

第 2 章 Oracle 数据库系统

Oracle 系统是一个大型数据库系统，要想全面理解和掌握 Oracle 数据库，必须熟悉 Oracle 的体系结构。本章主要讲述有关 Oracle9i 数据库的体系结构。

2.1 Oracle 数据库系统结构

2.1.1 Oracle 系统结构

Oracle9i 是以高级结构化查询语言（SQL）为基础的大型关系数据库，通俗地讲它是用方便逻辑管理的语言操纵大量有规律数据的集合，是目前最流行的客户/服务器（Client/Server）体系结构的数据库之一。对 Oracle9i 数据库系统本身而言，大致上可区分为两个主要部分：

- （1）Oracle9i 数据库实例（Oracle9i Instance）。
- （2）Oracle9i 数据库文件（Database files）。

Oracle 实例是指数据库拥有自己的系统全局区和相关数据库文件的 Oracle 服务器进程集。每个打开的 Oracle 数据库均有一个相关的 Oracle 实例支撑。简言之，Oracle9i 数据库实例是指数据库服务器的内存与相关处理程序，数据库文件则由操作系统内的各种文件组成。Oracle9i 服务器的基本架构如图 2.1 所示。

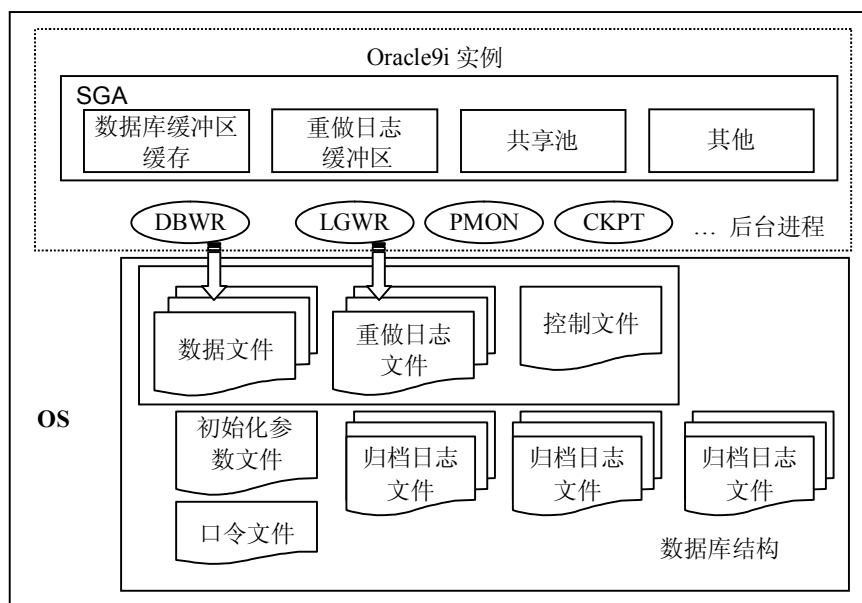


图 2.1 Oracle9i 服务器基本架构

图 2.1 的上半部为内存区域内的 Oracle9i 数据库实例，下半部则是位于操作系统的各种数据库文件。彼此之间是通过各个后台进程互相通信的。

Oracle 数据库系统是具有管理 Oracle 数据库功能的计算机系统。每一个运行的 Oracle 数据库与一个 Oracle 实例 (Instance) 相联系。一个 Oracle 实例是存取和控制数据库的软件机制。每一次在数据库服务器上启动一个数据库实例时，称为系统全局区 (System Global Area) 的内存区域 (简称 SGA) 被分配，有一个或多个 Oracle 进程被启动。该 SGA 和 Oracle 进程的结合称为一个 Oracle 数据库实例。一个实例的 SGA 和进程对内高效地管理数据库的所有数据，对外为用户和应用程序访问数据库提供服务。

在 Oracle 数据库系统的启动过程中，首先是实例启动，然后由实例装配 (Mount) 数据库。实例和数据库的关系如图 2.2 所示。

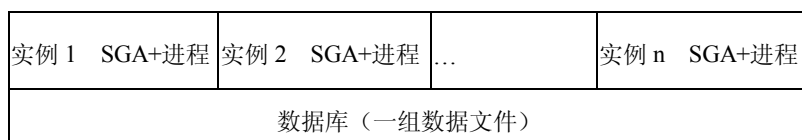


图 2.2 数据库与实例的关系图

一个 Oracle 数据库是数据的集合，被处理成一个单位。一个 Oracle 数据库有一个物理存储结构和一个逻辑存储结构。

数据库物理存储结构是由构成数据库的操作系统文件所决定的。每一个 Oracle 数据库由三种类型的文件组成：数据文件、日志文件和控制文件。数据库的文件为数据库信息提供真正的物理存储。

数据库逻辑存储结构是用户所涉及的数据库结构。一个 Oracle 数据库的逻辑结构由下列因素决定：

- (1) 一个或多个表空间。
- (2) 数据库模式对象 (即表、视图、索引、序列、同义词和存储过程等)。

逻辑存储结构如表空间 (Tablespace)、段 (Segment)、区间 (Extent) 和数据块 (Data Block) 将支配一个数据库的物理空间如何使用。模式是与每个 Oracle 数据库用户相关的一组数据库对象的集合。模式对象 (schema object) 用它们之间的联系组成了一个数据库的关系设计。

2.1.2 分布式 Oracle 数据库

物理上存放于网络的多个 Oracle 数据库，逻辑上可以看成一个大数据库。用户可以通过网络对异地数据库中的数据同时进行存取，而服务器之间的协同处理对于工作站用户及应用程序而言是完全透明的，开发人员无需关心网络的连接细节、无需关心数据在网络接点中的具体分布情况、也无需关心服务器之间的协调工作过程。图 2.3 给出了分布式 Oracle 数据库的示意图。

由网络相连的两个 Oracle 数据库之间通过数据库链接 (DB-Links) 建立访问机制，相

当于一方以另一方的某用户远程登录所做的操作。但 Oracle 采用的一些高级管理方法，如同义词（Synonme）等使我们觉察不到这个过程，似乎远端的数据就在本地。

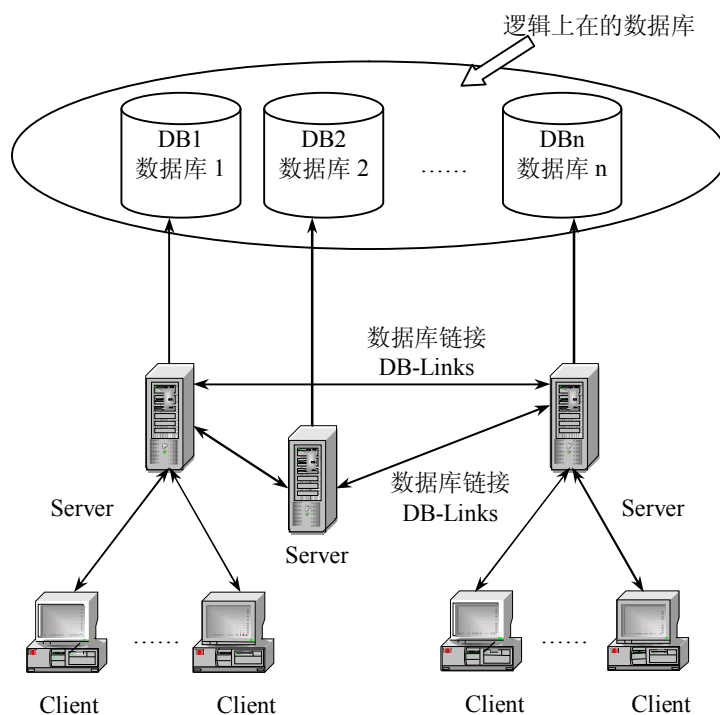


图 2.3 分布式 Oracle 数据库

2.2 Oracle 内存结构

内存是 Oracle 重要的信息缓存和共享区域，主要存储执行的程序代码、连接的会话信息以及程序执行期间所需数据和共享的信息等。Oracle 使用的主要内存结构包括系统全局区（System Global Area，简称 SGA）和程序全局区（Program Global Area，简称 PGA）。

2.2.1 系统全局区

系统全局区（SGA，System Global Area）是内存结构的主要组成部分，是 Oracle 为一个实例分配的一组共享内存缓冲区，保存着 Oracle 系统与所有数据库用户的共享信息，包括数据维护、SQL 语句分析和重做日志管理等，是实例的主要部分。

当启动 Oracle9i 数据库时，系统会先在内存中规划一个固定区域，用来存储每个使用者所需存取的数据以及 Oracle9i 运作时必备的系统信息。当实例启动时，SGA 的存储自动被分配；当实例关闭时，SGA 的存储被回收。SGA 包含数个重要区域，分别是：数据库缓冲区缓存、重做日志缓冲区、共享池和大块内存池。

1. 数据库缓冲区（Database Buffer Cache）

数据块缓冲区又称用户数据高速缓冲区，为所有与该实例相链接的用户进程所共享。

数据块缓冲区的容量受物理容量限制。

数据块缓冲区作为 SGA 的主要成员，用来存放从数据库数据文件中读取的数据，或者是使用者曾经处理过的数据。其用途在于有效减少存取数据时造成的磁盘读写动作，进而提高数据存取的效率。所有同时联机的用户都可以共享此缓冲区的数据。

当使用者第一次向 Oracle9i 送出数据查询请求时，Oracle9i 会先在数据库缓冲区的缓存内寻找该数据。如果欲查询的数据恰好已存在于缓存内，系统就直接对它们进行处理，这种情况我们称作缓存命中。反之，如果缓存内并没有使用者欲查询的数据，系统则首先搜索数据库缓冲区缓存，在其中查找空闲的数据块，之后将数据库数据文件中的数据读入这些空闲的数据块中处理。如果在缓存中所搜索的数据块数量达到指定的阈值仍未找到空闲的数据块时，用户进程将通知 Oracle 后台进程 DBW0 将缓存中一些修改过的数据写入磁盘，以腾出空间保存新的需要处理的数据。所以，Oracle 在修改数据时，并不是在每次修改数据后直接将它们写入磁盘，而是当所占用的缓存数据块达到一定数量后才写入一次。由于对内存操作远比磁盘操作速度快，所以这种多次修改一次写入的方法可以大大提高系统性能。

2. 重做日志缓冲区 (Redo Log Buffer)

重做日志缓冲区用于缓存数据库重做日志记录，以便数据库毁损时可以进行必要的恢复。对数据库进行修改的任何事务 (Transaction) 在记录到重做日志之前都必须首先放到重做日志缓冲区 (Redo Log Buffer) 中。重做日志缓冲区是专为此开辟的一块内存区域，重做日志缓存中的内容将被 LGWR 后台进程随时写入重做日志文件。

重做日志缓存是一个循环缓存区，在使用时从顶端向底端写入数据，然后再返回到缓冲区的起始点循环写入。

3. 共享池 (Shared Pool)

共享 SQL 池 (Shared SQL Pool) 相当于程序高速缓冲区，所有的用户程序都存放在共享 SQL 池中。

当使用者将 SQL 指令送至 Oracle 数据库后，系统将会先解析语法是否正确。解析时所需要的系统信息，以及解析后的结果将放置在共享池内。如果不同的使用者执行了相同的 SQL 指令，就可以共享已解析好的结果，加速 SQL 指令的执行速度。

共享池内包含数种不同用途的快速缓冲区，SQL 共享池主要可分为库缓存、数据字典缓存两类。

(1) 库缓存 (Library cache)。包含共享 SQL 区 (Shared SQL Area)、私有 SQL 区 (Private SQL Area) 以及 PL/SQL 程序单元区。解析完成的结果就放在共享 SQL 区内。

(2) 数据字典缓存 (Dictionary cache)。Oracle9i 在解析 SQL 语句时所需要的系统信息都存放在此缓冲区内，可能包含对象名称、列定义、权限设置和存储控制等，可以加快 Oracle 对 SQL 语句的执行速度。

4. 大块内存池 (Large pool)

大块内存池是数据库管理员的一个可选内存配置项，它主要用来为 Oracle 多线程服务器、服务器 I/O 进程以及数据库备份和恢复操作等分配连续的大块内存区域。

2.2.2 程序全局区

程序全局区 PGA (Program Global Area, PGA), 是单个 Oracle 进程 (包括服务器进程和后台进程) 使用的内存区域, 不属于实例的内存结构。它含有单个进程工作时需要的数据和控制信息, PGA 是非共享的, 只有服务进程本身才能够访问它自己的 PGA 区。

程序全局区又称为进程全局区 (Process Global Areas), 为指定进程所专用, 其他进程不能读写该内存区域中的任何数据。当用户连接服务器并建立会话时, Oracle 分配一个程序全局区。

程序全局区包含进程栈空间和会话信息, 在栈空间中存储会话变量、数组等。此外, 如果实例没有运行在多线程服务器环境中时, 用户会话中的专用 SQL 区也分配在程序全局区内。

2.2.3 排序区

排序区存在于请求排序的用户进程的内存中。由于排序需要内存空间, Oracle 利用该内存排序数据, 这部分空间称为排序区。该空间的大小为适应排序数据量的大小, 可增长, 但受初始化参数的限制。

2.2.4 软件代码区

软件代码区 (Software Code Area) 用于存储正在执行的或可以执行的程序代码。软件代码区是只读, 可安装成共享或非共享。Oracle 系统程序是共享的, 多个 Oracle 用户可存取它, 而不需要在内存中有多个副本。用户程序可以共享也可以不共享。

2.3 Oracle 进程

进程又称任务, 是操作系统中一个极为重要的概念。一个进程执行一组操作, 完成一个特定的任务。对 Oracle 数据库管理系统来说, 进程由用户进程、服务器进程和后台进程所组成。

进程与程序的区别在于前者是一个动态概念, 后者是一个静态实体。程序仅仅是指令的有序集合, 而进程则强调执行过程; 进程可以动态地创建, 完成任务后即会消亡。

2.3.1 服务器进程

服务器进程处理与应用程序相连的用户进程的请求, 它与用户进程相通讯, 为相连的用户进程的 Oracle 请求服务。

服务器进程处理用户进程的各种请求: 解释、编译和执行用户进程发出的 SQL 语句, 负责将需要处理的数据从数据库数据文件读入到 SGA 中的数据库缓冲区, 并将 SQL 语句执行结果返回给客户端应用程序。

服务器进程又分为专用服务器进程和共享服务器进程两类, 每个专用服务器只能服务于一个指定的用户进程; 而在多进程服务器配置下, 每个共享服务器进程则能同时服务于多个用户进程。

2.3.2 后台进程

为了使系统性能最好，协调多个用户，在多进程系统中使用一些附加进程，称为后台进程。在许多操作系统中，后台进程是在实例启动时自动建立。一个 Oracle 实例可以有許多后台进程，但它们不是一直都存在的。

1. 数据库写入进程 DBWn (Database Writer)

DBW 负责将 SGA 数据库缓冲区中的数据写入数据文件。Oracle 系统预设只会启动一个 DBWn (DBW0) 处理程序。但在一般的大型联机事务处理 (OLTP) 系统下，数据库改动情况可能十分频繁，用户可依实际需求额外配置其他的数据库写入处理程序 (DBW1-DBW9)，可以有效地提升 Oracle9i 写入数据文件的效率。

注意：在单一处理器的服务器系统中，配置额外的 DBWn 处理程序并没有实质性的帮助。

2. 日志写入进程 LGWR (Log Writer)

在事务被提交时，LGWR 会负责将重做日志缓冲区内的重做记录循序写入重做日志文件。当下面条件之一满足时，LGWR 进程写入一次：

- 用户进程提交事务。
- 自动周期性地动作，时间间隔超过 3 秒。
- 重做日志缓冲区的剩余空间不到 2/3。
- 数据库写入进程 DBWn 执行写入操作之前，它将数据库缓冲区中的数据写入数据文件。

3. 系统监视进程 SMON (System Monitor)

如果是因为停电或其他因素导致 Oracle 数据库常被不正常关闭，下一次启动数据库时将由 SMON 进行必要的数据库修复动作。其主要工作有：

- 数据库重新启动后的必要修复动作。
- 回收不用的临时段空间。
- 收集基于数据字典方式管理的表空间中的自由连续区域。
- 修复事务不正常结束造成的问题。

4. 进程监视进程 PMON (Process Monitor)

当某个用户进程异常终止时，PMON 会执行进程恢复操作。清除数据库缓存区中的缓存数据，并释放用户进程所占用的系统资源及其锁定的模式对象，并从系统活动进程列表中删除用户进程标识号。

PMON 也会定期检查各服务器进程以及分配器的状态，如果某个进程因故被挂起，也由 PMON 负责重新启动。

5. 检查点进程 CKPT (Checkpoint)

CKPT 会在适当的时候产生一个检查点事件，其意义为：

- (1) 确保缓冲区内经常被改动的数据定期被写入数据文件。

对于这点，也许大家会觉得非常疑惑。Oracle9i 不是会尽量将用户“最常使用的数据”保留在缓冲区内，以提高数据存取的效率吗？没错，但请大家思考一个问题：如果仅将最

不常使用的数据回写至数据文件，这些“经常使用”的数据反而会没机会存回硬盘！如果数据库发生毁损，这些数据只能从重做日志文件的记录中才能还原回来，无形中造成系统的额外负担。所以由 CKPT 掌控检查点时机，以确保这些数据照样可被妥善储存。

(2) 在检查点之后，因为所有更新过的数据已经回写至磁盘数据文件，万一 Oracle9i 需要进行实例恢复时，就不再需要检查点之前的重做记录，可缩短数据库重新启动的时间。此外，检查点发生后，CKPT 会先通知 DBWn 将数据库缓冲区缓存中的数据回写到数据文件，然后更新数据文件与控制文件的检查点信息。

6. 数据库恢复进程 RECO (Recover)

在 Oracle9i 分布式数据库环境中，RECO 处理程序会自动处理分布式事务失败时产生的问题。何谓分布式事务呢？简单的说，就是在同一个事务内针对多个数据库同时进行数据处理动作。与传统数据事务相比较之下，分布式事务的要求将严格许多！因为这些数据库可能分散在网络上，所以分布式事务的成功与否其中一个关键因素就在于网络传输速度与品质。试想，某个分布式事务需要将数据同时送进 5 个 Oracle9i 数据库，此时可能因为网络问题(例如断线或是响应等待时间过长)，导致其中一个数据库无法更新这笔事务数据。那么此事务就会发生问题，无法做最后的提交操作。这时 RECO 进程会自动重新登录包含在分布式事务中的各数据库，并尝试进行修复操作。

7. 日志存档进程 ARCn (Archiver)

Oracle9i 数据库设定为 ARCHIVELOG mode (存档日志模式) 时，ARCn 处理程序会在 DBA 执行 ALTER SYSTEM SWITCH LOGFILE 命令强制进行日志切换时自动将重做日志文件复制一份到指定的目录下；重做日志文件的副本称为存档日志文件。

每个 Oracle9i 实例最多可启动 10 个 ARCn 处理程序 (ARC0-ARC9)。

2.3.3 用户进程

用户进程为运行用户应用程序或 Oracle 工具所建立的进程，它们向 Oracle 服务器发送所要执行的 SQL 语句。

当用户运行一个应用程序时，系统就会为它建立一个用户进程。用户进程执行的是一个应用程序或 Oracle 工具程序的代码，以完成用户所指定的任务。用户进程不是实例。

2.4 Oracle 数据库物理存储结构

Oracle 数据库在物理上是存储于硬盘的各种文件。它是活动的，可扩充的，随着数据的添加和应用程序的增大而变化。如图 2.4 所示是 Oracle 数据库扩充前后在硬盘上的存储结构的示意图。

从数据的物理存储结构来看，Oracle 数据库由数据文件、重做日志文件和控制文件组成。

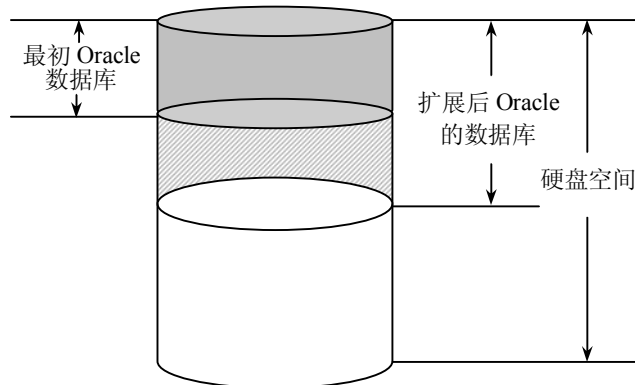


图 2.4 Oracle 数据库扩充前后在硬盘上的存储结构

2.4.1 数据文件

每一个 Oracle9i 数据库有一个或多个数据文件，而一个数据文件只能属于一个表空间。数据文件创建后可改变大小，创建新的表空间需要创建新的数据文件。数据文件一旦加入到表空间，就不能从这个表空间中移走，也不能和其他表空间发生联系。

如果数据库对象存储在多个表空间中，那么可以通过把它们各自的数据文件存放在不同的磁盘上来对其进行物理分割。

Oracle 数据库模式对象中的所有数据均存储在数据文件中，也就是说，数据文件是 Oracle 数据库的惟一物理存储对象，所以每个 Oracle 数据库至少由一个数据文件组成。数据文件用来存储数据库中的全部数据，如数据库表中的数据 and 索引数据。数据文件通常是后缀名为 .dbf 格式的文件。

从理论上讲，Oracle 对每个数据库中的数据文件数量没有限制。但是，操作系统往往限制每个进程能够同时打开的文件数量。所以，每个数据库中的数据文件数不能超过此限制。此外，太多的数据文件也将影响数据库的操作性能。因此，当数据库存储空间不够时，建议通过增加每个数据文件大小的方法来扩大整个数据库的存储空间，而尽量不要使用向数据库添加太多小数据文件的方法来扩大数据库的存储空间。

数据文件实际上是一个操作系统文件，这些文件中的数据格式是只有 Oracle 数据库系统才能够正确识别的二进制格式。虽然操作系统允许使用其他工具访问这些数据文件，但对数据文件的任何修改均可能导致 Oracle 数据库的崩溃，所以用户不要使用非 Oracle 工具来修改这些数据文件。

2.4.2 重做日志文件

除了数据文件外，最重要的 Oracle9i 数据库实体档案就是重做日志文件（multiplexed redo log）。Oracle 保存所有数据库事务的日志，这些事务被记录在联机重做日志文件中。当数据库中的数据遭到破坏时，可以用这些日志来恢复数据库。

重做日志文件记录数据库所做的全部变更（如增加、删除和修改），以便在系统发生

故障时，用它对数据库进行恢复。名字通常为 log*.dbf 格式。

根据在事务信息将被覆盖时，是否应该将文件归档，数据库分为以下两种归档模式：ARCHIVELOG（归档日志）模式和 NOARCHIVELOG（非归档日志）模式。

一个数据库至少需要两个重做日志文件。假定目前 Oracle9i 数据库拥有两个重做日志文件，分别是 Redo01.ora 与 Redo02.ora。因为重做日志文件是以循环方式使用，所以当 Redo01.ora 写满后就会轮到 Redo02.ora，这个动作称为日志切换。理论上，当 Redo02.ora 写满后应该会立刻从 Redo01.ora 重新开始写入。但实际上并非如此，Oracle9i 会等到 Redo01.ora 内记录的数据异动情况真正反应在数据文件之后，才会覆盖 Redo01.ora 中的数据。在一个大型联机事务处理（OLTP）系统下，如果数据交换十分频繁，建议不妨配置 5 个以上的重做日志文件，以免 Oracle9i 进行日志切换时还要等候 CKPT 进程执行检查点操作，从而影响系统效能。

在执行关键性任务的数据库系统中，为了避免硬件问题导致重做日志文件毁损（例如硬盘产生坏道），建议考虑设置多重重做日志文件。多重重做日志文件其实就是在其他硬盘内同时存放多份重做日志文件的复本。以图 2.5 为例，假定原来的重做日志文件为 Redo_A01.ora 与 Redo_A02.ora。作者在第二个磁盘又额外规划了两个重做日志文件：Redo_B01.ora 与 Redo_B02.ora（分别为 Redo_A01.ora 与 Redo_A02.ora 的复本）。这时 Redo_A01.ora 与 Redo_B01.ora 就构成所谓的重做日志群组（redo log group）；而 Redo_A01.ora 与 Redo_B01.ora 分别为该群组的成员（member）。

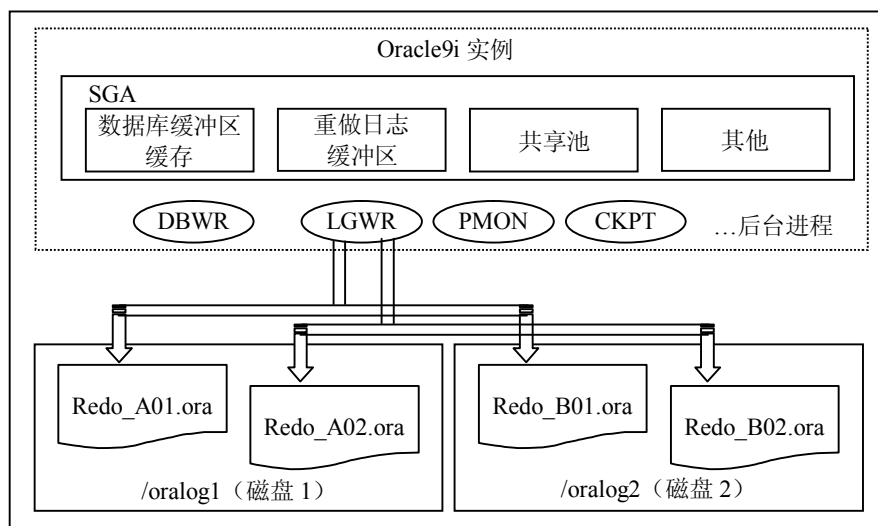


图 2.5 多重重做日志文件的配置方式

一旦以重做日志群组的方式规划重做日志文件，LGWR 会同时将变动记录写入相同群组的所有成员。也就是说，LGWR 会将数据同时写入 Redo_A01.ora 与 Redo_B01.ora，等到发生日志切换操作后，再同时写入 Redo_A02.ora 与 Redo_B02.ora。可想而知，万一这两个硬盘中的一个不幸损毁，重做日志文件将不受任何影响。

尽管多重重做日志文件的概念已具备容错功能，但是对于重做日志文件的写入效率却有一定程度的影响！这也是我们在规划之初就必须考虑的重点之一。

2.4.3 控制文件

每个 Oracle9i 数据库都有一个控制文件（名字通常为 Ctr*.ctl 格式），用以记录与描述数据库的物理结构。它们包括：

- (1) Oracle9i 数据库名称与建立时间。
- (2) 数据文件与重做日志文件名称及其所在位置。
- (3) 日志记录序列号（log sequence number）。
- (4) 数据库检查点信息。

每当数据库被激活时，Oracle9i 会在实例激活后立刻读取控制文件的内容，待所有数据库外部结构文件的信息都收集完毕后，数据库才会启动。为了避免控制文件毁损导致数据库系统停止，建议至少配置两个控制文件，并存放在不同的硬盘上。

控制文件中的内容只能由 Oracle 本身来修改。每个数据库必须至少拥有一个控制文件。一个数据库也可以同时拥有多个控制文件，但是一个控制文件只能属于一个数据库。从前面的介绍中我们知道，数据库的数据文件和重做日志文件均是一些操作系统文件，Oracle 在操作数据库时，怎样将指定数据库中的这些文件同普通的操作系统文件和其他数据库文件区分开来，这正是数据库控制文件的功能。对于 Oracle 数据库来说，如果把数据文件比作一个仓库，那么重做日志文件就相当于该仓库的货物进出记录账本，控制文件就是该仓库的管理中心。

2.4.4 配置文件

配置文件记录 Oracle 数据库运行时的一些重要参数。名字通常为 initSID*.ora 格式，其中 SID 相当于它所控制的数据库的标识符。

1. 服务器参数文件

Oracle9i 引入一个服务器参数文件（SPFILE），它是在 Oracle 数据库服务器端的初始化参数文件。存储在一个 SPFILE 中的初始化参数是永久的，它提供了由 Oracle 数据库服务器自我调节的一个基础。SPFILE 文件是二进制文件，不能使用文本编辑器浏览或编辑。Oracle 提供了浏览和查看相关参数的另外接口。

2. 数据库参数文件

初始化数据库参数文件记录数据库实例分配参数，它相当于 Oracle 实例引导文件，只有在它的引导下，Oracle 实例才能正确创建。

初始化参数文件包含以下主要内容：

- (1) 系统全局区相关的组态设定。
- (2) 其他各起始参数的名称与设定值。
- (3) 控制文件名称与存放路径。
- (4) 重做日志文件的相关设定。

可以利用任何文字编辑器来建立或修改初始化数据库参数文件，通常初始化数据库参数文件的名称为 `init.ora`。每个数据库和实例都有它自己惟一的 `init.ora` 文件。假定 Oracle9i 数据库实例的名称为 `student`（学生管理数据库），则参数文件存放在 `Oracle_Home\admin\student` 目录下（`Oracle_Home` 表示 Oracle 的根目录）。

初始化参数文件不仅控制 Oracle 实例的启动，其中的很多参数还影响系统的性能。合理设置这些参数，能够达到优化系统运行性能的目的。所以，数据库管理员要全面了解数据库初始化参数文件中各个参数的作用及取值范围，并根据实际环境情况进行设置，修改参数值。

2.5 Oracle 数据库逻辑存储结构

Oracle 数据库在逻辑上由许多表空间构成，主要分为系统表空间和非系统表空间。非系统表空间内存储着各项应用的数据、索引和程序等相关信息。当我们需要开发一个较大的 Oracle 应用系统时，应该创建它所独占的表空间，同时定义物理文件的存放路径和所占硬盘的大小。如图 2.6 所示为 Oracle 数据库逻辑结构与物理结构的对照关系。

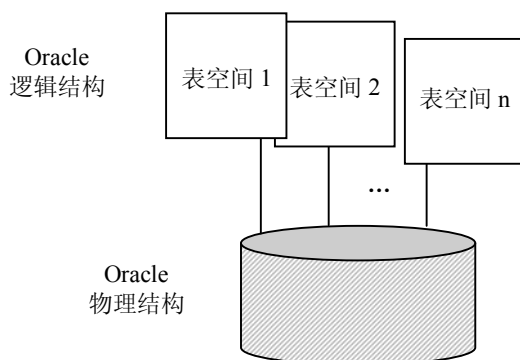


图 2.6 Oracle 数据库逻辑结构与物理结构的对照关系

Oracle 数据库的逻辑存储结构如表空间（Tablespace）、段（Segment）、区间（Extent）和数据块（Data Block）将支配一个数据库的物理空间如何使用。

2.5.1 表空间（Tablespace）

1. 表空间的分类

Oracle 中除了用户创建的存放数据对象的数据表空间外，还有以下类型的表空间：

- (1) 系统表空间（System TableSpace）。
- (2) 回退表空间（RollBack TableSpace）。
- (3) 临时表空间（Temp TableSpace）。
- (4) 用户表空间（User TableSpace）。
- (5) 索引表空间（Index TableSpace）。

2. 表空间的使用

在数据库中使用多个表空间具有如下优势：

(1) 能够将数据字典与用户数据分离开来，避免由于字典对象和用户对象保存在同一个数据文件中而产生的 I/O 冲突。

(2) 能够将回退数据与用户数据分离开来，避免由于硬盘损坏而导致永久性的数据丢失。

(3) 能够将表空间的数据文件分散保存到不同的硬盘上，平均分布物理 I/O 操作。

(4) 能够将某个表空间设置为脱机状态或联机状态，以便对数据库的一部分进行备份和恢复。

(5) 能够将某个表空间设置为只读状态，从而将数据库的一部分设置为只读状态。

(6) 能够为某种特殊用途专门设置一个表空间，例如临时表空间等，以优化表空间的使用效率。

(7) 能够更加灵活地为用户设置表空间限额。

3. 表空间中对象的存储

如果表空间只对应一个数据文件，则该表空间中所有的对象都存储在此数据文件中。

如果表空间对应于多个数据文件，可将一个对象的数据存储在该表空间的任意一个数据文件中，也可将同一个对象的数据分布在表空间的多个数据文件中。

Oracle 数据库的物理存储结构是由数据文件组成的，但在逻辑存储上，Oracle 将数据库中所有数据文件所占用的磁盘空间划分为一个或多个表空间进行存储管理。一个表空间可以包含数据库的多个数据文件，但一个数据文件只能属于一个表空间，而不能同时属于多个表空间。

Oracle9i 数据库允许 DBA 自行规划各种用途的表空间，以便存放不同类型的数据与数据库对象。举例来说，存放 Oracle 系统信息的数据字典可放在 SYSTEM 表空间内；应用系统数据可存放在 USER 表空间中；而所有数据表的索引则可单独存放在 INDEX 数据表空间内。

一般来说，Oracle9i 数据库至少应规划出 5 个以上的表空间，如图 2.7 所示。

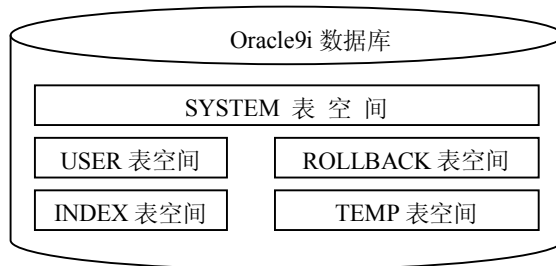


图 2.7 Oracle9i 数据库的表空间

2.5.2 段 (Segment)

表空间的下一级逻辑存储单位称作段，一个段只能存储同一种模式对象。根据段中所

存储的模式对象不同，段分为以下几类：

(1) 数据段。数据段中保存的是数据表中的记录。当用户建立表或聚簇时，Oracle 自动建立数据段。

(2) 索引段。在 Oracle 数据库中每个未分区索引都有一个索引段保存索引中的索引条目。对于一个分区索引，每个分区都有一个索引段保存它的数据。当执行 CREATE INDEX 语句建立索引或索引分区时，Oracle 自动创建索引段。

(3) 临时段。在执行查询、排序、创建索引等操作时，Oracle 使用临时段保存 SQL 语句解释和执行过程中所产生的临时数据。例如，当处理查询时，Oracle 可能会需要使用一些临时存储空间，用于临时保存解析过的查询语句以及在排序过程中产生的临时数据。Oracle 会自动在存储临时数据的表空间中，为操作分配临时段。

(4) 回退段。回退段用于存放数据修改之前的值（包括数据修改之前的位置和值）。即记录数据库中所有事务修改前的数据值，这些数据用于读一致性、回退事务和恢复数据库等操作。

2.5.3 区间 (Extent)

区间是由很多连续的数据块组成的数据库存储空间。Oracle9i 按需要以区间为单位为段分配空间。那么，使用区间究竟有何好处呢？当执行 DDL 指令建立数据库对象时（例如 CREATE TABLE 指令），Oracle9i 会先分配出一连串的数据块；直到这些数据块全部存满数据后，下次再分配一连串的数据块。这些连续的数据块就统称为区间。

举例来说，假定准备在 Oracle9i 数据库内存放 80 KB 的数据，Oracle9i 可以先分配一个区间，其大小可为数据块（8 KB）的 5 倍；然后视实际需要再分配第二个区间（大小还是 40 KB）。如此一来，只要两次空间分配动作就可以存放 80 KB 的数据了。试想，如果 Oracle9i 每次只分配 8 KB 的数据块来存放数据，总共需要 10 次空间分配操作才能放满 80 KB 的数据，效率就会比较低。由此可见，区间的确有其存在的必要性。

缺省情况下，数据库会使用表空间中存储的存储参数来管理他们的区间。其中，主要的几个存储参数为：

- INITIAL：段建立时分配的第一个区间的大小（单位：字节）。
- NEXT：段内分配的下一个增量区间大小（单位：字节）。
- MAXEXTENTS：分配给段的区间总数。
- MINEXTENTS：建立段时分配的区的总数，也就是段的最小区间数量。
- PCTINCREASE：为段分配的每个增量超过上一个区间的百分比。
- INITRANS：为初试数量的 DML 事务条目（并发访问数据块中的行）保留的预分配空间数量。
- NTKXTRANS：当多个事务同时访问同一数据块的行时，为数据块中每个 DML 事务的条目分配的空间。

2.5.4 数据块 (Data Block)

数据块也称逻辑块或 Oracle 块，它对应磁盘上一个或多个物理块。可以用 PCTFREE 和 PCTUSED 两个参数来优化数据块空间的使用。

- PCTFREE: 块中保留用于 UPDATE 操作的空间百分比，当数据占用的空间达到此上限时，新的数据将不能再插入到此块中。
- PCTUSED: 指定块中数据使用空间的最低百分比。

Oracle9i 数据库中最小的存储单元就称为数据块。数据块是在创建 Oracle9i 数据库之前就必须明确定义。

截至目前为止，无论是数据块、区间、段，甚至数据表空间，都只是逻辑上的数据存储概念。图 2.8 说明了 Oracle9i 数据库内各逻辑组成部分之间的阶层关系。

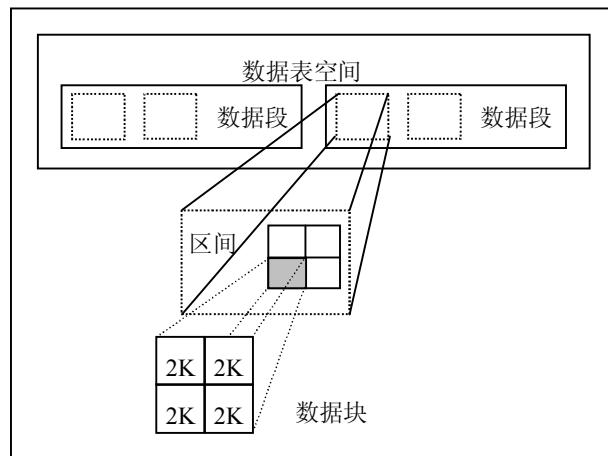


图 2.8 Tablespace、Segment、Extent、Data Block 之间的关系

从 Oracle9i 数据库的逻辑存储结构来看，我们可以总结如下：

(1) 数据是存放在每个数据块内，数据读写动作也将以数据块为单位。数据块是最小的存储单元，Oracle 常用的数据块大小是 2KB 或 4KB。

(2) 空间分配的操作则是引用区间的概念。一个区间由一连串的数据块所构成。

(3) 依实际数据量的多少，数据（或数据库对象）由多个区间所组成。

(4) 数据表空间内可划分不同的段，而段是由许多区间组合而成的。当段中的空间用完时，该段就获取另外的区间。由于段是一个物理实体，所以必须把它分配到数据库中的一个表空间中（放在表空间的一个数据文件中）。

(5) 在每个 Oracle9i 数据库内可规划不同的数据表空间，用以存放不同型态的数据。依照不同的数据处理性质，需要在数据表空间内划分出不同的区域，以存放不同的数据，这些区域称作“段”。例如，存放数据的区域称为“数据区段”；存放索引区域的称为“索引区段”。

2.6 Oracle9i 数据库简介

2.6.1 Oracle9i 数据库概述

Oracle9i 是一种关系数据库管理系统 (RDBMS)，关系数据库是按照二维表结构方式组织的数据集合。在应用数据库之前，熟悉 Oracle9i 数据库的基本对象，以确保数据库能很好地运行是很重要的。这里我们预先介绍 Oracle9i 中涉及的一些基本概念，为下面章节的学习做好准备。

数据库 (Database) 是一个数据容器，它包含了表、索引、视图、簇、过程、函数、包等对象，并对其进行统一的管理。用户只有和一个确定的数据库连接，才能使用和管理数据库中的数据。

1. 表空间 (TABLESPACE)

表空间是数据库的逻辑划分，每个数据库至少有一个表空间 (称作 SYSTEM 表空间)，每个表空间由一个或多个数据文件组成，表空间中其他逻辑结构的数据存储在这些数据文件中。

为便于管理和提高运行效率，可以使用一些附加表空间来划分用户和应用程序。例如，用户表空间供一般用户使用，回退表空间供回退段使用。

2. 表 (TABLE)

表是数据库中存放用户数据的对象，它包含一组固定的列。表中的列描述该表所跟踪的实体的属性。每个列都有一个名字和若干个属性。

3. 约束条件 (CONSTRAINT)

可以为一个表列创建约束条件，此时，表中的每一行都必须满足约束条件所规定的条件。约束条件有以下 5 种：

(1) 主键 (Primary key)：主键是表中的一列或多个列。为表定义主键有几个作用，主键包含的列不能输入重复的值，以保证一个表的所有行的惟一性；主键也不允许定义此约束的列为 NULL 值；主键在定义此约束的列中创建了惟一性的索引，利用这个索引可更快地检索表中的行。

(2) 默认 (DEFAULT) 约束条件：在表中插入一行数据但没有为列指定值时生成一个在定义表时预先指定的值。

(3) 检查 (CHECK) 约束条件：该约束条件确保指定列中的值符合一定的条件。CHECK 列约束条件不能引用一个独立表。非空值约束条件被数据库看作是一个 CHECK 约束条件。

(4) 惟一性 (UNIQUE) 约束条件：用于保证不是主键的那些列的惟一性。

(5) 外键 (FOREIGN KEY) 约束条件：该约束条件规定表之间的关系性质，一个外键使一个表的一列或多列与已定义为主键的表中一批相同的列相关联。当在已定义主键约束的表中更新列值时，其他表中定义有外键约束的列会被自动更新。

主键约束和外键约束保证关联表的相应行持续匹配，以便它们可以用在后面的关系连接中。在它们被定义为主键约束和外键约束后，不同表的相应列会自动更新，称为引用完

整性声明。

数据库的约束条件有助于确保数据的引用完整性。引用完整性保证数据库中的所有列引用都有效且全部约束条件都得到满足。

4. 分区 (PARTITION)

在非常大的数据库中,可以通过把一个大表的数据分成多个小表来简化数据库的管理。这些小表称作分区。除了对表进行分区外,还可以对索引进行分区。分区不仅简化了数据库的管理,还改善了数据库的应用性能。

在 Oracle9i 中,能够细分分区,创建子分区。例如,可以根据一组值分割一个表,然后再根据另一种分割方法分割分区。

5. 索引 (INDEX)

在关系数据库表中,一个行数据的物理位置无关紧要。为了能够找到数据,表中的每一行都用一个 RowID 来标识。RowID 告诉数据库这一行的准确位置,包括所在的文件、该文件中的块和该块中的行地址。

索引是帮助用户在表中快速地找到记录的数据库结构。它既可以提高数据库性能,又能够保证列值的惟一性。当 CREATE TABLE 命令中规定有 UNIQUE 或 PRIMARY KEY 约束条件子句时,Oracle 就会自动创建一个索引,也可以通过 CREATE INDEX 命令手工创建自己的索引。

注意: 索引结构表没有传统的 Oracle RowID,不过,其主键起一个逻辑 RowID 的作用。

6. 簇 (CLUSTER)

经常被同时访问的表在物理位置上可以存储在一起。为了将它们存储在一起,就要创建一个簇来管理这些表。表中相关的列称为簇键 (CLUSTER KEY)。簇键用一个簇索引来进行索引,对于簇中的多个表,簇键值只存储一次。在把任何行插入簇的表中之前,都必须先创建一个簇索引。

簇的另一种形式是散列簇。为确定行应存储的物理位置,对行的簇键使用散列函数 (Hashing function)。散列簇可以大大提高等值查询的效率。

7. 用户 (USER)

用户账号虽然不是数据库中的一个物理结构,但它与数据库中的对象有着重要的关系,这是因为用户拥有数据库的对象。例如,用户 SYS 拥有数据字典表,这些表中存储了数据库中其他对象的所有信息;用户 SYSTEM 拥有访问数据字典表的视图,这些视图供数据库其他用户使用。

为数据库创建对象(例如表)必须在用户账号下进行。可以对每一个用户账号进行自定义,以便将一个特定的表空间作为它的默认表空间。

把操作系统的账号和数据库账号联系在一起,这样可以不必既输入操作系统口令又输入数据库的口令。

8. 方案 (SCHEMA)

用户账号拥有的对象集称为用户的方案。可以创建不能注册到数据库的用户账号。

这样的用户账号提供一种方案，这种方案可以用来保存一组被其他用户方案分开的数据对象。

9. 同义词 (SYNONYM)

为了给不同的用户使用数据库对象时提供一个简单的、惟一标识数据库对象的名称，可以为数据库对象创建同义词。同义词有公用同义词和私有同义词两种。

10. 权限及角色 (ROLE)

为了访问其他账号所有的对象，必须首先被授予访问这个对象的权限。权限可以授予给某个用户或 PUBLIC，PUBLIC 把权限授予数据库中的全体用户。

可以创建角色即权限组来简化权限的管理。可以把一些权限授予一个角色，而这个角色又可以被授予多个用户。在应用程序中角色可以被动态地启用或禁用。

2.6.2 本书所涉及的样本数据库

1. Oracle9i 的数据类型

Oracle9i 数据库通常包含多个表，表是一个存储数据的实体，具有惟一的名称。可以说数据库实际上是表的集合，具体的数据或者信息都是存储在表中的。表是对数据进行存储和操作的一种逻辑结构，每一个表代表一个对象。例如，学生管理数据库中，会有学生基本信息表、教师基本信息表、学生成绩表、课程信息表、专业代码表、教研室信息表、课室信息表等。例如，学生成绩表就是由行和列组成的，通过表名和列名来识别数据。表中各列包含了列名、数据类型以及列的其他属性，行包含了若干列的数据。下面我们就介绍 Oracle9i 数据库表中的数据类型。

(1) 字符型数据类型。

1) CHAR (size)。存放固定长度的字符串，size 表示的是字符数据的最大宽度，其宽度为 1~2000 个字节，对应 1~2000 个单字节字符或 1~1000 个双字节字符（如汉字），如果不指定长度，缺省为 1 个字节长。CHAR 字符类型数据在声明时要设定其最大宽度。

2) NCHAR (size)。基于所使用语种的字符集的字符型数据类型。其他与 CHAR 数据类型相同。

3) VARCHAR2 (size)。表示可变长度字符串，其宽度范围为 1~4000 个字节，在使用时要声明其最大宽度 size。这种数据类型可以存放数字、字母以及 ASCII 码字符集（或者 EBCDIC 等数据库系统接受的字符集标准）中的所有符号。

如果数据长度没有达到最大值 n，Oracle9i 会根据数据大小自动调节列长度；如果数据前后有空格，Oracle9i 会自动将其删去。VARCHAR2 是最常用的数据类型。

4) VARCHAR (size)。目前 Oracle9i 版本中 VARCHAR 类型与 VARCHAR2 类型含义完全相同，而且 VARCHAR 将会独立成一种新的数据类型，不再受宽度的限制。

5) LONG。字符数据类型，用来存储变长字符串，最大长度限制是 2GB。例如在数据字典中，存放视图定义的正文就是使用了 LONG 数据类型。

这种数据类型无参数，用于不需要作字符串搜索的长串数据。如果要进行字符搜索就需要使用 VARCHAR2 类型。

LONG 是一种较老的数据类型，会逐渐被 BLOB、CLOB、NCLOB 等大的对象数据类型所取代。

(2) 数值类型数据。

NUMBER (m,n)。用来存储整数和浮点数，m 表示所有有效数字的位数，n 代表小数点以后的位数，n 缺省表示小数位为 0。如果实际数据超出设定精度则出现错误提示。

如：number (5,2) 允许数据的最大值是 99,999，数据超出了有效位数限制就会被截取多余的位数。例如在一行数据中的这个列输入 575.316，则真正保存到列中的数据是 575.32。又如：number (3,0)，输入 575.316，真正保存的数据是 575。

(3) 日期和时间类型数据。

DATE 这种数据类型无参数，用来存放日期和时间类型的数据，用 7 个字节分别描述年、月、日、时、分、秒。其日期可以表达公元前 4712 年 1 月 1 日到公元 4712 年 12 月 31 日的所有合法日期，缺省格式为 DD-MON-YY 分别对应日-月-年，例如 17-JUN-04。注意，月份的表达要用英文单词的缩写格式。日期的格式可以设置为中文格式，例如 17-06 月-04，表示 2004 年 6 月 17 日。

(4) 逻辑型变量。BOOLEAN (布尔) 逻辑型变量的值只有两个，true (真) 或 false (假)。逻辑类型变量在使用前先判断它的状态，然后根据其值是“真”或“假”来决定程序执行的逻辑。关系表达式的值就是一个逻辑变量。

(5) 二进制数据类型。

1) RAW (size)。RAW (size) 用来描述可变长二进制数据，在具体定义列的时候必须指明最大长度 size，Oracle9i 用这种格式来保存较小的图形文件或带格式的文本文件，如 Microsoft Word 文档。其优点在于：当数据在不同系统之间传输时，可以不做任何转换，方便了系统之间的操作。除此之外，RAW 和 VARCHAR2 类型没什么不同。RAW 类型最大宽度为 2000B。

RAW (size) 是一种较老的数据类型，会逐渐被 BLOB、CLOB、NCLOB 等大的对象数据类型所取代。

2) LONG RAW。这种数据类型无参数，用来描述可变长二进制数据，最大长度是 2GB。Oracle9i 用这种格式来保存较大的图形文件或带格式的文本文件，如 Microsoft Word 文档，以及音频、视频等非文本文件。与 RAW 的区别是它的最大宽度为 2GB，可用来存储图像、视频、音频等数据量大的数据，并可方便地在网上传输。

在同一张表中不能同时有 LONG 类型和 LONG RAW 类型。同样 LONG RAW 也是一种较老的数据类型，会逐渐被 BLOB、CLOB、NCLOB 等大的对象数据类型所取代。

(6) 大型对象数据类型。

大型对象可以包含没有结构特征的数据，对它的访问将比 LONG 或 LONG RAW 数据的访问更有效，对它的限制更少。大型对象 (LOB) 有几种类型，取决于你使用的字节的类型，Oracle9i 实实在在地将这些数据存储于数据库内部保存。可以对其执行读取、存储和写入等特殊操作。大型对象数据可分为以下几类：

1) BLOB 二进制大数据集，可存放数据宽度达 4GB。

2) CLOB 包含单个字符的大数据集，最大宽度为 4GB。

3) NCLOB 包含定宽的多位数字符的大的字符集，最大宽度为 4GB。

这三种大型对象 (LOB)，都可以用来保存较大的图形文件或带格式的文本文件，如 Microsoft Word 文档，以及音频、视频等非文本文件。

此外 Oracle9i 还支持一些其他数据类型。例如 BFILE，这种数据类型无参数，在数据库外部保存的大型二进制对象文件，最大长度也是 4GB。这种外部的 LOB 类型，通过数据库记录变化情况，但是数据的具体保存是在数据库外部进行的。Oracle9i 可以读取、查询 BFILE，但是不能写入。大小由操作系统决定。

2. 学生管理数据库 (student) 表结构

表由若干个列组成，每一个列的列名在表中应该是惟一的；其列值应该为 Oracle9i 可选的数据类型；大小则规定列值的最大长度；小数位用于定义数值型 NUMBER 列的小数位数；是否为空说明列值是否允许为空值；默认值用于定义不经输入所确定的列值。我们针对上述数据表作了如下设计，以帮助同学们创建数据表时参考使用。

(1) 学生基本信息表 (stud_info) 的结构设计。

序号	列名	数据类型	大小	小数位	是否为空	默认值	列名含义
1	STUD_ID	CHAR	10		N	0401010101	学生编号
2	NAME	VARCHAR2	8		N		姓名
3	BIRTHDAY	DATE					出生日期
4	GENDER	CHAR	2				性别
5	ADDRESS	NVARCHAR2	20				家庭地址
6	TELCODE	CHAR	12				电话
7	ZIPCODE	CHAR	6				邮编
8	MARK	NUMBER	3	0		560	入学成绩

(2) 教师基本信息表 (teacher_info) 的结构设计。

序号	列名	数据类型	大小	小数位	是否为空	默认值	列名含义
1	TEACHER_ID	CHAR	6		N	010101	教师编号
2	NAME	VARCHAR2	8		N		姓名
3	GENDER	CHAR	2				性别
4	AGE	NUMBER	2				年龄
5	TECH_TITLE	CHAR	10				职称
6	TELEPHONE	VARCHAR2	12				电话
7	SALARY	NUMBER	7	2		418	工资
8	COURSE_ID	CHAR	10				任教课程编码

(3) 学生成绩表 (stud_grade) 的结构设计。

序号	列名	数据类型	大小	小数位	是否为空	默认值	列名含义
1	STUD_ID	CHAR	10		N	0401010101	学生编号
2	NAME	VARCHAR2	8		N		姓名
3	COURSE_ID	CHAR	10				课程号
4	Grade	NUMBER	4	1			成绩

(4) 课程信息表 (lesson_info) 的结构设计。

序号	列名	数据类型	大小	小数位	是否为空	默认值	列名含义
1	COURSE_ID	CHAR	10		N	0401010101	课程号
2	COURSE_NAME	VARCHAR2 ()	24		N		课程名
3	COURSE_TYPE	CHAR	4		N	考试	课程类型
4	COURSE_MARK	NUMBER	3		N		课程学分
5	COURSE_TIME	NUMBER	3	1			课程学时

(5) 课程安排表 (teach_schedule, 课表) 的结构设计。

序号	列名	数据类型	大小	小数位	是否为空	默认值	列名含义
1	COURSE_ID	CHAR	10		N		课程号
2	COURSE_TIME	DATE					起始时间
3	COURSE_WEEK	NUMBER	2	0			教学周数
4	ROOM_ID	CHAR	6				教室
5	DEPTCODE	CHAR	2				院系编码
6	TEACHER_ID	CHAR	6				教师编号

(6) 院系代码表 (dept_code) 的结构设计。

序号	列名	数据类型	大小	小数位	是否为空	默认值	列名含义
1	DEPTCODE	CHAR	2		N	01	院系编号
2	DEPTNAME	VARCHAR2	20				院系名称

(7) 专业代码表 (specialty_code) 的结构设计。

序号	列名	数据类型	大小	小数位	是否为空	默认值	列名含义
1	SPECCODE	CHAR	6		N	040101	专业编号
2	SPECNAME	VARCHAR2	20				专业名称

(8) 教研室信息表 (staffroom_info) 的结构设计。

序号	列名	数据类型	大小	小数位	是否为空	默认值	列名含义
1	JYSH_ID	CHAR	4		N	0101	教研室编码
2	JYSH_NAME	VARCHAR2	20				教研室名称
3	JYSH_TYPE	CHAR	4			专业	教研室类型
4	JYSH_LEADER	VARCHAR2	8				教研室主任

(9) 教室信息表 (classroom_info) 的结构设计。

序号	列名	数据类型	大小	小数位	是否为空	默认值	列名含义
1	ROOM_ID	CHAR	6		N	010101	教室编码
2	ROOM_NAME	VARCHAR2	8				教室名称
3	ROOM_TYPE	VARCHAR2	10			普通	教室类型
4	ROOM_DEVICE	VARCHAR2	20			投影仪	教室设备
5	ROOM_SIZE	NUMBER	3	0		60	教室容量

3. 学生管理数据库表数据的样本

(1) 学生基本信息表 (stud_info):

STUD_ID	NAME	BIRTHDAY	GENDER	ADDRESS	TELCODE	ZIPCODE	MARK
0401010811	张源	12-5月-1986	男	北京市海淀区	010-64572345	100080	560
0401020201	赵明	08-6月-1986	男	上海市浦东区	021-64897232	201700	560
0401030101	王刚	01-2月-1986	男	天津市南开区	022-83572336	300000	560
0401050201	陈红	25-10月-1986	女	武汉市汉口区	027-74565350	430000	560
0401040112	孙强	06-7月-1986	男	重庆市沙坪坝	023-82342348	400000	560
0401020103	李伟	09-1月-1986	男	北京市大兴区	010-68902947	102600	560
0401010634	钱昆	12-6月-1986	男	广州市海珠区	020-45672453	510000	560
0401030213	郑芳	08-9月-1986	女	江苏省南京市	025-84573941	210000	560
0401010102	袁飞	03-11月-1986	男	湖南省长沙县	0731-4572345	410000	560
0401040123	孔荣	31-5月-1986	男	云南省昆明市	0871-2647535	650000	600
0401050127	张军	11-3月-1987	男	成都市和平区	028_85249567	610041	570
0401010704	王一明	03-3月-1986	男	甘肃省兰州市	0621-2221837	590000	573

其中：学号 STUD_ID 的 10 位数字说明→年级 (2 位)+院系编码 (2 位)+专业编码 (2 位)+班次 (2 位)+顺序号 (2 位)。例如，“0401010101”之中第 1~2 位的“04”表示 2004 级，第 3~4 位的“01”表示计算机工程系，第 5~6 位的“01”表示计算机应用技术专业，第 7~8 位的“01”表示班次编号，第 9~10 位的“01”表示张源同学的顺序号。

(2) 教师基本信息表 (teacher_info):

TEACHER_ID	NAME	GENDER	AGE	TECH_TITLE	TELEPHONE	SALARY	COURSE_ID
010101	刘娜	女	34	讲师	020-34601254	418	0401010102
010102	邵云鹏	男	45	教授	020-34607811	458	0401010105
010103	孙乐多	男	27	助教	020-34602622	380	0401010106
010104	赵一欧	女	26	助教	020-34603472	380	0401010101
010106	王吉林	男	32	讲师	020-34603232	418	0401010103
010105	王小悦	女	35	讲师	020-34603790	448	0401010104

其中: TEACHER_ID 的 6 位说明→院系编码 (2) +教研室编码 (2) +顺序号 (2)。
例如, “010101” →计算机工程系编码 (01) +计算机应用教研室编码 (01) +刘娜老师的顺序号 (01)。

(3) 学生成绩表 (stud_grade):

STUD_ID 学号	NAME 姓名	Course_ID 课程号	Grade 成绩
0401010811	张源	0401010102	90
0401020201	赵明	0401010105	89
0401030101	王刚	0401010103	87
0401050201	陈红	0401010101	91
0401040112	孙强	0401010104	83
0401020103	李伟	0401010106	86
0401010634	钱昆	0401010106	78
0401030213	郑芳	0401010106	95
0401010102	袁飞	0401010106	95
0401040123	孔荣	0401010106	83
0401050127	张军	0401010106	84

(4) 课程信息表 (lesson_info):

Course_ID (课程号)	Course_name (课程名)	Course_type (课程类型)	Course_mark (课程学分)	Course_time (课程学时)
0401010101	计算机专业英语	考查	1.5	30
0401010102	Java 程序设计	考试	3.5	60
0401010103	微型计算机原理	考试	3.5	60
0401010104	IT 市场营销	考查	1.5	30
0401010105	网络互联设备与配置	考查	3.5	60
0401010106	多媒体技术	考查	3	60

其中：课程号（COURSE_ID）10位数字的详细说明如下。

第1~2位	第3~4位	第5~6位	第7~8位	第9~10位
年级编号	院系编号	专业编号	学期编号	课程编号
“04”	“01”	“01”	“01”	“01”
04级	计算机系	计算机应用	第1学期	计算机导论

(5) 课程安排表（teach_schedule）:

COURSE_ID (课程号)	COURSE_TIME (起始时间)	COURSE_WEEK (使用周数)	ROOM_ID (教室)	DEPTCODE (院系编码)	TEACHER_ID (教师编号)
0401010101	30-8月-2004	15	120703	01	010104
0401010102	30-8月-2004	15	120704	01	010101
0401010103	30-8月-2004	13	120705	01	010106
0401010104	30-8月-2004	10	120706	01	010105
0401010105	30-8月-2004	19	120707	01	010102
0401010106	30-8月-2004	14	120708	01	010103

(6) 院系代码表（dept_code）:

DEPTCODE (院系编号)	DEPTname (院系名称)
01	计算机工程系
02	管理工程系
03	机电工程系
04	食品工程系
05	轻化工程系
06	通信工程系
07	外语工程系

(7) 专业代码表（specialty_code）:

SPECcode (专业编号)	SPECname (专业名称)
040101	计算机应用技术
040102	计算机网络技术
040103	计算机控制技术
040104	多媒体技术
040105	计算机软件技术
040106	计算机通信技术
040107	计算机管理技术

其中:SPECcode 的 6 位构成→学年(2)+院系编码(2)+专业顺序号(2),例如,“040101”之中第 1~2 位的“04”表示 2004 年,第 3~4 位的“01”表示计算机工程系,第 5~6 位的“01”表示计算机应用技术专业的顺序号。

(8) 教研室信息表 (staffroom_info):

JYSH_ID (教研室编码)	JYSH_NAME (教研室名称)	JYSH_TYPE (教研室类型)	JYSH_LEADER (教研室主任)
0101	计算机应用	专业	王二毛
0102	计算机网络	专业	李四冲
0103	计算机软件	专业	赵一生
0104	计算机管理	专业	汪三洋

其中:JYSH_ID 的 4 位组成→院系编码(2)+教研室编码(2),例如,“0101”→计算机工程系编码(01)+计算机应用教研室编码(01)。

(9) 教室信息表 (classroom_info):

ROOM_ID (教室编码)	ROOM_NAME (教室名称)	ROOM_TYPE (教室类型)	ROOM_DEVICE (教室设备)	ROOM_SIZE (教室容量)
120703	微机组装与维护	实训	微机、投影仪	40
120704	计算机网络	实验	交换机、路由器等	40
120705	数据库	计算机机房	微机、投影仪	60
120706	软件设计	计算机机房	微机、投影仪	60
120707	多媒体	计算机机房	微机、投影仪	60
120708		普通	白板、投影仪	120

其中:ROOM_ID(教室编码)的 6 位组成为 2 位楼栋号、2 位楼层号、2 位教室顺序号。例如,“120708”表示 12 栋 7 楼第 8 号教室。

上述 9 个表可以用图 2.9 来描述它们之间的相互关系。

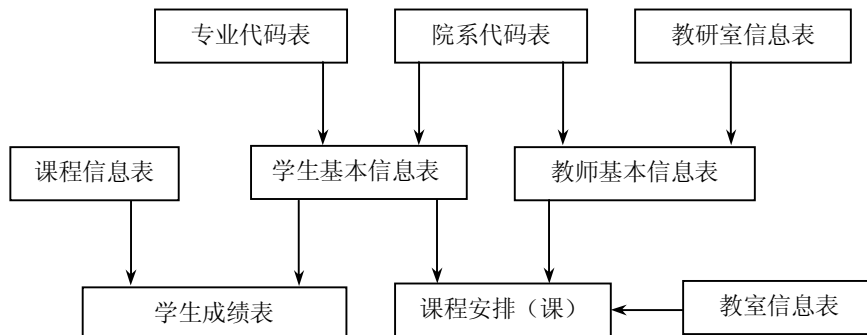


图 2.9 学生管理中数据表及它们之间的相互关系

课后习题

1. 简要回答下列问题。

- (1) 什么是 Oracle 的实例？它是一种什么机制？
- (2) 图示 Oracle 系统的构成（内存结构、主要的后台进程和物理数据库来描述）。
- (3) 图示 SGA 的主要组成。简述各部分的内容。
- (4) Oracle 的进程主要有哪几类？服务器进程的作用是什么？
- (5) 读写日志文件和数据库的进程是什么？
- (6) 段是什么概念？如何组成的？
- (7) Oracle 的常用表空间有哪些？
- (8) 如果 Oracle 的表空间满了，会出现什么问题？如何解决？
- (9) 为什么要给用户指定默认表空间和临时表空间？
- (10) 数据类型中 CHAR 和 VARCHAR2 有什么区别？

2. 选择最适合的答案。

- (1) 从下列选项中选择一个无效的数据类型（ ）。
A. CLOB B. VARCHAR2 C. TIME D. RAW
- (2) 每个数据库可以有几个控制文件：
A. 3 个 B. 7 个 C. 8 个 D. 12 个
- (3) 每个数据库至少要有几个重做日志文件？（ ）
A. 1 个 B. 2 个 C. 3 个 D. 任意个
- (4) 下列哪个文件用来记录应用程序对数据库的改变？（ ）
A. 数据文件 B. 控制文件 C. INIT.ORA D. 重做日志文件
- (5) 控制文件包含哪些信息？（ ）
A. 实例名称 B. 数据文件名称
C. 表空间名称 D. 重做日志文件名称
- (6) 当数据库创建时，哪个表空间将会自动生成？（ ）
A. USER 表空间 B. TOOLS 表空间
C. TEMP 表空间 D. SYSTEM 表空间
- (7) 关于表空间与数据文件的关系，下列哪些描述是对的？（ ）
A. 每个表空间至少含有一个数据文件
B. 一个表空间属于一个数据文件
C. 一个数据文件可以属于多个表空间
D. 一个数据文件只能属于一个表空间
- (8) 关于数据块的说法正确的是（ ）。
A. 数据块是数据库中最小的逻辑单元
B. 数据块由一系列扩展区组成

- C. 数据块的大小不能更改
 - D. 一系列连续的数据块组成扩展区
- (9) SGA 区包括 ()。
- A. 共享存储区
 - B. 重做日志缓冲区
 - C. 排序区
 - D. 数据库高速缓冲区
- (10) 下面哪个后台进程可实现实例启动时对其进行自动恢复? ()
- A. ARCH
 - B. PMON
 - C. SMON
 - D. CKPT