

第 4 章 Oracle 数据库创建与表空间维护



本章介绍手工和使用数据库配置助手 DBCA 创建 Oracle 数据库的方法，对与创建数据库相关的最佳灵活体系结构 OFA 和 Oracle 管理的文件 OMF 做了介绍。本章还介绍了表空间、回退段管理的基本知识。



- 创建 Oracle 数据库之前的准备工作
- 使用 DBCA 创建数据库
- 在 Linux 下手工创建数据库
- 表空间的创建与删除
- 使用撤销表空间
- 回退段的创建与删除

4.1 创建 Oracle 数据库

4.1.1 创建 Oracle 数据库概述

一个完整的数据库系统应包括它的物理结构、逻辑结构、内存结构和进程结构，如果要创建一个新的数据库，则这些结构都必须完整地建立起来。有两种创建数据库的方法，即手工使用命令行方法创建，以及使用 DBCA 向导工具创建。

1. 数据库创建的时机

一般在数据库安装结束后，Oracle 会自动创建一个预定义的启动数据库，如果这个数据库不符合要求，或要在一个服务器中创建多个数据库，或原数据库的物理结构已经被破坏，但是系统文件还可以使用，这时可以重新创建一个新的数据库。

2. 创建数据库的先决条件

要创建新的数据库，需要满足一些必要条件。包括安装需要的 Oracle 软件，为操作系统设置各种环境变量，以及为软件和数据库文件在磁盘建立目录结构；具有相应的操作系统权限和 Oracle 数据库的 SYSDBA 系统特权；要有充足的磁盘存储空间以容纳规划的数据库（对一个数据库的数据文件，至少应准备 1.2G 的磁盘空间）等。

3. 规划数据库的文件存储位置

为确保数据库系统的安全，Oracle 要求至少存在两个控制文件，并且把它们分开存放在不同磁盘上。同样，应将重做日志组内的各重做日志文件分开存放于不同的磁盘上。规划数据库的文件存储位置是一项重要的工作，可以基于如下几点来进行规划工作：

- (1) 解决磁盘空间及 I/O 争用。
- (2) 依据不同的数据生存期（如临时数据和持久的应用数据）。
- (3) 依据不同的数据管理特性（如人工或自动管理）。
- (4) 数据文件的命名要合理、规范。

4. 创建数据库的方法

Oracle 提供了创建数据库的向导工具——数据库配置助手（DataBase Configuration Assistant, DBCA），使用 DBCA 可以简单快捷地创建自定义的数据库结构。此外，还可以利用手工方式创建数据库（即使用 CREATE DATABASE 命令创建），这种方法是传统的创建数据库的方法，可适用于所有的 Oracle 版本。

5. 安装数据库之前的准备工作

准备工作包括规划数据库表和索引，并估计它们所需的空间大小；规划数据库包含的底层操作系统文件的布局；考虑采用 Oracle 管理的文件特性来创建和管理重做日志文件、控制文件；选择全局数据库名称；熟悉初始化参数和初始化参数文件；选择数据库字符集；选择标准的数据库块尺寸；使用撤销表空间来管理撤销记录，而不是用回退段；开发一套备份和恢复策略以防数据库失败；熟悉启动和关闭例程以及装载和打开数据库的准则和选项。

4.1.2 OFA 与 OMF

1. 最佳灵活体系结构

Oracle 10g 最佳灵活体系结构（Optimal Flexible Architecture, OFA），是指 Oracle 软件和数据库文件及目录的命名约定和存储位置规则，它使用户可以很容易地找到与 Oracle 数据库相关的文件集合。简单的说，OFA 就是 Oracle 公司推荐的一种在服务器上安装 Oracle 数据库系统的目录规划，以便维护 Oracle 数据库系统。

使用最佳灵活体系结构，能够简化数据库系统的管理工作，使数据库管理员更加容易地定位文件或添加文件。还可以将 Oracle 系统的执行文件、管理文件、数据文件分别存储到不同的硬盘上，从而有效地使用用户计算机系统中的所有存储空间，克服某个硬盘存储空间的限制。通过合理分配存储负载，还能降低单个硬盘存储速度方面的瓶颈，提高整个系统的存储效率。

OFA 将整个数据库系统分为 Oracle 软件产品、数据库例程的进程日志文件、数据库物理文件三部分，并分别放在不同的目录下。

安装 Oracle 数据库需要一个根目录，这就是 ORACLE_BASE，在这个目录下包括所有安装的 Oracle 产品和所有数据库例程的进程日志文件。一般将 Oracle 数据库软件所在的目录称为 Oracle 主目录，即 ORACLE_HOME，而将所有数据库例程的进程日志文件统一放到 ORACLE_BASE 下的 admin 目录下，并在该目录下为每个例程建立一个专用目录以管理它的进程日志文件。也就是说，在 ORACLE_BASE 下有两个目录，分别是 ORACLE_HOME 对应的目录和 admin 目录。

数据库物理文件所在的目录命名为 oradata，该目录可以位于 ORACLE_BASE 下面，也可以放到别的磁盘上。

下面是 OFA 的一般目录结构示意图。

Oracle 软件产品目录结构如下：

```
oracle_base/  
  /product  
    /release_number-----发行版本号，这个目录一般作为 ORACLE_BASE
```

```

/admin-----内核进程跟踪文件目录
  /inst_name---每个例程有专门的对应目录，以例程名命名
    /adump
    /bdump
    /pfile
    :
/client_1-----客户端第 1 次安装建立的客户端工具目录
  /bin-----客户端工具的可执行程序目录
  :
/db_1----第 1 次安装 Oracle 建立的主目录，该目录一般作为 ORACLE_HOME
  /bin-----可执行程序目录
  /dbs-----初始化参数、口令文件所在目录，Windows 有 database 目录
  /NETWORK-----监听进程等 Oracle Net 相关配置文件所在目录
  /RDBMS-----RDBMS 脚本文件所在目录
  /sqlplus-----SQL*Plus 工具所在目录
  :

```

数据库物理文件目录结构如下：

```

oradata/
  orcl/-----数据库 orcl 物理文件目录，每个数据库有一个对应目录
    SYSTEM01.DBF
    CONTROL01.CTL
    REDO01.LOG
    :
  CEMERP/-----数据库 CEMERP 物理文件目录
    SYSTEM01.DBF
    CONTROL01.CTL
    REDO01.LOG
    :

```

注意，Oracle 将重要的文件名和目录名大写！通过合理地规划目录结构，可以很方便地管理和维护 Oracle 数据库系统。比如物理备份时，只需将 `dbs`、`NETWORK` 和 `oradata` 目录下相应文件备份即可，操作简单而且清楚地知道文件在哪里。

2. Oracle 管理的文件

使用 Oracle 管理的文件（Oracle Managed Files, OMF）可以简化对 Oracle 数据库的管理。通过 OMF，不需要 DBA 直接管理 Oracle 数据库在操作系统上的文件。OMF 使用标准的文件系统接口创建和删除跟踪文件、审计文件、报警文件、核心文件等，并根据数据库对象而不是文件名来操作。

通过初始化参数，可以为特定类型的文件指定文件系统目录，并通过指定文件名规则，Oracle 可以确保为其创建唯一的 Oracle 管理的文件，并在不需要的时候将该文件删除。与 OMF 相关的初始化参数如下：

DB_CREATE_FILE_DEST: 设置所创建的数据文件、控制文件和联机日志的默认位置。该目录必须已存在，并且该目录必须具有可让 Oracle 在其中创建文件的权限。

DB_CREATE_ONLINE_LOG_DEST_n: 设置所创建的联机日志和控制文件的默认位置，`n` 表示多路备份的不同位置。如果在创建联机日志或控制文件时未指定文件名，则使用默认文件名。同样，该目录必须已存在，并且该目录必须具有可让 Oracle 在其中创建文件的权限。

以后只要创建文件时没有指定目录，就把文件放到参数指定的目录下。

OMF 使用例程名、进程名、表空间名、重做日志组号等格式字符串（形如 `%G`、`%R`、`%S`、`%T`、`%U` 等）自动命名相关文件。在不同的服务器操作系统下，格式字符串可能不同。在指

定文件名时，可以使用这些格式字符串，Oracle 将按照命名规则自动生成不重复的文件名。

在创建数据库时可以包含 LOGFILE、DATAFILE、UNDO TABLESPACE、TEMPORARY TABLESPACE 等子句来指定创建的文件名和地址。在创建数据库过程中未指定文件名，且已经配置了初始化参数 Oracle 管理的文件地址，则将相应文件创建为 Oracle 管理的文件。即，通过如下两步来创建 Oracle 管理的文件：

(1) 在初始化参数文件中设置 DB_CREATE_FILE_DEST 和 DB_CREATE_ONLINE_LOG_DEST_n 参数。

(2) 在 CREATE DATABASE 命令中只给出文件名（不要给出目录路径），Oracle 会自动将其创建在指定的目录并自动管理该文件。

4.1.3 使用数据库配置助手（DBCA）创建数据库

Oracle 数据库配置助手（简称 DBCA）是一个图形用户界面（GUI）工具，它可与 Oracle 通用安装程序进行交互，也可以单独使用，它的主要作用是简化数据库的创建过程。

Oracle 数据库配置助手向导指导整个数据库配置的选择过程。可以提供最少的输入，并让 Oracle 替用户作出决策，从而节省决定如何最好地设置数据库的参数或结构的时间。它也允许指定参数设置和文件分配方面的内容。

使用数据库配置助手可以创建或删除数据库，还可以对数据库选项进行配置，以便添加先前没有配置的选项。除此之外，数据库配置助手可以创建和管理数据库模板。用户可以先创建数据库定义的模板，然后对其进行修改，或者修改 Oracle 提供的模板。还可以先创建数据库模板，再进行复制。

下面简要说明在 Windows 环境下使用 DBCA 创建数据库的过程。

在 Windows 环境中选择【开始】/【程序】/【Oracle - OraDb10g_home1】/【配置和移植工具】/【Database Configuration Assistant】即可启动 DBCA。然后选择“创建数据库”引导用户开始创建数据库所需的设置工作。用户可根据需要选择一个模板来简化数据库的定义过程。DBCA 提供了如表 4-1 所示的几种典型数据库应用对应的数据库模板。

表 4-1 DBCA 为多种应用环境提供的模板

模板类型	说明
一般用途	选择此选项可以创建适用于混合工作负荷环境（OLTP 和数据仓库）的数据库。这是要安装的默认数据库类型
事务处理	选择此选项可创建一种适合 OLTP 环境的数据库环境。OLTP 环境中的数据库每天必须处理来自许多并发（同时连接的）用户的成千乃至上百万的事务。用户必须能够快速访问最新数据。因此，数据库的性能取决于数据吞吐量（速度）和数据可用性
定制数据库	选择此选项可以自定义方式创建数据库。选择“定制数据库”还可以自动为数据库配置共享服务器支持（又称为多线程支持）
数据仓库	选择此选项可创建一种适合数据仓库环境的数据库。数据仓库环境中的数据库必须处理各种各样的查询（通常是只读查询，如 SELECT 语句），包括从几条记录的简单读取到从许多不同的表中查询数千条记录的大量复杂查询。因此，数据库性能取决于响应时间

在配置安装的第二步选择一个类型的模板，单击“显示详细资料”将显示数据库默认参数，也可再单击“另存为 HTML 文件”保存数据库参数到磁盘文件。读者在此可体会 ORACLE_BASE, DB_NAME 以及数据库安装需要的其他参数的应用，搞清楚是哪些参数在影响不同用途的数据库的配置，为后面使用 CREATE DATABASE 命令打下基础。这些信息包

括以下方面：

- (1) 常用选项。
- (2) 初始化参数。
- (3) 字符集。
- (4) 数据文件和位置。
- (5) 控制文件及其位置。
- (6) 重做日志文件及重做日志组。

在选择模板的同时，可以指定数据库定义中是否包含数据文件，相应地会创建包含数据文件的数据库和不包含数据文件的数据库。

在模板选择完成后，DBCA 继续指导用户进一步定义数据库或者接受由 Oracle 推荐的默认的参数值和文件位置。包括提供一个全局数据库名称、指定要包含的数据库选项、决定采用的模式（共享服务器或专用服务器模式）以及最后指定初始化参数。初始化参数包括 SGA 的大小，以及数据、控制和重做日志文件设置、表空间大小、范围尺寸、存档格式和目标位置、跟踪文件目标位置、字符集规范等。

在完成定义数据库的各种参数的设置之后，可以将对参数的说明保存为数据库模板，并且生成用于创建数据库的脚本。该模板将自动添加到数据库创建模板的可用列表中，该脚本由用户在前面的界面中指定的数据库参数生成。在参数设置的最后，单击【完成】按钮开始创建数据库过程，用户只需等待操作完成即可。

假定要开发一个企业信息化软件，命名为现代企业管理 ERP 系统，该软件的 Oracle 数据库命名为 `cemerp`，则配置过程要点如下：

- (1) 启动 DBCA，在步骤 1 选择“创建数据库”，然后在步骤 2 选择“一般用途”。
- (2) 在步骤 3 “数据库标识”界面，全局数据库名和 SID 处均输入 `cemerp`。
- (3) 在步骤 4 “管理选项”，采用默认值。
- (4) 在步骤 5 “数据库身份证明”界面，选择“所有账户使用同一口令”，并在口令和确认口令处输入 `syspwd`，待数据库创建完毕再集中修改各用户的口令。
- (5) 在随后的步骤 6 “存储选项”、步骤 7 “数据库文件所在位置”、步骤 8 “恢复配置”采用默认值。
- (6) 在步骤 9 “数据库内容”界面，勾选“示例方案”，安装 HR 等示例方案。
- (7) 在步骤 10 “初始化参数”界面，均采用默认值。建议读者在该界面依次查看一下各选项卡下的内容，将 Oracle 数据库体系结构的知识结合起来理解。
- (8) 在步骤 11 配置“数据存储”，采用默认值即可。同样，建议读者将该界面上左边的树形结构展开，依次查看其中内容。该界面还有一个【文件位置】按钮，一定打开看看，其中显示了 `ORACLE_BASE`、`ORACLE_HOME`、`DB_NAME` 和 `SID` 值。
- (9) 步骤 12 是“创建选项”，勾选“创建数据库”和“生成数据库创建脚本”。“生成数据库创建脚本”建议初学者勾选，在数据库创建完成后可在 `ORACLE_BASE\admin\cemerp\scripts` 下查看到创建 `cemerp` 数据库的脚本文件。通过分析这些文件，了解 Oracle 是如何创建数据库的。单击【完成】按钮开始创建数据库过程，系统显示创建进度。
- (10) 在创建结束时，弹出 DBCA 口令管理窗口，单击【口令管理】按钮设置 `SYS` 口令为 `syspwd`，`SYSTEM` 口令为 `systempwd`，`SCOTT` 口令为 `tiger`，`HR` 账户口令为 `hrpwd`。
最后单击口令管理界面的【退出】按钮结束使用 DBCA 创建数据库 `cemerp` 的过程。
可以使用用户名 `scott`，口令 `tiger`，主机字符串 `cemerp` 登录 `SQL*Plus` 进行测试，还可以

从控制面板查看 cemerp 数据库的 Windows 服务例程情况。在 tnsnames.ora 文件中能看到 cemerp 主机字符串的定义情况，参见 3.1.3 节。

4.1.4 手工创建 Oracle 数据库

手工创建数据库需要在磁盘上建立相应的目录结构，其他步骤如下：

- (1) 决定例程标识符 (Instance Identifier, SID)。
- (2) 建立数据库管理员验证方法。
- (3) 创建初始化参数文件。
- (4) 在 SQL*Plus 中连接到例程。
- (5) 启动例程。
- (6) 发布 CREATE DATABASE 语句。
- (7) 创建其他表空间。
- (8) 运行脚本来构建数据字典视图。
- (9) 运行脚本来安装其他的选项。

下面以在 RedHat 5 企业版中创建 oradb 数据库为例说明手工创建 Oracle 的过程。关于在 Linux 下如何安装、配置 Oracle 数据库，参见附录。

1. 在磁盘建立相应的目录结构

以 oracle 用户登录 Linux，在终端下键入如下命令手工创建所需目录：

```
[oracle@localhost ~]$ mkdir -p /home/oracle/product/10.2.0/admin/oradb/adump
[oracle@localhost ~]$ mkdir -p /home/oracle/product/10.2.0/admin/oradb/bdump
[oracle@localhost ~]$ mkdir -p /home/oracle/product/10.2.0/admin/oradb/cdump
[oracle@localhost ~]$ mkdir -p /home/oracle/product/10.2.0/admin/oradb/dpdump
[oracle@localhost ~]$ mkdir -p /home/oracle/product/10.2.0/admin/oradb/udump
[oracle@localhost ~]$ mkdir -p /home/oracle/product/10.2.0/admin/oradb/pfile
[oracle@localhost ~]$ mkdir -p /home/oracle/product/10.2.0/oradata/oradb
```

2. 设置操作系统环境变量

在 oracle 用户终端下键入如下命令设置所需环境变量：

```
[oracle@localhost ~]$ ORACLE_BASE=/home/oracle/product/10.2.0
[oracle@localhost ~]$ export ORACLE_BASE
[oracle@localhost ~]$ ORACLE_HOME=$ORACLE_BASE/db_1
[oracle@localhost ~]$ export ORACLE_HOME
[oracle@localhost ~]$ ORACLE_SID=oradb
[oracle@localhost ~]$ export ORACLE_SID
```

3. 创建初始化参数文件

创建 Oracle 数据库需要使用到文本型初始化参数，参数名为 initSID.ora。本例中文本参数文件名为 initoradb.ora，应保存在 \$ORACLE_HOME/dbs 目录下。

在安装 Oracle10g 时，就会将一个示例 init.ora 文件存放在 \$ORACLE_HOME/dbs 中。因此可将该文件复制并命名为 initoradb.ora，然后对相应参数进行修改即可，需要修改的项包括 db_domain、db_name、control_files、background_dump_dest、user_dump_dest、core_dump_dest 和 undo_management 等。创建 oradb 数据库的初始化参数文件内容如下：

```
oradb._db_cache_size=188743680
oradb._java_pool_size=4194304
oradb._large_pool_size=4194304
oradb._shared_pool_size=83886080
oradb._streams_pool_size=0
```

```

*audit_file_dest='/home/oracle/product/10.2.0/admin/oradb/adump'
*.background_dump_dest='/home/oracle/product/10.2.0/admin/oradb/bdump'
*.compatible='10.2.0.1.0'
*.control_files='/home/oracle/product/10.2.0/oradata/oradb/control01.ctl','/home/oracle/product/10.2.0/oradata/oradb/control02.ctl','/home/oracle/product/10.2.0/oradata/oradb/control03.ctl'
*.core_dump_dest='/home/oracle/product/10.2.0/admin/oradb/cdump'
*.db_block_size=8192
*.db_domain=""
*.db_file_multiblock_read_count=16
*.db_name='oradb'
*.db_recovery_file_dest='/home/oracle/product/10.2.0/flash_recovery_area'
*.db_recovery_file_dest_size=2147483648
*.dispatchers=(PROTOCOL=TCP) (SERVICE=oradbXDB)
*.job_queue_processes=10
*.open_cursors=300
*.pga_aggregate_target=94371840
*.processes=150
*.remote_login_passwordfile='EXCLUSIVE'
*.sga_target=285212672
*.undo_management='AUTO'
*.undo_tablespace='UNDOTBS1'
*.user_dump_dest='/home/oracle/product/10.2.0/admin/oradb/udump'

```

4. 创建口令文件

创建 SYS 用户的口令文件，Linux 中文件名称为 `orapwSID.ora`。本例中为 `orapworadb.ora`，存放于 `$ORACLE_HOME/dbs` 目录。此处将口令指定为 `syspwd`，使用如下命令创建：

```
[oracle@localhost ~]$ orapwd file=$ORACLE_HOME/dbs/orapworadb.ora passwd=syspwd
```

也可手工复制一个口令文件，然后修改文件名为指定的名字即可。

5. 启动例程创建数据库

在设置环境并配置初始化参数文件 `initoradb.ora`、创建口令文件后，就可以通过在 SQL*Plus 中执行 `CREATE DATABASE` 命令创建数据库了。

`CREATE DATABASE` 命令指定了日志文件的数量和位置、SYSTEM 表空间、UNDO 表空间和 TEMP 表空间的位置和大小，以及数据库使用的字符集等。在该命令中设置的 UNDO 表空间要和初始化参数文件中的设置对应。

在创建数据库时，Oracle 10g 服务器仅识别 SYS 用户和 SYSDBA 角色。要创建数据库，必须以用户 SYS 和角色 SYSDBA 的身份连接到 SQL*Plus。

示例：

```

[oracle@localhost ~]$ sqlplus /nolog
SQL> CONN / AS SYSDBA
SQL> startup pfile=/home/oracle/product/10.2.0/db_1/dbs/initoradb.ora nomount
SQL> CREATE DATABASE oradb
 2  datafile '/home/oracle/product/10.2.0/oradata/oradb/system01.dbf'
 3     size 500M reuse autoextend on next 10240K maxsize unlimited
 4     extent management local
 5  sysaux datafile
 6     '/home/oracle/product/10.2.0/oradata/oradb/sysaux01.dbf'
 7     size 300M reuse autoextend on next 10240K maxsize unlimited
 8  default temporary tablespace temp
 9     tempfile '/home/oracle/product/10.2.0/oradata/oradb/temp01.dbf'
10     size 20M reuse autoextend on next 640K maxsize unlimited

```

```

11 undo tablespace UNDOTBS1
12   datafile '/home/oracle/product/10.2.0/oradata/oradb/undotbs01.dbf'
13   size 100M reuse autoextend on next 5120K maxsize unlimited
14   logfile
15 group 1 ('/home/oracle/product/10.2.0/oradata/oradb/redo01.log') size 50M,
16 group 2 ('/home/oracle/product/10.2.0/oradata/oradb/redo02.log') size 50M,
17 group 3 ('/home/oracle/product/10.2.0/oradata/oradb/redo03.log') size 50M
18 CHARACTER SET ZHS16GBK;

```

6. 建立数据字典和 PL/SQL 相关包

必须在创建数据库之后运行脚本 `catalog.sql` 和 `catproc.sql`，这两个脚本位于 `$ORACLE_HOME/rdbms/admin` 目录中。`catalog.sql` 脚本用于创建数据字典视图，`catproc.sql` 用于创建使用 PL/SQL 所需的包和过程。必须以 SYS 的身份运行这两个脚本。在创建脚本之前，确保数据库已打开。执行的命令如下：

```

SQL> @$ORACLE_HOME/rdbms/admin/catalog.sql
SQL> @$ORACLE_HOME/rdbms/admin/catproc.sql

```

执行 `pupbld.sql` 脚本，以允许所有用户都可以使用 SQL*Plus。必须以 SYSTEM 用户身份执行该脚本，命令如下（如果是刚创建的数据库，需要先修改 SYSTEM 用户的口令）：

```

SQL> CONN system/systempwd
SQL> @$ORACLE_HOME/sqlplus/admin/pupbld.sql

```

7. 创建 USERS 表空间

执行如下语句创建 USERS 表空间：

```

SQL> CONN / AS SYSDBA
SQL> CREATE TABLESPACE users
      2   DATAFILE '$ORACLE_BASE/oradata/oradb/users01.dbf' SIZE 20M;

```

然后运行 `utlsampl.sql` 脚本文件创建 SCOTT 示例账户：

```

SQL> @$ORACLE_HOME/rdbms/admin/utlsampl.sql

```

创建完 SCOTT 账户后，应该进行连接测试：

```

SQL> CONN scott/tiger
Connected.

```

8. Oracle Net 配置

数据库创建完成后，必须进行 Oracle Net 配置，这样用户才可以和新建的数据库连接。Oracle Net 配置包括两个部分，监听进程配置文件 `listener.ora` 及本地 Net 服务名配置文件 `tnsnames.ora` 的配置，这两个文件位于 `$ORACLE_HOME/NETWORK/admin/` 目录下。

在每一个 Oracle 服务器中都有监听进程，该进程决定用户是否可以通过某个网络协议与数据库进行连接。如果没有正确配置监听进程，则所有用户都不能连接到新创建的数据库上。

可直接用 vi 编辑器手工修改 `listener.ora` 文件，笔者计算机中该文件内容如下：

```

SID_LIST_LISTENER =
  (SID_LIST =
    (SID_DESC =
      (SID_NAME = PLSExtProc)
      (ORACLE_HOME = /home/oracle/product/10.2.0/db_1)
      (PROGRAM = extproc)
    )
  )
)

LISTENER =
  (DESCRIPTION_LIST =

```

```

        (DESCRIPTION =
          (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1))
          (ADDRESS = (PROTOCOL = TCP)(HOST = localhost.localdomain)(PORT = 1521))
        )
      )

```

同样直接用 vi 编辑器修改本地 Net 服务名配置文件 tnsnames.ora，笔者计算机中该文件内容如下（其中 ORADB 主机字符串部分是应该新添加的）：

```

ORADB =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = localhost.localdomain)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = oradb)
    )
  )
)
ORCL =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = localhost.localdomain)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = orcl)
    )
  )
)
EXTPROC_CONNECTION_DATA =
  (DESCRIPTION =
    (ADDRESS_LIST =
      (ADDRESS = (PROTOCOL = IPC)(KEY = EXTPROC1))
    )
    (CONNECT_DATA =
      (SID = PLSExtProc)
      (PRESENTATION = RO)
    )
  )
)

```

9. Linux 环境中多数据库例程的自动启动与关闭

在安装 Oracle 数据库软件时，创建了启动数据库 orcl。现在又手工创建了 oradb 数据库，可以考虑让这两个数据库随操作系统自动启动和关闭。

Oracle 10g 提供了 3 个文件用于数据库的自动启动及关闭，如果要在服务器上创建多个例程，可以结合使用这 3 个文件配置数据库的自动启动与自动关闭，3 个文件分别是/etc/oratab、\$ORACLE_HOME/bin/dbstart 和 \$ORACLE_HOME/bin/dbshut。

(1) 配置 oratab 文件。在 oracle 用户终端中执行 vi /etc/oratab，将其中各例程是否自动启动的标志改成 Y，表示都允许自动启动，即将 oratab 文件内容修改为：

```

orcl:/home/oracle/product/10.2.0/db_1:Y
oradb:/home/oracle/product/10.2.0/db_1:Y

```

(2) 修改 dbstart 文件。在 \$ORACLE_HOME/bin 目录（本例 \$ORACLE_HOME 为 /home/oracle/product/10.2.0/db_1）中找到 dbstart 文件，将 ORACLE_HOME_LISTNER 修改为监听 ORACLE_HOME 中所有例程，即将

```

ORACLE_HOME_LISTNER=/ade/vikrkuma_new/oracle

```

修改为

```

ORACLE_HOME_LISTNER=$ORACLE_HOME

```

(3) 修改 dbshut 文件。dbshut 文件默认只是关闭 oratab 文件中设置为 Y 的数据库，而不关闭监听器。为使其也能关闭监听器进程，需要在该文件末尾加入下面一行：

```
lsnrctl stop
```

(4) 建立系统启动、关闭时自动执行的 shell 脚本文件。以 root 登录系统，在 /etc/init.d 目录中创建一个 shell 脚本文件，命名为 oracle。执行 oracle 文件时，如果传入 start 参数则调用 dbstart 文件启动数据库及监听器，如果传入 stop 参数则调用 dbshut 文件关闭数据库及监听器。用 vi 打开该文件的命令如下：

```
[root@localhost ~]# vi /etc/init.d/oracle
```

该文件内容如下：

```
case "$1" in
start)
echo -n "Start oracle....."
su - oracle -c dbstart
echo -n "Starting oracle completed."
;;
stop)
echo -n "Shutdown oracle..... "
su - oracle -c dbshut
echo -n "Oracle stopped."
;;
restart)
echo -n "Resatrting oracle..... "
su - oracle -c dbshut
su - oracle -c dbstart
echo -n "Restarting ok."
;;
*)
echo "Usage: 'basename $0' start|stop"
exit 1
esac
```

然后设置文件 oracle 的可执行权限，命令如下：

```
[root@localhost ~]# chmod +x /etc/init.d/oracle
```

(5) 修改不同级别用户的启动软链接。为了在运行级别 3、5 能自动执行 /etc/init.d/oracle，以启动例程和监听程序，可以在 /etc/rc.d/rc3.d 和 /etc/rc.d/rc5.d 目录建立指向 /etc/init.d/oracle 的软链接 S99oracle。在 root 用户终端中执行如下命令：

```
[root@localhost ~]# ln -s /etc/rc.d/init.d/oracle /etc/rc.d/rc3.d/S99oracle
```

```
[root@localhost ~]# ln -s /etc/rc.d/init.d/oracle /etc/rc.d/rc5.d/S99oracle
```

(6) 多例程自动启动测试。重新启动计算机后，以 oracle 用户登录，等待例程启动完毕，在 oracle 用户终端中输入如下命令分别以 SCOTT 用户连接 orcl 和 oradb 两个例程进行测试，均应该能够正常连接到数据库：

```
[oracle@localhost ~]$ sqlplus scott/tiger@orcl
```

```
[oracle@localhost ~]$ sqlplus scott/tiger@oradb
```

10. Windows 环境下手工创建数据库

在 Windows 环境中手工创建数据库，过程基本与 Linux 相同，差别在于自动启动配置（Windows 为控制面板中的服务）。

在 Windows 中创建 Oracle 数据库服务例程的命令为 oradim.exe，该命令在 Windows 命令行上执行，当只输入命令 oradim 时可显示该命令的帮助信息。使用 oradim 可在 Windows 中创

建、删除 Oracle 数据库的例程服务，可设定例程服务的启动和关闭方式，还可通过它修改例程以前的设定值。使用 `oradim` 创建数据库例程服务的语法如下：

```
oradim -NEW -SID 数据库例程名 -INTPWD 口令字
        -STARTMODE auto
        -PFILE 数据库参数文件名及路径
```

其中 `INTPWD` 指定 `SYSDBA` 的口令。

一般情况下，应在创建口令文件之后、执行 `CREATE DATABASE` 命令之前运行 `oradim` 创建例程的服务。

在 Windows 环境中，建议使用 `DBCA` 来快捷、高效地完成数据库创建的各项工

4.1.5 数据库创建失败后的处理

如果因为某种原因使数据库创建失败，在准备重新创建数据库之前，应关闭相应例程并删除所有用 `CREATE DATABASE` 语句创建的文件。在纠正了导致数据库创建失败的错误之后再次运行相关的 SQL 脚本。

根据出错的情况不同，需要手工清除的内容也不同，参阅附录执行相应删除操作。

4.2 管理表空间

4.2.1 表空间管理概述

表空间是 Oracle 数据库内部数据的逻辑组织结构，是数据库的逻辑划分，对应于磁盘上的一个或多个物理数据文件。数据库中至少要包含一个 `SYSTEM` 表空间(含 `SYSAUX` 表空间)，还应该包含数据表空间、索引表空间、临时表空间和撤销 (`UNDO`) 表空间等多个表空间。

从用户角度看到的表、序列、索引、视图等对象，都存储在表空间这个逻辑结构里面，按照后面讲到的应用原则将各用户对象组织和管理起来。而表空间中的对象最终是在磁盘上存储的，因而一个表空间对应 1 个或多个物理数据文件。所以表空间将用户视图、数据库的逻辑结构和物理结构有机的结合起来。

深入理解表空间的类型、管理方式、特性和状态，掌握表空间与数据文件之间的关系，合理为数据文件安排磁盘空间，对于设计与实现一个 Oracle 数据库应用系统是至关重要的。

1. 表空间的特性

表空间具有如下特性：

- (1) 一个数据库可以有多个表空间，可以在数据库中创建和删除表空间。
- (2) 一个表空间只能属于一个数据库。
- (3) 一个表空间至少要有有一个数据文件。
- (4) 一个表空间的大小由其数据文件大小决定，等于各数据文件大小之和。
- (5) 除 `SYSTEM` 外的表空间可以被联机或脱机。
- (6) 一个数据文件只能属于一个表空间。
- (7) 方案对象可以跨表空间的数据文件存储，但不能跨表空间存储。当一个数据文件存储不下某个方案对象时，可以存储到该表空间下另一个数据文件中。
- (8) 可以为用户指定默认数据表空间，但该用户的方案对象可以存储到不同表空间中。
- (9) 应合理规划表空间大小，并指定用户在各表空间上的空间配额。

2. 表空间应用原则

应在一个数据库中创建多个表空间，根据需要存放不同的数据。数据库管理员在创建表空间时，应当仔细地考虑应用系统数据的特性，并据此合理规划表空间和磁盘文件。对表空间的使用应遵循分散（Separate）存储原则，具体包括如下情形：

(1) 在系统性能要求较高的应用环境中，创建表空间时使用裸设备存储数据文件。

(2) 将表、索引分开存放在不同的表空间中。分开存放表和索引，可避免磁盘 I/O 冲突，提高系统对事务的并行处理能力。

(3) 将访问频度高的表、索引分开存放在不同的表空间，并将这些表空间所使用的数据文件存储到不同的物理磁盘上。

(4) 对于数据量特别大，并发访问频繁的表、索引应考虑单独存放在一个表空间中，避免该表、索引和其他表、索引的磁盘访问竞争。为了进一步提高这些表、索引的读写速度，应考虑将表、索引进行分区存储到不同表空间中。

(5) 将日志和数据放置在不同的磁盘上。由于日志信息被连续、顺序地写入磁盘，将和数据的随机读取、更新方式相冲突。分开日志和数据的存放，将避免磁头的来回移动，提高 I/O 处理速度。

3. Oracle 表空间管理方法

在 Oracle 10g 之前有两种表空间管理方法，即本地管理和数据字典管理。本地管理方法是在表空间的数据文件中开辟一片磁盘空间，以位图的形式表示表空间中每一个区间的使用情况，这种管理方式的特点是空间分配与回收速度快、效率高。在本地管理方式下，创建表空间时不能指定分区的默认存储参数，只能将它们设置为自动分配或统一分配，并且在表空间创建后不能改变。数据字典管理方法将区间的使用信息存储在数据字典中，但可以指定分区分配参数。由于在进行存储分配和回收时需要不断地访问相关数据字典表，因此效率较低。两种表空间管理方法的差别参见表 4-2。

表 4-2 表空间管理方法的比较

管理方法	区间信息存储位置	区间大小	存取效率	空间管理和使用的灵活性	使用状态信息修改是否产生回退信息
本地管理	每个数据文件中	相同	高	低	不
数据字典管理	统一的数据字典中	可不同	低	高	是

在 Oracle 10g 中，基于性能的考虑，不再支持数据字典管理的表空间。

4. 表空间大小的估算

需要通过估算表、索引等方案对象的大小及数据块空间分配比例来估算表空间的大小。要确定一个表的大小，应收集如下信息：

(1) 表中每行的最大字节数。

(2) 该表最初要包括多少记录行。

(3) 每月、季、年表中记录的增长率。

(4) 表上数据操作的类型。

(5) 确定是哪种操作执行比较多，插入、删除将导致空间占用发生比较大的变化，修改和查询不会引起空间占用大的变化（修改 LOB 列除外）。

可以按照如下公式计算表的大小：

$$\text{表占用空间大小} = \text{最大行长} \times \text{初始行数} \times (1 + \text{PCTFREE}/100) \times \text{记录复合增长率}$$

其中 PCTFREE 是数据块保留用于更新的百分比，记录复合增长率指的是在系统的估计使用期内记录增长率的乘积。

在估算各表、索引等方案对象的大小后，将它们加起来就是表空间应具备的大小，真正创建的表空间一般应比这个数要再大一点。

4.2.2 表空间的创建与删除

1. 建立表空间的语法

使用 CREATE TABLESPACE 命令建立表空间。一般情况下，建立表空间由特权用户或 DBA 完成，如果要以其他用户身份建立表空间，则用户必须具有 CREATE TABLESPACE 系统特权。当执行 CREATE DATABASE 命令建立数据库时，Oracle 会自动建立 SYSTEM 和 SYSAUX 系统表空间，另外还会建立默认的临时表空间 TEMP 和撤销表空间 UNDOTBS1。

同时为了加强管理、提高数据库的性能，应该将不同类型的段部署到不同的表空间中。例如，应该建立专门存放表段的数据表空间、存放索引段的索引表空间、存放 UNDO 段的 UNDO 表空间等。

在 Oracle 10g 中，增加了大文件（bigfile）表空间，该表空间只能包含一个数据文件（或临时文件），但其数据块个数可以达到 4G 个，如果 1 个数据块 8KB，则该表空间可以存放 32TB 数据。采用传统的 CREATE TABLESPACE 创建的表空间均相应地称为小文件（smallfile）表空间，系统表空间总被创建为小文件表空间。

创建表空间的 CREATE TABLESPACE 命令语法如下：

```
CREATE [SMALLFILE | BIGFILE] [PERMANENT | TEMPORARY | UNDO] TABLESPACE tablespace
[DATAFILE|TEMPFILE] datafile_tempfile_spec [, datafile_tempfile_spec ... ]
[ { MINIMUM EXTENT integer [ K | M ]
  | BLOCKSIZE integer [ K | M ]
  | EXTENT MANAGEMENT LOCAL [AUTOALLOCATE | UNIFORM [SIZE integer [K|M]]]
  | SEGMENT SPACE MANAGEMENT { MANUAL | AUTO }
  | { ONLINE | OFFLINE }
  | { LOGGING | NOLOGGING }
  }
];
```

从上面语法可以看出，可以创建小文件表空间、大文件表空间、永久表空间、临时表空间和撤销表空间等几种类型的表空间，默认建立的是小文件永久数据表空间。在语法中的 tablespace 为要建立的表空间名，PERMANENT 子句声明本表空间用于存储持久数据，即为永久表空间，而 TEMPORARY 则声明表空间用于存放临时数据。Oracle 建议建立单独的临时表空间专门用于处理临时数据。datafile_tempfile_spec 为表空间的数据文件（或临时文件）定义，它的语法如下：

```
'path_filename' [SIZE integer [K|M]] [REUSE]
[AUTOEXTEND {OFF|ON [NEXT integer [K|M]]
              [MAXSIZE {UNLIMITED | integer [K|M]}] } ]
```

其中的 path_filename 为包含路径的数据文件名。在路径中可用 ORACLE_HOME 和 ORACLE_BASE 这样的 Oracle 环境变量。SIZE 指定数据文件的大小，可以 KB 和 MB 为单位。

{LOGGING | NOLOGGING} 子句用于声明是否对表、索引、实体化视图和分区等进行操作日志记载，默认值是 LOGGING。除非必要，应选择 NOLOGGING 以免影响系统性能。在对对象级的 logging_clause 子句值将覆盖表在空间一级指定的此参数值。

SEGMENT SPACE MANAGEMENT 子句声明表空间中段空间的管理方法，段空间一般采

用自动 (AUTO) 管理即可。

使用 BLOCKSIZE 可以定义使用非标准大小块的表空间。

2. 建立数据表空间

如果不指定 TEMPORARY 或 UNDO, 使用 CREATE TABLESPACE 命令默认建立的表空间就是数据表空间。应为每个用户建立专用的数据表空间存储用户数据, 一个用户可以使用多个数据表空间。

例 4.1 建立名称为 data_ts1 的数据表空间, 大小为 50M, 区间统一为 128KB 大小。

```
SQL> CONN system/systempwd@orcl
SQL> DROP TABLESPACE data_ts1 INCLUDING CONTENTS;
SQL> CREATE TABLESPACE data_ts1
2     DATAFILE '%ORACLE_HOME%\database\data_ts1.dbf' SIZE 50M REUSE
3     UNIFORM SIZE 128K;
```

3. 为索引建立表空间

在小型数据库中, 表和索引可以部署在同一个表空间中。但是在大型数据库环境中, 为了提高数据库的性能, 需要建立专门的索引表空间, 将表和索引部署到不同的表空间中。Oracle 没有单独的索引表空间类型, 索引使用数据表空间存储, 在定义索引时指定将索引存储到该表空间中。

4. 建立临时表空间

临时表空间 (Temporary tablespace) 是 Oracle 用于存储临时表的空间。查询时用到 Order by、Group by、Union 等操作将占用较大的临时存储区域, 因此临时表空间可设置大一些。

例 4.2 建立名称为 temp_ts1 的临时表空间, 该表空间使用 temp_ts1.dbf 文件存放临时数据。

```
SQL> CONN system/systempwd@orcl
SQL> DROP TABLESPACE temp_ts1 INCLUDING CONTENTS;
SQL> CREATE TEMPORARY TABLESPACE temp_ts1
2     TEMPFILE '%ORACLE_HOME%\database\temp_ts1.dbf' SIZE 20M REUSE
3     UNIFORM SIZE 128K;
```

需要注意的是, 在临时表空间上不能建立永久对象 (表、索引和簇), 而且当执行 CREATE USER 命令时, 如果不指定 TEMPORARY TABLESPACE 选项, 用户的排序操作就会使用默认的临时表空间 TEMP。可使用带 TEMPORARY TABLESPACE 子句的 ALTER USER 命令指定用户的临时表空间。

例 4.3 创建 10 号部门经理用户 EMP_MGR10, 指定该用户的数据表空间为 data_ts1, 临时表空间为 temp_ts1。授权该用户可以查看 SCOTT 用户下雇员表中的记录。关于如何控制该用户只能查看到 10 号部门记录的问题, 在 9.11 节“虚拟专用数据库”中讨论。

```
SQL> CONN system/systempwd@orcl
SQL> DROP USER emp_mgr10 CASCADE;
SQL> CREATE USER emp_mgr10 IDENTIFIED BY emp_mgr10pwd
2     DEFAULT TABLESPACE data_ts1 TEMPORARY TABLESPACE temp_ts1;
SQL> GRANT CONNECT TO emp_mgr10;
SQL> CONN scott/tiger@orcl
SQL> GRANT SELECT ON emp TO emp_mgr10;
```

5. 建立撤销表空间

为简化撤销 (UNDO) 数据的管理, 建议使用撤销表空间, Oracle 可以自动控制 UNDO 段的扩展和收缩, 不需要 DBA 的干预。注意, 在撤销表空间上同样不能建立永久对象。

如果创建了撤销表空间并指定系统的运行方式为自动撤销管理模式, Oracle 将使用撤销

表空间而不是使用回退段来存放更新操作的回退信息。要使用撤销表空间，需设置初始化参数 UNDO_MANAGEMENT 为 AUTO，并设置 UNDO_TABLESPACE 为指定的撤销表空间名。

例 4.4 创建和应用撤销表空间。

```
SQL> CONN system/systempwd@orcl
SQL> DROP TABLESPACE undo_ts1 INCLUDING CONTENTS;
SQL> CREATE UNDO TABLESPACE undo_ts1
2 DATAFILE '%ORACLE_HOME%\database\undo_ts1.dbf' SIZE 50M REUSE;
```

在数据库打开状态下只能使用一个 UNDO 表空间，为了使用新建立的 undo_ts1 表空间，必须切换到该表空间，执行如下语句：

```
SQL> ALTER SYSTEM SET undo_tablespace = undo_ts1;
```

执行完本语句后，究竟系统是否使用 undo_ts1 来管理撤销数据，还取决于初始化参数 UNDO_MANAGEMENT 的设置，只有设置为 AUTO 才会真正使用它来管理撤销数据。

6. 建立大文件表空间

大文件表空间是为超大型数据库设计的。对一个超大型数据库来说，如果采用传统的小文件表空间，那么系统中将包含数百个以至上千个数据文件，更新这些数据文件的头部信息（如写入检查点信息等）将会导致很大的系统开销。采用大文件表空间将大大减少系统中数据文件的个数，从而减少系统开销。

例 4.5 创建大文件表空间，并指定为 SCOTT 用户的默认数据表空间。

```
SQL> CONN system/systempwd@orcl
SQL> DROP TABLESPACE bigfile_ts1 INCLUDING CONTENTS;
SQL> CREATE BIGFILE TABLESPACE bigfile_ts1
2 DATAFILE '%ORACLE_HOME%\database\bigfile_ts1.dbf' SIZE 50M REUSE;
SQL> ALTER USER scott DEFAULT TABLESPACE bigfile_ts1;
```

需要注意的是，大文件表空间的段空间管理不能为手工（MANUAL），只能为自动（AUTO）。

4.2.3 表空间维护

1. 表空间信息查询

系统数据字典 DBA_TABLESPACES、DBA_SEGMENTS、DBA_FREE_SPACE、DBA_EXTENTS 中分别记载了表空间、段、自由表空间、扩展段的使用情况，在需要时可执行对这些数据字典的查询语句来了解系统中表空间的使用情况。执行如下语句可查询系统中的表空间：

```
SQL> SET PAGESIZE 30
SQL> COL tablespace_name FORMAT A12
SQL> SELECT tablespace_name,block_size,segment_space_management,
2 status,contents,allocation_type,bigfile
3 FROM dba_tablespaces;
```

TABLESPACE_N	BLOCK_SIZE	SEGMENT	STATUS	CONTENTS	ALLOCATION	BIG
SYSTEM	8192	MANUAL	ONLINE	PERMANENT	SYSTEM	NO
UNDOTBS1	8192	MANUAL	ONLINE	UNDO	SYSTEM	NO
SYSAUX	8192	AUTO	ONLINE	PERMANENT	SYSTEM	NO
TEMP	8192	MANUAL	ONLINE	TEMPORARY	UNIFORM	NO
USERS	8192	AUTO	ONLINE	PERMANENT	SYSTEM	NO
EXAMPLE	8192	AUTO	ONLINE	PERMANENT	SYSTEM	NO
TS1	8192	AUTO	ONLINE	PERMANENT	SYSTEM	NO
TS2	8192	AUTO	ONLINE	PERMANENT	SYSTEM	NO
TS3	8192	AUTO	ONLINE	PERMANENT	SYSTEM	NO
TS4	8192	AUTO	ONLINE	PERMANENT	SYSTEM	NO
MY_INDEX	8192	AUTO	ONLINE	PERMANENT	SYSTEM	NO
DATA_TS1	8192	AUTO	ONLINE	PERMANENT	UNIFORM	NO
TEMP_TS1	8192	MANUAL	ONLINE	TEMPORARY	UNIFORM	NO
UNDO_TS1	8192	MANUAL	ONLINE	UNDO	SYSTEM	NO
BIGFILE_TS1	8192	AUTO	ONLINE	PERMANENT	SYSTEM	YES

2. 存储空间调节

有两种改变表空间大小的方法，一种方法是改变数据文件的大小（可手工改变数据文件大小或让数据文件自动扩展），另一种方法是增加表空间的数据文件。注意，增加表空间数据文件使用的是 ALTER TABLESPACE 命令，改变数据文件大小使用的是 ALTER DATABASE 命令。

为表空间增加数据文件的 ALTER TABLESPACE 语句的语法如下：

```
ALTER TABLESPACE tablespace
  ADD DATAFILE filespec [autoextend_clause]
  [,filespec [autoextend_clause] ] ... ;
```

改变数据文件大小的 ALTER DATABASE 语句的语法如下：

```
ALTER DATABASE [database]
  DATAFILE 'filename' [, 'filename' ] ...
  RESIZE integer [K | M ] ;
```

允许数据文件自动扩展的 ALTER DATABASE 语句语法如下：

```
ALTER DATABASE DATAFILE filespec
  AUTOEXTEND {OFF|ON [NEXT n [K|M]] [MAXSIZE {UNLIMITED |n [ K | M ]}} ;
```

例 4.6 调整数据表空间 data_ts1 的大小。

```
SQL> CONN system/systempwd@orcl
--为表空间 data_ts1 增加一数据文件，大小为 1M
SQL> ALTER TABLESPACE data_ts1
  2   ADD DATAFILE '%ORACLE_HOME%\database\data_ts2.dbf' SIZE 1M;
--重置该数据文件大小为 2M
SQL> ALTER DATABASE
  2   DATAFILE '%ORACLE_HOME%\database\data_ts2.dbf' RESIZE 2M ;
--修改该数据库，允许该文件自动扩展，每次扩展 1M，文件最大扩展到 10M
SQL> ALTER DATABASE DATAFILE '%ORACLE_HOME%\database\data_ts2.dbf'
  2   AUTOEXTEND ON NEXT 1M MAXSIZE 10M;
```

3. 表空间状态维护

当要进行数据库维护时，可能需要将某个表空间暂时脱机或设置为只读。使用 ALTER TABLESPACE 命令修改表空间状态，语法如下：

```
ALTER TABLESPACE tablespace { ONLINE
  | OFFLINE [ NORMAL |TEMPORARY | FOR RECOVER ]
  | READ [ ONLY | WRITE ]
  };
```

4. 数据文件管理

当发现系统性能下降时，DBA 应当检查系统中数据文件的使用情况，可以执行如下语句查看数据库中数据文件的分布情况，看看它们是否存在 I/O 竞争：

```
COL file_name FORMAT A45
SELECT file_id, file_name, tablespace_name
  FROM dba_data_files
  ORDER BY file_id;
```

如果发现文件比较集中地存在于某个（些）磁盘上，可以考虑将它们分开部署到不同磁盘。按如下步骤操作即可：

(1) 执行 ALTER TABLESPACE tablespace OFFLINE 命令将对应的表空间脱机。

(2) 执行 HOST COPY source_disk_datafile dest_disk_datafile 命令将数据文件复制到分离的目标盘，或在操作系统上复制该文件到目标位置。

(3) 执行 ALTER TABLESPACE tablespace RENAME source_disk_datafile TO dest_disk_datafile 重命名表空间数据文件，将其数据文件定义为新位置的文件。

(4) 执行 ALTER TABLESPACE tablespace ONLINE 命令将对应的表空间联机。

5. 删除表空间

当表空间因为损坏无法恢复，或者当表空间不再需要时，可以删除该表空间。除了 SYSTEM 表空间不允许删除外，其他表空间都可以删除。删除表空间是由特权用户或 DBA 来执行的，如果要以其他用户身份删除表空间，则该用户必须具有 DROP TABLESPACE 系统特权。删除表空间同时可以级联删除它拥有的所有数据文件，如果未指定删除数据文件选项 (AND DATAFILES)，删除不用的数据文件的工作需要 DBA 在操作系统下手工完成。

删除各种表空间的语法都是一样的，删除表空间的 DROP TABLESPACE 命令语法如下：

```
DROP TABLESPACE tablespace
  [INCLUDING CONTENTS [AND DATAFILES]
  [CASCADE CONSTRAINTS]];
```

其中 tablespace 是要删除的表空间名字。

INCLUDING CONTENTS 指定删除表空间的同时删除表空间内容，如果不指定 INCLUDING CONTENTS 选项，而该表空间非空时，则提示错误。

CASCADE CONSTRAINTS 指定删除表空间时要删除其他表空间上表的参照完整性约束，这主要是指包含主键及唯一索引等表空间。如果存在完整性约束，而没有指定该参数，则 Oracle 会返回一个错误，并且不会删除该表空间。

例 4.7 删除 temp_ts1 表空间。

```
SQL> CONN system/systempwd@orcl
SQL> DROP TABLESPACE temp_ts1
 2   INCLUDING CONTENTS AND DATAFILES
 3   CASCADE CONSTRAINTS;
```

4.2.4 回退段的创建与删除

UNDO 段也称为回退段，用于存放事务所修改数据的旧值。Oracle 使用它来管理回退数据，实现读一致性以及数据库数据恢复等操作。例如，当执行 UPDATE emp SET sal=1000 WHERE empno=7934 语句时，如果 7934 原有的工资列值 sal 是 800，那么 800 将会被存放到 UNDO 段中，而新数据 1000 将会被存放到表段 EMP 中。

默认的回退段是 SYSTEM，在数据字典 DBA_ROLLBACK_SEGS 中能够查询到当前使用的回退段。当系统工作于自动撤销管理模式时，用户不能创建自己的回退段。需先修改参数 UNDO_MANAGEMENT 值为 MANUAL 才能使用用户自己创建的回退段。

1. 回退段的创建

要创建回退段，用户必须具有 CREATE ROLLBACK SEGMENT 系统权限。要包含新回退段的表空间必须是联机的。回退段的创建通常作为数据库创建脚本或进程的一部分，但用户可以在这之后添加更多的回退段。

使用 CREATE ROLLBACK SEGMENT 语句创建回退段，语法如下：

```
CREATE [PUBLIC] ROLLBACK SEGMENT rollback_segment
  [TABLESPACE tablespace][STORAGE storage];
```

其中 rollback_segment 为要创建的回退段名，tablespace 为回退段所在表空间名。

例 4.8 在撤销表空间 undo_ts1 中建立一回退段。

```
SQL> CONN system/systempwd@orcl
SQL> DROP ROLLBACK SEGMENT undo_ts1_rbs1;
SQL> CREATE ROLLBACK SEGMENT undo_ts1_rbs1
 2   TABLESPACE undo_ts1 STORAGE (initial 5K
```

```

3          next 5K
4          optimal 500K
5          minextents 15
6          maxextents 100);

```

新建立的回退段初始时是脱机的，必须发布一条 ALTER ROLLBACK SEGMENT 语句使其联机并且使它们能够为一个例程的事务所使用。使用带有 ONLINE 选项的 ALTER ROLLBACK SEGMENT 语句使回退段联机。如下语句实现使 undo_ts1_rbs1 回退段联机：

```
SQL> ALTER ROLLBACK SEGMENT undo_ts1_rbs1 ONLINE;
```

使用带有 OFFLINE 选项的 ALTER ROLLBACK SEGMENT 语句使回退段脱机。如下语句实现使 undo_ts1_rbs1 回退段脱机：

```
SQL> ALTER ROLLBACK SEGMENT undo_ts1_rbs1 OFFLINE;
```

可以改变回退段的存储参数，使用 ALTER ROLLBACK SEGMENT 语句修改回退段存储信息，语法如下：

```
ALTER [PUBLIC] ROLLBACK SEGMENT rollback_segment [STORAGE storage];
```

2. 删除回退段

当一个回退段的盘区在磁盘上碎片太多，或者该段需要在一个不同的表空间中重新定位时，可以删除回退段。在删除回退段之前，必须确定该回退段的状态是 OFFLINE，否则删除时会出现错误信息。用户必须拥有 DROP ROLLBACK SEGMENT 系统特权才能删除回退段。要删除一个回退段，使用 DROP ROLLBACK SEGMENT 语句，语法如下：

```
DROP ROLLBACK SEGMENT rollback_segment;
```

如下语句删除回退段 undo_ts1_rbs1：

```
SQL> DROP ROLLBACK SEGMENT undo_ts1_rbs1;
```



安装 Oracle 数据库管理系统软件和创建数据库是一项重要的基础工作，可以手工创建数据库，也可以使用 DBCA 创建数据库。在 Windows 和 Linux 环境下安装 Oracle 和创建数据库的过程有较大的差异，但是只要抓住一条主线，就是在准备环节配置环境变量、使用 CREATE DATABASE 命令创建数据库、执行脚本安装其他选项，创建数据库就相对容易理解了。表空间是所有方案对象的逻辑组织，合理设计和组织表空间能显著改善系统的性能。在系统中应根据数据的特性和使用要求，建立和应用不同的表空间。对于撤销数据，Oracle 建议采用撤销表空间进行自动管理。

综合应用与习题四

质检子系统（参见第3章）是给 HSS 公司开发的 ERP（企业资源计划）系统的一部分，在该 ERP 系统中还存在物料基本信息表（含物料的价格）、产品结构表、工艺路线表、加工工序表（含工时定额）、加工设备表、操作工岗位工资表（工资定额）、供应商等，这些表分属不同的子系统。其中物料基本信息表属于基础数据管理子系统，工资定额表属于成本子系统，供应商属于采购子系统，其他表属于技术子系统。在质检相关表中出现的公司内加工部门和人员信息属于人力资源子系统。

废品损失由 3 部分构成，分别是材料费、人工费、制造费，说明如下：

废品材料费的计算：查找产品结构表，按照产品结构提取出子件物料，从物料基本信息

表中查找该物料各子件的成本（为该子件综合了材料、人工、制造费用的总成本）。

废品工时费的计算：查找加工工序表，取出该物料的加工工序及各工序需要的加工时间，再取工资定额，计算该物料已经发生工序的工人的工资费用。

废品制造费的计算：查找产品加工工艺路线表，取出各工艺阶段所用的加工设备及其功率定额，计算该物料已经发生工序的燃料动力费。

现在给该公司的 ERP 系统创建一个专用数据库 HSS_ERP，在该数据库中建立以上各子系统的方案用户，每个方案对象保存在各自的表空间中，查询时也使用各自的临时表空间。系统采用自动管理撤销数据的工作模式。

根据以上信息，重新设计针对质检子系统的安全管理方案。在第 3 章未给出的表及结构等请自拟，可简化，能说明问题即可。

继续完成以下习题：

1. 简述手工创建数据库的过程。
2. 为什么要给用户指定默认表空间和临时表空间？
3. 创建表空间要定义它的增长策略，请解释各增长策略。
4. SYSTEM 表空间是执行哪个命令建立的？
5. 在哪些表空间上不能建立用户数据表？
6. 使用 DROP TABLESPACE 命令是否可以删除数据文件，为什么？
7. 简要说明管理表空间的原则。
8. 写出修改 cemerp 数据库中 SCOTT 用户在 USERS 表空间中空间限额为 10M 的相关命令。
9. 为赛德克公司创建一个名为 surtec 的 Oracle 数据库用户，密码为 surtecpwd。该用户需要一个 10G 的数据表空间，200M 的临时表空间。该用户不能拥有 SYSDBA 权限，但可以创建自己的数据库表等对象。按照相关政策规定，社保局（Oracle 用户名 SIB）可以查看该用户下 emp 表中的数据，并可修改该用户的养老保险缴费记录表（endowment_insurance）中缴费完成标志字段（flag）。写出能完成上述任务的 SQL 语句。
10. 写出完成如下任务的相关 SQL 语句和 SQL*Plus 命令，要求按照执行顺序编写。
 - （1）创建一个名为 test01 的 Oracle 数据库用户，初始密码为 test1234。
 - （2）以该用户登录并创建一个数据表空间 test_ts，数据文件为 test_ts.dbf，大小为 5MB。各参数设置如下：初始区间大小为 50k，扩展区间大小为 50k，可分配区间数为 2~50 个。
 - （3）为表空间 test_ts 增加数据文件 test_ts2.dbf，大小 10MB。
 - （4）重置数据文件 test_ts2.dbf 大小为 15MB。
11. 现有一个名为 pur_ts 的表空间，系统提示其所占磁盘空间已满。按如下要求完成扩充其空间任务，分别写出对应的 SQL 语句。
 - （1）为增加数据文件 pur_data.dbf，大小 800MB。
 - （2）重置数据文件 pur_data.dbf 的大小为 2GB。
 - （3）允许数据文件 pur_data.dbf 的区间自动扩展，每次扩展 30MB。