



目 录

开卷首语

数据库管理

2GB 问题

The Oracle DBA's Guide to NT

在 Oracle 中运行 OS 命令

Oracle DBA 日常管理

数据库开发

关于 Oracle 中大对象(lob)处理的一些方法和实例

Developer2000 系列讲座一

查看表结构的 script

ORACLE 认证

OCP 不完全指南

LZ0-024 考点总结

企业信息化

中小企业如何选择 MRP 系统

CRM 的核心、应用重点及分布实施步骤

Oracle Applications 11i 技术支持经验杂谈

论坛精华

制作人员

开卷首语

我们一直在成长.....

本刊所有文章版权归作者所有，如有转载，请与 webmaster@itpub.net 联系，多谢支持，一起保护我们的劳动成果。

>> 2001 ITPUB 诞生

2001 年是创世纪的一年。

2001 年是中国人扬眉吐气的一年，WTO，奥运会，世界杯围绕着我们。我们感到无比的幸福。

2001 年又是 IT 辛酸的一年，伴随着 911，整体陷入了滑坡。

2001 年在 <http://www.smiling.com.cn>，诞生一个小组名字是 oracle。

2001 年 10 月，oracle 小组搬家，全新的 ITPUB 诞生。

>> 决心

ITPUB 立志成为 IT 技术中心交流社区。

ITPUB 坚持自由平等开放的精神，这里容纳一切对技术的不同的见解，这里接纳不攻击别人的任何个性，目的只有一个，希望我们都能在 ITPUB 中交流，学习和进步。

>> 展望

2002 年是 ITPUB 关键的一年，我们将抓好以下几件事：

- 1) 论坛的稳定性
- 2) 电子杂志的稳定发行
- 3) 针对新手推出一些列栏目，活动。
- 4) 推出几期重量级的嘉宾聊天
- 5)

>> 祝福

2001 年不管它怎么样，过去的就让他过去吧。

2002 年对我们又是一个新的一年。

衷心的祝福大家工作顺利，学习进步，生活幸福,身体健康。

2002 年 ITPUB 需要大家更多呵护与支持，请相信我们，我们一直都在进步！

<http://www.itpub.net>

2GB 问题

-----2Gb or NOT 2Gb-File limits in ORACLE

本文描述"2GB"的问题。它阐述了为什么 2GB 是一个关键数字并且给出了如果文件大于 2GB 时一些你需要知道的东西。

本文主要阐述在 Unix 系统上的 2GB 问题。

讨论的主题：

为什么 2GB 是一个特别的数字

为什么使用 2GB 的数据文件

Export 和 2GB

SQL*Loader 和 2GB

ORACLE 和其他 2GB 问题

其他

为什么 2GB 是一个特殊的数字

许多 CPU 和系统调用接口(API)使用 32 位的字 (word)。这个 32 位就在许多系统上带来了限制。在许多情况下,标准文件操作的 API 使用 32 位有符号 word 来指定文件大小和在文件中的相对位置。有符号的 32 位 word 使用最高位为符号位,因此 31 位所能表示的最大值就是 0x7FFFFFFF (+2147483647),比 2GB 小 1。

2GB 或大于 2GB 的文件统称“大文件”。所以在 32 位环境中就会遇到许多问题。为了克服这些问题,现在的操作系统大多使用 64 位定义了新的系统调用。新版本的 ORACLE 使用了这些接口但是在处理“大文件”时还是有许多问题是需要注意的。

另一个比较特殊的数字是 4GB, 0xFFFFFFFF 作为无符号字所能表示的最大值就是 4294967295,比 4GB 少 1。加 1 则造成低 32 位变为 0x00000000 和一个进位,在 32 位体系中这个进位会被丢掉,因此 4GB 是另一个可能会发生问题的数字。

32 位影响着 ORACLE 的许多方面。为了使用大文件,你需要：

- 1.操作系统支持 2GB+的文件或裸设备
- 2.操作系统具有支持操作 2GB+文件 IO 的 API
- 3.ORACLE 使用这些 API

这对 ORACLE 意味着什么？

现今,大多数平台支持大文件并且有 64 位的 API。从 ORACLE 7.3 开始通常就使用这些 API 了,但是这依赖于平台、操作系统和 ORACLE 的版本。在多数情况下大文件支持是可行的,但在某些情况下需要一个专门的 patch。在本文写作时,在 ORACLE 中还有一些工具没有使用这些新的 64 位 API,比如 export, SQL*LOADER,当然这是依赖于平台的具体操作系统数据库版本的。

为什么使用 2GB+的数据文件

这里我们总结出使用大文件/设备作为 ORACLE 的数据文件的好处和缺点：

好处：

在大多数平台上，ORACLE7 最大支持 1022 个数据文件。每个文件如果小于 2GB，那数据库最大也超不过 2044GB。（在 ORACLE8 上这不是问题，ORACLE8 支持每个表空间 1022 个文件）。大文件的使用可以突破 2044GB 的限制。

对相对小的数据库来讲，大文件意味着更少的文件。也就意味着较少的文件处理及所需资源。

缺点：

恢复的单位更大。一个 2GB 的文件需要 15 分钟到一个小时的备份/恢复时间（依赖于备份介质和磁盘速度）。一个 8GB 的文件需要此时间的 4 倍。

并行备份/恢复操作会受影响。可能有操作系统特殊的限制：比如大于 2GB 的部分，也许异步 IO 只能串行(serialised)操作了。

操作大于 2GB 的文件也许需要补丁（patch），特殊的配置等，相对小文件来讲无形中引入了许多不可测因素，比如在一些 AIX 中。

使用大文件时需要注意的几点：

- 向操作系统厂商确认是否支持大文件并且如何配置他们

- 向操作系统厂商确认可支持的最大实际文件大小

- 向 ORACLE 支持确认是否需要补丁或在你的平台上（硬件、操作系统、ORACLE）是否有什么限制

- 当你升级操作系统或 ORACLE 时，检查以上所提。

- 确认是否正确的设置了系统以允许所有用户能使用大文件

- 确认备份脚本能处理大文件

注意还有一个使用大文件的限制。文件大小的具体数值依赖于数据库的 DB_BLOCK_SIZE 和平台。在大多数平台上（Unix，NT，VMS），文件大小限制为 4194302*DB_BLOCK_SIZE。

请查看 Alert [NOTE:112011.1]中的详细记录

重点注意事项：

当允许文件自动增长时需要特别小心。对 AUTOEXTEND 的文件限制 MAXSIZE 小于 2GB 是明智的。否则，由于[BUG:568232]，当文件增长超过 ORACLE 不能处理时可能会出现 ORA-600[3292]等错误。

大多数平台上 ORACLE 的数据文件包含一个特殊的头数据块，所以创建一个 2GB 的文件，实际需要大于 2GB 的空间。在 Unix 平台上，这个头大小通常是 DB_BLOCK_SIZE，但在裸设备上可能会更大。

2GB 的相关 ORACLE 错误：

ORA-01119 Error in creating datafile xxxx
ORA-27044 unable to write header block of file
SVR4 Error: 22: Invalid argument
ORA-19502 write error on file 'filename', blockno x (blocksize=nn)
ORA-27070 skgfdisp: async read/write failed
ORA-02237 invalid file size
KCF:write/open error dba=xxxxxx block=xxxx online=xxxx file=xxxxxxxx
file limit exceed.
Unix error 27, EFBIG

Export and 2Gb

2Gb Export File Size

本文写作时大多数的 export 版本使用默认的文件处理 API 来创建 export 文件。这意味着在相当多的平台上，不能导出大于 2GB 的文件。

以下是一些克服的方法：

通常可以导出大于 2GB 的文件到裸设备上。

在 Unix 上可以使用命名管道来压缩/分割文件

可以导出到磁盘上

ORACLE8i 允许导出到多个文件而不是一个大的文件

(译：方法可参考文章结尾处补充)

其他的 2GB 导出问题

ORACLE 最大 extent 的大小为 2GB.不幸的是许多发行版本的 ORACLE 中的 export 都有一个问题，就是当指定 compress=y 时，可能导出的文件中其 Next 存储子句会出现大于 2GB 的情况。这会导致即使指定了 ignore=y 时，import 也会出错。本问题可参见 [BUG:708790]和[NOTE:62436.1]

典型的 2GB+时 export 错误：

```
.. exporting table BIGEXPORT
EXP-00015: error on row 10660 of table
BIGEXPORT,
column MYCOL, datatype 96
EXP-00002: error in writing to export file
EXP-00002: error in writing to export file
EXP-00000: Export terminated unsuccessfully
```

在[BUG:185855]中还提到了一个问题：当导出全库时产生的 create tablespace 命令会使用 bytes 作为单位。当 import 时，生成的数据文件若大于 2GB，可能导致 ora-2237 错误。解决办法是先创建表空间(用 M 代替 bytes)，然后导入文件。

导出到磁带

export 中的 volsize 参数最大到 4GB. 在一些平台上只有 2GB。
8i 中已经修改了本问题。[BUG:490190]描述了本问题。

SQL*Loader and 2Gb

典型的，当 SQL*Loader 打开一个大于 2GB 的输入文件时会报如下错误：

SQL*Loader-500: Unable to open file (bigfile.dat)

SVR4 Error: 79: Value too large for defined data type

[NOTE:30528.1]中的例子可以修改后用于大的输入文件。

ORACLE 8.0.6 提高了对大的 discard 和 log 文件的支持。但最大的输入文件依平台不同而不同。

[BUG:948460]详细描述了输入文件的限制。

[BUG:749600]介绍了最大的 discard 文件大小。

ORACLE 和其他 2GB 问题

下面列出了其他的 2GB 问题

- 从 ORACLE 8.0.5 开始可以在大多数平台上使用 64 位 ORACLE 8.0.5 README 文件介绍了这些。见[NOTE:62252.1]

- DBV 也许不能扫描大于 2GB 的文件，并报“DBV-100”错。见[BUG:710888]

- 在建立大于 2GB 文件时，SQL 命令 datafile ... Size xxxxx 必须用‘M’或‘K’，否则报错‘ORA-02237’: invalid file size

- 在 ORACLE 7.3.4 之前表空间的 quotas 不能超过 2GB。如：

Eg : ALTER USER <username> QUOTA 2500M ON <tablespacename>
reports

ORA-2187 : invalid quota specification。见[BUG:425831]

解决办法是赋予 unlimited tablespace 权限。

- 使用 spool 时如果 spool 出的文件超过 2GB 也许会报错。

其他

具体各平台上文件大小限制

Platform	See
~~~~~	~~~
AIX (RS6000 / SP)	[NOTE:60888.1]
HP	[NOTE:62407.1]
Digital Unix	[NOTE:62426.1]

Sequent PTX	[NOTE:62415.1]
Sun Solaris	[NOTE:62409.1]
Windows NT	Maximum 4Gb files on FAT Theoretical 16Tb on NTFS ** See [NOTE:67421.1] before using large files on NT with ORACLE8 **2 There is a problem with DBVERIFY on 8.1.6 See [BUG:1372172] **3 There is a problem with 8.1.6 / 8.1.7 where an autoextend to 4Gb can cause a crash - see [BUG:1668488]

**后记：**这篇文章来自 metalink.ORACLE.com。本没想全文翻译的，因为其中许多详细的介绍都要参考该网站上的许多其他文章 - 我也知道读者读到一个问题，可又见不到具体的解决方法会很恼火，但无论时间精力都不可能让我全部找来，而且似乎也没有必要，翻译了本文就当起个头吧。

**补充：**

export 大文件可以采取的方法：

1. 裸设备

比如直接倒出到/dev/rvtest 等。

2. 命名管道(Unix 下)

```
mknod /tmp/imp_pipe p
compress < /tmp/exp_pipe > export.dmp.Z &
exp file=/tmp/exp_pipe userid=xxx/xxx tables=...
```

```
mknod /tmp/imp_pipe p
uncompress < export.dmp.Z>/tmp/imp_pipe &
imp file=/tmp/imp_pipe userid=xxx/xxx tables=...
```

3. 压缩/文件拆分:(以下只在 ksh 中有效：)

```
echo |exp file=>(compress | split -b 1024m - expdmp-) userid=xxx/xxx tables=...
```

```
echo | imp file=<(cat expdmp- * |zcat) userid=xxx/xxx tables=...
```

(下面由 chao_ping 补充)

4.可以直接倒出到磁带

比如 exp file=/dev/rmt0 ....

5. 可以在 ORACLE8i+版本里面，通过使用 filesize 和 file 相结合，倒出生成多个文件。



## The Oracle DBA's Guide to NT

### Windows NT 平台正在迅速占领中低端市场，你准备好了吗？

在 Oracle 最早在 80 年代中期进入数据库市场的时候，Oracle 选择的主要平台就是 Digital 公司的 VAX 小型机系统，在开发完 VAX 上面的 VMS 版本的 Oracle 以后，再把 Oracle 移植到别的平台的操作系统上，比如 DG-OS，MVS，Unix 等。如果你在 87 年左右就开始使用 Oracle，相信你那个时候用的应该是 VMS 系统。

在 80 年代晚期开始，Oracle 的主要市场转向了开放系统 Unix。在 90 年代中期到现在，绝大多数的 Oracle 系统都运行在 Unix 系统上，很多程序员都不得不艰难的从 VMS 转向到 Unix 平台。

时间走向了 2000 年，我们逐渐看到了另外一个发展方向 - 虽然没有向当时从 VMS 转向 Unix 那么明显：虽然 Unix 仍然占据着高端市场，但是越来越多的中低端用户开始转向 Windows 系统。我们不应该忽视这个现象的存在，应该开始接触和学习 Oracle 在 Windows 平台上的管理。

在这篇文章里面，我向大家介绍 Oracle 在 Unix 平台和在 Windows 平台上有哪些区别，以及在 Windows 平台上管理 Oracle 的常见问题。我们希望读者有一定的 Unix 和 Windows 操作系统基础知识。

### Oracle 在 Windows 系统上的体系结构

Oracle 一直对数据库的可移植性非常重视，实际上它也确实做的非常好。每个平台上的 Oracle 都有一致的操作方法和特性。在几乎所有平台上，都可以看到 Oracle 的存在。在各个不同的平台上，从 Intel 的 Windows95 上的 Oracle 到 IBM 的 S/MVS 上的 Oracle，Oracle 都是同样的一套通用的体系结构：

- 一块公共内存区域，称为 SGA。里面存放数据块缓冲，SQL 缓冲区，字典缓冲区，重做日志缓冲区等。如果配置成 MTS 结构的话，还有会话信息。

- 一些数据库后台进程，比如 dbwr，lgwr，pmon，smon，arc0 等。如果有分布式业务，并行数据库，作业队列等，还有一些别的相关的后台进程。

- Oracle 数据库用来存放表，索引，回滚段等数据的数据文件。

- 重做日志文件，记录在实例失败的时候回滚所需要的信息。

- 一些 Server 进程，用来服务来自客户进程的请求。如果配置成了共享服务器，那么就是 Dispatcher（调度进程）和一些共享的 Server 进程为客户请求服务。

- 一个 Listener，用来监听来自外部客户机的请求。

数据文件和日志文件一般都是用操作系统的文件系统上的文件或者使用操作系统的裸分区来实现，在不同操作系统都是一样的。但是 Oracle 的内存结构和进程结构在 Windows 平台上和别的平台有些区别。

在 Unix 下面，每个 Oracle 进程都是一个单独的 Unix 进程，包括 Oracle 后台进程和 Oracle Server 进程。SGA 存在于 Unix 的共享内存段里面。

下面这幅图表现了 Unix 下面 Oracle 的体系结构：

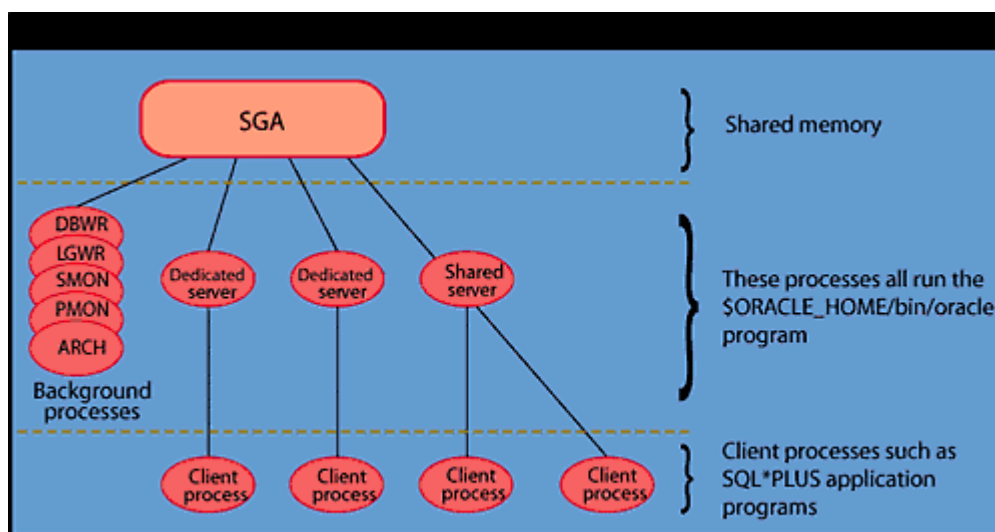


FIGURE 1. Oracle architecture under Unix.

但是 NT 平台上的 Oracle 有些区别。Oracle 充分利用了 NT 对线程的良好支持。下面的附图 2 里，我们可以比较一下 Unix 上的 Oracle 和 NT 上的 Oracle 的区别：在几乎所有操作系统里面，一个进程是不允许操作别的进程的内存空间的，因此在 Unix 下面要配置共享内存。但是在 NT 里面，因为整个 Oracle 就是一个单独的进程，每个 Oracle 的进程在 OS 都表现为 Oracle 进程的线程，因此不同线程允许操作共同的内存空间。

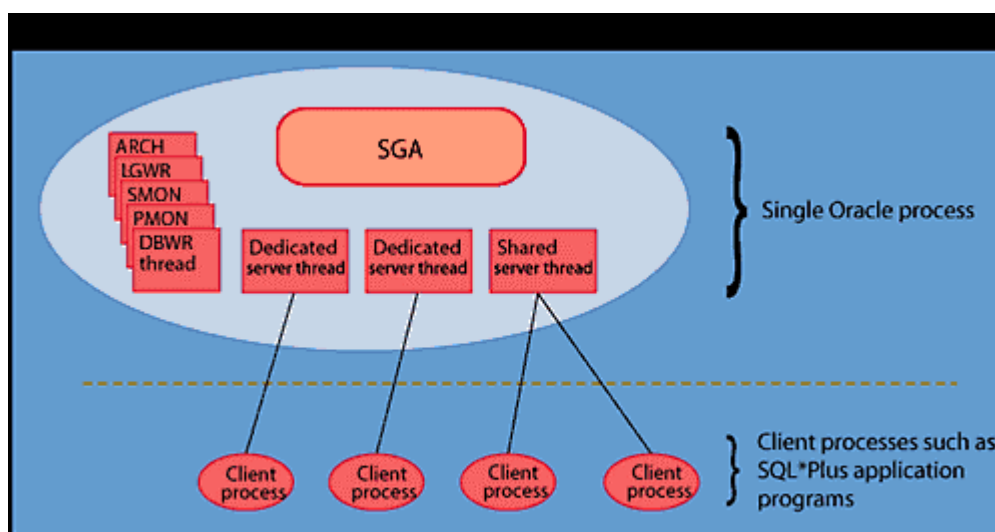


FIGURE 2. Oracle architecture under NT.

在 NT 平台上，每个 Oracle 实例都是一个 NT 进程。在这个进程里面，包括了所有的 Oracle 进程，每个 Oracle 进程实际上以线程的形式出现。NT 平台上的 Oracle 采用 NT 的多个线程/一个进程地方式来实现 Oracle。但是单个进程的模式决定了一个 Oracle 实例能够使用内存的数量。在 NT4.0 里面，每个进程能够寻址 4GB 的虚拟空间，但是有 2G 是保留作为操作系统预留的，Oracle 最多也只能使用 2GB 内存。

一眼看去，2G 内存对于大多数进程应该是够用了，但是记住：这里 Oracle 不但要存放 SGA，还要存放所有的会话信息等。在 NT4 里面，有两个办法可以考虑：使用 NT 企业版，这样，可以设置操作系统只保留 1GB 空间，Oracle 可以寻址 3Gb，或者使用 Alpha 版本的

NT 系统。

Oracle 的多线程服务器模式可以让很多个 Client 进程和少数共享 Server 进程通信，这样就大大减少了 Oracle 的服务器进程，从而减少了对内存的需求。NT 平台的 Oracle8 开始支持多线程服务器。如果进一步采用 Connection Pool 和 Concentrating 技术，可以更加减少对内存的需求。

在 Windows2000 里面，应用程序的可寻址空间达到了 32GB，这对于绝大多数的 Oracle 应用，都是足够了。

## 启动和关闭

在 Unix 环境下，我们通常使用 dbstart/dbshut，svrmgrl 或者 sqlplus 来启动/关闭。在 Oracle 启动的时候，Oracle 启动实例，Mount 数据库，Open 数据库，然后 Oracle 会自动 daemonize，成为后台进程，和登陆会话脱离，这样，当用户注销掉之后，Oracle 仍旧继续运行。

但是在 Windows 平台下，需要用户注销之后，Oracle 仍旧能够运行，Oracle 必须注册成为 NT 的一个服务。NT 下面的 Oracle 主要包含下面这些关键服务：

OracleServicesid 这个服务定义了 NT 机器上的一个 Oracle 实例。一般格式是 OracleService%SID%。比如我的 SID 是 ABC，那么相应的服务名字就是 OracleServiceABC。只有启动这个服务，才能够进一步启动 Oracle 实例，Mount 和 Open 数据库。但是也可以设置成实例自动 Open 数据库。

OracleStartsid 这个只是在 Oracle8 版本里面才出现的。在 Oracle8i 里面，这个服务和 OracleServiceSid 合并了。启动这个服务相当于就是执行了一个启动数据库的脚本。

OracleTNSListener 这个时 Oracle 的监听进程。一般一个默认 Listener 就够用了。如果设置了别的 listener 名字，就会产生相应的服务。

如果希望 Oracle 能够自动启动，可以设置相应服务属性为 Auto。

```
net start OracleServiceGDB1
net start OracleStartGDB1
net start OracleTNSListener80
```

我们可以通过 oradim 这个工具来创建，删除，编辑，启动和停止 OracleService 服务。但是我们一般建议使用 svrmgrl 或者 sqlplus 来关闭相应的服务。这是因为在低版本的 Oracle 里面，在 NT 一个级别停止 Oracle 服务，相当于在 Oracle 里面执行了 shutdown abort，这是很不规范的操作。虽然绝大多数情况下，Oracle 也都是能够自动恢复的，但是这个不是 Oracle 推荐的关闭数据库的方法。

从 Oracle8 和 8i 开始，Oracle 添加了注册表参数：ora_shutdown 或者 ora_<sid>_shutdown，Ora_shutdown_type，Ora_shutdown_timeout 等参数来控制从 NT 级别关闭数据库默认采取的方式。比如我们设定了 ora_<sid>_shutdowntype=I，ora_<sid>_shutdow_timeout=30，那么 Oracle 会先使用 shutdown immediate 来关闭，如果过了 30 秒还没有关闭成功，NT 使用 shutdown abort 来关闭数据库。

## 多个 Oracle 版本共存时候的考虑

在 Unix 下面管理多个版本的 Oracle 相对容易，我们只需要把不同的 Oracle 软件安装到不同的 Oracle_HOME 下面，然后切换的时候，指定环境变量 \$ORACLE_HOME 就可以了。

在 NT 下面，其实也差不多。不同的是 NT 默认使用注册表下面的键值：HKEY_LOCAL_MACHINE\SOFTWARE\ORACLE\ORACLE_HOME 来记录信息。

## 如何进行备份

和 Unix 上面的 Oracle 备份一样，我们同样有很多种可选的备份方案：冷备份，OS 热备份，exp 备份，Rman 备份等。对于 exp 和冷备份，在 Unix 和 NT 上没有什么区别。

NT 的备份工具可以备份到磁带，Oracle 的 ocopy 只能够备份到磁盘。因此如果使用 OS 备份，一般都需要有个中转地，让 Oracle 先备份到磁盘另外一个空间，再通过 NT 只得备份工具拷贝到磁带上。或者可以用 Rman 和相应的备份软件（如 Oracle 捆绑的 Legato）等实现直接备份到磁带。

## 定期实行备份

在 Unix 下面，我们通过 crontab 来实现定期的作业管理，同样，在 Windows 下面，我们也可以实现，通过 AT 或者计划任务。我们可以决定是只执行一次，还是一直执行等。

At 的基本命令格式是：

At 时间 [/every:date] [/next:date] 时间是 24 小时格式的，date 可以是每个月的 1-31 或者每星期的 M, T, W, TH, F, S, SU。

例如，我们希望系统能够在每天下午三点进行归档：

```
at 15:00 /every:m,t,w,th,f,s,su cmd /c  
c:\bin\archive.bat >c:\logs\archive.log
```

当然，要实现定期执行程序，我们必须把 Windows 的 Schedule Service 服务启动。

## 进行性能检查

监视数据库的性能，诊断性能问题和性能上的瓶颈，从而对数据库进行性能优化，这是 DBA 的主要职责之一。虽然理论上讲，在 NT 上和 Unix 上面进行性能调优的方法一样，但是实际使用的工具还是不一样的。

在 Unix 下面，我们一般通过 vmstat, sar, top 等工具来检查系统的整体性能和单独进程的情况。在 NT 下面，我们一般通过 NT 自带的进程管理器合性能监视器来检查服务器的整体性能。

NT 的性能监视器可以检查系统范围内很多的总体情况，比如总体的 CPU 使用，内存使用，网络使用，磁盘等信息。也可以检查某个单独线程的信息。

作为 NT 平台上的 Oracle DBA，必须对进程管理器和性能监视器非常熟悉。

Oracle 还自己提供了接口，利用 NT 的性能监视器来管理和监视 Oracle 数据库的性能。

## 监控每个线程使用的 CPU 资源

虽然在 NT 的进程管理器里面可以看到究竟哪些进程占用了过多的资源，但是无法深入进程，看究竟是那个线程占用了过多的资源。因为就如前面所说，Oracle 在 NT 上的实现是一个进程多个线程，从 OS 的角度，就是一个进程。我们可以通过下面的脚本，来监控究竟是那个 Oracle 的线程占用了多少资源。（注意，这里后台进程使用的 CPU 资料无法看到）。

**LISTING 1. CPU usage query.**

```
column program format a20
column username format a12

select p.spid thread , s.username ,
       decode(nvl(p.background , 0) , 1 , bg.description ,
              s.program ) program ,
       ss.value/100 CPU , physical_reads disk_io
from v$process p ,
     v$session s ,
     v$sesstat ss ,
     v$sess_io si ,
     v$bgprocess bg
where s.paddr=p.addr
and ss.sid=s.sid
and ss.statistic#=12
and si.sid=s.sid
and bg.paddr(+)=p.addr
order by ss.value desc;
```

可以通过 Oracle 的 OEM 工具：Oracle Top Session 来检查每个单独线程占用 CPU 等资源的情况。也可以使用单独的第三方工具，比如 dbtop.pl 脚本(werple.net.au~gharriso 可以下载)。

## 性能调优

在 NT 平台上对 Oracle 进行性能调优，其实基本原则和在 Unix 上进行性能调优没有什么区别：

- 调整 SQL，优化应用程序的设计

- 确保使用了合适的够用的硬件：包括 CPU，内存，网络资源，磁盘子系统等，能够符合应用程序的要求。

- 尽量把数据文件放到不同的物理磁盘上面，减少 IO 等待。

- 尽量把 Oracle 的 redo log 和归档日志文件，数据文件等放到不同的物理磁盘上。保证可恢复性和理想的性能。

- 给 Oracle 分配足够的 SGA，回滚段，排序区和临时表空间等内部资源。

下面这些是 NT 平台上特有的一些要考虑的方面：

- 对于 IO 非常密集的应用，可以考虑使用裸设备。在 NT 上面使用裸设备需要注意的东西和在 Unix 上用裸设备没有什么差别。可以使用 ocopy 或者 rman 来备份 NT 上面的裸设备上面的数据文件。

- 如果使用文件系统来存放数据文件，处于性能的考虑，两者差不多，但是基于安全性的考虑，应该优先使用 NTFS，因为它实现了权限控制，对于 Oracle 的安全比较重要。

- 由于 Windows 系统支持文件系统的异步 IO，一般不需要设置多个 dbwr 进程。

## BUGS? WHAT BUGS?

I may have painted a fairly rosy picture of the Oracle/NT environment. However, the NT



environment is no more immune to bugs than others. Although most bugs are either noncritical or have workarounds or patches, you may encounter a few of them while attempting to implement some of the features I've discussed. For example:

The Oracle8 facility to shut down a service cleanly is not functional in Oracle8.0.3.0 (see Oracle Product Change Request #513414).

Running in archivelog mode--and hence performing online backups--is not supported in Oracle7.3.2.1 (see alert #10802099).

The integration of Oracle logging into the NT performance monitor does not always work without manual intervention (see Oracle Product Change Requests #556846 and #520639).

## ORACLE ON NT: HERE TO STAY

I've given you just a taste of Oracle administration under Windows NT. Oracle administration is a big topic, and we've barely scratched the surface here. However, I think you'll see that transitioning Oracle from Unix to NT is neither particularly unpleasant nor troublesome, although there are challenges.

Both Unix and NT are here to stay, at least for the foreseeable future. While Oracle professionals who restrict themselves to the Unix environment will continue to be in demand, those who embrace both environments will have access to a greater range of Oracle installations and will experience greater variety in their professional lives. If you're an Oracle professional, you owe it to yourself to be at least familiar with Oracle on NT.

Guy Harrison is an independent Oracle consultant specializing in Oracle development and performance issues and author of Oracle SQL High Performance Tuning (Prentice Hall, 1997). He is currently working with Quest Software Inc. in Australia. You can contact Guy at [gharriso@werple.net.au](mailto:gharriso@werple.net.au) or through his home page at [werple.net.au/~gharriso/](http://werple.net.au/~gharriso/).

### 我们究竟使用什么文件系统？

在 Windows 上面，我们可以使用 FAT，FAT32，NTFS，压缩的 NTFS 和裸设备。那么究竟使用什么文件系统的性能更加优越呢？我们来做一个简单的试验，进行插入，索引扫描，和全表扫描，看看结果如何。

对于每个试验，我们通过导入一个 8MB 的文件，然后进行一个大的查询（涉及四个刚才导入的大表，通过索引进行查询）。测试使用的机器配置为 NT 4, Pentium Pro 256MHz, 单块 CPU，64M 内存，一个 2.5GB 的 IDE 硬盘。每次不同的测试之前，机器都重新启动一次。

从下面的测试结果我们可以看到，NTFS 和 FAT 的性能基本一样，裸设备的性能在各个方面都明显高于文件系统。NT 的压缩文件系统效率非常低下，不能考虑。

当然，在真正考虑使用什么文件系统的时候，还应该同时考虑到安全性，易管理性，如何进行备份等各个因素。

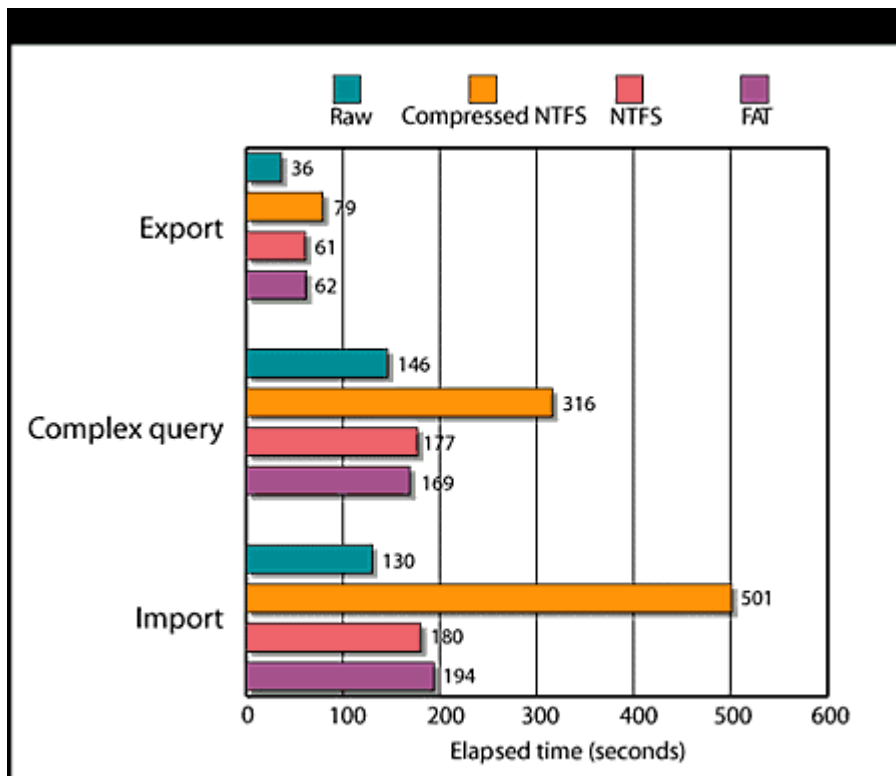


FIGURE 7. The results of a simple file-system benchmark.

原作：Guy Harrison

推荐：snowwhite2000

翻译改编：chao_ping

## 在 ORACLE 中运行 OS 命令

在 Oracle 8i 中，往往会出现要在存储过程中运行操作系统命令的情况。一般来说，利用 Oracle Enterprise Manager 设定作业时可以达到这个目的。但是由于 OEM 在设定作业缺乏灵活性，设定的作业的参数是固定的。在实际应用当中往往需要在 SQL 语句当中运行需要随时运行操作系统命令。Oracle 8i 没有直接运行 OS 命令的语句，我们可以利用 DBMS_PIPE 程序包实现这一要求。

DBMS_PIPE 通过创建管道，可以让至少两个进程进行通信。Oracle 的管道与操作系统的管道在概念上有相同的地方，但是在实现机制不同。

下面介绍实现具体步骤：

### 1 创建一个程序包，姑且起名叫 DAEMON，SQL 语句如下：

```
/*创建 daemon 程序包*/
CREATE OR REPLACE PACKAGE BODY daemon AS
/*execute_system 是实现运行 os 命令的函数*/
FUNCTION execute_system(command VARCHAR2,
timeout NUMBER DEFAULT 10)
RETURN NUMBER IS

status NUMBER;
result VARCHAR2(20);
command_code NUMBER;
pipe_name VARCHAR2(30);
BEGIN
pipe_name := DBMS_PIPE.UNIQUE_SESSION_NAME;
DBMS_PIPE.PACK_MESSAGE('SYSTEM');
DBMS_PIPE.PACK_MESSAGE(pipe_name);
DBMS_PIPE.PACK_MESSAGE(command);
/*向 daemon 管道发送表示命令的字符*/
status := DBMS_PIPE.SEND_MESSAGE('daemon', timeout);
IF status <> 0 THEN
RAISE_APPLICATION_ERROR(-20010,
'Execute_system: Error while sending. Status = ' || status);
END IF;

status := DBMS_PIPE.RECEIVE_MESSAGE(pipe_name, timeout);
IF status <> 0 THEN
RAISE_APPLICATION_ERROR(-20011,
'Execute_system: Error while receiving.
Status = ' || status);
END IF;
/*获取返回结果*/
```



```
DBMS_PIPE.UNPACK_MESSAGE(result);
IF result <> 'done' THEN
RAISE_APPLICATION_ERROR(-20012,
'Execute_system: Done not received. ');
END IF;

DBMS_PIPE.UNPACK_MESSAGE(command_code);
DBMS_OUTPUT.PUT_LINE('System command executed. result = ' ||
command_code);
RETURN command_code;
END execute_system;
/*stop 是让 daemon 停止*/
PROCEDURE stop(timeout NUMBER DEFAULT 10) IS
status NUMBER;
BEGIN
DBMS_PIPE.PACK_MESSAGE('STOP');
status := DBMS_PIPE.SEND_MESSAGE('daemon', timeout);
IF status <> 0 THEN
RAISE_APPLICATION_ERROR(-20030,
'stop: error while sending. status = ' || status);
END IF;
END stop;
END daemon;
```

通过 Sql*Plus 运行以上语句，将为当前用户创建 daemon 程序包。

2 创建在 OS 上运行的守护进程,监听由上面的 daemon 程序包发来的要求执行 OS 命令的语句.以下 Pro*C 的代码，必须由 pro*c 先进行预编译。

```
#include
#include

EXEC SQL INCLUDE SQLCA;

EXEC SQL BEGIN DECLARE SECTION;
char *uid = "scott/tiger";/*在这个地方改为你自己访问的用户,密码,服务名*/
int status;
VARCHAR command[20];
VARCHAR value[2000];
VARCHAR return_name[30];
EXEC SQL END DECLARE SECTION;

void
```

```
connect_error()
{
char msg_buffer[512];
int msg_length;
int buffer_size = 512;

EXEC SQL WHENEVER SQLERROR CONTINUE;
sqlglm(msg_buffer, &buffer_size, &msg_length);
printf("Daemon error while connecting:\n");
printf("%.s\n", msg_length, msg_buffer);
printf("Daemon quitting.\n");
exit(1);
}

void
sql_error()
{
char msg_buffer[512];
int msg_length;
int buffer_size = 512;

EXEC SQL WHENEVER SQLERROR CONTINUE;
sqlglm(msg_buffer, &buffer_size, &msg_length);
printf("Daemon error while executing:\n");
printf("%.s\n", msg_length, msg_buffer);
printf("Daemon continuing.\n");
}

main()
{
EXEC SQL WHENEVER SQLERROR DO connect_error();
EXEC SQL CONNECT :uid;
printf("Daemon connected.\n");

EXEC SQL WHENEVER SQLERROR DO sql_error();
printf("Daemon waiting...\n");
while (1) {
EXEC SQL EXECUTE
BEGIN
/*接收 deamon 发来的字符*/
:status := DBMS_PIPE.RECEIVE_MESSAGE('daemon');
IF :status = 0 THEN
/*取出字符*/
DBMS_PIPE.UNPACK_MESSAGE(:command);
END IF;
```

```
END;
END-EXEC;
IF (status == 0)
{
command.arr[command.len] = '\0';
/*如果是 stop,该进程就退出*/
IF (!strcmp((char *) command.arr, "STOP"))
{
printf("Daemon exiting.\n");
break;
}

ELSE IF (!strcmp((char *) command.arr, "SYSTEM"))
{
EXEC SQL EXECUTE
BEGIN
DBMS_PIPE.UNPACK_MESSAGE(:return_name);
DBMS_PIPE.UNPACK_MESSAGE(:value);
END;
END-EXEC;
value.arr[value.len] = '\0';
printf("Will execute system command '%s'\n", value.arr);
/*运行 os 命令*/
status = system(value.arr);
EXEC SQL EXECUTE
BEGIN
DBMS_PIPE.PACK_MESSAGE('done');
DBMS_PIPE.PACK_MESSAGE(:status);
:status := DBMS_PIPE.SEND_MESSAGE(:return_name);
END;
END-EXEC;

IF (status)
{
printf
("Daemon error while responding to system command.");
printf(" status: %d\n", status);
}
}
ELSE
{
printf
("Daemon error: invalid command '%s' received.\n",
command.arr);
```

```
}  
}  
ELSE  
{  
printf("Daemon error while waiting for signal.");  
printf(" status = %d\n", status);  
}  
}  
EXEC SQL COMMIT WORK RELEASE;  
exit(0);  
}
```

## Oracle DBA 日常管理

修改：2000/8/23

作者：Thomas B. Cox , with Christine Choi

目的：这篇文档有很详细的资料记录着对一个甚至更多的 ORACLE 数据库每天的，每月的，每年的运行的状态的结果及检查的结果，在文档的附录中你将会看到所有检查，修改的 SQL 和 PL/SQL 代码。

修改笔记： 1.1 在 Steve DeNunzio 的'existext.sql'中所识别的类型

1.2 固定的类型

1.3 加的 Gnu Public License；在重建 index 中增加 pctincr 0

1.4 加了一个最新的观点，是从 GEOCITIES 的主页上得到的。

<http://www.geocities.com/tbcox23>

### 目录

#### 1.日常维护程序

- A．检查已起的所有实例
- B．查找一些新的警告日志
- C．检查 DBSNMP 是否在运行
- D．检查数据库备份是否正确
- E．检查备份到磁带中的文件是否正确
- F．检查数据库的性能是否正常合理，是否有足够的空间和资源
- G．将文档日志复制到备份的数据库中
- H．要常看 DBA 用户手册

#### 2.晚间维护程序

- A．收集 VOLUMETRIC 的数据

#### 3.每周维护工作

- A．查找那些破坏规则的 OBJECT
- B．查找是否有违反安全策略的问题
- C．查看错误地方的 SQL*NET 日志
- D．将所有的警告日志存档
- E．经常访问供应商的主页

#### 4.月维护程序

- A．查看对数据库会产生危害的增长速度
- B．回顾以前数据库优化性能的调整
- C．查看 I/O 的屏颈问题
- D．回顾 FRAGMENTATION
- E．将来的执行计划
- F．查看调整点和维护

#### 5.附录

- A．月维护过程
- B．晚间维护过程
- C．周维护过程

#### 6.参考文献

## 一. 日维护过程

### A. 查看所有的实例是否已起

确定数据库是可用的,把每个实例写入日志并且运行日报告或是运行测试文件。当然有一些操作我们是希望它能自动运行的。

可选择执行:用 ORACLE 管理器中的 ' PROBE ' 事件来查看

### B. 查找新的警告日志文件

1. 联接每一个操作管理系统
2. 使用 ' TELNET ' 或是可比较程序
3. 对每一个管理实例,经常的执行 \$ORACLE_BASE/<SID>/bdump 操作,并使其能回退到控制数据库的 SID。
4. 在提示下,使用 UNIX 中的 ' TAIL ' 命令查看 alert_<SID>.log,或是用其他方式检查文件中最近时期的警告日志
5. 如果以前出现过的一些 ORA_ERRORS 又出现,将它记录到数据库恢复日志中并且仔细的研究它们,这个数据库恢复日志在 FILE 中

### C. 查看 DBSNMP 的运行情况

检查每个被管理机器的 ' DBSNMP ' 进程并将它们记录到日志中。

在 UNIX 中,在命令行中,键入 ps -ef | grep dbsnmp,将回看到 2 个 DBSNMP 进程在运行。如果没有,重启 DBSNMP。

### D. 查数据库备份是否成功

### E. 检查备份的磁带文档是否成功

### F. 检查对合理的性能来说是否有足够的资源

#### 1. 检查在表空间中有没有剩余空间。

对每一个实例来说,检查在表空间中是否存在有剩余空间来满足当天的预期的需要。当数据库中已有的数据是稳定的,数据日增长的平均数也是可以计算出来,最小的剩余空间至少要能满足每天数据的增长。

A) 运行 ' FREE.SQL ' 来检查表空间的剩余空间。

B) 运行 ' SPACE.SQL ' 来检查表空间中的剩余空间百分率

#### 2. 检查回滚段

回滚段的状态一般是在线的,除了一些为复杂工作准备的专用段,它一般状态是离线的。

a) 每个数据库都有一个回滚段名字列表。

b) 你可以用 V\$ROLLSTAT 来查询在线或是离线的回滚段的现在状态。

c) 对于所有回滚段的存储参数及名字,可用 DBA_ROLLBACK_SEGS 来查询。但是它不如 V\$ROLLSTAT 准确。

#### 3. 识别出一些过分的增长

查看数据库中超出资源或是增长速度过大的段,这些段的存储参数需要调整。

a) 收集日数据大小的信息,可以用 ' ANALYZE5PCT.SQL '。如果你收集的是每晚的信息,则可跳过这一步。

b) 检查当前的范围,可用 ' NR.EXTENTS.SQL '。

- c) 查询当前表的大小信息。
- d) 查询当前索引大小的信息。
- e) 查询增长趋势。

#### 4. 确定空间的范围。

如果范围空间对象的 NEXT_EXTENT 比表空间所能提供的最大范围还要大, 那么这将影响数据库的运行。如果我们找到了这个目标, 可以用 ' ALTER TABLESPACE COALESCE ' 调查它的位置, 或加另外的数据文件。

A) 运行 ' SPACEBOUND.SQL '。如果都是正常的, 将不返回任何行。

#### 5. 回顾 CPU, 内存, 网络, 硬件资源论点的过程

A) 检查 CPU 的利用情况 进到 x:\web\phase2\default.htm =>system metrics=>CPU 利用页, CPU 的最大限度为 400, 当 CPU 的占用保持在 350 以上有一段时间的话, 我们就需要查看及研究出现的问题。

#### G. 将存档日志复制到备用数据库中

如果有一个备用数据库, 将适当的存档日志复制到备用数据库的期望位置, 备用数据库中保存最近期的数据。

#### F. 经常查阅 DBA 用户手册

如果有可能的话, 要广泛的阅读, 包括 DBA 手册, 行业杂志, 新闻组或是邮件列表。

## 二. 晚间维护过程

大部分的数据库产品将受益于每晚确定的检查进程的运行。

### A. 收集 VOLUMETRIC 数据

#### 1. 分析计划和收集数据

更准确的分析计算并保存结果。

- a) 如果你现在没有作这些的话, 用 ' MK VOLFACT.SQL ' 来创建测定体积的表。
- b) 收集晚间数据大小的信息, 用 ' ANALYZE COMP.SQL '。
- c) 收集统计结果, 用 ' POP VOL.SQL '。
- d) 在空闲的时候检查数据, 可能的话, 每周或每个月进行。

我是用 MS EXCEL 和 ODBC 的联接来检查数据和图表的增长

## 三. 每周维护过程

### A. 查找被破坏的目标

- 1. 对于每个给定表空间的对象来说, NEXT_EXTENT 的大小是相同的, 如 12/14/98 缺省的 NEXT_EXTENT 的 DATAHI 为 1G, DATALO 为 500MB, INDEXES 为 256MB。
  - A) 检查 NEXT_EXTENT 的设置, 可用 ' NEXTEXT.SQL '。
  - B) 检查已有的 EXTENTS, 可用 ' EXISTEXT.SQL '。
- 2. 所有的表都应该有唯一的主键
  - a) 查看那些表没有主键, 可用 ' NO_PK.SQL '。
  - b) 查找那些主键是没有发挥作用的, 可用 ' DIS_PK.SQL '。
  - c) 所有作索引的主键都要是唯一的, 可用 ' NONUPK.SQL ' 来检查。
- 3. 所有的索引都要放到索引表空间中。运行 ' MKREBUILD_IDX.SQL '。
- 4. 不同的环境之间的计划应该是同样的, 特别是测试环境和成品环境之间的。

计划应该相同。

- a) 检查不同的 2 个运行环境中的数据类型是否一致，可用  
' DATATYPE.SQL '。
- b) 在 2 个不同的实例中寻找对象的不同点，可用  
' OBJ_COORD.SQL '。
- c) 更好的做法是，使用一种工具，象寻求软件的计划管理器那样的工具。

B . 查看是否有危害到安全策略的问题。

C . 查看报错的 SQL*NET 日志。

1. 客户端的日志。
2. 服务器端的日志。

D . 将所有的警告日志存档

E . 供应商的主页

1. ORACLE 供应商

<http://www.oracle.com>

<http://technet.oracle.com>

<http://www.oracle.com/support>

<http://www.oramag.com>

2. Quest Software

<http://www.quest.com>

3. Sun Microsystems

<http://www.sun.com>

#### 四 . 月维护过程

A . 查看对数据库会产生危害的增长速度

1. 从以前的记录或报告中回顾段增长的变化以此来确定段增长带来危害

B . 回顾以前数据库优化性能的调整

1. 回顾一般 ORACLE 数据库的调整点，比较以前的报告来确定有害的发展趋势。

C . 查看 I/O 的屏颈问题

1. 查看前期数据库文件的活动性，比较以前的输出来判断有可能导致屏颈问题的趋势。

D . 回顾 FRAGMENTATION

E . 计划数据库将来的性能

1. 比较 ORACLE 和操作系统的 CPU，内存，网络，及硬盘的利用率以此来确定在近期将会有的一些资源争夺的趋势
2. 当系统将超出范围时要把性能趋势当作服务水平的协议来看

F . 完成调整和维护工作

1. 使修改满足避免系统资源的争夺的需要，这里面包括增加新资源或使预期的停工。

#### 五 . 附录

A. 日常程序



```
--
-- free.sql
--
--To verify free space in tablespaces
--Minimum amount of free space
--document your thresholds:
--<tablespace_name> = <amount> m
--

SELECT tablespace_name, sum ( blocks ) as free_blk , trunc ( sum ( bytes ) /
(1024*1024) ) as free_m, max ( bytes ) / (1024) as big_chunk_k, count (*) as num_chunks
FROM dba_free_space
GROUP BY tablespace_name
```

### 1. Space.sql

```
--
-- space.sql
--
-- To check free, pct_free, and allocated space within a tablespace
--
-- 11/24/98

SELECT tablespace_name, largest_free_chunk
      , nr_free_chunks, sum_alloc_blocks, sum_free_blocks
      , to_char(100*sum_free_blocks/sum_alloc_blocks, '09.99') || '%'
      AS pct_free
FROM ( SELECT tablespace_name
      , sum(blocks) AS sum_alloc_blocks
      FROM dba_data_files
      GROUP BY tablespace_name
    )
    , ( SELECT tablespace_name AS fs_ts_name
      , max(blocks) AS largest_free_chunk
      , count(blocks) AS nr_free_chunks
      , sum(blocks) AS sum_free_blocks
      FROM dba_free_space
      GROUP BY tablespace_name )
WHERE tablespace_name = fs_ts_name
```

### 2. analyze5pct.sql

```
--
-- analyze5pct.sql
--
-- To analyze tables and indexes quickly, using a 5% sample size
-- (do not use this script if you are performing the overnight
```

```
-- collection of volumetric data)
```

```
--
```

```
-- 11/30/98
```

```
BEGIN
```

```
    dbms_utility.analyze_schema ( '&OWNER', 'ESTIMATE', NULL, 5 ) ;
```

```
END ;
```

```
/
```

### 3. nr_extents.sql

```
--
```

```
-- nr_extents.sql
```

```
--
```

```
-- To find out any object reaching <threshold>
```

```
-- extents, and manually upgrade it to allow unlimited
```

```
-- max_extents (thus only objects we *expect* to be big
```

```
-- are allowed to become big)
```

```
--
```

```
-- 11/30/98
```

```
SELECT e.owner, e.segment_type , e.segment_name , count(*) as nr_extents ,  
s.max_extents
```

```
, to_char ( sum ( e.bytes ) / ( 1024 * 1024 ) , '999,999.90') as MB
```

```
FROM dba_extents e , dba_segments s
```

```
WHERE e.segment_name = s.segment_name
```

```
GROUP BY e.owner, e.segment_type , e.segment_name , s.max_extents
```

```
HAVING count(*) > &THRESHOLD
```

```
    OR ( ( s.max_extents - count(*) ) < &&THRESHOLD )
```

```
ORDER BY count(*) desc
```

### 4. spacebound.sql

```
--
```

```
-- spacebound.sql
```

```
--
```

```
-- To identify space-bound objects. If all is well, no rows are returned.
```

```
-- If any space-bound objects are found, look at value of NEXT extent
```

```
-- size to figure out what happened.
```

```
-- Then use coalesce (alter tablespace <foo> coalesce;).
```

```
-- Lastly, add another datafile to the tablespace if needed.
```

```
--
```

```
-- 11/30/98
```

```
SELECT a.table_name, a.next_extent, a.tablespace_name
```

```
FROM all_tables a,
```

```
    ( SELECT tablespace_name, max(bytes) as big_chunk
```

```
FROM dba_free_space
GROUP BY tablespace_name ) f
WHERE f.tablespace_name = a.tablespace_name
      AND a.next_extent > f.big_chunk
```

## B. 每晚处理程序

### 1. mk_volfact.sql

```
--
-- mk_volfact.sql (only run this once to set it up; do not run it nightly!)
--
-- -- Table UTL_VOL_FACTS

CREATE TABLE utl_vol_facts
(
  table_name          VARCHAR2(30),
  num_rows            NUMBER,
  meas_dt             DATE
)
TABLESPACE platab
STORAGE (
  INITIAL      128k
  NEXT         128k
  PCTINCREASE  0
  MINEXTENTS   1
  MAXEXTENTS   unlimited
)
/

-- Public Synonym

CREATE PUBLIC SYNONYM utl_vol_facts FOR &OWNER..utl_vol_facts
/

-- Grants for UTL_VOL_FACTS

GRANT SELECT ON utl_vol_facts TO public
/

2. analyze_comp.sql
--
-- analyze_comp.sql
--
BEGIN
```

```
sys.dbms_utility.analyze_schema ( '&OWNER','COMPUTE');
END ;
/
3. pop_vol.sql
--
-- pop_vol.sql
--
insert into utl_vol_facts
select table_name
      , NVL ( num_rows, 0) as num_rows
      , trunc ( last_analyzed ) as meas_dt
from all_tables          -- or just user_tables
where owner in ('&OWNER') -- or a comma-separated list of owners
/
commit
/
```

## C. 每周处理程序

### 1. nexttext.sql

```
--
-- nexttext.sql
--
-- To find tables that don't match the tablespace default for NEXT extent.
-- The implicit rule here is that every table in a given tablespace should
-- use the exact same value for NEXT, which should also be the tablespace's
-- default value for NEXT.
--
-- This tells us what the setting for NEXT is for these objects today.
--
-- 11/30/98
```

```
SELECT segment_name, segment_type, ds.next_extent as Actual_Next
      , dt.tablespace_name, dt.next_extent as Default_Next
FROM dba_tablespaces dt, dba_segments ds
WHERE dt.tablespace_name = ds.tablespace_name
      AND dt.next_extent != ds.next_extent
      AND ds.owner = UPPER ( '&OWNER' )
ORDER BY tablespace_name, segment_type, segment_name
```

### 2. existtext.sql

```
--
-- existtext.sql
--
```

```
-- To check existing extents
--
-- This tells us how many of each object's extents differ in size from
-- the tablespace's default size. If this report shows a lot of different
-- sized extents, your free space is likely to become fragmented. If so,
-- this tablespace is a candidate for reorganizing.
--
-- 12/15/98
```

```
SELECT segment_name, segment_type
, count(*) as nr_exts
, sum ( DECODE ( dx.bytes,dt.next_extent,0,1) ) as nr_illsized_exts
, dt.tablespace_name, dt.next_extent as dflt_ext_size
FROM dba_tablespaces dt, dba_extents dx
WHERE dt.tablespace_name = dx.tablespace_name
AND dx.owner = '&OWNER'
GROUP BY segment_name, segment_type, dt.tablespace_name, dt.next_extent
```

### 3. No_pk.sql

```
--
-- no_pk.sql
--
-- To find tables without PK constraint
--
-- 11/2/98
```

```
SELECT table_name
FROM all_tables
WHERE owner = '&OWNER'
MINUS
SELECT table_name
FROM all_constraints
WHERE owner = '&&OWNER'
AND constraint_type = 'P'
```

### 4. disPK.sql

```
--
-- disPK.sql
--
-- To find out which primary keys are disabled
--
-- 11/30/98
```

```
SELECT owner, constraint_name, table_name, status
FROM all_constraints
WHERE owner = '&OWNER' AND status = 'DISABLED' AND constraint_type = 'P'
```

## 5. nonuPK.sql

```
--
-- nonuPK.sql
--
-- To find tables with nonunique PK indexes. Requires that PK names
-- follow a naming convention. An alternative query follows that
-- does not have this requirement, but runs more slowly.
--
-- 11/2/98
```

```
SELECT index_name, table_name, uniqueness
FROM all_indexes
WHERE index_name like '&PKNAME%'
      AND owner = '&OWNER' AND uniqueness = 'NONUNIQUE'
```

```
SELECT c.constraint_name, i.tablespace_name, i.uniqueness
FROM all_constraints c, all_indexes i
WHERE c.owner = UPPER ( '&OWNER' ) AND i.uniqueness = 'NONUNIQUE'
AND c.constraint_type = 'P' AND i.index_name = c.constraint_name
```

## 6. mkrebuild_idx.sql

```
--
-- mkrebuild_idx.sql
--
-- Rebuild indexes to have correct storage parameters
--
-- 11/2/98
```

```
SELECT 'alter index ' || index_name || ' rebuild '
      , 'tablespace INDEXES storage '
      || ' ( initial 256 K next 256 K pctincrease 0 ) ; '
FROM all_indexes
WHERE ( tablespace_name != 'INDEXES'
      OR next_extent != ( 256 * 1024 )
      )
AND owner = '&OWNER'
/
```

## 7. datatype.sql

```
--  
-- datatype.sql  
--  
-- To check datatype consistency between two environments  
--  
-- 11/30/98
```

```
SELECT  
    table_name,  
    column_name,  
    data_type,  
    data_length,  
    data_precision,  
    data_scale,  
    nullable  
FROM all_tab_columns -- first environment  
WHERE owner = '&OWNER'  
MINUS  
SELECT  
    table_name,  
    column_name,  
    data_type,  
    data_length,  
    data_precision,  
    data_scale,  
    nullable  
FROM all_tab_columns@&my_db_link -- second environment  
WHERE owner = '&OWNER2'  
order by table_name, column_name
```

#### [8. obj_coord.sql](#)

```
--  
-- obj_coord.sql  
--  
-- To find out any difference in objects between two instances  
--  
-- 12/08/98
```

```
SELECT object_name, object_type  
FROM user_objects  
MINUS  
SELECT object_name, object_type  
FROM user_objects@&my_db_link
```

## 六. 参考文献

1. Loney, Kevin Oracle8 DBA Handbook
2. Cook, David Database Management from Crisis to Confidence  
[<http://www.orapub.com/>]
3. Cox, Thomas B. The Database Administration Maturity Model



## 关于 oracle 中大对象 (lob) 处理的一些方法和实例

整理：蓝带鱼 参与讨论：蓝带鱼，coolsword，zl_king，jlandzpa，youngcow，flytiger

在 oracle 中，有 4 个大对象 (lobs) 类型可用，分别是 blob, clob, bfile, nclob。

下面是对 lob 数据类型的简单介绍：

blob: 二进制 lob，为二进制数据，最长可达 4GB，存贮在数据库中。

clob: 字符 lob，字符数据，最长可以达到 4GB，存贮在数据库中。

bfile: 二进制文件；存贮在数据库之外的只读型二进制数据，最大长度由操作系统限制。

nclob：支持对字节字符集(nultibyte character set)的一个 clob 列。

对于如何检索和操作这些 lob 数据一直是 oracle 数据库开发者经常碰到的问题。下面我将在 oracle 对 lob 数据处理的一些方法和技巧，介绍给读者，希望能够对读者以后的开发有所帮助。

oracle 中可以用多种方法来检索或操作 lob 数据。通常的处理方法是通过 dbms_lob 包。其他的方法包括使用 api(application programming interfaces)应用程序接口和 oci (oracle call interface) oracle 调用接口程序。

一、在 oracle 开发环境中我们可以用 dbms_lob 包来处理！dbms_lob 包功能强大，简单应用。既可以用来读取内部的 lob 对象，也可以用来处理 bfile 对象。但处理两者之间，还有一点差别。处理内部 lob 对象 (blob, clob) 时，可以进行读和写，但处理外部 lob 对象 bfile 时，只能进行读操作，写的操作可以用 pl/sql 处理。另外用 sql 也可以处理 lob，但要注意 sql 仅可以处理整个 lob，不能操作 lob 的数据片。

在 dbms_lob 包中内建了 read(), append(), write(), erase(), copy(), getlength(), substr() 等函数，可以很方便地操作 lob 对象。这里不做深入讨论，读者可以参看相关的书籍。

对于 pl/sql，下面介绍一种技巧，用动态的 pl/sql 语句处理 clob 对象来传替表名！

### example 1.

动态 PL/SQL, 对 CLOB 字段操作可传递表名 table_name, 表的唯一标志字段名 field_id, clob 字段名 field_name, 记录号 v_id, 开始处理字符的位置 v_pos, 传入的字符串变量 v_clob, 修改 CLOB 的 PL/SQL 过程：updateclob

```
create or replace procedure updateclob(
```

```
    table_name      in varchar2,
    field_id        in varchar2,
    field_name      in varchar2,
    v_id            in number,
    v_pos           in number,
    v_clob          in varchar2)
```

```
is
```

```
    lobloc clob;
    c_clob varchar2(32767);
    amt     binary_integer;
    pos     binary_integer;
    query_str varchar2(1000);
```

```
begin
```

```
    pos:=v_pos*32766+1;
```

```
amt := length(v_clob);
c_clob:=v_clob;
query_str :='select '||field_name||' from '||table_name||' where '||field_id||'= :id for
update ';
--initialize buffer with data to be inserted or updated
EXECUTE IMMEDIATE query_str INTO lobloc USING v_id;
--from pos position, write 32766 varchar2 into lobloc
dbms_lob.write(lobloc, amt, pos, c_clob);
commit;
exception
when others then
rollback;
end;
```

/用法说明：

在插入或修改以前，先把其它字段插入或修改，CLOB 字段设置为空 empty_clob()，然后调用以上的过程插入大于 2048 到 32766 个字符。

如果需要插入大于 32767 个字符，编一个循环即可解决问题。

查询 CLOB 的 PL/SQL 函数：getclob

```
create or replace function getclob(
    table_name      in varchar2,
    field_id        in varchar2,
    field_name      in varchar2,
    v_id            in number,
    v_pos           in number) return varchar2
is
    lobloc  clob;
    buffer  varchar2(32767);
    amount  number := 2000;
    offset  number := 1;
    query_str  varchar2(1000);
begin
    query_str :='select '||field_name||' from '||table_name||' where '||field_id||'= :id ';
    --initialize buffer with data to be found
    EXECUTE IMMEDIATE query_str INTO lobloc USING v_id;
    offset:=offset+(v_pos-1)*2000;
    --read 2000 varchar2 from the buffer
    dbms_lob.read(lobloc,amount,offset,buffer);
    return buffer;
exception
when no_data_found then
    return buffer;
end;
```

用法说明：

用 select getclob(table_name,field_id,field_name,v_id,v_pos) as partstr from dual;

可以从 CLOB 字段中取 2000 个字符到 partstr 中，编一个循环可以把 partstr 组合成 dbms_lob.getlength(field_name)长度的目标字符串。

二、对于在其他不同的开发环境，例如 vc,vb,pb,java 等环境下对 lob 的处理，处理方法不尽相同，在这里将简要举几个例子来说明不在 oracle 开发环境下对 lob 的处理。

### (一) 在 pb 中的处理

#### exampler 2.

```
string ls_path,ls_filename,ls_jhdh
long ll_num,ll_count,rtn
blob ole_blob
ll_num=dw_lb.getrow()
if ll_num>0 then ls_jhdh=dw_lb.object.ct_njhhdh[ll_num]
select count(*) into :ll_count from sj_jh_jhfjb where ct_jhdldxbh='1' and ct_jhdh=:ls_jhdh
and ct_jdldxbh=:is_jdlx;
if ll_count>0 then
rtn=messagebox("提示","是否要修改此附件",question!,yesno!,1)
if rtn=1 then
SELECTBLOB ct_jhfjnr INTO le_blob from sj_jh_jhfjb where ct_jhdldxbh='1' and
ct_jhdh=:ls_jhdh and ct_jdldxbh=:is_jdlx;
ole_1.objectdata =ole_blob
If ole_1.activate(offsite!) <> 0 Then
MessageBox("OLE Activate","不能激活")
Return -1
end If
end if
else
messagebox("提示","没有附件")
end if
end if
```

### (二) 在 vb 中的处理

在 vb 中处理大对象，一般可以用 OO4O(oracle objects for ole)来处理大对象。这里介绍一种不用 OO4O 处理大对象 blob 的方法。

下面这段程序可以将一个文件（文本文件，doc 文件，图象文件等）保存到数据库中，并可以将其从数据库读出

需要两个 commandbutton

cmd1 名称 cmdsave caption 保存

cmd2 名称 cmdread caption 读取

一个 cmddialog 控件

同时需要创建一张表 t_demo(字段 id 类型 number,;字段 text 类型 blob;)

#### example 3.

Option Explicit

Dim rn As ADODB.Connection

Public Function CreateDataSource(DataSource As String, UserID As String, Password As String) As Boolean

```
On Error GoTo DbConErr:
Set rn = New ADODB.Connection
With rn
    .ConnectionString = "Provider=OraOledb.Oracle.1;" & _
        "password=" & Password & ";" & _
        "User ID =" & UserID & ";" & _
        "Data Source=" & DataSource & ";" & _
        "Locale Identifier=2052"
    .Open
End With
CreateDataSource = True
Exit Function
DbConErr:
CreateDataSource = False
End Function

Private Sub cmdRead_Click()
Dim rs As New ADODB.Recordset
rs.ActiveConnection = rn
rs.LockType = adLockOptimistic
rs.CursorLocation = adUseClient
rs.Source = "select * from t_demo"
rs.Open
ComDlgDir.DialogTitle = "保存文件"
ComDlgDir.Filter = "*.*)"
ComDlgDir.ShowSave
Call BlobToFile(rs.Fields("text"), ComDlgDir.filename)
Set rs = Nothing
Exit Sub
Set rs = Nothing
End Sub

Private Sub cmdsave_Click()
Dim rs As New ADODB.Recordset
rs.ActiveConnection = rn
rs.LockType = adLockOptimistic
rs.CursorLocation = adUseClient
rs.Source = "select * from t_demo"
rs.Open
rs.AddNew
ComDlgDir.DialogTitle = "选取文件"
ComDlgDir.ShowOpen
rs.Fields("id").Value = 1
If ComDlgDir.filename <> "" Then
```

```
Call FileToBlob(rs.Fields("text"), ComDlgDir.filename)
rs.Update
End If
Set rs = Nothing
Exit Sub
Set rs = Nothing
End Sub

Private Sub Form_Load()
If Not CreateDataSource("sid", "systemp", "manager") Then
MsgBox "Connection failure!"
End If
End Sub

fld As ADODB.Field, filename As String, Optional ChunkSize As Long = 8192)
Dim fnum As Integer, bytesleft As Long, bytes As Long
Dim tmp() As Byte
If (fld.Attributes And adFldLong) = 0 Then
Err.Raise 1001, , "field doesn't support the GetChunk method."
End If
If Dir$(filename) = "" Then Err.Raise 53, , "File not found"
fnum = FreeFile
Open filename For Binary As fnum
bytesleft = LOF(fnum)
Do While bytesleft
bytes = bytesleft
If bytes > ChunkSize Then bytes = ChunkSize
ReDim tmp(1 To bytes) As Byte
Get fnum, , tmp
fld.AppendChunk tmp
bytesleft = bytesleft - bytes
Loop
Close #fnum
End Sub

Sub BlobToFile(fld As ADODB.Field, filename As String, Optional ChunkSize As Long =
8192)
Dim fnum As Integer, bytesleft As Long, bytes As Long
Dim tmp() As Byte
If (fld.Attributes And adFldLong) = 0 Then
Err.Raise 1001, , "field doesn't support the GetChunk method."
End If
If Dir$(filename) <> "" Then Kill filename
fnum = FreeFile
```

```
Open filename For Binary As fnum
bytesleft = fld.ActualSize
Do While bytesleft
    bytes = bytesleft
    If bytes > ChunkSize Then bytes = ChunkSize
    tmp = fld.GetChunk(bytes)
    Put #fnum, , tmp
    bytesleft = bytesleft - bytes
Loop
Close #fnum
End Sub
```

### (三) 用 jdbc 处理 lob

#### example 4.

首先是 Getting BLOB and CLOB Locators from a Result Set

// Select LOB locator into standard result set.

```
ResultSet rs =stmt.executeQuery ("SELECT blob_col, clob_col FROM lob_table");
```

```
while (rs.next())
```

```
{// Get LOB locators into Java wrapper classes.
```

```
    oracle.jdbc2.Blob blob = (oracle.jdbc2.Blob)rs.getObject(1);
```

```
    oracle.jdbc2.Clob clob = (oracle.jdbc2.Clob)rs.getObject(2);
```

```
    [...process...]
```

```
}
```

然后是 Read BLOB data from BLOB locator.

```
InputStream byte_stream = my_blob.getBinaryStream();
```

```
byte [] byte_array = new byte [10];
```

```
int bytes_read = byte_stream.read(byte_array);
```

和 Writing BLOB Data

```
java.io.OutputStream outstream;
```

```
// read data into a byte array
```

```
byte[] data = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
```

```
// write the array of binary data to a BLOB
```

```
outstream = ((BLOB)my_blob).getBinaryOutputStream();
```

```
outstream.write(data);
```

还有 Passing a BLOB Locator to a Prepared Statement

```
OraclePreparedStatement ops = (OraclePreparedStatement)conn.prepareStatement
```

```
"INSERT INTO blob_table VALUES(?)";
```

```
ops.setBLOB(1, my_blob);
```

```
ops.execute();
```

最后应该注意：

insert 的时候一定要用 empty_blob()初始化

```
stmt.execute ("insert into my_blob_table values ('row1', empty_blob());
```

### (四) 在 pro*c 中的处理

PRO*C 可以用三种方式对 LOB 字段处理。

- 1、The DBMS_LOB package inside PL/SQL blocks.
- 2、OCI (Oracle Call Interface) function calls.
- 3、Embedded SQL statements.

Embedded SQL statements.的方式简单而且比较灵活。OTN 上提供一个例子：

In this example we will be reading data from a BLOB with an unknown arbitrary length into a buffer and then writing the data from the buffer into an external file.

Our buffer is small, so depending on the size of the BLOB we are reading, we may be able to read the BLOB value into the buffer in a single READ statement or we may be required to utilize a standard polling method instead.

First we start off with oci.h and some simple local variable declarations

**example 5.**

```
#include <oci.h>
```

```
OCIBlobLocator *blob ;
```

```
FILE *fp ;
```

```
unsigned int amt, offset = 1 ;
```

Now we need a buffer to store the BLOB value and then write to the file from:

```
#define MAXBUFLLEN 5000
```

```
unsigned char buffer[MAXBUFLLEN] ;
```

```
EXEC SQL VAR buffer IS RAW(MAXBUFLLEN) ;
```

Allocate the BLOB host variable and select a BLOB which we will READ:

```
EXEC SQL ALLOCATE :blob ;
```

```
EXEC SQL SELECT a_blob INTO :blob FROM lob_table WHERE ... ;
```

We can then open the external file to which we will write the BLOB value:

```
fp = fopen((const char *)"image.gif", (const char *)"w") ;
```

If the buffer can hold the entire LOB value in a single READ we need to catch the NOT FOUND condition to signal LOB READ termination:

```
EXEC SQL WHENEVER NOT FOUND GOTO end_of_lob ;
```

Now do our first READ. We set the amount to the maximum value of 4 Gigabytes. It is larger than our buffer so if the LOB doesn't fit we will READ using a polling mode:

```
amt = 4294967295 ;
```

```
EXEC SQL LOB READ :amt FROM :blob AT offset INTO :buffer ;
```

If we get here then it means that the buffer was not large enough to hold the entire LOB value, so we must write what we have using binary I/O and continue reading:

```
(void) fwrite((void *)buffer, (size_t)MAXBUFLLEN, (size_t)1, fp) ;
```

We use a standard polling method to continue reading with the LOB READ inside of an infinite loop. We can set up the NOT FOUND condition to terminate the loop:

```
EXEC SQL WHENEVER NOT FOUND DO break ;
```

```
while (TRUE)
```

```
{
```

During polling, the offset is not used so we can omit it in subsequent LOB READs.

We need the amount, however, because it will tell us how much was READ in the last READ invocation

```
EXEC SQL LOB READ :amt FROM :blob INTO :buffer ;
```

```
(void) fwrite((void *)buffer, (size_t)MAXBUFLen, (size_t)1, fp);  
}
```

Here, we have reached the end of the LOB value. The amount holds the amount of the last piece that was READ. During polling, the amount for each interim piece was set to MAXBUFLen, or the maximum size of our buffer:

end_of_lob:

```
(void) fwrite((void *)buffer, (size_t)amt, (size_t)1, fp);
```

### (五) 在 delphi 中的处理

对于 lob 字段而言, 个人认为其使用比 long 类型有很大的灵活性, 而且 lob 字段可以保存各类的数据, 可以保存图片, 大量的文字, 现就 clob 跟 blob 两种类型加以说明, 其中 blob 保存图片信息, clob 保存大量文字。

#### example 6.

Create table test_table

(c_no number(1) not null,

c_blob blob,

c_clob clob,

constraint pk_test_table primary key (c_no));

unit Unit1;

interface

uses

Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, DBCtrls, Grids, DBGrids, DB, DBTables, ExtDlgs;

type

TForm1 = class(TForm)

Database1: TDatabase; //用于连接数据库

Table1: TTable; //获取表信息

DataSource1: TDataSource;

DBGrid1: TDBGrid;

DBMemo1: TDBMemo; //显示 c_clob 字段内容

DBImage1: TDBImage; //显示 c_blob 字段内容

Button1: TButton; //插入按钮

Button2: TButton; //保存按钮

Table1C_NO: TFloatField; //Tfiled

Table1C_BLOB: TBlobField;

Table1C_CLOB: TMemoField;

OpenPictureDialog1: TOpenPictureDialog; //从文件获取图片

OpenDialog1: TOpenDialog; //从文件获取文字

procedure Button1Click(Sender: TObject);



```
    procedure Button2Click(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    Form1: TForm1;

implementation

{$R *.dfm}

procedure TForm1.Button1Click(Sender: TObject);
begin //插入操作
    with Table1 do
        begin
            Insert; //将表状态置为插入状态
            if OpenPictureDialog1.Execute then //获得图片信息
                Table1C_BLOB.LoadFromFile(OpenPictureDialog1.FileName);
            if OpenDialog1.Execute then //获得文字信息
                Table1C_CLOB.LoadFromFile(OpenDialog1.FileName);
        end;
    end;
end;

procedure TForm1.Button2Click(Sender: TObject);
begin //提交插入内容
    try
        Table1.Post;
    except
        Application.MessageBox('错误发生', '警告', 0);
    end;
end;

end.
```

**注意：**

openpiceturedilog 只能打开 dmp,ico,wmf 等文件，事先需要将图片文件格式保存成这几类；

在文字字段不从文件获得时，可以手动输入。

本例只是对 lob 字段的一个小小的探索，用法不当及需改正之处，还请多多指教。

注:本篇文章大部分例子均取自论坛，如有侵犯您的版权，请来信告知，我们会做相应处理。

Developer2000 系列讲座之一 (作者:索马里)

## Developer2000 开发 Step by Step

Oracle 是国际有影响力的资料库管理系统, Developer2000 作为 Oracle 的第二代开发工具是一个功能强大,方便灵活,学习简单的开发工具. 作为 Oracle 的开发人员,应对其要有些了解.

本文采用一步一步图解的方式,向您简单介绍 Developer 简单的程序的开发方式.让大家对 Developer2000 有所了解

本文所用到的表是 demo 下的 Dept 和 Emp 两个表. 环境: 繁体 win95 + Form 5.0, 程序的功能是查询管理各个部门中的员工.

最后程序的界面如图所示:

The screenshot shows the 'Developer2000 Forms Runtime for Windows 95 / NT - [WINDOW1]' window. It features a menu bar with 'Action', 'Edit', 'Query', 'Block', 'Record', 'Field', 'Window', and 'Help'. Below the menu is a toolbar with various icons for file operations and navigation. The main area contains a form with three input fields: '部門編號' (Department Number) with value '10', '部門名稱' (Department Name) with value 'ACCOUNTING', and '地點' (Location) with value 'NEW YORK'. Below these fields is a table displaying employee data. The table has six columns: '員工編號' (Employee Number), '員工名稱' (Employee Name), '職務' (Job Title), '聘用日期' (Hire Date), '收入' (Salary), and '獎金' (Bonus). The data rows are as follows:

員工編號	員工名稱	職務	聘用日期	收入	獎金
7782	CLARK	MANAGER	2081-06-09	2450	
7839	KING	PRESIDENT	2081-11-17	5000	
7934	MILLER	CLERK	2082-01-23	1300	

图 1

一 开发前的准备工作:

1. 去 [otn.oracle.com](http://otn.oracle.com) 注册帐号免费下载 Developer2000 软件
2. 安装并配置好 Developer2000
3. 设置好数据库别名, 本文使用 server 并检查 demo/demo 下的 Dept 和 Emp 两个表是否正常,

二 从开始,程序中打开 Form Builder,跳出如下对话框

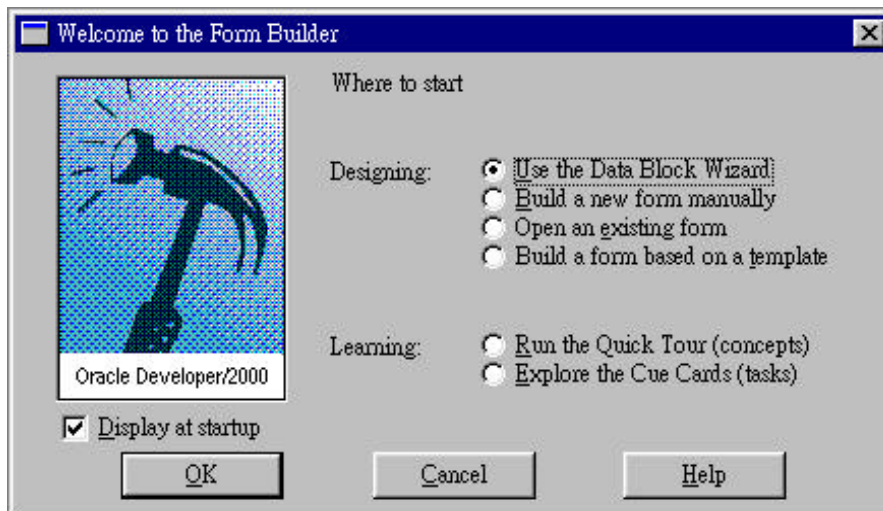


图 2

选择 ok,进入下一页

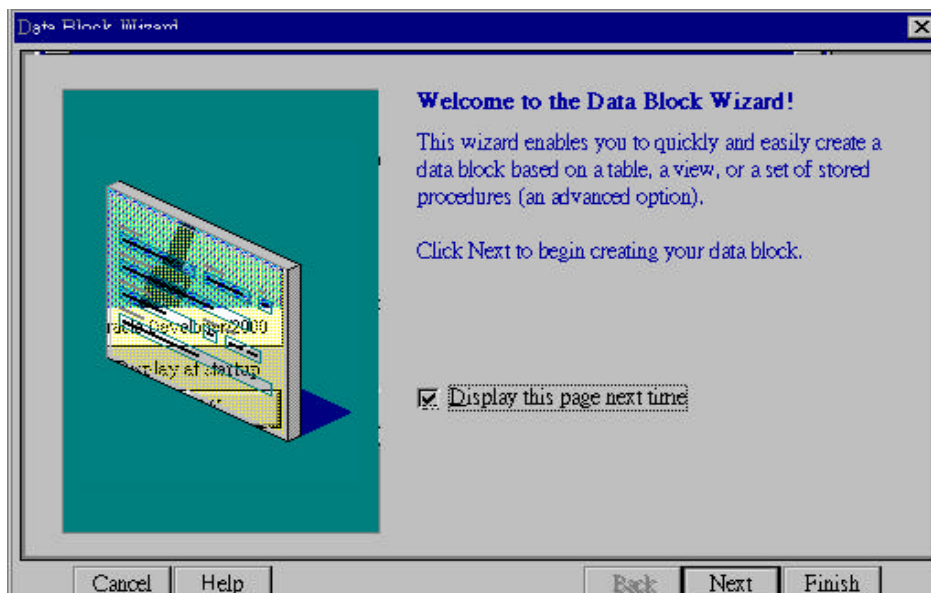


图 3

这个页面是介绍性页面,按 Next 进入下一步

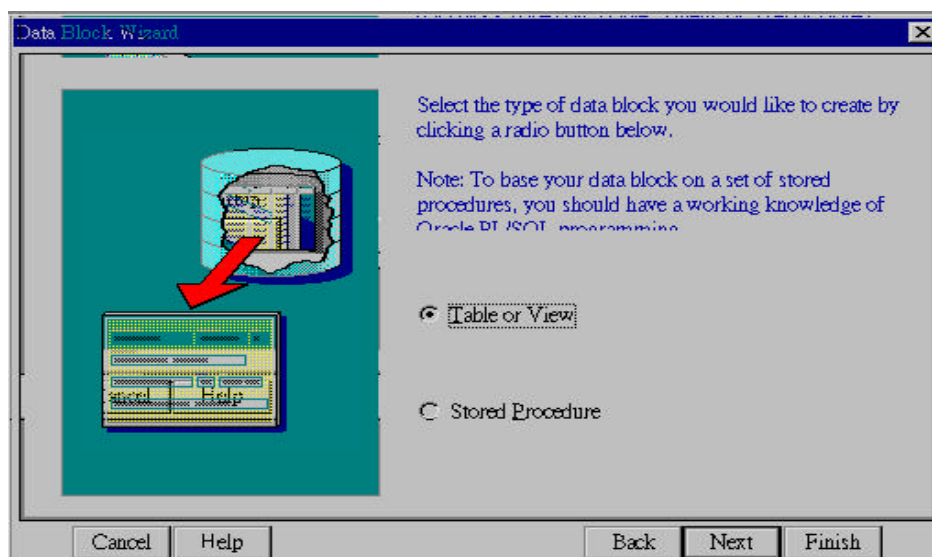


图 4

这个页面是让选择数据块(Datablock)的类型.我们暂时不做介绍,选择默认 Table or View 进入下一个页面;

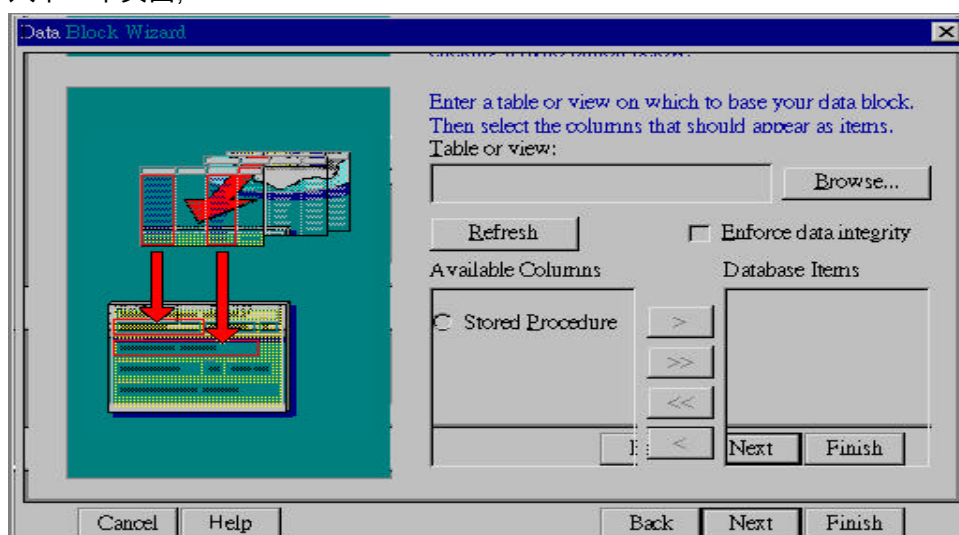


图 5

这里让大家选择表和要检索数据库的表的栏位,我们点击 Browse...显示

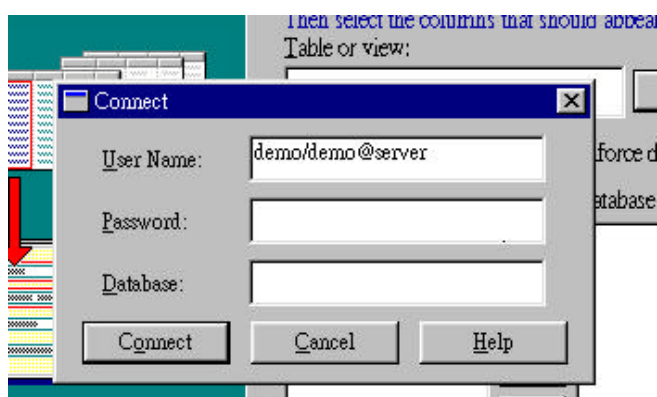


图 6

键入 demo/demo@server 按 Connect 进行数据库连接,

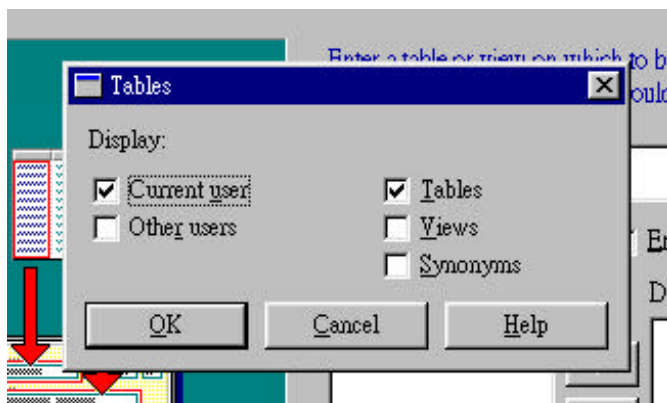


图 7

进入这个画面,由于我们查询的是表所以,我们直接用默认值,选择 ok

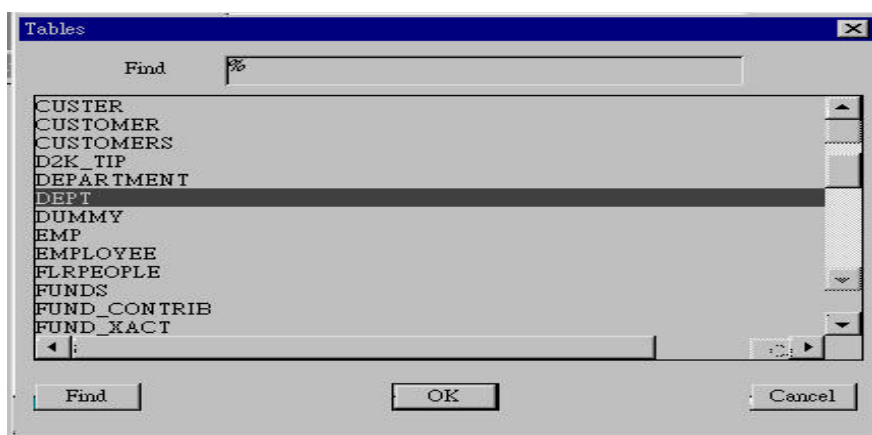


图 8

列表框显示 demo 用户下的所有的表,我们首先建立部门的,所以选上 Dept 表.按 ok.如下图所示:

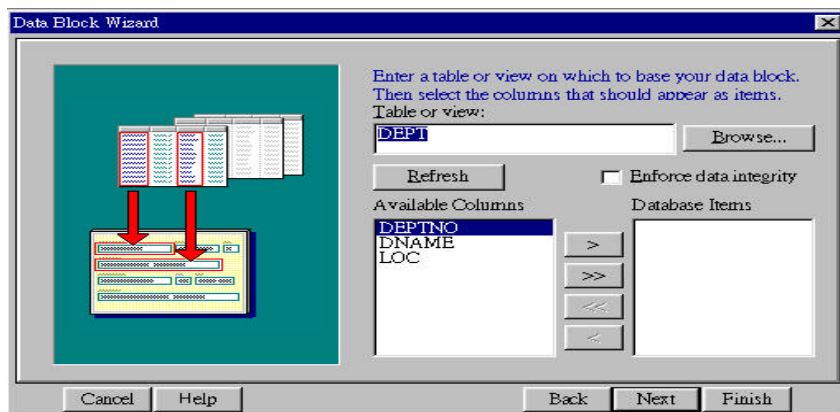


图 9

Available Columns 下的列表框显示了所有 Dept 表的栏位,右边的 Database Items 栏位表示我们在程序中需要检索的栏位,由于我们该表中所有的栏位都需要操作,所以我们按中间的 >> 按钮把左边所有的栏位移动到右边的列表框中,然后按 Next 进入下一步。

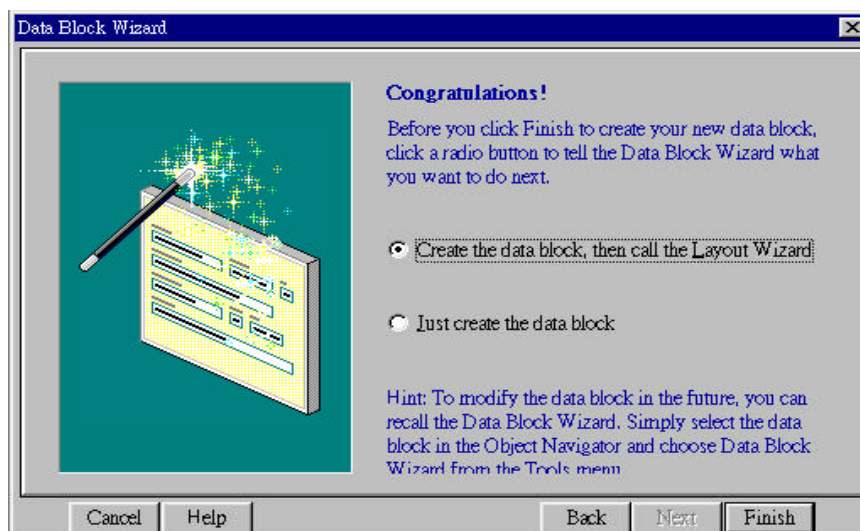


图 10

到这个页面我们用向导的形式完成了程序中 Datablock 的制作，下面我们所要做的就是将 Datablock 的栏位放置到 Canvase(画布)上。也就是说，Form 的实质是通过 Datablock 与数据行交互操作，而通过把 Datablock 中的栏位放置在 Canvas 上从而使程序与使用者建立了一个可视化的环境。打个比喻，就象北方的窗花一样，我们已经把红纸张已经剪好了一个一个的花的形状，就像我们用建立好了 Datablock，然后我们要做的就是将剪好的花贴在窗户上一样。

上图中有两个选项，一就是通过 Form 向导的形式建立整个画面的布局。二就是手工添加。我们在这里选择默认的 layout 向导形式。按 Finish。

显示下图：

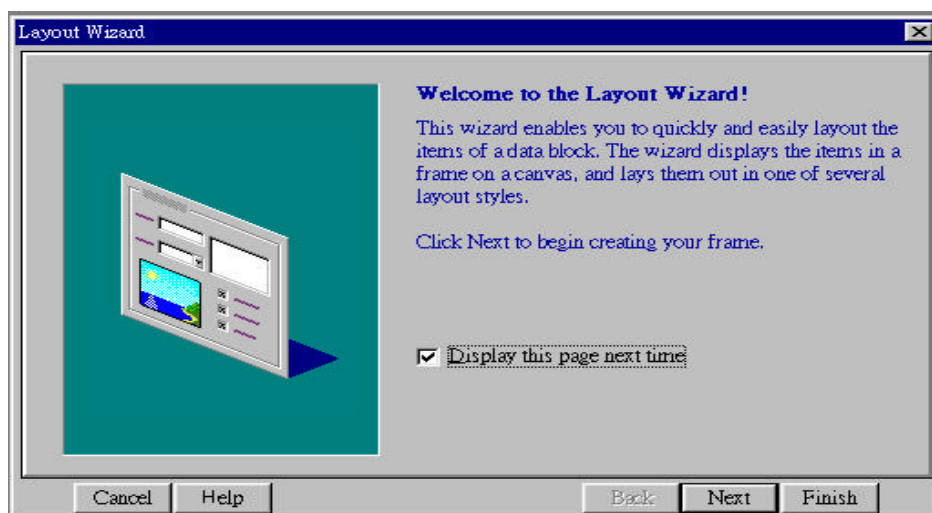


图 11

这里是一般性的介绍欢迎页面，可以不看，按 Next 按钮进入下一个页面：



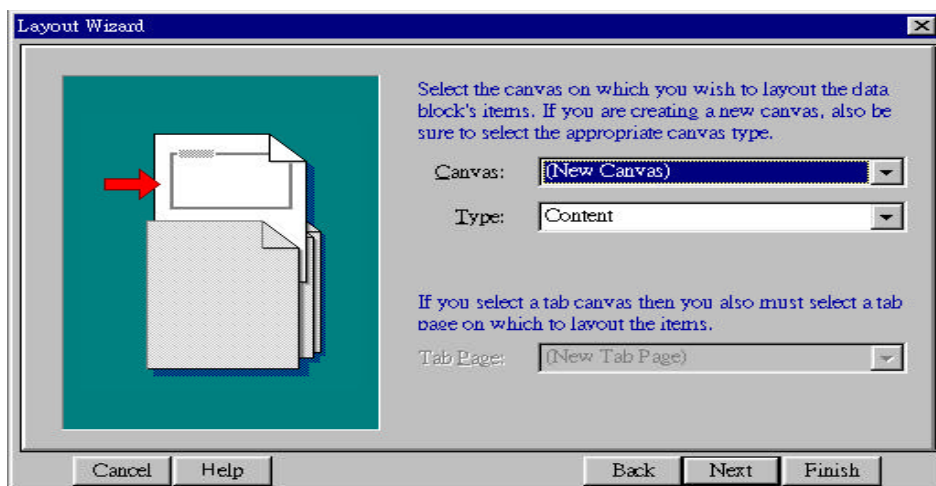


图 12

这个页面让我们选择我们要把显示的栏位放在哪个 Canvas 上，由于我们尚未建立任何 Canvas，所以我们只有 New Canvas 这个选项。下面的 Type 则是该 Canvas 类型。我们使用 Content，其它类型我们以后介绍。按 Next 进入下一步：

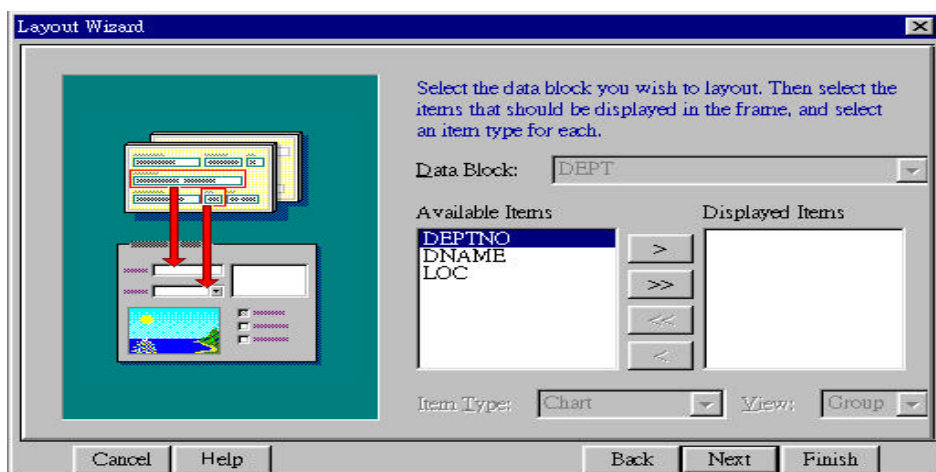


图 13

在左边的 Available Item 列表框内，我们列出了所有的在我们上面所建立 Datablock 时所含的栏位，由于我们所有的栏位都要显示出来，所以我们暗中见得的 >> 按钮全部转移到 Displayed Item 列表框中。按 Next 进入下一步。

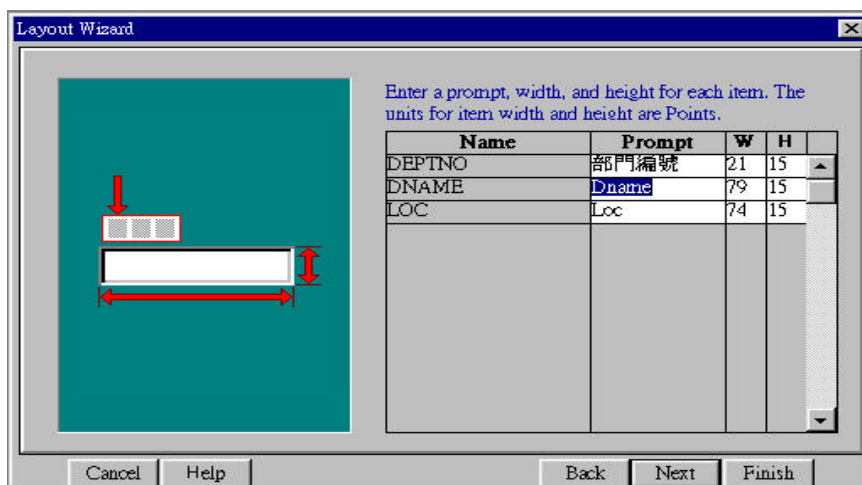


图 14

这一步是规定在 Canvas 中显示栏位的 Prompt(标题), 左边的图很形象的说明了 Prompt 代表甚么。这里还让大家规定栏位宽度, 和高度。建议大家不使用它的 Prompt, 在后面的系列文章中我们会做说明。为了本文以后说明方便我们还是采用了栏位的 Prompt, 修改里面的名称为合适的名称。我们的名称是 Deptno 命名为部门编号, Dname 命名为部门名称, Loc 则命名为地点。显示栏位的宽度和高度我们在完成向导后再调整, 按 Next 进入下一步

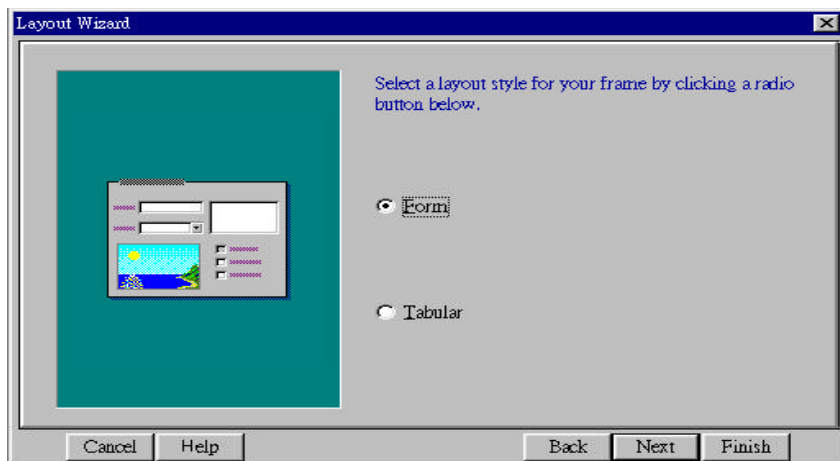


图 15

到这一步让大家选择 Canvas 上放置的这些栏位的显示方式。大家通过选择 Form 或者 Tabular 的同时查看左边的图形显示来体会这两种方式的区别, 一般来说 Form 风格通常用在单纪录的 Datablock 里面, Tabular 风格通常使用在多纪录的 Datablock 里面。由于我们程序的目的是通过选择 Dept 来显示该部门下的员工, 所以我们每次只显示一次纪录, 所以我们采用 Form 风格。按 Next 继续。

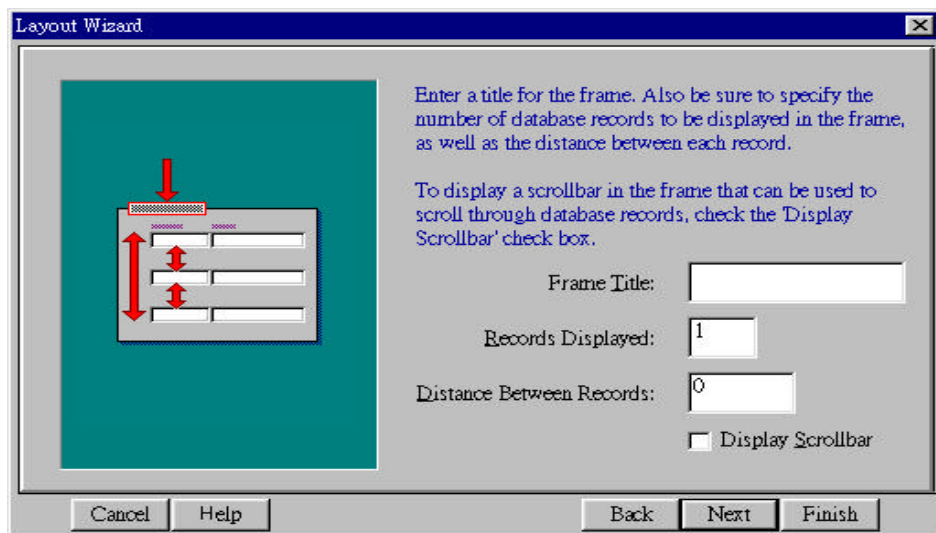


图 16

这个画面需要定置的是框架名称, 纪录的显示数, 以及每个纪录间显示的距离, 由于我们每次只有一条纪录, 所以我们全部采用默认值, 按 Next 继续。



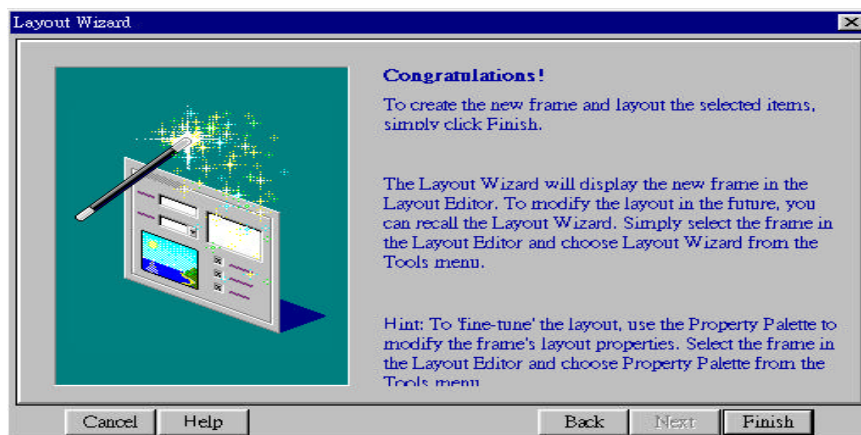


图 17

到这一步,恭喜,我们已经创建了基本的 Block 和 Canvas,按 Finish 完成进入主画面,这里是我们的 Object Navigator 环境,如果不在,请选择菜单 Tools 下的,Object Navigator 或者快捷键 F3 进入,

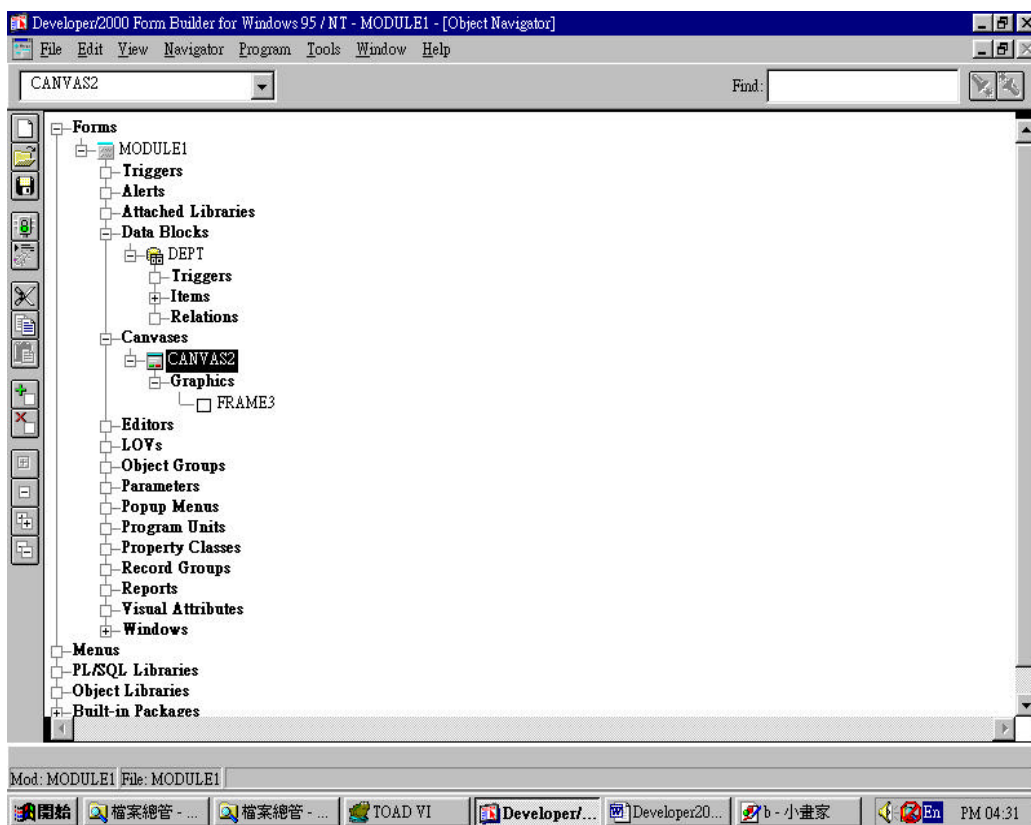
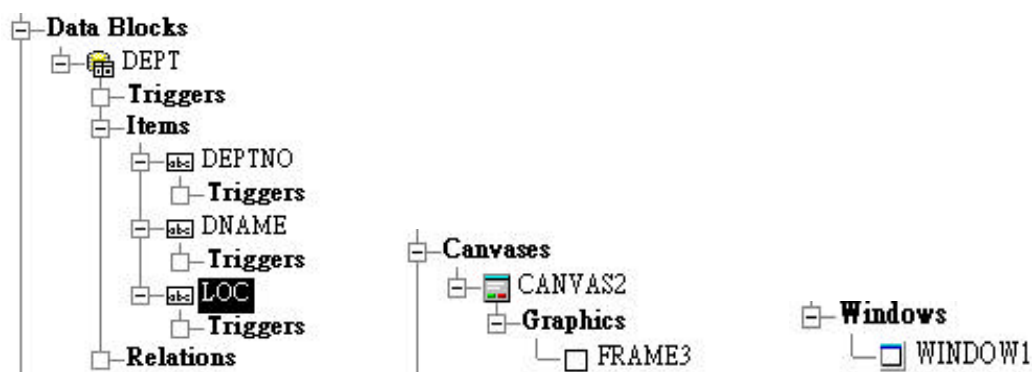
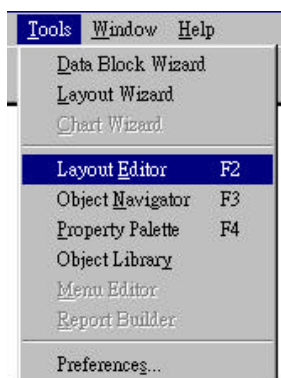


图 18

我们看看多出了那些东西:



这里多除了一个以该表的名称命名的 Datablock, 和一个 Canvas, 下面还多出了一个 Window, windows 是包含可视化 Form 对象的容器, 而 Canvas 是把可视化对象“画”在上面的一个接口, 为了看到这个 Canvas 上的东西, 我们必须把 Canvas 放在 windows 上. 我们在以后的章节详细介绍这三者的关系.. 下面我们看看我们的 Canvas 上有了那些东西. 他们自己的关系, 我们已经通过向导程序自动生成了



我们选择工具条上的 Tools 下的 Layout Editor, 或按 F2 快捷键进入 Layout 编辑器, 我们在这里, 可以调整 Canvas 上各个对象的位置等.

打开后如下图显示

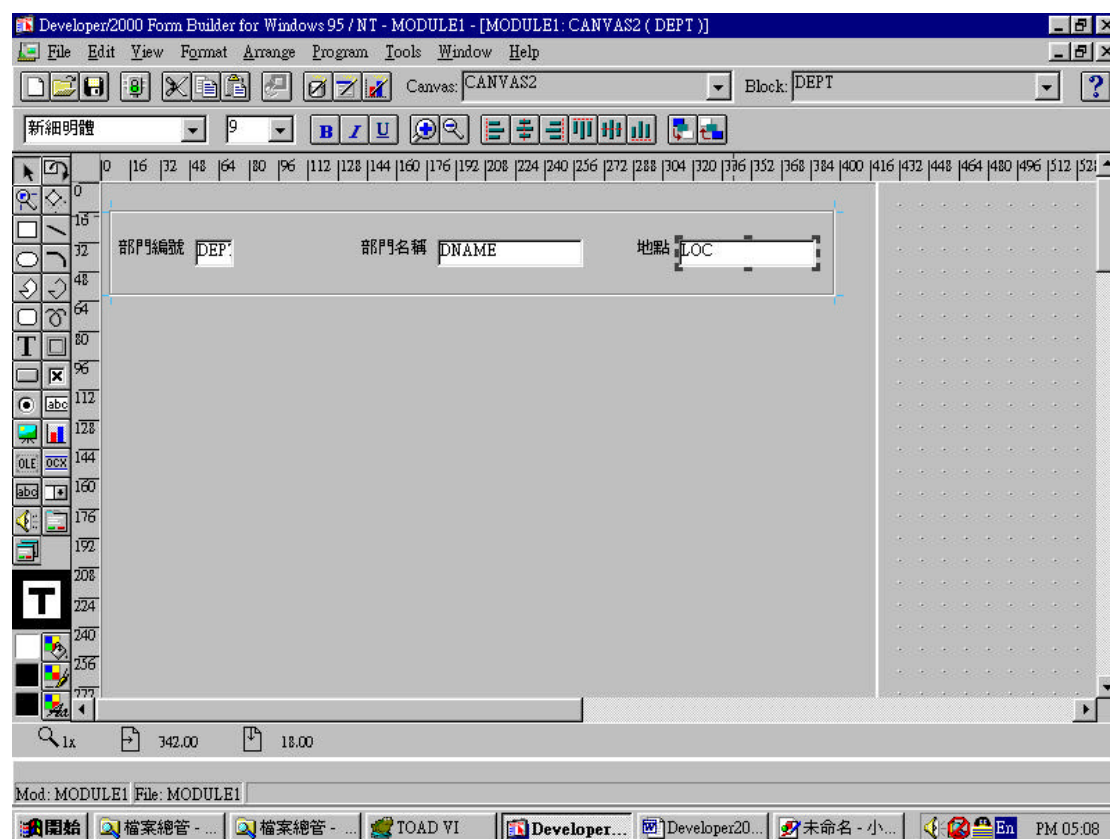


图 19

这里向导已经自动的帮助我们完成了也调整,大家可以看看,如果不满意,可以通过拖拉径行调整,具体的操作我们也以后介绍,中间是三个显示的栏位,最外面包围着的是一个 Frame 框. 这里我们开始修改外面的 Frame 框的属性, 点选住 Frame 框,选择菜单 Tools 下的 Property Palette 或按快捷键 F4 进入属性编辑器,这里显示了该 Frame 框的全部属性,仔细查找到如下图属性 Update Layout:

Layout Data Block	DEPT
Update Layout	Manually
Layout Style	Manually
Frame Alignment	Automatically
Single Object Alignment	Locked
	Start






其中有三个选项,Manually,Automatically,Locked,这里默认设置为 Automatically,意思是当更改了 Frame 框的大小,Layout 随即自动备更新,Manually,是指 Frame 跟新后,需要手工径行 Layout 更新,Locked 是把 Layout 锁定,不能径行更新.这里建议使用 Manually,这样对以后径行调整,非常方便.


好了,我们编译运行一下我们目前的成果把,选择菜单栏上的 Program 下的 Run Form 或者按快捷键 Ctrl+R,我们看看,目前有了甚么成果.



图 20

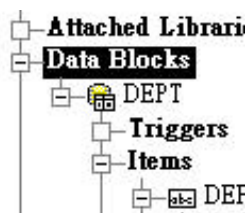
这里在我们程序运行在 Form 的 Runtime 环境中,Form Builder 已经帮我们指定了一些常见操作,如,查询,按条件查询,上一条纪录,下一条纪录,存盘,打印,插入,删除等,我们所要做的只是能使用它就可以了,这里是 Developer2000 工具集的优越之处,当然我们也可以自定义菜单,自己制作菜单,这在以后介绍.

好了,我们把当前光标放在部门编号的文字框中,然后按下按钮作为执行查询,这样我们就能看到文字框里数据库里取出了数据,按钮分别是,第一条,上一条,下一条,最后一条纪录的意思,如果操作有错误,会在下面状态栏上显示,如果不太清楚,可以在菜单 Help 下的 Display Error 显示详细错误,好了,大家可以熟悉一下这里的环境.

最后按退出程序,回到程序编辑环境.

下面我们将通过向导往程序里添加员工的信息显示那块,最终达到的效果如本文图 1 张图所示,操作中,通过我们选择部门纪录,随即显示该部门所有的员工的信息.

回到我们的编辑环境里,看看我们是不是在 Object Navigator 环境里,如果不在,请按 F3 进入.选住 DataBlocks,如下图所示:




按左边的工具条上的按钮,我们增加一个 Datablock,显示如下对话框



图 21

询问时用 Block 向导生成还是,使用手工,这里我们还是采用向导,这样,我们回到了图 3,按上面所述的顺序,再进行添加.注意要选择 Emp 表,  
由于 Dept 和 Emp 表之间有外键关系,所以要进行到下面图 22

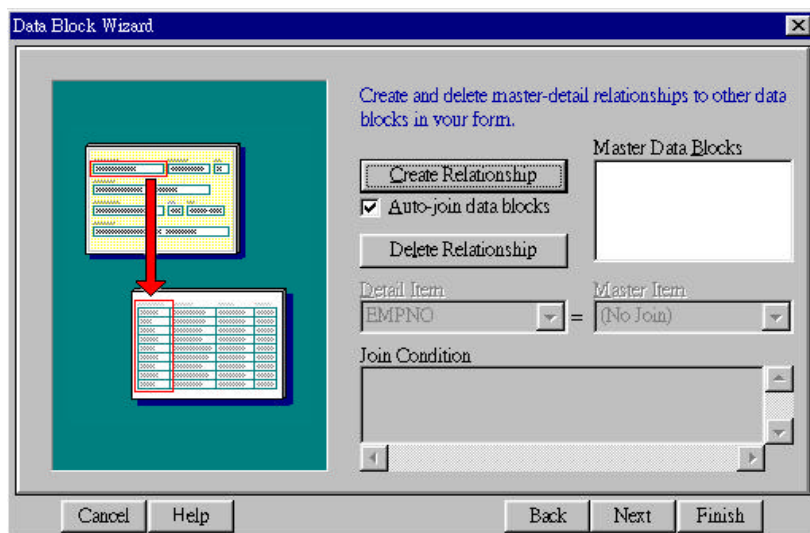
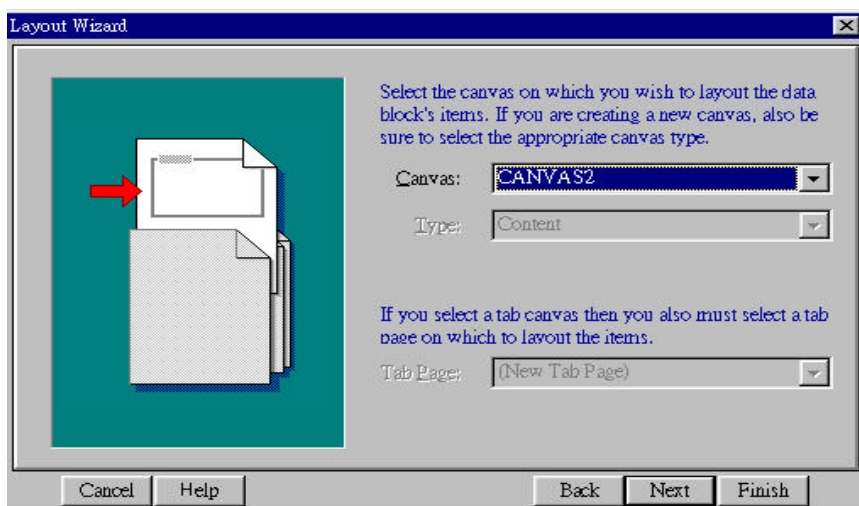
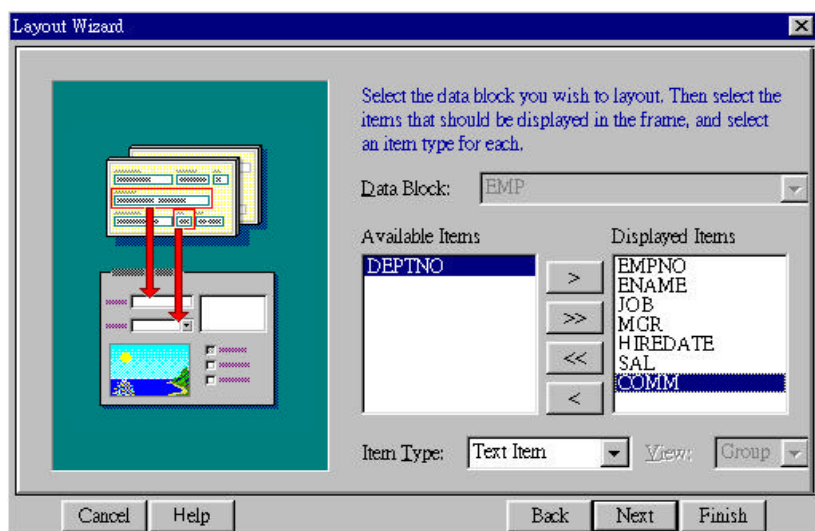


图 22

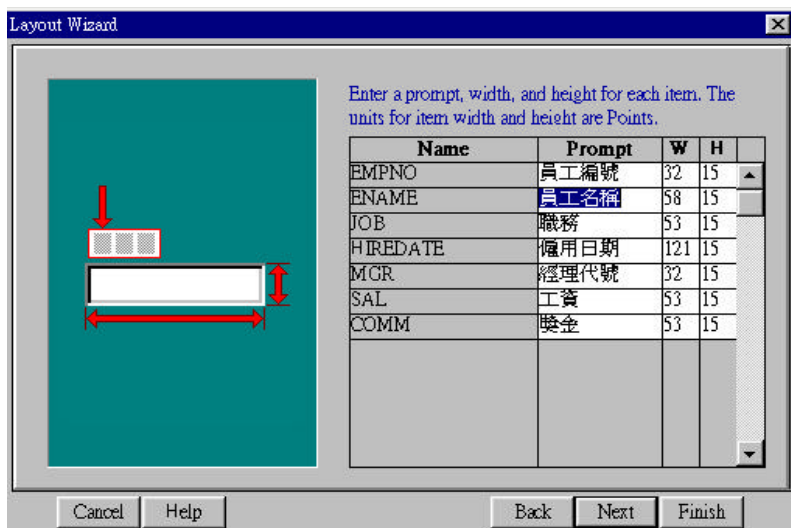
如果要建立一个主从关系的 Datablock,这里要创建一个关系(Relationship),我们点击 Create Relationship,由于 Dept 和 Emp 表之间有外键关系,所以我们可以看到里面有与表 Dept 连接的 Emp_foreign_KEY 这样的外键,我们选择,则在右边 Master Data Blocks 里面 Dept 字样,按 Next 继续,下面同上面创建 Dept 一样的画面,问是否要调用 Layout 的向导,我们选择默认,按 Next 继续,经过欢迎介绍界面后,到达如下图所示的页面



这里,我们准备把 Emp Datablock 中的栏位放在跟 Dept Datablock 一个 Canvas 上,我们不用选择,程序默认放在一个页面上,按 Next 继续到达下一个页面

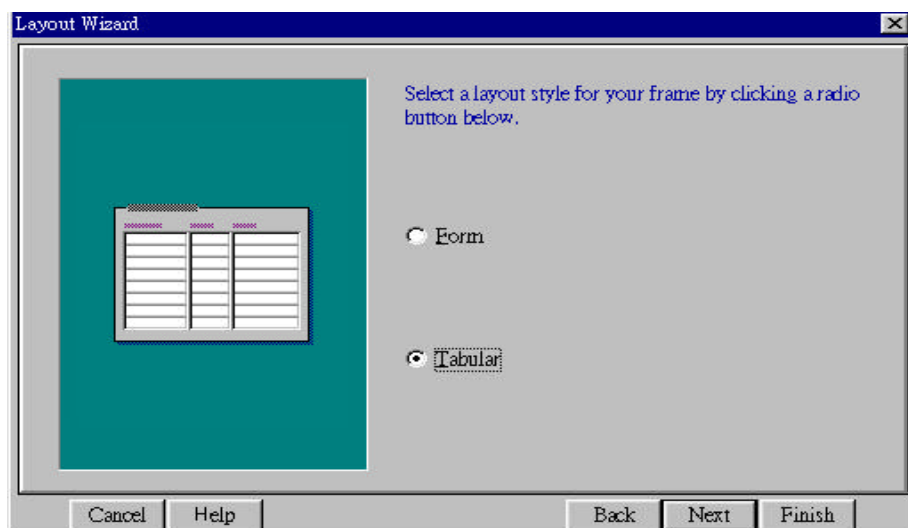


这个页面虽然 Deptno 在 Datablock 中,但由于它与 Dept 表的 Deptno 重复,我们不重复显示,所以不准备把 Emp Datablock 中的 Deptno 栏位也同时放在上面,所以除了 Deptno 外,其他都用 > 按钮归入 Displayed Item,按 Next 继续

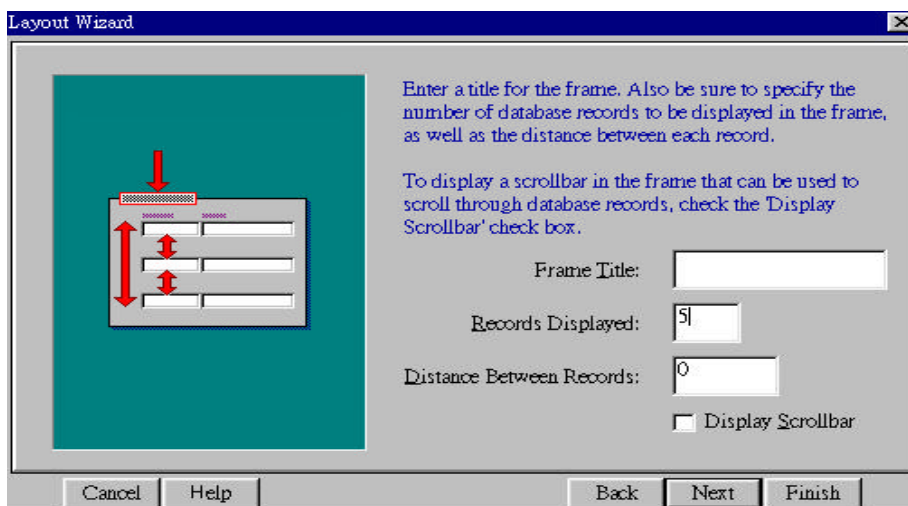


同样我们也修改 Prompt(标题),Empno 命名为员工编号,Ename 命名为员工姓名,Job 命名为职务,Hiiredate 命名为聘用日期,Mgr 命名为经理代号,Sal 命名为工资,Comm 命名为奖金,按 Next 继续





由于一个部门可以有多个员工,我们选择 Tabular 形式,按 Next 进入下一步




下一步我们修改一次显示的纪录数,我们显示 5 条,勾选上 Display Scrollbar 选框,按 Next 进入下一个页面,按 Finish,成功完成

进入 Object Navigator 我们发现了不光在 Datablock,Canvas 上多了东西,还在 Tigger 下面多了很多东西,我们暂时不管它,我们按 F4,进入 Layout Editor,看看我们一个基本的程序框架已经做好了,如果对上面的栏位放置不满意,可以调整,调整前特别建议也把 Emp 的 Frame 框的属性 Update Layout 改为 Manully,方便做调整.

Ok,一切大功告成,让我们运行一下,按菜单上的 Program 下的 Run Form 或者快捷键 Ctrl+R,一切搞定,把光标移动到部门编号的文本框里,然后按查询按钮,如下图


員工編號	員工名稱	職務	僱用日期	經理代號	工資	獎金
7782	CLARK	MANAGER	09-JUN-2081	7839	2450	
7839	KING	PRESIDENT	17-NOV-2081		5000	
7934	MILLER	CLERK	23-JAN-2082	7782	1300	

我们可以看到,ACCOUNTING 部门下面有三位员工.详细情况也列在表中.按下一条纪录按钮,我们会发现部门编号改动后,下面员工也随即改动,很方便.

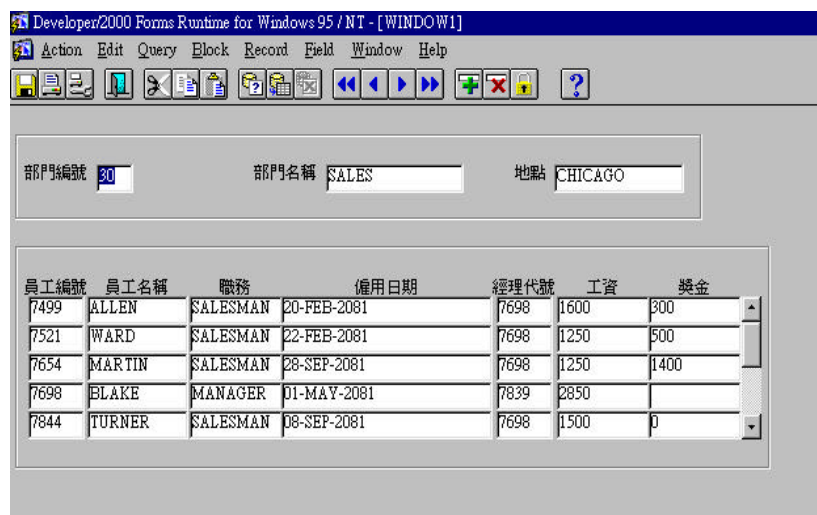
下面我们介绍一下 Runtime 环境中  按钮的用法,这个按钮名称是 Enter Query,我们可以利用它径行条件查询,点击该按钮,画面出现空白,其它很多按钮失效,此时进入 Enter Query 模式,如下图所示:

員工編號	員工名稱	職務	僱用日期	經理代號	工資	獎金

比如我们要查询部门编号为 30 的员工情况,我们在部门编号的文本框中输入 30




完成后,按下 Execute Query 按钮 ,查询完成,如下图所示:





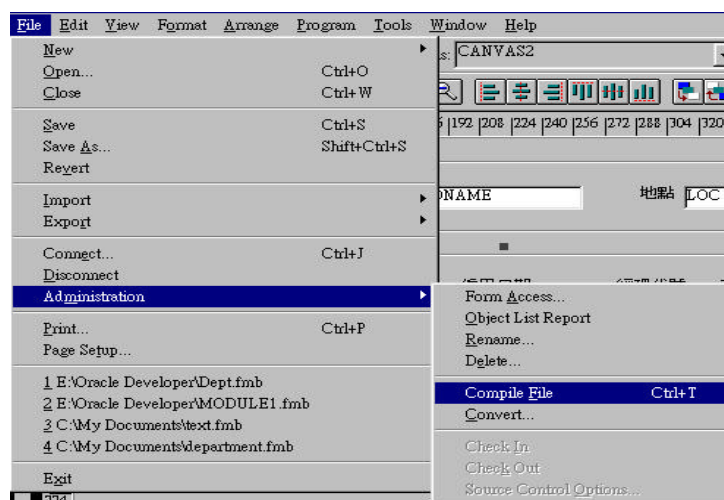
員工編號	員工名稱	職務	僱用日期	經理代號	工資	獎金
7499	ALLEN	SALESMAN	20-FEB-2081	7698	1600	300
7521	WARD	SALESMAN	22-FEB-2081	7698	1250	500
7654	MARTIN	SALESMAN	28-SEP-2081	7698	1250	1400
7698	BLAKE	MANAGER	01-MAY-2081	7839	2850	
7844	TURNER	SALESMAN	08-SEP-2081	7698	1500	0

我们就查询除了部门为 30 的员工情况,

按钮和为插入,删除纪录,在插入删除操作后,请一定要按按钮,才能把改动存入数据库中。

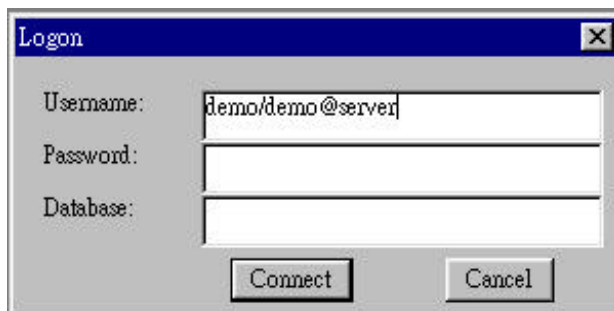
下面我们开始编译生成 Fmx 文件,才能在可视化的 Runtime 环境中执行,Form 的源文件则保存为 Fmb 文件,注意不能编译成 windows 下可执行的 exe 文件.所以 Form 的运行需要安装 Form 的 Runtime 环境.

先保存文件上述的源文件到一个目录,并命名,我们命名为 dept.fmb,然后执行菜单 File 下的 Administration 下的 Compile File.如图所示



完成后,我们将再同一个目录下看到已经生成了与源文件同名的 FMX 文件,到这个地方,我们完成了该程式的生成过程,该程式还有很多值得修改的地方,我将围绕这个程式展开对 Developer2000 的讲座.

我们来运行一下刚才的程式,双击 Dept.fmx,程序跳出对话框,如下图:



输入需要改程序连接数据库的用户名,密码和别名,即可进入我们刚才的画面

通过上面的介绍,大家应该对 Form 的开发过程有了很大的认识,不用输入任何 sql 语句,或者任何 PL/SQL 程序便完成了程式的编写工作,可以大大提高开发效率,当然我们可以利用手工来编写更为复杂的程序,或者给程序添加更多的功能,这是我们以后要讨论的.

最后请大家对本文一些错误之处提出意见,谢谢!

下一章讲开始介绍 developer2000 的安装和构成.以及程序构造原理,请大家期待。

## 查看表结构的 script

Littlelolo

对初学 Oracle 的人来说，会在一些简单的问题面前不知所措，比方说，表结构是什么，表的主键，外键是什么，有哪些索引，等等。

以下是我在 sql*plus 写的一段关于 table 的 script，希望能抛砖引玉，对初学者有帮助。

本 script 可以返回 table structure, primary key, foreign key, index, tablespace,.

方法：

1)保存以下语句在文件中，eg: c:\plus\table.sql

2)在 sql*plus 中运行，eg: SQL>@ c:\plus\table.sql

3)在提示后输入 table name, table's owner

```
/****** start *****/
```

```
set serveroutput on size 1000000 verify off feedback off echo off
```

```
accept table_name prompt 'Please input the table name: '
```

```
accept table_owner prompt 'Please input the owner: '
```

```
declare
```

```
cursor cur_columns(iv_table_name varchar2,
```

```
iv_owner varchar2) is
```

```
select column_name,
```

```
data_type,
```

```
data_length,
```

```
data_precision,
```

```
data_scale,
```

```
nullable,
```

```
column_id,
```

```
data_default
```

```
from all_tab_columns
```

```
where table_name = iv_table_name
```

```
and owner = iv_owner
```

```
order by column_id;
```

```
rec_columns cur_columns%rowtype;
```

```
--
```

```
cursor cur_constraint(iv_table_name varchar2,
```

```
iv_owner varchar2) is
```

```
select constraint_name,
```

```
constraint_type,
```

```
search_condition,
```

```
r_owner,
```

```
r_constraint_name,
```

```
delete_rule,
```

```
status
```

```
from all_constraints
```

```
where table_name = iv_table_name
and owner = iv_owner;
--
cursor cur_cons_columns(iv_constraint_name varchar2,
iv_owner varchar2) is
select column_name,
table_name
from all_cons_columns
where constraint_name = iv_constraint_name
and owner = iv_owner
order by position;
--
cursor cur_index(iv_table_name varchar2,
iv_owner varchar2) is
select index_name,
lower(tablespace_name) tablespace_name
from all_indexes
where table_name = iv_table_name
and owner = iv_owner;
--
cursor cur_ind_columns(iv_index_name varchar2,
iv_owner varchar2) is
select lower(column_name) column_name
from all_ind_columns
where index_name = iv_index_name
and index_owner = iv_owner
order by index_name, column_position;
--
v_table_name all_tables.table_name%type;
v_owner all_tables.owner%type;
v_tablespace_name all_tables.tablespace_name%type;
v_pre_column_name all_tab_columns.column_name%type;
v_pre_column varchar2(200);
v_constraint varchar2(200);
v_index varchar2(200);
v_r_table_name all_cons_columns.table_name%type;
begin
v_table_name := upper('&table_name');
v_owner := upper('&table_owner');
--
select lower(tablespace_name)
into v_tablespace_name
from all_tables
where table_name = v_table_name
```

```

and owner = v_owner;
-- -----
-- get all columns --
-- -----
open cur_columns(v_table_name,v_owner);
loop
fetch cur_columns into rec_columns;
--
if v_pre_column_name is not null then
if v_pre_column_name = lower(rec_columns.column_name) then
dbms_output.put_line(v_pre_column);
dbms_output.put_line(' ' || v_tablespace_name);
dbms_output.put_line('/');
elsif rec_columns.column_id = 2 then
dbms_output.put_line('create table ' || lower(v_table_name) || ' (');
dbms_output.put_line(v_pre_column || ',');
else
dbms_output.put_line(v_pre_column || ',');
end if;
end if;
--
exit when cur_columns%notfound;
--
v_pre_column_name := lower(rec_columns.column_name);
v_pre_column := rpad(lower(rec_columns.column_name), 31, ' ');
if rec_columns.data_type = 'NUMBER' then
if rec_columns.data_precision is null then
v_pre_column := v_pre_column || 'number';
elsif rec_columns.data_scale = 0 then
v_pre_column := v_pre_column || 'number(' || to_char(rec_columns.data_precision) || ')';
else
v_pre_column := v_pre_column || 'number(' ||
to_char(rec_columns.data_precision) || ',' ||
to_char(rec_columns.data_scale) || ')';
end if;
elsif instr(rec_columns.data_type, 'CHAR') > 0 then
v_pre_column := v_pre_column || lower(rec_columns.data_type) || '('
|| to_char(rec_columns.data_length) || ')';
else
v_pre_column := v_pre_column || lower(rec_columns.data_type);
end if;
--
if rec_columns.data_default is not null then
v_pre_column := rpad(v_pre_column, 46, ' ') || ' default ' ||

```

```
rtrim(rec_columns.data_default);
end if;
--
if rec_columns.nullable = 'N' then
if length(v_pre_column) > 46 then
v_pre_column := v_pre_column || ' not null';
else
v_pre_column := rpad(v_pre_column, 46, ' ') || ' not null';
end if;
end if;
--
end loop;
close cur_columns;
--
for rec_c in cur_constraint(v_table_name,v_owner) loop
if rec_c.constraint_type = 'P' then
v_constraint := ' (';
for rec_column in cur_cons_columns(rec_c.constraint_name,
v_owner) loop
v_constraint := v_constraint || lower(rec_column.column_name) || ', ';
end loop;
v_constraint := substr(v_constraint, 1, length(v_constraint)-2) || ')';
dbms_output.put_line('alter table ' || lower(v_table_name));
dbms_output.put_line(' add constraint ' || rec_c.constraint_name || ' primary key');
dbms_output.put_line(v_constraint);
--
select lower(tablespace_name)
into v_tablespace_name
from all_indexes
where table_name = v_table_name
and index_name = rec_c.constraint_name
and owner = v_owner;
dbms_output.put_line(' using index tablespace ' || v_tablespace_name);
dbms_output.put_line('/');
elsif rec_c.constraint_type = 'R' then
v_constraint := ' (';
for rec_column in cur_cons_columns(rec_c.constraint_name,
v_owner) loop
v_constraint := v_constraint || lower(rec_column.column_name) || ', ';
end loop;
v_constraint := substr(v_constraint, 1, length(v_constraint)-2) || ')';
dbms_output.put_line('alter table ' || lower(v_table_name));
dbms_output.put_line(' add constraint ' || rec_c.constraint_name || ' foreign key');
dbms_output.put_line(v_constraint);
```

```
v_constraint := '(';
for rec_column in cur_cons_columns(rec_c.constraint_name,
v_owner) loop
v_constraint := v_constraint || lower(rec_column.column_name) || ',';
v_r_table_name := lower(rec_column.table_name);
end loop;
v_constraint := substr(v_constraint, 1, length(v_constraint)-2) || ')';
dbms_output.put_line(' references ' || v_r_table_name);
dbms_output.put_line(v_constraint);
dbms_output.put_line('/');
elsif rec_c.constraint_type = 'U' then
v_constraint := '(';
for rec_column in cur_cons_columns(rec_c.constraint_name,
v_owner) loop
v_constraint := v_constraint || lower(rec_column.column_name) || ',';
end loop;
v_constraint := substr(v_constraint, 1, length(v_constraint)-2) || ')';
dbms_output.put_line('alter table ' || lower(v_table_name));
dbms_output.put_line(' add constraint ' || rec_c.constraint_name || ' unique');
dbms_output.put_line(v_constraint);
--
select lower(tablespace_name)
into v_tablespace_name
from all_indexes
where table_name = v_table_name
and index_name = rec_c.constraint_name
and owner = v_owner;
dbms_output.put_line(' using index tablespace ' || v_tablespace_name);
dbms_output.put_line('/');
elsif rec_c.constraint_type = 'C' and
instr(rec_c.search_condition, 'IS NOT NULL') = 0 then
dbms_output.put_line('alter table ' || lower(v_table_name));
dbms_output.put_line(' add constraint ' || rec_c.constraint_name || ' check');
dbms_output.put_line(' (' || ltrim(rtrim(rec_c.search_condition)) || ')');
dbms_output.put_line('/');
end if;
end loop;
--
for rec_i in cur_index(v_table_name,v_owner) loop
dbms_output.put_line('create index ' || rec_i.index_name);
v_index := ' on ' || lower(v_table_name) || '(';
for rec_c in cur_ind_columns(rec_i.index_name, v_owner) loop
v_index := v_index || rec_c.column_name || ',';
end loop;
```

```
v_index := substr(v_index, 1, length(v_index)-2) || ' ';
dbms_output.put_line(v_index);
dbms_output.put_line(' tablespace '||rec_i.tablespace_name);
dbms_output.put_line('/');
end loop;
exception
when no_data_found then
dbms_output.put_line('No this table!');
when others then
dbms_output.put_line(sqlcode||sqlerrm);
end;
/
set serveroutput off verify on feedback on
/***** end *****/
```



## OCP 不完全指南

作者:easyfree

1. OCP 及考试科目介绍
2. 科目章节分布明细
3. 教材列表
4. 已有资源列表
5. 考试简介
6. 相关网站资源

### 一. OCP 及考试科目介绍

OCP 是 Oracle Certified Professional 的缩写,意即 Oracle 认证专家,是 Oracle 公司于 1997 年推出的国际上承认的 Oracle 水平证书。

目前 Oracle 已有的考试有:

#### 1、Oracle Database Administrator

ORACLE7.3 4 门

1Z0-001 Introduction to Oracle:SQL and PL/SQL 中/英

1Z0-003 Oracle7.3: Database Administrator 英

1Z0-004 Oracle7.3: Performance Tuning 英

1Z0-005 Oracle7.3: Backup and Recovery 英

ORACLE8 5 门 2000/08 已作废

1Z0-001 Introduction to Oracle:SQL and PL/SQL 中/英

1Z0-013 Oracle8: Database Administrator 英

1Z0-014 Oracle8: Performance Tuning workshop 英

1Z0-015 Oracle8: Backup and Recovery workshop 英

1Z0-016 Oracle8: Network Administration 英

ORACLE8i 5 门

1Z0-001 Introduction to Oracle:SQL and PL/SQL 中/英

1Z0-023 Oracle8i: Database Administrator 英

1Z0-024 Oracle8i: Performance Tuning workshop 英

1Z0-025 Oracle8i: Backup and Recovery workshop 英

1Z0-026 Oracle8i: Network Administration 英

ORACLE9i

升级考试

1Z0-010 Oracle8:New Features for Administration 英 Oracle7.3 --> Oracle8 升级考试

1Z0-020 Oracle8i:New Features for Administration 英 Oracle8 --> Oracle8i 升级考试

#### 2、Oracle Application Developer

Release 1

1Z0-001 Introduction to Oracle:SQL and PL/SQL 中/英

1Z0-101 Developer PL/SQL Program Units 英

1Z0-111 Developer/2000 Form4.5 (1) 英

1Z0-112 Developer/2000 Form4.5 (2) 英

1Z0-113 Developer/2000 Report2.5 英

Release 2

1Z0-001 Introduction to Oracle:SQL and PL/SQL 中/英

1Z0-101 Developer PL/SQL Program Units 英

1Z0-121 Developer/2000 Build Forms (1) 英

1Z0-122 Developer/2000 Build Forms (2) 英

1Z0-123 Developer/2000 Build Report 英

升级考试

1Z0-110 Oracle Developer ,Release 2 New Features ?????

### 3、Oracle Database Operator

1Z0-401 Database Operator

### 4、Oracle Java Developer

Level 1: Sun Certified Programmer

1Z0-501: Sun Certified Programmer for the Java 2 Platform

Level 2: Certified Solution Developer

1Z1-502: Oracle JDeveloper: Develop Database Applications with Java (Oracle JDeveloper, Release 2)

or

1Z1-512: Oracle JDeveloper: Develop Database Applications with Java (Oracle JDeveloper, Release 3)

1Z0-503: Object Oriented Analysis and Design with UML

Level 3: Certified Enterprise Developer

1Z0-504: Enterprise Connectivity with J2EE

1Z0-505: Enterprise Development on the Oracle Internet Platform

### 5、Oracle Financial Applications Consultant

1Z0-210: Financial Management R11

1Z0-220: Applied Technology R11

1Z0-230: Procurement R11

or

1Z0-240: Order Fulfillment

由于 ORACLE 公司会根据具体情况调整其考试内容,所以详细的信息参考:  
<http://education.oracle.com>,以 Oracle 公司官方网站的内容为准

通用信息:

#### 注册:

(1) Oracle 公司授权 prometric 公司进行 OCP 认证考试,考生必须注册 prometric,获取全球唯一的 ID 后才能参加 OCP 考试

(2) Hong Kong, Taiwan, Australia, Indonesia, Malaysia, New Zealand, The Philippines, Thailand, and Singapore 等地的考生可以在网站 [www.2test.com](http://www.2test.com) 进行在线注册,中国大陆考生访问:[www.sylvan.com.cn](http://www.sylvan.com.cn) (目前无法使用)

(3) 可以通过国内的 prometric 授权考点进行注册

(4) 需要修改个人信息(通信地址,email,电话等)时可以:

A 通知考试中心修改

B 直接发 mail 给 [rebecca.dimaio@prometric.com](mailto:rebecca.dimaio@prometric.com) 或者打热线电话 1.800.891.EXAM (3926)

目前我没有找到可以在网上修改个人信息的方法,如果哪位 fans 知道,请提供一下,谢谢

**考试：**

- (1) OCP 考试提供 Upgrade Path 和 Core Path 两种方式,一般升级考试难一些
- (2) OCP 考试的科目顺序没有规定,仅对升级要求必须逐级递增
- (3) 考试提前预约的时间各个地区和考试中心不完全相同,一般中国大陆为 3 天,可以向各自的考试中心咨询
- (4) OCP 考试要求当某门考试失败后必须间隔 30 天以上才允许重考
- (5) 每门考试成绩在考试完成后立刻打印交给考生保存(注意必须加盖 prometric 公司考试印章), 考试例外(参见 Oracle 公司考试指南)
- (6) 当 Oracle 宣布某项认证考试退役(过期),考生仍然可以有 6 个月的时间完成该认证考试;但如果在截止日期前没有完成升级考试,则必须通过全部考试课程才能获取新的证书
- (7) 从第一门考试时间开始必须一年内完成全部认证课程的考试 -- 没有找到官方说明,考试中心对此也没有明确说明,个人认为不重要

**证书：**

证书应在通过所以认证课程考试之后 30 天内送达,否则:

A 向考试中心咨询

B 直接发 mail 给 fulfillment@prometric.com,提供姓名,candidate ID,当前 mail 地址

**培训课程简介：****1Z0-001 Introduction to Oracle:SQL and PL/SQL****课程描述：**

从基础 SQL 语句开始学习,帮助学员了解如何使用 Oracle 的工具和方法来满足实际的企业信息需求.

学习如何编写 PL/SQL 块,如何编写 Procedure 和 Trigger.

学习如何设计和建立数据库结构,如何保存,检索,更新和显示关系型数据库中的数据.

**课程内容：**

SQL 命令

限定和排序数据

Singh-Row 函数

从多个表中显示数据使用组函数合计数据

子查询/多列子查询/用 SQL*plus 生成满足的报表操作数据

创建并管理表/约束/建视图

其它数据对象(序列、索引)用户

SQL 讨论/PL/SQL 中的变量

PL/SQL 对数据进行操作

结构控制 PL/SQL 记录和表处理

游标的处理

例外的处理.

适用对象

数据库系统管理员

数据库设计人员

网页设计人员

应用开发人员

## 1Z0-023 Oracle8I Database Administration

### 课程描述：

从数据库的基本结构着手,引导学员掌握 Oracle 数据库的物理结构,逻辑结构.并从中掌握 Oracle 数据库的管理方法和技巧.

学习使用管理工具来启动、关闭数据库,创建数据库,管理文件和数据存储,并管理用户及其权限.

通过大量的问题分析和解决使得 DBA 将获得必要的知识和技能来设置,维护和解决 Oracle 数据库故障.

### 课程内容：

数据库结构

数据库的安装配置

常用数据字典视图和标准化

控制文件的维护

Redo log 文件的维护

表空间和数据文件的维护

回滚段的管理

临时段的管理

表的管理

索引的管理

数据的完整性维护

使用簇式索引结构的表

装载和重组数据

用户和角色的管理

参数文件的管理

权限的管理

审计

国际语言支持的使用.

适用对象

系统管理员

数据库管理员

网络管理员

系统设计人员

应用开发人员.

## 1Z0-025 Oracle8I Backup and Recovery Workshop

### 课程描述：

介绍数据库的备份和恢复策略的关键任务的规划和实施.

讲述备份和恢复技巧,并测试各种不同类型的备份,故障复原和恢复方案.

讲述适用于 Oracle7 和 Oracle8 数据库环境的备份,恢复操作以及 Oracle8 RMAN (Recovery Manager) .

通过实际操作,练习基于关键任务企业业务所要求的备份方法.

对 Oracle 的各种不同的备份方法作比较,学员将能利用学习到的这些方法选择/制订出一个最能适合于本部门的备份方案.

### 课程内容：

备份与恢复的考虑  
ORACLE 恢复的结构与过程  
ORACLE 的备份与恢复配置  
不使用 ORACLE RMAN 的物理备份  
不用 ORACLE RMAN 的完全恢复  
用归档进行不完全 ORACLE 恢复  
ORACLE 的导入与导出功能  
附加的恢复问题  
ORACLE 的修复故障功能  
ORACLE 的 RMAN 一览  
ORACLE Recovery Catalog 的建立和维护  
用 RMAN 备份  
用 RMAN 重建和恢复  
ORACLE 的备用数据库.  
适用对象  
系统管理员  
网络管理员  
数据库管理员  
系统设计人员  
应用开发人员  
技术支持专业人员

### 1Z0-024 Oracle8I Performance Tuning Workshop

#### 课程描述：

向学员解释一系列可以用来改进 Oracle8 服务器性能的调整步骤.

本课程注重于数据库而非操作系统性能问题.通过对各种不同类型的性能分析例子和数据分  
析,学员将获得实际的数据库调整经验.

利用各种工具,学员还将学习在管理数据库时如何识别和排除故障以及解决相关问题.

#### 课程内容：

性能调整概叙  
ORACLE 的报警与跟踪文件  
工具与动态性能视图  
调整共享池  
调整高速缓冲区  
调整日志缓冲区  
数据库配置与 I/O 问题  
有效使用 ORACLE 数据块  
优化排序操作  
调整回滚段  
监视、检测锁冲突  
对不同的应用有不同的 SQL 和调整考虑  
管理一个混合工作量  
使用 ORACLE Expert 调整  
多线程服务调整论点

适用对象  
系统管理员  
网络管理员  
数据库管理员  
项目经理  
系统设计人员  
应用开发人员  
技术支持专业人员

### 1Z0-026 SQL*Net and Network Administration

#### 课程描述：

了解关于 Net8 的组件信息，并通过学习如何配置简单到较复杂的 Net8 环境。  
通过展示潜在的网络安全，以及 Oracle 对这些问题的可选的安全解决方案。  
提供 Net8 调试工具和调试信息的解释。

#### 课程内容：

Net8 体系结构介绍  
Net8 服务器端配置  
集中命名管理  
多线索的使用和配置  
连接管理器的配置  
网络安全  
适用对象  
系统管理员  
网络管理员  
数据库管理员

## 二. 科目章节分布明细

代码 科目 总题数 通过题数 通过率 考试时间(分钟)

-----

1Z0-001	Introduction to Oracle:SQL and PL/SQL	57	39	68%	120
1Z0-003	Oracle 7.3: Database Administration	60	90		
1Z0-004	Oracle 7.3: Performance Tuning Workshop	60	90		
1Z0-005	Oracle 7.3: Backup and Recovery Workshop	60	90		
1z0-010	Oracle8:New Features for Administration	60	90		
1z0-013	Oracle8:Database Administration	63	90		
1z0-014	Oracle8:Performance Tuning Workshop	61	90		
1z0-015	Oracle8:Backup and Recovery Workshop	60	90		
1z0-016	Oracle8:Network Administration	62	90		
1Z0-020	Oracle8i:New Features for Administration	59			
1.	Java in the Database	1			
2.	Optimizer and query	7			
3.	Summary management	6			
4.	Indexes and Index-Organized Tables	3			
5.	Object Relational Features and LOBs	3			

6. Partitioning improvements 6
7. ORACLE Universal Installer:Migration and Upgrade 3
8. Tablespace Management 6
9. Database Resource Manager 2
- 10.Manageability Enhancements 3
- 11.Availability and Recoverability Enhancements 8
- 12.Features of NET8 1
- 13.SQL*Plus,PL/SQL and NLS Enhancements 3
- 14.Database Security 2
- 1Z0-023 Oracle8i: Database Administrator 64 42 66% 90
1. Oracle Architecture Components 4
2. Getting Started with the Oracle Server 2
3. Managing an Oracle Instance 4
4. Creating a Database 2
5. Creating Data Dictionary Views and Standard Packages 4
6. Maintaining the Control File 3
7. Maintaining the Redo log files 5
8. Managing Tablespace and Data files 7
9. Storage Structure and Relationships 3
- 10.Managing Rollback Segments 3
- 11.Managing Tables 4
- 12.Managing Indexes 2
- 13.Maintaining Data Integrity 2
- 14.Loading Data 3
- 15.Reorganizing Data 2
- 16.Managing Password Security and Resources 4
- 17.Managing Users 3
- 18.Managing Privileges 2
- 19.Managing Roles 2
- 20.Using National Language Support 3
- 1Z0-024 Oracle8i: Performance Tuning workshop 57 38 67% 90
1. Tuning overview 1
2. Oracle alert & trace files 2
3. Util. & dynamic perf. views 3
4. Tuning the shared pool 7
5. Tuning the buffer cache 7
6. Tuning the redo log buffer 3
7. DB conf. & IO issues 7
8. Using oracle blocks eff. 4
9. Optimizing sort oper. 4
- 10.Tuning rollback seg. 4
- 11.Mon. & det. lock contention 5
- 12.SQL tuning 5
- 13.Managing mix. workload 2

- 14.Oracle expert 1
- 15.MTS 2
- 1Z0-025 Oracle8i: Backup and Recovery workshop 60 42 70% 90
  - 1. recovery structure and process 5
  - 2. backup and recovery configuration 7
  - 3. physical backup 8
  - 4. complete recovery without rman 5
  - 5. incomplete recovery with archiving 3
  - 6. export and import 3
  - 7. additional recovery issue 3
  - 8. utilities for troubling 4
  - 9. rman overview 2
  - 10.recovery catalog creation and maintenances 6
  - 11.backup using rman 5
  - 12.recovery using rman 4
  - 13.standby db 5
- 1Z0-026 Oracle8i: Network Administration 59 41 69% 90
  - 1. Networking Overview 1
  - 2. Basic Net8 Architecture 5
  - 3. Basic Net8 Server-Side Configuration 8
  - 4. Basic Net8 Client-Side Configuration 8
  - 5. Centralized Naming Concepts 5
  - 6. Oracle Names Usage and Configuration 7
  - 7. Multithreaded Server Usage and Configuration 9
  - 8. Connection Manager Usage and Configuration 5
  - 9. Troubleshooting the Network Environment 7
  - 10 Security in the Network Environment 4
- 1Z0-101 Develop PL/SQL Program Units 57 39 68% 90
- 1z0-111 Developer/2000 Forms 4.5I 60 120
- 1z0-112 Developer/2000 Forms 4.5II 60 120
- 1z0-113 Developer/2000 Reports 2.5 67 90
- 1Z0-120 Developer/2000 Release 2 New Features 60 45 75% 90
- 1Z0-121 Developer/2000: Build Forms I Rel. 2 60 46 76% 90
- 1Z0-122 Developer/2000: Build Forms II Rel. 2 60 44 73% 90
- 1Z0-123 Developer/2000: Build Reports Rel. 2 60 43 72% 90
- 1Z0-130 Oracle Forms Developer Rel.1-6/6i New Features 60 34 57%
- 1Z0-131 Build Internet Applications I Rel. 6/6i 60 45 75%
- 1Z0-132 Build Internet Applications II Rel. 6/6i 60 46 77%
- 1Z0-210 Oracle Financial Applications:  
Financial Management, R11 69 90
- 1Z0-220 Oracle Financial Applications:



Applied Technology,R11 61 90  
1Z0-230 Oracle Financial Applications:Procurement,R11 77 90  
1Z0-240 Oracle Financial Applications:Order Fulfillment, R11 60 90  
1Z0-401 Oracle8: Database Operator 70 120  
1Z0-411 Oracle8i: Internet Database Operator  
1Z0-501 [Oracle candidates only] Sun Certified  
Programmer for the Java 2 Platform 59 120  
1Z0-502 Oracle JDeveloper (Rel.2):Develop Database  
Applications With Java 59 90  
1Z0-503 Object-Oriented Analysis and Design (Oracle) 58, 55** 90  
** There are 58 questions in Form A and 55 questions in Form B for this exam

### 三. 教材列表

课程编号 课程名称 教材编号

-----  
1Z0-001 Introduction to Oracle:SQL and PL/SQL Using Procedure Bulger m03989 ,  
m03990 , m03991 , m03992(学生手册 )  
m03993 , m03994 , m03995 , m03996(教师手册 )  
1Z0-020 Oracle8i:New Features for Administration M08353, M08354, M08355 ( 学生手册 )  
M08356, M08357, M08358 ( 教师手册 )  
1Z0-023 Enterprise DBA Part 1A: Architecture and Administration m09005 , m09006 ( 学  
生手册 )  
m09007 , m09008 ( 教师手册 )  
1Z0-024 Enterprise DBA Part 1B: Backup and Recovery Workshop m09095 , m09096 ( 学  
生手册 )  
m09097 , m09098 ( 教师手册 )  
1Z0-025 Enterprise DBA Part 2: Performance and Tuning m09213 , m09214 ( 学生手册 )  
m09215 , m09216 ( 教师手册 )  
1Z0-026 Enterprise DBA Part 3: Network Administration m09125 ( 学生手册 )  
m09126 ( 教师手册 )  
1Z0-101 Develop PL/SQL Program Units M06103 (PPT)  
1Z0-121 Developer/2000 Build Forms I Rel. 2 M06151, M06152  
1Z0-122 Developer/2000 Build Forms II Rel. 2 M08751 , M08752  
1Z0-123 Developer/2000 Build Reports Rel. 2 M08694, M08695  
1Z0-130 Oracle Forms Developer Rel.1-6/6i New Features M08491  
1Z0-131 Build Internet Applications I Rel. 6/6i M08604, M08605, M08606 (IG)  
1Z0-132 Build Internet Applications II Rel. 6/6i M08749, M08750 ( 学生手册 )  
M08751, M08752 ( 教师手册 )  
M08608 (PPT)

Oracle 官方教材一般会由以下几部分构成：

Instructor Guide  
student guide 或 Participant Guide  
Transparencies

## Electronic Presentation

其中 student guide 或 Participant Guide 就是听 oracle 讲座的时候会得到的课本，即所谓学生手册。

Instructor Guide 会在 student guide 或 Participant Guide 的基础上增加课时安排信息，即所谓老师手册，两者内容基本一致。

Transparencies 和 Electronic Presentation 是教师讲课的时候使用的幻灯片。

它们的编号一般按照 student guide 或 Participant Guide、Instructor Guide、Transparencies、Electronic Presentation 的顺序编号。

M013907 Administrating Oracle9i Application Server Student Guide

M03989 Introduction to Oracle: SQL and PL/SQL Participant Guide Vol. 1 Les1-6

M03990 Introduction to Oracle: SQL and PL/SQL Participant Guide Vol. 2 Les8-17

M03991 Introduction to Oracle: SQL and PL/SQL Participant Guide Vol. 3 Les18-26

M03992 Introduction to Oracle: SQL and PL/SQL Participant Guide Vol. 4 Practice

M04802 Oracle Express Relational Access Manager Instructor Guide

M04886 Oracle Express Database Design and Control Student Guide Vol. 1

M04887 Oracle Express Database Design and Control Student Guide Vol. 2

M05759 Oracle Express Database Design and Control Slides PPT

M06089 Introduction to Data Warehousing Instructor Guide

M06103 Developer PLSQL ProgramUnits PPT

M06149 Developer2000 Build Forms I_V1 SG

M06150 Developer2000 Build Forms I_V2 SG

M06151 Developer2000 Build Forms I_V1 IG

M06152 Developer2000 Build Forms I_V2 IG

M06180 Developer2000 Build Forms II_V1(Lesson1-10)

M06181 Developer2000 Build Forms II_V2(Lesson11-14)

M? Developer2000 Build Reports

M06268 Oracle8: Backup and Recovery Workshop Instructor Guide Vol. 1

M06269 Oracle8: Backup and Recovery Workshop Instructor Guide Vol. 2

M06291 Oracle8: Database Administration Instructor Guide Vol. 1

M06292 Oracle8: Database Administration Instructor Guide Vol. 2

M06293 Oracle8: Database Administration Instructor Guide Vol. 3

M07081 Oracle8: Parallel Server Implementation Student Guide

M07082 Oracle8: Parallel Server Implementation Instructor Guide

M07126 Develop Application With Oracle Express Objects Instructor Guide Vol. 1

M07127 Develop Application With Oracle Express Objects Instructor Guide Vol. 2

M07182 Develop Applications with Oracle Express Objects Transparencies PPT

M07349 Introduction to Oracle Financial Analyzer Instructor Guide Vol. 1

M07350 Introduction to Oracle Financial Analyzer Instructor Guide Vol. 2

M07391 Oracle Financial Analyzer for Administrators Instructor Guide Vol. 1

M07392 Oracle Financial Analyzer for Administrators Instructor Guide Vol. 2

M07861 Sql Statement Turning Workshop_O8 pdf

M08353 Oracle 8i: New Feature for Administrators Participant Guide Volume 1

M08354 Oracle 8i: New Feature for Administrators Participant Guide Volume 2

M08355 Oracle 8i: New Feature for Administrators Participant Guide Volume 3  
M08356 Oracle 8i: New Feature for Administrators Instructor Guide Vol. 1  
M08357 Oracle 8i: New Feature for Administrators Instructor Guide Vol. 2  
M08358 Oracle 8i: New Feature for Administrators Instructor Guide Vol. 3  
M08359 Transparancies  
M08360 Electronic Presentation  
M08491 Oracle Developer: Release 1 to Release 6 New Features Instructor Guide  
M08567 ORACLE ENTERPRISE MANAGER V2 Student Guide Vol. 1  
M08568 ORACLE ENTERPRISE MANAGER V2 Student Guide Vol. 2  
M08604 Oracle Developer - Build Forms I_V1 pdf  
M08605 Oracle Developer - Build Forms I_V2 pdf  
M08606 Oracle Developer - Build Forms I_V3 pdf  
M08608 Oracle Developer - Build Forms I_PPT pdf  
M08751 Oracle Developer: Build Forms II Instructor Guide Vol. 1  
M08752 Oracle Developer: Build Forms II Instructor Guide Vol. 2  
M08763 Data Warehousing Fundamentals Instructor Guide Vol. 1  
M08764 Data Warehousing Fundamentals Instructor Guide Vol. 2  
M08694_Oracle Developer - Build Reports_V1 pdf  
M08695_Oracle Developer - Build Reports_V2 pdf  
M09005 Enterprise DBA: Architecture and Administration Student Guide Vol. 1  
M09006 Enterprise DBA: Architecture and Administration Student Guide Vol. 2  
M09007 Enterprise DBA: Architecture and Administration Instructor Guide Vol. 1  
M09008 Enterprise DBA: Architecture and Administration Instructor Guide Vol. 2  
M09095 Enterprise DBA: Backup and Recovery Workshop Student Guide Vol. 1  
M09096 Enterprise DBA: Backup and Recovery Workshop Student Guide Vol. 2  
M09097 Enterprise DBA: Backup and Recovery Workshop Instructor Guide Vol. 1  
M09098 Enterprise DBA: Backup and Recovery Workshop Instructor Guide Vol. 2  
M09125 Enterprise DBA: Network Administration Student Guide  
M09126 Enterprise DBA: Network Administration Instructor Guide  
M09213 Enterprise DBA: Performance and Tuning Student Guide Vol. 1  
M09214 Enterprise DBA: Performance and Tuning Student Guide Vol. 2  
M09215 Enterprise DBA: Performance and Tuning Instructor Guide Vol. 1  
M09216 Enterprise DBA: Performance and Tuning Instructor Guide Vol. 2

#### 四. 考试资源列表

这里列出的是网上可以找到的资料

课程编号 作者 资料编号/名称 文件格式 备注

-----  
1Z0-001 Oracle m03989 , m03990 , m03991 , m03992 PDF  
未知 Oracle Class Handouts -- SQL&PL/SQL PDF/PPT  
www.coolcram.com CoolCramBible for 1Z0-001 PDF  
www.brainbuzz.com Cramsession for 1Z0-001 PDF  
未知 OCP for Nokia Tutorial Notes 1Z0-001 DOC

未知 CH7 -- Exam Terms and Concepts PDF BOOK  
Troy Techologies Study Guide for 1Z0-001 PDF  
Sybex OCP DBA Study Guides: SQL&PLSQL PDF BOOK  
Sybex SQL&PLSQL Flashcards EXE TEST  
STS 1Z0-001e_v2.22a EXE 296 题  
STS 1Z0-001e_v2.29d EXE 307 题  
CHEET SHEETS 1Z0-001e_20010206 DOC 175 题  
未知 Exam1_Introduction to SQL&PLSQL PDF 120 题  
measure_up setup001 EXE

---

1Z0-013 [www.coolcram.com](http://www.coolcram.com) CoolCramBible for 1Z0-013 PDF  
[www.brainbuzz.com](http://www.brainbuzz.com) Cramsession for 1Z0-013 PDF  
未知 OCP for Nokia Tutorial Notes 1Z0-013 DOC  
未知 CH8 -- Exam Terms and Concepts PDF BOOK  
Oracle M06295 PPT  
Troy Techologies Study Guide for 1Z0-013 PDF  
STS 1Z0-013e_v2.22a EXE 300 题  
CHEET SHEETS 1Z0-013e_20010318 DOC 112 题  
未知 Exam2_Database Administration PDF 120 题  
未知 DOC 62 题

---

1Z0-014 [www.brainbuzz.com](http://www.brainbuzz.com) Cramsession for 1Z0-014 PDF  
未知 OCP for Nokia Tutorial Notes 1Z0-014 DOC  
Oracle PPT  
Troy Techologies Study Guide for 1Z0-014 PDF  
STS 1Z0-014e_v2.20 EXE 300 题  
STS 1Z0-014e_v2.22a EXE 300 题

---

1Z0-015 [www.brainbuzz.com](http://www.brainbuzz.com) Cramsession for 1Z0-015 PDF  
未知 OCP for Nokia Tutorial Notes 1Z0-015 DOC  
未知 CH9 -- Exam Terms and Concepts PDF BOOK  
未知 Study Notes DOC 笔记  
Troy Techologies Study Guide for 1Z0-015 PDF  
STS 1Z0-015e_v2.22a EXE 300 题  
未知 Exam3_Backup and Recovery Workshop PDF 120 题

---

1Z0-016 [www.coolcram.com](http://www.coolcram.com) CoolCramBible for 1Z0-016 PDF  
[www.brainbuzz.com](http://www.brainbuzz.com) Cramsession for 1Z0-016 PDF  
未知 OCP for Nokia Tutorial Notes 1Z0-016 DOC  
未知 CH11 -- Exam Terms and Concepts PDF BOOK  
Troy Techologies Study Guide for 1Z0-016 PDF  
STS 1Z0-016e_v2.15g EXE 293 题  
STS 1Z0-016e_v2.23N EXE 293 题  
CHEET SHEETS 1Z0-016e_20010317 DOC 118 题

---

---

1Z0-020 Oracle M08356, M08357, M08358 PDF

Oracle M08360 PPT

Peter N. Crompton Study Guide PDF 笔记

未知 CH12 -- Exam Terms and Concepts PDF BOOK

STS 1Z0-020e_v2.28t EXE 286 题

STS 1Z0-020e_v2.29d EXE 253 题

未知 Oracle8i Upgrade New Features PDF 120 题

---

1Z0-023 Contemporary Tech Quick Study Guide for 1Z0-023 PDF

[www.brainbuzz.com](http://www.brainbuzz.com) Cramsession for 1Z0-023 PDF

Sybex OCP DBA Study Guide: DBA_Admin&BackRec PDF BOOK

STS 1Z0-023e_v2.27g EXE 300 题

STS 1Z0-023e_v2.29d EXE 300 题

STS 1Z0-023w_v2.32l EXE 291 题

measure_up setup023 EXE

---

1Z0-024 Contemporary Tech Quick Study Guide for 1Z0-024 PDF

Sybex OCP DBA Study Guide: Perf&Netadm PDF BOOK

STS 1Z0-024e_v2.28t EXE 300 题

measure_up setup024 EXE

---

1Z0-025 Contemporary Tech Quick Study Guide for 1Z0-025 PDF

未知 Oracle 8i B&R Quick Study Guide DOC

Sybex OCP DBA Study Guide: DBA_Admin&BackRec PDF BOOK

STS 1Z0-025e_v2.28t` EXE 315 题

---

1Z0-026 Contemporary Tech Quick Study Guide for 1Z0-026 PDF

Sybex OCP DBA Study Guide: Perf&Netadm PDF BOOK

STS 1Z0-026e_v2.29b EXE 268 题

CHEET SHEETS 1Z0-026e_20010206 DOC 175 题

---

1Z0-101

CHEET SHEETS 1Z0-101e_20010209 DOC 175 题

---

未知 OCP DBA Guide HTML OCP8

1Z0-401 未知 OCP DBO Certification Exam Guide HTML OCP8

未知 OCP DBA Certification Exam Guide HTML OCP8

---

1Z0-411 Sybex Oracle8i DBO Study Guide PDF BOOK

Sybex Oracle8i DBO Flashcards EXE TEST

---

## 五. 考试简介

### 1. 选择授权考试中心

目前 Oracle 委托 Prometric 公司进行 OCP 认证考试, 请大家到下面这个网址查看国内各地的授权考试中心, 注意询问是否有 OCP 考试, 并且在考试中心注意是否有 prometric 的授权证书

[http://www.prometric.com.cn/aptcquery.asp?page_id=84](http://www.prometric.com.cn/aptcquery.asp?page_id=84)

## 2. 认证考试那天

请您在预约时间前至少 15 分钟到达测试中心。如果您迟到超过 15 分钟, 测试中心人员将不会给您安排座位, 否则会干扰其他人的考试。

签署测试日志, 准备出示至少两种身份证件。一种必须是政府颁发的带照片的身份证, 两种证件上必须有您的签名。

测试中心管理员将给您简单介绍一下情况, 领您到考试的计算机终端。在考试开始前, 测试中心人员将给您提供关于考试系统和设备使用的计算机化指导, 使您熟悉考试程序。情况介绍包括检查试题、输入答题的指导, 怎样访问您回答某些测试试题所需要的演示。请注意, 测试中心人员并不是试题专家, 对于测试内容, 他们不能给您任何咨询。还请注意, 到了规定的时间, 计算机的程序设计将终止考试, 剩下没有回答的问题将被视作不正确。

请注意: 不许将纸张、书籍、包、计算机、个人记事本 (personal organizers) 计算器等带入考场。考试时用的草稿纸将由测试中心人员提供。考试结束时, 测试中心人员收回并销毁所有草稿纸。

## 3. 获得您的考试结果

完成测试时可从测试中心得到考试成绩单。如果您参加 beta 考试, 则在您的测试评判和打分后把成绩单邮寄给您。

Prometric 把所有考试结果立即发送到 Oracle 认证项目管理部门。

请在您的档案里保留您的测试结果复份。

## 4. 补考

报考人考试不及格必须至少等 30 天后再进行补考, 这一政策盖无例外。Oracle 鼓励您利用与成绩单一起提供的评测反馈意见, 以便复习需要进一步准备或学习的部分。补考前, 报考人可以利用在线测试内容清单 (Test Content Checklists), 发现在技能方面需要多加注意之处。

## 5. 更多需要考虑的事项

全部考试都是基于计算机进行的。认证考试可以包含多项选择、填空和交互式图解问题。仔细阅读全部试题, 包括所有回答选项, 因为有些题不只有一个正确答案。您需要选出最佳答案或全部正确的答案。有些试题还包括展示, 它们将在屏幕上呈现给您, 您在回答问题时可以查阅这些展示。

安排好自己的进度。了解试题的总数, 分配好时间, 以便进行您的测试。先回答那些容易的试题, 然后再花时间回答看来难一些的试题。所有试题都是平均打分的, 所以, 您做的看来容易的试题的得分, 与做的难一些的试题的得分都是一样的。测试项目允许您在那些想回过头再做的试题上做出标记。在考试最后给您呈现复查屏幕, 上面显示出您已答过的题目、遗漏的题目、做出标记的题目或回答不完整的题目。

猜测没有把握试题。您的测试得分是基于您正确答案的数量。题答错了是不会罚扣的，所以猜一猜对您是有好处的。仔细看好题目和所有回答选项。您能答多少题目就先答多少，剩下的题目可以猜测。

复查您的答案。如果完成考试后您还有剩余时间，检查您的答案，特别是那些您觉得把握不大的试题。

每一门考试实行多种形式。您预先不知道会遇到哪种形式的考试。如果您在一种形式的考试失败了，选择了重考，就会给您不同的形式的考试，这种形式同样测试您的技能和知识。

## 六. 相关站点资源

Oracle 考试认证 官方介绍：

<http://www.oracle.com/global/cn/education/certification/index.html?ocptracks.html>

Oracle 官方报考指南下载：

<http://www.oracle.com/global/cn/education/certification/index.html?ocpguides.html>

Oracle 模拟器 simulator 介绍：

<http://www.oracle.com/global/cn/education/certification/simulator/index.html?content.html>

本文内容主要参考网上信息，因时间较长，具体来源等目前无法确定，仅在此感谢那些提供过信息资料的网友们。

因内容较多,部分资料未经核实，同时也有所欠缺，希望广大 fans 们给予补充，谢谢。



## LZ0-024 考点总结

2001/12/01 Write by easyfree

### 1. Course Intraduction 0

### 2. Tuning overview 1

(1) 调整的先后次序

1. Tune the design. -- Application designers
2. Tune the application. -- Application developers
3. Tune memory. --|
4. Tune I/O. |-- DBA
5. Tune contention. |
6. Tune the operating system. --|

(2) 调整的基点和主要的可测量的目标

Database hit percentages 命中率是 base line

SQL statements access the smallest possible number of Oracle blocks 访问尽可能少的数据块 response time, database availability, memory utilization(页面交换)等

### 3. Oracle Alert and Trace Files 2

(1) USER_DUMP_DEST 和 BACKGROUD_DUMP_DEST

USER_DUMP_DEST: SQL_TRACE, DEAD LOCK, 用户 session 中 sql 语句的执行情况

BACKGROUD_DUMP_DEST: Alert.log, 系统后台进程的错误信息

(2) Alert.log 文件的特性和内容

- 启动时不存在则自动创建,存在 BACKGROUD_DUMP_DEST 路径下
- 文件尺寸一直增长,需要人工清除
- 包含内容有 :

Internal errors (ORA-600), and block corruption errors (ORA-1578)

影响数据库结构,参数的操作,以及命令:CREATE DATABASE, STARTUP, SHUTDOWN, ARCHIVE LOG, RECOVER

实例启动时的非缺省参数 -- 启动时写入

控制文件和在线表空间备份

未完成的检查点

(3) SQL_TRACE 设置的两个级别

Instance 参数中设定

Session        ALTER        SESSION        SET        SQL_TRACE=TRUE        /  
DBMS_SYSTEM.SET_SQL_TRACE_IN_SESSION(sid,serial#,TRUE/FALSE)

(4) 参数 MAX_DUMP_FILE_SIZE 限制 User Trace 文件的 O/S Blocks

### 4. Utilities and Dynamic Performance Views 3

(1) Anylyze 执行之后查询信息的字典:

DBA_TABLES and DBA_TAB_COLUMNS table storage within extents and blocks

DBA_HISTOGRAMS and INDEX_HISTOGRAM data about nonindexed and indexed column data distribution.

DBA_INDEXES and INDEX_STATS data about index within extents and blocks and indexation usefulness.



## (2) 关于 Latch

Latch 的类型: Willing-To-Wait 和 Immediate

Gets/Misses/Sleeps 的意义

Reports.txt 中记录的 Latch 的 hit ratio 应 $\geq 99\%$ ,即 redo latch 竞争应 $\leq 1\%$

DBA 可调整的 Latch 争用区域: Redo allocation latch/Redo copy latch/LRU latch

## (3) 动态视图

特点: 是 X\$表和 V\$_的同义词,属于 sys,在 V\$FIXED_TABLE 中,NOMOUNT 和 MOUNT 时可以查询,TIMED_STATISTICS=TRUE 记录 WAIT_TIME

主要的视图

%EVENT 事件类视图 包含 wait for event 的统计 -- 注意%WAIT, wait for events or resource 的信息

-- V\$SYSTEM_EVENT , V\$SESSION_EVENT

%STAT 统计类视图 V\$SYSSTAT(Instance 启动以来的统计)

%CACHE 内存方面的统计视图,主要有 V\$LIBRARYCACHE 和 V\$ROWCACHE(数据字典的...),V\$DB_OBJECT_CACHE

PX% 并行处理的信息

V\$SESSION 记录所有用户连接的 type(BACKGROUND/USER)

V\$LOCK 和 V\$LATCH 的区别: 前者是现有锁信息, 后者是锁争用统计信息

V\$SYSTEM_EVENT(所有 session 从 Instance 启动以来)/V\$SESSION_EVENT(每个 session)/V\$SESSION_WAIT(当前活动 session)

V\$SESSION_WAIT.WAIT_TIME -- 要获取 WAIT_TIME 的值, 必须要将 TIMED_STATISTICS=TRUE(动态可改)

> 0 The session's last wait time

= 0 The session is currently waiting

= -1 The value was less than 1/100 of a second

= -2 The system cannot provide timing information -- TIMED_STATISTICS=FALSE

## (4) 关于 UTLBSTAT and UTLESTAT 工具

特点: 需要 SYSDBA,创建了一些统计表和视图并在结束时删除,DEFERENCE 记录开始和结束时统计的差异,report.txt

TIMED_STATISTICS=TRUE,统计期间发生中断需要重新运行

Report.txt 的内容

Library Cache 涉及 SQL,PL/SQL 语句执行

System 涉及 buffer cache 和逻辑读写

Wait events 涉及等待的 CPU 时间

Latch 涉及内存中锁的争用,redo allocation/redo copy/LRU

Rollback contention 涉及 undo header, 等待 rollback header 中的事务 slot

Buffer Busy Wait 涉及 data block,segment header,undo header 争用

Dictionary cache 涉及数据字典的 get/miss

I/O 涉及数据文件的读写

Period of measurement UTLBSTAT,UTLESTAT 开始和结束的时间

## 5. Tuning the Shared Pool 7

### (1) 关于 Shared Pool

特性: SHARED_POOL_SIZE 决定大小, library cache + data dictionary cache + UGA + large pool

调整 Shared Pool 的原因: shared pool 的 miss 比 database buffer cache 的 miss 影响大,library cache 首要

(2) 关于 Large objects

特性: use LRU

tuning: generic code/bind variable/防止空间不足 age out 而 reload/防止 object 更改而 re-parse

大的匿名块->小的过程/pin/reserve space for large objects

keep: 方法: DBMS_SHARED_POOL.KEEP / .UNKEEP

需要 keep 的 object: 常用的包/常用的 trigger/sequence

使用: Instance 启动时 keep 防止碎片/ALTER SYSTEM FLUSH SHARED_POOL 来 flush shared pool(no keep)

视图: V\$LIBRARYCACHE(GETHITRATIO>90%,否则优化语句应用; reloads/pins 应 <=1%)

V\$SGASTAT/V\$SQLAREA/V\$SQLTEXT/V\$DB_OBJECT_CACHE(过程等占用内存)

参数: OPEN_CURSORS/SESSION_CACHED_CURSORS

CURSOR_SPACE_FOR_TIME -- 除非 RELOADS in V\$LIBRARYCACHE 一直为 0, 否则保持缺省值:false

预留空间: V\$SHARED_POOL_RESERVED

当 REQUEST_FAILURES 大于 0 并且不断增长 (ORA-4031), 可以相应增大 SHARED_POOL_RESERVED_SIZE & SHARED_POOL_SIZE 的值

REQUEST_MISS = 0 并且不再增长 或 FREE_MEMORY>=50%*SHARED_POOL_RESERVED_SIZE,考虑减少 reserved size

SHARED_POOL_RESERVED_SIZE 初始为 SHARED_POOL_SIZE 的 10%

(3) 关于数据字典 cache

特性: 启动时任何 sql 语句都将导致 cache miss, GETMISSES 几乎不可能为 0

调整: 调整 SHARED_POOL_SIZE 的大小而间接地调整 dictionary cache

report.txt 中: GET_MISS/GET_REQS < 15%

字典: V\$ROWCACHE -- SUM(GETMISSES)/SUM(GETS) < 15%, 否则应增大 SHARED_POOL_SIZE

(4) 关于 UGA

包括: User Session Data(sort area & private SQL area) 和 Cursor State, MTS 中创建

存储: 专用服务器-->PGA, MTS-->shared pool, 使用 MTS 的总内存<=使用专用服务器内存

调整: 查阅视图 V\$MYSTAT,( V\$STATNAME, V\$SESSTAT 查询 space usage for MTS user),计算: SUM(VALUE)

(5) 关于 Large Pool

特性: LARGE_POOL_SIZE 需明确设置,若未设置使用 shared pool 分配

The Oracle library cache and buffer cache will never allocate memory from the large pool

用途: I/O 服务进程 /oracle 备份回复 /MTS/ 并行操作 (PARALLEL_AUTOMATIC_TUNING=TRUE)

字典: v\$sgastat

## 6. Tuning the Buffer Cache 7

(1) 关于命中率

公式:  $\text{Hit Ratio} = 1 - (\text{physical reads} / (\text{db block gets} + \text{consistent gets}))$  -- 分母是 request 的总数,包括内存和磁盘读取

-- 因为这些统计数据是实例启动后收集的,所以不要启动后立刻进行计算,因为这时 buffer cache 可能是空的

视图: V\$SYSSTAT( name,value)

指标: Hit Ratio 应 $\geq$ 90%, 否则需要增加 DB_BLOCK_BUFFERS

(2) 关于 Multiple Buffer Pools

-- There are at least 50 blocks per latch

-- 总数不能超过 DB_BLOCK_BUFFERS 和 DB_BLOCK_LRU_LATCHES ,否则 mount 时候出错

-- 三种

KEEP: 保存最有可能重用的 object

RECYCLE: 保存很少被重用的 object

DEFAULT: 始终存在,大小等于单个 buffer cache, 尺寸定义=DB_BLOCK_BUFFERS-其它 buffer

-- V\$BUFFER_POOL_STATISTICS: consistent gets statistics for multiple buffer caches

-- 如何计算 KEEP buffer pool 的数据

ANALYZE ... ESTIMATE STATISTICS

获取 objects 的大小:将 DBA_TABLES, DBA_INDEXES, and DBA_CLUSTERS 中的 blocks 相加

(3) 关于 LRU Latches

特性: 每个 latch 最少控制 50 个 buffer, 最小=1,缺省=1/2*CPU, 对每个 DBWn 进程有一个 Latch,命中率应 $\geq$ 99%

视图: V\$LATCH and V\$LATCHNAME

参数: DB_BLOCK_LRU_LATCHES(单 cpu 系统中,不要超过 CPU,在多 buffer pool 中)

(4) 关于 Free List

特性: freelist 决定哪个 block 可以用于 insert

视图: V\$SESSION_WAIT class = 'segment header'

DBA_SEGMENTS segment 当前存在的 freelist 的数量

V\$WAITSTAT SELECT class, count, time FROM v\$waitstat WHERE class = 'segment header';

V\$SYSTEM_EVENT event='buffer busy waits'

解决竞争三个步骤:

1. 查询 V\$SESSION_WAIT,获取 FILE, BLOCK, and ID

2. 查询 DBA_SEGMENTS 和 V\$SESSION_WAIT,获取发生竞争的 segment 信息

3. 重建 object,增加 freelist

(5) V\$CACHE 与 V\$BUFFER_POOL 的区别

-- V\$CACHE: 监控每个 object 占用的 buffer pool block 数量/由 catparr.sql 创建/用于 OPS

to determine the number of blocks in the RECYCLE buffer pool

-- V\$BUFFER_POOL: Describes multiple buffer pools

(6) 关于 table cache 的说法

目标: 对未 cache 的 table 进行 full scan,block 都在 MRU 端,可以 cache table 放在 LRU

端

- 方式: : 1. Create a table using the CACHE clause -- create table ...cache/nocache  
2. Alter a table using the CACHE clause -- alter table ... cache/nocache  
3. Code the CACHE hint clause into a query -- 查询中使用 cache 提示

注意: 过多的 cache table 可能使 buffer cache 过分拥挤(overcrowd)

## 7. Tuning the Redo Log Buffer 3

(1) 怎样设置 redo log file 可以加快 ARCHIVELOG MODE 下数据库的恢复速度?

-- Create small redo log files 增加归档的次数

(2) 关于调整 redo log buffer

V\$SESSION_WAIT: 事件: 'Log Buffer Space'说明空间等待(写入 log buffer 比 LGWR 写出快)

解决: 增加参数 LOG_BUFFER,将 log buffer 移到更快的 disk

V\$SYSSTAT 事件: 'redo buffer allocation retries'说明新的 entries 写入覆盖已写入 disk 的 entries 的空间等待

'redo log space requests'说明活动 log file 写满,等待 Oracle server 磁盘空间分配

'redo entries'上述'redo buffer allocation retries'/ 'redo entries'应<=1%

解决: 增加 log buffer/improve checkpoint 或归档进程

V\$SYSTEM_EVENT 事件: 'log file switch completion'说明 LOG SWITCH 的 wait

事件: alert.log 文件中有"CHECKPOINT NOTCOMPLETE.",说明 LGWR 等待 DBWn 完成一个 CHECKPOINT

'Log File Switch (Checkpoint Incomplete)'

解 决 : 调 整 参 数 LOG_CHECKPOINT_INTERVAL and LOG_CHECKPOINT_TIMEOUT

Check the size and number of redo log groups

(3) 关于 NOLOGGING

可以使用 NOLOGGING 模式的 SQL 语句: CTAS/CREATE INDEX/ALTER INDEX REBUILD

DRICT LOAD Sqlloader direct path: Set the NOLOGGING attribute.

(4) REDO LOG BUFFER 的特性

-- 参数 LOG_BUFFER 决定尺寸,大小必须是 OS block size 的倍数,一般为最大 block size 的 4 倍

-- Frequent COMMIT statements lead to a smaller buffer size requirement

-- Larger redo log buffer sizes reduce log file I/O

-- The tuning goal for the redo log buffer is to ensure that there is sufficient space for the server processes.

-- redo log buffer 分配内存过多将减少分配给其他 areas 的内存

(5) V\$SESSION_WAIT 中字段 SECONDS_IN_WAIT 的信息说明了什么?

-- 指明等待空间的时间(由于 log switch 未发生) -- buffer 填充比 LGWR 写出快

-- 也指明了 redo log file 所在的 disk I/O 竞争

## 8. Database Configuration and I/O Issues 7

(1) Alert.log file 出现信息:"Checkpoint not complete; unable to allocate file."意味着什么?

-- LGWR waited for the checkpoint to finish.

- (2) 影响 Full table scan I/O 的参数是什么?
  - DB_FILE_MULTIBLOCK_READ_COUNT
- (3) Local Managed tablespace 的特性和优点
  - 没有字典, 很少出现一致性问题, 可以有上千个 extent 而不涉及性能问题不必重组, Extent 分配信息存在表空间本地(bitmap)
- (4) 视图 V\$FILESTAT 的用途
  - 监控每个磁盘文件的 disk I/O 活动情况和物理读写情况
- (5) 关于表空间的一些特点
  - 表和索引分表空间存放, 用户不指定表空间(包括临时表空间)将使用 system 表空间
  - RBS 仅用于存放 rollback segment
  - system 表空间仅包含属于 sys 用户的 objects, 其它用户应不允许在 system 表空间中创建 object
- (6) 条带化文件的参数和手工命令
  - DB_FILE_MULTIBLOCK_READ_COUNT * DB_BLOCK_SIZE
  - CREATE TABLE / ALTER TABLE ALLOCATE
- (7) 检测 Full table scan 的视图
  - V\$SYSSTAT where name like '%table scans%'
  - V\$SESSION_LONGOPS
- (8) 关于 redo log 的配置
  - 归档日志不能放在 raw 设备上
  - 定义多个归档进程并写在不同目标磁盘上
  - LGWR 与 ARCn 分磁盘
- (9) 决定参数 DB_BLOCK_MAX_DIRTY_TARGET 参数是否合适的事件和视图是什么?
  - V\$SYSTEM_EVENT 的"WRITE COMPLETE WAITS"事件
- (10) I/O slaves 是模拟异步 I/O, 提供 nonblocking I/O 应用

## 9. Using Oracle Blocks Efficiently 4

- (1) Row Chaining 和 Row Migration 的特点
  - 行链接: Insert or Update 引起, 一个 Row 存在不同的 block 中, 可以通过设置大的 DB_BLOCK_SIZE 来减少行链接的发生, 但无法避免(LOB)
  - 行迁移: Update 引起, 由于 PCTFREE 过小, ROWID 未变, 可以通过增大 PCTFREE 的值减少行迁移的发生
  - 影响: 产生额外的 I/O
- (2) Block size 大小的优缺点
  - Large: 有益于 very large rows and sequential reads, 管理代价小, 适合 DSS
  - Small: 有益于 small rows(减少 block 争用), random reads(内存中 reuse 机会少), 管理代价高, 适合 OLTP
- (3) 排除行迁移的步骤 eliminate row migration
  1. ANALYZE TABLE ... LIST CHAINED ROWS;
  2. Copy the rows to another table.
  3. Delete the rows from the original table.
  4. Insert the rows from step 2 back into the original table.
- (4) 如何发现和检查行迁移/行链接 -- ANALYZE table COMPUTE STATISTICS;
- (5) 分析索引的命令和视图?

```
-- ANALYZE INDEX index_name VALIDATE STRUCTURE;
-- INDEX_STATS
(6) 大的 Extents 的优缺点
-- 优点: 避免动态扩展,减少 multiblock reads
-- 缺点: 不易获得连续的空间, 初始时浪费空间
-- 其它: 全表扫描和索引查询的性能不受 extent 的个数影响
(7) 什么时候需要考虑 PCTFREE 和 PCTUSED 的更改
-- PCTFREE: 需要经常更改的 row,考虑增大,对 DSS,可以考虑设置=0
-- PCTUSED: 可能需要删除的 row,考虑增大
(8) HWM 的特点
-- 初始时设置,存储在 segment 的 header block 中
-- ALTER TABLE <table_name> DEALLOCATE UNUSED... 回收 HWM 之上的空间,
TRUNCATE 回收全部
-- 全表扫描从 first block --> HWM
(9) REBUILD 索引的一些特点
-- REBUILD 比 DROP/RECREATE 速度快
-- UNRECOVERABLE 与 LOGGING 不兼容
-- 若 delete entries 占当前 entries 20%以上,则可重建索引
```

#### 10. Optimizing Sort Operations 4

```
(1) 需要进行排序的操作有哪些?
-- Index creation -- 在 building B-tree 之前 sort indexed values
-- Parallel insert operation involving index maintenance -- 并行插入操作涉及索引维护
-- ORDER BY or GROUP BY clauses
-- DISTINCT values selection -- 为排除重复行
-- UNION, INTERSECT, or MINUS operators -- 排除重复行
-- Sort-merge joins -- 执行一个涉及到两个表连接,而连接字段没有索引
(2) 参数 SORT_AREA_RETAINED_SIZE 的作用是什么?
-- 排序操作完成后保留的 sort area 大小
-- 大于该参数部分的内存释放返回给 UGA
(3) 涉及 sort 信息的几个相关视图
-- V$SORT_SEGMENT 显示 active individual sort segments 上的用户数量,扩展
extents 的数量
-- V$SORT_USAGE 显示临时段正在使用的情况,和 V$SESSION 结合查询经常使用
较大临时段的 USER
-- V$SYSSTAT 显示所有 sorts (memory),sorts (disk),sorts (rows)的数量(since
instance startup),比例(<5%)
(4) 计算并行查询所需要的内存数量的公式? -- SORT_AREA_SIZE * 2 * degree of
parallelism
(5) 如何设定和更改 SORT_AREA_SIZE 的值?
-- INIT.ORA 中定义
-- ALTER SESSION or ALTER SYSTEM DEFERRED
(6) 关于参数 SORT_MULTIBLOCK_READ_COUNT
-- 缺省=2
-- 该参数意味着: 每次 sort 从 a temporary segment 中读取的 block 的 number
```

-- 提高此值可以改善 sort

## 11. Tuning Rollback Segments 4

(1) ORA-01555: snapshot too old 错误的原因

-- 正在查询的 block 已经被重用了,此 block 的 SCN 比查询开始时的 SCN 新

-- 在回滚段 header 中的 transaction slot 已经被重用了

-- 回滚段中的 undo data 在一个提交后被覆盖了(an Interested Transaction List (ITL) entry in a data block has been reused)

(2) 回滚段的特性

--STORAGE:INITIAL:NEXT(128K 分界),PCTINCREASE=0,MINEXTENTS=20,MAXEXTENTS=UNLIMITED,RBS 表空间动态扩展

-- 循环方式,不能跨越,自动扩展,设定 OPTIMAL 回收

-- DELETE 需要保存整个记录,INSERT 只需 ROWID,INDEX 列需要 old data,old index,new index

(3) 控制 redo 的命令? -- SET TRANSACTION USE ROLLBACK SEGMENT ...

(4) 回滚段调整的目标 Tuning Goals

1. 事务无需等待回滚段的访问,需要等待足够的回滚段

2. 在正常运行中,回滚段不要扩展

3. 无论是大的还是特殊的事务,都不应因回滚段空间的不足而失败

4. 数据读取总是能从 read-consistent images 中获取需要的内容

(5) OLTP 事务的回滚段管理

一般事务较小,所以分配小的回滚段

一般规则: 决定于并发工作量(一个 concurrent job 一个回滚段),一个 RBS 4 个事务,最多 200 个 user

(6) 查询哪个视图可以获得事务写入回滚段的字节数量? -- V\$ROLLSTAT

(7) 标识回滚段 header 竞争的三个视图:

-- V\$ROLLSTAT.WAITS/GETS>1%(5%)

-- V\$WAITSAT.COUNT 中 CLASS='undo header'的值<>0

-- V\$SYSTEM_EVENT.EVENT='Undo Segment Tx Slot'(也可能是 buffer cache 过小)

(8) 为什么在设置回滚段存储参数时 NEXT=INITIAL?

-- 因为回滚段采用循环使用的方式,大的事务可能用到小的回滚段,一段时间后将同样大小 ???

## 12. Monitoring and Detecting Lock Contention 5

(1) 死锁的特点和解决

-- 两个以上用户互相等待对方锁定的数据资源

-- Oracle 自动检测 deadlock 并通过回滚检测出的 deadlock 的语句来解决死锁,但注意,仅回滚了语句而没有回滚该事务,返回 ORA-00060 错误

下一步应当手工回滚剩余的事务

-- DeadLock 更多地发生在事务明确地覆盖了 Oracle Server 的缺省锁,对分布式 deadlock,处理同非分布式相同

-- DeadLock 检测出后,会记录在 USER_DUMP_DEST 下的 trace file 中,包括 lock 的 row id

-- ALTER TABLE ... ENABLE/DISABLE TABLE LOCK; -- 在并行服务器中,允许/禁止 DML,DDL,LOCK on a table



(2) Lock 相关的两个表的用途

V\$LOCK -- 显示回滚段和 slot 的数量以及修改的表的 ID for 当前 hold 的 lock

V\$LOCKED_OBJECT -- 显示当前 hold 的 lock 的 object ID

(3) 释放锁涉及的视图和命令

-- 查询 V\$SESSION 获得: SID,SERIAL# / 引起竞争的 row

-- ALTER SYSTEM KILL SESSION 'SID,SERIAL#';

-- COMMIT & ROLLBACK

(4) 导致锁竞争的几种情况

-- users who do not regularly commit changes

-- developers who code unnecessarily long transactions

-- developers use unnecessarily high locking levels.

### 13. SQL Issues and Tuning Considerations for Different Applications 5

(1) Bitmap 索引和 B-tree 索引的比较

B-tree Bitmap

-----  
Suitable for high-cardinality columns Suitable for low-cardinality columns

Updates on keys relatively inexpensive Updates to key columns very expensive

Inefficient for queries using OR predicates Efficient for queries using OR predicates

Row-level locking Bitmap segment-level locking

More storage Less storage

Useful for OLTP Useful for DSS

适合查询表的一小部分(<5%)

(2) 优化器的稳定性特点

-- 重写 sql

-- 创建包含 hints 的 stored outline

-- 应用中强制访问路径

-- SQL 语句尽量保持匹配

(3) 为什么应用 SQL 的 tuning 很重要?

-- 应用中的语句影响最大, 如果它包含低效率的 SQL 语句,那么底层的 tuning 没有什么效果

(4) OLTP 和 DSS 的比较,理解什么样的应用属于什么样的系统

OLTP DSS

-----  
Performs index searches More full table scans

Uses B-tree indexes Uses bitmap indexes

Uses reverse key indexes Uses IOT tables

Needs more, small rollback segments Fewer, large rollback segment

Should not use parallel query Employs parallel query for large operations

PCTFREE according to expected update activity PCTFREE can be set to 0

Shared code and bind variables Literal variables and hints

Uses ANALYZE indexes Histograms generation

(5) TKPROF 的用途?

-- 格式化 SQLTRACE session 的 output

(6) 使用 cluster 的好处?



- 减少存储
- 减少 disk I/O 和访问时间
- (7) What should be the tuning priority for a DSS application?
- Use the optimal access path in the execution plan.
- (8) CBO 的特点
- 统计驱动
- 优化器检查每个 SQL,识别所有可能的访问路径并计算各自的成本,选择最少的逻辑

读数量

- (9) Reverse Key Index 在什么情况下适用?
- 一直顺序增长的(ever-increasing)key,如:订单号,对指定范围的区域扫描不合适
- (10) IOT 有哪些特点
- 主键值没有重复存储(索引和表在一起),节省空间
- 对等值或区间查询可提供快速的 key-based 访问
- 存储整个 row
- 使用 row overflow 区存放 nonkey 值
- (11) 对于混合系统的推荐设置
- 配置不同的回滚段
- 设置两套参数,放在不同的 pfile 中
- MTS 可在高峰时间使用(peak-time),not for DSS

#### 14. Managing a Mixed Workload 2

- (1) Resource Manager 包含的内容:
  - Resource Plan
  - Resource consumer groups
  - Resource plan directives
- (2) 数据库建立时的缺省资源计划为: SYSTEM_PLAN
- (3) 初始化参数 RESOURCE_MANAGER_PLAN 设置的资源计划名称必须在数据库中  
存在,否则报错,需要 shutdown,可用 ALTER SYSTEM 修改并立即生效
- (4) 过程包 DBMS_RESOURCE_MANAGER 的几个过程
  - CREATE_PENDING_AREA() 一个 instance 同一时间点只能建立一个 pending area
  - CREATE_CONSUMER_GROUP()
  - CREATE_PLAN()
  - VALIDATE_PENDING_AREA()
  - SUBMIT_PENDING_AREA()
  - GRANT_SWITCH_CONSUMER_GROUP() SET_INITIAL_CONSUMER_GROUP()
  - SWITCH_CURRENT_CONSUMER_GROUP()
  - SWITCH_CONSUMER_GROUP_FOR_SESS()
  - SWITCH_CONSUMER_GROUP_FOR_USER()
- (5) 当前数据库 RM 设置的几个视图:
  - V\$SESSION.RESOURCE_CONSUMER_GROUP 显示 session 的当前 group
  - V\$RSRC_PLAN 显示获得的资源计划
  - V\$RSRC_CONSUMER_GROUP 包含资源消耗组的统计信息

#### 15. Tuning with Oracle Expert 1

- (1) In addition to the rules, what else is the basis for an Oracle Expert tuning session?
- scope defined at the beginning of the tuning session

(2) Oracle Expert 为生成建议需要收集的数据是什么?

-- data in the Schema class

(3) 如何执行 Oracle Expert 生成的调整建议?

-- 使用 Oracle Expert 生成的.ora 参数文件替换 init.ora 文件来执行建议

-- 为执行推荐的 SQL 语句和结构, 检查 Oracle Expert 生成的.txt 文件,将包含 string TBS 的地方替换成相近的信息

(4) 几个 class 的特性

-- Database class 从 DBA_%,V\$视图中收集数据

-- Instance class 高峰时期采集样本数据,推荐收集 10 个以上的样本,以获取更好的调整建议

-- Collection Schema class 用于优化访问 schema 的路径

--Storage optimization of tables, indexes, and clusters

--Index structuring

--Database file I/O distribution

--Table, index, and cluster placement

-- Environment class 不能自动收集,需要: 手工录入/.xdl 导入

-- Workload class 描述数据库日常使用情况,事务请求等, 从:sql cache/sql history/.xdl from previous tuning session

(5) Oracle Expert 提供的 reports 包括:

-- Session Data report 提供关于一个存在 repository 中的 tuning session 的收集数据的明细信息

-- Analysis report 描述 Oracle Expert 生成的建议的基本理由,它一直存在 repository 中,直至生成另一个分析报告

(6) Oracle Expert 采集的数据类别:

-- Instance/Application/Structure/Database/Schema/Environment/Workload

(7) Oracle Expert 的特性

1. 设定 tuning session 的范围

2. 采集数据: 获取综合的信息,包括 database, instance, schema,environment, and workload.

3. Viewing and editing the collected data and rules -- The data is organized in a hierarchical format.

4. Analyzing the data to generate recommendations -- 在收集并按需要编辑了数据之后,Oracle Expert 分析并生成调整建议

5. Reviewing the Oracle Expert recommendations -- 根据调整建议取舍某部分,重新生成新的调整建议报告

6. Generating scripts for implementing the recommendations -- 确定调整建议后生成参数文件和 SQL 脚本日后运行

## 16. Multithreaded Server Tuning Issues 2

(1) 几个相关视图

-- V\$CIRCUIT virtual circuits 的信息,用户通过调度进程和服务进程连接到数据库的信息

-- V\$QUEUE 包含多线程信息队列的信息,用于检查共享服务上是否有竞争 (sum(wait)/sum(totalq),单位 1/100 秒)

-- V\$DISPATCHER 调度进程的信息,检查 busy rate(sum(busy)/(sum(idle)+sum(busy)))

应小于 50%,否则需增加 dispatcher)

-- V\$SHARED_SERVER 正在运行的共享服务进程的信息

(2) MTS 的特性

-- 提供一个共享进程的方式,但不能提高执行速度,可以支持更多的并发用户访问数据库

-- Shared servers use the UGA for sorts, 如果配置 large pool,UGA is stored in the large pool

-- MTS is not intended for batch processing or decision support. 并非用于批量处理和 DSS

(3) 限制 Oracle server 动态启动共享服务数量的两个参数是什么?

-- PROCESSES

-- MTS_MAX_SERVERS

(4) 参数 MTS_SERVERS 设置了 shared servers 的个数,当共享服务 idle 时能否删除?

-- 不能,只有 MTS_SERVERS 较小而 Oracle Server 自动启动的共享服务进程在 idle 时才会自动删除

[17. Workshop 0](#)

## 中小企业如何选择 MRP 系统

魏治

### 前言

对于很多实施大系统的朋友来说,他们感兴趣的并不是这类几万十几万就敢卖的"ERP 系统",但这类系统确实存在,毕竟在数量上还是以中小企业居多,他们的投资能力有限。开发和使用小"ERP 系统"也是很多朋友的应用 ERP 系统的启蒙课,这类系统对人员的要求不高,是他们迈向 ERP 顾问的第一个台阶。

也正如朋友们所看到的一样,这类系统的实施极少有"成功"者。是软件的原因或是公司自身的问题或是.....。

### 本文将分为三个部分:

- 1、从软件公司和工厂两方面介绍他们对小型"ERP 系统"的想法。
- 2、分析工厂在选择小型"ERP 系统"前应准备的工作。
- 3、介绍软件公司在宣传小型"ERP 系统"时常用的猫腻及其应对方法。

### 一、需求的产生

听到 N 个朋友在抱怨 ERP(MRP) 系统实施失败,自己也忍不住要总结一番,或对或错,希望大家能予以探讨。(考虑到大多数的 MRP 软件开发商都喜欢以 ERP 命名自己的产品,我也就随大流称其为 ERP 了。)

这里所说的小企业实施 ERP 系统,主要指计划 ERP 项目总投资在 50 万以下的企业---或许企业本身不小,可在电脑化管理方面却不大肯投资,于是也给很多很多软件公司造就了生存的机会。

对工厂而言,产生这类需求的因素主要包括:

A、实际运作的过程中发现确实需要加强管理了,指望着 ERP 系统能提高管理的效率,可公司里确实不能投入过多的资金。

B、实际运作的过程中发现需要加强管理了,可公司里不愿意投入过多的资金。还常有公司领导要求下属员工买一套盗版的回来用用。

C、看到人家工厂利用电脑管理,觉得挺好,想找一套系统来用用。

而三类原因中以 B、C 居多。

不怎么样的软件开发商加上不明确的 ERP 实施目的也就照成了小企业实施 ERP 系统大多以失败告终。

对于五十万以下的"ERP 软件"无外乎由以下几种途径提供:

A、小软件公司进行开发。

他们本着薄利多销的原则进行销售,卖一单算一单,能赚就赚,反正几个开发人员的工资就那么多,亏也亏不到哪去。至于服务嘛,呵呵呵,能给你改个 BUG 就已经很不错了。当然,讲究信誉的公司也不是没有,但由于资源和开发水准的问题,当客户比较多时就容易出现服务质量降低的情况,进而损害公司的形象。

B、由台湾进入大陆的 MRP 软件。

除了汉康、汉普等比较知名的软件公司以外,台湾还有很多很多小软件开发商,他们的客户群体以在大陆的台资企业为主。或许是大部份中小台资企业的投资习惯吧,各方面都计算得非常精细,选择软件往往也是先考虑价格再考虑性能,这类软件供应商也将研发的基地搬移到了劳动成本低廉的大陆,靠着一两千一个月的工资招聘一大堆刚刚毕业或稍有工作经

验的人在原有系统的基础上开发。价格和人情因素是他们销售的诀窍。

C、软件开发人员私下开发的 MRP 软件。

不由得你不相信,很多在软件公司或工厂内进行 ERP/MRP 软件开发的人员也在考虑自己接单,软件嘛可以在现有的基础上进行修改,最多修改一下界面,靠着几个朋友帮忙推销系统进行简单的原始资本积累----谁都不甘于人下!这类软件的价格自然也是最低的,几万甚至几千都敢卖,一个城市几十万家工厂谁查去?个别能力强的开发人员也能和电脑公司携手,宣传品印出来,谁能分出来这是什么产品?

可以肯定的是,很多小型 ERP 产品开发者都参照了某种或某几种先进的 ERP 系统(或许是软件本身比较成熟与资料相对容易以得到,参考的软件以 MFG/PRO 系统居多),也翻看了不少的资料。

软件开发的工具也以 PB、VB、DELPHI,后台数据库大多采用 MS SQL。

## 二、选择前的准备

ERP 只是有钱企业的事情,小企业就不应该上 ERP 系统了吗?答案自然是否定的,可如何上,应该考虑哪些问题?又应该注意些什么?待我慢慢说来。

作为选择 ERP 系统的主管,首先应该明确:购买 ERP 软件的目的是什么。

这看似是个极其简单的问题,可真正懂得去思考的人并不多,这里或许还包含了 BPR 的内容在其中。

### 1、需要思考。老板的想法是怎样的。

或许是一时兴起,或许是遇到了什么问题,都没弄明白就找来的系统,老板肯定不满意。对于职员来说就必须明了领导的想法。

### 2、计划的花费是多少。

很多人都习惯这么理解:先找了再说,如果 100 万的软件合适就选 100 万的,如果 10 万的软件合适就选 10 万的,可这绝不是个好的方法。在老板的心里其实早就有杆称----SAP 是好呀,可要投入 500 万,他肯吗?同样也不可能把几百万系统的资料和几万块进销存的数据一并交上去吧。需要注意的是:所面对的软件供应商的价格一般是"能屈能伸",百万他敢说能做,或许几万块也能考虑。

### 3、公司内部哪些地方需要调整。

毕竟是实施 ERP 系统,很多流程就与采用手工管理方式存在差异。软件虽然便宜,可该改的地方不改,最终也达不到管理的效果。与大公司不同的是,小型企业的员工一般身兼数职,所需要进行的调整也少得多,但小企业更容易存在管理的漏洞或不规范的地方,如生产计划或库存管理方面的问题,实施 ERP 系统之前如能先行调整,整个实施过程将事半功倍。

很多情况下,老板的要求并不高,只需要通过内部职能的简单调整就能达到,这时又何必非得用 ERP 系统呢?

### 4、实施 ERP 系统的步骤。

实施 ERP 系统没有一步登天的。先做哪一部分,再做哪一部分事先得考虑清楚,如果领导认为 ERP 只是个软件,那么在选择软件之前也必须纠正他的想法,否则软件买回来三个月看不到效果他可会烦。详尽的计划也能让老板了解实施系统的过程中可能遇到的问题,必要时给予有力支持。

需要指出的是,这里所规划的实施步骤并不是实际实施的步骤,仅仅作作为一种参考。

### 5、公司内部的数据资料

选择 ERP 系统之前,必须准备公司运作的实际数据供系统测试之用。如销售订单数据、生产计划数据、工程清单数据等等。这类数据在其他文章中已有详细介绍,在此从略。

### 三、选择小型 ERP 系统时应该注意的问题

个人认为，选择小型 ERP 系统应按以下几个重点依次考察：

- A、软件公司的实力
- B、软件产品的质量
- C、软件的实施情况
- D、其他事项

#### 1、软件公司的实力

这是个很广泛的话题，可以从几个方面考察软件公司的真正实力：

##### A、软件的开发人员数量。

只有三五个开发人员的公司能开发出什么产品？可事实上这样的公司很多，尤其在使用 FOXPRO 进行开发的年代，这更是常事。

也有个别开发人员敢于叫嚣自己开发出的产品比知名的 XXX 软件更好更先进，当然我不否认他个人的能力，但在此提醒各位：三个臭皮匠胜过诸葛亮。而且你也必须考虑：当用户更多时，该公司还有没有人力进行技术支持。

##### B、软件系统开发的时间

近两年才开发出的产品几乎没经过什么实际测试，很多问题自然难以暴露。用得越多、时间越长，低级错误也就越少(不是没有错误)，很多错误只能在实践中才能发觉。

在实际的运作中主要还是查看该公司软件以前的版本----常有公司把新出的软件定名为 10.0，可 1.0 跟 10.0 之间连界面与功能都没有任何改动。

##### C、软件的开发文档

一个正规的软件公司，必定有正规的软件开发文档，几个高级程序员信手捻来的程序或许很好，一旦出现人员变动.....。即使人员没有进行更动，在开发了三个月后还要回过头去修改原来的程序，记忆也难免会出现遗漏。所以，我个人认为：正规的公司必定有正规的开发文档。

#### 2、软件产品的质量

软件产品的质量主要通过两个方面了解：

首先，可以从该公司已有的客户处了解，最好能在原有客户那里参观。这是最直接的方法，但在实际的操作中这也是最不现实的方法。

作为软件公司，他能提供的一般只是实施比较成功的客户，由于 ERP 系统涉及到了企业内部的管理信息，能允许别人参观的企业与软件公司之间的关系一定相当紧密----这样的参观没有太多实用价值。当然，这也证明该软件公司的确具有成功案例。

软件公司也常会虚构出一些客户（尤其是和公司业务相近的客户），对于很多刚刚研制出产品的软件公司，他们担心客户“成功经验”看得太重，只好虚构客户。一个城市里有几万家工厂，能查到吗？即使有这样的工厂，也很难知道其内部的管理模式。到了要求参观的时刻，总能找出些理由搪塞。

其次，试用软件产品。

作为小型的 ERP 系统，由于对自己产品不甚放心，通常他们会采用些借口搪塞。当然也有部分企业已经生产了相对成熟的产品，还是愿意提供给客户试用。试用的过程中应注意：要测试公司内能用到的所有类型资料。如，生产、销售、工程等等。虽然不一定要输入全部的资料，但每个大类的产品资料都应予以测试。财务系统则还需要测试月结处理和年度关帐等。在测试过程中发现的所有问题均应予以登记，以考察软件公司对错误的处理情况。具体输入时也可以采用“摧残”的办法，即输入一些明显的错误资料，



检查系统的容错能力。我曾遇到这样一个例子：销售定单的数量居然允许保存字母，但运行 MRP 时系统崩溃。

其三，测试软件的运行速度和操作灵活性。

真正使用 ERP 系统时必定有大量的资料需要输入，如果此时只能用鼠标操作，那录入资料的速度将大大降低。同时也因用户同时操作而造成系统反应变慢，这会影响到用户操作系统时的情绪。

### 3、软件的实施情况

作为一套管理系统，并不是买来就能用的，还需要一个实施的过程，很多成熟的软件和管理正规的工厂本能够通过 ERP 系统提高经济效益，却因实施过程不顺畅而导致应用失败。所以购买软件时也应考察软件公司的实施能力（对于小公司而言，主要指培训指导能力）。

选型的过程中，可以考察软件公司的培训方式和培训文档。同时也应同实施人员交流，了解其对 ERP 系统的理解程度，甚至可以要求其试讲。

对于软件公司的实施能力或许是最难量化的指标，具体操作时因具体情况而异。

### 4、其他应注意的事项

应用小型 ERP 系统的过程中最大的麻烦就是软件公司出现异常。如主要开发人员出现变动，售后服务人员出现变动，甚至整个公司垮台。

当然，这类问题并不一定在选择 ERP 系统时便能完全察觉，但我们也应采取相关的防范措施。

A、选择比较成熟的软件，有可能的话最好能拿到系统的源程序，以便自行修改（这种方式需要投入额外的人力和资金，除非特殊情况，不建议采用）

B、选择具备二次开发能力的软件。如果软件本身比较成熟，且具备基本的二次开发功能（最少也因包含自定义报表开发），那么在使用的过程中将很少受制于人。

C、在合同中明确规定意外情况发生时的处理方法。具体的使用过程中，对报表的修改应该是十分频繁的事情，各个时期的需求不同，如果软件不具备自定义报表的开发功能，那么每次修改都将依赖软件公司。对工厂或对软件公司而言时间、经济成本都不合算。

## 四、总结

ERP 系统的选择直接关系到公司该项目应用的成败，故在供应商的选择方面必须谨慎小心，如果公司领导对该项目没有迫切的需要与认识，建议还是不要启动为佳。

对于软件的选择，除了对产品的考察外，更重要的是对软件供应商的考察。

ERP 系统项目也号称“一把手工程”，无论项目的大小都需要得到公司高层的大力支持，而这种支持也与项目负责人员与其的适当沟通息息相关。由于涉及到各个层面的利益关系，领导只有了解详细了解项目的进度和选型实施过程中的问题与利弊之后才能给予有效的支持。这点和其他项目所谓的“我放权，你去做”不尽相同。

## CRM 的核心、应用重点及分布实施步骤

李际

CRM, 中文直译的全名是"客户关系管理"。它起源于上个世纪 90 年代初中期, 在中期得到了迅猛的发展。现在一度被认为是接替 ERP 的下一个管理软件应用热点。仅仅从 SEIBEL 在 NASDAQ 上的表现就可以知道资本市场对于其的信心与希望。我想就 CRM 的核心理念及其与其他理论发展的内在联系、CRM 应用的重点与其组成、CRM 分布实施的步骤三点做简明论述。

CRM 的核心理念其实只有一个词组: ONE TO ONE MARKETING(一对一营销)。对于这个名次解释已经有太多的普及性宣传, 不过是否能真正理解它是一个大问题。我理解的 ONE TO ONE MARKETING 是真正彻底的个性化营销。这个个性化并不仅仅停留在你计划营销的个体对象上, 因为影响你营销结果的其实有许多相关的个体。比如讲, 你是面向企业客户的供应商, 但是你不能仅仅停留在企业个体的单一对象上进行营销。有权力作出决定和有能力影响作出决定的所有企业内外的部门、个人都应该你营销的对象。我们一定知道整合营销这个名词。整合营销按照大众化的观点应该是整合自己的所有资源与人力去进行系统化的营销, 而从另一个角度看, 被营销的各个对象都被系统化地组织起来也应该是整合营销的另一个方面。

ONE TO ONE MARKETING 的理念还体现了营销理论的变迁。营销理论大致经历了从面向全社会的整体营销到面向特定群体的团体营销到现在的面向单一个体的个性化营销。其变迁的背后其实反映的是社会的进步与供求关系的变化。社会从早期物质的大量缺乏到后期物质相对满足供给到现在物质过剩, 供求关系从早期几乎处于垄断的绝对供小于求到后期的供求基本平衡以至现在相对供大于求都在幕后主导着营销理论的变化。这里需要说明我指的物质过剩并不是绝对意义上的物质过剩而是能够满足客户需求的物质过剩, 所以我在供求关系变化中提到相对的供大于求即指的是相对与客户需求, 供大于求。而从绝对的供应量来看, 供求关系并未失衡到一个极端的比值。这似乎很难理解而且有些自相矛盾。

我举一个例子来帮助大家理解这个论点。我们都知道世界上有很多地方穷得连满足生存的粮食都不够, 可是由于种种原因, 其他地方的粮食因为当地的产量远远超出当地的实际需求而造成粮食过剩。所以在这种情况下, 我们并不能说粮食是供大于求(因为其实是资源分配不均的问题), 只能相对的该些产粮大国的粮食供大于求。这些情况的本质起因因为其复杂性与多样性所以并不在此讨论。正是因为上述的变化, 原来只需要满足全社会需求的产品再也不能如此畅销, 因为大量的相同产品打破了原先动态平衡(意指动态稳定)的供应关系, 迫使厂商不得不把自己的产品定位于更细的一个群体以谋求供应关系恢复平衡。同理造成厂商开展 ONE TO ONE MARKETING。我还能从许多相关理论来解释这种营销理论变化。关于生产的理论同样经历了从满足对于产品的需求到满足使用的需求最终满足体验的需求的变化。

第一种其实是提供功能的使用, 是彻底物质化的。而第二种、第三种是非物质的, 其中第二种是提供价值, 而第三种是满足客户感性的需求。因为社会的进步使客户不再注意物质的价值了。具体讲, 第一种理论你可以解释当初汽车发明造成的轰动, 因为它满足了人们轻松远足的需求, 提供的是代步的功能; 第二种理论可以解释珠宝业存在的事实, 因为它满足了需要炫耀的心里需求, 提供的是珠宝的货币价值(而不是它的耐磨、耐高温等工业上的使



用功能，就好像金刚石与钻石的区别一样)；第三种理论可以解释唱片业的存在意义，因为它满足了客户需要发泄、寻找精神寄托的体验需求。特别是新近流行的自己给自己灌录唱片更是证明了 ONE TO ONE MARKETING 理论是基于市场需求而产生。而这种自灌唱片的业务不就是 ONE TO ONE MARKETING 在该行业应用的成功案例吗？再参考市场理论，它相似地经历了从满足客户需求到引导客户需求乃至超越客户需求与期望的变迁。这些变迁一是反映了因为供求关系的相对变化导致市场的顺势变化二是间接反映了 ONE TO ONE MARKETING 在此处的折射。所谓超越客户需求就是大家耳熟能详的"120%的满意度"或"惊喜体验（惊喜度）"等类似的提法。只有实现了 ONE TO ONE MARKETING，才能真正实现所谓的超越客户需求的目标。

为了帮助大家理解，我再次举一个例子。在 ONE TO ONE MARKETING 的具体理论中，有一条细节就是要注意个体客户的个性化信息，比如客户的生日等。一个与你有业务往来的客户从来不会主动有意识希望得到你的生日祝福，因为他觉得与你仅仅是业务关系，而生日祝福应该来自于诸如亲人、朋友等处。一旦你给他发出生日祝福，他一定会非常地惊喜，意即超越了他的期望值。根据他的经验，因为生日祝福只可能来自于亲人于朋友，所以你也就成了他的朋友而不是简单的生意上的合作者。市场理论发展到今天，提出的以超越客户需求与期望为目标的理论业从侧面与 ONE TO ONE MARKETING 的诞生与发展相辅相成。而通过对 ONE TO ONE MARKETING 的一系列解释自然也就已经解释了 CRM，大家应该可以对于其有一个深刻的理解了。CRM 的应用重点无非是客户管理与执行、客户信息管理与共享、潜在商机的管理与发现三点。这里面的组成各不相同。客户管理与执行由营销自动化与销售自动化两个部分。营销自动化一般是由企业市场部门使用的工具，主要是通过大量自动化的工作来实现普遍而非特定的客户管理，其目的无非是客户兴趣的唤醒、客户关系的维持与增进，充其量是为下一步销售自动化做铺垫。具体的应用有自动化的邮寄资料或个性化服务（比如生日祝福、预定的新品广告）等。销售自动化一般由销售部门使用，主要通过对于具体的目标客户的销售流程的管理与控制提高成交与再次成交的绩效。它的一些具体的手段有根据不同的销售阶段进行进阶管理（这同时也体现了内部管理地精神），根据客户诸如个人喜好等进行个性化营销，通过快速、有效的搜索与关联信息的提示帮助克服销售过程中的问题。客户信息管理与共享由知识中心与 CALL CENTER（电话中心）组成。

**知识中心帮助相关人员能够寻找到有助于解决当前问题的方法。**它有点类似于过去常提到的专家系统。知识中心不同于一般的文档管理的最大不同点在于需要在大量的数据中迅速寻找并配对，这对于计算机技术是一个很大的挑战。目前绝大多数的 CRM 产品仅仅能提供一个雏形。而 CALL CENTER 可能是 CRM 中发展最早与最成熟的应用了。早期的 CRM 几乎就等同于 CALL CENTER。它主要有电话排队、自动提示等功能实现电话服务。它另一个特点是高度强调与硬件的集成。所以目前这方面的著名厂商均是由硬件向软件转移过程中发展出来的。无论是客户管理与执行还是客户信息管理与共享都是对于显性知识的管理，而潜在商机的管理与发现就是对于隐性知识的挖掘了。两者结合起来正好构成了知识管理的两个应用点。所谓潜在商机指的是原本并未表露出来的商机，就好像隐性知识是隐藏在显性知识（即你已知知识）的背后，你是不知道是否有你需要的知识而且这知识是什么都是未知的。只有通过分析与挖掘才能知道有没有和是什么。这个应用是利用数据仓库技术实现的。

同样，为了帮助大家理解，我介绍一个经典案例。美国有一家超市长期苦于无法提高啤酒的销售量。通过数据仓库技术的利用与长期数据的积累后，该超市发现一个有趣的情况：大多数购买婴儿尿片的都是男性。于是它采取了一项匪夷所思的举动，把尿片与啤酒陈列在

同一行货架上。结果因为购买尿片的同时，绝大多数男性都会随手买些啤酒使啤酒的销售量在短期内翻了几番！谁能事先想到啤酒居然与尿片还有什么关联性而且两者分属于不同分类的个人消费品，传统做法从来不可能把两者扯上任何关系。这就是潜在商机的发现，也是数据仓库作为计算机技术在营销上的商业意义的体现。做任何事情都不可能一蹴而就，实施 CRM 也不例外。必然需要一步步的实施，但是分布实施的步骤又如何确定呢？我觉得很难一概而论。根据行业属性的区别、企业个体差异的区别会造成针对性的不同。

比如讲，对于有普遍义务的行业，诸如：邮政、通讯，其重点一定是 CALL CENTER 去解决客户服务与咨询的需求；对于以重复消费为特征的行业，诸如：百货零售业，其重点一定是营销自动化去激发客户消费的意愿；对于以薄利多销为特征的行业，诸如：交通运输业，其重点一定是数据仓库去发现克服成本与开辟新商机的可能...不过我仅假设一个以制造为特征的企业作一个泛泛的建议：先实现销售自动化继而实现营销自动化，第三步是建立知识中心，在此基础上开展对数据仓库的使用，最后完成 CALL CENTER 的建设。我这样建议的原因是各个应用对于该类型企业重要性与使用频率的递减性考虑。而且也考虑了实现的成本与难易程度。从低成本、简单易行处开始可以逐步渐进，增加企业实施 CRM 的信心，应用 CRM 的决心。而且这些难度的增加并不仅仅是实施工作量的增加，而是需要前一步基础性工作完成作为基础。例如没有销售自动化与营销自动化中长期积累的准确、及时、大量的数据就不可能指望数据仓库可以提供正确、巨大的发现。我已经就 CRM 的核心理念及其与其他理论发展的内在联系、CRM 应用的重点与其组成、CRM 分布实施的步骤三点做了简明论述，其中许多的理论虽然来自于国际上公认的理论体系但结合了个人的理解与见解。所以可能与传统的理论体系在结构组织与论点展开上略有不同，希望是通过对于理论的学习帮助大家能有深刻而非形而上学的理解与应用。

### 中小型 ERP 市场及软件的特点

ERP，原本是大型企业的专有产品。所以一开始，许多的软件都设计成大型机系统，包含了极其复杂、一应俱全的功能。无论是从 TCO（总体拥有成本）还是适用性上都让中小型企业望而却步。目前大型企业均已纷纷添置 ERP 系统，而随着市场竞争的加剧、中小企业的崛起和 ERP 厂商本身扩展客户的需要，面向中小企业的 ERP 市场越来越红火，大有赶超大型 ERP 市场的趋势，本文仅就中小型 ERP 的需求及中小型 ERP 软件的特色做一讨论。中小型企业虽然没有大型企业在生产及业务上如此地复杂，但是“麻雀虽小，五脏俱全。”它还是有着一整套进、销、存、计划、生产和财务成本管理的需求。与大型企业所不同的仅仅是在于各个功能不象大型企业要求的如此复杂、全面。换言之，有关 MRPII 方面的功能是各种生产型企业的必须。中小型企业由于受资金的限制，不可能像大型企业般一掷千金，花上十几、几十万美元引进 ERP。他们往往只有几十万、甚至十几万的预算。钱多好办事，钱少一样要办好事！企业对于 ERP 的虔诚往往使我们这些专业从事 ERP 的人士被感动。要知道，这些钱很可能就是该企业当年非生产资料采购中最大的一笔预算。所以，中小型企业往往是抱着少花钱多办事的原则。

中小型企业自身规模尚不足与大型企业相提并论，所以对 ERP 的详细需求就不如大型企业。许多在大型企业所特有的需求在中小型企业就没有必要。反言之，大型 ERP 软件一直引以为荣的“功能全面”在中小型市场有可能成为“操作繁琐”的代名词。也因为中小型企业缺乏专业的 IT 人员，所以对于系统的可维护性及简便性提出了高要求。中小型企业随着自

身的发展，纷纷引进了许多管理制度，例如：成本独立核算、多公司、多生产方式、跨行业生产、工商业并行等。这些对 ERP 系统就提出了极高的要求，许多针对单一行业设计的系统就显得捉襟见肘。 目前在国内中小型软件正处在国内外争雄的时期。国内的中小型软件从以用友的"M"系列、金蝶"K3"系列为代表的产品向下延伸到博科、安易乃至许多仅仅几千元的 MIS 系统不一而足。国外的产品从"JPOBS"集团的面向中小型企业的解决方案到以 QAD、SYMIX 为代表的 TIER2 软件乃至为中小型企业设计的 SCALA、MARCOLA、MRP9000 等等也是林林总总。 国内的软件由于起步较晚，使用的大多是新的技术，这一点正好与国外的软件略有不同。

所以国内软件有的强调"技术先进性"，颇得 IT 人员的好评，国外软件强调"成熟性、功能性"，又博得应用人员的青睐。 由大型软件演变而来的中小型系统可能多多少少带有过去"大而全"的遗风。一些小型企业无论从资金还是适用性上仍是很难接受。许多中型软件由于具有较好的延展性，比较多获得用户的首肯。但是事物总是具有两个方面的，中型软件的延展性本身也成为被频繁讨论的话题。对于较大规模的企业略显不足，对于小规模的企业又略感庞大。中小型软件由于针对性较强，也占有大量的市场。这些产品具有操作简便、功能适用的特点，是一批企业的首选。 由此看来，中小型市场也是众说纷纭、众口难调。如果依笔者观点，应该视具体情况考虑。一些已经发展相当成熟、有大规模经营潜力的企业可以重点了解大型软件的解决方案；一些发展多年、有着固定模式的企业可以采用中型软件；一些仍在起步阶段或者希望通过 ERP 改进管理的企业可以从中小型软件着手，不必一步到位。

如果企业是以生产、计划等为重点，国外软件有着多年的经验，这部分本身也是 ERP 实施的难点所在；如果企业是以进、销、存和财务总帐（MIS）为重点，这相对简单的部分国内软件已经具备能力。特别值得一提的是，在大约 30-50 万人民币范围内，是目前国内外软件厂商争夺最激烈的领域。具备上述投资额规模的企业也是可选方案最多的，所以在货比三家的同时也要谨防挑花眼。仍建议以"功能性、实用性"为重点。 纵观国内中小型 ER 市场应是相当活跃的，在很长的一段时间内将撑起国内 ERP 市场的半壁江山。这也是为什么众多厂商纷纷推出中小型企业解决方案，许多中小型软件厂商一改过去姿态，自豪的声称专为中小型企业设计的原因。

## Oracle Applications 11i 技术支持经验杂谈

Kevon zeng

从事 Oracle Applications 的技术支持快要一年，虽然 11i 版本并不象人们期待那样能够另人满意，但是经历了一段过渡期，11.5.4 总算能给喜欢 oracle applications 的朋友带来了一点希望。三个月前在跟同事聊天时，我们在开玩笑说"自从 11 升级到 11i，我们就象上了贼船，我们没有回头路，我们只能硬着头皮，咬着牙，希望能看到大洋的彼岸"，现在总算依稀可见大洋的彼岸，看来 oracle 这盗贼还是有点良心，虽然让我们等了太久，但是没把我们半途丢下已经是不错（阿 Q 一下），但愿我们能够顺利到达彼岸，但愿 oracle 不要使喜欢 oracle 产品的发烧友们再次伤透了心。也许有好多的但愿，但这都是由于中国的 ERP 还处于起步阶段，我们不得不向国外的老大哥低头，希望有一天我们可以对 Oracle ,SAP ,Siebel ,PeopleSoft 等 ERP 厂家说"不"，更希望我们的 ERP 厂商能够在世界 ERP 的舞台上与这些老牌的 ERP 厂家相抗衡，也许这离我们有点遥远，但如果我们大家一起努力吧，也许会有那么一天的！

废话说了这么多（呵呵，没办法，最近久未写稿，发点牢骚也是应该吗，过去写稿总希望多写点废话，多赚点稿费，现在给大家写稿，没稿费，但也值得，因为我们一起探讨，一起成长），该进入正题，不然有人就要打我啦！

首先，我声明我是一个 Oracle Applications 初学者（有点"冒天下之大不韪"吧），下面的所有问题和解决方法以 FAQ 形式出现，权作抛砖引玉，希望有更多的 Oracle Applications 高手加入我们的行列，介绍他们的经验，让我们一起进步。

### 1. 在 oracle applications 11i 中如何快速获得上次查询的 SQL 语句

方法：我们不必使用 SQL Trace 来获得，有一个简单的办法，即：帮助->诊断->检查->在"块"中输入"system"，在"字段"输入"last_query"，然后把鼠标移动到"值"，你就可以看到你所要的 SQL 语句，简单吧！

### 2. 如何更改 Applications 中 LOG 的存放位置

方法：(以 AIX 为例)在 11i 中默认 LOG 的存放位置是各个模块下 log 目录，例如 FND 模块 log 的存放目录就是 \$FND_TOP/log/，可能有些用户想统一管理 log，可以采取下面这种方法得到解决：

更改 applprod 用户（用于启动 FormSever , ReportServer 和并发管理等）的 .profile，增加两行：

```
APPLCSF=/u10/oracle/prodcomn
export APPLCSF
```

然后重新以 applprod 登录，重新启动 FormSever , ReportServer 和并发管理等（一般有个 shell 文件，例如 start.sh），至于为什么能起作用？原因很简单，Applications 首先查看 APPLCSF 环境变量是否设置，一旦设置，所有的 log 均存放在 \$APPLCSF/log；否则就存放在各个模块的 log 目录

### 3. 如何检测 11i 的数据库是否符合 CBO ( Cost Based Optimizer ) ?

方法：CBO 对整个 11i 的性能影响很大，如果你的数据库不满足 CBO 的条件，那么象订单的查询，总账的 drilldown，MRP 等速度就会慢得无法忍受，那么如何检测呢？11i 提供了一个检测工具 \$FND_TOP/sql/AFCHKCBO.sql，如果不满足条件，会以 * 显示，但是 optimizer_max_permutations 最好设置为 2000，optimizer_features_enable 为 8.1.7；如果



_push_join_predicate 没有设置为 true ,说明你的系统的性能还有很大提升的余地 ,在 11.5.3 中由于存在一些 bugs ,这个参数不能设置为 true ,所以建议 11.5.3 的用户尽快升级到 11.5.4 或者 11.5.5 ,这样就能解决 OM,GL,MRP 的性能问题。

#### 4. 如何收集统计数据？

方法：在 11i 中，你如果启用了 CBO，你就要定期的收集统计数据，因为 CBO 利用统计数据来优化 SQL 语句，那么如何收集统计数据？有两种方法，一种在 11i 系统中以系统管理员的身份提交并发请求，包括"统计数据收集模式"、"所有收集列的统计数据"、"收集列的统计数据"等（有关用法，请参见 Oracle Applications System Administrator's Guide）；另一种方法是执行 SQL 语句（其实并发请求的原理跟它是一样），Oracle 提供 fnd_stats 包，例如 exec fnd_stats.gather_schema_statistics('ONT')用来收集 OM 模块的统计数据，exec fnd_stats.gather_schema_statistics('ALL')用来收集所有模块的统计数据（如果大家对 fnd_stats 这个包感兴趣的话,可以用 pl/sql developer 打开来研究）。在我们的系统中，我一般采取定期提交"统计数据收集模式"对所有的模块收集统计数据（每隔 12 小时），然后在特殊情况下，手工优化某个模块（OM 模块比较常用）

#### 5. 如何清除一些历史数据？

方法：Applications 11i 的许多模块都提供清除相关历史数据，例如 fnd 模块提供"清除过时的工作流运行时数据"的并发请求，你可以清除 OM 模块的旧的工作流的历史数据（目前我们系统中每隔 3 个月自动清除）；fnd 模块提供"汇总和清除并发请求统计"，你可以清除提交并发请求产生的 log（目前我们系统每隔 7 天自动清除）。大家有兴趣可以去研究一下其他模块的提供的清除历史数据的并发请求，毕竟系统用了一阶段，数据量剧增，给备份，硬件和整个系统性能都会有很大的影响，做一些必要的清除工作也许能解决燃眉之急，而且能减少整个硬件的投资

#### 6. 如何编译 Forms？

方法：经常我们要修改 Applications 11i 的 Forms，所有 Forms 的源文件都放在 \$AU_TOP/forms/ZHS（US）下，你可以利用 FTP 取回来修改，再利用 FTP 传到 \$AU_TOP/forms/ZHS（US）下覆盖原文件（记住覆盖前要备份原文件和它所对应的.fmx 文件），那么如何编译修改文件，你可以使用 f60gen 来编译，例如：

```
f60gen INVTMTX.fmb userid=apps/apps
output_file=$INV_TOP/forms/ZHS/INVTMTX.fmx
```

#### 7. 如何获得某个 Forms 文件的版本号？

方法：当你跟踪 iTar 时，Oracle 工程师有时会叫你提供某个 Forms 版本，你可以采用如下方法获得（以 AIX 为例）：Strings filename|grep Header

#### 8. 如何自动 retry 通知（Notification）？

方法：作为 Applications 系统管理员，经常要处理一些无关紧要的通知，为了解决这个问题，我特意研究了其 retry 过程，大家可以把以下 sql 语句作为并发程序，定期执行，这样就不用再去一个一个处理：

```
REM
+=====
=====+
```

```
REM ||
REM | FILENAME |
REM | WFNR.sql version |
REM ||
REM | USAGE |
REM | sqlplus <appsuser/pwd> @WFNR.sql |
REM ||
REM | HISTORY |
REM | 05/06/2001 kevon zeng Created |
REM | 27/06/2001 kevon zeng Modified | |
REM
+=====
=====+
declare
vh_fnames varchar2(255);
vh_fvalues varchar2(255);
vh_fdocnames varchar2(255);
vh_counter varchar2(255);
cursor cr_nid is
select n.notification_id
from WF_NOTIFICATION_ATTRIBUTES NA,
WF_MESSAGE_ATTRIBUTES_VL MA,
WF_NOTIFICATIONS N
where n.status='OPEN'
and n.message_type='WFERROR'
and NA.NOTIFICATION_ID = N.NOTIFICATION_ID
and MA.MESSAGE_NAME = N.MESSAGE_NAME
and MA.MESSAGE_TYPE = N.MESSAGE_TYPE
and MA.NAME = NA.NAME
and MA.SUBTYPE = 'RESPOND';
v_nid wf_notifications.notification_id%type;
icount number;
begin
icount:=0;
open cr_nid;
loop

fetch cr_nid into v_nid;
exit when cr_nid%notfound;
vh_fnames:='RESULT#LOOKUP#WFERROR_RETRY';
vh_fvalues:='RETRY';
vh_fdocnames:='';
wfa_html_util.SetAttribute(v_nid, vh_fnames, vh_fvalues, vh_fdocnames);
wf_engine.preserved_context := FALSE;
```

```
wf_notification.Respond(v_nid, null, 'SYSADMIN');
icount:=icount+1;
end loop;
fnd_file.put_line(fnd_file.log,'deal with '||icount||' lines');
end;
```

### 9. 如何处理大量处于 Fulfillment 阶段的订单？

方法 在 11.5.3 或 11.5.4 订单工作流经常停在 Fulfillment 阶段 (这似乎是 oracle 的 bug), 你必须手工一条一条去处理, 于是我就写了下面一段 SQL 语句, 然后做成并发程序, 定期运行, 就解决了手工处理的麻烦

```
rem HEADER
rem $Header: wfretryFulfill version 1.0 $
rem INFORMATION
rem created by kevon zeng 2001.6.26
rem NAME
rem WFRF.sql - WorkFlow Handle Fulfillment Error
rem DESCRIPTION
rem if fulfillment error happen,retry fulfillment

set verify off
declare
icount number;
v_item_key wf_item_activity_statuses_v.item_key%type;
cursor crFulfillmentError is
select item_key
from wf_item_activity_statuses_v
where item_type = 'OEOL'
and activity_name = 'FULFILL_LINE'
and activity_status_code = 'ERROR';
begin
icount:=0;
fnd_file.put_line(fnd_file.log,'all line_ids to deal with:');
open crFulfillmentError;
loop
fetch crFulfillmentError into v_item_key;
exit when crFulfillmentError%notfound;
wf_engine.handleerror(itemtype => 'OEOL',itemkey => v_item_key,
activity => 'FULFILL_LINE',command =>'RETRY');
fnd_file.put_line(fnd_file.log,v_item_key || ';');
icount:=icount+1;
end loop;
fnd_file.put_line(fnd_file.log,'deal with '||icount||' lines');
end;
/
```

```
set verify on;
commit;
```

#### 10. 如何调试 Applications 11i 中的包？

方法：我想大家都有自己的经验，我经常使用的一种方法，就是建立一个叫 temp 中间表，然后往这个中间表写入调试信息；往往 oracle 也提供了很多调试的包，例如 fnd_file 等

#### 11. 价目表丢失了某些 item 的价格

方法：在输入订单行时，系统报错说"此价目表不包含此 item 价格"，但在价目表增加这一 item 时，系统又会报错说此记录一存在（这可能是升级过程导致的错误），采取的方法是提交"QP：维护 QP 限定词中的异常数据"并发请求。在使用 oracle 提供的并发请求之前，我们曾经对 pricing list 进行研究，发觉是 qualification_ind 字段在作怪，我们就使用了下列语句进行手工更新

```
update qp_list_lines qll set qll.qualification_ind=4 where list_line_id=?
update qp_pricing_attributes set qualification_ind=4 where list_line_id=?
```

其中？代表 item 所对应的 list_line_id,你可以使用下面语句获得:

```
SELECT qll.list_header_id
,qll.list_line_id
,qll.qualification_ind
,msi.inventory_item_id
FROM qp_list_lines qll, qp_pricing_attributes qpa, mtl_system_items_b msi
WHERE qpa.product_attr_value = to_char(msi.inventory_item_id)
AND qpa.list_line_id = qll.list_line_id
AND qll.list_line_type_code IN ('PLL', 'PBH')
AND qpa.pricing_phase_id = 1
--AND qpa.qualification_ind IN (4, 6, 20, 22)
AND msi.segment1 = ?
```

其中？代表 item 的编号

但是为什么 qualification_ind 有 3, 4, 6, null 等的数字，查看了 reference guide，只是说这是个跟性能相关字段，不知道后台价格引擎跑了些什么，希望哪位大虾知道告知一下。

（未完待续）

不知大家对这样文章是否感兴趣，如果大家感兴趣的话，我将在下一期的电子杂志中继续把我的一些经验分享给大家，也希望大家向我提供更多的经验，或者直接向电子杂志投稿，如果大家的经验足够多的话，我们将按 module 进行分类，并以 FAQ 的形式放在我们的论坛中。

最后，还是那句老话，"让我们大家一起努力吧！"。



## 论坛精华

### 数据库管理员论坛：

[一次数据库起不来的经历，供参考](#)  
[为什么我的 oracle 进程始终占用 90%cpu 的时间？](#)  
[Oracle 手工建立数据库的完整步骤](#)  
[转贴\]优化调整 Oracle 8i 数据库](#)  
[如何确定操作系统读缓冲区的大小呢？](#)

[更多](#)

### Oracle 开发：

[三层体系结构中查询处理的问题？](#)  
[function 能否返回数组？](#)  
[你们是如何调试存储过程的？？](#)  
[在优化 sql 时一般遵循什么原则？](#)  
[Develop2K 6.0 常见问题解答](#)

[更多](#)

### Oracle 认证：

[通过 OCP 8iDBA 5 门考试的经验](#)  
[很久以前写的几个脚本,日常用的](#)  
[培训: 023 lesson one](#)  
[培训: 023 Lesson Two](#)  
[培训: 023 Lesson Three](#)  
[培训: 023 Lesson Four](#)

[更多](#)

### ERP&CRM：

[基本理论---历史演变，管理内核和最新发展](#)  
[Oracle9iAS Web Cache 多种拒绝服务和缓冲区溢出漏洞\[中联绿盟翻译整理\]](#)  
[basic thesis----实施模式与风险防范](#)  
[sap 46+oracle8i+安装手册](#)  
[chao ,here a 3 useful website:](#)

[更多](#)

## 制作人员

主编：

Oldwain

编辑：

ITPUB 各版块斑竹：

chao_ping , guo , owen , parrotao , snowwhite2000 , coolsword , flytiger , jlandzpa , xiaomiao ,  
billicon , easyfree , yyg , kevon_zeng , w39 , WilliamGui (排名不分先后)

排版、PDF 制作：

痴痴地等

ITPUB 管理团队

2002-1-1 创刊