一、ORACLE 的启动和关闭
1、在单机环境下
要想启动或关闭 ORACLE 系统必须首先切换到 ORACLE 用户，如下
su - oracle

a、启动 ORACLE 系统
oracle>svrmgrl
SVRMGR>connect internal
SVRMGR>startup
SVRMGR>quit

b、关闭 ORACLE 系统
oracle>svrmgrl
SVRMGR>connect internal
SVRMGR>shutdown
SVRMGR>quit

启动 oracle9i 数据库命令：
$ sqlplus /nolog

SQL*Plus: Release 9.2.0.1.0 - Production on Fri Oct 31 13:53:53 2003

Copyright (c) 1982, 2002, Oracle Corporation. All rights reserved.

SQL> connect / as sysdba
Connected to an idle instance.
SQL> startup^C

SQL> startup
ORACLE instance started.

2、在双机环境下
要想启动或关闭 ORACLE 系统必须首先切换到 root 用户，如下
su — root

a、启动 ORACLE 系统
hareg —y oracle

b、关闭 ORACLE 系统
hareg —n oracle

Oracle 数据库有哪几种启动方式

说明：

有以下几种启动方式：
1、startup nomount
非安装启动，这种方式启动下可执行：重建控制文件、重建数据库

读取 init.ora 文件，启动 instance，即启动 SGA 和后台进程，这种启动只需要 init.ora 文件。

2、startup mount dbname
安装启动，这种方式启动下可执行：
数据库日志归档、
数据库介质恢复、

使数据文件联机或脱机，
重新定位数据文件、重做日志文件。

执行“nomount”，然后打开控制文件，确认数据文件和联机日志文件的位置，
但此时不对数据文件和日志文件进行校验检查。

3、startup open dbname
先执行“nomount”，然后执行“mount”，再打开包括 Redo log 文件在内的所有数据库文件，
这种方式下可访问数据库中的数据。

4、startup，等于以下三个命令
startup nomount
alter database mount
alter database open

5、startup restrict
约束方式启动
这种方式能够启动数据库，但只允许具有一定特权的用户访问
非特权用户访问时，会出现以下提示：
ERROR:
ORA-01035: ORACLE 只允许具有 RESTRICTED SESSION 权限的用户使用

6、startup force
强制启动方式
当不能关闭数据库时，可以用 startup force 来完成数据库的关闭
先关闭数据库，再执行正常启动数据库命令

7、startup pfile=参数文件名
带初始化参数文件的启动方式
先读取参数文件，再按参数文件中的设置启动数据库
例：startup pfile=E:Oracleadminoradbpfileinit.ora

8、startup EXCLUSIVE
二、用户如何有效地利用数据字典
ORACLE 的数据字典是数据库的重要组成部分之一，它随着数据库的产生而产生，随着数据库的变化而变化，
体现为 sys 用户下的一些表和视图。数据字典名称是大写的英文字符。

数据字典里存有用户信息、用户的权限信息、所有数据对象信息、表的约束条件、统计分析数据库的视图等。
我们不能手工修改数据字典里的信息。

很多时候，一般的 ORACLE 用户不知道如何有效地利用它。

dictionary 全部数据字典表的名称和解释，它有一个同义词 dict
dict_column 全部数据字典表里字段名称和解释

如果我们想查询跟索引有关的数据字典时，可以用下面这条 SQL 语句:

SQL>select * from dictionary where instr(comments,'index')>0;

如果我们想知道 user_indexes 表各字段名称的详细含义，可以用下面这条 SQL 语句:

```
SQL>select column_name,comments from dict_columns where table_name='USER_INDEXES';
```

依此类推，就可以轻松知道数据字典的详细名称和解释，不用查看 ORACLE 的其它文档资料了。

下面按类别列出一些 ORACLE 用户常用数据字典的查询使用方法。

1、用户

查看当前用户的缺省表空间

```
SQL>select username,default_tablespace from user_users;
```

查看当前用户的角色

```
SQL>select * from user_role_privs;
```

查看当前用户的系统权限和表级权限

```
SQL>select * from user_sys_privs;
SQL>select * from user_tab_privs;
```

2、表

查看用户下所有的表

```
SQL>select * from user_tables;
```

查看名称包含 log 字符的表

```
SQL>select object_name,object_id from user_objects
  where instr(object_name,'LOG')>0;
```

查看某表的创建时间

```
SQL>select object_name,created from user_objects where object_name=upper('&table_name');
```

查看某表的大小

```
SQL>select sum(bytes)/(1024*1024) as "size(M)" from user_segments
  where segment_name=upper('&table_name');
```

查看放在 ORACLE 的内存区里的表

```
SQL>select table_name,cache from user_tables where instr(cache,'Y')>0;
```

3、索引

查看索引个数和类别

```
SQL>select index_name,index_type,table_name from user_indexes order by table_name;
```

查看索引被索引的字段

```
SQL>select * from user_ind_columns where index_name=upper('&index_name');
```

查看索引的大小

```
SQL>select sum(bytes)/(1024*1024) as "size(M)" from user_segments
  where segment_name=upper('&index_name');
```

4、序列号

查看序列号，last_number 是当前值

```
SQL>select * from user_sequences;
```

5、视图

查看视图的名称

```
SQL>select view_name from user_views;
```

查看创建视图的 select 语句

```
SQL>set view_name,text_length from user_views;
SQL>set long 2000;          说明：可以根据视图的 text_length 值设定 set long 的大小
SQL>select text from user_views where view_name=upper('&view_name');
```

6、同义词

查看同义词的名称

```
SQL>select * from user_synonyms;
```

7、约束条件

查看某表的约束条件

```
SQL>select constraint_name, constraint_type,search_condition, r_constraint_name
  from user_constraints where table_name = upper('&table_name');

SQL>select c.constraint_name,c.constraint_type,cc.column_name
  from user_constraints c,user_cons_columns cc
  where c.owner = upper('&table_owner') and c.table_name = upper('&table_name')
  and c.owner = cc.owner and c.constraint_name = cc.constraint_name
  order by cc.position;
```

8、存储函数和过程

查看函数和过程的状态

```
SQL>select object_name,status from user_objects where object_type='FUNCTION';
SQL>select object_name,status from user_objects where object_type='PROCEDURE';
```

查看函数和过程的源代码

```
SQL>select text from all_source where owner=user and name=upper('&plsql_name');
```

三、查看数据库的 SQL

1、查看表空间的名称及大小

```
select t.tablespace_name, round(sum(bytes/(1024*1024)),0) ts_size
from dba_tablespaces t, dba_data_files d
where t.tablespace_name = d.tablespace_name
group by t.tablespace_name;
```

2、查看表空间物理文件的名称及大小

```
select tablespace_name, file_id, file_name,
round(bytes/(1024*1024),0) total_space
from dba_data_files
order by tablespace_name;
```

3、查看回滚段名称及大小

```
select segment_name, tablespace_name, r.status,
(initial_extent/1024) InitialExtent,(next_extent/1024) NextExtent,
max_extents, v.curext CurExtent
From dba_rollback_segs r, v$rollstat v
Where r.segment_id = v.usn(+)
order by segment_name ;
```

4、查看控制文件

```
select name from v$controlfile;
```

5、查看日志文件

```
select member from v$logfile;
```

6、查看表空间的使用情况

```
select sum(bytes)/(1024*1024) as free_space,tablespace_name
from dba_free_space
group by tablespace_name;

SELECT A.TABLESPACE_NAME,A.BYTES TOTAL,B.BYTES USED, C.BYTES FREE,
(B.BYTES*100)/A.BYTES "% USED",(C.BYTES*100)/A.BYTES "% FREE"
FROM SYS.SM$TS_AVAIL A,SYS.SM$TS_USED B,SYS.SM$TS_FREE C
WHERE A.TABLESPACE_NAME=B.TABLESPACE_NAME AND A.TABLESPACE_NAME=C.TABLESPACE_NAME;
```

7、查看数据库库对象

```
select owner, object_type, status, count(*) count# from all_objects group by owner, object_type,
status;
```

8、查看数据库的版本

```
Select version FROM Product_component_version
Where SUBSTR(PRODUCT,1,6)='Oracle';
```

9、查看数据库的创建日期和归档方式

```
Select Created, Log_Mode, Log_Mode From V$Database;
```

四、ORACLE 用户连接的管理

用系统管理员，查看当前数据库有几个用户连接：

```
SQL> select username,sid,serial# from v$session;
```

如果要停某个连接用

```
SQL> alter system kill session 'sid,serial#';
```

如果这命令不行,找它 UNIX 的进程数

```
SQL> select pro.spid from v$session ses,v$process pro where ses.sid=21 and ses.paddr=pro.addr;
```

说明：21 是某个连接的 sid 数

然后用 kill 命令杀此进程号。

五、SQL*PLUS 使用
a、近入 SQL*Plus
$sqlplus 用户名/密码

退出 SQL*Plus

```
SQL>exit
```

b、在 sqlplus 下得到帮助信息
列出全部 SQL 命令和 SQL*Plus 命令
SQL>help
列出某个特定的命令的信息
SQL>help 命令名

c、显示表结构命令 DESCRIBE
SQL>DESC 表名

d、SQL*Plus 中的编辑命令
显示 SQL 缓冲区命令
SQL>L

修改 SQL 命令
首先要将待改正行变为当前行
SQL>n
用 CHANGE 命令修改内容
SQL>c/旧/新
重新确认是否已正确
SQL>L

使用 INPUT 命令可以在 SQL 缓冲区中增加一行或多行
SQL>i
SQL>输入内容

e、调用外部系统编辑器
SQL>edit 文件名
可以使用 DEFINE 命令设置系统变量 EDITOR 来改变文本编辑器的类型，在 login.sql 文件中定义如下一行
DEFINE_EDITOR=vi

f、运行命令文件
SQL>START test
SQL>@test

常用 SQL*Plus 语句
a、表的创建、修改、删除
创建表的命令格式如下：
create table 表名 （列说明列表）；

为基表增加新列命令如下：
ALTER TABLE 表名 ADD （列说明列表）
例：为 test 表增加一列 Age，用来存放年龄
```
  sql>alter table test
    add （Age number(3)）；
```

修改基表列定义命令如下：
ALTER TABLE 表名
MODIFY （列名 数据类型）
例：将 test 表中的 Count 列宽度加长为 10 个字符
```
  sql>alter atble test
    modify （County char(10)）；
```

b、将一张表删除语句的格式如下：
DORP TABLE 表名；
例：表删除将同时删除表的数据和表的定义
sql>drop table test

c、表空间的创建、删除

六、ORACLE 逻辑备份的 SH 文件

完全备份的 SH 文件：exp_comp.sh

```
rq=` date +"%m%d" `

su - oracle -c "exp system/manager full=y inctype=complete file=/oracle/export/db_comp$rq.dmp"
```

累计备份的 SH 文件：exp_cumu.sh

```
rq=` date +"%m%d" `

su - oracle -c "exp system/manager full=y inctype=cumulative file=/oracle/export/db_cumu$rq.dmp"
```

增量备份的 SH 文件: exp_incr.sh

```
rq=` date +"%m%d" `

su - oracle -c "exp system/manager full=y inctype=incremental file=/oracle/export/db_incr$rq.dmp"
```

root 用户 crontab 文件
/var/spool/cron/crontabs/root 增加以下内容

```
0 2 1 * * /oracle/exp_comp.sh

30 2 * * 0-5 /oracle/exp_incr.sh

45 2 * * 6 /oracle/exp_cumu.sh
```

当然这个时间表可以根据不同的需求来改变的，这只是一个例子。

七、ORACLE 常用的 SQL 语法和数据对象

一.数据控制语句 (DML) 部分

1.INSERT (往数据表里插入记录的语句)

```
INSERT INTO 表名(字段名 1, 字段名 2, ……) VALUES ( 值 1, 值 2, ……);
INSERT INTO 表名(字段名 1, 字段名 2, ……) SELECT (字段名 1, 字段名 2, ……) FROM 另外的表名;
```

字符串类型的字段值必须用单引号括起来, 例如: ’ GOOD DAY’
如果字段值里包含单引号’ 需要进行字符串转换, 我们把它替换成两个单引号''.
字符串类型的字段值超过定义的长度会出错, 最好在插入前进行长度校验.

日期字段的字段值可以用当前数据库的系统时间 SYSDATE, 精确到秒
或者用字符串转换成日期型函数 TO_DATE(‘2001-08-01’,’ YYYY-MM-DD’)
TO_DATE()还有很多种日期格式, 可以参看 ORACLE DOC.
年-月-日 小时:分钟:秒 的格式 YYYY-MM-DD HH24:MI:SS

INSERT 时最大可操作的字符串长度小于等于 4000 个单字节, 如果要插入更长的字符串, 请考虑字段用 CLOB 类型,
方法借用 ORACLE 里自带的 DBMS_LOB 程序包.

INSERT 时如果要用到从 1 开始自动增长的序列号, 应该先建立一个序列号

CREATE SEQUENCE 序列号的名称（最好是表名+序列号标记） INCREMENT BY 1 START WITH 1
MAXVALUE 99999 CYCLE NOCACHE;
其中最大的值按字段的长度来定，如果定义的自动增长的序列号 NUMBER(6)，最大值为 999999
INSERT 语句插入这个字段值为：序列号的名称.NEXTVAL

2.DELETE （删除数据表里记录的语句）

DELETE FROM 表名 WHERE 条件;

注意：删除记录并不能释放 ORACLE 里被占用的数据块表空间. 它只把那些被删除的数据块标成 unused.

如果确实要删除一个大表里的全部记录，可以用 TRUNCATE 命令，它可以释放占用的数据块表空间
TRUNCATE TABLE 表名;
此操作不可回退.

3.UPDATE （修改数据表里记录的语句）

UPDATE 表名 SET 字段名 1=值 1，字段名 2=值 2，…… WHERE 条件;

如果修改的值 N 没有赋值或定义时，将把原来的记录内容清为 NULL，最好在修改前进行非空校验;
值 N 超过定义的长度会出错，最好在插入前进行长度校验..

注意事项:
A. 以上 SQL 语句对表都加上了行级锁,
确认完成后，必须加上事物处理结束的命令 COMMIT 才能正式生效,
否则改变不一定写入数据库里.
如果想撤回这些操作，可以用命令 ROLLBACK 复原.

B. 在运行 INSERT，DELETE 和 UPDATE 语句前最好估算一下可能操作的记录范围,
应该把它限定在较小（一万条记录）范围内,. 否则 ORACLE 处理这个事物用到很大的回退段.
程序响应慢甚至失去响应. 如果记录数上十万以上这些操作，可以把这些 SQL 语句分段分次完成,
其间加上 COMMIT 确认事物处理.
二.数据定义（DDL）部分

1.CREATE（创建表，索引，视图，同义词，过程，函数，数据库链接等）

ORACLE 常用的字段类型有
CHAR 固定长度的字符串
VARCHAR2 可变长度的字符串
NUMBER(M,N) 数字型 M 是位数总长度，N 是小数的长度
DATE 日期类型

创建表时要把较小的不为空的字段放在前面，可能为空的字段放在后面

创建表时可以用中文的字段名，但最好还是用英文的字段名

创建表时可以给字段加上默认值，例如 DEFAULT SYSDATE
这样每次插入和修改时，不用程序操作这个字段都能得到动作的时间

创建表时可以给字段加上约束条件
例如 不允许重复 UNIQUE，关键字 PRIMARY KEY

2.ALTER （改变表，索引，视图等）

改变表的名称
ALTER TABLE 表名 1 TO 表名 2;

在表的后面增加一个字段

ALTER TABLE 表名 ADD 字段名 字段名描述;

修改表里字段的定义描述
ALTER TABLE 表名 MODIFY 字段名 字段名描述;

给表里的字段加上约束条件
ALTER TABLE 表名 ADD CONSTRAINT 约束名 PRIMARY KEY (字段名);
ALTER TABLE 表名 ADD CONSTRAINT 约束名 UNIQUE (字段名);

把表放在或取出数据库的内存区
ALTER TABLE 表名 CACHE;
ALTER TABLE 表名 NOCACHE;

3.DROP　　(删除表, 索引, 视图, 同义词, 过程, 函数, 数据库链接等)

删除表和它所有的约束条件
DROP TABLE 表名 CASCADE CONSTRAINTS;

4.TRUNCATE (清空表里的所有记录, 保留表的结构)

TRUNCATE 表名;

三.查询语句 (SELECT) 部分

SELECT 字段名 1, 字段名 2, …… FROM 表名 1, [表名 2, ……] WHERE 条件;

字段名可以带入函数
　例如: COUNT(*), MIN(字段名), MAX(字段名), AVG(字段名), DISTINCT(字段名),
　　TO_CHAR(DATE 字段名,'YYYY-MM-DD HH24:MI:SS')

NVL(EXPR1, EXPR2)函数
解释:
IF EXPR1=NULL
　　RETURN EXPR2
ELSE
　　RETURN EXPR1

DECODE(AA，V1，R1，V2，R2....)函数
解释:
IF AA=V1 THEN RETURN R1
IF AA=V2 THEN RETURN R2
..…
ELSE
RETURN NULL

LPAD(char1,n,char2)函数
解释:
字符 char1 按制定的位数 n 显示，不足的位数用 char2 字符串替换左边的空位

字段名之间可以进行算术运算
例如: (字段名 1*字段名 1)/3

查询语句可以嵌套
例如: SELECT …… FROM
(SELECT …… FROM 表名 1, [表名 2, ……] WHERE 条件) WHERE 条件 2;

两个查询语句的结果可以做集合操作
例如: 并集 UNION(去掉重复记录), 并集 UNION ALL(不去掉重复记录), 差集 MINUS, 交集 INTERSECT

分组查询
SELECT 字段名 1, 字段名 2, …… FROM 表名 1, [表名 2, ……] GROUP BY 字段名 1
[HAVING 条件] ;

两个以上表之间的连接查询

SELECT 字段名 1, 字段名 2, …… FROM 表名 1, [表名 2, ……] WHERE
表名 1.字段名 = 表名 2. 字段名 [AND ……] ;

SELECT 字段名 1, 字段名 2, …… FROM 表名 1, [表名 2, ……] WHERE
表名 1.字段名 = 表名 2. 字段名(+) [AND ……] ;

有(+)号的字段位置自动补空值

查询结果集的排序操作, 默认的排序是升序 ASC, 降序是 DESC

SELECT 字段名 1, 字段名 2, …… FROM 表名 1, [表名 2, ……]
ORDER BY 字段名 1, 字段名 2 DESC;

字符串模糊比较的方法

INSTR(字段名, '字符串')>0
字段名 LIKE '字符串%' ['%字符串%']

每个表都有一个隐含的字段 ROWID, 它标记着记录的唯一性.

四.ORACLE 里常用的数据对象 (SCHEMA)

1.索引 (INDEX)

CREATE INDEX 索引名 ON 表名 (字段 1, [字段 2, ……]);
ALTER INDEX 索引名 REBUILD;

一个表的索引最好不要超过三个 (特殊的大表除外), 最好用单字段索引, 结合 SQL 语句的分析执行情况, 也可以建立多字段的组合索引和基于函数的索引

ORACLE8.1.7 字符串可以索引的最大长度为 1578 单字节
ORACLE8.0.6 字符串可以索引的最大长度为 758 单字节

2.视图 (VIEW)

CREATE VIEW 视图名 AS SELECT …. FROM ….. ;
ALTER VIEW 视图名 COMPILE;

视图仅是一个 SQL 查询语句, 它可以把表之间复杂的关系简洁化.

3.同义词 (SYNONMY)
CREATE SYNONYM 同义词名 FOR 表名;
CREATE SYNONYM 同义词名 FOR 表名@数据库链接名;

4.数据库链接 (DATABASE LINK)
CREATE DATABASE LINK 数据库链接名 CONNECT TO 用户名 IDENTIFIED BY 密码 USING '数据库连接字符串' ;

数据库连接字符串可以用 NET8 EASY CONFIG 或者直接修改 TNSNAMES.ORA 里定义.

数据库参数 global_name=true 时要求数据库链接名称跟远端数据库名称一样

数据库全局名称可以用以下命令查出
SELECT * FROM GLOBAL_NAME;

查询远端数据库里的表
SELECT …… FROM 表名@数据库链接名;

五.权限管理 (DCL) 语句

1.GRANT 赋于权限
常用的系统权限集合有以下三个:
CONNECT(基本的连接), RESOURCE(程序开发), DBA(数据库管理)
常用的数据对象权限有以下五个:
ALL ON 数据对象名, SELECT ON 数据对象名, UPDATE ON 数据对象名,
DELETE ON 数据对象名, INSERT ON 数据对象名, ALTER ON 数据对象名

GRANT CONNECT, RESOURCE TO 用户名;
GRANT SELECT ON 表名 TO 用户名;
GRANT SELECT, INSERT, DELETE ON 表名 TO 用户名 1, 用户名 2;

2.REVOKE 回收权限

REVOKE CONNECT, RESOURCE FROM 用户名;
REVOKE SELECT ON 表名 FROM 用户名;
REVOKE SELECT, INSERT, DELETE ON 表名 FROM 用户名 1, 用户名 2;

查询数据库中第 63 号错误:
select orgaddr,destaddr from sm_histable0116 where error_code='63';

查询数据库中开户用户最大提交和最大下发数: select MSISDN,TCOS,OCOS from ms_usertable;

查询数据库中各种错误代码的总和:
select error_code,count(*) from sm_histable0513 group by error_code order
by error_code;

查询报表数据库中话单统计种类查询。
select sum(Successcount) from tbl_MiddleMt0411 where ServiceType2=111
select sum(successcount),servicetype from tbl_middlemt0411 group by servicetype