

**期末项目设计报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 题 目 | 基于Oracle的外卖系统的数据库设计 | | |
| 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 | 何浩文 | 学号 | 201810414111 |
| 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 | 10 |  |
| PL/SQL设计 | 存储过程和函数设计正确 | 25 |  |
| 备份方案 | 备份方案设计正确 | 25 |  |
| 容灾方案 | DataGuard设计正确 | 10 |  |
| **得分合计** | | |  |

2021 年 5 月 20 日

目录

[目录 2](#_Toc74608503)

[一、 书店图书销售系统 3](#_Toc74608504)

[1. 项目背景 3](#_Toc74608505)

[2. 创建数据库 3](#_Toc74608506)

[3. 创建表空间 3](#_Toc74608507)

[4. 创建用户与角色，并给用户授权 4](#_Toc74608508)

[5. 系统用表 5](#_Toc74608509)

[5. 数据初始化 14](#_Toc74608510)

[二、 实验总结 20](#_Toc74608511)

# 书店图书销售系统

## 项目背景

随着信息时代的到来，IT产业和Internet获得了飞速发展，计算机应用已渗透到了各个领域，网络应用越来越普遍，而且走进了千家万户，一直使用手工记账的小镇书店老板，也想购买一个图书销售系统，但想先体验一下计算机的快捷方便。因此，设计了如下测试用例，供其初步体验！

## 创建数据库

1. CREATE PLUGGABLE DATABASE cjhpdb admin user cjh IDENTIFIED by 123 file\_name\_convert=
2. ('/home/oracle/app/oracle/oradata/orcl/pdbseed','/home/oracle/app/oracle/oradata/orcl/cjhpdb');

## 创建表空间

切换pdb：

1. alter session set container=cjhpdb;



创建表空间：

1. CREATE TABLESPACE USERS02
2. DATAFILE
3. '/home/oracle/app/oracle/oradata/orcl/cjhpdb/pdbcjh\_users02\_1.dbf'
4. size 100M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED,
5. '/home/oracle/app/oracle/oradata/orcl/cjhpdb/pdbcjh\_users02\_2.dbf'
6. SIZE 100M AUTOEXTEND ON next 256M MAXSIZE UNLIMITED
7. EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;
8. CREATE TABLESPACE USERS03
9. DATAFILE
10. '/home/oracle/app/oracle/oradata/orcl/cjhpdb/pdbcjh\_users03\_1.dbf'
11. size 100M AUTOEXTEND ON NEXT 256M MAXSIZE UNLIMITED,
12. '/home/oracle/app/oracle/oradata/orcl/cjhpdb/pdbcjh\_users03\_2.dbf'
13. SIZE 100M AUTOEXTEND ON next 256M MAXSIZE UNLIMITED
14. EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;



## 创建用户与角色，并给用户授权

1. *-- 创建usercjh用户：*
2. CREATE USER usercjh IDENTIFIED BY 123
3. DEFAULT TABLESPACE "USERS02"
4. TEMPORARY TABLESPACE "TEMP";
5. *-- 给用户usercjh分配表空间：*
6. ALTER USER usercjh QUOTA UNLIMITED ON USERS02;
7. ALTER USER usercjh QUOTA UNLIMITED ON USERS03;
8. *-- 创建角色并给用户分配权限：*
9. CREATE ROLE cjhrole1;
10. GRANT connect,resource,CREATE VIEW TO cjhrole1;
11. GRANT cjhrole1 to usercjh;
12. CREATE ROLE cjhrole2;
13. GRANT dba TO cjhrole2;
14. GRANT cjhrole2 to usercjh;



## 系统用表

系统用表共四张，分别为：**藏书信息表(book)**、**进货表(stock)**、**出货表(sold)**、**用户表(users)**。

**藏书信息表**：用来存放书店现有图书的信息。

**进货表**：用来存放书店进货信息。

**出货表**：用来存放书店卖出的图书信息。

**用户表**：用来存放书店员工信息。

* 1. 藏书信息表(book)

藏书信息表(book) 表1

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 描述 |
| bookno | varchar2 | 6 | 图书编号，定义为主键 |
| bookname | varchar2 | 20 | 书名， 唯一 |
| species | varchar2 | 5 | 图书分类， 不允许为空 |
| bookprice | number | (7,2) | 图书单价， 不允许为空 |
| publish\_house | varchar2 | 20 | 出版社， 不允许为空 |
| publishdate | date |  | 出版日期， 不允许为空 |
| author | varchar2 | 10 | 作者， 不允许为空 |
| number | number | 4 | 藏书量， 大于0 |

创建BOOK表：

1. CREATE TABLE BOOK
2. (
3. BOOKNO NUMBER(20, 0) NOT NULL
4. , BOOKNAME VARCHAR2(20 BYTE)
5. , SPECIES VARCHAR2(5 BYTE) NOT NULL
6. , BOOKPRICE NUMBER(7, 2) NOT NULL
7. , PUBLISH\_HOUSE VARCHAR2(20 BYTE)
8. , PUBLISHDATE DATE
9. , AUTHOR VARCHAR2(10 BYTE) NOT NULL
10. , NUMBERS NUMBER
11. , CONSTRAINT SYS\_C0010008 PRIMARY KEY
12. (
13. BOOKNO
14. )
15. USING INDEX
16. (
17. CREATE UNIQUE INDEX SYS\_C0010008 ON BOOK (BOOKNO ASC)
18. LOGGING
19. TABLESPACE USERS02
20. PCTFREE 10
21. INITRANS 2
22. STORAGE
23. (
24. BUFFER\_POOL DEFAULT
25. )
26. NOPARALLEL
27. )
28. ENABLE
29. )
30. TABLESPACE USERS02
31. PCTFREE 10
32. INITRANS 1
33. STORAGE
34. (
35. BUFFER\_POOL DEFAULT
36. )
37. NOCOMPRESS
38. NOPARALLEL
39. PARTITION BY RANGE (PUBLISHDATE)
40. (
41. PARTITION P1 VALUES LESS THAN (TO\_DATE(' 2018-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN'))
42. LOGGING
43. TABLESPACE USERS02
44. PCTFREE 10
45. INITRANS 1
46. STORAGE
47. (
48. BUFFER\_POOL DEFAULT
49. )
50. NOCOMPRESS NO INMEMORY
51. , PARTITION P2 VALUES LESS THAN (TO\_DATE(' 2019-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN'))
52. LOGGING
53. TABLESPACE USERS02
54. PCTFREE 10
55. INITRANS 1
56. STORAGE
57. (
58. BUFFER\_POOL DEFAULT
59. )
60. NOCOMPRESS NO INMEMORY
61. , PARTITION P3 VALUES LESS THAN (TO\_DATE(' 2020-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS\_CALENDAR=GREGORIAN'))
62. LOGGING
63. TABLESPACE USERS03
64. PCTFREE 10
65. INITRANS 1
66. STORAGE
67. (
68. BUFFER\_POOL DEFAULT
69. )
70. NOCOMPRESS NO INMEMORY
71. , PARTITION P4 VALUES LESS THAN (MAXVALUE)
72. LOGGING
73. TABLESPACE USERS03
74. PCTFREE 10
75. INITRANS 1
76. STORAGE
77. (
78. BUFFER\_POOL DEFAULT
79. )
80. NOCOMPRESS NO INMEMORY
81. );
82. ALTER TABLE BOOK
83. ADD CONSTRAINT SYS\_C0010009 UNIQUE
84. (
85. BOOKNAME
86. )
87. USING INDEX
88. (
89. CREATE UNIQUE INDEX SYS\_C0010009 ON BOOK (BOOKNAME ASC)
90. LOGGING
91. TABLESPACE USERS02
92. PCTFREE 10
93. INITRANS 2
94. STORAGE
95. (
96. BUFFER\_POOL DEFAULT
97. )
98. NOPARALLEL
99. )
100. ENABLE;
101. ALTER TABLE BOOK
102. ADD CONSTRAINT SYS\_C0010007 CHECK
103. (numbers>=0)
104. ENABLE;

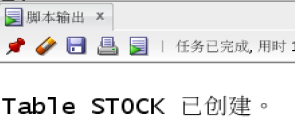
5.2、 进货表(stock)

进货表(stock) 表2

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 描述 |
| iono | varchar2 | 6 | 进货编号，定义为主键 |
| bookno | varchar2 | 6 | 图书编号，定义为外键 |
| numbers | number |  | 进货量， 大于等于0 |
| in\_price | number |  | 进价，不能为空 |
| in\_time | date |  | 进货时间，不能为空 |

建表语句：

1. CREATE TABLE STOCK
2. (
3. IONO VARCHAR2(6 BYTE) NOT NULL
4. , BOOKNO NUMBER(20, 0)
5. , NUMBERS NUMBER
6. , IN\_PRICE NUMBER NOT NULL
7. , IN\_TIME DATE NOT NULL
8. , CONSTRAINT SYS\_C0010018 PRIMARY KEY
9. (
10. IONO
11. )
12. USING INDEX
13. (
14. CREATE UNIQUE INDEX SYS\_C0010018 ON STOCK (IONO ASC)
15. LOGGING
16. TABLESPACE USERS02
17. PCTFREE 10
18. INITRANS 2
19. STORAGE
20. (
21. BUFFER\_POOL DEFAULT
22. )
23. NOPARALLEL
24. )
25. ENABLE
26. )
27. LOGGING
28. TABLESPACE USERS02
29. PCTFREE 10
30. INITRANS 1
31. STORAGE
32. (
33. BUFFER\_POOL DEFAULT
34. )
35. NOCOMPRESS
36. NO INMEMORY
37. NOPARALLEL;
38. ALTER TABLE STOCK
39. ADD CONSTRAINT FK\_STOCK FOREIGN KEY
40. (
41. BOOKNO
42. )
43. REFERENCES BOOK
44. (
45. BOOKNO
46. )
47. ENABLE;
48. ALTER TABLE STOCK
49. ADD CONSTRAINT SYS\_C0010017 CHECK
50. (numbers>=0)
51. ENABLE;



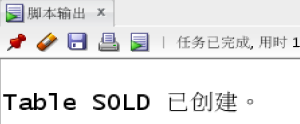
* 1. 出货表(sold)

出货表(sold) 表3

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 描述 |
| oono | varchar2 | 6 | 出货编号，定义为主键 |
| bookno | varchar2 | 6 | 图书编号，定义为外键 |
| numbers | number |  | 出货量， 大于等于0 |
| out\_price | number |  | 售卖价格，不能为空 |
| out\_time | date |  | 售卖时间，不能为空 |

建表语句：

1. CREATE TABLE SOLD
2. (
3. OONO VARCHAR2(6 BYTE) NOT NULL
4. , BOOKNO NUMBER(20, 0)
5. , NUMBERS NUMBER
6. , OUT\_PRICE NUMBER NOT NULL
7. , OUT\_TIME DATE NOT NULL
8. , CONSTRAINT SYS\_C0010013 PRIMARY KEY
9. (
10. OONO
11. )
12. USING INDEX
13. (
14. CREATE UNIQUE INDEX SYS\_C0010013 ON SOLD (OONO ASC)
15. LOGGING
16. TABLESPACE USERS02
17. PCTFREE 10
18. INITRANS 2
19. STORAGE
20. (
21. BUFFER\_POOL DEFAULT
22. )
23. NOPARALLEL
24. )
25. ENABLE
26. )
27. LOGGING
28. TABLESPACE USERS02
29. PCTFREE 10
30. INITRANS 1
31. STORAGE
32. (
33. BUFFER\_POOL DEFAULT
34. )
35. NOCOMPRESS
36. NO INMEMORY
37. NOPARALLEL;
38. ALTER TABLE SOLD
39. ADD CONSTRAINT FK\_SOLD FOREIGN KEY
40. (
41. BOOKNO
42. )
43. REFERENCES BOOK
44. (
45. BOOKNO
46. )
47. ENABLE;
48. ALTER TABLE SOLD
49. ADD CONSTRAINT SYS\_C0010012 CHECK
50. (numbers>=0)
51. ENABLE;



5.4、用户表(users)

用户表(users) 表4

|  |  |  |  |
| --- | --- | --- | --- |
| 列名 | 数据类型 | 长度 | 描述 |
| username | varchar2 | 10 | 用户名，定义为主键 |
| password | varchar2 | 15 | 密码， 不允许为空 |
| sal | number |  | 月薪，可以为空 |
| comm | number |  | 奖金，可以为空 |
| remark | varchar2 | 15 | 用户类型，允许为空 |

建表语句：

1. CREATE TABLE BO\_USER
2. (
3. BO\_USERNAME VARCHAR2(10 BYTE) PRIMARY KEY
4. , BO\_PASSWORD VARCHAR2(15 BYTE) NOT NULL
5. , SAL NUMBER
6. , COMM NUMBER
7. , REMARK VARCHAR2(15 BYTE)
8. )
9. LOGGING
10. TABLESPACE USERS02
11. PCTFREE 10
12. INITRANS 1
13. STORAGE
14. (
15. BUFFER\_POOL DEFAULT
16. )
17. NOCOMPRESS
18. NO INMEMORY
19. NOPARALLEL;

tablet表;

|  |  |  |  |
| --- | --- | --- | --- |
| T\_ID | NUMBER(10) | 10 | 主键 |
| T\_DATE | DATE |  | 记录日期 |

建表语句：

1. CREATE TABLE TABLET
2. (
3. T\_ID NUMBER(10, 0) PRIMARY KEY
4. , T\_DATE DATE
5. )
6. LOGGING
7. TABLESPACE USERS02
8. PCTFREE 10
9. INITRANS 1
10. STORAGE
11. (
12. BUFFER\_POOL DEFAULT
13. )
14. NOCOMPRESS
15. NO INMEMORY
16. NOPARALLEL;

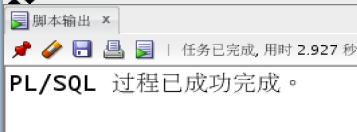


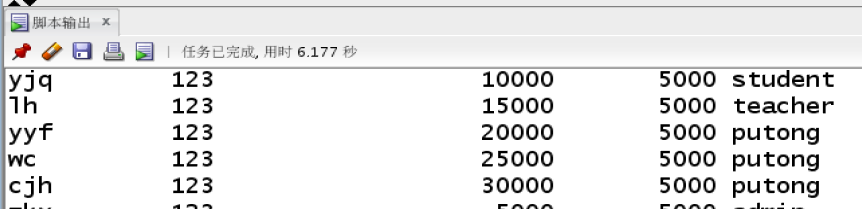
## 数据初始化

用户表插入数据

插入5万条数据：

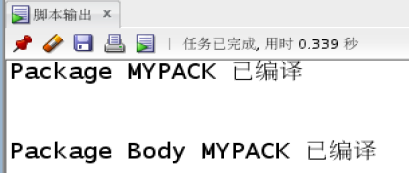
1. declare
2. u\_name VARCHAR2(10);
3. u\_pwd VARCHAR2(15);
4. u\_sal NUMBER;
5. u\_comm NUMBER;
6. u\_re VARCHAR2(15);
7. begin
8. */\**
9. system login:
10. ALTER USER "TEACHER" QUOTA UNLIMITED ON USERS;
11. ALTER USER "TEACHER" QUOTA UNLIMITED ON USERS02;
12. ALTER USER "TEACHER" QUOTA UNLIMITED ON USERS03;
13. \*/
14. */\*v\_order\_detail\_id:=1;*
15. delete from order\_details;
16. delete from orders;
17. \*/
18. for i in 1..50000
19. loop
20. *--    if i mod 6 =0 then*
21. *--      dt:=to\_date('2018-01-01','yyyy-mm-dd')+(i mod 60); --PARTITION\_2018*
22. *--    elsif i mod 6 =1 then*
23. *--      dt:=to\_date('2019-01-01','yyyy-mm-dd')+(i mod 60); --PARTITION\_2019*
24. *--    elsif i mod 6 =2 then*
25. *--      dt:=to\_date('2020-01-01','yyyy-mm-dd')+(i mod 60); --PARTITION\_2020*
26. *--    end if;*
27. */\*V\_EMPLOYEE\_ID:=CASE I MOD 6 WHEN 0 THEN 11 WHEN 1 THEN 111 WHEN 2 THEN 112*
28. WHEN 3 THEN 12 WHEN 4 THEN 121 ELSE 122 END;
29. \*/
30. *--插入订单*
31. u\_name:=CASE I MOD 6 WHEN 0 THEN 'zkx' WHEN 1 THEN 'yjq' WHEN 2 THEN 'lh'
32. WHEN 3 THEN 'yyf' WHEN 4 THEN 'wc' ELSE 'cjh' END;
33. u\_pwd:='123';
34. u\_sal:=CASE I MOD 6 WHEN 0 THEN 5000 WHEN 1 THEN 10000 WHEN 2 THEN 15000
35. WHEN 3 THEN 20000 WHEN 4 THEN 25000 ELSE 30000 END;
36. u\_comm:=5000;
37. u\_re:=CASE I MOD 6 WHEN 0 THEN 'admin' WHEN 1 THEN 'student' WHEN 2 THEN 'teacher'
38. ELSE 'putong' END;
39. insert */\*+append\*/* into BO\_USER (BO\_USERNAME,BO\_PASSWORD,SAL,COMM,REMARK)
40. values (u\_name,u\_pwd,u\_sal,u\_comm,u\_re);
41. *--插入订单y一个订单包括3个产品*
42. *--    v:=dbms\_random.value(10000,4000);*
43. *--    v\_name:='computer'|| (i mod 3 + 1);*
44. *--    insert /\*+append\*/ into ORDER\_DETAILS(ID,ORDER\_ID,PRODUCT\_NAME,PRODUCT\_NUM,PRODUCT\_PRICE)*
45. *--      values (v\_order\_detail\_id,v\_order\_id,v\_name,2,v);*
46. *--    v:=dbms\_random.value(1000,50);*
47. *--    v\_name:='paper'|| (i mod 3 + 1);*
48. *--    v\_order\_detail\_id:=v\_order\_detail\_id+1;*
49. *--    insert /\*+append\*/ into ORDER\_DETAILS(ID,ORDER\_ID,PRODUCT\_NAME,PRODUCT\_NUM,PRODUCT\_PRICE)*
50. *--      values (v\_order\_detail\_id,v\_order\_id,v\_name,3,v);*
51. *--    v:=dbms\_random.value(9000,2000);*
52. *--    v\_name:='phone'|| (i mod 3 + 1);*
53. *--*
54. *--    v\_order\_detail\_id:=v\_order\_detail\_id+1;*
55. *--    insert /\*+append\*/ into ORDER\_DETAILS(ID,ORDER\_ID,PRODUCT\_NAME,PRODUCT\_NUM,PRODUCT\_PRICE)*
56. *--      values (v\_order\_detail\_id,v\_order\_id,v\_name,1,v);*
57. *--    --在触发器关闭的情况下，需要手工计算每个订单的应收金额：*
58. *--    select sum(PRODUCT\_NUM\*PRODUCT\_PRICE) into m from ORDER\_DETAILS where ORDER\_ID=v\_order\_id;*
59. *--    if m is null then*
60. *--     m:=0;*
61. *--    end if;*
62. *--    UPDATE ORDERS SET TRADE\_RECEIVABLE = m - discount WHERE ORDER\_ID=v\_order\_id;*
63. *--    IF I MOD 1000 =0 THEN*
64. *--      commit; --每次提交会加快插入数据的速度*
65. *--    END IF;*
66. end loop;
67. end;





创建程序包

1. create or replace PACKAGE MyPack IS
2. */\**
3. 包MyPack中有：
4. 一个函数:Get\_Count，
5. 一个过程:Get\_UserType 统计每个用户类型的总数
6. \*/
7. FUNCTION Get\_Count RETURN NUMBER;
8. PROCEDURE Get\_UserType;
9. END MyPack;
10. create or replace PACKAGE BODY MyPack IS
11. FUNCTION Get\_Count RETURN NUMBER
12. AS
13. N NUMBER(20,2);
14. BEGIN
15. SELECT COUNT(\*) into N  FROM BO\_USER;
16. RETURN N;
17. END;
18. PROCEDURE Get\_UserType
19. AS
20. LEFTSPACE VARCHAR(2000);
21. begin
22. *--通过LEVEL判断递归的级别*
23. LEFTSPACE:=' ';
24. *--使用游标*
25. for v in
26. (SELECT REMARK,SUM(REMARK) U\_SUM FROM BO\_USER
27. GROUP BY REMARK)
28. LOOP
29. DBMS\_OUTPUT.PUT\_LINE(LPAD(LEFTSPACE,(1+1),' ')||
30. V.REMARK||' '||v.U\_SUM);
31. END LOOP;
32. END;
33. END MyPack;



测试Get\_Count函数：

SELECT MyPack.Get\_Count as 总记录数 from dual;

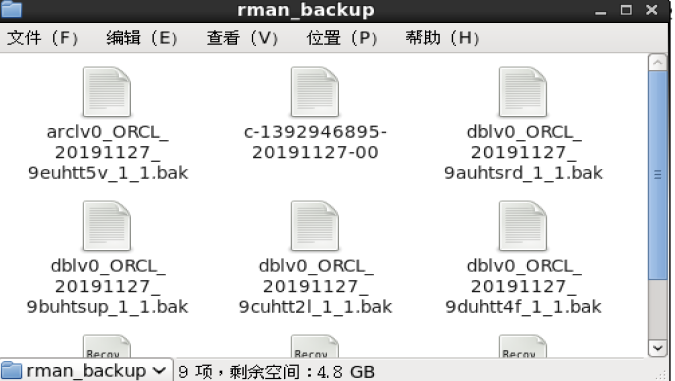


备份

1. 编写脚本文件rman\_cjh\_level0.sh
2. *#rman\_level0.sh*
3. #!/bin/sh
4. export NLS\_LANG='SIMPLIFIED CHINESE\_CHINA.AL32UTF8'
5. export ORACLE\_HOME=/home/oracle/app/oracle/product/12.1.0/dbhome\_1
6. export ORACLE\_SID=orcl
7. export PATH=$ORACLE\_HOME/bin:$PATH
8. rman target / nocatalog msglog=/home/oracle/rman\_backup/lv0\_`date +%Y%m%d-%H%M%S`\_L0.log << EOF
9. run{
10. configure retention policy to redundancy 1;
11. configure controlfile autobackup on;
12. configure controlfile autobackup format for device type disk to '/home/oracle/rman\_backup/%F';
13. configure default device type to disk;
14. crosscheck backup;
15. crosscheck archivelog all;
16. allocate channel c1 device type disk;
17. backup as compressed backupset incremental level 0 database format '/home/oracle/rman\_backup/dblv0\_%d\_%T\_%U.bak'
18. plus archivelog format '/home/oracle/rman\_backup/arclv0\_%d\_%T\_%U.bak';
19. report obsolete;
20. delete noprompt obsolete;
21. delete noprompt expired backup;
22. delete noprompt expired archivelog all;
23. release channel c1;
24. }
25. EOF
26. Exit

开始备份：

1. [oracle@oracle-pc ~]$ chmod 755 rman\_cjh\_level0.sh
2. [oracle@oracle-pc ~]$ ./rman\_
3. bash: ./rman\_: 没有那个文件或目录
4. [oracle@oracle-pc ~]$ ./rman\_cjh\_level0.sh
5. RMAN> 2> 3> 4> 5> 6> 7> 8> 9> 10> 11> 12> 13> 14> 15> 16>  RMAN> [oracle@oracle-pc ~]$
6. [oracle@oracle-pc ~]$



# 实验总结

完成该oracle数据实验基本掌握了数据库的建表操作，并能够创建出符合要求的表，将表与表之间建立联系。并且能够建立出表的视图，方便对图的查看。在进行备份和容灾操作的过程中，能够使用linux命令对数据库进行操作，极大地熟悉了linux的操作，同时也对oracle数据库的操作有了一定的了解。备份与容灾的操作是非常重要的，防止在数据库发生错误的时候，造成数据丢失，在进行这一实验过程的时候，利用了两台虚拟机，分别对主库与备库进行操作，在完成备份之后还进行了模拟删除数据，并成功恢复了数据。

同时在操作的过程中也应该更加认真仔细，在出现问题的时候应该要具备解决问题的能力，例如应该分析错误提示，找到错误的原因，并通过查询资料解决该问题。因为在数据库备份的过程中，等待时间较长，我们也应该耐心等待，不能够进行其他不利于数据库备份的操作。

在这次的实验过后，熟悉了oracle的基本命令，但是在oracle的学习中还远远不够，所以在学习oracle数据库的路程中还任重道远。