Oracle数据库

笔记本： 数据库
创建时间： 2015/12/15 21:37
标签： 数据库

Oracle数据库
2015年12月15日
21:37

一，SQL*plus使用

1. **help index** 显示所有SQL *plus的命令。**Help+命令** 显示该命令的帮助信息。
2.

N 删除行n
N m 删除行n到行m
N* 删除行n到当前行
N last 行n到最后
* 当前行
* n 当前行到行n
* last 当前行到最后
Last 最后一行
Bre 清除分组定义
Buff 清除缓冲区
Col 清除列的显示属性
Comp 清除计算设置
Scr 清屏
SQL 清除SQL缓冲区
Timi 清除定时
a+text 在当前行末尾添加文本。 **c /旧字符串/新字符串** 替换修改。**Cl+选项** 清除某项设置。
Del+选项 删行的内容。
Input +text 当前行添加一行。
List+选项 显示行的内容
Pro+text 显示文本，但不作
为行存储。**Desc+object** 列出表、视图、存储过程等的定义信息。**Def+变量/变量=text** 定义一个变量/显示变量。**Acc+变量+类型+prompt+‘提示信息’** 提示用户输入变量值。**Set+环境变量+value** 设置SQL *plus的环境。**Run** 运行SQL缓冲区中SQL语句。

3. SQL *plus使用技巧

Set timing on/off 跟踪语句执行时间。**Ctrl+C** 终止正在运行的结果。**Column 列名 format A字符长度（heading ‘’ 修改列标题）** 格式化运行结果。**Spool +uri路径** 把查询结果输出到文件，**spool off** 停止输出。**#+**命令 在输入SQL命令的过程中临时先运行一个SQL *plus命令。
Set autotrace on explain： 显示执行SQL的细节。

4. 脚本文件

Sav 文件路径+文件名 【create】【replace】【app】 将SQL缓冲区内容保存到指定文件（.sql文件）中。**Get文件路径+文件名** 从指定文件加载SQL指令。
Spo +文件路径；spool 将SQL*plus的各种操作及运行结果复制到文件中（.ist）文件。即从spool指令开始讲屏幕所有结果复制到文件，直到**spool off；spool**指令为止。
Start 文件路径 【arg1，arg2....】 运行脚本【】内为希望传递给脚本文件的参数。Arg1代替所有的&1，arg2代替所有的&2(这叫做替换变量)。

5. **conn 用户名/密码【@<连接标识符>】 as sysoper/sysdba/sysasm** 指定用户建立新的连接。

Disc 断开当前用户的连接，但不退出SQL*plus。

6. 替换变量（&）

使用&&可以避免多次提示输入变量值，要改变值要先用undefine清除该变量。

定义替换变量：**def 变量名【=值】**。使用该变量时，格式为&变量名。Define 变量名 查询替换变量情况，**define** 查看所有变量（都是char类型）。

Acc**变量名【num ,char,date,binary_float,binary_double,format ,def】【prompt text|nopr】【hide】** acc可以指定变量的数据类型，prompt为指示信息，hide隐藏输入。

清除替换变量：**undef 变量名**。

prompt【text】 将提示信息或空行输出到屏幕。

Pause【text】暂停脚本运行，将text输出到屏幕。

7，绑定变量（：）相当于常量

Var 变量名

【number|char|char()|nchar|nchar()|varchar2()|nvarchar2()|clob|nclob|refcursor|binary_float|binary_double】

使用绑定变量时必须在绑定变量前加：（并且字符串不用‘’）。

赋值时：**exec ： 变量 ： =值**。显示定义的绑定变量：**print v1 v2 v3** 。

Show 系统变量 显示系统变量的值

Show all 显示所有系统变量值

Show errors 显示错误信息

Show paramet [参数名] 显示初始化参数

Show release 显示数据库版本

Show SGA 显示SGA的大小（DBA权限）

Show sqlcode 显示操作之后的状态代码

Show user 显示当前用户

8，sqlplus环境变量

Show 显示当前sqlplus环境中的系统变量、错误信息、初始化信息等。

Set 系统变量 值 设置系统变量的值。Help Set 查看set可以更改的所有变量。

Store set 文件路径 【cre|rep|app】 将当前环境设置保存到脚本文件（.sql），可以用start重现当前设置，达到保存设置的作用。

9，常用系统变量

Arraysize 从数据库中一次提取的行数，默认为15。值越大，网络开销越小，内存增加。

Autocommit 设置自动提交，默认off 。

Closep 列之间的分隔符，默认为空格。Headsep 表示在哪里将标题分行显示，默认为|。

Echo 用start执行脚本时，控制是否显示脚本中正字啊之星的sql语句，默认为off。

Feedback 在结果集后面显示返回的行数，默认off。

Heading 是否显示查询结果的列标题，默认为on。

Linesize 每行的宽度，默认为80。Pagesize 每页显示的行数，默认为14。

Newpage 设置分割页与页之间的空白行数，默认为1。

Sqlprompt sqlplus的命令提示符 默认为SQL>。

Time 是否在命令提示符钱显示系统时间，默认为off。

Trimspool 将spool输出中每行后面多余的空格去掉，默认为off。

Underline 设置下划线字符，默认为_。

Verify 是否列出替换变量赋值前后的比较，默认为on。
10，格式化查询结果
9 禁止显示前面的0
1. 强制显示前面的0
$ 显示美元符号
L 显示本地货币符号
. 小数点位置
， 千分隔符
S 正负符号位
Col 原列名 heading 新列名 justify left/right/center 修改列标题，justify为标题的对齐方式。
Col 数值列 format 格式标准 对数值列规定格式。格式样式用到这些元素作占位符。
Col 列1 like 列2 将列1的格式复制给列2。
Col 列名 显示当前列的格式。**Col** 显示所有列的格式。**Null text** 空值时显示的文本。
Col 列名 clear 恢复默认的格式。 **Clear columns** 恢复所有的行。**Col 列名 on/off** 指定的格式是否开启。
Bre on列名 skip n/skip page 限制重复列，skip n表示列值变化时跳过多少行，skip page 跳过一页。On可以连用同时规定多个列。除了列，还可以是row、expression、report。
Sum 和
Min 最小值
Max 最大值
Avg 平均值
STD 标准方差
Variance 协方差
Count 非空的个数
Number 行数
Compute 函数名 [lab text] of {expr|column|alias} on {expr|col|alias|report|row}
Col <n> 在第n列开始显示
Skip <n> 跳过n列
Tab <n> 插入n个制表键
Left 左对齐
Center 剧中对齐
Right 右对齐
Bold 加粗
Format text 格式化
对各个分组进行各种汇总计算，总是与break on配合使用。Lab为汇总描述信息，of为要计算的列/表达式，on为该列值变化时产生一个汇总。
Tti 位置 文本 off/on 设置页标题。Bti {省略} 设置页脚注。Reph {省略} 设置报告标题。Repf {省略} 设置报告脚注。
获取系统日期
Column today new_value time
Select to_char（sysdate，‘YYYY-MM-DD’） today from dual；
第几页
Bti right “第” format 99 SQL.PNO “页”

二，SQL基础

1.
地区编码的意思
1是指europe
2是指american
3是指asia
4是指middle east
数据库中日期格式是“DD-MON-YY”，sql语句中日期必须符合改格式，可以使用to_date()转换。例如：2008-3-20转换为默认格式为20-3月-08，即**to_date（‘2008-3-20’，‘YYYY-MM-DD’）**。单引号！！

2. 列的别名要用双引号！字符串常量用单引号。

3. Select region_id **||’是指’ ||** region_name as 地区编码的意思 from regions的查询结果为：

4. 在select和from中间加入case表达式，可以对查询的结果进行处理。例如：

```
select region_id,region_name,
 2  case region_id
 3    when 1 then '欧洲'
 4    when 2 then '美洲'
 5    when 3 then '亚洲'
 6    when 4 then '中东和非洲'
 7  end 中文地区名 from regions;
REGION_ID REGION_NAME              中文地区名
---------- ------------------------- ----------
         1 Europe                    欧洲
         2 Americas                  美洲
         3 Asia                      亚洲
         4 Middle East and Africa    中东和非洲
```

5. 如果select使用了distinct，则用于排序的列必须是所查询的列，否则会出错。分组（group by）函数只能用于列、orderby、having子句中，除count（*）外，其他函数都会忽略null值的行。Having后面只能用分组函数本身，不能用列的别名；where不能使用分组函数。
6. 尽量避免出现笛卡尔积，应在from子句中有n个表时，在where子句中至少有n-1个连接条件。
7. **SQL92的连接语法**

表1 Inner join表2 on连接条件 内连接：当连接中的列满足连接条件时才返回相应的行。若列值为null，不返回相应行（而在外连接中返回）。 多个表时inner join on连用。

Using对on子句进行简化，条件：相等连接；连接条件中列必须同名。**Using（连接条件中同名的列名1，列名2…）**。 在select中使用using内的列名时不能有表的限定，直接用列名。

8.
Union all 返回两个结果集所有行
Union 返回所有行，没有重复行
Minus 返回第一个结果集中有，第二个结果集没有的行
Intersect 返回两个结果集都有的行。
集合查询：将多个select结果集合并到一起。

格式：**select语句1 集合运算符 select语句2 order by 列名**。

多行运算符：
In， not in,
exists, not exists,

all, any
两个结果集的列个数及数据类型要相同，order by只能使用第一个结果集的列，只能有一个order by。
9，多列子查询：使用（列1，列2..）= （select列1，列2，.....）。例如：select first_name,last_name,salary,job_id from employees
Create table/view
表名 （列...） as select语句
where (**salary,job_id**) = (select s**alary,job_id** from employees
where employee_id=198)
Insert into 表名（列....）
Select 语句
在DDL,DML中也可以使用子查询：
Create table中：复制已经存在的表结构并自动插入数据。
Update 表
Set （列.....）=
Select 语句
Create view 中：创建的视图必须依据表，所以要用select。
Insert 中：将一个表的数据插入到另一个表。
Delete 中：在delete的where语句中可以使用子查询。
Update 中：在update中可以讲新的列值设置成子查询结果。
事务性质：
原子性（A）
一致性（C）
隔离性（I）
持久性（D）
Commit 提交事务
Rollback 回退事务
Savepoint sp设置保存点
Rollback to [savepoint] sp 会退到保存点
Set transaction设置事务属性
Set constrants 设置可延迟约束的检验时机
10，**事务控制**：是用户定义的一组操作序列，由多条（1条）SQL语句组成，是数据操作的逻辑单位。**事务控制只限于DML**。
自动提交事务：执行一条DDL语句（create table，alter Table，drop table，altersystem）；执行一条DCL语句（Grant，revoke，audit，noaudit，rename）；断开数据库连接。
提交：提交的工作简单固定，花费的时间相同。提交的重要性：会话A插入记录，但在提交之前，另一个会话B是查不到该记录的；A提交之后，B才能查得到。
回退：回退全部事务到上一个commit成功后的状态，回退的时间取决于要撤销的数据量。
保存点：一个事务的某些中间标识，有了保存点可以将事务会退到指定保存点，而不必回退所有操作。
隔离级别：
Read committed
read uncommitted
Repeatable read
Serializable
设置事务属性：对即将开始的事务性质的控制，如是否可读写，是否只读，隔离级别等，**设置事务必须是事务中的第一个语句**。**Set transaction read write** 设置可读写，该事务中可以执行DML；**Set transaction read only** 设置只读，不能执行DML（在A设置只读，在B中虽然可以更新数据提交，但在A中不会查询到更新后的数据，说明只读事务可以取得特定时间点的数据）。**Set transaction isolation level +级别** 设置“读已提交事务”的隔离级别。
可延迟约束检验：为一个事务涉及的可延迟约束指定检验时机。
set Constraints [all|约束1.....] [deferred|immediate]，deferred为提交时检

验，immediate为每条DML之后立即检验（和不可延迟约束一样）。

各隔离等级对读现象的处理能力：

隔离等级	脏读	不可重复读	幻像读
Read uncommitted	可能	可能	可能
Read committed	不会	可能	可能
Repeatable read	不会	不会	可能
Serializable	不会	不会	不会

11，SQL中的各种函数（仅常用）

单行数字函数：abs()绝对值；acos()反余弦;ceil()最小整数；cos()余弦;cosh()双余弦;exp()幂；floor()最大整数；ln()自然对数；log(,)前为底的对数；mod(,)余数；Power(y,x) y为底的x次幂;round()四舍五入；sign() 正负检验；sinh()双正弦；tanh()双正切；trunc(x,[y]) 截取数字。**Greatest（e1,e2,e3…）**找出其中的最大值；**least（e1,e2,e3…）**找出其中的最小值。**Nvl（e1,e2）** 如果e1不是null，返回e1，否则返回e2；**nvl2（e1,e2,e3）**e1不是null，返回e2，否则返回e3（e2，e3不能是long类型）。

单行字符函数：**ascii(c)** 返回ASCII值；**chr(n)** 返回ASCII对应字符；**concat(c1,c2)** 连接c1c2;**initcap(c)** 返回首字母大写；**instr(c1,c2 [n,m])** 在c1中从n开始搜索c2第m次出现；**length(c)**返回长度;lower()、upper()大小写；**Lpad(c1,n[c2])** 在c1左边补充c2，直到达到n长度。**Ltrim(c1[,c2])** 去掉c1左边包含c2的字符(**rpad,rtrim**是在右边)；**replace(c1,c2[,c3])**将c1中的c2替换成c3（**translate**相同）；**substr(c1,m[,n])**返回c1的子串，从m开始，长度n；**trim([c1] c2 from c3)**
从c3的c1处(leading,trailing,both)删除c2。**Decode(x,if1,then1,if2,then2…else)**
将x与if1,if2…比较，相等返回thenn，否则返回else。

Year，month,day
Min,second,
Timazone_hour
Timezone_minute
Timezone_region
Timezone_abbr

单行日期时间函数：**alter session set NLS_DATE_FORMAT=格式** 设置当前会话的日期格式，**NLS_DATE_FORMAT**为空时采用默认的日期格式。**To_char(x[,fmt])** 将x按fmt格式转换为字符串，x为日期/数字。**To_timestamp(c)** 将c转换为timestamp数据(时间类型和date差不多)。**Numtodsinterval(n,c)** 将数字n转化为c(day,hour,minute,second)类型数据；**Numtoyminterval(n,c)** 将n转化为c(year,month)类型数据，**To_dsinterval(c)** 将字符串c转换为interval day to second类型(时间间隔存储)数据；**To_yminterval(c)** 将c转换为interval year to month数据（时间间隔）。这四个主要用于原有时间上增加n。**Add_months(d,n)**在日期d加上n个月。**Current_date，current_timestamp(p)**,**localtimestamp(p)**当前会话的日期，p表示精度。**Dbtimezone** 数据库所在时区。**Extract（c from d）** 在日期d中取指定的部分。**Last_day(d)** 日期d所在月份最后一天。**Months_between(d1,d2)** 日期d1，d2之间相差的月份。**Round(d[,fmt])** 日期的四舍五入（年 7月1号，月16号，天12:00）。**Sessiontimezone** 返回上次alter session后数据库时区偏移量。**Sys_extract_utc(ts)** 返回ts(timestamp类型)对应的格林尼治时间。**Sysdate，systimestamp** 当前数据库时间。Trunc(d[,fmt]) 根据fmt指定的精度截断日期d。

常用日期格式代码及说明		
日期代码	**格式说明**	**示例**
BC、AD	公元前、后	2002 AD
AM、PM	上、下午	09 AM（上午9点）
Dy/DY	星期(首字母大写)/（大写）	Mon/MON,Tue/TUE
DAY/Day	星期全拼	MONDAY/Monday
D/DD/DDD	周天数/月天数/年天数	1~7/1~31/1~365
w/ww	本月/年第几周	1~5/1~53
MM/MON	两位数月份/月份全拼	01/Jan(一月)
MONTH/Month	月份全拼	JANUARY/January

YYYY/YYY/YY/Y	不同位数表示年份	2012/012/12/2
YEAR/Year	英文的年份	TOW THOUSAND-EIGHT
HH、HH12	12小时制小时	1~12
HH24	24小时制小时	1~24
MI	分钟	0~59
SS/SSSS	分钟秒数/天秒数	0~59/0~86399

单行转换函数：从一种数据类型到另一种数据类型。好多都在上面介绍了。**to_number(c[,fmt])** 将fmt格式的字符串c转换成数字类型。**Asciistr(c)** 将c中非字符转换为ASCII。**Bin_to_num(n1,n2,n3...)** 将n1，n2，n3等组成的二进制转换为十进制。**Cast(c AS t)** 将表达式c转换为数据类型t（内置或自定义）。**Unistr(c)** 字符串对应的unicode字符。

常用数字格式代码及说明		
数字代码	**格式说明**	**示例**
9	正—数位；负—负号	9999.9->123.4;9999.9->-123.4
0	前导或结尾位置的0	0009.90->0012.30
,/G	组分隔符（，不能出现在小数点后）	9,99->1,23;9G99->1,23
./D	小数点	
$/L	美元符号/本地货币符号	$999->$123;￥999->￥123
EEEE	科学计数法	9.0EEEE->1.2E+05
FM	没有前导或末尾空白	FM99.99->.1
MI/PR/S	末尾负号、<>负数、前导负号	999MI->123-;999PR-><123>
RN/rn、X	大写、小写罗马数字、十六进制数	Declare 声明部分 Begin 执行部分 （End；） Exception 异常处理 end； XX->FC

DBMA_OUTPUT.PUT_LINE() 输出变量的值，类似于printf（）。

三 PL/SQL基础

1. PL/SQL程序结构

组成部分：声明部分、执行部分（必需的）、异常处理部分。注意：**区分大小写！单行注释为--。**

声明变量：变量名 [constant] 类型 [not null] :=/default 变量值。Constant表示常量，not null表示不能为空，赋值号：=或只用default。可以从表中取数据赋值：**select 列 into 变量 from.....。**

2. 数据类型

Number [(p,s)] 定义固定长度的数字，p(1~38)表示总位数，s(-84~127)表示小数点后位数。

Char(n) 定义固定长度的字符串，n可选表示长度，2000字节。

Varchar2(n) 定义可变长度的字符串，n必需，4000个字节内。

Boolean 定义布尔变量，取值为true、false、null。

Date、timestamp(s) 日期类型。Timestamp是date的扩展，秒有小数位，还多了上下午和时区。

Interval [year(y) to month]、Interval [day(d) to second(s)] 年月的时间间隔，y可选0~4；天小时分钟秒之间的时间间隔，d,s可选0~9。时间间隔可以与timestamp数据进行加减运算。

Lob(blob,clob,bfile) 存储最大为4G的无结构大数据块。Blob为含图片图像的大文件；clob为只有文本的大文件；bfile为声音视频。

%type 使一个变量的类型与另一个已经定义的变量总是保持一致。例如：v_salary employees.salary%type。注意要用.指定表名。

Type 记录名 is record (
V1 type1 [not null] [:=value]
V2 type2 [not null] [:=value]
V3 type3 [not null] [:=value]
);

%rowtype 定义一行的记录的变量。例如：v_emp employees%rowtype; 使用行中数据时：v_emp.employee_id。

Record 记录数据类型（类似于结构体）使用里面的变量要使用.运算符。定义的是一种类型，要使用要再定义该类型的变量。

Varray pl/sql中的数组类型。定义语法为：

Type 数组名 is varray(size) of元素类型 [not null]；定义的也是一种

Type 表名 is table of 元素类型 [not null]
Index by [binary_integer | PLS_integer | varray2];

类型，要使用要再定义该类型的变量。

Table 记录表（索引表）类型，是对记录的扩展，类似二维数组(可以一维可以二维)。

3. 集合API方法：作用于数组、记录表的函数等。

调用格式：**集合名.方法名 [（参数.....）]**

常用集合API函数	
名称	**说明**
Count	赋值了的成员数目
Delete	删除所有成员
Delete(n)、delete(n,m)	删除第n个成员、删除n到m个成员
Exists（n）	Din个成员是否存在。
Extend、extend(n)	添加null成员**Varray类型**
Extend(n,i)	添加n个成员，值与第i个成员相同**Varray类型**
First、last	成员最低、最高下标
Limit	**Varray类型**出现的最高下标
Next(n)、prior(n)	返回下标n的成员的后一个、前一个成员的下标
Trim、trim(n)	删除末尾成员

4,流程控制

条件控制：

If 条件 then
语句段1
Else
语句段2
End if;
	If 条件 then
	Elsif 条件
	Elsif 条件
	Else
	End if;
		Case [条件]
		When 条件 then
		When 条件 then
		End case;

循环控制：

<<循环标签>> Loop 语句段 Exit 语句 End loop[循环标签];	<<循环标签>> While 条件loop 语句段 End loop [循环标签];	<<循环标签>> For变量 in [reverse] 初值..终值 loop 语句段 End loop [循环标签]; (使用reverse时初终值调换)

5，游标

For 变量（不用声明） in 游标名 loop
语句段
End loop；

可以省略open、fetch、close和循环的步骤。

声明游标 **cursor 游标名 [(参数表)] [return datatype] is select语句；**

打开游标：**open 游标名 [(参数表)];**

提取游标：**fetch 游标名 into 变量或记录变量；**

游标%isopen 游标是否打开； 游标%found 最近一次提取是否成功

游标%notfound 最近一次提取是否失败；

游标%rowcount 最近一次提取到数据行的行序号

关闭游标：**close 游标名；**

使用游标更新删除数据：在声明游标时select后面加 **for update【nowait】**，更新数据时在update/delete后加**where current of 游标名**。

游标变量：可以多次定义（不同的select语句），定义的是类型，变量要再声明。

Exception
When 异常错误名1 [or 异常错误名2]
Then 语句段1
When 异常错误名3 [or 异常错误名4]
Then 语句段2
When others then 语句段3
End；

Type 游标类型 is ref cursor [return 返回类型];(有return的为强类型。)

使用时：**open 游标类型变量 for select语句；**

6，异常处理：

都是oracle判断

预定义异常：PL/SQL预定义了一些异常，可以在安装目录下的stdspec.sql文件中查看。预定义异常虽然是被隐含引发的，但要在异常处理部分用名称处理相应的异常。

非预定义异常：该中异常要自己声明，格式为

异常名 exception；（声明异常）

Pragma exception_init(异常名，oracle错误代码)；（与某个错误联系）

自定义异常：也需要声明异常，但不需要与某个错误联系，只需要使用raise抛出异常。（类似throw）

SQLCODE 无参数，返回错误代码（用户自定义+1；没有异常0；未找到数据+100）。**SQLERRM(error_code)**根据错误代码返回对应的错误消息文本。Raise_application_error(错误代码，错误消息[,true|false]) 自定义错误代码和消息文本，错误代码必须在-20000~-20999之间，该方法不能在匿名块使用。

形参=>实参——按名称调用

7，过程和函数

过程：**创建过程**

Create [or replace] procedure 过程名 （replace覆盖同名过程）
(参数 [in|out|in out] 类型 [default 值]....)
Is |as 声明部分 （in模式：实参传值，只读不能写）
Begin 执行部分 （out模式：形参为null，形参传值给实参）
Exception 异常处理部分
End 过程名；

调用过程：在SQL*PLUS中药使用exec来调用，在PL/SQL块中直接调用。

删除过程：drop procedure 过程名；

函数：一般用户计算和返回一个值。

创建函数：

Create [or replace] function 函数名
(参数 [in|out|in out] 类型 [default 值]....)
Return type
Is |as 声明部分
Begin 执行部分
Return 返回值
Exception 异常处理部分
End 函数名；

Select line,text from user_source
Where name='对象名' and type=' 类型';
Select * from user_objects
Where object_type in ('PROCEDURE','FUNCTION');

删除函数：drop function 函数名；

管理过程和函数
查看创建的过程函数名称：
查看创建的过程函数源代码：查看编译错误：**show errors。**
重新编译过程和函数：**alter procedure|function|view|trigger|** 名称 **compile。**

8，程序包

程序包说明部分：

Create [or replace] package 包名
Is | as
[Pragma serially_reusable;]
公有数据类型定义
公有变量声明
公有常量声明
公有异常错误声明
公有游标声明
公有函数声明
公有过程声明
End 包名；

程序包体部分：

Create [or replace] package body 包名
Is | as
[Pragma serially_reusable;]
私有数据类型定义
私有变量声明
私有常量声明
私有异常错误声明
私有函数声明及定义
私有过程声明及定义
公有游标定义
公有函数定义
公有过程定义
Begin 初始化部分
End 包名；

重新编译程序包：**alter package 包名 compile [package|body|specification];**
删除包：**drop package body 包名**；（仅删除包体）**drop package 包名**；

9,触发器

触发器分为DML触发器、instead of触发器、系统触发器。（里面可以使用inserting、updating 列名、deleting条件谓词）。

DML触发器：

语句级触发器：	行级触发器：

Create [or replace] trigger 名称 Before|after Insert|update|delete or.... On 表名 PL/SQL程序块|call 过程	Create [or replace] trigger 名称 Before|after Insert|update[of列名]|delete or.... On 表名 For each row

行级触发器中在列名前加上old.或new.表示列变化前后的值。在PLSQL中前面加：，在when中不用加。

Instead of触发器：只能定义在复杂视图（不可更新的视图）中，通过触发器执行更新。DML触发器除了执行触发它的DML语句，海之行触发器内部语句；instead of触发器只执行内部语句。

Create or replace trigger 名称
Instead of Insert\|update[of列名]\|delete or....
On 视图名
For each row——instead of触发器只能是行级触发器
When 条件
PL/SQL程序块\|call 过程

Startup和shutdown触发器使用when语句。
Servererror可以用errno检测错误。
Logon和logoff可以用userid和username检测用户id和名称。

系统触发器：基于对数据库进行的操作触发，如DDL、启动关闭DB、连接断开等。

Create [or replace] trigger 名称 Before|after DDL事件|数据库事件 On DB|schema When 条件 PL/SQL程序块|call 过程	常见事件： Create、drop、alter、grant、revoke、rename（DB重命名）,audit（审计）、noaudit（停止审计）、logon（登录DB）、logoff（退出DB）、startup（启动DB实例）、shutdown（关闭DB）、servererror。

常用事件的属性函数：

函数	返回类型	说明
Ora_client_ip_address	Varchar2(20)	返回登录DB客户机的IP
Ora_database_name	Varchar2(50)	返回DB名称
Ora_dict_obj_name	Varchar2(30)	返回DDL操作的DB对象
Ora_dict_obj_owner	Varchar2(30)	返回DDL操作的对象所有者
Ora_dict_obj_type	Varchar2(20)	返回DDL操作对象类型
Ora_is_alter_column(列 in varchar2)	Boolean	对alter事件，如果正在更新某列返回true
Ora_is_drop_column(列 in varchar2)	Boolean	对drop事件，如果正在删除某列返回true
Ora_login_user	Varchar2(30)	返回登录的用户名
Ora_is_servererror(err_no in BINARY_INTEGER	Boolean	如果发生err_no错误返回true
Ora_sysevent	Varchar2(20)	触发该触发器的事件名称

查询触发器：（**user_triggers,all_triggers,dba_triggers**视图）
例如：select trigger_name,description,trigger_body from user_triggers where table_name='regons';

禁用/启用触发器：alter trigger 名称 disable|enable;——单个触发器：**alter table 表名 disable|enable all triggers;**——一个表全部触发器。

重新编译触发器：**alter trigger 名称 cpmpile**

删除触发器：**drop trigger 名称;**

四，表空间（sys创建）

1,**create user 用户名 identified by 密码** 创建DB用户及密码。
Drop user 用户名 cascade 删除用户（同时会删除该用户创建的对象）。**Alter user 用户名 identified by 密码 account unlock** 将锁定的用户解锁。
Select * from session_roles 查询当前用户所具有的角色。
2,本地管理下**区分配方式：uniform**（统一分配，相同大小）、**system**（自动分配，小表取小区，大表取大区）。
段空间管理方式：manual（用空闲列表管理段的空闲数据块）、**auto**（用位图管理已用和空闲数据块）。
表大小=最大行长x行数x（1+pctfree/100）x预留的百分比。
一个能避免竞争的表空间层次结构：

系统表空间			
临时表空间			
撤销表空间			
工具表空间			
用户表空间（示例表空间）			
应用程序1数据表空间		应用程序2索引表空间	
App2分区表1分区1表空间	App2分区表1分区2表空间		大对象数据表空间

3,创建表空间
创建表空间语法：大文件小文件　表空间类型
Create [smallfile|bigfile] [permanent|temporary|undo] tablespace
Datafile|tempfile datafile_clause1....
不能同时使用
[logging | nologging]
[extent management dictonary | local]　区管理方式
[autoallocate | uniform [size integer [K|M]]]　区分配方式
Show parameters db_block_size
[default storage_clause] 基本不用了
[segment space management auto |manual] 段空间管理方式
[blocksize integer [K|M]]　指定块的大小，不用默认
[minimum extent integer [K|M]]　区最小空间
[online | offline [normal | temporary |immediate |for recover]];
其中，datafile_clause（数据文件子句）语法：
'path_filename suffix' size integer [K|M] 使用绝对路径
[reuse]　重现数据文件，即先删除后创建，对表空间无效
[autoextend off|on]
[next integer [K|M]] 指定扩展的大小
[maxsize unlimited |integer [K|M]]
其中，storage_clause（默认存储参数子句）语法：
Storage (initial integer[K|M] 第一个区大小
Next integer[K|M] 下一个区和后续区大小
Pctincrease integer 指定第3个区及后续区在前面区基础上增加的百分比
Minextents integer 区的个数
Maxextents {integer | unlimited}) 区的最大个数
注意：1，区管理方式默认为本地管理，不明显指定区管理方式时不能创建字典管理的空间。
2，表空间默认为永久表空间(permanent)。段空间管理方式默认为manual。
3，datafile_clause中使用autoextend on就不能再指定区分配方式了。

4,自己指定blocksize时，要建立告诉缓冲区参数（db_2K_cache_size, db_4K_cache_size, db_8K_cache_size, db_16K_cache_size, db_32K_cache_size）才行，方法：**alter system set db_2K_cache_size=2K**等来临时设置会话的参数值。
5，默认创建的是smallfile表空间。Bigfile仅在本地管理方式中支持，bigfile的段管理方式不能是manual，也不能再向其中添加数据文件。**Alter database set default bigfile/smallfile tablespace** 更改表空间类型。
6，创建临时表空间(temporary)时要用tempfile指定文件路径，并且不能使用autoallocate的区分配方式，不能使用auto的段管理方式。
4,修改表空间
表空间的修改包括扩展表空间的存储空间（增加数据文件大小、增加数据文件、设置自动扩展、更改空间使用警告阀值）、修改表空间状态（读写、联机、脱机）和属性（位置、名称）等。

修改表空间语法：	针对数据文件的语法：
Alter tablespace 表空间名 [add datafile datafile_clause....] [add tempfile datafile_clause....] [rename to 新表空间名] [logging | nologging] [extent management dictionary | local] [autoallocate | uniform size integer [K|M]] [default storage_clause] [permanent | temporary] [minimum extent integer [K|M]] [read {write | only}] [online | offline [normal | temporary | immediate |for recover]] [begin backup |end backup];	Alter database [datafile '路径' **resize** integer [K|M]] [tempfile '路径' resize integer [k|M]] [**autoextend** on next inerger maxsize integer] [default tablespace 表空间名](默认表空间) [default temporary tablespace]（默认临时表空间）

注意：1，临时表空间不能够脱机。设置只读后，不能执行insert、update、delete操作，但是可以执行drop删除表空间的对象。当表空间脱机时，不能修改其读写状态。
2，当表空间或任何数据文件处于offline状态，不能修改表空间名称。
3,查询数据库默认表空间：**select username,default_tablespace,temporary_tablespace from dba_users;**
4,一个永久表空间不能设置为数据库的临时表空间。
5，删除表空间
应先将表空间脱机，再将其删除：
Drop tablespace 表空间名
[including contents] **表空间非空时使用（但数据文件要手动删除）**
[including contents casecade constraints] **同时删除与它相关的数据文件**
[including contents and datafiles] **连同数据文件一起删除**
6，查询表空间信息

查询基本信息	Seelct tablespace_name,extent_management,allocation_type,segment_space_management,status,contents,block_size from **dba_tablespaces**
表空间名称和存储参数	Select tablespace_name,initial_extent,next_extent,min_extents,max_extents,pct_increase,bigfile from **dba_tablespaces**
数据文	Select tablespace_name,file_name,blocks,bytes,autoextentsible,maxbytes From **dba_data_files**;

件信息	
临时文件信息	Select tablespace_name,file_name,blocks,bytes,autoextentsible,maxbytes From **dba_temp_files**;
空闲空间大小	Select tablespace_name,sum(bytes/1024/1024) from **dba_free_space** Group by tablespace_name
数据段信息	Select segment_name,segment_type,tablespace_name,extent_id,bytes,blocks From **user_extents**;

五，管理数据文件

1，数据文件：绝对文件号（相对于DB）、相对文件号（相对于TS）。出于速度和性能考虑，采取先内存（SGA）后外存的方法减少磁盘的I/O操作。Db_files初始化参数决定了数据库中数据文件的最大值，maxdatafiles是控制文件能保存的数据文件记录数，maxdatafiles会自动适应da_files。

2，创建数据文件

Create tablespace datafile|tempfile 创建表空间时创建数据文件

Alter tablespace ..add datafile|tempfile 向表空间添加数据文件时

3，查询数据文件的信息

在**dba_data_files**和**dba_temp_files**视图中查找数据文件信息和临时文件信息。

在**dba_tablespaces**中查找表空间的名称和默认存储参数，

如initial_extent和next_extent，是否是大文件等。

V$datafile动态视图与**dba_data_files**可以查询到许多动态的东西。它从控制文件角度看待数据文件信息，可以获得同步信息。

4，改变数据文件大小

不能创建没有指定大小的表空间或数据文件，也不能创建一个小于最小值的数据文件。

- 设置自动扩展：最好同时限定最大大小。可以在**create database**、**alter database**、**create tablespace**、**alter tablespace**的autoextend on子句中设置自动扩展。取消自动扩展使用autoextend off。
- 手动改变数据文件大小：在alter tablespace、alter database中使用**resize integer [K|M]**重新设置大小（也可以缩小文件）。

5，修改数据文件可用性

- 要修改文件可用性，要在archivelog模式下（默认为noarchivelog模式）。设置**archivelog**模式如下：

Select log_mode from v$database 查看目前的模式

Shutdown immediate 关闭数据库

Startup mount 以mount状态启动数据库

Alter database archivelog 设置为archivelog模式(改回去只改此句)

Alter database open 按新模式打开数据库

- 修改可用性：要使数据文件脱机：**alter databse datafile ‘’ offline;**要再次恢复联机，必须先进行介质恢复：**recover datafile ‘’;** 再进行联机：**alter databse datafile ‘’ online;**

6，移动(重命名)数据文件

- **Alter tablespace rename datafile ‘’ to ‘’**——移动同一个表空间的文件。使用alter修改数据文件位置或名称时，只是改变了数据文件的指针，其物理地

址并没有改变，因此要自己移动文件或**host move 原位置 新位置**命令（**先移动后使用alter**）。使用该命令要数据库**open**并且数据文件所在的**表空间脱机**。

- **Alter database rename file " to "**——移动多个表空间的文件，但这几个表空间要都脱机。
- System、sysaux等系统表空间不能脱机，所有要使数据库处于**mount**状态。并且移动要在alter database open之前。

7，删除数据文件

- 首先使表空间脱机，drop tablespace有三个选项（见上面），删除表空间后就可以手动删除数据文件或在命令中指定删除（仅对最新表空间有效）。

六，启动和关闭DB

1，初始化参数文件：pfile(可编辑纯文本文件)、spfile(不可编辑二进制文件)。Pfile的默认命名：**init%ORACLE_SID%.ora**;spfile的默认命名：**spfile%ORACLE_SID%.ora**。必须通过pfile来创建spfile。

Create pfile=" from spfile=":通过spfile导出形成pfile文件。**Create spfile=' ' from pfile="**:由pfile生成spfile。已经由例程使用的spfile不能重新创建和覆盖。

2，部分基本初始化参数：

参数	**说明**
Db_name	数据库的名称
Db_domain	分布式网络中网络域名，与db_name组成全局DB
Db_block_size	标准oracle块的大小
Db_nK_cache_size	非标准数据高速缓存的大小
Java_pool_size	Java池的大小，不小于20M
Sga_target	所有服务器进程可占用的SGA区的大小
Db_recovery_file_dest	恢复文件的文件夹位置
Remote_login_passwdfile	特权用户的验证方式，none操作系统方式、shared多个DB公用一个口令文件、exclusive口令只用于一DB
Undo_management	撤销管理的模式，auto使用undo表空间，manual使用回退段
Undo_tablespace	启动例程时使用的undo表空间
Processes	连接oracle并发进程的最大个数
Open_cursors	单会话可以同时打开游标的最大数目

3，更改初始化参数

Alter session set ...：只更改当前会话的参数设置。**Alter system set...commet=' ' scope=[spfile|memory|both]**：适用于例程的所有会话。

Scope子句	说明
Spfile	只更改spfile文件，对当前例程没有影响，在下一次启动生效，静态参数只能用此方式。
Memory	只更改内存或当前例程的参数，立即生效，但不保留（pfile默认）
Both	同时更改spfile文件和内存参数。（spfile默认）

Comment为参数更新添加相关的注释。Alter system更改参数范围是有限的，可以再v$parameter视图中查看（见后面）。

4，查看初始化参数

- **Show parameters spfile:**显示当前使用的spfile文件名称。**Show parameters 参数名**：显示该参数的参数值。
- **V$parameter视图：**该视图显示当前正在使用的参数值。其中，**isses_modifiable**表示是否可以由alter session修改（true可以修改）；**issys_modifiable**表示时候可以由alter system修改（immediate直接修改、defered需要defered选项修改）；**ismodified**表示时候已经更改过（modified为session修改的、system_mod为system修改的）。
- **V$apparameter视图：**显示当前例程使用的spfile的内容。

5，数据库启动的步骤；

启动例程（即nomount模式，读取初始化参数文件、分配内存启动后台进程

等）-》装载DB（即mount模式，可以增减重命名数据文件、DB的恢复、改变归档模式等）-》打开DB（即open模式）。

可以使用**alter database mount**和**alter database open**进入到下一个模式，但不可以回退到上一个模式。

6，关闭数据库

Shutdown [normal | transactional | immediate | abort];

- **Normal选项**：与shutdown作用相同，等待当前所有用户主动断开连接，所用时间取决于用户主动断开连接的时间。
- **Transactional选项**：等待当前未提交的活动事务提交完毕，主动断开用户的连接，可以保证用户不会丢失当前工作的信息。
- **Immediate选项**：任何当前未提交的事务均被回退，主动断开用户连接。如果有很多事务，会耗费很多时间。
- **Abort选项**：强行退出，立即结束正在执行的SQL语句，未提交事务不回退。可能丢失一部分数据信息，DB的完整性也会破坏，需要在下一次启动进行数据恢复。

7，启动数据库

Startup [nomount |mount | open | force] [restrict] [pfile=''']

- **Nomount选项**：只创建了例程，没有装载DB。可以进行的操作：运行创建新DB的脚本；重建控制文件。
- **Mount选项**：创建例程，装载DB，为打开DB做好准备。可以进行操作：增加、删除、重命名数据文件和重做日志文件；执行DB的完全恢复；改变DB的归档模式（archivelog和noarchivelog）。
- **Open选项**：正常启动。
- **Force选项**：正常启动遇到了困难，使用force先异常关闭DB，然后重启，不需要shutdown语句关闭DB。
- **Restrict选项**：将DB启动到open，但那只有restricted session权限的用户可以访问数据库，即受限会话模式。可以进行的操作：数据库数据的导入导出；数据装载操作；暂时阻止一般用户使用数据；DB移植和升级。**Alter system disable/enable restricted session**使一般用户可以使用DB。
- **Pfile选项**：指定使用文本初始化参数文件。

8，其他操作

- 更改DB读写方式（mount模式下）：**alter database open read only/write**。
- 暂停和重新开始DB：**alter system suspend** 暂停；**alter system resume** 重新开始。
- 管理会话：每个会话都有SID(标识符)和serial#(序列号)，可以在**v$session**视图中查询连接到DB的会话用户信息。
- 强行结束连接：**alter system kill session ‘sid，serial#’**；sid和serial#一起表示要终止的会话。如果允许先完成当前的事务再终止：**alter system disconnect session ‘sid，serial#’ post_transaction**。Post_transaction用immediate替换，立即回退事务（kill可以使用）。

七，网络服务与网络管理

1，oracle网络（oracle net services）最基本的功能是在一个网络中建立并维护应用程序和oracle服务器的连接。同版本的DB使用**数据库名**连接和该版本的网络驱动，并且可以进行复制与传输；不同版本的DB使用**数据库例程名**连接和较低版本的网络驱动。

2，数据库全局名：**数据库名.数据库域名**。DB的全局名就是DB的服务名。Oracle net用**连接描述符**作为客户机访问数据库服务的路径：

```
(DESCRIPTION =
  (ADDRESS=(PROTOCOL=TCP) (HOST=主机名) (PORT=端口号))
  (CONNECT_DATA=(SERVER=DEDICATED|SHARED) (SERVICE_NAME=全局名))
)
```

网络服务名是数据库服务在客户机端的名称，储存在客户机配置文

件tnsname.ora中，代替连接描述符（一个描述符可以配置多个网络服务名）。客户机连接口令：**用户名/口令@网络服务名**。

3，**监听程序**：运行在服务器端的一个单独的服务进程，网络回话是通过监听程序建立的，职责是通过监听端口号监听客户机的连接请求，并管理客户机与服务器之前的网络通信量。在服务器端配置文件listener.ora中。一台服务器可以运行多个监听程序，一个监听程序可以监听多个协议地址。

4，网络配置工具介绍：

- 监听程序配置文件：listener.ora（服务器端配置）；网络服务名配置文件：tnsnames.ora（客户端配置）；命名方法配置文件：sqlnet.ora（如果要配置命名方法，在客户端和服务器端要配置一致的文件）。路径：**%ORACLE_HOME%\NETWORK\ADMIN**。
- **Oracle net manager（ONM）**：客户端和服务器端都可以使用，可以完成：概要文件、网络服务名、监听程序。
- **Oracle enterprise manager(OEM)**：基于web的工具，在启动非默认监听程序方面有不可替代作用。
- **Oracle net configuration assistant（ONCA）**：可以配置DB的监听协议、命名方法、网络服务名和目录使用的配置。
- **lsnrctl**：在cmd输入lsnrctl进入该命令再使用services +监听名查看监听的信息。**Tnsping**：客户端的网络服务检测工具，检查客户机与监听程序的服务连接情况。

5，管理监听程序

- **服务器端添加监听**：在ONM默认监听程序中添加新地址端口，保存网络配置即可。
- **客户端网络服务名配置**：在ONM中选择**服务命名**，单击+号在向导中设置链接信息，最后进行测试以验证链接是否可用，最后文件-保存网络配置。
- **链接的端口问题**：端口号是DB与外界联系的通道，每个组件都有一个预先分配的端口号范围。Oracle允许的端口范围为：1~65535，端口号小于1024的是由操作系统中授权的进程使用的。

6，轻松连接命名方式

该方式不需要再客户端进行任何配置工作，还可以用主机IP代替主机名。格式：**用户名/口令@主机名：端口号/服务名**。服务名是可选的，即全局数据库名。

命名方式优先级：在ONM的**概要文件命令选项**中，可以设置优先级，其中EZCONNECT为轻松命名方式，TNSNAMES为本地命名方式。

7，拒绝和允许访问的客户机

在sqlnet.ora中保存该信息，**tcp.exclude_nodes:**被拒绝的客户端；**tcp.invited_nodes**:被允许的客户端；**tcp.vaildnode_checking**：是否对客户机访问权限进行检查。

在ONM的**概要文件一般信息的访问选项**中可以设置拒绝和允许的客户机。修改保存在sqlnet.ora中。

八，管理权限

1,权限分类

- 系统权限：系统级控制数据库存取和使用，执行某SQL语句的能力，是否能启动启动BD，连接BD，修改DB参数，是否能创建删除修改对象等。例如：**grant select any dictionary to 用户**：授予用户查询数据字典的系统权限。新创建的用户没有create session权限，不能连接登录数据库，授予该权限：**grant create session to 用户**。收回权限将grant改为revoke，to改为from。
- 对象权限：对象（表、索引、视图、过程）级控制数据库存取和使用，即访问其他用户方案对象的能力。例如：**Grant 权利 on 对象to用户** 给某用户授予权利。**Revoke 权利 on对象from用户** 撤销某用户权利。

2，授予系统权限

Grant { system_prililege | ALL [prilileges] }…

To {user |role |PUBLIC}..... 用户、角色公共用户组
[with admin option]; 表示可以将这些系统权限传递给别人

- 注意：专有权限一般只授予DB管理员。
- 新创建的用户即使有create table权限还是不能创建表，因为没有分配可使用表空间的配额：**alter user 用户 quota 大小 on 表空间名**。但无需在临时表空间分配配额。
- 查询系统权限：**system_privilege_map** 所有系统权限；**v$pwfile_users** 所有被授予sysdba、sysoper专有权限的用户信息；**dba_sys_privs** 授予所有用户和角色的系统权限；**session_privs** 当前会话可以使用的系统权限；**user_sys_privs** 授予当前用户的系统权限；

3，回收系统权限

Revoke { system_prililege | ALL [prilileges] }...
From {user |role |PUBLIC}.....

- 回收权限不会级联，A将权限a授予B，B又授予C，当B权限回收时，C仍然有a权限。

4. 授予对象权限

Grant {object_privilege |ALL [privilege] [列列表] }...
On [schema.]object 方案名/对象
To {user | role |PUBLIC }....
[with grant option] 可以传递权限，不能授予角色
[with hierarchy option];

- Insert，update，references这3种对象权限可以授予表视图中的列。
- **SQL*loader**：Oracle允许只读方式访问不在数据库中的表，要创建**目录对象**保存操作系统文件的位置：**create directory 目录对象名 as ‘路径’**。授予读取目录对象的权限：**grant read on directory 目录对象名 to 用户**。注意一定要有directory，回收时也是。可以使用select查询文件，用insert将文件装载到数据库。
- 查询对象权限：**dba_tab_privs** 数据库中所有用户角色的对象权限；**usr_tab_privs** 当前用户的对象权限；**dba_col_privs** 数据库中所有用户角色的列对象权限；**user_tab_privs_made** 当前用户授予别人的权限；**user_col_privs_made** 当前用户授予别人的列对象权限；**user_tab_privs_recd** 当前用户被授予的权限，类同**user_col_privs_recd**。

5，回收对象权限

Revoke {object_privilege |ALL [privilege] [列列表] }...
On [schema.]object
From {user | role |PUBLIC }....
[cascade constraints]; 回收references权限时使用

- 列权限必须一起回收，不能只回收部分列。
- 回收权限会级联，A将权限a授予B，B又授予C，当B权限回收时，C的a权限也丢失。
- A和B同时将权限a授予C，当删除A和B时，C仍然有a权限。

九，管理角色

1, 角色：将不同的权限集中在一起并进行命名。**预定义角色**是oracle自动创建一些对象，在%oracle_home%\RDBMS\ADMIN的desc.bsq脚本中。

- **Connect**：可以使用户链接登录DB，应该讲此角色授予任何需要访问DB的用户。
- **Resource**：可以使用户与其相关的方案，创建、更改、删除方案和对象，一般授予开发者和必须创建方案对象的用户。此角色的用户要再授予connect才能连接数据库。
- **DBA**：拥有所有的系统权限，默认授予sys和system。
- **Exp_full_database**用于数据库的导出操作；**imp_full_database**用于执行数据库的导入操作。
- **Execute_catalog_role,select_catalog_role,delete_catalog_role:**分别为执

行PL/SQL包，所有数据字典的select权限，所有数据字典的delete权限。

2，**创建角色**

Create role 角色名 {[identified by password] | [identified externally]}

其中By password要求用户启用该角色前先验证口令，默认是禁用的，启用角色：**set role 角色名 identified by password**。如果是externally则启用该角色前要通过一个外部服务。

3，角色权限管理

- 角色的系统权限和对象权限**和用户授予权限的语法一样**。**注意：**不能将系统权限unlimited tablespace和对象权限with grant option和with hierarchy option选项授予角色；不能用一条grant语句同时授予系统权限和对象权限。**回收权限和用户回收权限语法一样**。
- 更改角色：只能更改角色的验证方式，不能更改角色的名称。

Alter role 角色名 {[not identified] | [identified by password] | [identified externally]}

4，用户角色管理

- **给用户授予角色**：若给角色授予权限时有with admin option，则用户可以讲角色里德权限单独授予其他用户。

Grant role1 role2....
To {user | role |public}....
[with admin option] 默认隐含有的

- **回收用户角色**：无论是系统权限还是对象权限，回收角色都不会级联。

Revoke role1,role2....
From {user | role |public}....

- **更改用户默认角色**：all表示出了except的角色其余都是默认角色；none表示都不是默认角色。

Alter user 用户
Default role {role1.....|all [except rolex] | none}

- **启动和禁用角色**:只是在当前会话临时禁用，不会保存。Role1表示要启动的角色，except后为要禁用的角色，none为所有禁用。

Set role {role1 [identified by password]....| all except rolex | none};

- **删除角色**：**drop role**，删除该角色后就不再具有该用户的权限了

5，查询角色信息

- Dba_roles:数据库存在的所有角色；
- Dba_role_privs:授予用户或角色的角色;
- Use_role_privs:当前用户的角色；
- Role-sys_privs:授予角色的系统权限；
- Role_tab_privs:授予角色的对象权限；
- Session_roles:当前会话启用的角色。

6，PL/SQL过程与权限角色的关系

过程函数或包都是根据**定义者的命名空间和权限**来执行的，即只要定义者对某方案对象没有操作权限，该过程就不操作该对象。

当过程的定义者用authid current_user指定使用当前用户（调用者）的命名空间和权限后，调用者就可以利用该过程在自己 的方案中进行操作。

操作其他方案的对象要直接授予权限，不能通过角色简介授予。

十，管理概要文件

1,**创建概要文件：**

Create profile 概要文件名 limit
[cpu_per_session {n | unlimited | default}]
[cpu_per_call {n | unlimited | default}]
[connect_time {n | unlimited | default}]
[idle_time {n | unlimited | default}]
[sessions_per_user {n | unlimited | default}]
[logical_reads_per_session {n | unlimited | default}]
[logical_reads_per_call {n | unlimited | default}]
[private_sga {n [K|M] | unlimited | default}]
[composite_limit {n | unlimited | default}]
[failed_login_attempts {n | unlimited | default}]

[password_lock_time {n | unlimited | default}]_
[password_life_time {n | unlimited | default}]
[Password_grace_time {n | unlimited | default}]
[Password_reuse_max {n | unlimited | default}]
[Password_reuse_time {n | unlimited | default}]
[password_verify_function [function_name | null | default]]

- N是表示最大值的整数，unlimited表示没有显示，default表示使用默认概要文件(default)的值。
- 必须创建当前没有的文件，而且default概要文件不能再创建。

2，**更改概要文件**

Alter profile 概要文件名 limit
以下要修改的和创建时选项相同

- 对于默认概要文件default，不能创建和删除，但可以更改，更改某个参数，则使用该参数的其他文件都隐含的改变。
- Alter profile可以更改存在的选项值，也可以向概要文件中添加新的选项。

3，**分配概要文件**

- 可以再创建用户的时候分配：**profile profile_name**；
- 可以用改变用户的方式分配：**alter user 用户名profile 文件名。**

4，**删除概要文件**

Drop profile 文件名 [cascade]

- Cascade选项为为删除已经分配过的文件用的。

5，使用概要文件管理口令

- **登录失败和账户锁定：FAILED_LOGIN_ATTEMPTS**表示多少次登录失败后锁定用户；**PASSWORD_LOCK_TIME**表示锁定的用户多少天后自动解锁。
- **口令有效期和宽限期：PASSWORD_LIFE_TIME**口令的有效期；**pASSWORD_GRACE_TIME**口令的宽限期，不更改则账户过期，不能登录。手动使用户账户过期：**alter user 用户名 password expire**；
- **口令历史记录：PASSWORD_REUSE_MAX**口令在历史数据字典(user_history$)保存的最大个数；**pASSWORD_REUSE_TIME口令**口令重新使用的间隔天数。
- **口令复杂性校验函数：PASSWORD_VERIFY_FUNCTION**。在oracle主目录\RDBMS\ADMIN下的utlpwdmg.sql为一个校验函数的脚本，可以参考来定义校验函数。

6，使用概要文件管理内核资源

- **Sessions_per_user：**每个用户最多可以开启多少个SQL*PLUS。
- **logical_reads_per_session：**每次会话的最大读取次数，超过次数就会提示错误信息并注销。
- **composite_limit：**显示每个会话的组合资源，组合资源算法：

cost=（0*cpu_per_session）+(10*logical_reads_per_session)+(2*connect_time)+(0*private_sga)

加权可以再resource_cost中查看。

7，查询概要文件信息

视图	**说明**
Dba_profiles	所有概要文件及其参数值
Dba_users	用户所使用的概要文件信息
User_password_limits	分配给用户的口令资源参数
User_resource_limits	分配给用户的内核资源参数
Resource_cost	组合权限的加权值

十一，管理用户

1，oracle预设用户

用户名	口令	说明
Ctxsts	Ctxsys	Oracle test账户
Dbsnmp	Dbsnmp	由OEM管理代理人组件使用的账户，监测管理DB
Mddata	Mddata	由oracle spatial使用的方案的账户，用于存储geocoder和routershuju
Mdsys	Mdsys	Oracle spatial和intermedia locator的管理员账户

Dmsys	Dmsys	执行数据挖掘的账户
Olapsys	Manager	用于创建olap元数据的账户
Ordplugins	Ordplugins	Oracle intermedia插件的账户
Ordsys	Ordsys	Oracle intermedia插件的管理员账户
Sys		数据库管理员
Sysman		使用OEM执行数据库管理
System	Manager	与sys一样为管理员
Wksys		在oracle ultra search使用的账户

2，创建用户

Create user 用户名 [identified by 口令 | identified externally]
[default tablespace 默认表空间名]
[temporary tablespace 临时表空间名]
[quota [n K|M |unlimited] on 默认表空间名]
[quota [n K|M |unlimited] on 其他表空间名]
[profile [概要文件名 | default]]
[password expire]
[account lock] | [account unlock]

3,更改用户

alter user 用户名 [identified by 口令 | identified externally]
[default tablespace 默认表空间名]
[temporary tablespace 临时表空间名]
[quota [n K|M |unlimited] on 默认表空间名]
[quota [n K|M |unlimited] on 其他表空间名]
[profile [概要文件名 | default]]
[password expire]
[account lock] | [account unlock]
[default role [角色名 | all except 角色名 | none]

4,删除用户

Drop user 用户名[cascade] cascade用于已经拥有对象的用户
不能删除当前连接到数据库的用户。

5，identified externally外部身份验证

外部身份可以使客户操作系统，也可以是网络服务。这些外部验证账户格式：**XY\Z**，x为os authent prefix的值(默认为OPS$)，Y为DB服务器的计算机名，Z为操作系统的用户名。设置步骤：

- **确保sqlnet.authentication_services=(NTS)**。此参数在oracle主目录NETWORK\ADMIN中的sqlnet.ora文件中。同时remote_os_authent默认为false表示只有服务器所在计算机可以验证。
- **确认os_authent_prefix即前缀为OPS$**。使用前缀的好处是，如果DB中一些操作系统验证的账户具有不同的前缀，就可以修改改制启用禁止这些账户。
- **创建操作系统账户，必须授予administrators组**。
    - **创建数据库账户**，例如："ops$huge-PC\huge"。之后重启数据库，使用**connect **就可以登录。

DBA的操作系统验证：oralce安装后会在操作系统下建一个ORA_DBA的特殊组，将操作系统账户加入改组就可以，不需要再数据库创建账户等复杂的操作。

6，查询用户账户

视图	说明
Dba_users	数据库所有用户，包括概要文件
All_users	显示所有用户，但不描述他们
User_users	当前用户详细信息
User_password_limits	用户的口令资源参数
User_resource_limits	用户的内核资源参数
Dba_ts_quotas	所有用户在各个表空间的配额
User_ts_quotas	当前用户在各个表空间的配额
Dba_role_privs	授予所有用户的角色
User_role_privs	授予当前用户的角色

Dba_sys_privs	用户拥有的系统权限
Dba_tab_privs	用户拥有的对象权限
V$session	当前会话信息
V$sesstat	当前会话的统计信息
V$pwfile_users	特权用户的信息
Dba_tables	所有用户创建的表的信息
User_tables	用户所创建的表的信息

十二，管理表

1，创建表

```
Create [ global temporary] table [schema.]table_name
(col1  type1  [default 默认值] [列约束].....[表约束])
[on commit {delete | preserve} rows]   --指定哪种临时表
[organization [heap | index | external]]   --指定索引表还是外部表
[partition by...(...)]
[tablespace  表空间]
[logging | nologging]      --是否记录日志
[compress | nocompress]   --是否压缩
```

创建临时表：临时表的结构是一直存在的，但是数据时临时的，删除时都要有cascade constraint后缀。

- 事务临时表：数据只在当前事务内有效，提交或回退事务后消失，通过**on commit delete rows**指定。
- 会话临时表：通过**on commit preserve rows**指定，关闭会话后数据消失。

创建索引表：将表数据和索引数据放到一块，和带有索引的标准表是不一样的。索引表只适合那些存放数据变动较小的表。

- **Organization index**表示创建一个索引表，必须指定主键。
- **Pctthreshold**子句：定义数据块中为索引表保留的空间百分比，超过的行的任何部分都被存到溢出数据段。
- **Including**子句：控制哪些非键列要与键列存储在一起。
- **Overflow tablespace**子句：指定溢出数据段所在的表空间。

从一个表创建另一个表：基于已有的表或视图的列定义新表的列，并把查到的数据放入新表。

- **Create table 表名 as select语句**。不能指定tablespace，不能使用nologging选项，不能改变列的数据类型和长度。约束条件及列的默认值不会被复制。

2，更改变

添加列：新添加的列总是位于表的末尾。

```
Alter table 表名 add (列的定义...);
```

更改列：如果数据符合新长度可以缩小varchar的长度；更改数据类型，该列值必须为null；减小数值类型的精度和标度，相关列必须为null。

```
Alter table 表名 modify(l列名 列属性...)
```

删除列：可以将要删除的列设置为unused(部分删除),unused的列和删除的列没什么区别，但仍存在表中。

```
Alter table 表名 drop(列名) [cascade constraint]；

Alter table 表名 set unused(列...) [cascade constraint]   --部分删除
Alter table 表名 drop unused columns;    --完全删除unused的列
```

添加注释：

```
Comment on table 表名 is  '注释'
```

3，重新组织表

重命名：

- **Rename 旧表名 to 新表名**:该语句只能更改自己方案对象的名称。
- **Alter table 表名 rename to 新名**；该语句可以修改其他方案对象。

重新组织：将一个非分区表移动到一个新的数据段，或其他表空间**alter table...move...**。表的索引会失效，且默认不移动LOB数据。

4，删除表

删减表：即删除表中部分数据，可以使用delete语句、
先drop后create或**truncate**(截断)执行删减操作。**Truncate table 表名**：快速高效删除表中所有记录，不会影响触发器和其他对象和授权，且自动回收空间。
删除表：**drop table 表名 [cascade constraints]**
5,约束条件
约束的状态：

Enable validate	对新插入和原有数据都进行约束检查
Enable novalidate	对新插入和更新的数据进行约束检查
Disable validate	对原有数据检测，不允许进行DML操作
Disable novalidate	完全没有约束限制

定义约束：not null只能列级定义，不能再大对象和二进制类型的列上定义约束。

列级约束	[constraint 约束名] 约束类型 [条件]
表级约束	[constraint 约束名] 约束类型 {[列1...] | [条件] }

添加约束：可以在表定义时定义，也可以在之后用**alter table...add**添加约束。增加not null约束时必须用**modify代替add**。在定义外键约束时还可以用**on指定引用行为**，即父表记录被删除要怎么处理子表外键列（**delete cascade | delete set null**）。
删除约束：删除not null时用**modify代替drop**。删除外键约束可能需要cascade同时删除引用。

指定名称	Alter table ... drop constraint 约束名
指定某类型	Alter table ...frop 类型(约束名)

修改约束名:**alter table ...rename constraint 旧名 to 新名。**
设置约束状态：要先禁用所有子表的外键约束，再禁用主键约束。
Alter table ....disable/enable [validate | novalidate] constraint 约束名
设置约束的延迟检查：

- 对于不可延迟约束(not deferrable)，创建之后不能改变检查时机。
- 对于可延迟约束(deferrable)，可以指定initially immediate或initially deferred改变时机。前者是立即检查，后者是延迟检查。
- 延迟检查允许暂时违反检查，但不能通过提交。

Alter table... modify constraint 约束名 initially deferred/immediate --单个约束
Set constraints all deferred --全部延迟
Set constraints all immediate --取消延迟

6，外部表
外部表是不在数据库中，而在操作系统中存储，oracle以**只读**的方式访问。创建外部表实际上只是创建了外部表的元数据，对应外部表。
创建外部表：可以直接查询，也可以用insert装载到数据库。

create table 表名(列定义) organization external (type oracle_loader default directory 目录对象 access parameters (records delimited by newline fields terminated by ',' missing field values are null (列名列表)) location ('外部文件名'));	创建外部表前要先创建目录对象，里面保存着外部表，要有create any directory权限： **Create directory 目录对象 as ‘目录路径’;** 文件里分隔符一定是英文的,不能有空格

更改外部表：

- 更改默认目录：先将文件移到新目录，创建新目录：**Create directory 目录对象 as ‘目录路径’**，再更改目录：**alter table 表名 default directory 新目录**。
- 更改OS文件名：**alter table 外部表名 location(‘新外部表名’);**
- 修改访问分隔符：**alter table 外部表名 access parameters(fields terminated by ‘新分隔符’);**

删除外部表：先删除表：drop table...；再删除目录：drop directory ...

7,查询表信息

Dba_tables	数据库中所有的表	**All_tables**	用户可访问的所有表
User_tables	用户拥有的表	**D/a/u_tab_columns**	表视图簇的列信息
D/a/u_tab_comments	表和视图的注释	**D/a/u_col_comments**	列的注释
D/a/u_external_tables	外部表的属性信息	**D/a/u_external_locations**	外部表的来源
D/a/u_unused_col_tabs	部分删除(unused)的列	**D/a/u_partial_drop_tabs**	没有完全完成drop column操作的表

8，查看约束信息

D/a/u_constraints	约束的基本描述信息	**D/a/u_cons_columns**	约束被定义在那些字段上

十三，索引和视图

1,管理索引的策略

位图索引适合基数较小(1%)的，and/or/not的列	某列需要进行null相关的查询，选位图索引
B树索引会占用大量空间，位图索引空间小	一般不需要为很小的列创建索引
经常查询到的记录数目少于15%，可以建索引	不能对long，blob大对象类型建立索引
创建索引时将经常查询到的列放到前面	经常插入的表的索引要设置较高的**pctfree**(空闲)
Parallel选项，提高索引创建速度，默认不使用	**Nologging**选项，不产生重做日志，默认不使用
Compute statistics选项生成关于索引的统计信息	**Reverse**选项按逆序存储索引块字节，使B平衡
Nosort选项建索引同时对表数据排序	Reverse和nosort不能同时使用

2，创建索引

```
Create [unique] | [bitmap] index 索引名     --bitmap指定位图索引
On 表名（列名[ASC | DESC].... | 函数/表达式 ）  --函数索引
[tablespace 表空间名]
[pctfree n1]                --将来为insert预留的空间百分比
[storage(initial n2)]       --指定存储空间区分配第一个区大小
[compress n3] | [nocompress]
[logging] | [nologging]
[online]                    --指创建索引时允许DML操作，默认不执行
[conpute statistics]
[reverse] | [nosort]
```

- 在某列创建了unique索引，之后不能再该列创建其他索引。多列时，索引的列的顺序不同是合法的。
- 函数索引：实际上存储了事先计算过的值，下次查询时不需要计算where条件，不能是数值类型和null。

3，更改索引

- **Alter index。。。rename to。。。**：重命名。
- **Alter index 索引名 coalesce**：合并索引，可以清楚索引里的碎片。
- **Alter index 索引名 rebuild 选项**：重建索引，可以消除碎片和使用各种选项。如果频繁执行update和delete，应定期重建索引。
- **Alter index 函数索引名 enable/disable**:禁用和启用函数索引。
- **Alter index 索引名 allocate extent（size ...）**：分配给索引段空间；
- **alter index 索引名 deallocate unused**：释放分配的空间。
- **alter index 索引 monitoring usage**：使索引处于被监视的状态。之后在**v$object_usage**中查询该索引的使用情况。**Nomonitoring usage**关闭监视。

4，查看索引信息

D/a/u_indexes	所有表上的索引信息
D/a/u_ind_columns	数据库中表的索引列，即列的信息
D/a/u_ind_expressions	所有表的函数索引的函数或表达式
Index_stats Index_histogram	存储最后一条analyst index..validate structure产生的信息
V$object-usage	包含被监视的索引的信息
User_segments	索引占用的表空间和大小

5，视图是由select定义的一个逻辑表，**只有定义没有数据**，是“虚表”。视图的操作和表一样，所有针对视图的操作会影响到表。

Create [or replace] [force] view 视图名(列列表) --force强制创建，不考虑表的存在
As select语句
[with check option] [comstraint 约束名] --检查新插入数据是否能通过where条件
[with read only] --视图只读

- 简单视图：基于单个表，不包含任何函数，表达式和分组数据。
- 连接视图：基于多个表或视图创建，目的是为了简化连接查询。
- 复杂视图：select语句中包含函数、表达式和分组数据等视图。当select包含函数或表达式时，必须定义**列别名**。
- 强制创建的视图带有编译错误，处于失效状态，等到创建了表会自动重新编译视图。

6，更改视图

- 视图重定义：**create or replace**。视图重编译：alter view ...

7，删除视图：**drop view**

8，查询视图信息

D/a/u_updatable_columns	显示所有表和视图所有可修改的列
D/a/u_views	数据库中所有视图
D/a/u_tab_columns	描述数据库中的表、视图和簇的列

十四，同义词、序列

1，**同义词**：方案对象的别名。作用：可以简化对象访问；可以提高对象访问安全性。同义词不占用空间，只在数据字典里定义。

- **公有同义词**：由PUBLIC拥有，所有用户都可以使用。例如sys用户的数据字典视图。
- **方案同义词**：由创建它的用户拥有，也叫私有同义词。无论是否授予其他用户使用自己方案同义词对应的对象的权限，都不能使用方案同义词，**加上方案前缀才能使用**。

2，创建同义词

Create [or replace] synonym 同义词名 for 对象 --创建方案同义词
Create [or replace] public synonym 同义词名 for 对象 --创建公用同义词

在自己方案里创建指向其他方案对象的同义词，在授予访问该对象的权限后，可以访问该同义词。
在没有创建对象情况下也可以创建同义词，之后补上对象即可。

3，删除同义词

删除方案同义词：**drop synonym 同义词名**；
删除公用同义词：**drop public synonym 同义词名**；

4，查看同义词信息

D/a/u_synonyms:数据库中所有同义词；**d/a/u_db_links**:所有数据库连接名。

5，序列：命令的顺序编号生成器，可以设置为递增或递减，有界或无界，循环或不循环等方式。序列也不占空间，只在字典里有定义。

Create sequence 序列名
[start with n1] --第一个序列号
[increment by n2] --增量，默认为1，n2可正可负
[{maxvalue n3 | nomaxvalue}]
[{minvalue n4 | nominvalue}]
[{cache n5 | nocache}] --在缓存中可以预分配的序列号个数
[{cycle | nocycle}] --达到最值时是否循环，再从n1开始
[order]; --按顺序生成序列号

6，使用序列

引用当前序列值：sequence_name.**currval**;
引用下一个序列值：sequence_name.**nextval**;
在第一次引用currval前，要先引用一次nextval用于初始化序列，否则出现错误。
在**SQL语句中够可以直接引用序列值**。

7，更改序列：**alter sequence**语句。在对序列更改时将对视缓存里的序列值。起始值不能修改。

更改nextval：先改变increment by 值，然后选择nextval多次达到理想的值，再将increment 编译改回原值。

8，查看序列信息

D/a/u_sequences:数据库中所有序列。

十五，备份和恢复

1，备份和恢复方法

- **逻辑备份和恢复**：利用oracle的实用工具软件将数据库数据进行卸载和装入。具有多种方式，实现不同操作系统的数据传输，不同oracle版本的数据传输。
- **脱机备份和恢复**：关闭数据库情况下对数据文件的物理备份和恢复，简单直接。
- **联机备份和恢复**：数据库打开状态下进行备份和恢复，数据库必须处于归档模式(archivelog)，用户仍然可以访问数据。

2，数据泵(expdp和impdp)

- 创建目录对象：目录对象用于对于OS的某个目录。**Create directory as 路径**。要事先在OS中创建目录
    - 使用expdp**估计导出文件大小**：在cmd中输入：

expdp 用户/密码 schemas=用户方案 estimate_only=y estimate=statistics nologfile=y。

- 使用expdp**导出表**：cmd中输入：

expdp 用户/密码 directory=目录对象 dumpfile=导出文件名(.dmp) logfile=目录对象:日志文件(.log) tables=要导出的表;

- 使用expdp**导出方案**：cmd输入：

expdp 用户/密码directory=目录对象dumpfile=导出方案名(.dmp) logfile=目录对象:日志文件(.log) schemas=方案 filesize=大小 job_name=导出作业名称

- 使用expdp**导出表空间**：cmd输入

expdp 用户/密码dumpfile=目录对象:导出名(.dmp) logfile=目录对象:日志文件(.log) tablespace=表空间

- 使用impdp**导入数据**：cmd输入

Impdp 用户/密码 dumpfile=目录对象:导入名(.dmp) logfile=目录对象:日志文件(.log) content=data_only tables=表名

- 使用impdp**导入表**：cmd输入

Impdp 用户/密码 dumpfile=目录对象:导入名(.dmp) logfile=目录对象:日志文件(.log) tables=表名

- 使用impdp导入方案：cmd输入

Impdp用户/密码directory=目录对象dumpfile=导入方案名(.dmp) logfile=目录对象:日志文件(.log)

- 使用impdp导入表空间：cmd输入：

Impdp directory=目录对象dumpfile=导入表空间名(.dmp) logfile=目录对象:日志文件(.log) tablespace=表空间

3，脱机备份和恢复

要备份的文件有一下几类：

- 参数文件：在database目录中；
- 网络连接文件：在NETWORK\ADMIN目录中；
- 控制文件：查询视图v$controlfile；
- 数据文件：查询视图dba_data_files;
- 联机重做日志文件:查询视图v$logfile；

此方式备份就是上述几类文件的复制和转移，注意要在数据库关闭的状态下进行。