

**期末项目设计报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 题 目 | 基于Oracle的某项目的数据库设计 | | |
| 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 |  | 学号 |  |
| 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

**基于Oracle的图书销售系统的数据库分析与设计**

**成都大学信息科学与工程学院**

| **学号** | **班级** | **姓名** |
| --- | --- | --- |
| 201810414420 | 软件(本)18-4 | 马超 |

**1. 概述**

* 基于Oracle的书籍销售管理系统的实现。
* 管理员的功能主要有：一是添加书籍，二是查询书籍，三是修改书籍，以及对于用户的管理。
* 用户的功能是能够查找书籍，并且将书籍加入到购物车中，查询书籍的详细信息。
* 系统自动计算每个用户的购物车商品总金额。

**2. 数据表设计**

管理员表：

| **字段** | **类型** | **约束** | **默认值** | **说明** |
| --- | --- | --- | --- | --- |
| id | int | primary key |  | 管理员id |
| password | varchar | not null |  | 密码 |
| admin | varchar | not null |  | 角色 |

员工表：

| **字段** | **类型** | **约束** | **默认值** | **说明** |
| --- | --- | --- | --- | --- |
| id | int | primary key |  | 员工id |
| password | varchar | not null |  | 密码 |
| username | varchar | not null |  | 用户名 |
| phone | carchar | not null |  | 电话 |
| address | varcahr | not null |  | 地址 |

商品表：

| **字段** | **类型** | **约束** | **默认值** | **说明** |
| --- | --- | --- | --- | --- |
| id | int | primary key |  | 商品id |
| bookname | varchar | not null |  | 书名 |
| pricenumber | varchar | not null |  | 价格 |
| describe | varchar | not null |  | 描述 |

购物车表：

| **字段** | **类型** | **约束** | **默认值** | **说明** |
| --- | --- | --- | --- | --- |
| id | int | primary key |  | id |
| amount | varchar | not null |  |  |
| pid | varchar | not null |  |  |
| bookuser\_id |  | foreign key |  |  |

论坛表：

| **类型** | **约束** | **默认值** | **说明** |
| --- | --- | --- | --- |
| int | primary key |  | id |
| content | not null |  |  |

**3. 创建表空间**

创建两个表空间mc01和mc02

 Create Tablespace mc01  
 datafile  
 '/home/oracle/app/oracle/oradata/orcl/pdborcl/mc01\_1.dbf'  
  SIZE 100M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED,  
 '/home/oracle/app/oracle/oradata/orcl/pdborcl/mc01\_2.dbf'  
  SIZE 100M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED  
 EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;  
 ​  
 ​  
 Create Tablespace mc02  
 datafile  
 '/home/oracle/app/oracle/oradata/orcl/pdborcl/mc02\_1.dbf'  
  SIZE 100M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED,  
 '/home/oracle/app/oracle/oradata/orcl/pdborcl/mc02\_2.dbf'  
  SIZE 100M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED  
 EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;

1. 创建角色和用户

**用户默认使用表空间mc01** **创建第一个角色和用户**

* 创建角色mc将connect,resource,create view授权给mc
* 创建用户mmcc
* 分配90M空间给lh并将角色mc1授权给用户lh

 CREATE ROLE mc;  
 ​  
 GRANT connect,resource,CREATE VIEW TO mc;  
 ​  
 CREATE USER mmcc IDENTIFIED BY 123 DEFAULT TABLESPACE mc01 TEMPORARY TABLESPACE temp;  
 ​  
 ALTER USER mmcc QUOTA 60M ON mc01;  
 ​  
 GRANT mc1 TO mmcc;  
 ​

创建第二个角色和用户

创建角色mc2，将connect,resource权限给mc2 创建用户mmcc2 分配60M空间给mmcc2并将角色mc2授权给用户mmcc2

 CREATE ROLE mc2;  
 ​  
 GRANT connect,resource TO mc2;  
 ​  
 CREATE USER mmcc2 IDENTIFIED BY 123 DEFAULT TABLESPACE mc01 TEMPORARY TABLESPACE temp;  
 ​  
 ALTER USER mmcc2 QUOTA 60M ON mc01;  
 ​  
 GRANT mc2 TO mmcc2;

**4. 创建管理员表**

 CREATE TABLE ADMINISTRATOR   
 (  
  ID NUMBER(\*, 0) NOT NULL   
 , PASSWORD VARCHAR2(20 BYTE) NOT NULL   
 , ADMIN VARCHAR2(20 BYTE) NOT NULL   
 , CONSTRAINT ADMINISTRATOR\_PK PRIMARY KEY   
   (  
    ID   
   )  
   USING INDEX   
   (  
       CREATE UNIQUE INDEX ADMINISTRATOR\_PK ON ADMINISTRATOR (ID ASC)   
      LOGGING   
       TABLESPACE mc01   
      PCTFREE 10   
      INITRANS 2   
       STORAGE   
       (   
        BUFFER\_POOL DEFAULT   
       )   
      NOPARALLEL   
   )  
   ENABLE   
 )   
 LOGGING   
 TABLESPACE mc01   
 PCTFREE 10   
 INITRANS 1   
 STORAGE   
 (   
  BUFFER\_POOL DEFAULT   
 )   
 NOCOMPRESS   
 NO INMEMORY   
 NOPARALLEL;

**5. 创建用户表**

 CREATE TABLE BOOKUSER   
 (  
  ID NUMBER(\*, 0) NOT NULL   
 , PASSWORD VARCHAR2(20 BYTE) NOT NULL   
 , USERNAME VARCHAR2(50 BYTE) NOT NULL   
 , PHONE VARCHAR2(20 BYTE) NOT NULL   
 , ADDRESS VARCHAR2(30 BYTE) NOT NULL   
 , REGISTRATIONDATE DATE NOT NULL   
 , CART\_ID NUMBER(\*, 0) NOT NULL   
 , CONSTRAINT U\_PK PRIMARY KEY   
   (  
    ID   
   )  
   USING INDEX   
   (  
       CREATE UNIQUE INDEX U\_PK ON BOOKUSER (ID ASC)   
      LOGGING   
       TABLESPACE mc01   
      PCTFREE 10   
      INITRANS 2   
       STORAGE   
       (   
        BUFFER\_POOL DEFAULT   
       )   
      NOPARALLEL   
   )  
   ENABLE   
 )   
 TABLESPACE mc01   
 PCTFREE 10   
 INITRANS 1   
 STORAGE   
 (   
  BUFFER\_POOL DEFAULT   
 )   
 NOCOMPRESS   
 NOPARALLEL   
 PARTITION BY RANGE (REGISTRATIONDATE)   
 SUBPARTITION BY RANGE (REGISTRATIONDATE)   
 (  
   PARTITION DATE2018 VALUES LESS THAN (TO\_DATE(' 2018-12-31 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN'))   
   TABLESPACE mc01   
  PCTFREE 10   
  INITRANS 1   
   STORAGE   
   (   
    BUFFER\_POOL DEFAULT   
   )   
  NOCOMPRESS NO INMEMORY   
   (  
    SUBPARTITION DATE2018\_3 VALUES LESS THAN (TO\_DATE(' 2018-03-31 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN'))   
    NOCOMPRESS NO INMEMORY    
   , SUBPARTITION DATE2018\_6 VALUES LESS THAN (TO\_DATE(' 2018-06-30 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN'))   
    NOCOMPRESS NO INMEMORY    
   , SUBPARTITION DATE2018\_9 VALUES LESS THAN (TO\_DATE(' 2018-09-30 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN'))   
    NOCOMPRESS NO INMEMORY    
   , SUBPARTITION DATE2018\_12 VALUES LESS THAN (TO\_DATE(' 2018-12-31 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN'))   
    NOCOMPRESS NO INMEMORY    
   )    
 , PARTITION DATE2019 VALUES LESS THAN (TO\_DATE(' 2019-12-31 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN'))   
   TABLESPACE mc01   
  PCTFREE 10   
  INITRANS 1   
   STORAGE   
   (   
    BUFFER\_POOL DEFAULT   
   )   
  NOCOMPRESS NO INMEMORY   
   (  
    SUBPARTITION DATE2019\_3 VALUES LESS THAN (TO\_DATE(' 2019-03-31 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN'))   
    NOCOMPRESS NO INMEMORY    
   , SUBPARTITION DATE2019\_6 VALUES LESS THAN (TO\_DATE(' 2019-06-30 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN'))   
    NOCOMPRESS NO INMEMORY    
   , SUBPARTITION DATE2019\_9 VALUES LESS THAN (TO\_DATE(' 2019-09-30 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN'))   
    NOCOMPRESS NO INMEMORY    
   , SUBPARTITION DATE2019\_12 VALUES LESS THAN (TO\_DATE(' 2019-12-31 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN'))   
    NOCOMPRESS NO INMEMORY    
   )    
 );  
 ​

**6. 创建商品表**

 CREATE TABLE COMMODITY   
 (  
  ID NUMBER(\*, 0) NOT NULL   
 , PID NUMBER(\*, 0) NOT NULL   
 , BOOKSNAME VARCHAR2(20 BYTE) NOT NULL   
 , PRICE NUMBER NOT NULL   
 , DESCRIBE VARCHAR2(50 BYTE) NOT NULL   
 , NUM NUMBER(\*, 0) NOT NULL   
 , ADMIN\_ID NUMBER(\*, 0) NOT NULL   
 , CONSTRAINT COMMODITY\_PK PRIMARY KEY   
   (  
    ID   
   )  
   USING INDEX   
   (  
       CREATE UNIQUE INDEX COMMODITY\_PK ON COMMODITY (ID ASC)   
      LOGGING   
       TABLESPACE mc01   
      PCTFREE 10   
      INITRANS 2   
       STORAGE   
       (   
        BUFFER\_POOL DEFAULT   
       )   
      NOPARALLEL   
   )  
   ENABLE   
 )   
 LOGGING   
 TABLESPACE mc01   
 PCTFREE 10   
 INITRANS 1   
 STORAGE   
 (   
  BUFFER\_POOL DEFAULT   
 )   
 NOCOMPRESS   
 NO INMEMORY   
 NOPARALLEL;

**7. 创建购物车表**

* 用户表字段BOOKUSER\_ID为购物车表的外键
* 购物车采用引用分区

 CREATE TABLE CART   
 (  
  ID NUMBER(\*, 0) NOT NULL   
 , AMOUNT NUMBER(\*, 0) NOT NULL   
 , PID NUMBER(\*, 0) NOT NULL   
 , BOOKUSER\_ID NUMBER(\*, 0) NOT NULL   
 , CONSTRAINT CART\_PK PRIMARY KEY   
   (  
    ID   
   )  
   USING INDEX   
   (  
       CREATE UNIQUE INDEX CART\_PK ON CART (ID ASC)   
      LOGGING   
       TABLESPACE mc01   
      PCTFREE 10   
      INITRANS 2   
       STORAGE   
       (   
        BUFFER\_POOL DEFAULT   
       )   
      NOPARALLEL   
   )  
   ENABLE   
 , CONSTRAINT CART\_BOOKUSER FOREIGN KEY  
   (  
  BOOKUSER\_ID   
   )  
   REFERENCES BOOKUSER  
   (  
  CART\_ID   
   )  
   ENABLE   
 )   
 PCTFREE 10   
 PCTUSED 40   
 INITRANS 1   
 STORAGE   
 (   
  BUFFER\_POOL DEFAULT   
 )   
 NOCOMPRESS   
 NOPARALLEL   
 PARTITION BY REFERENCE (CART\_BOOKUSER)   
 (  
   PARTITION DATE2018\_3   
  LOGGING   
   TABLESPACE mc01   
  PCTFREE 10   
  INITRANS 1   
   STORAGE   
   (   
    BUFFER\_POOL DEFAULT   
   )   
  NOCOMPRESS NO INMEMORY    
 , PARTITION DATE2018\_6   
  LOGGING   
   TABLESPACE mc01   
  PCTFREE 10   
  INITRANS 1   
   STORAGE   
   (   
    BUFFER\_POOL DEFAULT   
   )   
  NOCOMPRESS NO INMEMORY    
 , PARTITION DATE2018\_9   
  LOGGING   
   TABLESPACE mc01   
  PCTFREE 10   
  INITRANS 1   
   STORAGE   
   (   
    BUFFER\_POOL DEFAULT   
   )   
  NOCOMPRESS NO INMEMORY    
 , PARTITION DATE2018\_12   
  LOGGING   
   TABLESPACE mc01   
  PCTFREE 10   
  INITRANS 1   
   STORAGE   
   (   
    BUFFER\_POOL DEFAULT   
   )   
  NOCOMPRESS NO INMEMORY    
 , PARTITION DATE2019\_3   
  LOGGING   
   TABLESPACE mc01   
  PCTFREE 10   
  INITRANS 1   
   STORAGE   
   (   
    BUFFER\_POOL DEFAULT   
   )   
  NOCOMPRESS NO INMEMORY    
 , PARTITION DATE2019\_6   
  LOGGING   
   TABLESPACE mc01   
  PCTFREE 10   
  INITRANS 1   
   STORAGE   
   (   
    BUFFER\_POOL DEFAULT   
   )   
  NOCOMPRESS NO INMEMORY    
 , PARTITION DATE2019\_9   
  LOGGING   
   TABLESPACE mc01   
  PCTFREE 10   
  INITRANS 1   
   STORAGE   
   (   
    BUFFER\_POOL DEFAULT   
   )   
  NOCOMPRESS NO INMEMORY    
 , PARTITION DATE2019\_12   
  LOGGING   
   TABLESPACE mc01   
  PCTFREE 10   
  INITRANS 1   
   STORAGE   
   (   
    BUFFER\_POOL DEFAULT   
   )   
  NOCOMPRESS NO INMEMORY    
 );

**8. 创建论坛表**

 CREATE TABLE TABLE1   
 (  
  ID INT NOT NULL   
 , CONTENT NVARCHAR2(50) NOT NULL   
 , CONSTRAINT TABLE1\_PK PRIMARY KEY   
   (  
    ID   
   )  
   ENABLE   
 );

**9. 创建视图计算每个用户购车中单个商品的总价**

 create or replace  view view\_SinglePriceSum  
 as   
 select b.id,b.username,co.booksname,(co.price\*ca.amount) pricesum from COMMODITY co,cart ca,BOOKUSER b where co.pid=ca.pid and ca.BOOKUSER\_ID =b.id;  
 select \* from view\_SinglePriceSum;  
 ​

**10. 插入用户，商品，购物车数据**

 declare  
  id number(38,0);  
  username varchar2(50);  
  phone varchar2(20);  
  address varchar2(30);  
  REGISTRATIONDATE date;  
  booksname varchar2(50);  
  price number(5,2);  
  num number(38,0);  
  amount number(38,0);  
     
 begin  
   for i in 1..50000  
   loop  
     if i mod 2 =0 then  
      REGISTRATIONDATE:=to\_date('2018-5-6','yyyy-mm-dd')+(i mod 60);  
     else  
      REGISTRATIONDATE:=to\_date('2019-5-6','yyyy-mm-dd')+(i mod 60);  
     end if;  
 ​  
     --插入用户  
    id:=SEQ\_ORDER\_ID.nextval;  
    username := 'aa'|| 'aa';  
    username := 'wang' || i;  
    phone := '131785693' || i;  
  booksname := '唐诗三百首版本号' || i;  
  address :='成都'|| '四川';  
  price :=(dbms\_random.value() \* 100);  
  num :=(i mod 5);  
     insert /\*+append\*/ into bookuser (id,password,username,phone,address,REGISTRATIONDATE,cart\_id)  
       values (id,username,username,phone,address,REGISTRATIONDATE,id);  
  --插入货品  
    
  insert into commodity(id,pid,booksname,price,describe,num,admin\_id)  
  values (id,id,booksname,price,'good',num,1);  
  --插入购物车  
  amount :=(id mod 3 ) + 1;  
  insert into cart(id,amount,pid,bookuser\_id)  
  values (id,amount,id,id);  
 ​  
     IF I MOD 1000 =0 THEN  
       commit;   
     END IF;  
   end loop;  
    
 end;

**11. 创建程序包，存储过程**

**创建程序包**

* 函数getcartsumprice计算每个用户的购物车商品总金额
* 存储过程adduser插入用户信息

 create or replace PACKAGE book\_package Is  
    function getcartsumprice(user\_id number) return number;  
    procedure adduser(password varchar2,username varchar2,phone varchar2,address varchar2,registerdate VARCHAR2);  
 end book\_package;

**创建函数、存储过程**

 create or replace PACKAGE body book\_package Is  
    
        function getcartsumprice(user\_id number) return number as  
           begin  
             declare cart\_sum number;  
  query\_sql varchar2(200);  
             begin  
  query\_sql:='select sum(pricesum) from view\_SinglePriceSum where ID=' || user\_id;  
               execute immediate query\_sql into cart\_sum;  
   return cart\_sum;  
             end;  
         end getcartsumprice;  
                   procedure addUser(password varchar2,username varchar2,phone varchar2,address varchar2,registerdate varchar2) as  
             begin  
               declare maxId number;  
               begin  
                 select max(id) into maxId from bookuser;  
                 insert into bookuser values(maxId+1,password,username,phone,address,to\_date(registerdate,'yyyy-mm-dd'),maxId+1);  
                 commit;  
               end;  
             end adduser;  
     end book\_package;

使用存储过程adduser插入用户数据

 set serveroutput on  
 declare  
 begin  
 BOOK\_PACKAGE.addUser('131','cwd','125626','hongkong','2019-05-02');  
 end;

**12. 表空间使用状况**

 SELECT a.tablespace\_name "表空间名",  
 total "表空间大小",  
 free "表空间剩余大小",  
 (total - free) "表空间使用大小",  
 total / (1024 \* 1024 \* 1024) "表空间大小(G)",  
 free / (1024 \* 1024 \* 1024) "表空间剩余大小(G)",  
 (total - free) / (1024 \* 1024 \* 1024) "表空间使用大小(G)",  
 round((total - free) / total, 4) \* 100 "使用率 %"  
 FROM (SELECT tablespace\_name, SUM(bytes) free  
 FROM dba\_free\_space  
 GROUP BY tablespace\_name) a,  
 (SELECT tablespace\_name, SUM(bytes) total  
 FROM dba\_data\_files  
 GROUP BY tablespace\_name) b  
 WHERE a.tablespace\_name = b.tablespace\_name

**13. 备份**

**全库0级备份(只作一次)**

 run{  
 configure retention policy to redundancy 1;  
 configure controlfile autobackup on;  
 configure controlfile autobackup format for device type disk to '/home/student/rman\_backup/%F';  
 configure default device type to disk;  
 crosscheck backup;  
 crosscheck archivelog all;  
 allocate channel c1 device type disk;  
 allocate channel c2 device type disk;  
 allocate channel c3 device type disk;  
 backup incremental level 0 database format '/home/student/rman\_backup/level0\_%d\_%T\_%U.bak';  
 report obsolete;  
 delete noprompt obsolete;  
 delete noprompt expired backup;  
 delete noprompt expired archivelog all;  
 release channel c1;  
 release channel c2;  
 release channel c3;  
 }

**全库完全恢复**

* oracle登录linux,不是student用户,dedicated专用连接模式
* 需要全库停机，需要oracle用户
* sys登录到orcl，查看全库的数据文件

 $ sqlplus / as sysdba  
 SQL> select file\_name from dba\_data\_files;  
 ​  
 - 全库停机  
 $rman target /  
 RMAN> shutdown immediate; 或者 shutdown abort;  
 RMAN> exit  
 ​  
 - 数据文件改名，模拟文件损失  
 $mv /home/student/pdb\_ly/pdbtest\_users02\_1.dbf /home/student/pdb\_ly/pdbtest\_users02\_1.dbf2  
 ​  
 - 全库恢复  
 $rman target /  
 RMAN> startup mount;  
 RMAN> restore database;  
 RMAN> recover database;  
 RMAN> alter database open;