

# 实验一 Oracle 数据库的安装与部署

## 一、实验目的

1. 掌握 oracle 数据库安装和配置的方法
2. 掌握 SQL\*PLUS 工具及其常用命令
3. 熟悉常用的数据字典，能够借助数据字典获得相关 oracle 数据库信息。
4. 了解 OEM 管理器的各项功能

## 二、实验环境

- 1、硬件设备：计算机局域网，服务器 1 台，终端计算机每人一台
- 2、软件系统： Windows 操作系统， Oracle11g 数据库系统。

## 三、实验内容

### (一) Oracle 数据库安装与配置

同学们，请根据自己电脑的操作系统是 32 位或 64 位，到官网下载相应的安装程序 (QQ 群提供下载)，同时看看磁盘空间、内存等是否满足安装条件。

#### 1. Oracle11g 安装

采用的是基于 Java 引擎的 Oracle Universal Installer 安装工具自动安装。分为固定 IP 服务器、无固定 IP 的个人工作站安装两种情况，一般要求在分配有固定 IP 地址的计算机上进行。为实验方便密码设置为 **orcl**。(Oracle11g 建议密码：必须是大写字母加小写字母加数字，而且必须是 8 位以上。)

- (1) 固定 IP 安装模式：略，按照默认方式安装即可。
- (2) 动态 IP 的笔记本电脑安装：首先安装微软的环回适配器 Microsoft Loopback Adapter (win10 为 Microsoft KM-TEST)，并将其作为住适配器，设置一个固定私有 IP，例如：**192.168.1.100**，然后再进行正常安装，如图所示。然后进行正常安装即可。



- (3) 安装完毕，查看控制面板服务是否起来。

Network List Service	识别计算机已连接的网络，收集和存储这些网络的属性，并在更改这...	已启动	手动	本地服务
Network Location Awareness	收集和存储网络的配置信息，并在此信息被修改时向程序发出通知。...	已启动	自动	网络服务
Network Store Interface Service	此服务向用户模式客户端发送网络通知(例如，添加/删除接口等)。...	已启动	自动	本地服务
Office Software Protection Platform	Enables the download, installation, and enforcement of digital...	已启动	手动	网络服务
OracleServiceXE		已启动	手动	本地系统
OracleXETNSListener		已启动	手动	本地系统
Plug and Play	使计算机在极少或没有用户输入的情况下能识别并适应硬件的更改。...	已启动	自动	本地系统
Portable Device Enumerator Service	强制可移动大容量存储设备的组策略。使应用程序(如 Windows Me...	已启动	手动	本地系统
Power	管理电源策略和电源策略通知传递。	已启动	自动	本地系统
Print Spooler	将文件加载到内存供稍后打印	已启动	自动	本地系统
QPCore Service	腾讯安全服务	已启动	自动	本地系统

## 2. Oracle 数据库 12C 安装

(1) 执行安装目录中 setup，加载设置驱动程序，安装程序要求提供您的电子邮件地址以获取最新的安全问题和更新，可以单击下一步按钮跳过。



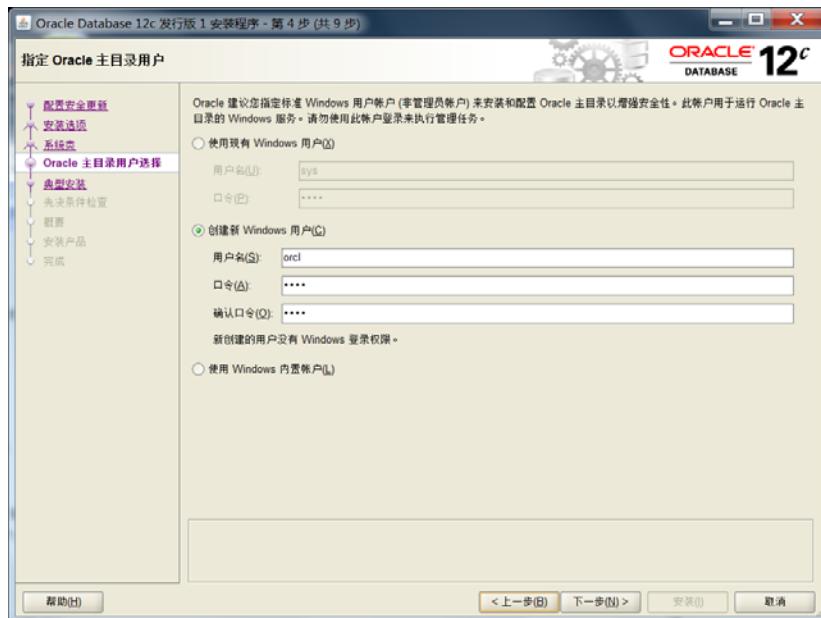
(2) Oracle 安装程序询问是要创建和配置数据库，仅安装数据库软件还是仅升级现有数据库。如果首次安装 Oracle 数据库，请选择第一个选项 (Create and configure a database)，然后单击下一步按钮。



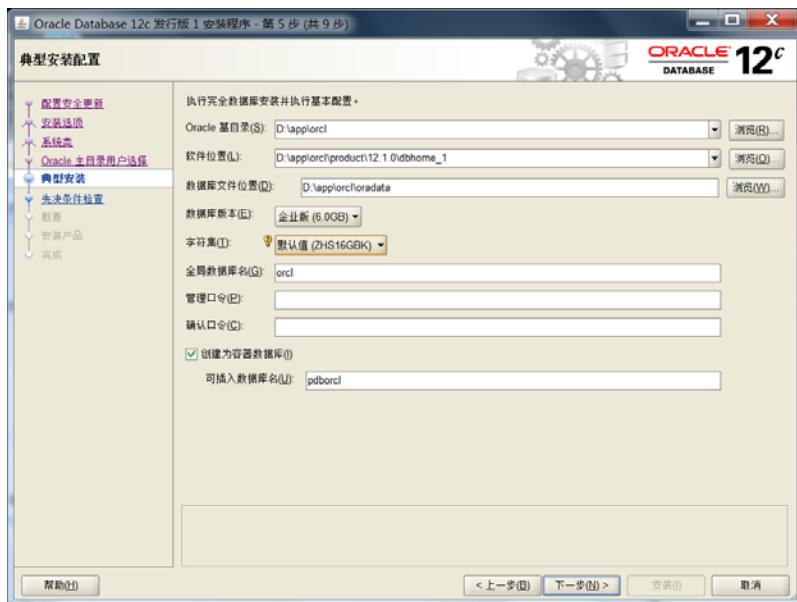
(3) 安装程序允许您选择系统类型。因为我们实验教学是在桌面计算机上安装 Oracle，而不是服务器，因此，这里选择第一个选项：桌面类(Desktop class)，然后单击下一步按钮。



(4) 在这一步骤中，允许指定要安装和配置 Oracle Home 以增强安全性的 Windows 用户帐户。选择第三个选项：“使用 Windows 内置帐户”。



(5) 选择要安装 Oracle 数据库的文件夹，全局数据库名称和密码，可插拔数据库名称。



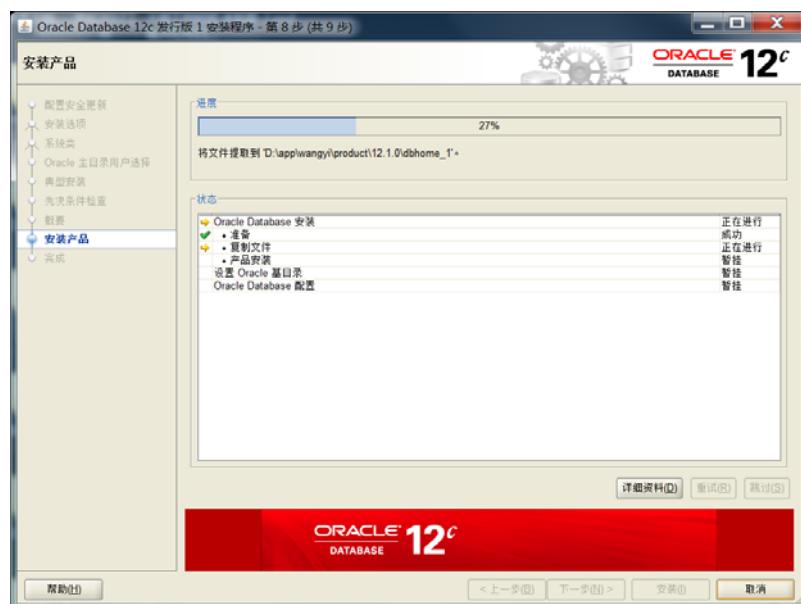
(6) 安装程序执行前提条件检查。



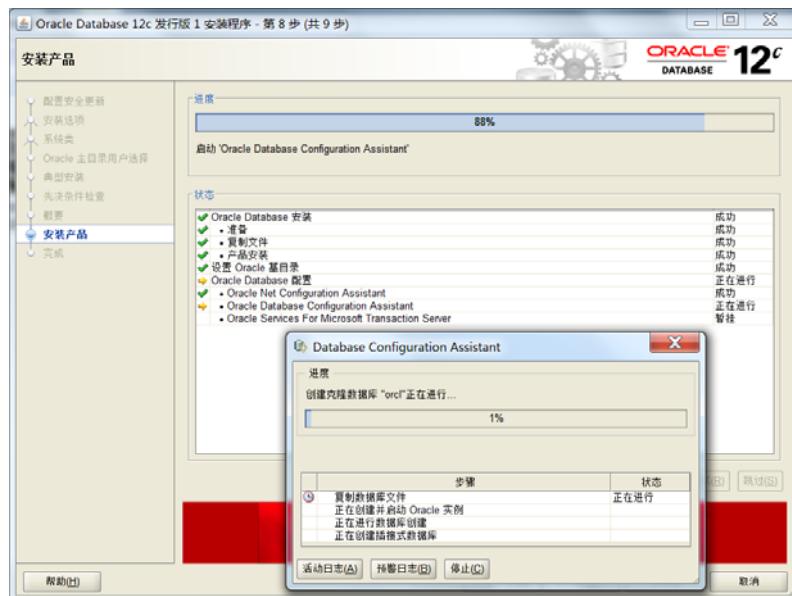
(7) 显示信息的摘要，例如全局设置，数据库信息等。如果一切正常，可查看信息，然后单击安装按钮。



(8) 安装程序开始安装 Oracle 数据库。安装速度取决于您的计算机性能，需要几分钟的时间才能完成。



安装进展接近结束时，系统弹出数据库配置助手窗口。



可设置 system、sys 管理账户的口令。



(9) 单击确定, 完成安装。检查一下控制面板中 oracle 实例、监听器是否启动正常,



**OracleServiceORCL:** 数据库实例服务，是 Oracle 核心服务该服务，是数据库启动的基础 orcl 是 Oracle 的实例标识，默认的实例为 orcl。

**OracleOraDb11g\_home1TNSListener:** 监听器服务，服务只有在数据库需要远程访问的时候才需要。（通过客户端访问需启动）。

**OracleDBConsoleorcl:** Oracle 数据库控制台服务，运行企业管理器 OEM 需要启动这个服务。

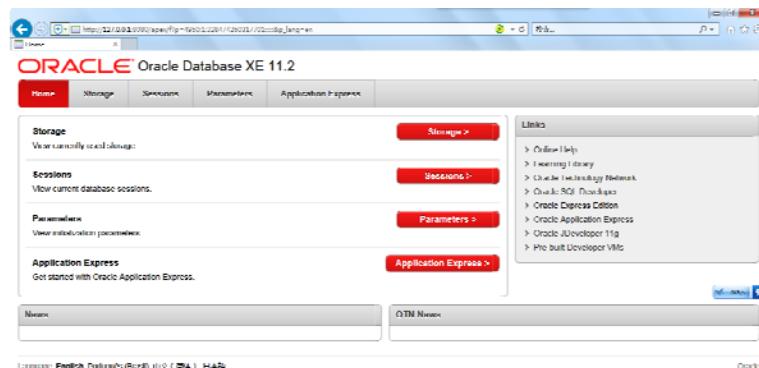
**Oracle ORCL VSS Writer Service:** Oracle 卷映射拷贝写入服务，能够让存储基础设施（比如磁盘，阵列等）创建高保真的时间点映像。

**OracleJobSchedulerORCL:** Oracle 作业调度（定时器）服务，ORCL 是 Oracle 实例标识。

**OracleMTSRecoveryService:** 服务端控制。该服务允许数据库充当一个微软事务服务器 MTS、COM/COM+对象和分布式环境下的事务的资源管理器。

**OracleOraDb11g\_home1ClrAgent:** OEM 代理服务，接收和响应来自 OEM 控制台的任务和事件请求，只有使用 OEM 管理数据库时才需要。

(10) 通过 OEM 熟悉 Oracle 的管理界面，熟悉常用的表空间、系统中各类用户以及各种操作性能检测、管理界面。

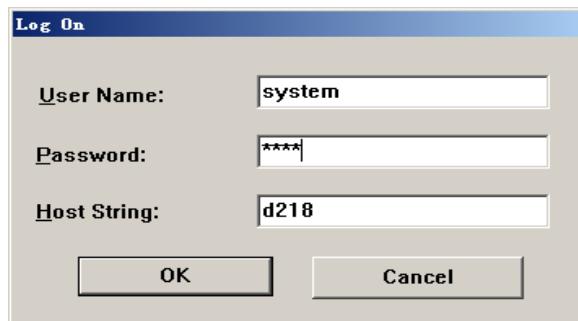


## (二) SQL\*PLUS 的使用

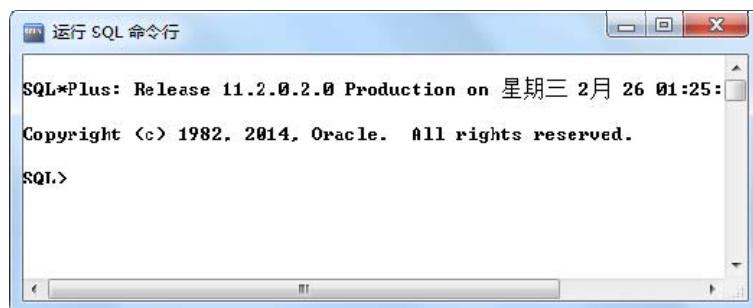
SQL Plus 是一个基于命令行的数据库操作工具。

### 1. 进入（启动）SQL\*PLUS

在 Windows 环境下, 双击 SQL\*PLUS 图标或从程序组找 SQL\*PLUS , 出现登录窗口, 如图所示。



输入正确的数据库用户名、密码和连接字符串后, 本次实验可采用 scott 为用户名, tiger 为密码, 图中 d218 为主机字符串 (SID 名)。若连接数据库成功, 则会出现如下提示符: SQL>, 如图。



2. 退出 SQL\*PLUS 是 SQL>exit, 则退回到 Windows 桌面。
3. 掌握不同用户登录 oracle 系统的方式: 在 windows 操作系统桌面点击[开始]后, 在运行中填写 sqlplus, 然后在窗口中键入以下语句, 以不同方式连接 Oracle:

```
conn scott/tiger      //普通用户
conn sys/orcl as sysdba    //sys 用户必须带上 as sysdba
conn system/orcl
```

## (三) PL/SQL 集成环境使用

1. 启动 PL/SQL Developer, 使用用户名和密码登录, 用户名 scott, 密码 tiger, 数据库为 orcl, 登陆连接方式为 normal, 见图 1。(如果是 sys 用户, 连接方式为 sysdba)



2. 登陆后，点击[文件]菜单项，打开[新建]，出现图 2，选择[Command Windows]项，出现对话框，见图 3.

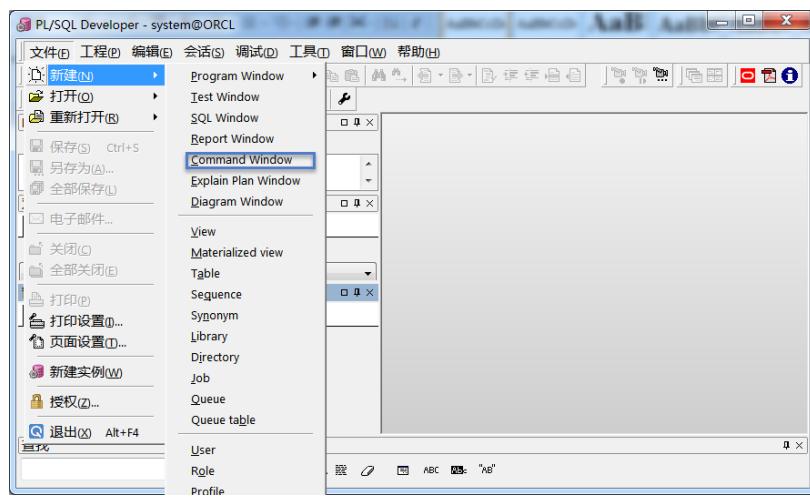


图 2

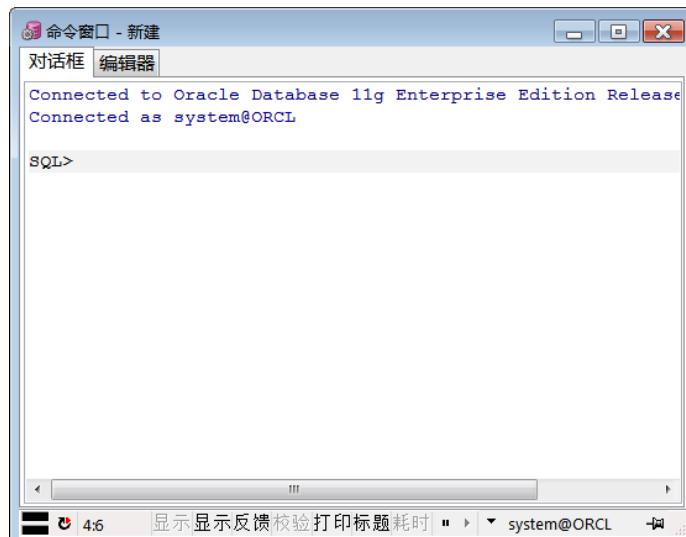


图 3

3. 本实验操作内容可在命令窗口内操作。

#### 4. 在 sqlplus 环境下熟悉常见的命令

(1) 显示和设置环境变量，可以用来控制输出的各种格式

linesize :设置显示行的宽，默认是 80 个自己字符

sql>show linesize

sql>set linesize 120

pagesize :设置每页显示的行数默认是 14，用法同 linesize

(2) 交互式命令 & :可以替代变量，而该变量在执行时需要用户输入

sql>select \* from emp where job='&job';

oracle 会提示用户输入值

(3) 查看帮助

help index

(4) 查看对象信息

describe 命令

例如 desc emp desc scott.emp 查看表结构

(5) 其他基本操作

熟练使用 save、get 和 start 和@保存和运行 sql 脚本、运用 edit 编辑。

sql>start mypro.sql;

sql>@d:\test\mypro.sql

(6) show 命令，显示当前环境变量信息

Show all 全部 68 个系统变量

Show user Show release Show SGA Show parameters Show errors

(7) 使用 select 语句熟悉 scott 模式下四个表的内容，掌握之间的关系。

实例恢复。

## 实验二 SQL\*PLUS 及 PL/SQL Developer

### 一、实验目的

1. 掌握 SQL\*PLUS 及 PL/SQL Developer 工具的使用方法；
2. 掌握常用的 SQL\*PLUS 命令；
3. 掌握常见的数据字典使用。

### 二、实验环境

- 1、硬件设备：计算机局域网，服务器 1 台，终端计算机每人一台
- 2、软件系统： Windows 2000/XP 客户机操作系统；  
Oracle11g 服务端数据库系统，客户端工具。

### 三、实验内容

#### (一) SQL\*PLUS 工具的使用

1. startup 启动的阶段模式：
  - nomount：启动实例(准备内存、进程)；
  - mount：打开控制文件；
  - open：从控制文件中找到数据文件和日志文件等；
2. shutdown 关闭四种方式
  - abort：直接关闭，不安全模式
  - immediate：保证会话完成的关闭
  - transactional：保证事务完成的关闭方式
  - normal：最安全的关闭方式
3. 运行 SQLPLUS 工具，在开始菜单的运行中敲入“*cmd*”，在打开的窗口执行 *sqlplus*。登录账户名和密码，**登录密码是不显示**，光标不移位。

```
C:\Users\Administrator>sqlplus
SQL*Plus: Release 11.2.0.2.0 Production on 星期三 3月 10 08:44:03 2021
Copyright <c> 1982, 2014, Oracle. All rights reserved.

请输入用户名: system
输入口令: 
连接到:
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production
SQL>
```

4. 以默认操作系统用户身份连接数据库，as sysdba 后缀表示数据库管理员权限。

*conn / as sysdba;*

请尝试试着采用带密码方式，看看是否起作用。

5. 显示当前用户名：*show user*；

```
SQL> conn scott/orcl;
已连接。
SQL> show user
USER 为 "SCOTT"
```

6. 显示有用户名和帐户的状态: `select username,account_status from dba_users;` 可以看到有锁定[LOCKED]、有的过期锁定[EXPIRED&LOCKED]

USERNAME	ACCOUNT_STATUS
TUSTER	OPEN
BOOKSTORE	OPEN
BOOKUSER	OPEN
SYS	OPEN
SYSTEM	OPEN
ANONYMOUS	OPEN
VY	OPEN
STU	OPEN
SCOTT	OPEN
WANGYI	OPEN
FLOWS_FILES	LOCKED
APEX_040000	LOCKED
APEX_PUBLIC_USER	LOCKED
OUTLN	EXPIRED & LOCKED
MDSYS	EXPIRED & LOCKED
CTXSYS	EXPIRED & LOCKED
DBSNMP	EXPIRED & LOCKED
XDB	EXPIRED & LOCKED
APPQOSSYS	EXPIRED & LOCKED
HR	EXPIRED & LOCKED
DIP	EXPIRED & LOCKED
ORACLE_OCM	EXPIRED & LOCKED
X\$NULL	EXPIRED & LOCKED

已选择23行。

7. 切换身份到某用户, 例如 scott 用户: `conn scott/tiger;`  
 8. 显示当前用户有哪些表: `select * from tab;` 注意 system 和 scott 用户看到表数量不同。

TNAME	TABTYPE	CLUSTERID
ADVISOR	TABLE	
BINS?p1d8kZfQSukAZ?C0rq1CA==\$0	TABLE	
BONUS	TABLE	
CLASSROOM	TABLE	
COURSE	TABLE	
DEPARTMENT	TABLE	
DEPT	TABLE	
EMP	TABLE	
EMP_HISTORY	TABLE	
INSTRUCTOR	TABLE	
MLDN_TAB	TABLE	
PREREQ	TABLE	
SALGRADE	TABLE	
SECTION	TABLE	
STUDENT	TABLE	
TAKES	TABLE	
TEACHES	TABLE	
TIME_SLOT	TABLE	
TRIGGER_DATA	TABLE	

已选择19行。

9. 将 SCOTT 帐号解锁/加锁: `alter user scott account unlock(lock);` 应以 system 用户操作

```
SQL> alter user scott account lock;
用户已更改。

SQL> conn scott/orcl;
ERROR:
ORA-28000: the account is locked

警告: 您不再连接到 ORACLE。
```

10. 查看 EMP 的表结构及记录内容: `desc emp;`

`select empno,ename from emp;`

名称	是否为空?	类型
EMPNO	NOT NULL	NUMBER(4)
ENAME		VARCHAR2(10)
JOB		VARCHAR2(9)
MGR		NUMBER(4)
HIREDATE		DATE
SAL		NUMBER(7,2)
COMM		NUMBER(7,2)
DEPTNO		NUMBER(2)

11. 在 windows 下以 administrator 身份查看 SGA:

`connect / as sysdba`

`show sga` (不支持 PL/SQL Developer)

SQL> connect / as sysdba
已连接。
SQL> show sga
Total System Global Area 1071333376 bytes
Fixed Size 1375792 bytes
Variable Size 645923280 bytes
Database Buffers 419430400 bytes
Redo Buffers 4603904 bytes

使用数据字典查询: `select name,value/1024/1024 from v$sga;`

查看共享池的信息: `show parameter shared_pool_size;`

SQL> show parameter shared_pool_size
NAME TYPE VALUE
shared_pool_size big integer 0

`select value/1024/1024 from v$parameter where name ='shared_pool_size';`

查看 CACHE 信息: `show parameter db_cache_size`

`select value/1024/1024 from v$parameter where name ='db_cache_size';`

`db_cache_size` 等 SGA 中的参数为 0, 意味着是自动管理, 动态分配内存大小

12. SQL\*PLUS 中的运行命令。

导入 scott. sql, 运行命令: `@d:\scott.sql`; 路径为自己定义

相同的命令有: strat、run 和/

- `start`: 运行指定文件, 直接显示结果;
- `run`: 运行缓冲区内的内容;
- `/`: 执行已经使用过的内容

13. spool: 将 SQL\*Plus 中的输出结果复制到一个指定的文件中。

请尝试在 SQL\*plus 中设置 spool on; 操作一段时间然后在设置为 spool off; 查看输出文件 wangyi.txt:

`spool d:\wangyi.txt;`

14. 格式化查询命令:

(1) column: 9 代表任意一个数字字符, \$ 为美元符号, L 显示本地区域货币符号, A 表示字符, 后面接长度。

- 将 emp 表 ename 字段以 15 字符宽度显示: `col ename format A15;`
  - 将 enmae 字段改为中文: `col ename heading 姓名;`
  - 将 sal 字段以本地货币符号显示: `col sal format L9999.9`
- (2) `pagesize`: 设置页面大小, 默认 14: `set pagesize 30`
- (3) `linesize`: 设置列宽度, 默认 80: `set linesize 200`
15. 显示执行时间: `set timing on/off`
16. 设置显示校验: `set verify on/off`
17. 临时变量&: `select * from emp where ename = &var;`
- 18: 注释语句:
  - `/* */`: 一段范围注释
  - `Remark:` 一行注释
19. 将上述操作在 PL/SQL Develop 中操作一遍熟悉环境, 注意权限问题。

## (二) 常见数据字典

常用数据字典请大家在实验课过程中不断积累掌握, 对于 oracle 体系结构相关的数据字典主要有以下几个。

数据文件: `dba_data_files` 查看对应数据文件的信息

`v$datafile`: 记录数据文件的动态信息

控制文件: `v$controlfile`;

日志文件: `v$log`

实例信息: `v$instance`

提示: 以 `dba_` 开头的数据字典名称记录数据库实例的所有对象信息。如 `dba_tables`。  
 以 `all_` 开头的数据字典记录用户对象的信息和被授权访问的对象信息。  
 以 `user_` 开头的数据字典名称记录用户对象信息, 如 `user_tables`。

1. `v$instance`: 查看当前数据库实例, 查询当前实例, 出现以下提示

```
SQL> select instance_name from v$instance;
select instance_name from v$instance
```

`ORA-00942: 表或视图不存在`

为什么? 采用 scott 登陆, 权限不够。

改用 system 用户登录, 继续使用上述 select 查询, oracle 11g express 数据库得到当前实例 xe。

SQL> select instance_name from v\$instance;
INSTANCE_NAME
-----
xe
SQL>

可以使用 show parameter service\_names; 查看当前实例，但必须具备相应的权限。

SQL> show parameter service_names		
NAME	TYPE	VALUE
service_names	string	XE

## 2. show parameter sga: 查看当前 SGA 设置

SQL> show parameter sga;		
NAME	TYPE	VALUE
lock_sga	boolean	FALSE
pre_page_sga	boolean	FALSE
sga_max_size	big integer	1G
sga_target	big integer	0

## 3. dba\_data\_files: 查看数据库中所有数据文件的名称位置, 所属表空间, 以及大小。

FILE_NAME	FILE_ID	TABLESPACE_NAME	BYTES	BLOCKS	STATUS	RELATIVE_FNO	AUTOEXTENSIBLE	MAXBYTES
D:\APP\ADMINISTRATOR\ORADATA\ORCL\USERS01.DBF	4	USERS	5242880	640	AVAILABLE	4	YES	3435972198
D:\APP\ADMINISTRATOR\ORADATA\ORCL\UNDOTBS01.DBF	3	UNDOTBS1	104857600	12800	AVAILABLE	3	YES	3435972198
D:\APP\ADMINISTRATOR\ORADATA\ORCL\SYSAUX01.DBF	2	SYSAUX	650117120	79360	AVAILABLE	2	YES	3435972198
D:\APP\ADMINISTRATOR\ORADATA\ORCL\SYSTEM01.DBF	1	SYSTEM	734003200	89600	AVAILABLE	1	YES	3435972198
D:\APP\ADMINISTRATOR\ORADATA\ORCL\EXAMPLE01.DBF	5	EXAMPLE	104857600	12800	AVAILABLE	5	YES	3435972198
D:\APP\ADMINISTRATOR\ORADATA\ORCL\BOOKSPACE.DBF	6	BOOKSPACE	52428800	6400	AVAILABLE	6	YES	104857600

MAXBLOCKS	INCREMENT_BY	USER_BYTES	USER_BLOCKS	ONLINE_STATUS
4194302	160	4194304	512	ONLINE
4194302	640	103809024	12672	ONLINE
4194302	1280	649068544	79232	ONLINE
4194302	1280	732954624	89472	SYSTEM
4194302	80	103809024	12672	ONLINE
12800	640	51380224	6272	ONLINE

上面显示所有字段，可以只显示感兴趣字段。

Desc dba\_data\_files: 查看表结构如下。

Name	Type	Nullable	Default	Comments
FILE_NAME	VARCHAR2(513)	Y		Name of the database data file
FILE_ID	NUMBER	Y		ID of the database data file
TABLESPACE_NAME	VARCHAR2(30)	Y		Name of the tablespace to which the file belongs
BYTES	NUMBER	Y		Size of the file in bytes
BLOCKS	NUMBER	Y		Size of the file in ORACLE blocks
STATUS	VARCHAR2(9)	Y		File status: "INVALID" or "AVAILABLE"
RELATIVE_FNO	NUMBER	Y		Tablespace-relative file number
AUTOEXTENSIBLE	VARCHAR2(3)	Y		Autoextensible indicator: "YES" or "NO"
MAXBYTES	NUMBER	Y		Maximum size of the file in bytes
MAXBLOCKS	NUMBER	Y		Maximum size of the file in ORACLE blocks
INCREMENT_BY	NUMBER	Y		Default increment for autoextension
USER_BYTES	NUMBER	Y		Size of the useful portion of file in bytes
USER_BLOCKS	NUMBER	Y		Size of the useful portion of file in ORACLE blocks
ONLINE_STATUS	VARCHAR2(7)	Y		Online status of the file

## 4. v\$controlfile: 查看控制文件，SELECT name FROM v\$controlfile; express 版默认一个控制文件，企业版则有多个，分别如下图。

SQL> SELECT name FROM v\$controlfile;	
NAME	
C:\ORACLEXE\APP\ORACLE\ORADATA\XE\CONTROL.DBF	

```
SQL> SELECT name FROM v$controlfile;
NAME
-----
D:\APP\ADMINISTRATOR\ORADATA\ORCL\CONTROL01.CTL
D:\APP\ADMINISTRATOR\FLASH_RECOVERY_AREA\ORCL\CONTROL02.CTL
```

### 5. v\$logfile: 查看日志文件信息, select \* from v\$logfile;

```
SQL> col member format A80
SQL> select group#,status,type,member,is_recovery_dest_file from v$logfile;
GROUP# STATUS TYPE MEMBER IS_
----- -----
2 ONLINE C:\ORACLEXE\APP\ORACLE\FAST_RECOVERY_AREA\XE\ONLINELOG\01_MF_2_G8DWG9Z0_.LOG YES
1 ONLINE C:\ORACLEXE\APP\ORACLE\FAST_RECOVERY_AREA\XE\ONLINELOG\01_MF_1_G8DWG9TQ_.LOG YES
```

Member 字段太长, 可通过格式化命令缩短显示。

对于安装 oracle 企业版同学, 查询日志文件显示不同, 如下。

```
SQL> select * from v$logfile;
GROUP# STATUS TYPE MEMBER IS_RECOVERY_DEST_FILE
----- -----
3 ONLINE D:\APP\ADMINISTRATOR\ORADATA\ORCL\REDO03.LOG NO
2 ONLINE D:\APP\ADMINISTRATOR\ORADATA\ORCL\REDO02.LOG NO
1 ONLINE D:\APP\ADMINISTRATOR\ORADATA\ORCL\REDO01.LOG NO
```

可用 v\$log 查看重做日志文件的结构、属性以及动态变化情况

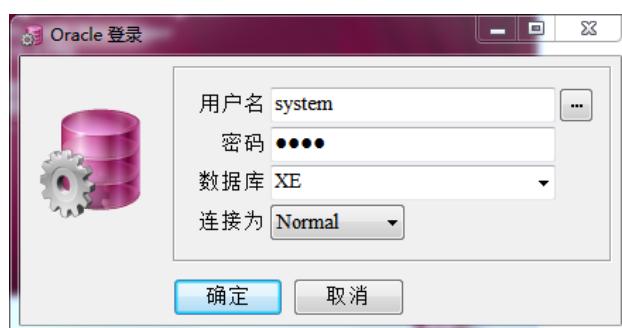
```
SQL> select * from v$log;
GROUP# THREAD# SEQUENCE# BYTES BLOCKSIZE MEMBERS ARCHIVED STATUS FIRST_CHANGE# FIRST_TIME NEXT_CHANGE# NEXT_TIME
----- -----
1 1 94 52428800 512 1 NO INACTIVE 3033405 2016/3/19 1 3064990 2016/3/21 2
2 1 95 52428800 512 1 NO CURRENT 3064990 2016/3/21 2 201474976710
3 1 93 52428800 512 1 NO INACTIVE 3008818 2016/3/10 3 3033485 2016/3/19 1
```

### 6. dba\_\all\_\user\_前缀的数据字典: 使用 select 语句测试数据查询情况

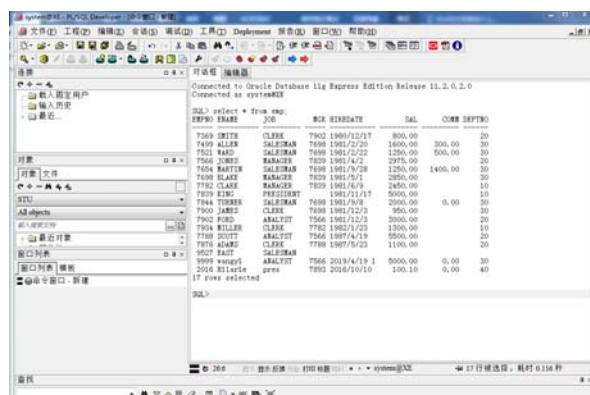
*dba\_tables: all\_tables; user\_tables*

#### (三) PL/SQL develop 使用

##### 1. 登录界面



##### 2. 操作界面



## 实验三 SQL 基础语句

### 一、实验目的

1. 掌握常见的 DDL、DML 和 DCL 用法。
2. 掌握 SELECT 语句的语法结构，熟练使用 WHERE、GROUP BY、HAVING 和 ORDER BY 子句。
3. 熟练掌握 INSERT、UPDATE\DELETE 和 MERGE 语句使用；
4. 掌握字符串函数、数字函数、日期时间函数和聚合函数。

### 二、实验环境

- 1、硬件设备：计算机局域网，服务器 1 台，终端计算机每人一台
- 2、软件系统： Windows 客户机操作系统；  
Oracle 11g 服务端数据库系统，客户端工具。

### 三、实验内容

#### 一、实验环境检查

在实验之前，请同学们查看上机环境是否正常。

- (一) 所使用的计算机名称(在[我的电脑](#)图标上点右键查看[属性](#)中[计算机名](#))，确保与 oracle 安装路径下的配置文件 tnsnames.ora 和 listener.ora 文件中 host 值保持一致，如下图所示。

```
XE =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = superb)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = XE)
    )
  )

EXTPROC_CONNECTION_DATA =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1))
    )
    (CONNECT_DATA =
      (SID = PLSExtProc)
      (PRESENTATION = RO)
    )
  )

ORACLR_CONNECTION_DATA =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1))
    )
    (CONNECT_DATA =
      (SID = CLRExtProc)
      (PRESENTATION = RO)
    )
  )
```

图 1. tnsnames.ora 配置文件

```
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = PLSExtProc)
      (ORACLE_HOME = C:\oraclexe\app\oracle\product\11.2.0\server)
      (PROGRAM = extproc)
    )
    (SID_DESC =
      (SID_NAME = CLRExtProc)
      (ORACLE_HOME = C:\oraclexe\app\oracle\product\11.2.0\server)
      (PROGRAM = extproc)
    )
  )

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1))
      (ADDRESS = (PROTOCOL = TCP)(HOST = superb)(PORT = 1521))
    )
  )

DEFAULT_SERVICE_LISTENER = (XE)
```

图 2. listener.ora 配置文件

(二)打开计算机运行(快捷键“**按住 windows 键+R 键**”),在运行框中输入“**services.msc**”并点击回车, 查看 oracle 相关实例服务和监听器服务是否起来。如没有启动, 需要手动启动, 如图 3。

 OracleJobSchedulerXE	禁用	本地系统
 OracleMTSRecoveryService	手动	本地系统
 OracleServiceXE	已启动	本地系统
 OracleXEClrAgent	手动	本地系统
 OracleXETNNSListener	已启动	本地系统

图 3. oracle 相关服务

## 二、 scott. sql 脚本解读

使用“@路径\scott.sql”，导入 scott.sql 脚本。

```

1 Rem Copyright (c) 1990 by Oracle Corporation
2 Rem NAME
3 Rem   UTLSAMPLI.SQL
4 Rem FUNCTION
5 Rem NOTES
6 Rem MODIFIED
7 Rem goudrey 06/28/95 - Modified for desktop seed database
8 Rem glumpkin 10/21/92 - Renamed from SQLBLD.SQL
9 Rem blinden 07/27/92 - Added primary and foreign keys to EMP and DEPT
10 Rem rlim 04/29/91 - change char to varchar2
11 Rem mmcore 04/08/91 - use unlimited tablespace priv
12 Rem pritto 04/04/91 - change SYSDATE to 13-JUL-97
13 Rem Mendels 12/07/90 - bug 30123;add to_date calls so language independent
14 Rem
15 Rem
16 rem
17 rem $Header: utlsampli.sql 7020100.1 94/09/23 22:14:24 cli Generic<base> $ sqlbld.sql
18 rem
19 rem 原scott脚本存放在oracle 11g express存放在C:\oraclexe\app\oracle\product\11.2.0\server\rdbms\admin目录下
20 rem-
21 /* 代码注释由 Wangyi, 仅供参考*/
22 SET TERMOUT OFF          /* 显示脚本中的命令的执行结果, 缺省为on; */
23 SET ECHO OFF              /* 显示START启动的脚本中的每个SQL命令, 缺省为on; */
24
25 rem CONDON   Invoked in RDBMS at build time. 29-DEC-1988
26 rem OATES:   Created: 16-Feb-83
27
28 GRANT CONNECT,RESOURCE,UNLIMITED TABLESPACE TO SCOTT IDENTIFIED BY TIGER;
29
30 ALTER USER SCOTT DEFAULT TABLESPACE USERS; /* 授予scott用户CONNECT, RESOURCE角色和系统权限unlimited tablespace (隐含在dba, resource角色中) */
31 ALTER USER SCOTT TEMPORARY TABLESPACE TEMP; /* 将scott的默认表空间设置为users */
32 CONNECT SCOTT/TIGER
33 DROP TABLE DEPT; /* 用户scott连接数据库 */
34 CREATE TABLE DEPT /* 删掉已存在的DEPT表 */
35   (DEPTNO NUMBER(2) CONSTRAINT PK_DEPT PRIMARY KEY, /* 设置表的主键为deptno字段, 主键名为PK_DEPT, PRIMARY KEY为保留字 */
36   DNAME VARCHAR2(14),
37   LOC VARCHAR2(13));
38
39 DROP TABLE EMP;
40 CREATE TABLE EMP /* 创建emp表 */
41   (EMPNO NUMBER(4) CONSTRAINT PK_EMP PRIMARY KEY,
42    ENAME VARCHAR2(10),
43    JOB VARCHAR2(9),
44    MGR NUMBER(4),
45    HIREDATE DATE, /* 设置MGR为数值型, 长度4个字符 */
46    SAL NUMBER(7,2), /* 设置HIREDATE字段为日期型, 默认"DD-MM-YY"格式 */
47    COMM NUMBER(7,2), /* 设置SAL长度为7个字符. 小数点后2位 */
48    DEPTNO NUMBER(2) CONSTRAINT FK_DEPTNO REFERENCES DEPT); /* 设置deptno为外键, 数值来自于dept表, 两个表的相应列取值保持一致 */
49 INSERT INTO DEPT VALUES (10,'ACCOUNTING','NEW YORK'); /* 同dept表中插入一系列值, 字段间用半角逗号隔开 */
50 INSERT INTO DEPT VALUES (20,'RESEARCH','DALLAS');
51 INSERT INTO DEPT VALUES (30,'SALES','CHICAGO');
52 INSERT INTO DEPT VALUES (40,'OPERATIONS','BOSTON');
53 /* 向emp表插入一组数据 */
54 INSERT INTO EMP VALUES (7369,'SMITH','CLERK','7902,to_date(''17-12-1980'',''dd-mm-yyyy''),800,NULL,20); /* 利用to_date函数将字符串转为日期型'dd-mm-yyyy' */
55 INSERT INTO EMP VALUES (7499,'ALLEN','SALESMAN','7698,to_date(''20-2-1981'',''dd-mm-yyyy''),1600,300,30);
56 INSERT INTO EMP VALUES (7521,'WARD','SALESMAN','7698,to_date(''22-2-1981'',''dd-mm-yyyy''),1250,500,30);
57 INSERT INTO EMP VALUES (7566,'JONES','MANAGER','7839,to_date(''2-4-1981'',''dd-mm-yyyy''),2975,NULL,20);
58 INSERT INTO EMP VALUES (7654,'MARTIN','SALESMAN','7698,to_date(''28-9-1981'',''dd-mm-yyyy''),1250,1400,30);
59 INSERT INTO EMP VALUES (7698,'BLAKE','MANAGER','7839,to_date(''1-5-1981'',''dd-mm-yyyy''),2850,NULL,30);
60 INSERT INTO EMP VALUES (7782,'CLARK','MANAGER','7839,to_date(''9-6-1981'',''dd-mm-yyyy''),2450,NULL,10);
61 INSERT INTO EMP VALUES (7788,'SCOTT','ANALYST','7566,to_date(''13-JUL-87'')-85,3000,NULL,20); /* 插入数据字段日期转换格式与其他记录不同 */
62 INSERT INTO EMP VALUES (7839,'KING','PRESIDENT','NULL,to_date(''17-11-1981'',''dd-mm-yyyy''),5000,NULL,10);
63 INSERT INTO EMP VALUES (7844,'TURNER','SALESMAN','7698,to_date(''8-9-1981'',''dd-mm-yyyy''),1500,0,30);
64 INSERT INTO EMP VALUES (7876,'ADAMS','CLERK','7788,to_date(''13-JUL-87'')-51,1100,NULL,20);
65
66 INSERT INTO EMP VALUES (7900,'MAMES','CLERK','7698,to_date(''3-12-1981'',''dd-mm-yyyy''),950,NULL,30);
67
68 INSERT INTO EMP VALUES (7902,'FORD','ANALYST','7566,to_date(''3-12-1981'',''dd-mm-yyyy''),3000,NULL,20);
69
70 INSERT INTO EMP VALUES (7934,'MILLER','CLERK','7782,to_date(''23-1-1982'',''dd-mm-yyyy''),1300,NULL,10);
71
72 DROP TABLE BONUS;
73 CREATE TABLE BONUS /* 创建工资补贴表bonus */
74   (
75    ENAME VARCHAR2(10), /* 雇员姓名 */
76    JOB VARCHAR2(9), /* 雇员工作 */
77    SAL NUMBER, /* 基本工资 */
78    COMM NUMBER /* 奖金 */
79   );
80
81 DROP TABLE SALGRADE; /* 工资等级表 */
82 CREATE TABLE SALGRADE /* 工资等级表 */
83   (
84    GRADE NUMBER, /* 对应等级编号 */
85    LOSAL NUMBER, /* 对应等级最低工资 */
86    HISAL NUMBER ); /* 对应等级最高工资 */
87
88 INSERT INTO SALGRADE VALUES (1,100,1200);
89 INSERT INTO SALGRADE VALUES (2,1201,1400);
90 INSERT INTO SALGRADE VALUES (3,1401,2000);
91 INSERT INTO SALGRADE VALUES (4,2001,3000);
92 INSERT INTO SALGRADE VALUES (5,3001,9999);
93
94 COMMIT; /* 显式提交 */
95
96 SET TERMOUT ON /* 设置执行脚本显示为on状态 */
97 SET ECHO ON /* 设置系统交互式为on状态 */

```

### 三、数据库基本查询技术

请大家以 scott 用户登录 PL/SQL Developer，对 emp 表进行相关查询，见图 4。



图 4. 查询界面

1. 使用字符串连接符号查询 emp 表信息，各字段值用 “，” 间隔输出。

2. 查询工资收入在 1000 元和 1500 元之间人员信息；

3. 查询 emp 表人员中以 J 开头的人员信息；

4. 查询部门编号为 20 且入职日期为 81 年的员工信息；

5. 查询 10 和 20 部门员工信息并按照工资降序排列；

请再考虑使上述查询结果中津贴（comm）为空的记录排在前面。

6. 查询各部门员工信息并按照部门升序、工资降序排列结果；

7. 统计 30 部门的员工数量；

8. 查询所有员工的平均工资；

9. 查询各部门的平均工资；

10. 查看 emp 表的所有职位数；

11. 查看 emp 表中最低和最高工资情况；

12. 查看各部门薪资小计大于 2000 的人员信息；

13. 向 emp 表添加若干条记录，数据自拟。

14. 用 delete 语句删除上调语句添加的数据；
15. 查询每部门的工资小计以及总工资数量。
16. 查询没有部门主管的人员信息。
17. 查找津贴 comm 高于工资 sal 的员工信息
18. 查询各个部门经理的收入情况。
19. 查询员工的基本信息以及其主管姓名
20. 显示工资比 Blake 高的所有员工的姓名和工作

#### 四、实验要求

- (一) 请独立完成实验，实验报告文件名以“学号+姓名”命名。
- (二) 提交方式：雨课堂提交，截止 3 月 18 日 24 时

# 实验四 表的完整性

## 一、实验目的

- (一) 掌握表空间创建与管理;
- (二) 掌握表的定义和创建;
- (三) 掌握表的完整性约束定义和使用;

## 二、实验环境

- (一) 硬件设备：计算机局域网，服务器 1 台，终端计算机每人一台
- (二) 软件系统：Windows XP 以上客户机操作系统；  
Oracle11g 服务端数据库系统，客户端工具 PL/SQL developer。

## 三、实验内容

### (一) 实验环境检查

在实验之前，请同学们按照如下步骤查看上机环境是否正常。

1. 所使用的计算机名称（在〔我的电脑〕图标上点右键查看〔属性〕中〔计算机名〕），确保与 oracle 安装路径下的配置文件 tnsnames.ora 和 listener.ora 文件中 host 值保持一致，如下图所示。

```
XE =
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP)(HOST = superb)(PORT = 1521))
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = XE)
  )
)

EXTPROC_CONNECTION_DATA =
(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1))
  )
  (CONNECT_DATA =
    (SID = PLSExtProc)
    (PRESENTATION = RO)
  )
)

ORACLR_CONNECTION_DATA =
(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1))
  )
  (CONNECT_DATA =
    (SID = CLRExtProc)
    (PRESENTATION = RO)
  )
)
```

图 1. tnsnames.ora 配置文件

```
SID_LIST_LISTENER =
(SID_LIST =
  (SID_DESC =
    (SID_NAME = PLSExtProc)
    (ORACLE_HOME = C:\oraclexe\app\oracle\product\11.2.0\server)
    (PROGRAM = extproc)
  )
  (SID_DESC =
    (SID_NAME = CLRExtProc)
    (ORACLE_HOME = C:\oraclexe\app\oracle\product\11.2.0\server)
    (PROGRAM = extproc)
  )
)

LISTENER =
(DESCRIPTION_LIST =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1))
    (ADDRESS = (PROTOCOL = TCP)(HOST = superb)(PORT = 1521))
  )
)

DEFAULT_SERVICE_LISTENER = (XE)
```

图 2. listener.ora 配置文件

2. 打开计算机运行（快捷键“[按住 windows 键+R 键](#)”），在运行框中输入“[services.msc](#)”

并点击回车，查看 oracle 相关实例服务和监听器服务是否起来。如没有启动，需要手动启动，如图 3。

OracleJobSchedulerXE	禁用	本地系统
OracleMTSRecoveryService	手动	本地系统
OracleServiceXE	已启动	本地系统
OracleXEClrAgent	手动	本地系统
OracleXETNSListener	已启动	本地系统

图 3. oracle 相关服务

## (二) 表空间的定义

掌握表空间的定义和使用，蓝色斜体字要根据自己数据存放情况修改。

1.永久表空间： **myspace**

```
create tablespace myspace
  datafile 'D:\app\wangyi\mydata\data.dbf'
  size 10M
  autoextend on
  next 5M
  maxsize 50M;
```

2.创建临时表空间： **mytemp**

```
create temporary tablespace mytemp
  tempfile 'D:\app\wangyi\mydata\mytemp.dbf'
  size 10M
  autoextend on
  next 2M
  maxsize 20M;
```

3.创建撤销表空间： **myundo**

```
create undo tablespace myundo
  datafile 'D:\app\wangyi\mydata\myundo.dbf'
  size 10M
  autoextend on
  next 5M
  maxsize 50M
```

4.查询表空间

```
select tablespace_name from user tablespaces;
```

5.查询表空间数据文件

```
SELECT tablespace_name , file_name , bytes FROM dba_data_files WHERE
tablespace_name = 'MYSPACE'
```

6. 为当前表空间扩容 50M

```
ALTER TABLESPACE wy
```

```
ADD DATAFILE
'D:\app\wangyi\mydata\data2.dbf'
```

```
SIZE 10M  
AUTOEXTEND ON NEXT 5M MAXSIZE 40M,
```

### (三) 验证实验: 用户创建与管理

#### 1. 创建用户 *tuster*

```
create user tuster  
identified by admin  
default tablespace myspace  
quota 20M on myspace; //当前用户可用表空间最大值, 可设为 unlimited
```

#### 2. 查看当前用户会话信息

```
Select sid,serial#,username,status,logon_time,machine  
FROM v$session WHERE username IS NOT NULL
```

#### 3. 修改用户密码为 orcl

```
alter user tuster identified by orcl
```

#### 4. 修改用户的锁定状态

```
alter user tuster account lock[unlock]
```

#### 5. 设定用户密码为过期状态

```
alter user tuster password expire
```

#### 6. 删除用户, 包括用户所有数据文件

```
drop user tuster cascade
```

注意 cascade 使用

### (四) 验证内容: 表的完整性约束。

约束是保证表中数据完整性和一致性的机制, 要求熟练掌握以下五种常见的约束, 可在列上定义也可单独定义约束。

- 主键约束: primary key
- 唯一约束: unique;
- 非空约束: not null;
- 外键约束: foreign key

创建 scott 账户及其中的表, *emp* 员工表 (*empno,ename,job,mgr,hiredate,sal,comm.,deptno*)  
*dept* 部门表 (*deptno, dname, loc*)

#### 1. 创建 dept 表, 注意完整性约束。

```
create table dept(  
    deptno number(10) primary key,  
    dname varchar2(30) not null,  
    loc varchar2(30) check (loc in('Tianjin','Hebei','Shanghai')) //限定地点  
)
```

添加几条数据到创建的 dept 表中

- (1) `insert into dept values(1001, 'Sale Market', 'Tianjin');` 正常添加
- (2) `insert into dept values(1002, '技术部', 'Shandong');` 添加非 check 条件的数据, 出现违例。

```
| SQL> insert into dept values(1002,'技术部','Shandong');  
| insert into dept values(1002,'技术部','Shandong')  
| ORA-02290: 违反检查约束条件 (SYSTEM.SYS_C0011668)
```

## 2. 创建 emp 表

```
create table emp(
    empno number(10) primary key,
    ename varchar2(30) not null,
    job varchar2(30),
    mgr varchar2(30),
    hiredate number(10),
    sal number(10),
    comm number(10),
    deptno number(10),
    foreign key(deptno) references dept(deptno)
)
```

添加一些数据并进行测试。

**删除表:** `drop table emp; drop table dept;`

3. 主键的设定也可在表列定义的后面，作为表级约束

例如：

```
create table categoryinfo
(
    categoryID varchar(10),
    categoryname varchar2(30),
    primary key(categoryID))
```

4. 假如在创建表时没有添加约束，可以通过以下语句后期添加（也可移除约束）

```
alter table categoryinfo add constraint PK_category primary key(categoryID)
alter table categoryinfo drop constraint PK_category
```

**(五) 实践内容：请创建 SCT 表空间和用户（需授权），完成学生选课数据库模式中表的完整性约束定义，需要提交报告。**

1. 学生表

`Student(S# char(8),Sname char(10),Ssex char(2), Sage integer, D# char(2), Sclass char(6))`  
S#:学号; Sname: 学生姓名; Ssex:性别; Sage:年龄; D#:所属系; Sclass:所属班级;

2. 系别表

`Dept(D# char(2),Dname char(10),Dean char(10))`  
D#:系别; Dname:系名, Dean:系主任

3. 课程表

`Course(C# char(3),Cname char(12),Chours integer,Credit float(1),T# char(3))`  
C#:课程号; Cname:课程名; Chours:学时; Credit:学分; T#:教师编号;

4. 教师表

`Teacher(T# char(3), Tname char(10),D# char(2),Salary float(2))`  
T#:教师编号; Tname:教师姓名; D#:所属学院; Salary:薪资

## 5.选课表

SC(S# char(8), C# char(3),Score float(1))

S#:学号; C#: 课程号; Score:成绩

## 四、实验要求

(一) 请独立完成实验，实验报告文件名以“学号+姓名”命名。

(二) 提交方式：雨课堂提交，截止 3 月 31 日 23:55 分

# 实验五 模式与权限

## 一、实验目的

- (一) 掌握系统权限与对象权限;
- (二) 掌握用户角色定义与管理;
- (三) 掌握概要文件使用。

## 二、实验环境

- (一) 硬件设备：计算机局域网，服务器 1 台，终端计算机每人一台
- (二) 软件系统： Windows XP 以上客户机操作系统；  
Oracle11g 服务端数据库系统，客户端工具 PL/SQL developer。

## 三、实验内容

### (一) 实验环境检查

在实验之前，请同学们按照如下步骤查看上机环境是否正常。

1. 所使用的计算机名称（在（我的电脑）图标上点右键查看（属性）中（计算机名）），确保与 oracle 安装路径下的配置文件 tnsnames.ora 和 listener.ora 文件中 host 值保持一致，如下图所示。

```
XE =
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP)(HOST = superb)(PORT = 1521))
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = XE)
  )
)

EXTPROC_CONNECTION_DATA =
(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1))
  )
  (CONNECT_DATA =
    (SID = PLSExtProc)
    (PRESENTATION = RO)
  )
)

ORACLE_CONNECTION_DATA =
(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1))
  )
  (CONNECT_DATA =
    (SID = CLRExtProc)
    (PRESENTATION = RO)
  )
)
```

图 1. tnsnames.ora 配置文件

```
SID_LIST_LISTENER =
(SID_LIST =
  (SID_DESC =
    (SID_NAME = PLSExtProc)
    (ORACLE_HOME = C:\oraclexe\app\oracle\product\11.2.0\server)
    (PROGRAM = extproc)
  )
  (SID_DESC =
    (SID_NAME = CLRExtProc)
    (ORACLE_HOME = C:\oraclexe\app\oracle\product\11.2.0\server)
    (PROGRAM = extproc)
  )
)

LISTENER =
(DESCRIPTION_LIST =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1))
    (ADDRESS = (PROTOCOL = TCP)(HOST = superb)(PORT = 1521))
  )
)

DEFAULT_SERVICE_LISTENER = (XE)
```

图 2. listener.ora 配置文件

2. 打开计算机运行（快捷键“**按住 windows 键+R 键**”），在运行框中输入“**services.msc**”

并点击回车，查看 oracle 相关实例服务和监听器服务是否起来。如没有启动，需要手动启动，如图 3。

OracleJobSchedulerXE	禁用	本地系统
OracleMTSRecoveryService	手动	本地系统
OracleServiceXE	已启动	本地系统
OracleXEClrAgent	手动	本地系统
OracleXETNSListener	已启动	本地系统

图 3. oracle 相关服务

## (二) 权限的的定义和使用

新建用户无任何权限。权限分为系统权限授权和对象权限授权。

### 1. 系统权限

系统权限指对资源操作的权限，DBA 是数据库中级别最高的用户，拥有最高权限，能够对各种资源进行操作，oracle 中有超过 100 多种的系统权限，且随着数据库版本发展，权限也增多。

(1) 查询当前 oracle 数据库具有的系统权限数量，如下图。

```
SQL 窗口 - select count(*) from sys.system_privilege_map;
SQL | 输出 | 统计
1 | select count(*) from sys.system_privilege_map;
| COUNT(*) |
| 1 | 208 |
& 1 of 1 | sys@XE AS SYSDBA |
```

(2) 两个特殊的系统权限。SYSDBA、SYSOPER，与体系结构部分学习密切相关，SYSDBA 可以创建数据库，改变表空间等。SYSOPER 执行 startup 或 shutdown、alter database open 等权限。

(3) 查询以授权给用户的系统权限 USER\_SYS\_PRIVS；

(4) 请熟悉常见的系统权限

有关用户的系统权限		
1	CREATE USER:	创建其他的用户（需具有 DBA 角色权限）
2	ALTER USER	修改其他用户的设置
3	DROP USER	删除其他用户
有关表的系统权限		
1	CREATE TABLE	用户在自己的模式中创建表
2	ALTER ANY TABLE	修改任何模式中所创建的表
3	CREATE ANY TABLE	在任何模式中创建表
4	DELETE ANY TABLE	删除任何用户表中的数据和视图中的数据
5	UPDATE ANY TABLE	修改任何用户的表中的数据和视图中的数据

6	DROP ANY TABLE	删除任何模式中所创建的表
有关表空间的系统权限		
1	CREATE TABLESPACE	创建表空间的权限
2	DROP TABLESPACE	删除表空间的权限
3	ALTER TABLESPACE	修改表空间的权限
4	UNLIMITED TABLESPACE	使用全部表空间的权限
有关索引的系统权限		
1	CREATE ANY INDEX	在任何模式中创建索引的权限
2	DROP ANY INDEX	在任何模式中删除索引的权限
3	ALTER ANY INDEX	在任何模式中修改索引的权限
有关会话的系统权限		
1	CREATE SESSION	连接数据库的权限
2	ALTER SESSION	修改数据库会话的权限
有关概要文件系统权限		
1	CREATE PROFILE	创建概要文件
2	ALTER PROFILE	修改概要文件
3	DROP PROFILE	删除概要文件
其他的系统权限（视图、序列等）		
1	CREATE VIEW	用户在自己的模式中创建视图的权限
2	CREATE ANY VIEW	为任意用户创建视图权限
3	CREATE SEQUENCE	在用户自己的模式中创建序列号的权限
4	CREATE PROCEDURE	在用户自己的模式中创建过程的权限

(5) 创建 tuster 用户，进行授权验证。

```
Grant create session,create table to tuster
```

```
Revoke create session from tuster
```

测试是否可以使用 tuster 创建表？

(6) 请授权给 tuster 上表的系统权限，然后 tuster 把对应权限传给另一个新建用户 tuster2，请考虑带有 with admin option 开关情况下的系统权限授权情况，然后完成权限的回收。

## 2. 对象权限

对象权限指对表、序列和触发器等操作的权限。包括对表和视图的 select、insert、update、delete、alter 等，对 execute 子程序、alter sequence 等。

(1) 查询授权给当前用户的对象权限 USER\_TAB\_PRIVS；

(2) 对象权限授权及撤销

```
Grant select,update on emp to tuster;
```

Revoke select on emp from tuster --请注意收回后用户权限变化

请考虑带有 with grant option 选项开关情况下的对象权限授权情况，测试用户之间授权传递及回收。

(3) 将访问 scott 模式下 emp 表 ename、hiredate 字段查询权限授予 tuster;

(4) 用数据字典查看建立账户的状态。

The screenshot shows an Oracle SQL Developer window titled "SQL 窗口 - Select username,account\_status from dba\_users where username='TUSTER'". The query window contains the SQL statement: "Select username,account\_status from dba\_users where username='TUSTER';". The results window displays a single row with columns "USERNAME" and "ACCOUNT\_STATUS". The row shows "TUSTER" in the first column and "OPEN" in the second column. The status bar at the bottom indicates "1行被选择, 耗时 0.030 秒".

USERNAME	ACCOUNT_STATUS
TUSTER	OPEN

请练习查询用户信息: all\_users, dba\_users, user\_users 注意三者区别。

### (三) 角色 role

1. 角色创建

(1) conn tuster/orcl

(2) 创建与授权

Create role user\_role identified by user; //如果 not identified 则表明不需要验证。

```
Grant select,update,insert on emp to user_role;
```

```
Grant user_role to tuster;
```

(3) 删除角色: drop role user\_role;

2. oracle 内置角色:**CONNECT, RESOURCE, DBA** 是 ORACLE 系统的三个内置角色:

**CONNECT**: 授予终端用户，能够连接到 ORACLE 数据库，对其他用户的表有访问权限时，做 SELECT、UPDATE、INSERT、DELETE 等操作。

**RESOURCE**: 是授予开发人员的，能在自己的方案中创建表、序列、视图等。

**DBA**: 拥有全部特权，是系统最高权限，可以创建数据库结构。

对用户进行 DBA 授权

```
grant connect,resource,DBA to tuster;
```

3. 查询角色所拥有的权限个数

```
select role, count(*) from role_sys_privs group by role;
```

The screenshot shows an Oracle SQL Developer interface with a SQL window titled "SQL 窗口 - select role, count(\*) from role\_sys\_privs group by role;". The results are displayed in a table:

ROLE	COUNT(*)
1 EXP_FULL_DATABASE	11
2 AQ_ADMINISTRATOR_ROLE	6
3 DBA	202
4 OEM_ADVISOR	3
5 RECOVERY_CATALOG_OWNER	11
6 SCHEDULER_ADMIN	6
7 RESOURCE	8
8 IMP_FULL_DATABASE	80
9 DATAPUMP_EXP_FULL_DATABASE	2
10 CONNECT	1
11 DATAPUMP_IMP_FULL_DATABASE	14
12 OEM_MONITOR	7
13 LOGSTDBY_ADMINISTRATOR	1

At the bottom of the window, it says "13 行被选择, 耗时 0.0s".

4. 使用 v\$session 查看当前数据库中已建立的会话信息。

```
Select sid,serial#,username,status,logon_time,machine
FROM v$session WHERE username IS NOT NULL
```

#### (四) 概要文件

目的是存放数据库系统资源或者数据库使用限制的内容，oracle 默认概要文件为 default。

1. 创建概要文件 testpro，设置用户密码失效天数为 30 天。

```
SQL> create profile testpro
  2  limit
  3  password_life_time 30
  4 /
Profile created
```

2. 查询概要文件：select distinct profile from dba\_profile;

3. 删除概要文件 drop profile tusterpro

#### (五) 综合题

创建名为 stu 的表空间，大小为 100M，数据存储在 d:\test 目录下，再创建一个用户并授予相应权限，该用户完成在表空间下建立学生表 student 和专业表 major。

**student** (学号、姓名、性别、班级、班长、专业编号、考试时间、成绩、备注)

- [1] 各字段需定义完整性约束：学号为主键；
- [2] 班长字段为本班某个学生学号；
- [3] 姓名不能为空；性别取值仅能为男和女；
- [4] 专业编号为外键，取值来源于 major，形如 181031；
- [5] 成绩要求大于 0 分，小于 710.

**major** (专业编号，专业名称，归属学院，开办时间，备注)，

- [1] 专业编号为主键，专业名称为非空。
- [2] 向个表中填加一些相关数据验证。



# 实验六 视图与序列

## 一、实验目的

(一) 掌握视图的定义与使用;

(二) 掌握序列的定义与使用。

## 二、实验环境

(一) 硬件设备: 计算机局域网, 服务器 1 台, 终端计算机每人一台

(二) 软件系统: Windows XP 以上客户机操作系统;

Oracle11g 服务端数据库系统, 客户端工具 PL/SQL developer。

## 三、实验内容

### (一) 实验环境检查

在实验之前, 请同学们按照如下步骤查看上机环境是否正常。

1. 所使用的计算机名称 (在 [我的电脑](#) 图标上点右键查看 [属性](#) 中 [\(计算机名\)](#)), 确保与 oracle 安装路径下的配置文件 tnsnames.ora 和 listener.ora 文件中 host 值保持一致, 如下图所示。

```
XE =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = superb)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = XE)
    )
  )

  EXTPROC_CONNECTION_DATA =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1))
    )
    (CONNECT_DATA =
      (SID = PLSExtProc)
      (PRESENTATION = RO)
    )
  )

  ORACLE_CONNECTION_DATA =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1))
    )
    (CONNECT_DATA =
      (SID = CLRExtProc)
      (PRESENTATION = RO)
    )
  )
```

图 1. tnsnames.ora 配置文件

```
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = PLSExtProc)
      (ORACLE_HOME = C:\oraclexe\app\oracle\product\11.2.0\server)
      (PROGRAM = extproc)
    )
    (SID_DESC =
      (SID_NAME = CLRExtProc)
      (ORACLE_HOME = C:\oraclexe\app\oracle\product\11.2.0\server)
      (PROGRAM = extproc)
    )
  )

  LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1))
      (ADDRESS = (PROTOCOL = TCP)(HOST = superb)(PORT = 1521))
    )
  )

  DEFAULT_SERVICE_LISTENER = (XE)
```

图 2. listener.ora 配置文件

2. 打开计算机运行(快捷键[按住 windows 键+R 键](#)), 在运行框中输入[“services.msc”](#)并点击回车, 查看 oracle 相关实例服务和监听器服务是否起来。如没有启动, 需要手动启

动，如图 3。

OracleJobSchedulerXE	禁用	本地系统
OracleMTSRecoveryService	手动	本地系统
OracleServiceXE	已启动	手动
OracleXEClrAgent	手动	本地系统
OracleXETNSListener	已启动	手动

图 3. oracle 相关服务

## (二) 视图的定义和使用

1. 在 system 用户下创建基于 emp 表的视图 emp\_view，并对该视图进行查询；

```
create view emp_view
as
select * from scott.emp where sal>1000 with check option
```

对创建的视图进行查询，如下图，少了 1000 元以下人员信息：

EMPNO	ENAME	JOB	HIREDATE	SAL	COMM	DEPTNO
7499	ALLEN	SALESMAN	7698 1981/2/20	1600.00	300.00	30
7521	WARD	SALESMAN	7698 1981/2/22	1250.00	500.00	30
7566	JONES	MANAGER	7839 1981/4/2	2975.00		20
7654	MARTIN	SALESMAN	7698 1981/9/28	1250.00	1400.00	30
7698	BLAKE	MANAGER	7839 1981/5/1	2850.00		30
7782	CLARK	MANAGER	7839 1981/6/9	2450.00		10
7788	SCOTT	ANALYST	7566 1987/4/19	3000.00		20
7839	KING	PRESIDENT	1981/11/17	5000.00		10
7844	TURNER	SALESMAN	7698 1981/9/8	1500.00	0.00	30
7876	ADAMS	CLERK	7788 1987/5/23	1100.00		20
7902	FORD	ANALYST	7566 1981/12/3	3000.00		20
7934	MILLER	CLERK	7782 1982/1/23	1300.00		10

思考：创建包含雇员信息及部门名称和工作地点的视图如何写？

2. 创建基于一个 emp 表的视图 emp\_view2，并且对子查询中检索的 sal 列进行数学计算，查询工资上涨 20% 以后的工资大于 2500 的员工信息。

```
create or replace view emp_view2
as
select empno,ename,job,hiredate,sal "原工资",sal*1.2 "涨后工资额" from scott.emp
```

查询创建的视图 emp\_view2，如下图。

EMPNO	ENAME	JOB	HIREDATE	原工资	涨后工资额
7369	SMITH	CLERK	1980/12/17	800.00	960
7499	ALLEN	SALESMAN	1981/2/20	1600.00	1920
7521	WARD	SALESMAN	1981/2/22	1250.00	1500
7566	JONES	MANAGER	1981/4/2	2975.00	3570
7654	MARTIN	SALESMAN	1981/9/28	1250.00	1500
7698	BLAKE	MANAGER	1981/5/1	2850.00	3420
7782	CLARK	MANAGER	1981/6/9	2450.00	2940
7788	SCOTT	ANALYST	1987/4/19	3000.00	3600
7839	KING	PRESIDENT	1981/11/17	5000.00	6000
7844	TURNER	SALESMAN	1981/9/8	1500.00	1800
7876	ADAMS	CLERK	1987/5/23	1100.00	1320
7900	JAMES	CLERK	1981/12/3	950.00	1140
7902	FORD	ANALYST	1981/12/3	3000.00	3600
7934	MILLER	CLERK	1982/1/23	1300.00	1560

## (三) 序列的定义与使用

使用序列值需引用的伪列是 *nextval* 而不是 *currval* 伪列。

- NEXTVAL 返回序列中下一个有效的值，任何用户都可以引用，应在 CURRVAL 之前指定；
- CURRVAL 中存放序列的当前值。

1. 首先创建一个表 student，字段包括 sid(4), sname(varchar2(8)), 并且 sid 为主键。

建表语句参考如下：

```
CREATE TABLE student (
    sid NUMBER(4) PRIMARY KEY ,
    sname VARCHAR2(8) NOT NULL
)
```

2. 创建序列 student\_seq, 语句如下：

```
CREATE SEQUENCE student_seq
    START WITH 1001
    INCREMENT BY 1
    NOCACHE
    NOCYCLE
    ORDER
```

注释：可通过查询 dual 表获得当前序列值情况。

3. 对 student 表添加如下数据，然后查看一下结果，注意观察主键生成情况。

```
insert into student(sid,sname) values(student_seq.nextval,'晋文东');
insert into student(sid,sname) values(student_seq.nextval,'郭德纲');
insert into student(sid,sname) values(student_seq.nextval,'CMU');
insert into student(sid,sname) values(student_seq.nextval,'马克斯');
```

4. 查询新添加的记录，观察序列的产生。

```
select * from student;
```

5. 测试序列的两个伪列值

```
select student_seq.nextval from dual;
select student_seq.curval from dual
```

6. 思考：可以新建一个用户，通过两用户切换了解序列使用时两个伪列值的变化。

#### (四) 了解 ROWID

唯一标识表上的一行，快速定位单行记录。ROWID 有 18 位，每一位均采用 64 位编码，分别用 A~Z、a~z、0~9、+、/共 64 个字符表示。A 表示 0，B 表示 1，……Z 表示 25；a 表示 26，……z 表示 51；0 表示 52，……，9 表示 61；+表示 62，/表示 63；

测试代码：

```
sql>create table test_rowid ( id number, row_id rowid);
表已创建。
sql>insert into test_rowid values ( 1, null );
已创建 1 行。
sql>update test_rowid set row_id = rowid where id =1;
已更新 1 行。
sql>commit;
提交完成。
sql> select rowid, row_id from test_rowid;
```



# 实验七 PL/SQL 程序设计（1）

## 一、实验目的

- (一) 掌握 PL/SQL 程序块的基本结构。
- (二) 掌握变量的定义与使用，熟练掌握%type 和%rowtype 的使用。
- (三) 掌握 PL/SQL 循环语句和条件控制语句。

## 二、实验环境

- (一) 硬件设备：计算机局域网，服务器 1 台，终端计算机每人一台
- (二) 软件系统： Windows XP 以上客户机操作系统；  
Oracle11g 服务端数据库系统，客户端工具 PL/SQL developer。

## 三、实验内容

### (一) 实验环境设置

在实验之前，请同学们按照如下步骤查看上机环境是否正常。

1. 所使用的计算机名称（在（我的电脑）图标上点右键查看（属性）中（计算机名）），确保与 oracle 安装路径下的配置文件 tnsnames.ora 和 listener.ora 文件中 host 值保持一致，如下图所示。

```
XE =
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP)(HOST = superb)(PORT = 1521))
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = XE)
  )
)

EXTPROC_CONNECTION_DATA =
(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1))
  )
  (CONNECT_DATA =
    (SID = PLSExtProc)
    (PRESENTATION = RO)
  )
)

ORACLR_CONNECTION_DATA =
(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1))
  )
  (CONNECT_DATA =
    (SID = CLRExtProc)
    (PRESENTATION = RO)
  )
)
```

图 1. tnsnames.ora 配置文件

```
SID_LIST_LISTENER =
(SID_LIST =
  (SID_DESC =
    (SID_NAME = PLSExtProc)
    (ORACLE_HOME = C:\oraclexe\app\oracle\product\11.2.0\server)
    (PROGRAM = extproc)
  )
  (SID_DESC =
    (SID_NAME = CLRExtProc)
    (ORACLE_HOME = C:\oraclexe\app\oracle\product\11.2.0\server)
    (PROGRAM = extproc)
  )
)

LISTENER =
(DESCRIPTION_LIST =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1))
    (ADDRESS = (PROTOCOL = TCP)(HOST = superb)(PORT = 1521))
  )
)

DEFAULT_SERVICE_LISTENER = (XE)
```

图 2. listener.ora 配置文件

2. 打开计算机运行（快捷键“按住 windows 键+R 键”），在运行框中输入“services.msc”

并点击回车，查看 oracle 相关实例服务和监听器服务是否起来。如没有启动，需要手动启动，如图 3。

OracleJobSchedulerXE	禁用	本地系统
OracleMTSRecoveryService	手动	本地系统
OracleServiceXE	已启动	本地系统
OracleXEClrAgent	手动	本地系统
OracleXETNSListener	已启动	本地系统

图 3. oracle 相关服务

## (二) 程序阅读

1. 阅读以下程序，改正错误并进行调试。

```
declare
    v_number NUMBER;
    c_var constant varchar2(20);
begin
    v_number=30;
    v_number=v_number+1;
    c_var='让我们未来相聚尚德湖畔';
    DBMS_OUTPUT.PUT_LINE('当前的变量值是: '||v_number);
    DBMS_OUTPUT.PUT_LINE('当前文本输出: '||c_var);
end;
```

2. 使用%type 和%rowtype 定义变量

- %type: 用户声明单独的变量。
- %rowtype: 用户声明表示表、视图或游标的完整行的记录变量时使用。

使用 scott 模式下 emp 职员表，完成以下程序块编写。

(1) 使用%type 定义变量输出员工号 empno 为 7866 的员工姓名。

(2) 利用%rowtype 改写上述程序代码，输出该学生 7866 员工的详细信息。

## (三) 程序设计

1. 写出打印 9\*9 口诀表的程序（请补充代码或者修改错误，调试程序通过）

```
1*1=1
2*1=2    2*2=4
3*1=3    3*2=6    3*3=9
4*1=4    4*2=8    4*3=12   4*4=16
5*1=5    5*2=10   5*3=15   5*4=20   5*5=25
6*1=6    6*2=12   6*3=18   6*4=24   6*5=30   6*6=36
7*1=7    7*2=14   7*3=21   7*4=28   7*5=35   7*6=42   7*7=49
8*1=8    8*2=16   8*3=24   8*4=32   8*5=40   8*6=48   8*7=56   8*8=64
9*1=9    9*2=18   9*3=27   9*4=36   9*5=45   9*6=54   9*7=63   9*8=72   9*9=81
```

可在 SQL>提示符下输入 set serveroutput on，打开输入输出开关，然后输入脚本，也可在 pl/sql developer 环境下进行脚本验证。

```
declare
```

```

v_number1 NUMBER(3);
v_number2 NUMBER(3);
Begin
  FOR v_number1 IN 1..9
    LOOP
      FOR v_number2 IN _____
        DBMS_OUTPUT.put(v_number1||'*'||v_number2||'='||v_number1*v_number2||' ');
        DBMS_OUTPUT.put_line(' ');
      END LOOP
    END

```

2. 使用分支结构完成对输入员工编号后进行加薪:如果是职员加薪 10%, 销售加薪 15%, 经理加薪 20%,

(1) 在 pl/sql developer 中的命令窗口中进行测试, 使用 set serveroutput on 设置交互响应。

```

DECLARE
  c_manager CONSTANT NUMBER:=0.2;
  c_salesman CONSTANT NUMBER:=0.15;
  c_clerk CONSTANT NUMBER:=0.1;
  v_job VARCHAR(100); --定义职位
BEGIN
  SELECT job into v_job FROM scott.emp WHERE empno=&empno1;-- 查找制定编号员工信息
  IF v_job='CLERK' THEN
    UPDATE scott.emp SET sal=sal*(1+c_clerk) WHERE empno=&empno1;
  elsif v_job='SALSMAN' THEN
    UPDATE scott.emp SET sal=sal*(1+c_salesman) WHERE empno=&empno1;
  elsif v_job='MANAGER' THEN
    UPDATE scott.emp SET sal=sal*(1+c_manager) WHERE empno=&empno1;
  end if;
  --输出加薪后的信息
  DBMS_OUTPUT.put_line('已经为员工 '&empno1||' 成功加薪! ');
EXCEPTION  --处理异常
  WHEN NO_DATA_FOUND THEN
    DBMS_OUTPUT.put_line('没有找到员工数据! ');
END;

```

(2) 以上程序运行有何问题? 如何修改?

(3) 请考虑能否用一条语句完成对所有员工加薪。

## 附 (2) 修改参考

```
DECLARE
c_manager CONSTANT NUMBER:=0.2;
c_salesman CONSTANT NUMBER:=0.15;
c_clerk CONSTANT NUMBER:=0.1;
v_job VARCHAR(100); --定义职位
v_empno number:=&empno;
BEGIN
    SELECT job into v_job FROM scott.emp WHERE empno=v_empno;--查找制定编号
    员工信息
    IF v_job='CLERK' THEN
        UPDATE scott.emp SET sal=sal*(1+c_clerk) WHERE empno=v_empno;
    elsif v_job='SALSMAN' THEN
        UPDATE scott.emp SET sal=sal*(1+c_salesman) WHERE empno=v_empno;
    elsif v_job='MANAGER' THEN
        UPDATE scott.emp SET sal=sal*(1+c_manager) WHERE empno=v_empno;
    end if;
    --输出加薪后的信息
    DBMS_OUTPUT.put_line('已经为员工 '||v_empno||' 成功加薪! ');
    EXCEPTION --处理异常
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.put_line('没有找到员工数据! ');
END;
```

# 实验八 SQL 查询-2

## 一、实验目的

- (一) 掌握各类子查询和连接查询。
- (二)
- (三)

## 二、实验环境

- (一) 硬件设备：计算机局域网，服务器 1 台，终端计算机每人一台
- (二) 软件系统： Windows XP 以上客户机操作系统；  
Oracle11g 服务端数据库系统，客户端工具 PL/SQL developer。

## 三、实验内容

### (一) 实验环境设置

在实验之前，请同学们按照如下步骤查看上机环境是否正常。

1. 所使用的计算机名称（在[\(我的电脑\)](#)图标上点右键查看[\(属性\)](#)中[\(计算机名\)](#)），确保与 oracle 安装路径下的配置文件 tnsnames.ora 和 listener.ora 文件中 host 值保持一致，如下图所示。

```
XE =
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP)(HOST = superb)(PORT = 1521))
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = XE)
  )
)

EXTPROC_CONNECTION_DATA =
(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1))
  )
  (CONNECT_DATA =
    (SID = PLSExtProc)
    (PRESENTATION = RO)
  )
)

ORACLR_CONNECTION_DATA =
(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1))
  )
  (CONNECT_DATA =
    (SID = CLRExtProc)
    (PRESENTATION = RO)
  )
)
```

图 1. tnsnames.ora 配置文件

```
SID_LIST_LISTENER =
(SID_LIST =
  (SID_DESC =
    (SID_NAME = PLSExtProc)
    (ORACLE_HOME = C:\oraclexe\app\oracle\product\11.2.0\server)
    (PROGRAM = extproc)
  )
  (SID_DESC =
    (SID_NAME = CLRExtProc)
    (ORACLE_HOME = C:\oraclexe\app\oracle\product\11.2.0\server)
    (PROGRAM = extproc)
  )
)

LISTENER =
(DESCRIPTION_LIST =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1))
    (ADDRESS = (PROTOCOL = TCP)(HOST = superb)(PORT = 1521))
  )
)

DEFAULT_SERVICE_LISTENER = (XE)
```

图 2. listener.ora 配置文件

2. 打开计算机运行(快捷键“[按住 windows 键+R 键](#)”), 在运行框中输入“[services.msc](#)”

并点击回车，查看 oracle 相关实例服务和监听器服务是否起来。如没有启动，需要手动启动，如图 3。

OracleJobSchedulerXE	禁用	本地系统
OracleMTSRecoveryService	手动	本地系统
OracleServiceXE	已启动	本地系统
OracleXEClrAgent	手动	本地系统
OracleXETNSListener	已启动	本地系统

图 3. oracle 相关服务

## (二) 查询语句实践

请同学们根据 scott 模式有关表，编写查询语句完成以下查询。

1. 利用 HAVING 子句实现查询哪些部门的员工平均工资小于全体员工的平均工资

```
SELECT deptno , AVG(sal) FROM scott.emp GROUP BY deptno
HAVING AVG(sal) < (
    SELECT AVG(sal) FROM scott.emp
);
```

The screenshot shows the SQL window with the following code:

```
SQL 输出 统计
SELECT deptno , AVG(sal) FROM scott.emp GROUP BY deptno
HAVING AVG(sal) < (
    SELECT AVG(sal) FROM scott.emp
);
```

The results table shows the average salary for each department compared to the overall average:

DEPTNO	AVG(SAL)
1	30 2128.57142857143
2	20 2675
3	40 99

Below the table, it says "3 行被选择, 耗时 0.124 秒".

备注：首先找出全体员工的平均工资

2. 利用 some 操作实现获得工资大于任意一个部门的平均工资的员工信息

```
SELECT empno, ename, sal, deptno  FROM scott.emp
where sal>some (
    SELECT AVG(sal) FROM scott.emp
    GROUP BY deptno
);
```

The screenshot shows the SQL window with the following code:

```
SQL 输出 统计
SELECT empno, ename, sal, deptno  FROM scott.emp
where sal>some (
    SELECT AVG(sal) FROM scott.emp
    GROUP BY deptno
);
```

The results table shows employees whose salary is greater than the average salary of any department:

EMPNO	ENAME	SAL	DEPTNO
1	2020 T TURNER	3000.00	15
2	7788 SCOTT	5500.00	20
3	7839 KING	5000.00	10
4	9999 WANGYI	5000.00	30
5	7902 FORD	3000.00	20
6	7566 JONES	2975.00	20
7	7698 BLAKE	2850.00	30
8	7782 CLARK	2450.00	10
9	7844 ADLER	2400.00	30
10	7499 ALLEN	1600.00	30
11	7934 MILLER	1300.00	10
12	7654 MARTIN	1250.00	30
13	7521 WARD	1250.00	30
14	7876 ADAMS	1100.00	20
15	7900 JAMES	950.00	30
16	7369 SMITH	800.00	20

Below the table, it says "16 行被选择, 耗时 0.124 秒".

说明： some 相当于用 or

3. 使用 ALL 操作获得工资大于所有部门的平均工资的员工信息

```
SELECT empno, ename, sal, deptno FROM scott.emp
WHERE sal > ALL (
    SELECT AVG(sal) FROM scott.emp
    GROUP BY deptno );
```

The screenshot shows the SQL window with the following code:

```
1 SELECT empno, ename, sal, deptno FROM scott.emp
2 WHERE sal > ALL (SELECT nvl(AVG(sal),0) FROM scott.emp
3                      GROUP BY deptno
4
5 );
```

Below the code is a result grid displaying four rows of employee data:

	EMPNO	ENAME	SAL	DEPTNO
▶	1	KING	5000.00	10
▶	2	wangyi	5000.00	30
▶	3	SCOTT	5500.00	20
▶	4	Trump	8888.00	10

At the bottom of the window, it says "4 行被选择, 耗时 0.0".

4. 查询每个部门中工资最低的员工信息

```
select empno,ename,sal,deptno from scott.emp
where sal in ( select MIN(sal) from scott.emp
group by deptno )
```

The screenshot shows the SQL window with the following code:

```
1 select empno,ename,sal,deptno from scott.emp
2 where sal in (
3     select MIN(sal) from scott.emp
4         group by deptno
5 )
```

Below the code is a result grid displaying four rows of employee data:

	EMPNO	ENAME	SAL	DEPTNO
▶	1	SMITH	800.00	20
▶	2	JAMES	950.00	30
▶	3	MILLER	1300.00	10
▶	4	Hilarie	99.00	40

At the bottom of the window, it says "4 行被选择, 耗时 0.033 秒".

5. 查询 emp 表各个部门中，哪些员工的工资高于其所在部门的平均工资水平

```
select empno,ename,sal,deptno from scott.emp outer
      where sal > ( select avg(sal) from scott.emp inner
                      where inner.deptno=outer.deptno );
```

The screenshot shows the SQL window with the following code:

```
1 select empno,ename,sal,deptno from scott.emp
2 where sal in (
3     select MIN(sal) from scott.emp
4         group by deptno
5 )
```

Below the code is a result grid displaying four rows of employee data:

	EMPNO	ENAME	SAL	DEPTNO
▶	1	SMITH	800.00	20
▶	2	JAMES	950.00	30
▶	3	MILLER	1300.00	10
▶	4	Hilarie	99.00	40

At the bottom of the window, it says "4 行被选择, 耗时 0.033 秒".

说明：本题为相关子查询，主查询的每一行都执行一次子查询。

6. 将员工编号为 7839 的员工的工资设置为平均工资

```
UPDATE scott.emp  
SET sal = (  
    SELECT AVG(sal) FROM scott.emp  
)  
WHERE empno = 7839 ;
```

7. 删除工作地点在 NEW YORK 的所有员工信息

```
DELETE FROM scott.emp  
WHERE deptno IN (  
    SELECT deptno FROM scott.dept WHERE loc = 'NEW YORK'  
) ;
```

8. 利用 emp 表和 dept 表验证笛卡尔乘积现象

```
SELECT empno , ename, sal , e.deptno , d.deptno , dname  
FROM scott.emp e , scott.dept d  
WHERE e.deptno = d.deptno;
```

9. 利用等值连接查询 ACCOUNTING 部门的员工信息

```
SELECT empno , ename , sal ,d.deptno, dname ,  
FROM scott.emp e inner join scott.dept d  
ON e.deptno=d.deptno  
Where dname='ACCOUNTING';
```

### (三) 选做预习内容

1. 左外连接：结果中除了显示满足连接条件的行外，还显示 JOIN 关键字左侧表中所有满足检索条件的行，备注：可通过在右侧加 (+) 表示做外连接。

```
SELECT empno , ename , sal ,s.grade  
FROM scott.emp e LEFT OUTER JOIN scott.salgrade s  
ON e.sal BETWEEN s.losal AND s.hisal;
```

2. 利用右外连接查询 emp 和 dept 包含的所有部门和编号。右外连接的结果中除了显示满足连接条件的行外，还显示 JOIN 右侧表中所有满足检索条件的行。

```
SELECT distinct e.deptno,d.deptno  
FROM scott.emp e RIGHT OUTER JOIN scott.dept d  
ON e.deptno=d.deptno
```

3. 全外连接：结果中除了显示满足连接条件的行外，还显示JOIN两侧表中所有满足检索条

件的行。

```
SELECT distinct e.deptno,d.deptno  
FROM scott.emp e FULL OUTER JOIN scott.dept d ON e.deptno=d.deptno
```

4. 使用交叉连接查询 emp 表和 dept 表中，部门编号为 10 的员工信息和部门信息

```
Select empno,ename,sal,e.deptno,dname from scott.emp E  
CROSS JOIN scott.dept d  
Where e.deptno=10 and dname='SALES';
```

5. 使用 UNION ALL 操作符，对 scott 用户的 emp 表进行操作，获得员工编号大于 7800 或者所在部门编号为 10 的员工信息。使用 ORDER BY 语句将结果集按照 deptno 列升序排列输出。

```
SELECT EMPNO,ENAME,SAL,DEPTNO  
FROM SCOTT.EMP WHERE EMPNO>7800  
UNION ALL  
Select empno,ename,sal,deptno  
from scott.emp where deptno=10  
Order by deptno ASC;
```

6. 使用 INTERSECT 操作符获得员工编号大于 7800 且所在部门编号为 10 的员工信息

```
SELECT EMPNO,ENAME,SAL,DEPTNO  
FROM SCOTT.EMP WHERE EMPNO>7800  
INTERSECT  
Select empno,ename,sal,deptno  
from scott.emp where deptno=10
```

7. 使用 MINUS 操作符获得员工编号大于 7800 但是所在部门编号不是 10 的员工信息。

```
SELECT EMPNO,ENAME,SAL,DEPTNO  
FROM SCOTT.EMP WHERE EMPNO>7800  
MINUS  
Select empno,ename,sal,deptno  
from scott.emp where deptno=10
```

8. 自连接：是 SQL 语句中经常要用的连接方式，使用自连接可以将自身表的一个镜像当作另一个表来对待，从而能够得到一些特殊的数据。

得到每个员工及其主管名称的明细表

```
SELECT e1.ename||' works for '|e2.ename "Employees and Their Managers"  
FROM scott.emp e1,scott.emp e2  
WHERE e1.mgr = e2.empno  
ORDER BY e1.ename;
```

得到为姓名以 A 的主管打工的员工信息

```
SELECT e1.ename||' works for'||e2.ename "Employees work for Their Managers"  
FROM scott.emp e1,scott.emp e2  
WHERE e1.mgr = e2.empno AND e1.ename LIKE 'A%'  
ORDER BY e1.ename;
```

# 实验八 SQL 查询-2

## 一、实验目的

(一) 掌握各类子查询和相关子查询。

(二) 掌握连接查询。

## 二、实验环境

(一) 硬件设备：计算机局域网，服务器 1 台，终端计算机每人一台

(二) 软件系统： Windows XP 以上客户机操作系统；

Oracle11g 服务端数据库系统，客户端工具 PL/SQL developer。

## 三、实验内容

### (一) 实验环境设置

在实验之前，请同学们按照如下步骤查看上机环境是否正常。

1. 所使用的计算机名称（在〔我的电脑〕图标上点右键查看〔属性〕中〔计算机名〕），确保与 oracle 安装路径下的配置文件 tnsnames.ora 和 listener.ora 文件中 host 值保持一致，如下图所示。

```
XE =
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP)(HOST = superb)(PORT = 1521))
  (CONNECT_DATA =
    | (SERVER = DEDICATED)
    | (SERVICE_NAME = XE)
  )
)

EXTPROC_CONNECTION_DATA =
(DESCRIPTION =
  (ADDRESS_LIST =
    | (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1))
  )
  (CONNECT_DATA =
    | (SID = PLSExtProc)
    | (PRESENTATION = RO)
  )
)

ORACLE_CONNECTION_DATA =
(DESCRIPTION =
  (ADDRESS_LIST =
    | (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1))
  )
  (CONNECT_DATA =
    | (SID = CLRExtProc)
    | (PRESENTATION = RO)
  )
)
```

图 1. tnsnames.ora 配置文件

```
SID_LIST_LISTENER =
(SID_LIST =
  (SID_DESC =
    | (SID_NAME = PLSExtProc)
    | (ORACLE_HOME = C:\oraclexe\app\oracle\product\11.2.0\server)
    | (PROGRAM = extproc)
  )
  (SID_DESC =
    | (SID_NAME = CLRExtProc)
    | (ORACLE_HOME = C:\oraclexe\app\oracle\product\11.2.0\server)
    | (PROGRAM = extproc)
  )
)

LISTENER =
(DESCRIPTION_LIST =
  (DESCRIPTION =
    | (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1))
    | (ADDRESS = (PROTOCOL = TCP)(HOST = superb)(PORT = 1521))
  )
)

DEFAULT_SERVICE_LISTENER = (XE)
```

图 2. listener.ora 配置文件

2. 打开计算机运行(快捷键“[按住 windows 键+R 键](#)”), 在运行框中输入“`services.msc`”并点击回车，查看 oracle 相关实例服务和监听器服务是否起来。如没有启动，需要手动启

动，如图 3。

OracleJobSchedulerXE	禁用	本地系统
OracleMTSRecoveryService	手动	本地系统
OracleServiceXE	已启动	手动
OracleXEClrAgent	手动	本地系统
OracleXETNSListener	已启动	手动

图 3. oracle 相关服务

## (二) 查询语句实践

请同学们根据 scott 模式有关表，编写查询语句完成以下查询。

- 利用 HAVING 子句实现查询哪些部门的员工平均工资小于全体员工的平均工资

```
SELECT deptno , AVG(sal) FROM scott.emp GROUP BY deptno  
HAVING AVG(sal) < (  
    SELECT AVG(sal) FROM scott.emp  
);
```

DEPTNO	AVG(SAL)
30	2128.57142857143
20	2675
40	99

备注：首先找出全体员工的平均工资

- 利用 some 操作实现获得工资大于任意一个部门的平均工资的员工信息

```
SELECT empno, ename, sal, deptno  FROM scott.emp  
where sal>some (  
    SELECT AVG(sal) FROM scott.emp  
    GROUP BY deptno  
);
```

EMPNO	ENAME	SAL	DEPTNO
2020	TRUMAN	8588.00	10
7788	SCOTT	5500.00	20
7839	KING	5000.00	10
9999	WANGYI	5000.00	30
7902	MILLER	3000.00	20
7566	WARD	2975.00	20
7698	BLAKE	2850.00	30
7782	CLARK	2450.00	10
7844	TURNER	2000.00	30
7499	ALLEN	1600.00	30
7934	MILLER	1300.00	10
7654	MARTIN	1250.00	30
7521	WARD	1250.00	20
7876	ADAMS	1100.00	20
7900	JAMES	950.00	30
7369	SMITH	800.00	20

说明：some 相当于用 or

- 使用 ALL 操作获得工资大于所有部门的平均工资的员工信息

```

SELECT empno, ename, sal, deptno FROM scott.emp
WHERE sal > ALL (
    SELECT AVG(sal) FROM scott.emp
    GROUP BY deptno );

```

The screenshot shows the SQL window with the following query:

```

1 SELECT empno, ename, sal, deptno FROM scott.emp
2 WHERE sal > ALL (SELECT nvl(AVG(sal),0) FROM scott.emp
3                      GROUP BY deptno
4
5 );

```

The results table displays four rows of employee data:

	EMPNO	ENAME	SAL	DEPTNO
▶	1	7839 KING	5000.00	10
	2	9999 wangyi	5000.00	30
	3	7788 SCOTT	5500.00	20
	4	2020 Trump	8888.00	10

Execution details at the bottom: 4 行被选择, 耗时 0.0.

#### 4. 查询每个部门中工资最低的员工信息

```

select empno,ename,sal,deptno from scott.emp
where sal in ( select MIN(sal) from scott.emp
                group by deptno )

```

The screenshot shows the SQL window with the following query:

```

1 select empno,ename,sal,deptno from scott.emp
2 where sal in (
3             select MIN(sal) from scott.emp
4             group by deptno
5
)

```

The results table displays four rows of employee data:

	EMPNO	ENAME	SAL	DEPTNO
▶	1	7369 SMITH	800.00	20
	2	7900 JAMES	950.00	30
	3	7934 MILLER	1300.00	10
	4	2016 Hilarie	99.00	40

Execution details at the bottom: 4 行被选择, 耗时 0.033 秒.

#### 5. 查询 emp 表各个部门中，哪些员工的工资高于其所在部门的平均工资水平

```

select empno,ename,sal,deptno from scott.emp outer
        where sal > ( select avg(sal)  from scott.emp inner
                        where inner.deptno=outer.deptno      );

```

The screenshot shows the SQL window with the following query:

```

1 select empno,ename,sal,deptno from scott.emp
2 where sal in (
3             select MIN(sal) from scott.emp
4             group by deptno
5
)

```

The results table displays four rows of employee data:

	EMPNO	ENAME	SAL	DEPTNO
▶	1	7369 SMITH	800.00	20
	2	7900 JAMES	950.00	30
	3	7934 MILLER	1300.00	10
	4	2016 Hilarie	99.00	40

Execution details at the bottom: 4 行被选择, 耗时 0.033 秒.

说明：本题为相关子查询，主查询的每一行都执行一次子查询。

#### 6. 将员工编号为 7839 的员工的工资设置为平均工资

```
UPDATE scott.emp  
SET sal = (  
    SELECT AVG(sal) FROM scott.emp  
)  
WHERE empno = 7839 ;
```

7. 删除工作地点在 NEW YORK 的所有员工信息

```
DELETE FROM scott.emp  
WHERE deptno IN (  
    SELECT deptno FROM scott.dept WHERE loc = 'NEW YORK'  
);
```

8. 利用 emp 表和 dept 表验证笛卡尔乘积现象

```
SELECT empno , ename, sal , e.deptno , d.deptno , dname  
FROM scott.emp e , scott.dept d  
WHERE e.deptno = d.deptno;
```

9. 利用等值连接查询 ACCOUNTING 部门的员工信息

```
SELECT empno , ename , sal ,d.deptno, dname ,  
FROM scott.emp e inner join scott.dept d  
ON e.deptno=d.deptno  
Where dname='ACCOUNTING';
```

### (三) 连接查询

1. 左外连接：结果中除了显示满足连接条件的行外，还显示 JOIN 关键字左侧表中所有满足检索条件的行，备注：可通过在右侧加 (+) 表示做外连接。

```
SELECT empno , ename , sal ,s.grade  
FROM scott.emp e LEFT OUTER JOIN scott.salgrade s  
ON e.sal BETWEEN s.losal AND s.hisal;
```

2. 利用右外连接查询 emp 和 dept 包含的所有部门和编号。右外连接的结果中除了显示满足连接条件的行外，还显示 JOIN 右侧表中所有满足检索条件的行。

```
SELECT distinct e.deptno,d.deptno  
FROM scott.emp e RIGHT OUTER JOIN scott.dept d  
ON e.deptno=d.deptno
```

3. 全外连接：结果中除了显示满足连接条件的行外，还显示JOIN两侧表中所有满足检索条件的行。

```
SELECT distinct e.deptno,d.deptno  
FROM scott.emp e FULL OUTER JOIN scott.dept d ON e.deptno=d.deptno
```

4. 使用交叉连接查询 emp 表和 dept 表中，部门编号为 10 的员工信息和部门信息

```
Select empno,ename,sal,e.deptno,dname from scott.emp E  
CROSS JOIN scott.dept d  
Where e.deptno=10 and dname='SALES';
```

5. 使用 UNION ALL 操作符，对 scott 用户的 emp 表进行操作，获得员工编号大于 7800 或者所在部门编号为 10 的员工信息。使用 ORDER BY 语句将结果集按照 deptno 列升序排列输出。

```
SELECT EMPNO,ENAME,SAL,DEPTNO  
FROM SCOTT.EMP WHERE EMPNO>7800  
UNION ALL  
Select empno,ename,sal,deptno  
from scott.emp where deptno=10  
Order by deptno ASC;
```

6. 使用 INTERSECT 操作符获得员工编号大于 7800 且所在部门编号为 10 的员工信息

```
SELECT EMPNO,ENAME,SAL,DEPTNO  
FROM SCOTT.EMP WHERE EMPNO>7800  
INTERSECT  
Select empno,ename,sal,deptno  
from scott.emp where deptno=10
```

7. 使用 MINUS 操作符获得员工编号大于 7800 但是所在部门编号不是 10 的员工信息。

```
SELECT EMPNO,ENAME,SAL,DEPTNO  
FROM SCOTT.EMP WHERE EMPNO>7800  
MINUS  
Select empno,ename,sal,deptno  
from scott.emp where deptno=10
```

8. 自连接：是 SQL 语句中经常要用的连接方式，使用自连接可以将自身表的一个镜像当作另一个表来对待，从而能够得到一些特殊的数据。

得到每个员工及其主管名称的明细表

```
SELECT e1.ename||" works for "||e2.ename "Employees and Their Managers"  
FROM scott.emp e1,scott.emp e2  
WHERE e1.mgr = e2.empno  
ORDER BY e1.ename;
```

得到为姓名以 A 的主管打工的员工信息

```
SELECT e1.ename||" works for "||e2.ename "Employees work for Their Managers"
```

```
FROM scott.emp e1,scott.emp e2  
WHERE e1.mgr = e2.empno AND e1.ename LIKE 'A%'  
ORDER BY e1.ename;
```

# 实验九 PL/SQL 程序设计 (2)

## 一、实验目的

- (一) 掌握游标的定义和使用。
- (二) 掌握存储过程、函数的定义和使用。

## 二、实验环境

- (一) 硬件设备：计算机局域网，服务器 1 台，终端计算机每人一台
- (二) 软件系统： Windows XP 以上客户机操作系统；  
Oracle11g 服务端数据库系统，客户端工具 PL/SQL developer。

## 三、实验内容

### (一) 实验环境设置

在实验之前，请同学们按照如下步骤查看上机环境是否正常。

1. 所使用的计算机名称（在[我的电脑](#)图标上点右键查看[属性](#)中[\(计算机名\)](#)），确保与 oracle 安装路径下的配置文件 tnsnames.ora 和 listener.ora 文件中 host 值保持一致，如下图所示。

```
XE =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST = superB) (PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = XE)
    )
  )

EXTPROC_CONNECTION_DATA =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC1))
    )
    (CONNECT_DATA =
      (SID = PLSExtProc)
      (PRESENTATION = RO)
    )
  )

ORACLE_CONNECTION_DATA =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC1))
    )
    (CONNECT_DATA =
      (SID = CLRExtProc)
      (PRESENTATION = RO)
    )
  )
)
```

图 1. tnsnames.ora 配置文件

```
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = PLSExtProc)
      (ORACLE_HOME = C:\oraclexe\app\oracle\product\11.2.0\server)
      (PROGRAM = extproc)
    )
    (SID_DESC =
      (SID_NAME = CLRExtProc)
      (ORACLE_HOME = C:\oraclexe\app\oracle\product\11.2.0\server)
      (PROGRAM = extproc)
    )
  )

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC1))
      (ADDRESS = (PROTOCOL = TCP) (HOST = superB) (PORT = 1521))
    )
  )

DEFAULT_SERVICE_LISTENER = (XE)
```

图 2. listener.ora 配置文件

2. 打开计算机运行（快捷键“[按住 windows 键+R 键](#)”），在运行框中输入“[services.msc](#)”并点击回车，查看 oracle 相关实例服务和监听器服务是否起来。如没有启动，需要手动启

动，如图 3。

OracleJobSchedulerXE	禁用	本地系统
OracleMTSRecoveryService	手动	本地系统
OracleServiceXE	已启动	手动
OracleXEClrAgent	手动	本地系统
OracleXETNSListener	已启动	手动

图 3. oracle 相关服务

## (二) 游标的使用。

掌握显式游标的定义与使用，以 scott 模式 emp 表为例结合游标属性编写代码。在 file/new/test windows 或者 sql windows 测试语句。

- %ISOPEN: 游标是否打开，打开后返回 true
- %FOUND : 检测提取行数据情况，提出数据返回真。
- %NOTFOUND: 未提出数据，返回真
- %ROWCOUNT: 累计到目前为止使用 fetch 提取的行数

1. 简单游标使用。请大家完成有关代码，调通程序。

```
declare
    tname _____;
cursor cur(id varchar2)
is
    select ename from student where empno=id;
begin
    if cur%ISOPEN then
        DBMS_OUTPUT.PUT_LINE('游标已经被打开！');
    else
        open cur('2020');
    end if;
    fetch cur into tname;
    close cur;
    DBMS_OUTPUT.PUT_LINE('名字是:' || tname);
end;
```

运行结果：根据大家使用数据不同结果可能不同。

2. 用FOR循环改写下列程序，实现数据输出。

```
declare
    tname varchar2(10);
cursor mycur
is
    select ename from scott.emp;
begin
    open mycur;
    loop
        fetch mycur into tname;
        exit when mycur%NOTFOUND or mycur%notfound is null;
        dbms_output.put_line('游标的 rowcount 值是:' || mycur%rowcount);
    end loop;
    close mycur;
end;
```

## (三) 存储过程的定义和使用

使用的 scott 模式中的 emp 表，改写并完善以下给定代码。

- (1) 无参存储过程：阅读代码实现修改 emp 表中 ename 字段值，下面代码可供参考，需要根据上机实例进行修改。

```
create or replace procedure update_emp
AS
begin
    update scott.emp SET ename= '乔布斯'
    where empno='2020';
end update_stu;
/
```

- (2) 含有输入参数存储过程：对上面（1）进行有参改写，均为输入参数。

```
Create or replace procedure update_emp(temp_id number,temp_name in varchar2)
AS
begin
    update emp SET ename=temp_name where empno=temp_id;
end update_emp;
/
```

对比一下调用方法优缺点：

- exec update\_emp(2020,'乔布斯')
- exec update\_emp('ename=>'乔布斯',empno =>2020)

- (3) 编写过程 enquiry\_dept，实现向 dept 表添加新记录时，反馈是否已存在相同编号。

```
CREATE PROCEDURE enquiry_dept (
    t_deptno      dept.deptno%TYPE,
    t_dname       dept.dname%TYPE,
    t_loc         dept.loc%TYPE
)
AS
    deptcount    NUMBER;           -- 保存查询员工编号计数
BEGIN
    SELECT COUNT (*) INTO deptcount FROM dept WHERE deptno = t_deptno;
    IF deptcount > 0     THEN
        raise_application_error (-20001, '已经存在相同编号的员工！');
    END IF;
    INSERT INTO dept(deptno, dname, loc)  VALUES (t_deptno, t_dname, t_loc);
    COMMIT;
END;
```

- (4) 练习利用 out 参数实现存储过程的返回值

```
Create or replace procedure select_emp
(temp_id in varchar2, temp_name OUT varchar2) AS
begin
    select ename INTO temp_name from emp where empno=temp_id;
end select_emp;
```

/

- sql>variable temp\_ename varchar2(10);
- sql>Exec select\_emp(2020,:temp\_ename);
- sql>print temp\_ename;

运行结果： 乔 丹

#### (四) 函数的定义与使用

利用 emp 表实现根据给定员工编号得到员工名字的函数。

```
Create or replace Function get_ename(temp_id number)
return varchar2
AS
temp_name scott.emp.ename%TYPE;
BEGIN
    select ename into temp_name from scott.emp where empno=temp_id;
    return temp_name;
END get_ename;
```

调用方法： select get\_ename(7839) from dual;

# 实验十、十一 SQL 查询&PL/SQL 程序设计（2）

## 一、实验目的

- (一) 掌握 rownum 与 rowid 的含义和使用。
- (二) 熟悉掌握存储过程、函数的定义和使用。
- (三) 掌握包的定义和使用。

## 二、实验环境

- (一) 硬件设备：计算机局域网，服务器 1 台，终端计算机每人一台
- (二) 软件系统： Windows XP 以上客户机操作系统；  
Oracle11g 服务端数据库系统，客户端工具 PL/SQL developer。

## 三、实验内容

### (一) 实验环境设置

在实验之前，请同学们按照如下步骤查看上机环境是否正常。

1. 所使用的计算机名称（在[我的电脑](#)图标上点右键查看[属性](#)中[\(计算机名\)](#)），确保与 oracle 安装路径下的配置文件 tnsnames.ora 和 listener.ora 文件中 host 值保持一致，如下图所示。

```
XE =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST = superb) (PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = XE)
    )
  )

  EXTPROC_CONNECTION_DATA =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC1))
    )
    (CONNECT_DATA =
      (SID = PLSExtProc)
      (PRESENTATION = RO)
    )
  )

  ORACLE_CONNECTION_DATA =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC1))
    )
    (CONNECT_DATA =
      (SID = CLRExtProc)
      (PRESENTATION = RO)
    )
  )
```

图 1. tnsnames.ora 配置文件

```
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = PLSExtProc)
      (ORACLE_HOME = C:\oraclexe\app\oracle\product\11.2.0\server)
      (PROGRAM = extproc)
    )
    (SID_DESC =
      (SID_NAME = CLRExtProc)
      (ORACLE_HOME = C:\oraclexe\app\oracle\product\11.2.0\server)
      (PROGRAM = extproc)
    )
  )

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC1))
      (ADDRESS = (PROTOCOL = TCP) (HOST = superb) (PORT = 1521))
    )
  )

DEFAULT_SERVICE_LISTENER = (XE)
```

图 2. listener.ora 配置文件

2. 打开计算机运行（快捷键[按住 windows 键+R 键](#)），在运行框中输入[“services.msc”](#)

并点击回车，查看 oracle 相关实例服务和监听器服务是否起来。如没有启动，需要手动启动，如图 3。

OracleJobSchedulerXE	禁用	本地系统
OracleMTSRecoveryService	手动	本地系统
OracleServiceXE	已启动	本地系统
OracleXEClrAgent	手动	本地系统
OracleXETNSListener	已启动	本地系统

图 3. oracle 相关服务

## (二) rownum 与开窗函数

1. 查找员工信息及部门名称信息按部门排序，显示 emp 表 rowid 与 rownum 字段。进一步深入理解 rownum 用法。

2. 查询 emp 中工资排名前 5 名的员工信息，工资按照由高到低排序。

3. 查询 emp 表中工资排名第 5-10 名的员工姓名、工资，工资按照由高到低排序。

4. 使用开窗函数查询 emp 中各部门的工资总和。

开窗函数：

指定了分析函数工作的数据窗口大小，这个数据窗口大小可能会随着行的变化而变化。

开窗聚集函数 over (partition by ... order by ...), 开窗聚集函数可以是: row\_number(), sum(), count(\*), avg() 等。

## (三) 存储过程的定义和使用

创建一个用户表 Users，字段包括四个字段（序号、姓名、用户类型、密码）其中各字段约束要求如下：

- (1) 序号为整数值，作为主键。
- (2) 姓名长度为 8 个字符，不能重复；
- (3) 密码为 8 位字符型。

1. 创建测试用数据表

```
create table users(  
    UserId number primary key,  
    UserName varchar2(8) unique not null,  
    UserPwd varchar2(8)  
)
```

2. 插入若干条测试数据，格式如下：

```
insert into users values(1,'王自健','wang');  
insert into users values(2,'郭德纲','guo');  
insert into users values(3,'周立波','zhou');  
insert into users values(4,'乔布斯','jack');
```

使用 select 语句查看表中数据是否已经存在

3. 写出新用户注册的存储过程 user\_reg，用户注册时，判断用户名是否重复，如果注册

成功，给出提示，不成功，给出系统提示“您的用户名已存在！”。

(1) 使用序列作为添加新用户时产生用户编号

```
create sequence UserId_seq
start with 1
increment by 1
minvalue 1 maxvalue 99
nocycle
order
nocache
```

(2) 创建存储过程

```
create procedure user_reg( user_name varchar2, user_pwd varchar2,
                           user_prompt out varchar2)
is
  t_user users%rowtype;
begin
  select * into t_user from users where UserName=user_name;
  user_prompt := '您的用户名已经存在';
  EXCEPTION
    when NO_DATA_FOUND then
      insert into users values (UserId_seq.Nextval,user_name,user_pwd);
      user_prompt := '注册成功，您的帐号是:'||user_name;
      commit;
end user_reg;
```

检验一下存储过程工作情况

```
variable temp_stu_name varchar2(20);
exec user_reg( '杨绛', '2016', :temp_stu_name );
print temp_stu_name;
```

运行界面：

```
SQL> exec user_reg('杨绛','2016',:temp_stu_name);
PL/SQL procedure successfully completed
temp_stu_name
-----
注册成功，您的帐号是:杨绛

SQL> print temp_stu_name;
temp_stu_name
-----
注册成功，您的帐号是:杨绛
```

注意：请大家考虑是否有其他改写方法。

```
create procedure user_reg( user_name varchar2, user_pwd varchar2,
                           user_prompt out varchar2)
is
  t_user users%rowtype;
begin
  insert into users values (UserId_seq.Nextval,user_name,user_pwd);
  user_prompt := '注册成功，您的帐号是:'||user_name;
```

```

exception
when DUP_VAL_ON_INDEX then
    user_prompt := '您的用户名已经存在！';
end user_reg;

```

4. 创建函数在用户登录时判断用户账号是否存在，如果账户名和密码都正确允许登录。

```

create procedure user_login (user_name varchar2,
                            user_pwd varchar2,
                            user_prompt out varchar2)
is
    t_user users%rowtype;
begin
    select * into t_user from users where UserName=user_name;
    if t_user.UserPwd =user_pwd then
        user_prompt := '登录成功！';
    else
        user_prompt := '密码错误，不能登录！';
    end if;
exception
    when no_data_found then
        user_prompt := '账户不存在！';
end user_login;

```

验证以上代码

```

variable temp_prompt varchar2(20);
exec user_login( '郭德纲' , 'guo' , :temp_prompt);
print temp_prompt;

```

运行结果：

```

SQL> variable temp_prompt varchar2(20);
SQL> exec user_login( '郭德纲' , 'guo' , :temp_prompt);
PL/SQL procedure successfully completed
temp_prompt
-----
登录成功

SQL> print temp_prompt;
temp_prompt
-----
登录成功

```

5. 创建修改密码的存储过程 user\_update\_pwd，要求输入原密码，检查密码是否匹配以后再进行更新操作。

```

create or replace procedure user_update_pwd( username users.UserName%type,
                                            useroldpwd users.UserPwd%type,
                                            usernewpwd users.UserPwd%type,
                                            user_prompt out varchar2)
is
    temp_pwd users.UserPwd%type;

```

```

begin
    select UserPwd into temp_pwd from users where UserName = username;
    dbms_output_puntline(temp_pwd);
    if (temp_pwd=useroldpwd ) then
        update users set UserPwd=usernewpwd where UserName=username;
        user_prompt := '密码修改成功！';
    else
        user_prompt := '原密码错误！';
    end if;
end user_update_pwd;
/

```

验证以上存储过程，将郭德纲的密码改为‘gang’

```

variable temp_prompt varchar2(20);
exec user_update_pwd( '郭德纲' , 'guo' , 'gang' , :temp_prompt );
print temp_prompt;

```

运行图如下：

```

SQL> variable temp_prompt varchar2(20);
SQL> exec user_update_pwd( '郭德纲' , 'guo' , 'gang' , :temp_prompt );
PL/SQL procedure successfully completed
temp_prompt
-----
密码修改成功！

```

### (三) 包的创建和使用，包由规范和包体构成

根据已实现的一个存储过程和一个函数，按照包的定义将该存储过程和函数打包。

过程 add\_stu\_proc:

```

create or replace procedure add_stu_proc(stu_id number,stu_number char,stu_name
varchar2,stu_sex char,stu_deptid number)
is
begin
    insert into student values(stu_id,stu_number,stu_name,stu_sex,stu_deptid);
end add_stu_proc;

```

函数find\_stu\_func:

```

create or replace function find_stu_func(stu_id number)
return varchar2 is
    stu_name student.studentname%TYPE;
begin
    select studentname into stu_name from student where studentid=stu_id;
    return stu_name;
end find_stu_func;

```

制作包

第一步：包规范

```

Create or replace package stu_package
as

```

```
procedure add_stu_proc(stu_id number,stu_number char,stu_name varchar2,stu_sex
char,stu_deptid number);
FUNCTION find_stu_func(stu_id number) return varchar2;
End stu_package;
```

第二步：包体

```
Create or replace package body stu_package IS
procedure add_stu_proc(stu_id number,stu_number char,stu_name varchar2,stu_sex
char,stu_deptid number) is
begin
    insert into student values(stu_id,stu_number,stu_name,stu_sex,stu_deptid);
end add_stu_proc;

function find_stu_func(stu_id number) return varchar2 is
    stu_name student.studentname%TYPE;
begin
    select studentname into stu_name from student where studentid=stu_id;
    return stu_name;
end find_stu_func;
End stu_package;
```

第三步：命令窗口下调用包

```
EXEC stu_package.add_stu_proc(20081039,'121212','Mike Dell. ','男',20);
```

# 实验十二 PL/SQL 程序设计（触发器）

## 一、实验目的

- (一) 掌握触发器定义与使用。
- (二) 通过触发器与序列的综合应用解决实际数据库问题。

## 二、实验环境

- (一) 硬件设备：计算机局域网，服务器 1 台，终端计算机每人一台
- (二) 软件系统：Windows XP 以上客户机操作系统；  
Oracle11g 服务端数据库系统，客户端工具 PL/SQL developer。

## 三、实验内容

### (一) 实验环境设置

在实验之前，请同学们按照如下步骤查看上机环境是否正常。

1. 所使用的计算机名称（在[我的电脑](#)图标上点右键查看[属性](#)中[\[计算机名\]](#)），确保与 oracle 安装路径下的配置文件 tnsnames.ora 和 listener.ora 文件中 host 值保持一致，如下图所示。

```
XE =
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP) (HOST = superB) (PORT = 1521))
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = XE)
  )
)

EXTPROC_CONNECTION_DATA =
(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC1))
  )
  (CONNECT_DATA =
    (SID = PLSExtProc)
    (PRESENTATION = RO)
  )
)

ORACLE_CONNECTION_DATA =
(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC1))
  )
  (CONNECT_DATA =
    (SID = CLRExtProc)
    (PRESENTATION = RO)
  )
)
```

图 1. tnsnames.ora 配置文件

```
SID_LIST LISTENER =
(SID_LIST =
  (SID_DESC =
    (SID_NAME = PLSExtProc)
    (ORACLE_HOME = C:\oraclexe\app\oracle\product\11.2.0\server)
    (PROGRAM = extproc)
  )
  (SID_DESC =
    (SID_NAME = CLRExtProc)
    (ORACLE_HOME = C:\oraclexe\app\oracle\product\11.2.0\server)
    (PROGRAM = extproc)
  )
)

LISTENER =
(DESCRIPTION_LIST =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC1))
    (ADDRESS = (PROTOCOL = TCP) (HOST = superB) (PORT = 1521))
  )
)

DEFAULT_SERVICE_LISTENER = (XE)
```

图 2. listener.ora 配置文件

2. 打开计算机运行（快捷键“[按住 windows 键+R 键](#)”），在运行框中输入“[services.msc](#)”并点击回车，查看 oracle 相关实例服务和监听器服务是否起来。如没有启动，需要手动启

动，如图 3。

OracleJobSchedulerXE	禁用	本地系统
OracleMTSRecoveryService	手动	本地系统
OracleServiceXE	已启动	手动
OracleXEClrAgent	手动	本地系统
OracleXETNSListener	已启动	手动

图 3. oracle 相关服务

## (二) 触发器及其使用

掌握下列四种触发器的定义和使用。

DML 触发器 (insert、update、delete)

DDL 触发器 (create、alter、drop)

系统事件触发器 (start、shutdown)

Instead of 触发器

- 利用触发器实现对用户修改的某个表的字段其原始内容和修改时间自动记录到 record 表。下述代码需要根据上机数据库进行修改。

(1) 首先创建要记录内容的表 record，共有两个字段：

```
Create table record(  
    content varchar2(100),    -- 修改内容  
    rtime timestamp          -- 修改时间  
)
```

(2) 对 student 表创建一个触发器

```
Create or replace trigger update_stu_trigger  
after update on student  
for EACH ROW  
Begin  
    Insert into record values('执行update前数据是  
studentid='||:old.studentid||',studentname='||:old.studentname||'sex='||:old.studentsex||',deptid='||:  
old.deptid,sysdate);  
End update_stu_trigger;
```

(3) 调用法： update student set studentname='我是谁' where studentid='20081003';

2.DDL 触发器:当创建数据表时，自动对创建者信息记录到 my\_create\_log 表中。

首先创建上述表，

```
Create table my_create_log(username varchar2(10),  
    objname varchar2(30),objtype varchar2(30),create_date timestamp);
```

其次建立触发器

```
Create or replace trigger my_create_trigger  
after create  
on database  
BEGIN
```

```
Insert into my_create_log values(user,sys.dictionary_obj_name,sys. dictionary_obj_type,sysdate);
```

```
END create_trigger;
```

然后创建一实例数据表

```
createtable test(test varchar2(10));
```

最后查看记录表

```
select * from my_create_log;
```

3.系统事件触发器：某一系统事件后引起触发，例如登陆数据库后自动记录登陆者和登陆时间等信息。

```
create logon_table(username varchar2(10),time timestamp)
```

```
create trigger logon_on_trigger  
after logon  
on database  
Begin  
    Insert into logon_table values(user,sysdate);  
End logon_trigger;
```

### (三) 触发器综合训练

创建一个表 student(*stuid*,*name*,*birthday*,*sex*,*deptid*)，其中 *stuid* 为主键，数值型，*name* 非空，*birthday* 为日期型，*sex* 取值仅能为男或女，*deptid* 为外键，外键引用自系别表 dept(*deptid*,*dname*,*loc*)，请完成：

1. 写出创建表 student 表和 dept 表的 SQL 语句；
2. 定义一个触发器，实现用户添加数据时主键值自动生成。
3. 定义一个触发器，实现用户进行数据修改时，记录用户信息、修改时间和两个表信息到一个新表 record 中。

四、实验要求：综合训练需要提交实验报告（雨课堂布置）

# 实验十三 SQL 查询优化

## 一、实验目的

- (一) 掌握查询计划的使用;
- (二) 掌握常见的 SQL 优化方法。

## 二、实验环境

- (一) 硬件设备: 计算机局域网, 服务器 1 台, 终端计算机每人一台
- (二) 软件系统: Windows XP 以上客户机操作系统;  
Oracle11g 服务端数据库系统, 客户端工具 PL/SQL developer。

## 三、实验内容

### (一) 实验环境设置

在实验之前, 请同学们按照如下步骤查看上机环境是否正常。

1. 所使用的计算机名称 (在 (我的电脑) 图标上点右键查看 (属性) 中 (计算机名)), 确保与 oracle 安装路径下的配置文件 tnsnames.ora 和 listener.ora 文件中 host 值保持一致, 如下图所示。

```
XE =
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP) (HOST = superb) (PORT = 1521))
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = XE)
  )
)

EXTPROC_CONNECTION_DATA =
(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC1))
  )
  (CONNECT_DATA =
    (SID = PLSExtProc)
    (PRESENTATION = RO)
  )
)

ORACLE_CONNECTION_DATA =
(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC1))
  )
  (CONNECT_DATA =
    (SID = CLRExtProc)
    (PRESENTATION = RO)
  )
)
```

图 1. tnsnames.ora 配置文件

```
SID_LIST_LISTENER =
(SID_LIST =
  (SID_DESC =
    (SID_NAME = PLSExtProc)
    (ORACLE_HOME = C:\oraclexe\app\oracle\product\11.2.0\server)
    (PROGRAM = extproc)
  )
  (SID_DESC =
    (SID_NAME = CLRExtProc)
    (ORACLE_HOME = C:\oraclexe\app\oracle\product\11.2.0\server)
    (PROGRAM = extproc)
  )
)

LISTENER =
(DESCRIPTION_LIST =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC1))
    (ADDRESS = (PROTOCOL = TCP) (HOST = superb) (PORT = 1521))
  )
)

DEFAULT_SERVICE_LISTENER = (XE)
```

图 2. listener.ora 配置文件

2. 打开计算机运行 (快捷键“**按住 windows 键+R 键**”), 在运行框中输入“**services.msc**”并点击回车, 查看 oracle 相关实例服务和监听器服务是否起来。如没有启动, 需要手动启

动，如图 3。

OracleJobSchedulerXE	禁用	本地系统
OracleMTSRecoveryService	手动	本地系统
OracleServiceXE	已启动	手动
OracleXEClrAgent	手动	本地系统
OracleXETNSListener	已启动	手动

图 3. oracle 相关服务

## (二) 执行计划

Oracle 数据库中执行计划是一条查询语句在 ORACLE 中的执行过程或访问路径的描述，可通过查看执行计划进行 SQL 语句优化以提升性能。通过执行计划可以看到 SQL 的执行路径，逻辑读，物理读等信息，根据这些信息判断 SQL 优化情况。

1. 查看执行计划：查看执行计划可参照如下步骤进行：

- (1) 设置执行计划开关：set autotrace on
- (2) SQL>EXPLAIN PLAN FOR **你编写的 sql 语句**;
- (3) select \* from table (*dbms\_xplan.display\_cursor()*) ;显示 SQL 语句的真实执行计划

2. 有关数据字典：使用 desc 查看数据字典的结构，

(1) v\$sql : 存储的是具体的 SQL 和执行计划相关信息，常见字段如下，其他字段请大家自行查看。

- SQL\_TEXT 和 SQL\_FULLTEXT: 字段表示当前正在执行的 SQL 文本
- EXECUTIONS 在共享池内执行次数
- OPTIMIZER\_COST 优化器对于 sql 给出的成本
- HASH\_VALUES 解析产生的哈希值
- SERVICE 服务名
- CPU\_TIME 执行和获取数据使用的 CPU 时间（毫秒）
- ELAPSED\_TIME SQL 执行时间（毫秒）

查询数据字典 v\$sql，以 **sys 身份登录** 后了解关于 SQL 语句的执行耗費情况。

```
select hash_value,service,executions,optimizer_cost from v$sql
```

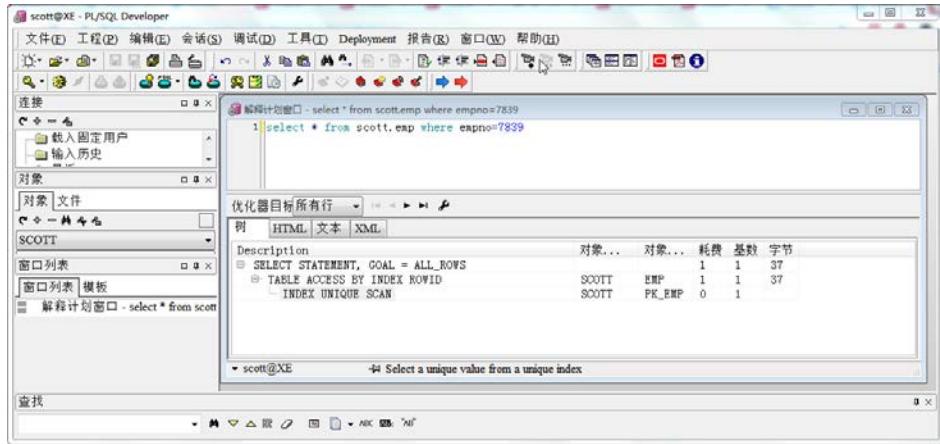
```
where sql_text like '%SELECT * FROM SCOTT.EMP%';
```

HASH_VALUE	SERVICE	I	EXECUTIONS	OPTIMIZER_COST
1436427812 XE			1	1
905985831 XE			1	1
1486636396 XE			1	1
4094449978 XE			5	1
1015791636 XE			2	2
1275322486 XE			1	1
6 rows selected				

以下数据字典使用，请大家自行练习。

- (2) v\$sql\_plan: 使用查询语句获得已执行的，尚在内存中的 sql 语句执行计划。
- (3) v\$sqltext : 存储完整的 SQL，sql 被分割存储；
- (4) v\$sqlarea: 存储的 SQL 和一些相关的信息，比如累计的执行次数，逻辑读，物理读等统计信息；

3. 在 PL/SQL 中查看执行计划，在文件中选择新建后，选择解释计划窗口。输入 sql 语句后，执行，如图。



### (三) 常见的 SQL 优化技术练习

以 scott 模式下 emp 表和 dept 表为例, 改写完成以下 SQL 语句的优化验证, 对比时间耗费以及采用的执行计划。

`set autotrace on` --包含执行计划和统计信息

`set timing on` --设置时间显示

1. 查询时以具体字段替代使用\*

```
select * from scott.emp;
```

2. 利用 where 子句代替 having 子句

```
SELECT deptno,avg(sal) FROM scott.emp GROUP BY deptno HAVING deptno > 10;
```

3. 建立一个 test 表, 插入若干记录体验 truncate 替代 delete 语句效果,truncate 删除后不能恢复。

```
delete from test
```

```
truncate table test
```

4. 保持数据完整性条件下多用 commit 语句

自己写一个 SQL 测试

5. 用表连接替代多表查询, 将行记录较少的表连接放到后面, 作为驱动表。

```
select empno,ename,deptno from scott.emp where
deptno=(select deptno from scott.dept where dname='ACCOUNTING')
```

6. 使用 exists 替代 in

```
select * from scott.emp where deptno in ( select deptno from dept where deptname like
'%ACCOUNTING' )
```

7. 使用 exists 替代 distinct

```
select distinct e.depno,d.dname from emp e,dept d where
s.depno=d.depno;
```

8. 使用<=替代<,将 7935 换成 7934

```
select * from scott.emp where empno<7935
```

9. 使用完全限定列法

```
select empno,ename,mgr,sal, deptno,dname from emp,dept
```

10. 表连接中 from 子句顺序，由右至左处理，最右的表作为驱动表。

```
select e.empno,e.ename,e.sal,d.deptno,d.dname  
from dept d, emp e ;
```

11. where 子句中顺序:右至左解析，本例使用 scott.emp

```
select * from scott.emp e where sal>1000 and job='MANAGER'  
and 3<(select count(*) from scott.emp where mgr=7839)
```

12. 创建索引，提高查询效率.对比有无索引时查询效率。

```
create index stuid_index on student(studentid)
```

## 实验十四 数据库连接

### 一、实验目的

1. 掌握 ADO.NET 数据库访问技术。
2. 掌握 JDBC 数据库连接技术。

### 二、实验环境

1. 硬件设备：计算机局域网，服务器 1 台，终端计算机每人一台
2. 软件系统：Windows 2000/XP 客户机操作系统； Oracle11g 服务端数据库系统，客户端工具。

### 三、实验内容

#### (一) 实验环境设置

在实验之前，请同学们按照如下步骤查看上机环境是否正常。

1. 所使用的计算机名称（在（我的电脑）图标上点右键查看（属性）中（计算机名）），确保与 oracle 安装路径下的配置文件 tnsnames.ora 和 listener.ora 文件中 host 值保持一致，如下图所示。

```
XE =
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP)(HOST = superb)(PORT = 1521))
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = XE)
  )
)

EXTPROC_CONNECTION_DATA =
(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1))
  )
  (CONNECT_DATA =
    (SID = PLSExtProc)
    (PRESENTATION = RO)
  )
)

ORACLR_CONNECTION_DATA =
(DESCRIPTION =
  (ADDRESS_LIST =
    (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1))
  )
  (CONNECT_DATA =
    (SID = CLRExtProc)
    (PRESENTATION = RO)
  )
)
```

图 1. tnsnames.ora 配置文件

```
SID_LIST_LISTENER =
(SID_LIST =
  (SID_DESC =
    (SID_NAME = PLSExtProc)
    (ORACLE_HOME = C:\oraclexe\app\oracle\product\11.2.0\server)
    (PROGRAM = extproc)
  )
  (SID_DESC =
    (SID_NAME = CLRExtProc)
    (ORACLE_HOME = C:\oraclexe\app\oracle\product\11.2.0\server)
    (PROGRAM = extproc)
  )
)

LISTENER =
(DESCRIPTION_LIST =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1))
    (ADDRESS = (PROTOCOL = TCP)(HOST = superb)(PORT = 1521))
  )
)

DEFAULT_SERVICE_LISTENER = (XE)
```

图 2. listener.ora 配置文件

2. 打开计算机运行（快捷键“[按住 windows 键+R 键](#)”），在运行框中输入“`services.msc`”

并点击回车，查看 oracle 相关实例服务和监听器服务是否起来。如没有启动，需要手动启动，如图 3。

OracleJobSchedulerXE	禁用	本地系统
OracleMTSRecoveryService	手动	本地系统
OracleServiceXE	已启动	本地系统
OracleXEClrAgent	手动	本地系统
OracleXETNSListener	已启动	本地系统

图 3. oracle 相关服务

## (二) ADO.NET 对象简介

Connection: 连接对象，根据连接的字符串连接数据库，通过打开和关闭两个方法使用数据库。

Command: 命令对象，通过设置 sql 语句实现增删改操作。

DataReader: 数据读取对象，接收 command 命令查询得到的结果，获取只读数据，仅用于显示。仅能用 Command 对象的 ExecuteReader() 对象创建。

DataAdapter: 数据库适配器对象，把查询结果填充到 DataSet 数据集对象中。是 Connection 对象和 DataSet 对象的桥梁。

DataSet: 数据集对象，存储数据的虚拟表。

## (三) 通过编写代码实现数据库访问

```
private void Form1_Load(object sender, EventArgs e)
{
    OracleConnection conn = new OracleConnection(@"data source=orcl;user id=scott;password=tiger;unicode=TRUE");
    conn.Open();
    OracleCommand cmd = new OracleCommand();
    cmd.CommandText = "SELECT * FROM scott.emp ORDER BY empno";
    cmd.CommandType = CommandType.Text;
    cmd.Connection = conn;
    OracleDataReader dr = cmd.ExecuteReader();
    BindingSource bs = new BindingSource();
    bs.DataSource=dr;
    dataGridView1.DataSource = bs;
}
```



## (四) JDBC 方式

### 1. 数据库驱动类加载

Java 连接数据库前，将数据库驱动类加载到 JVM 中，实现在 JDBC 管理器中的注册。

```
Class.forName("Oracle.jdbc.driver.OracleDriver");
```

Class.forName() 为 Java 注册类语句。

### 2. 建立数据库连接

确定连接目标 URL 地址，格式为： JDBC 协议：域名或 IP 地址：端口号/数据库名称）

```
static String URL="jdbc:oracle:thin:@localhost:1521:hsb"
3. 获取数据库连接对象
connection conn=DriverManager.getConnection(url,username,password)
4. 创建数据库 SQL 对象
Statement st=conn.createStatement();
5. 向 oracle 发送 SQL 执行语句
```

```
String sql="select * from scott.emp"
Boolean value=st.executeQuery(sql);
6. 结果集遍历
```

```
getString(string columnLabel)
getInt(string columnLabel)
getData(string columnLabel)
```

```
7. 对象关闭
调用 Close()关闭所创建的各类对象， 释放资源。
```

#### (五) JDBC 参考代码

```
package jdbcHomework;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;

public class dbUtil {
    public static Connection getConnection(){
        Connection conn=null;

        try {
            String url="jdbc:oracle:thin:@127.0.0.1:1521:orcl";
            String user="scott";
            String password="tiger";

            Class.forName("oracle.jdbc.driver.OracleDriver");//加载数据驱动
            conn = DriverManager.getConnection(url, user, password);// 连接数据库

        } catch (ClassNotFoundException e) {
            e.printStackTrace();
            System.out.println("加载数据库驱动失败");
        }catch(Exception e){
            e.printStackTrace();
            System.out.println("连接数据库失败");
        }
    }
}
```

```
    }
    return conn;
}
public static void close(Connection conn, PreparedStatement ps, ResultSet rs){
    try {
        if(rs!=null){
            rs.close();
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }

    try {
        if(ps!=null){
            ps.close();
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }

    try {
        if(conn!=null){
            conn.close();
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

}
```

## 实验十五 数据库备份与恢复

### 一、实验目的

1. 掌握 imp、exp 的使用，掌握备份和恢复表、用户模式、数据库。
2. 掌握联机备份和恢复方法。
3. 了解 RMAN 的使用。

### 二、实验环境

1. 硬件设备：计算机局域网，服务器 1 台，终端计算机每人一台
2. 软件系统：Windows 2000/XP 客户机操作系统； Oracle11g 服务端数据库系统，客户端工具。

### 三、实验内容

#### (一) 实验环境设置

在实验之前，请同学们按照如下步骤查看上机环境是否正常。

1. 所使用的计算机名称（在[我的电脑](#)图标上点右键查看[属性](#)中[\(计算机名\)](#)），确保与 oracle 安装路径下的配置文件 tnsnames.ora 和 listener.ora 文件中 host 值保持一致，如下图所示。

```
XE =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP) (HOST = superb) (PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = XE)
    )
  )

EXTPROC_CONNECTION_DATA =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC1))
    )
    (CONNECT_DATA =
      (SID = PLSExtProc)
      (PRESENTATION = RO)
    )
  )

ORACLR_CONNECTION_DATA =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC1))
    )
    (CONNECT_DATA =
      (SID = CLRExtProc)
      (PRESENTATION = RO)
    )
  )
```

图 1.tnsnames.ora 配置文件

```
SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = PLSExtProc)
      (ORACLE_HOME = C:\oraclexe\app\oracle\product\11.2.0\server)
      (PROGRAM = extproc)
    )
    (SID_DESC =
      (SID_NAME = CLRExtProc)
      (ORACLE_HOME = C:\oraclexe\app\oracle\product\11.2.0\server)
      (PROGRAM = extproc)
    )
  )

LISTENER =
  (DESCRIPTION_LIST =
    (DESCRIPTION =
      (ADDRESS = (PROTOCOL = IPC) (KEY = EXTPROC1))
      (ADDRESS = (PROTOCOL = TCP) (HOST = superb) (PORT = 1521))
    )
  )

DEFAULT_SERVICE_LISTENER = (XE)
```

图 2.listener.ora 配置文件

2. 打开计算机运行(快捷键[按住 windows 键+R 键](#))，在运行框中输入[“services.msc”](#)

并点击回车，查看 oracle 相关实例服务和监听器服务是否起来。如没有启动，需要手动启动，如图 3。

OracleJobSchedulerXE	禁用	本地系统
OracleMTSRecoveryService	手动	本地系统
OracleServiceXE	已启动	本地系统
OracleXECIrlAgent	手动	本地系统
OracleXETNSListener	已启动	本地系统

图 3. oracle 相关服务

## (二) 向导方式备份

采用向导方式使用 exp 和 imp

```
C:\Users\Administrator>exp
Export: Release 11.2.0.2.0 - Production on 星期三 6月 10 10:22:45 2020
Copyright (c) 1982, 2009, Oracle and/or its affiliates. All rights reserved.

用户名: scott
口令:

连接到: Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production
输入数组提取缓冲区大小: 4096 >

导出文件: EXPDAT.DMP > c:\wy.dmp
<2>U<用户>, 或 <3>T<表>: <2>U >
导出权限 <yes/no>: yes >
导出表数据 <yes/no>: yes >
压缩区 <yes/no>: yes >
· 正在导出引用完整性约束条件
· 正在导出触发器
· 正在导出索引类型
· 正在导出位图, 功能性索引和可扩展索引
· 正在导出后期表活动
· 正在导出实体化视图
· 正在导出快照日志
· 正在导出作业队列
· 正在导出刷新组和子组
· 正在导出维
· 正在导出 post-schema 过程对象和操作
· 正在导出统计信息
成功终止导出, 没有出现警告。
```

(1) 请大家读懂上述导出过程描述，使用 imp 进行导入 c:\wy.dmp。

(2) 使用 exp 导出 scott 模式下的 emp 和 dept 表。

2. 命令行方式：带有命令参数，联系按照三种方式进行数据备份和恢复。

(1) 用户模式： 导出用户所有对象以及对象中的数据；

`exp scott/orcl file=d:\wy.dmp owner=(scott);`

(2) 表模式： 导出用户所有表或者指定的表；

`exp scott/orcl file=d:\wytables.dmp tables=(dept);`

(3) 整个数据库： 导出数据库中所有对象。

`exp cott/orcl file=d:\wydatabase.dmp full=y;`

(4) `exp scott/orcl tables=(emp, dept) file=d:\mydata.dmp grants=y;`

### (三) 联机备份

请按照以下操作过程操作备份和恢复。

**SQL> conn sys/orcl as sysdba;**

**SQL> archive log list**

数据库日志模式	非存档模式
自动存档	禁用
存档终点	USE_DB_RECOVERY_FILE_DEST
最早的联机日志序列	119
当前日志序列	121

**SQL> shutdown immediate;**

数据库已经关闭。

已经卸载数据库。

ORACLE 例程已经关闭。

**SQL> startup mount;**

ORACLE 例程已经启动。

Total System Global Area	1071333376 bytes
Fixed Size	1375792 bytes
Variable Size	645923280 bytes
Database Buffers	419430400 bytes
Redo Buffers	4603904 bytes

数据库装载完毕。

**SQL> alter database archivelog;**

数据库已更改。

**SQL> alter database open;**

数据库已更改。

**SQL> alter tablespace users begin backup; //备份开始**

表空间已更改。

**SQL> alter tablespace users end backup; //备份结束**

表空间已更改。

恢复过程如下：

**SQL> alter system archive log current;**

系统已更改。

**SQL> alter system switch logfile;**

系统已更改。

**SQL> shutdown immediate;**

数据库已经关闭。

已经卸载数据库。

ORACLE 例程已经关闭。

SQL> **startup;**

ORACLE 例程已经启动。

```
Total System Global Area 1071333376 bytes
Fixed Size           1375792 bytes
Variable Size        645923280 bytes
Database Buffers    419430400 bytes
Redo Buffers         4603904 bytes
```

数据库装载完毕。

数据库已经打开。

SQL> **shutdown immediate;**

数据库已经关闭。

已经卸载数据库。

ORACLE 例程已经关闭。

SQL> **startup**

ORACLE 例程已经启动。

```
Total System Global Area 1071333376 bytes
Fixed Size           1375792 bytes
Variable Size        645923280 bytes
Database Buffers    419430400 bytes
Redo Buffers         4603904 bytes
```

数据库装载完毕。

ORA-01157: 无法标识/锁定数据文件 4 - 请参阅 DBWR 跟踪文件

ORA-01110: 数据文件 4: 'D:\APP\ADMINISTRATOR\ORADATA\ORCL\USERS01.DBF'

SQL> **set linesize 200;**

SQL> **select \* from v\$recover\_file;**

FILE#	ONLINE	ONLINE_	ERROR	CHANGE#	TIME
4	ONLINE	ONLINE	FILE NOT FOUND	0	

SQL> **alter database datafile 4 offline drop;**

数据库已更改。

SQL> **alter database open;**

数据库已更改。

将数据文件拷贝回原路径存放

SQL> **recover datafile 4;**

ORA-00279: 更改 3782020 (在 06/13/2020 11:31:22 生成) 对于线程 1 是必需的

ORA-00289: 建议:

D:\APP\ADMINISTRATOR\FLASH\_RECOVERY\_AREA\ORCL\ARCHIVELOG\2020\_06\_13\  
O1\_MF\_1\_121\_COWCX0WH\_.ARC

ORA-00280: 更改 3782020 (用于线程 1) 在序列 #121 中

指定日志: {<RET>=suggested | filename | AUTO | CANCEL}

**Auto //用户输入**

ORA-00279: 更改 3787896 (在 06/13/2020 11:59:27 生成) 对于线程 1 是必需的

ORA-00289: 建议:

D:\APP\ADMINISTRATOR\FLASH\_RECOVERY\_AREA\ORCL\ARCHIVELOG\2020\_06\_13\  
O1\_MF\_1\_122\_COWGOG0Z\_.ARC

ORA-00280: 更改 3787896 (用于线程 1) 在序列 #122 中

已应用的日志。

完成介质恢复。

SQL> **alter database datafile 4 online;**

数据库已更改。

恢复完成

#### (四) 数据泵技术

Oracle 中创建**目录对象**时，需要使用 CREATE DIRECTORY 语句。并且要对使用目录的账户进行授权 (grant)。Directory 对象包含 Read, Write 两个权限。Create 之前应提前建立物理文件目录。备份的数据将来就存储在这个目录里。

例子：导出 bookstore/admin 用户下的某个表

1. 连接 Oracle 数据库。

SQL> conn sys/orcl as sysdba

Connected.

2. 创建一个操作目录，同时需要使用操作系统命令在硬盘上创建这个物理目录。

SQL> CREATE DIRECTORY MY\_DIR AS 'd:\mydata';

Directory created.

3. 授予用户权限。

SQL> GRANT READ,WRITE ON DIRECTORY MY\_DIR TO bookstore;

Grant succeeded.

4. 导出数据表

C:>expdp bookstore/admin directory=MY\_DIR DUMPFILE=wy.dmp tables=student

同样方法，请大家自行练习导出，数据库、用户模式和表空间。

数据泵导入方法

使用 impdp 导入上例导出的数据

C:>Impdp bookstore/admin directory=MY\_DIR dumpfile=wy.dmp tables=student

**思考：**emp 与 empdp 工具的作用和性能

### (五) RMAN (选做内容)

Recovery Manager 是一个以客户端方式运行的备份与恢复工具, RMAN 可以用来备份和还原数据库文件、归档日志和控制文件。它也可以用来执行完全或不完全的数据库恢复。

RMAN 特点: (1)跳过未使用的数据块; (2)采用定制算法压缩备份; (3)执行增量备份; (4)支持块级别的恢复

使用 RMAN 进行备份和恢复/操作应具备以下条件: (1) 数据库处于归档模式。(2) 创建了恢复目录。(3) RMAN 连接到目标数据库。(4) RMAN 进行相应的参数设置。

1.Conn sys/admin as sysdba

2.select dbid,name,log\_mode,platform\_name from v\$database;

如果处于非归档模式，则需要:

执行 shutdown immediate;关闭数据库和实例。

startup mount;启动数据库，不打开实例。

alter database archivelog;切换实例为归档模式。

3. alter database open 打开实例

4.create tablespace rman\_ts datafile ‘d:\oracle\rman\_ts.dbf’ size 200M; //建恢复目录表空间

5.create user rman identified by rman default tablespace rman\_ts temporary tablespace temp;

//创建 RMAN 用户

6.grant connect,recovery\_catalog\_owner,resource to rman;//对 rman 授权

7. c:>rman catalog rman/rman; //在目录数据库中创建恢复目录

启动 RMAN

8. rman>Create catalog tablespace rman\_ts。//使用 rman\_ts 表空间创建恢复目录。

9. rman target sys/admin catalog rman/rman //连接目标数据库

10.rman>register database;//store control file to dest //注册数据库

11.Run {

allocate channel dev1 type disk;

backup database;

release channel dev1;

}