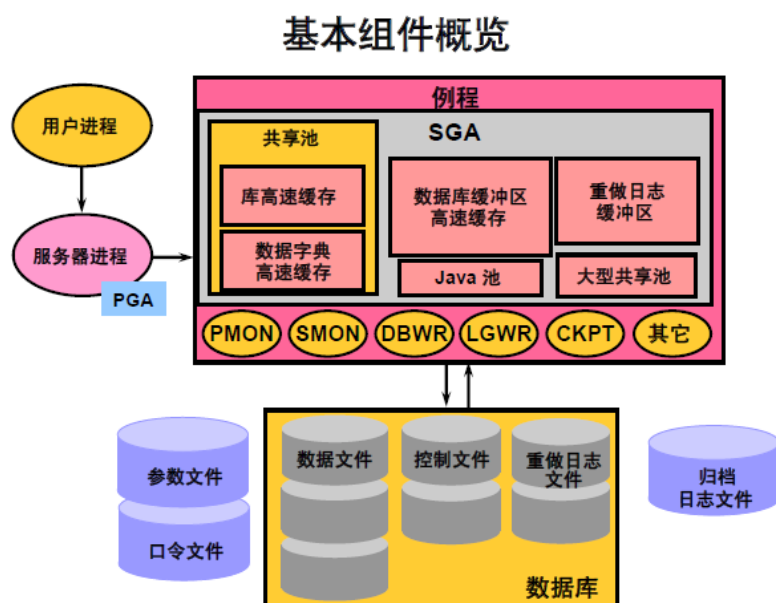


目录

第一章 Oracle 体系结构组件.....	1
1. 组成结构(P27).....	1
2. Oracle 实例(P30).....	2
3. 内存结构(P35).....	2
4. 进程结构(P52).....	6
5. 处理 SQL 语句(P66).....	9
第三章 管理 Oracle 实例.....	10
1. 启动数据库(P113).....	10
2. 以只读模式打开数据库(P123).....	13
3. 关闭数据库(P125).....	14
第四章 创建数据库.....	17
1. 创建的前提条件(P146).....	17
2. 创建数据库(P150).....	17
3. 手动创建数据库(P157).....	18
第五章 使用数据字典和动态性能视图.....	21
1. 数据字典内容(P177).....	21
2. 动态性能表(P182).....	22
第六章 维护控制文件.....	23
1. 控制文件的内容(P195).....	23
第八章 管理表空间和数据文件.....	25
1. 表空间类型(P240).....	25
2. 表空间的空间管理(P245).....	26
3. 缺省临时表空间(P253).....	27
4. 缺省临时表空间的限制(P257).....	28
第十章 管理还原数据.....	29
1. 还原段用途(P323).....	29
附 1 建表 SQL 示例.....	30
附 2 增删改查 SQL 示例.....	31
1. 增加 (INSERT)	31
2. 删除 (DELETE)	31
3. 修改 (UPDATE)	32
4. 查询 (SELECT)	32

第一章 Oracle 体系结构组件

1. 组成结构(P27)



基本组件概览

Oracle 体系结构包括很多基本组件，本课将详细介绍这些组件。

- **Oracle 服务器**：Oracle 服务器中包含多种文件结构、进程结构和内存结构；但是，处理 SQL 语句时，并非所有这些结构都会用到。某些结构用于提高数据库的性能，确保该数据库在遇到软件或硬件错误时可以恢复，或者执行维护该数据库所需的其它任务。Oracle 服务器包括一个 Oracle 实例和一个 Oracle 数据库。

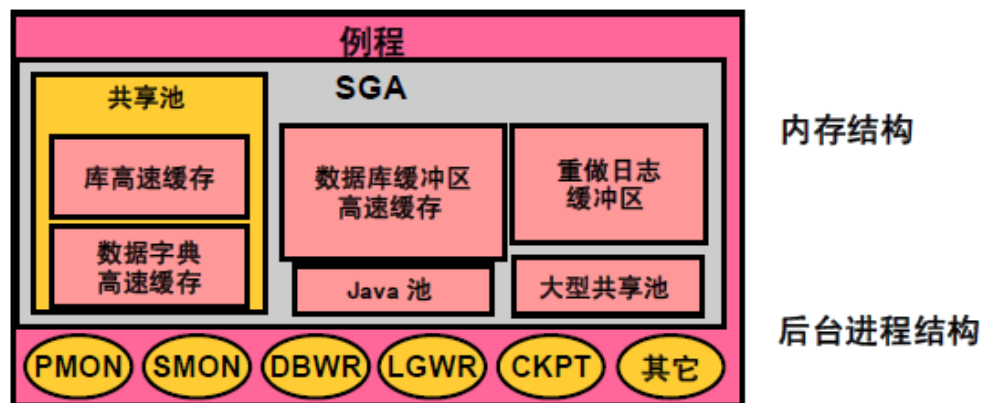
- **Oracle 实例**：Oracle 实例是后台进程和内存结构的组合。只有启动实例后，才能访问数据库中的数据。每次启动实例时，会分配系统全局区(SGA)并启动 Oracle 后台进程。后台进程代表调用进程执行各种功能。它们把为每个用户运行的多个 Oracle 程序所处理的功能统一起来。后台进程执行输入/输出(I/O)，并监视其它 Oracle 进程来提高并行性，从而使性能和可靠性更加优越。

- **Oracle 数据库**：Oracle 数据库包含操作系统文件（也称为数据库文件），这些文件为数据库信息提供了实际的物理存储。数据库文件用于确保数据一致性并能在实例失败时得以恢复。

2. Oracle 实例(P30)

Oracle 实例：

- 是一种访问 Oracle 数据库的方式
- 始终打开一个，并且只打开一个数据库
- 由内存结构和后台进程结构组成



Oracle 实例由系统全局区(SGA) 内存结构和用于管理数据库的后台进程组成。实例是通过使用特定于每个操作系统的方法来标识的。实例一次只能打开和使用一个数据库。

3. 内存结构(P35)

Oracle 的内存结构由两个内存区组成，分别是：

- 系统全局区(SGA)：在实例启动时分配，是 Oracle 实例的基本组件
- 程序全局区(PGA)：在服务器进程启动时分配

系统全局区(SGA)

- SGA 包括以下几种内存结构：
 - 共享池
 - 数据库缓冲区高速缓存

- 重做日志缓冲区
- 其它结构（例如锁定和栓锁管理以及统计数据）
- 大型共享池
- Java 池

SGA 也称作共享全局区，用于存储数据库进程共享的数据库信息。它包含有关 Oracle 服务器的数据和控制信息，在 Oracle 服务器所在计算机的虚拟内存中分配。

要查看 SGA 内存的分配情况，可以使用下面的语句：SHOW SGA

SGA 是动态的

- 大小由 SGA_MAX_SIZE 参数指定
- 由 SGA 组件以粒组为单位进行分配和跟踪
- 连续的虚拟内存分配
- 粒组大小由估算的 SGA_MAX_SIZE 总计大小确定

共享池

- 用于存储：
 - 最近执行的 SQL 语句
 - 最近使用的数据定义
- 它包括以下两个与性能相关的关键内存结构：
 - 库高速缓存
 - 数据字典高速缓存
- 其大小由 SHARED_POOL_SIZE 参数确定

数据库缓冲区高速缓存

- 存储已从数据文件中检索到的数据块的副本
- 能够大幅提高获取和更新数据时的性能

- 通过 LRU 算法管理
- 主块的大小由 DB_BLOCK_SIZE 确定
- 由独立的子高速缓存组成：
 - DB_CACHE_SIZE
 - DB_KEEP_CACHE_SIZE
 - DB_RECYCLE_CACHE_SIZE
- 大小可以进行动态调整
- 设置 DB_CACHE_ADVICE 可收集用于预测不同高速缓存大小行为的统计信息
- 统计信息由 V\$DB_CACHE_ADVICE 显示

重做日志缓冲区

- 记录对数据库数据块所做的全部更改
- 主要用于恢复
- 其中记录的更改称作重做条目
- 重做条目包含用于重新构造或重做更改的信息
- 大小由 LOG_BUFFER 定义

大型共享池

- SGA 中的可选内存区
- 分担了共享池的一部分工作
- 用于：
 - 共享服务器的会话内存(UGA)
 - I/O 服务器进程
 - 备份和恢复操作或 RMAN

- 并行执行消息缓冲区
- 将 PARALLEL_AUTOMATIC_TUNING 设置为 TRUE
- 不使用 LRU 列表
- 大小由 LARGE_POOL_SIZE 确定

Java 池

- 存储 Java 命令的服务分析要求
- 在安装并使用 Java 时是必需的
- 大小由 JAVA_POOL_SIZE 参数确定

程序全局区(PGA)

- 为连接到 Oracle 数据库的每个用户进程保留的内存
- 在创建进程时分配
- 在终止进程时回收
- 仅供一个进程使用

程序全局区或进程全局区(PGA) 是内存区，它包含有关单个服务器进程或单个后台进程的数据和控制信息。PGA 在创建进程时分配，并在终止进程时回收。与由若干个进程共享的 SGA 相比，PGA 是仅供一个进程使用的区。

PGA 内存的内容会因不同情况而变化，这取决于实例是在专用服务器配置还是在共享服务器配置下运行。一般来讲，PGA 内存包括下列组件：

- 专用 SQL 区：包含绑定信息和运行时内存结构之类的数据。发出 SQL 语句的每个会话均拥有一个专用 SQL 区。提交同一 SQL 语句的每个用户都拥有自己的使用单个共享 SQL 区的专用 SQL 区。因此，许多专用 SQL 区都与同一个共享 SQL 区相关联。

一个游标的专用 SQL 区可以分成以下两个区：

- 永久区：包含绑定信息，并且只在关闭游标时释放

– 运行时区：在执行请求时的第一步创建。对于 INSERT、UPDATE 和 DELETE 命令，该区在执行语句后释放，对于查询操作，该区只在提取所有行或取消查询后释放。

4. 进程结构(P52)

Oracle 利用了以下几种进程的优势：

- 用户进程：在数据库用户请求连接到 Oracle 服务器时启动
- 服务器进程：与 Oracle 实例相连接，在用户建立会话时启动
- 后台进程：在 Oracle 实例启动时启动

用户进程

- 请求与 Oracle 服务器交互的程序
- 必须先建立连接
- 不与 Oracle 服务器直接交互

服务器进程

- 直接与 Oracle 服务器交互的程序
- 执行生成的调用并返回相关结果
- 可以是专用服务器或共享服务器

后台进程

维护并加强物理结构与内存结构之间的关系

- 必备的后台进程：
 - DBWn PMON CKPT
 - LGWR SMON
- 可选的后台进程：
 - ARCn LMDn RECO

- CJQ0 LMON Snnn
- Dnnn Pnnn
- LCKn QMNn

数据库写入程序(DBWn)

在以下情况下写入：

- 出现检查点
- 灰数据缓冲区达到阈值
- 没有空闲缓冲区
- 出现超时
- 执行了 RAC ping 请求
- 表空间处于 OFFLINE 状态
- 表空间处于 READ ONLY 状态
- 对表执行 DROP 或 TRUNCATE 操作
- 对表空间执行 BEGIN BACKUP 操作

日志写入器(LGWR)

LGWR 在以下情况下写入：

- 提交时
- 三分之一填满时
- 有 1 MB 的重做时
- 每隔三秒
- DBWn 写入前

系统监控程序(SMON)

职责：

- 实例恢复

- 前滚重做日志中的更改
- 打开数据库供用户访问
- 回退未提交的事务处理

- 合并空闲空间
- 回收临时段

过程监视器(PMON)

实例失败后，通过以下方法

进行清理：

- 回退事务处理
- 释放锁
- 释放其它资源
- 重新启动已失效的调度程序

检查点(CKPT)

职责包括：

- 在检查点发信号给 DBWn
- 使用检查点信息更新数据文件的标头
- 使用检查点信息更新控制文件

归档程序(ARCn)

- 可选的后台进程
- 设置 ARCHIVELOG 模式时自动归档联机重做日志
- 保留数据库的全部更改记录

5. 处理 SQL 语句(P66)

处理查询(SELECT)：

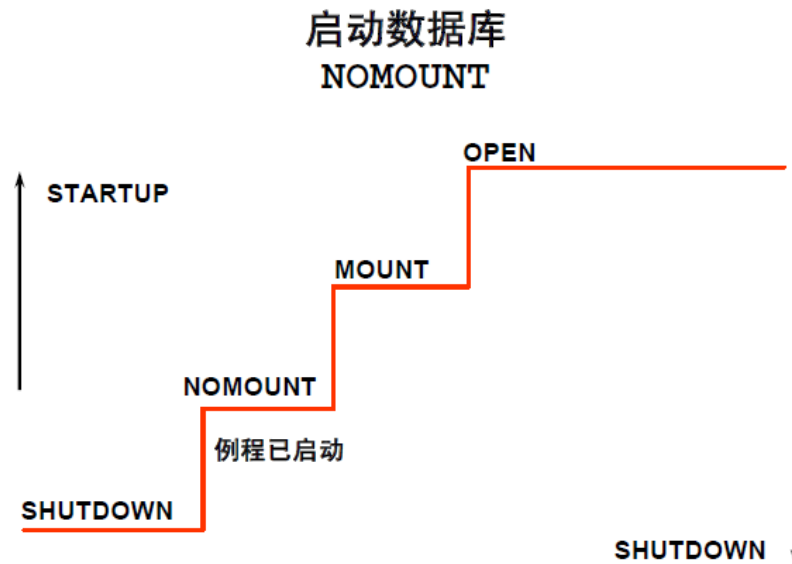
- 语法分析：
 - 搜索同一语句
 - 检查语法、对象名和权限
 - 锁定语法分析过程中使用的对象
 - 创建和存储执行计划
- 绑定：获取变量值
- 执行：处理语句
- 提取：将结果行返回用户进程

处理 DML 语句(INSERT\UPDATE\DELETE)：

- 语法分析：与处理查询时的语法分析阶段相同。
- 绑定：与处理查询时的绑定阶段相同。
- 执行：
 - 如果数据库缓冲区高速缓存中不存在某些数据块和还原块，服务器进程就从数据文件将它们读入数据库缓冲区高速缓存。
 - 服务器进程锁定要进行修改的行。还原块用于存储数据的前像，以便在需要时回退 DML 语句。
 - 数据块记录数据的新值。
 - 服务器进程将数据的前像记录到回退块中，并更新数据块。这两种更改都是在数据库缓冲区高速缓存中进行的。数据库缓冲区高速缓存中所有已更改的块都标记为灰数据缓冲区，即与磁盘中相应的块不同的缓冲区。
 - DELETE 或 INSERT 命令的处理使用类似的步骤。DELETE 命令的前像包含已删除行中的列值，而 INSERT 命令的前像中包含行的位置信息。

第三章 管理 Oracle 实例

1. 启动数据库(P113)



启动实例(NOMOUNT)：

仅在创建数据库或重新创建控制文件过程中，实例才会在 NOMOUNT 阶段启动。

启动实例包括下列任务：

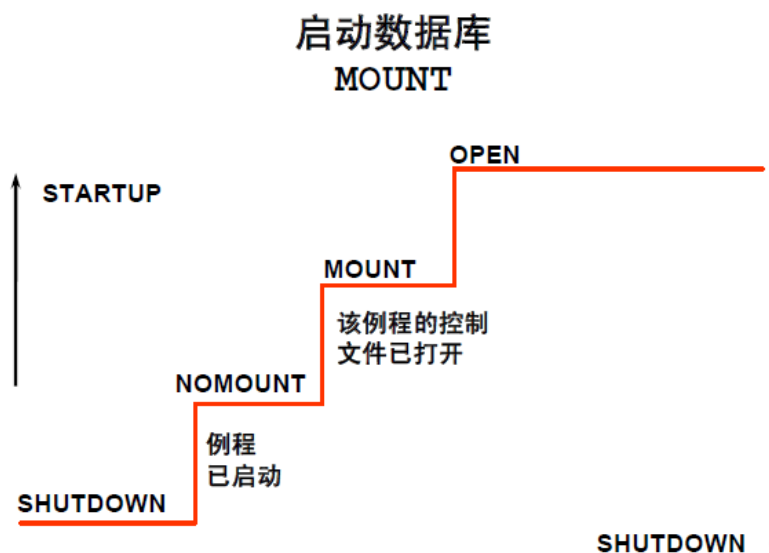
- 按以下顺序从\$ORACLE_HOME/dbs 读取初始化文件：
 - 首先读取 spfileSID.ora
 - 如果找不到，则读取 spfile.ora
 - 如果仍然找不到，则读取 initSID.ora

使用 STARTUP 指定 PFILE 参数以覆盖缺省行为。

- 分配 SGA
- 启动后台进程

- 打开 alertSID.log 文件和跟踪文件

必须在初始化文件中使用 DB_NAME 参数对数据库命名，或使用 STARTUP 命令命名。



加载数据库(MOUNT)：

若要执行特定的维护操作，可启动实例并加载数据库，但不要打开数据库。

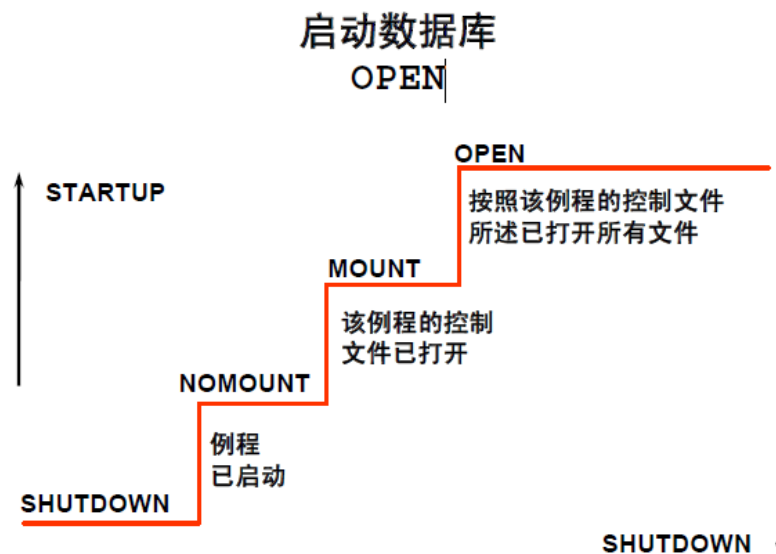
例如，在以下任务中必须加载数据库但不要打开数据库：

- 重命名数据文件
- 启用和禁用重做日志归档选项
- 执行完全数据库恢复

加载数据库包括以下任务：

- 使数据库与以前启动的实例关联
- 定位并打开参数文件中指定的控制文件

• 读取控制文件以获取数据文件和重做日志文件的名称和状态。但是，在此时不进行数据文件和联机重做日志文件是否存在的检查。



打开数据库(OPEN)：

正常的数据库操作指启动实例、加载数据库和打开数据库。通过正常的数据库操作，任何有效用户都可以连接到数据库并执行一般的数据访问操作。

打开数据库包括以下任务：

- 打开联机数据文件
- 打开联机重做日志文件

如果在尝试打开数据库时有任何数据文件或联机重做日志文件不存在，Oracle 服务器将返回错误消息。

在这个最后阶段中，Oracle 服务器验证所有数据文件和联机重做日志文件是否可以打开，并检查数据库的一致性。如果需要，系统监视(SMON) 后台进程将启动实例恢复操作。

STARTUP 命令

STARTUP [FORCE] [RESTRICT] [PFILE=filename]

[OPEN [RECOVER]][database]

|MOUNT

|NOMOUNT]

(注：这不是完整的语法。)

其中：

- OPEN：使用户能够访问数据库
- MOUNT：为某些 DBA 活动加载数据库，但不允许用户访问数据库
- NOMOUNT：创建 SGA 并启动后台进程，但不允许访问数据库
- PFILE=parfile：允许使用非缺省参数文件配置实例

2. 以只读模式打开数据库(P123)

- 以只读模式打开数据库

STARTUP MOUNT ALTER DATABASE OPEN READ ONLY;

- 此模式可用于：
 - 执行查询
 - 使用本地管理的表空间执行磁盘排序
 - 使数据文件（而不是表空间）脱机和联机
 - 执行脱机数据文件和表空间的恢复

只要数据库尚未以读写模式打开，就能以只读模式打开。该功能对于备用数据库从生产数据库卸载查询处理尤其有用。

如果查询需要使用临时表空间（例如，进行磁盘排序），当前用户必须将本地管理的表空间分配为缺省的临时表空间；否则查询会失败。对于用户 SYS，需要有本地管理的表空间。

只读模式不限制无需生成重做数据即可更改数据库状态的数据库恢复或操作。例如，在只读模式中：

- 可使数据文件脱机和联机。
- 可以执行脱机数据文件和表空间的恢复。

磁盘在写入其它文件，如控制文件、操作系统审计线索、跟踪文件和警报日志文件时，可仍然保持只读模式。

3. 关闭数据库(P125)

关闭数据库

关闭模式	A	I	T	N
允许建立新连接	否	否	否	否
等待到当前会话结束	否	否	否	是
等待到当前事务处理结束	否	否	是	是
强制执行检查点操作并关闭文件	否	是	是	是

关闭模式：

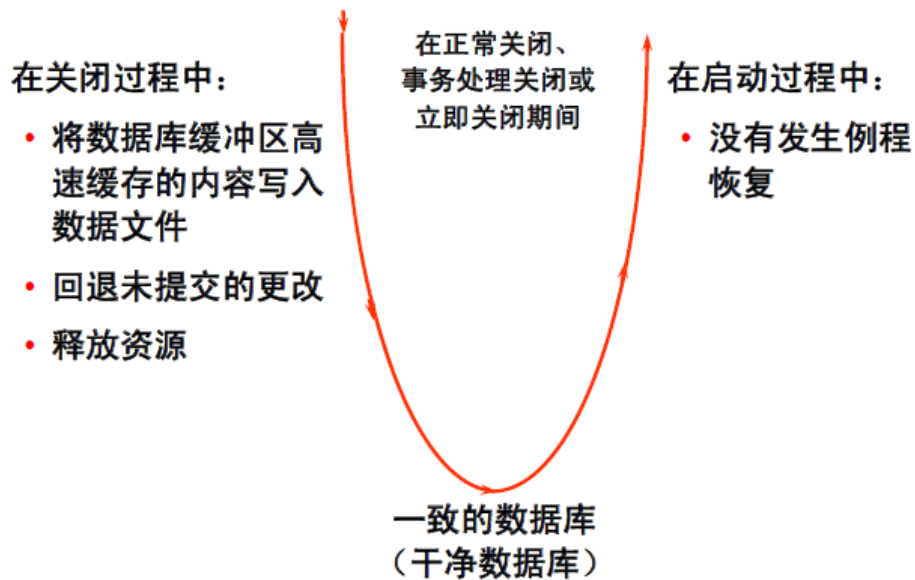
- A = ABORT
- I = IMMEDIATE
- T = TRANSACTIONAL
- N = NORMAL

关闭数据库这一操作将导致对所有物理结构进行操作系统脱机备份，并使修改过的静态初始化参数在重新启动后生效。

要关闭实例，必须使用以下命令以 SYSOPER 或 SYSDBA 身份进行连接：

SHUTDOWN [NORMAL | TRANSACTIONAL | IMMEDIATE | ABORT]

关闭选项



正常关闭：

“正常”是缺省的关闭模式。正常的数据库关闭在下列情况下进行：

- 未建立新连接。
- Oracle 服务器等待所有用户断开后才完成关闭。
- 数据库和重做缓冲区中的内容已写入磁盘。
- 后台进程已终止，SGA 已从内存中删除。
- Oracle 在关闭实例前将关闭并卸装数据库。
- 下一次启动将不要求实例恢复。

事务处理关闭：

事务处理关闭防止客户机丢失工作。事务处理数据库关闭在下列情况下进行：

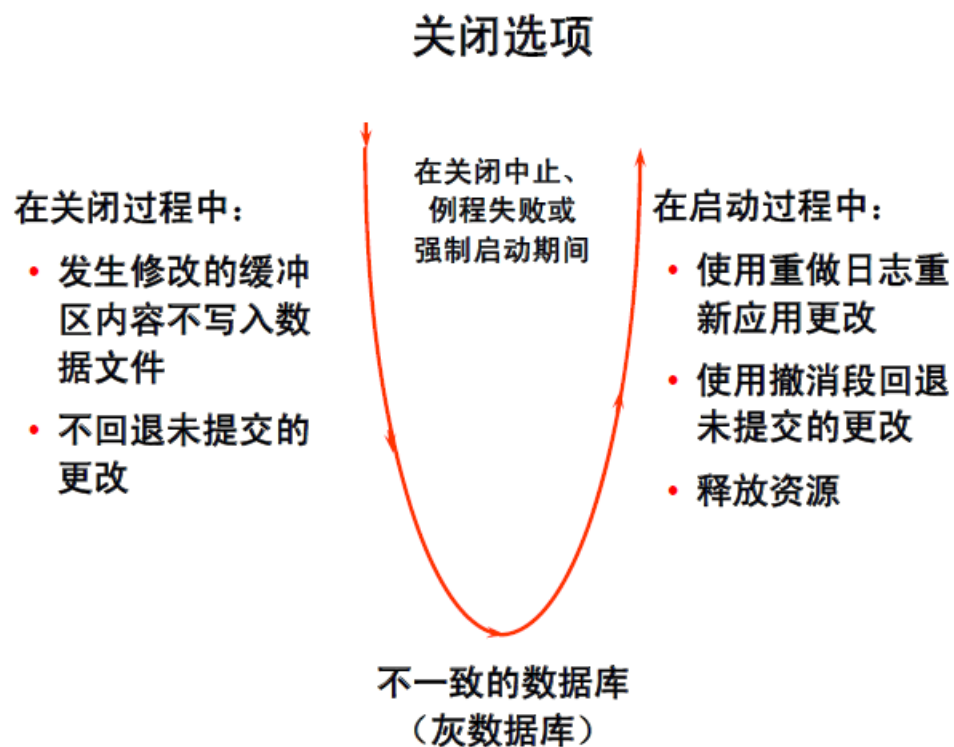
- 没有客户机可以在此特定实例上启动新事务。
- 当客户机结束正在进行的事务时，断开客户机。
- 当所有事务都已完成后立即关闭。

- 下一次启动将不要求实例恢复。

立即关闭：

立即关闭数据库在下列情况下进行：

- 由 Oracle 处理的当前 SQL 语句未完成。
- Oracle 服务器不等待当前连接到数据库的用户断开。
- Oracle 回退活动的事务并断开所有连接的用户。
- Oracle 在关闭实例前将关闭并卸装数据库。
- 下一次启动将不要求实例恢复。



关闭中止：

如果“正常”和“立即关闭”选项不起作用，可以中止当前数据库实例。中止实例在下列情况下进行：

- Oracle 服务器所处理的当前 SQL 语句被立即终止。
- Oracle 不等待当前连接数据库的用户断开。

- 数据库和重做缓冲区中的内容不写入磁盘。
- 未提交的事务不回退。
- 在不关闭文件的情况下实例被终止。
- 数据库不关闭或被卸装。
- 下次启动要求恢复实例，该操作将自动进行。

注：建议您不要备份处于不一致状态的数据库。

第四章 创建数据库

1. 创建的前提条件(P146)

要新建数据库，您必须具备以下条件：

- 已授权的帐户，通过以下方式之一验证：
 - 操作系统
 - 口令文件
- 有足够的内存可用于启动实例
- 有足够的磁盘空间可用于计划要创建的数据库

创建数据库需要具备 SYSDBA 权限。使用操作系统验证或口令文件验证即可授予这些权限。

创建数据库之前，确保有足够的内存可用于 SGA、Oracle 可执行程序 and 进程。请参考操作系统安装和管理指南。

计算数据库所需的磁盘空间，包括联机重做日志文件、控制文件和数据文件。

2. 创建数据库(P150)

可通过以下方式创建 Oracle 数据库：

- Oracle Universal Installer
- Oracle Database Configuration Assistant
 - 图形用户界面
 - 基于 Java
 - 由 Oracle Universal Installer 启动
 - 可独立使用
- CREATE DATABASE 命令

共有三种创建数据库的方式：使用 Oracle Universal Installer 在 Oracle9i 安装中自动创建；使用 Oracle Database Configuration Assistant (DBCA)；或使用 CREATE DATABASE 命令通过创建 SQL 脚本来创建数据库。

Database Configuration Assistant 是一个用来简化数据库创建操作的图形用户界面，它既能与 Oracle Universal Installer 交互使用，也可以独立使用。DBCA 基于 Java，可以从任何带 Java 引擎的平台启动。

安装 Oracle Server 的过程中，Oracle Universal Installer 会启动 DBCA，接着 DBCA 将自动创建一个初始数据库。对于用或不用 DBCA，创建或不创建初始数据库，以及是否在安装后作为独立的应用程序启动 DBCA 来创建数据库，您都有充分的选择自由。

如果使用的是较早版本的 Oracle 软件，还可以对现有数据库进行移植或升级。

3. 手动创建数据库(P157)

- 为实例和数据库选择唯一的名称。
- 选择一个数据库字符集。
- 设置操作系统变量。
- 创建初始化参数文件。
- 在 NOMOUNT 阶段启动该实例。
- 创建并执行 CREATE DATABASE 命令。

- 打开数据库。
- 运行脚本以生成数据字典并完成创建后的步骤。
- 根据需要创建其它表空间。

详细步骤：

- 为实例和数据库选择唯一的名称。
- 选择一个数据库字符集。

必须定义一个数据库字符集。同时还可选择定义一个国家字符集。例如：

- 字符集 Character set AL32UTF16
- 国家字符集 AL16UTF16 (National character set AL16UTF16)

有关可供使用的各种字符集的信息，请参考“使用全球化支持”一课。

- 设置操作系统变量。

需要设置四个环境变量：ORACLE_HOME、ORACLE_SID、PATH、LD_LIBRARY_PATH。

- ORACLE_HOME：安装 Oracle9i 服务器的顶级目录。
- ORACLE_SID：可由用户定义的、分配给数据库实例的名称。用于区分运行在同一台机器上的不同数据库实例
- PATH：定义操作系统查找可执行程序时要搜索的目录。
- LD_LIBRARY_PATH：定义所需的库文件的存储目录。

- 创建初始化参数文件。

初始化参数文件是通过随安装过程安装的 init.ora 示例文件而创建的。复制 init.ora 示例文件，将其命名为 initSID.ora。针对要创建的数据库的具体需要来修改该文件。如果要使用 SPFILE，则必须首先创建 PFILE。

- 在 NOMOUNT 阶段启动该实例。

以具有 SYSDBA 权限的用户 SYS 身份连接。要创建数据库，数据库必须处于 NOMOUNT 状态。

- 创建并执行 CREATE DATABASE 命令。

- 创建包含 CREATE DATABASE 命令的 SQL 脚本。以具有 SYSDBA 权限的用户 SYS 身份连接到 SQL*Plus。当数据库处于 NOMOUNT 状态时，执行该脚本。

- 如果要创建的数据库是通过“Oracle 管理文件” (Oracle Managed Files, OMF) 来管理操作系统文件的，那么 CREATE DATABASE 命令的简化程度将非常明显。

- 打开数据库。

必须首先打开数据库，然后再运行脚本，创建数据字典并完成创建后的步骤。

- 运行脚本。

- 创建数据库后必须运行两个脚本：catalog.sql 和 catproc.sql。这两个脚本都必须以具有 SYSDBA 权限的用户 SYS 身份运行。执行脚本前，数据库必须处于 OPEN 状态。

- catalog.sql：在基表和动态性能视图上创建视图及其同义词。它还启动其它脚本，为以下各项创建对象：

- PL/SQL 基本环境，包括 PL/SQL 数据类型的声明、预定义异常、内置过程和函数、SQL 操作等

- 审计

- 导入/导出

- SQL*Loader

- 已安装选项

- catproc.sql：创建使用 PL/SQL 所需的程序包和过程。此外，此脚本还创建用于扩展 RDBMS 功能的若干 PL/SQL 程序包，以及用于预警、管道、logminer、大对象、对象、排队、复制和其它内置选项的程序包视图。

- pupbld.sql：创建名为“产品用户配置文件” (Product User Profile) 的表以及相关的过程。运行此脚本将在用户每次连接到 SQL*Plus 时防止生成警告消息。

注：必须以用户 SYSTEM 的身份运行此脚本。

- 创建其它表空间。

- 应该根据数据库的需要创建其它表空间。

第五章 使用数据字典和动态性能视图

1. 数据字典内容(P177)

数据字典提供有关以下方面的信息：

- 逻辑数据库结构和物理数据库结构
- 对象的定义和空间分配
- 完整性约束
- 用户
- 角色
- 权限
- 审计

数据字典包含以下内容：

- 数据库内所有方案对象的定义，这些对象包括表、视图、索引、簇、同义词、序列、过程、函数、程序包、触发器等等
- 已为方案对象分配的空间量以及它们当前使用的空间量
- 列的缺省值
- 完整性约束信息
- Oracle 用户的名称
- 已授予每个用户的权限和角色
- 审计信息，如有哪些人访问或更新了各种方案对象

2. 动态性能表(P182)

- 虚拟表
- 记录当前的数据库活动
- 在数据库可操作时不断更新
- 通过内存和控制文件访问信息
- 用于监控和优化数据库
- 由 SYS 用户拥有
- 同义词以 V\$ 开头
- 在 V\$FIXED_TABLE 中列出

在 Oracle 服务器的整个操作过程中，它将当前数据库活动记录在称为动态性能视图的一组虚拟表中。只有数据库处于运行状态时，这些虚拟表才驻留在内存中，反映数据库操作的实时状况。它们指向内存和控制文件中的实际信息源。

这些表不是真正的表，大多数用户都无法访问它们；但是数据库管理员可以在这些视图上查询、授予 SELECT 权限并创建视图。这些视图有时称为固定视图，因为数据库管理员无法更改或删除这些视图。

动态性能表由 SYS 拥有，它们的名称均以 V_\$ 开头。在这些表上先创建视图，然后再为这些视图创建公共同义词。同义词名以 V\$ 开头。例如，V\$DATAFILE 视图包含有关数据库中数据文件的信息，而 V\$FIXED_TABLE 视图包含有关数据库中所有动态性能表和视图的信息。

动态性能表可使用户了解到以下信息：

- 该对象是否处于联机状态并可用
- 该对象是否已打开
- 目前持有哪些锁
- 该会话是否处于活动状态

动态性能表示例

- V\$CONTROLFILE
- V\$DATABASE
- V\$DATAFILE
- V\$INSTANCE
- V\$PARAMETER
- V\$SESSION
- V\$SGA
- \$SPPARAMETER
- V\$TABLESPACE
- V\$THREAD
- V\$VERSION

第六章 维护控制文件

1. 控制文件的内容(P195)

控制文件中包含以下条目：

- 数据库名称和标识符
- 创建数据库的时间戳
- 表空间的名称
- 数据文件和重做日志文件的名称和位置
- 当前重做日志的序列号
- 检查点信息
- 还原段的开始和结尾
- 重做日志归档信息

- 备份信息

控制文件中的信息包括下列内容：

- 数据库名称取自初始化参数 DB_NAME 所指定的名称或 CREATE DATABASE 语句中所用的名称。
- 当创建数据库时会记录数据库标识符。
- 创建数据库时还会记录创建数据库的时间戳。
- 当在数据库中添加、重命名或删除数据文件或重做日志时，会更新相关数据文件和联机重做日志文件的名称和位置。
- 当添加或删除表空间时会更新表空间信息。
- 在日志切换过程中会记录重做日志历史信息。
- 归档日志的位置和状态会在归档时记录。
- 备份的位置和状态由“恢复管理器” (Recovery Manager) 实用程序记录。
- 在进行日志切换时记录当前日志序列号。
- 在建立检查点时记录检查点信息。

控制文件由以下两种类型的部分组成：

- 可重用
- 不可重用

可重用部分存储“恢复管理器” (Recovery Manager) 的信息，如备份数据文件名和备份重做日志文件名。只有“恢复管理器” (Recovery Manager) 才能以循环方式重新使用这些部分。

第八章 管理表空间和数据文件

1. 表空间类型(P240)

- SYSTEM 表空间
 - 随数据库创建
 - 包含数据字典
 - 包含 SYSTEM 还原段
- 非 SYSTEM 表空间
 - 用于分开存储段
 - 易于空间管理
 - 控制分配给用户的空间量

为加强控制和方便维护，DBA 创建了表空间。Oracle 服务器识别两种类型的表空间：

SYSTEM 和所有其它表空间。

SYSTEM 表空间：

- 随数据库创建
- 所有数据库均需要
- 包括数据字典（内含存储程序单元）
- 包含 SYSTEM 还原段
- 应不包括用户数据，尽管允许这样做

非 SYSTEM 表空间：

- 支持更灵活地管理数据库

- 将还原段、临时段、应用程序数据段和应用程序索引段分开
- 根据备份要求将数据分开
- 分开动态和静态数据
- 控制分配给用户对象的空间量

2. 表空间的空间管理(P245)

- 本地管理的表空间：
 - 在表空间内管理空闲区
 - 使用位图来记录空闲区
 - 每一位与一个块或一组块相对应
 - 位的数值指明是空闲还是已占用
- 字典管理的表空间：
 - 由数据字典管理空闲区
 - 在分配或回收区时更新对应的表

表空间以区为单位分配空间。可使用以下两种不同方法来跟踪创建的表空间中的空闲空间和已占用空间：

本地管理的表空间：在表空间内通过位图管理区。位图中的每个位对应于一个块或一组块。分配了某个区或释放了某个区可重新使用时，Oracle 服务器更改位图值以显示块的新状态。从 Oracle9i 开始，在本地管理已成为缺省设置。

字典管理的表空间：由数据字典管理区。Oracle 服务器将在分配或回收区时更新数据字典中对应的表。

本地管理的表空间

- 减少了对数据字典表的争用
- 分配或回收空间时不生成还原数据

- 无需合并

```
CREATE TABLESPACE userdata DATAFILE '/u01/oradata/userdata01.dbf' SIZE
500M EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K;
```

在本地管理表空间的优点：

- 本地管理可以避免循环空间管理操作，但是这种操作在字典管理的表空间中却有可能发生。一旦消耗或释放某个区的空间会产生另一个消耗或释放操作（消耗或释放还原段或数据字典表内的空间）时，它就会发生。

- 由于本地管理的表空间在数据字典表中不记录空闲空间，从而减少了对这些表的争用。

- 区的本地管理可自动跟踪邻近的空闲空间，因而无须合并空闲区。

- 本地管理的区大小可由系统自动确定。

- 对区的位图进行更改不会生成还原信息，因为它们不更新数据字典中的表（表空间限额信息等特殊情况除外）。

字典管理的表空间

- 在数据字典中管理区

- 存储在表空间中的每个段都可以有不同的存储子句

- 需要合并

```
CREATE TABLESPACE userdata DATAFILE '/u01/oradata/userdata01.dbf' SIZE
500M EXTENT MANAGEMENT DICTIONARY DEFAULT STORAGE (initial 1M NEXT 1M
PCTINCREASE 0);
```

字典管理的表空间内的段可具有自定义的存储设置。这比本地管理的表空间更灵活，但效率要低得多。

3. 缺省临时表空间(P253)

- 指定数据库范围内的缺省临时表空间

- 避免使用 SYSTEM 表空间存储临时数据

- 可使用以下命令进行创建：

- CREATE DATABASE

- 在本地管理

- ALTER DATABASE

ALTER DATABASE DEFAULT TEMPORARY TABLESPACE temp;

创建未指定缺省临时表空间的数据库时，分配给任意用户的，未使用 TEMPORARY TABLESPACE 子句创建的表空间是 SYSTEM 表空间。此时，alert_sid.log 中会记录一条警告消息，指出 SYSTEM 表空间是缺省临时表空间。在创建数据库期间创建缺省临时表空间可防止将 SYSTEM 表空间用作临时空间。

创建数据库后，可通过创建临时表空间然后改变数据库来设置缺省临时表空间。

```
SQL> ALTER DATABASE DEFAULT TEMPORARY TABLESPACE temp;
```

定义后，未显式分配到临时表空间的用户将被分配到该缺省临时表空间。缺省临时表空间可通过使用 ALTER DATABASE DEFAULT TEMPORARY TABLESPACE 命令随时进行更改。更改缺省临时表空间后，分配到该缺省临时表空间的所有用户将被重新分配到新的缺省表空间。

4. 缺省临时表空间的限制(P257)

不能对缺省临时表空间执行下列操作：

- 将其删除，除非已经有新的缺省临时表空间
- 使其脱机
- 更改为永久表空间

删除缺省临时表空间

您只有指定了一个新的缺省表空间后，才能删除旧的缺省临时表空间。必须使用 ALTER DATABASE 命令才能将缺省临时表空间更改为新的缺省值。旧的缺省临时表空

间仅在新的缺省临时表空间可用时才会被删除。分配到旧的缺省临时表空间的用户将被自动重新分配到新的缺省临时表空间。

更改缺省临时表空间的类型

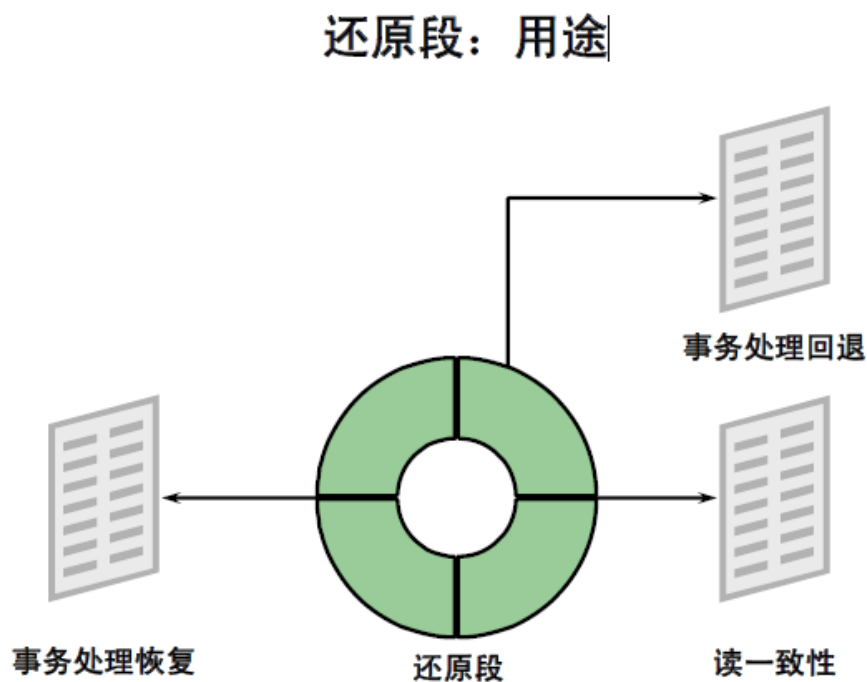
由于缺省临时表空间必须是 SYSTEM 表空间或临时表空间，因此，无法将缺省临时表空间更改为永久类型。

使缺省临时表空间脱机

使表空间脱机后，例如在进行脱机备份、维护或更改使用该表空间的应用程序时，其他用户将无法使用对应的这部分数据库内容。由于上述情况都不适用于临时表空间，因此无法使缺省临时表空间脱机。

第十章 管理还原数据

1. 还原段用途(P323)



事务处理回退

当某事务处理修改表中某行时，被修改的列的旧映像（要还原的数据）将存储在还原段中。如果将该事务处理回退，则 Oracle 服务器通过将还原段中的值写回到该行来恢复原始值。

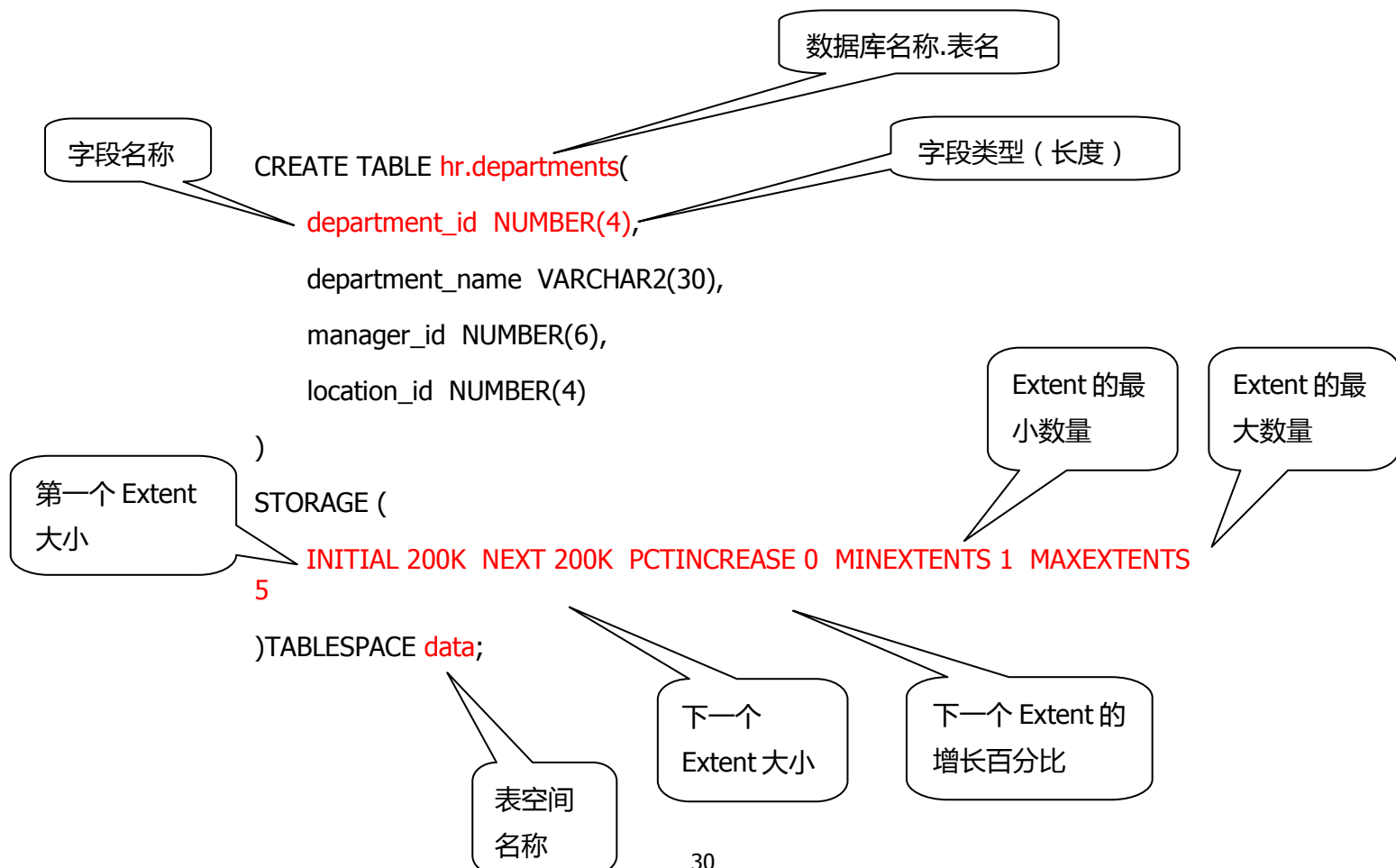
事务处理恢复

如果实例在事务处理正在进行时失败，那么 Oracle 服务器需要在数据库再次打开时还原所有未提交的更改。这种回退操作是事务处理恢复的一部分。之所以有可能恢复事务处理，原因在于对还原段所做的更改同样受重做日志文件的保护。

读一致性

在事务处理正在进行时，数据库中的其他用户不应看到这些事务处理所做的任何未提交更改。此外，也不应从某条语句中看到该语句开始执行后所提交的任何更改。还原段中的旧值（要还原的数据）也可用于为读者提供给定语句的一致映像。

附 1 建表 SQL 示例



附 2 增删改查 SQL 示例

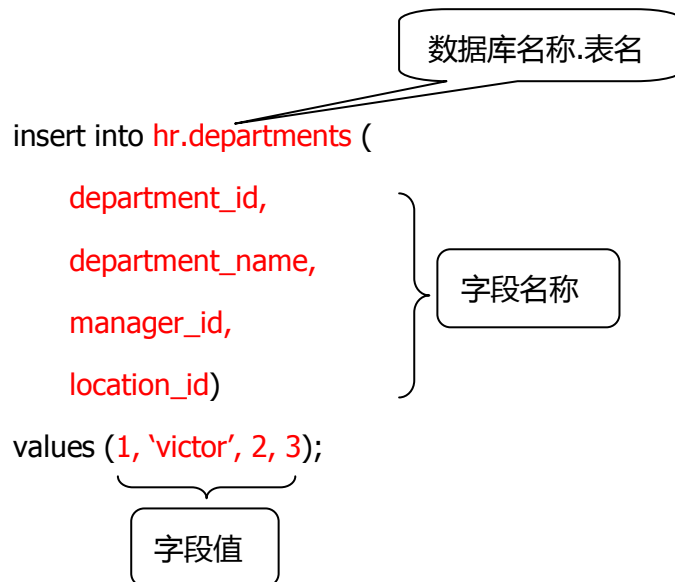
1. 增加 (INSERT)

insert into **hr.departments** (
 department_id,
 department_name,
 manager_id,
 location_id)
values (**1, 'victor', 2, 3**);

数据库名称.表名

字段名称

字段值



2. 删除 (DELETE)

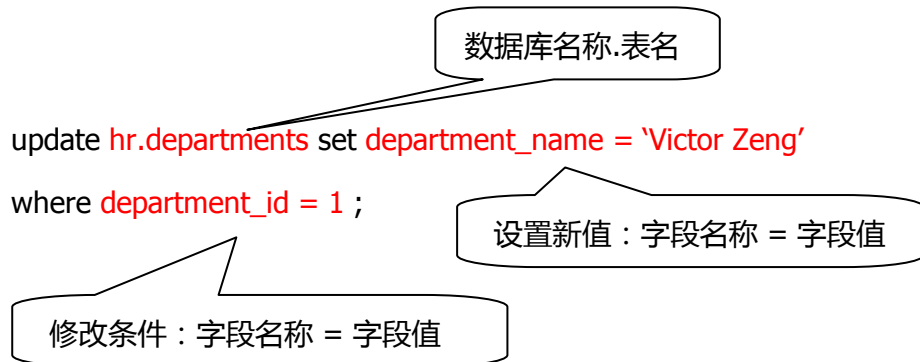
delete from **hr.departments** where **department_id = 1** ;

数据库名称.表名

删除条件：字段名称 = 字段值



3. 修改 (UPDATE)



4. 查询 (SELECT)

