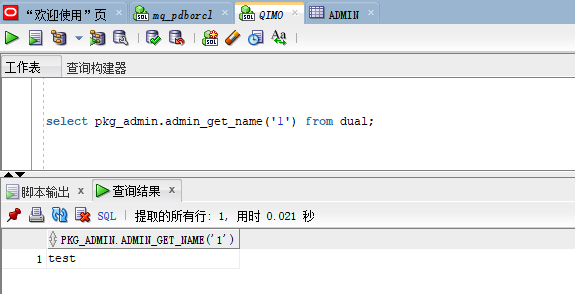


图 7-9 日用品商城测试admin程序包的函数

|  |
| --- |
| exec pkg\_admin.admin\_update\_Tel('1','1335094249');  select \* from admin where admin\_id='1'; |

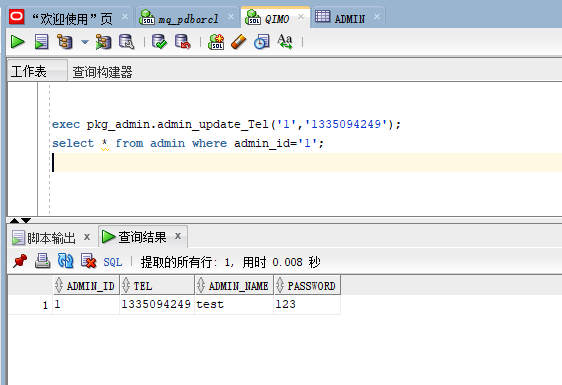


图 7-10日用品商城测试admin程序包的函数

## 数据库备份

数据库备份有冷备份、热备份、物理备份三种方法。

冷备份：将数据库停下来，然后把数据库的PGDATA目录拷贝下来就可以。

热备份：

1、使用数据库的PIRT方法进行热备份。PIRT的原理--时间点恢复(Point In Time Recovery)

2、使用文件系统或设备级别的快照功能完成备份。因为使用了快照，所以也能让备份出来的

本系统拟采用冷备份的方式进行备份。

步骤

关闭数据库

|  |
| --- |
| shutdown immediate； |

(2) 复制文件

1）复制三个文件夹

|  |
| --- |
| oradata(datafile， controlfile，redo);  flash\_recovery\_area datafile;  flash\_recovery\_area controlfile;  flash\_recovery\_area redo; |

2）dbhome\_1下的内容

|  |
| --- |
| database(PWDfile、pfile);  dbs(spfile);  NETWORK/ADMIN(listener.ora、tnsnames.ora); |

(3) 恢复数据库

1）创建一个和原来一样的数据库。(安装路径和数据库名必须和原来一致)

2）停止数据库shutdown immediate；

3）复制安装目录下的admin、oradata、flash\_recovery\_area覆盖，复制database(PWDfile、pfile)

覆盖

4）启动数据库 startup;

备注：

无归档模式：将备份文件拷贝回原来的目录即可，然后启动数据库。

归档模式：

①将数据文件、控制文件拷贝回原来目录

②SQL>startup mount;

SQL>recover database using backup controlfile;

③将相应的归档日志和联机日志拖到CMD命令窗口进行跑日志。

④SQL>alter database open resetlogs;

## 参考文献

[1]Oracle 12c 数据库基础教程[M] 赵卫东、刘永红，于曦北京：科学出版社，2017

[2]oracl 12C dataguard搭建 [https://blogcsdnnet/s754520480/article/details/79789771](https://blog.csdn.net/s754520480/article/details/79789771)

[3]Oracle 数据库应用 [https://githubcom/zwdcdu/oracle](https://github.com/zwdcdu/oracle)