

Oracle 用户对象操作

Oracle 用户对象操作包括用户对表的创建、查询、修改、删除操作。为了加速查询速度,用户还必须建立按索引组织的表、簇表和分区表等,以便快速查询数据。为了隐藏逻辑数据的复杂性、使复杂的查询简单化,本章还介绍了视图及实体化视图的使用方法,使用户数据访问更加方便,系统安全性更好。本章还讲解了 Oracle 数据类型、序列的概念及使用。

通过本章的学习,读者应掌握以下重点内容:

- Oracle 表的创建及操作。
- 簇表和分区表的创建与使用。
- 表索引的种类及使用。
- 视图及实体化视图的概念及应用。

3.1 Oracle 表

Oracle 表是数据库中最基本、最重要的对象,用户应用数据库管理数据是通过表中数据的操作来完成的。为了进一步改善数据表的并发访问、连接和查询效率,Oracle 设立了多种形式的表。

3.1.1 Oracle 表的类型

在 Oracle 数据库中,表分为系统表和用户表两大类。系统表一般由数据库系统自行创建和管理,用于存放 Oracle 数据库的系统参数。用户表则是真正存储用户信息的地方,用户可以程序访问的方式或手工交互操作的方式对其操作,完成数据库管理的各项业务活动。

1. 系统表

Oracle 数据库通过系统表存放系统参数,可以通过 SQL * PLUS,以用户 system 权限进入,通过 select 命令来查询。

(1) 在如图 3-1 所示的 Log On 对话框中的 User Name 文本框中输入 system/sysqwp@orcl as sysdba, 单击 OK 按钮。

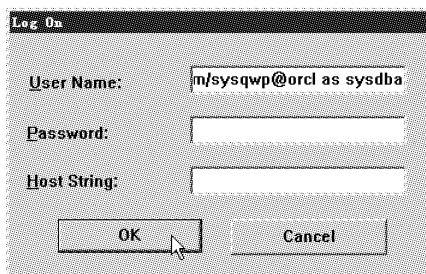


图 3-1 以 system 权限登录

(2) 出现如图 3-2 所示的 SQL * Plus 运行环境, 等待用户键入操作命令或运行程序。

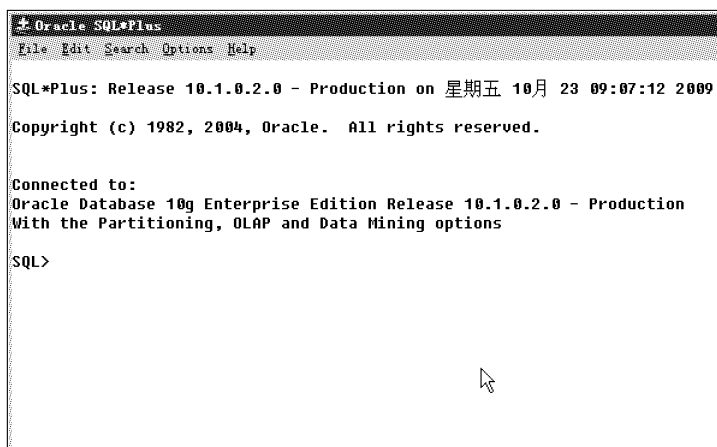


图 3-2 进入 SQL * Plus

(3) 键入 SQL * Plus 会话环境设置 set 命令和 select 查询命令即可查询系统表的内容, 如:

```
SQL>set pagesize 300;           --设置屏幕显示页的行数
SQL>set linesize 1000;          --设置屏幕显示页的列数
SQL>select * from v$controlfile; --查询系统表 v$controlfile 中控制文件的信息
```

屏幕显示 v\$controlfile 表中控制文件的路径如图 3-3 所示。

深入学习后, 用户有必要关心系统表中的内容, 以下几个重要的系统表都可以用上面介绍的查询语句进行读取, 例如:

```
SQL>select * from v$datafile;    --查询系统表 v$datafile 中数据文件信息
SQL>select * from v$log;         --查询系统表 v$log 中日志文件信息
SQL>select * from v$process;     --查询系统表 v$process 中处理器的信息
```

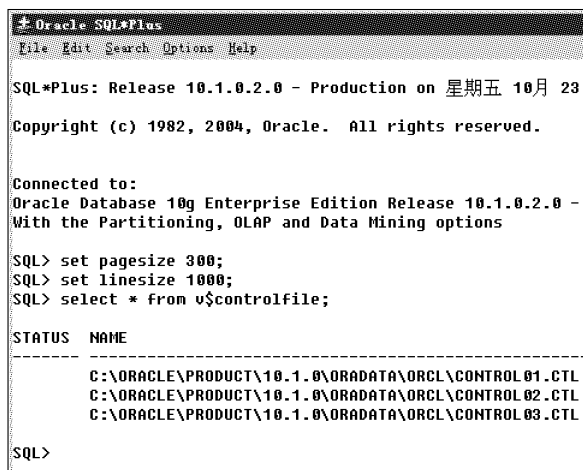


图 3-3 查看系统表的内容

```
SQL>select * from v$session;      --查询系统表 v$session 中会话信息
SQL>select * from v$transaction;  --查询系统表 v$transaction 中事务信息
SQL>select * from v$resource;     --查询系统表 v$resource 中资源信息
SQL>select * from v$sga;         --查询系统表 v$sga 中系统全局区的信息
```

2. 用户表

Oracle 用户表主要有以下 5 种类型。

- (1) 关系表：用来存储用户的永久数据。
- (2) 临时表：供系统使用，如查询结果等。
- (3) 按索引组织的表：用来存储数据和索引。
- (4) 外部表：数据库外部数据文件，只读。
- (5) 嵌套表：一个表被另外一个表引用。

其中关系表是用来存储和管理用户数据的对象。在 Oracle 中，还可对关系表创建分区表和簇表，大大加快了表的访问速度。

3.1.2 Oracle 数据类型

在关系表中，每一行对应一个对象，对象的各个属性对应关系表的各个列名。表 3-1 是属性列允许采用的常用标量数据类型。

需要注意以下两点：

(1) 由于 Oracle 不保证 Varchar 类型与今后新版本兼容，如果想与 Oracle 新版本的数据库兼容就不要用 Varchar，而要使用数据类型 Varchar 2。

(2) 由于 Long 型字段不能用于 GROUP BY 和 ORDER BY 子句及一个表只能包含一个 Long 类型的列，因此当需要使用一个或多个 Long 型时，均可以用 Clob 对象类型来



代替,一个表中允许有多个 Clob 对象类型。

表 3-1 常用标量数据类型

数据类型	说明
Char	定长字符串,长度 ≤ 2000 个字符
Varchar2	可变长字符串,长度 ≤ 2000 个字符
Number(P,S)	数值型,P(1~38):有效数字位,S(-84~127):小数位数
Date	日期型
TimeStamp	时间戳类型,可更精确地表示时间
Long	可变长字符串,长度 $\leq 2\text{GB}$,由于子查询建表时不能被复制,所以不建议使用
Raw	可变长二进制数据类型,长度 $\leq 4\text{KB}$
Long raw	可变长二进制数据类型,最大长度为 2GB,可用 Blob 代替
Blob	大二进制对象,长度 $\leq 4\text{GB}$
Clob	大字符串对象类型,长度 $\leq 4\text{GB}$
Nclob	国际字符集,大字符串对象类型,长度 $\leq 4\text{GB}$
Bfile	保存在外部的二进制文件类型,长度 $\leq 4\text{GB}$

3.1.3 表结构的创建

Oracle 数据表的操作始于数据表结构的创建,主要任务是根据确定的数据对象,确定属性列名和属性值类型,建立一个数据表的结构框架。创建数据表结构的方法主要有两种,最简单的是通过 SQL 命令创建表结构,方便、快捷,修改容易,可重复性好。还可以通过 OEM 创建,可视化程度高,修改更加方便。

1. SQL 命令创建表结构

在 SQL * Plus 中可以使用 SQL 命令操作数据库,也可以使用 SQL * Plus 命令设置 Oracle 环境变量及显示参数,命令字符不区分大小写。

用 SQL 命令创建表结构语法如图 3-4 所示。

从图 3-4 中可以看出,SQL 建表语法分为 4 个部分。

(1) 语法主干段:该部分为建表的语法主体,不可缺省。如果仅有语法主体段,就是简单表的创建语法,这时存储参数段和优化参数段均采用默认值。

(2) 存储参数段:该部分通过设定分区参数,使数据表存储性能得到优化。

(3) 优化参数段:该部分对每个数据块做了更精细的设置,使每个数据块存储时充分留有“余地”,以供数据修改时使用,保证修改后的数据仍然存放在一起,不会因为数据修改造成存储性能逐渐变差,确保 Oracle 数据库整体性能始终在最佳状态。

(4) 建表方式段:通过查询建表是一种快速建表方式,灵活、快捷,功能非常强大,在 SQL 命令建表和编程中经常使用。

语法主干段	CREATE TABLE[项目名.表名](		#SQL命令动词, 项目名可省略
	{列名1 数据类型 [NULL NOT NULL] [PRIMARY KEY]		#列名1, 类型, 非空约束, 主键约束
	[列级完整性约束] [[表级完整性约束]]		#约束
	,{列名2 数据类型 [NULL NOT NULL]		#列名2, 类型, 非空约束, 主键约束
	[列级完整性约束] [[表级完整性约束]]		#约束
存储参数段	...)		
	[STORAGE(		#存储参数段
	INITIAL n		#设置初始分区大小, n为KB或MB
	NEXT n		#下一个分区大小
	MINEXTENTS n		#最小分区数, 默认为1
	MAXEXTENTS n		#最大分区数, 单表最大99, 注意限制用户过多占有存储空间
	PCTINCREASE n)		#下一个分区增大的百分比, 设置为0时各个分区大小相同
优化参数段	[TABLESPACE 表空间名]		#设置表空间, 优化数据表的访问性能
	[PCTFREE n]		#指定在每个数据块中留出数据修改的空间, 优化存储性能
	[PCTUSED n]		#指定每个数据块最多使用空间百分比
	[INITRANS n]		#初始并发事务量, n可设置为5
	[MAXTRANS n]		#最大并发事务量, n可设置为25以上
建表方式段	[AS QUERY];		#使用此句是通过查询创建表, 没有此句就是直接建立表结构

图 3-4 SQL 建表语法图

在 SQL 建表语法中还有一些优化设置参数,例如 COLUMN 存储参数子句用来设置每一列的存储参数。CACHE 和 NOCACHE 设置是否使用缓存存储数据块,默认值是 NOCACHE。当设置了 CACHE,使用缓存存储数据块后,可以进一步加快数据的访问速度。

2. 用 SQL 命令创建简单表

1) 确定表的列名和数值类型

进入 Scott 用户中,Oracle 提供了一个公司员工表 emp,供读者练习数据表操作。下面通过多次使用这个 emp 表,熟练掌握数据表操作的基本命令和方法。

在 SQL * Plus 中,键入 SQL * Plus 的 DESCRIBE 命令:

```
SQL>DESCRIBE emp ✓ --SQL * Plus 的查看表结构命令,emp 是员工表
```

按回车键后,屏幕显示出员工表 emp 的表结构,如图 3-5 所示。

在图 3-5 中,员工表 emp 由以下 8 个属性列所构成。

(1) EMPNO: 员工编号。约束为非空字段,4 位数值型数据。



Name	Null?	Type
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)

图 3-5 员工表 emp 的表结构

(2) ENAME: 员工姓名。可变长字符串类型,10 个字符长。

(3) JOB: 工种。可变长字符串类型,9 个字符长。

(4) MGR: 上司编号。4 位数值型数据。

(5) HIREDATE: 聘用日期。日期型数据。

(6) SAL: 工资。7 位数值型数据,2 位小数。

(7) COMM: 奖金。7 位数值型数据,2 位小数。

(8) DEPTNO: 部门编号。2 位数值型数据。

2) 用 SQL 语句编写简单表结构

【例 3-1】 根据图 3-5 SQL 建表语法图中的“语法主干+存储参数+优化参数”格式来建立数据表,为了方便练习和对比,仿照 emp 的表结构建立一个新的表 emp_new1。

```
SQL>CONN scott/tiger@orcl          --以 scott 用户连接数据库
Connected.                          --屏幕显示已连接
SQL>DROP TABLE emp_new1;          --删除与待建数据表同名文件
表已删除                            --屏幕显示已删除
SQL>CREATE TABLE emp_new1(        --新建一个 emp_new1 数据表
2 EMPNO      NUMBER(4) Primary key, --把主键约束放在类型后面
3 ENAME      VARCHAR2(10),          --各列定义必须用半角','分隔开
4 JOB        VARCHAR2(9),           --以下各属性列和类型均与 emp 表相同
5 MGR        NUMBER(4),             --...
6 HIREDATE   DATE,                 --...
7 SAL        NUMBER(7,2),           --...
8 COMM       NUMBER(7,2),           --...
9 DEPTNO     NUMBER(2))             --如果在这一行结束,存储参数和优化参数默认
10 STORAGE(                          --设置存储参数
11     INITIAL 64KB                  --初始分区大小,以 B、KB 或 MB 为单位
12     NEXT 64KB                     --下一个分区大小
13     MINEXTENTS 1                  --最小分区数为 1 个分区
14     MAXEXTENTS 10                 --最大分区数
15     PCTINCREASE 0)               --下一个分区不增大
```

```

16 TABLESPACE users           --设置表空间名为 users
17     PCTFREE 10              --每个数据块保留 10%空闲空间用于以后修改
18     PCTUSED 40              --每个数据块至少要利用 40%的空间
19     INITRANS 10             --初始并发数为 10,最大并发数为 30
--用分号结束,分号后面不能加注释!
20     MAXTRANS 30;
21 /                           -- '/' 是开始运行符
表已创建                      -- 屏幕显示表已经建立
SQL> desc emp_new1            -- 键入查看表结构命令, 屏幕显示如图 3-6 所示

```

Name	Null?	Type
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)

图 3-6 新建表 emp_new1 的表结构

通过对比可以看出,新建表 emp_new1 与员工表 emp 表结构完全相同。通过这个例子,使用户迅速掌握建立表结构的方法,用同样的方法,可以对其他对象建立表结构。

3. 使用 OEM 可视化建表

在 Oracle 数据库服务器中使用企业管理器 OEM(客户机为 OEM)可视化建表,非常方便学习和设计。创建表的步骤如下:

(1) 进入企业管理器 OEM 的对象“表”。方法是:首先在浏览器地址栏键入并执行 `http://localhost:5500/em` 后,在企业管理器登录界面中输入用户名 sys、口令 sysqwp、连接身份选择 sysdba,单击“登录”按钮,即可以最高权限登录到 Oracle 企业管理器 OEM。在“管理”页面中选择方案中的“表”,打开表管理页面,如图 3-7 所示。

(2) 在“方案”文本框中输入 scott,在“对象名”文本框中输入 emp_new2,单击右下角的“创建”按钮,进入“表组织”页面,默认该页面的选项后单击右下角“继续”按钮,打开“一般信息”页面。

(3) 输入名称 emp_new2,方案 scott 后,仿照 emp 表的数据填写 emp_new2 表的各列名和数据类型和值长度。需要注意的是,EMPNO 名称要处于选中状态。在填写 SAL 时,number 类型大小设为 7;比例是小数位数,输入 2,即 2 位小数,如图 3-8 所示。



图 3-7 表管理页面



图 3-8 列信息设置页面

(4) 添加表约束。单击图 3-9 左下角“约束条件”选项,屏幕显示“表约束条件”页面,单击页面右侧“添加”按钮,对当前表添加主键(Primary key)约束,单击“添加”按钮,屏幕显示“对何列添加 PRIMARY 约束条件”页面,在“可选列”中选择 EMPNO 到“所选列”中,单击“确定”按钮。

(5) 选择“存储”标签,在事务处理数“初始值”中填写 10,“最大值”中填写 30,其他保持默认值不变。

(6) 选择“选项”标签,屏幕出现线程设置页,在该页中的“启动行移动”选默认,“并行值”中填写 10,并选中高速缓存选项。当基本设置完成,单击“显示 SQL”按钮,屏幕显示刚才设置过程形成的 SQL 命令:

```
CREATE TABLE "SCOTT"."EMP_NEW2" ( "EMPNO" NUMBER(4) NOT NULL , "ENAME" VARCHAR2(10) ,
"JOB" VARCHAR2(9) , "MGR" NUMBER(4) , "HIREDATE" DATE , "SAL" NUMBER(7, 2) ,
"COMM" NUMBER(7, 2) , "DEPTNO" NUMBER(2) , PRIMARY KEY ("EMPNO") VALIDATE) INI-
TRANS 10 MAXTRANS 30 PARALLEL 10 CACHE,
```

该命令与在 SQL * Plus 下手工键入的程序非常相似,仅格式略有不同,如图 3-9 所示。



图 3-9 显示生成的 SQL 语句

当全部参数设置好以后,可以单击图 3-10 右下角的“确定”按钮,在 OEM 中完成建立表结构的全过程。在 SQL * Plus 中用 DESC emp_new2 显示其表结构,与 emp 表完全相同,如图 3-5 所示。

4. 使用查询子句创建数据表

利用查询子句建立新表,不但能备份原表使其不被破坏,还能选择其部分列创建新表。创建新表时,以查询条件为源,可以快速在新表中得到满足查询条件的数据。

SQL 语法是:

```
CREATE TABLE 表名 [(列名,列名,...)] AS 子句;
```

需要注意的是:

(1) 原表中的 long 型字段不会复制到新表中。



图 3-10 结束设置用“确定”按钮

(2) 原表的完整性约束不会被复制到新表中,需要另外增加完整性约束。

(3) 新表的列类型不能变,但列名可以改变。

【例 3-2】 使用查询子句创建新表 emp_new3,新表取原表全部字段。

```
SQL>CONN scott/tiger@orcl
SQL>Drop table emp_new3;           --删除已有的 emp_new3 表
SQL>Create table emp_new3 as select * from emp;   --创建一个新表,原样备份 emp 表
SQL>Set pagesize 300
SQL>Set linesize 1000
SQL>DESC emp_new3;                 --查看表结构
SQL>Select * from emp_new3;        --查询并显示新建表中所有数据,如图 3-11 所示
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-12月 -80	800		20
7499	ALLEN	SALESMAN	7698	20-2月 -81	1600	300	30
7521	WARD	SALESMAN	7698	22-2月 -81	1250	500	30
7566	JONES	MANAGER	7839	02-4月 -81	2975		20
7654	MARTIN	SALESMAN	7698	28-9月 -81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-5月 -81	2850		30
7782	CLARK	MANAGER	7839	09-6月 -81	2450		10
7788	SCOTT	ANALYST	7566	19-4月 -87	3000		20
7839	KING	PRESIDENT		17-11月 -81	5000		10
7844	TURNER	SALESMAN	7698	08-9月 -81	1500	0	30
7876	ADAMS	CLERK	7788	23-5月 -87	1100		20
7900	JAMES	CLERK	7698	03-12月 -81	950		30
7902	FORD	ANALYST	7566	03-12月 -81	3000		20
7934	MILLER	CLERK	7782	23-1月 -82	1300		10

图 3-11 备份 emp 表到 emp_new3 并查看全部数据

【例 3-3】 使用查询子句创建新表 emp_new3,新表取原表 5 个字段。