

IBM

推荐使用教材

DB2 数据库 管理与应用教程

庄济诚 编著



清华大学出版社

DB2 数据库管理与应用教程

庄济诚 编著

清华大学出版社
北 京

内 容 简 介

IBM 公司的 DB2 通用数据库为企业提供领先的商业解决方案, Version 8.1 是其最新版本。本书着重于概念的介绍, 包括 DB2 的基础概念、数据库与表格空间、数据库中的对象、数据的搬运、备份与恢复、数据锁定、数据库系统效能、安全控管、基础 SQL 指令及高级 SQL 指令等。本书在理论讲解的同时, 还辅以适当的范例帮助读者了解应用的方式。

本书适合 DB2 初学者使用, 也可以为具有一定数据库管理或编程经验的数据库管理员或程序员提供参考。

本书繁体字版书名为《爱上 DB2 资料库管理与应用》, 由文魁资讯股份有限公司出版, 版权属庄济诚所有。本书简体字中文版由文魁资讯股份有限公司授权清华大学出版社独家出版。未经本书原版出版者和本书出版者书面许可, 任何单位和个人均不得以任何形式或任何手段复制或传播本书的部分或全部内容。

北京市版权局著作权合同登记号 图字: 01-2003-8767

版权所有, 翻印必究。

本书封面贴有清华大学出版社激光防伪标签, 无标签者不得销售。

图书在版编目(CIP)数据

DB2 数据库管理与应用教程/庄济诚编著.—北京: 清华大学出版社, 2003
ISBN 7-302-07805-X

I.D… II.庄… III.关系数据库—数据库管理系统, DB2 IV.TP311.138

中国版本图书馆 CIP 数据核字(2003)第 116040 号

出 版 者: 清华大学出版社 地 址: 北京清华大学学研大厦

<http://www.tup.com.cn> 邮 编: 100084

社 总 机: 010-62770175 客户服务: 010-62776969

责任编辑: 刘 颖

封面设计: 陈刘源

印 刷 者: 北京昌平环球印刷厂

装 订 者: 三河市化甲屯小学装订二厂

发 行 者: 新华书店总店北京发行所

开 本: 185×260 印张: 16.75 字数: 398 千字

版 次: 2004 年 2 月第 1 版 2004 年 2 月第 1 次印刷

书 号: ISBN 7-302-07805-X/TP·5689

印 数: 1~4000

定 价: 28.00 元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系调换。联系电话: (010)62770175-3103 或(010)62795704

前 言

我在 IBM 软件事业处的工作,有很大一部分是教育训练,对象包括客户、经销商及 IBM 公司的同事。一开始所采用的是规定的教材,但是总觉得内容不是很符合学员的需要,前后逻辑顺序也有问题。于是从 4、5 年前我开始编辑自己用的教材,将平日所看到的用户需求及遇到的问题纳入教材中,同时也加入 DB2 新版本的每个功能,所以我每次授课的内容总有些不同。我常对上课的学员感到抱歉的是,我的教材的说明文字不够详尽,所以除了上课外,学员很难自行在课后研读教材。于是我兴起了编写本书的念头,希望以我教材中的大纲为基础,配合详细的说明与范例,以及比较合逻辑的顺序,让读者在阅读本书后,能对 DB2 有清楚的认识。

本书是以 DB2 Universal Database Version 8.1 为基础,内容着重于概念的讲述,希望让读者对 DB2 的许多功能与操作从原理层面开始了解,如 DB2 的结构与环境、磁盘空间的运用、备份与恢复、数据锁定、索引结构的运用、SQL 指令的创建等,都有详尽的概念说明。在说明的同时,也配合适当的范例让读者了解应用的方式,而这些范例也都是以在 DB2 V 8.1 的环境执行为例。本书并没有将所有指令与语法详尽列出,读者若要了解 DB2 各项功能的完整指令语法,仍需要查阅 DB2 的正式参考手册。

我一直都觉得关系型数据库是一种很“人性化”的系统软件,不论是结构、原理或操作,都是那么直接与合理。当然现在 DB2 的功能已经非常复杂而强大,但是仍然可以从简单的层级循序渐进地进入 DB2 的殿堂。所以导入学习的阶段是很重要的,而是否有一本适当的参考书籍,对于学习者会有很大的影响。本书的目的就在于此,希望借助本书,让你享受学习 DB2 的乐趣,让你也和我一样,觉得 DB2 功能这么强大,又这么易懂易上手,让你禁不住爱上 DB2。

作者简介:

庄济诚,台湾交通大学计算机工程系学士、美国南加州大学计算机工程硕士。曾任美国 IBM 公司 Santa Teresa Laboratory DB2 产品研发工程师、美国 Informix 公司全球支持中心资深工程师,现任台湾 IBM 公司软件事业处资深工程师。译有《DB2 ——新世纪的对象关连式数据库》和《DB2 Universal Database 完全指南》。

编者

2003 年 10 月

目 录

第 1 章 认识 DB2.....	1	2.2.7 扩充与缩小表格空间.....	41
1.1 什么是实例	1	2.2.8 表格空间中加入容器.....	41
1.1.1 多个实例并存	1	2.2.9 在表格空间中删除数据容器 ...	42
1.1.2 使用哪一个实例	2	2.2.10 不同大小的数据容器上	
1.1.3 建立实例	2	数据的分布	43
1.1.4 启动与停止实例	3	2.2.11 缩减数据容器的大小.....	44
1.1.5 实例的管理	3	2.2.12 BEGIN NEW STRIPE	
1.1.6 Administration 服务器	4	SET 选项	45
1.2 DB2 的存取界面	5	2.2.13 db2empfa 指令.....	46
1.2.1 交互式界面	5	2.2.14 表格空间与系统	
1.2.2 应用程序界面	9	效能的关系	46
1.2.3 管理者界面	13	第 3 章 数据库中的对象.....	47
1.3 建立 DB2 客户机/服务器	16	3.1 对象名称与模式名称.....	47
1.3.1 DB2 服务器的通信设置.....	16	3.2 表格	49
1.3.2 DB2 客户机的通信设置.....	17	3.2.1 数据类型	49
1.4 DB2 与系统环境参数	19	3.2.2 生成表格	50
1.4.1 系统环境参数	19	3.2.3 更改表格定义.....	50
1.4.2 DB2 注册配置文件.....	19	3.2.4 LIKE 子句	51
1.4.3 数据库设置参数	21	3.3 别名	51
1.4.4 实例(Database Manager)		3.4 限制	52
设置参数	22	3.4.1 NOT NULL 限制.....	52
1.5 常用的 DB2 指令	23	3.4.2 字段默认值	52
1.6 常用的 DB2 系统指令	28	3.4.3 惟一限制	53
第 2 章 数据库与表格空间	31	3.4.4 检查限制	53
2.1 什么是数据库	31	3.4.5 主键限制	54
2.2 表格空间	32	3.4.6 Foreign Key Constraint.....	54
2.2.1 数据容器	33	3.5 视图	57
2.2.2 表格空间、表格及		3.5.1 什么是视图	57
容器的关系	34	3.5.2 只读视图	59
2.2.3 表格空间的种类	35	3.5.3 查询视图	60
2.2.4 建立表格空间	35	3.5.4 用连接所有视图	
2.2.5 DMS 表格空间与 SMS		模拟范围分区.....	61
表格空间的比较	38	3.6 实例化的查询表(MQT)	62
2.2.6 列出表格空间	39	3.7 索引	62

3.8 触发器	63	5.6.1 事务日志文件的位置	104
3.8.1 什么是触发器	63	5.6.2 激活事务日志文件	105
3.8.2 建立与删除触发器	66	5.6.3 存档事务日志文件	105
3.8.3 前触发器	67	5.7 事务日志文件的结构	106
3.8.4 后触发器	69	5.7.1 循环日志	106
3.9 序号对象	74	5.7.2 线性日志	107
3.10 识别字段	75	5.8 数据库回存	108
3.11 数据加密	76	5.9 Rollforward database 命令	109
3.12 db2look 指令	78	5.10 事务日志文件的备份	112
第 4 章 数据的传输	80	5.10.1 手动备份事务日志文件	112
4.1 数据文件的格式	80	5.10.2 自动备份事务日志文件	112
4.2 输出功能	81	5.11 联机备份	113
4.3 输入功能	83	5.11.1 联机备份的回存	113
4.4 加载功能	85	5.12 有关备份与恢复的数据库参数	114
4.4.1 加载数据查询	89	5.13 Circular Logging 与 Linear Logging 的比较	116
4.4.2 索引模式	90	5.14 变动部分备份	116
4.4.3 检查挂起	91	5.14.1 为何需要变动部分备份	116
4.4.4 SET INTEGRITY 指令	91	5.14.2 Incremental Backup 的种类	117
4.4.5 联机加载数据	92	5.14.3 如何设定数据库支持 Incremental Backup	118
4.4.6 DUMPFIL 选项	92	5.14.4 Incremental Backup 的 例子	118
4.4.7 加载的全部过程	93	5.14.5 Cumulative Backup 的 回存	118
4.5 输入/加载数据与数据库的恢复	93	5.14.6 Delta Backup 的回存	119
4.5.1 加载执行失败或中断	94	5.15 救回删除掉的表格	120
4.5.2 数据加载完成后的 备份挂起	94	5.16 Turning off Logging	123
4.6 输入/加载数据指令的额外选项	96	5.16.1 Long Transaction 所造成的 事务日志爆满	123
第 5 章 备份与恢复	99	5.16.2 表格的 NOT LOGGED 属性	125
5.1 数据库毁坏的原因	99	5.16.3 NOT LOGGED 属性的 副作用	126
5.2 什么是事务?	99	5.17 回存到其他数据库	126
5.3 DB2 的恢复类型	99	5.18 表格空间的备份与恢复	127
5.3.1 Crash Recovery	100	5.18.1 表格空间的 Minimum Recovery Time	128
5.3.2 Restore/Version Recovery	100	5.18.2 表格之间的相关时间点	130
5.3.3 Rollforward Recovery	100		
5.4 数据库备份	100		
5.5 DB2 的缓冲池与日志缓冲	103		
5.5.1 缓冲池	103		
5.5.2 日志缓冲	104		
5.6 事务日志文件	104		

第 6 章 数据锁定	132	7.6.4 Index Only Access	163
6.1 锁定在何时发生	132	7.6.5 索引中包括其他字段	164
6.2 Lock Table 指令	133	7.6.6 WHERE 条件的选择性	164
6.3 DB2 锁定的对象	133	7.6.7 索引建议器	165
6.3.1 数据库锁定	133	7.6.8 丛集索引	167
6.3.2 表格空间锁定	134	7.6.9 索引结构的副作用	169
6.3.3 表格锁定	134	7.7 缓冲池	169
6.3.4 记录锁定	134	7.7.1 表格空间与缓冲池的关系	169
6.4 锁定的模式	134	7.7.2 缓冲池中的数据清除 (Page Cleaning)	171
6.5 锁定的长度	136	7.8 数据预取	171
6.6 锁定等待	138	7.8.1 预取与 SMP Parallelism	172
6.7 锁定超时	138	7.8.2 Sequential Detection	172
6.8 锁定升级	139	7.8.3 Prefetch Size	173
6.9 SQL 执行计划与锁定的关系	139	7.8.4 I/O Server	174
6.10 如何避免因 Relation Scan 造成的锁定等待	142	7.9 Extended Storage	174
6.11 不同的 Isolation Level 造成 不同的执行结果	143	7.10 数据重整(Reorganization)	175
6.11.1 Isolation Level CS 的 情况	144	7.10.1 REORGCHK	176
6.11.2 Isolation Level RR 的 情况	144	7.10.2 表格重整	179
6.11.3 Isolation Level RS 的 情况	145	7.10.3 索引重整	180
第 7 章 数据库系统效能	147	7.10.4 联机表格重整	180
7.1 了解 DB2 的最优化处理器	147	7.11 MQT	182
7.2 DB2 Explain 工具	149	7.12 Multi Dimensional Clustering	185
7.2.1 db2expln	149	7.12.1 MDC 的结构	185
7.2.2 dynexpln	149	7.12.2 三个维度的 MDC 表格	188
7.2.3 Visual Explain	151	7.12.3 如何建立 MDC 表格	189
7.3 系统表格中的统计信息	153	7.12.4 MDC 表格对查询的 帮助	190
7.4 RUNSTATS 指令	155	7.13 平行处理	191
7.5 最优化处理等级	156	7.13.1 设定平行处理	191
7.6 索引	158	7.13.2 平行处理的执行计划	192
7.6.1 索引的结构	158	7.13.3 选择平行处理等级	193
7.6.2 建立索引的方法	159	7.13.4 限制平行处理等级	193
7.6.3 索引与 SQL 查询的关系	160	第 8 章 安全管理	195
		8.1 身份鉴别	195
		8.2 权限控制	196
		8.2.1 权限	197
		8.2.2 特权	198

8.2.3 Authority 与 Privilege 的 授予及收回	199	9.3.2 UPDATE 指令	226
8.2.4 Public 组	201	9.3.3 Delete 指令	228
8.2.5 权限设定的 GUI 接口	202	9.4 特殊记录器	229
8.2.6 检查权限的时机	203	第 10 章 高级 SQL 指令	230
第 9 章 基础 SQL 指令	205	10.1 CASE 表达式	230
9.1 示例数据库	205	10.1.1 CASE 表达式的 简单形式	230
9.2 查询数据	207	10.1.2 CASE 表达式的 一般形式	231
9.2.1 表达式	208	10.2 子查询	234
9.2.2 Predicate	209	10.3 数量子查询	237
9.2.3 搜索条件	212	10.4 表格表达式	238
9.2.4 格式连接	212	10.5 表格函数	240
9.2.5 字段函数	213	10.6 Explicit Join	242
9.2.6 分组	214	10.6.1 内部链接	243
9.2.7 Subselect	215	10.6.2 Outer Join	244
9.2.8 Fullselect 与 Literal Table	217	10.7 公用表格表达式	245
9.2.9 SELECT 指令	221	10.8 递归 SQL	249
9.2.10 VALUES 指令	223	10.9 Recursion with computation	251
9.3 更改数据	224	10.10 递归式查询	254
9.3.1 INSERT 指令	224		

第 1 章 认识 DB2

本章的目的是要让读者了解 DB2 的环境,包括 DB2 的实例、交互式界面、应用程序界面、管理者界面与工具、DB2 的客户机/服务器设置及 DB2 的环境参数。本章最后也列出了一些常用的 DB2 指令及 DB2 系统指令供读者参考。

1.1 什么是实例

实例是一个独立的数据库系统环境。一个实例可以拥有多个数据库,在一个实例中可以使用的数据库可以是它本地拥有的数据库,也可以是通过 UDB 的客户机/服务器设置方法所连接的其他实例(不论是同一部服务器或是其他服务器上的实例)中的数据库。

实例是通过 db2start 指令启动,在实例启动后,应用程序或交互式界面的用户才可以连接到数据库,然后对 DB2 下达 SQL 指令进行数据处理与查询,执行 DB2 指令,或执行 DB2 的系统指令。

1.1.1 多个实例并存

在一台服务器上只需要安装一次 DB2,然后就可以建立一个或多个实例。在一台服务器上建立多个实例,可以有下列的用途:

- 一个实例用在 Production Instance,另一个当作测试开发的 Testing Instance。
- 用分开的实例执行不同的应用系统,针对不同应用系统的特性,分别调整实例的参数。
- 将不同机密程度的应用系统与数据分在不同的实例上执行,可以用作适度的安全区分。
- 将几个应用系统分在不同的实例上执行,在某个实例无法运行时,可以减少对应用系统的影响程度。

当然,在一台服务器上启动多个实例也有一些负面影响:

- 每一个实例都会消耗系统资源,实例越多,就消耗越多的系统资源。
- DBA 管理工作量增加。

图 1.1 是在一部服务器上建立两个实例的示意图。在一部服务器上只要安装一次 DB2,但是可以建立多个实例。图中一个实例叫做 INSTANCE_1,另一个叫做 INSTANCE_2,它们各有实例的设置参数(Database Manager Configuration,简称 DBM CFG),可以用以设置或改变这个实例的行为、容量或性能。INSTANCE_1 中有两个数据库(DB_1 与 DB_2),INSTANCE_2 中也有两个数据库(DB_3 与 DB_4)。每个数据库各有自己的一组设置参数(Database Configuration,简称 DB CFG),可以改变数据库的行为、容量和性能。每个数据库也有自己的一组系统表(System Tables,或称为 System Catalog Tables 或 Catalog Tables),用来记录该数据库中的所有对象(如 Tablespace、Table、Index、Trigger、存储过程等)及权

175562/04

限等。

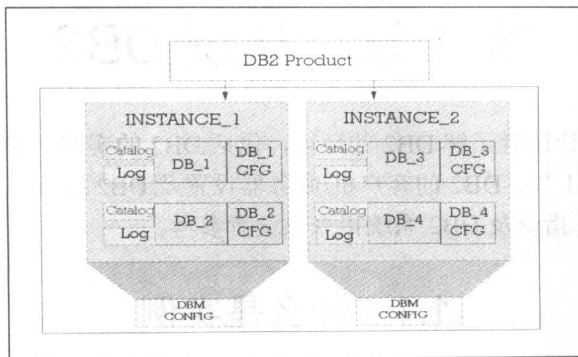


图 1.1

每个数据库也都各自拥有一组事务日志(Transaction Logs)，用来记录这数据库中所有的情况，以便在数据库受损时执行数据库恢复(Recovery)的操作。

1.1.2 使用哪一个实例

如果一部计算机上有多个实例，当你登录到这台计算机后，你用到的是哪一个实例？这决定于 DB2INSTANCE 这个环境参数。图 1.2 中，一部计算机上建立了两个实例，一个名为 inst1，一个名为 inst2。当用户登录这台计算机后，如果将环境参数 DB2INSTANCE 设置为 inst1，他所使用的就是 inst1 这个实例，他就可以看到 database1 这个数据库。

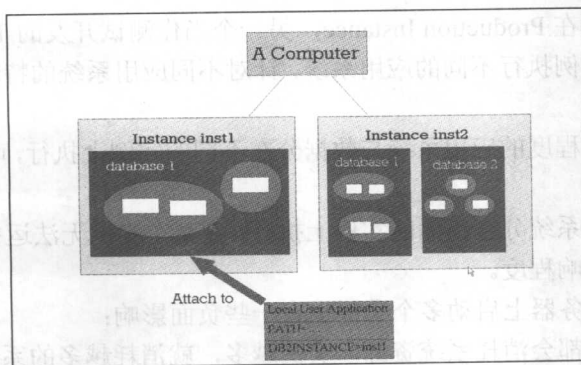


图 1.2

1.1.3 建立实例

在 Windows 环境中，DB2 光碟上的 setup.exe 除了会安装 DB2 之外，也会自动建立一个 DB2 实例，这个实例默认的名字是 DB2。而在 UNIX 环境中，DB2 光碟上的 db2setup 除了安装 DB2 之外，也会建立一个 DB2 实例，这个实例默认的名字是 db2inst1。

除了在安装 DB2 时所自动建立的实例之外，还可以另外建立实例。在 Windows 环境 DB2 的安装目录 sqllib\bin 中，或是 Unix 环境 DB2 的安装目录 /usr/opt/db2_08_01/instance

中,有一个名为 `db2icrt` 的执行文件,可以用来建立实例。例如,在 Windows 上:

```
db2icrt production
```

这个指令会建立一个名为 **Production** 的 DB2 实例。

下面的指令会在 UNIX 上建立一个名为 `db2inst2` 的 DB2 实例:

```
db2icrt -w 32 -s WSE -p 50000 -u db2fenc2 db2inst2
```

这个指令中用到了几个选项:

- **-w**: 这个选项用来指定所建立的是 32 位实例还是 64 位的实例。
- **-s**: 可以指定 WSE(Workgroup Server Edition)或 ESE(Enterprise Server Edition)。即使你所安装的 DB2 是 ESE 版本,也可以选择建立 WSE 的实例。
- **-p**: 指定这个实例所用的端口号,在 DB2 客户机连接时使用。
- **-u**: 用来指定 DB2 执行 Fenced User Defined Function 或是 Fenced 存储过程所用的用户账号。

1.1.4 启动与停止实例

如果是用 `db2setup`(UNIX)或是 `setup.exe`(Windows)在安装 DB2 时所建立的实例,安装程序已经将实例设置成在服务器开机时自动启动。也可以用 `db2start` 来启动实例,或是用 `db2stop` 来停止实例。

例如,

```
db2start
```

这个指令可以启动 DB2 实例。

```
db2stop
```

这个指令可以停止 DB2 实例。但是如果还有用户,这个指令将无法成功。

```
db2stop force
```

这个指令可以强迫所有的用户离开,再将 DB2 实例停止。

1.1.5 实例的管理

可以用 `db2ilist` 指令(DB2 Windows 安装目录 `sqllib\bin` 中,或 UNIX 安装目录 `/usr/opt/db2_08_01/instance` 中),将一部计算机上所建立的实例列出来,如图 1.3 所示。



图 1.3

可以用 `db2idrop` 指令将实例删除。当然要确定这个实例中的数据库是可以丢弃，或是已将数据作备份后移到他处使用；也要确定你已经用 `db2stop` 将实例停止，才可以将实例删除掉。例如，

```
db2idrop db2inst2
```

1.1.6 Administration 服务器

以上所讨论的实例，都是用来建立数据库，及开发与执行应用系统的。在安装 DB2 时，安装程序还会自动建立另一个特殊的实例，称为 DB2 Administration 服务器(简称为 DAS)。DAS 是用来支持 DB2 的一些工具，如 Configuration Assistant、控制中心、任务中心及 Development Center 所设计的。DAS 的功用包括：

- 让管理者可以从 Administration 客户机上执行其他计算机上的 DB2 实例及数据库的管理工作。
- 可以定义、排程及执行 DB2 或是操作系统的指令与工作。
- 可以协助设置 DB2 数据库的客户机/服务器结构。

如图 1.4 所示，在一部服务器上建立了两个实例(Instance1 与 Instance2)，Instance1 上有 A 与 B 两个数据库，Instance2 上有 C 与 D 两个数据库。在一个客户机上安装了 DB2 Administration 客户端。服务器上启动了 DAS，我们就可以在客户机上，通过 DB2 Administration 客户端所提供的 Configuration Assistant，自动寻找并连接到 A、B、C 与 D 数据库，完成客户机/服务器的设置。也可以通过 DB2 Administration 客户端的控制中心，执行所有对 A、B、C 与 D 数据库的管理工作。也可以通过 DB2 Administration 客户端的任务中心，定义、排程与执行 A、B、C 或 D 数据库的应用程序或指令，或是 Server Machine 上的操作系统指令(OS Commands)。

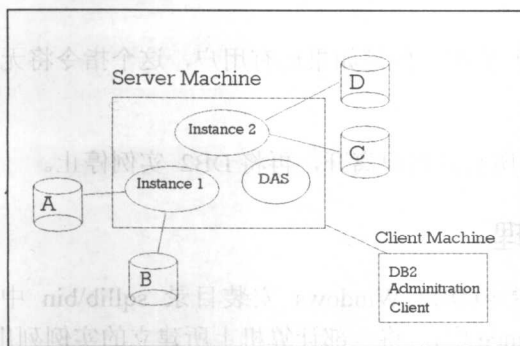


图 1.4

一部计算机上只可以有一个 DAS，DB2 的安装程序将 DAS 设置为在开机时可以自动启动。也可以用下面的指令启动 DAS：

```
db2admin start
```

或是用下面的指令来停止 DAS：

```
db2admin stop
```

1.2 DB2 的存取界面

在 DB2 环境中，用户与应用程序可以通过不同的方式和界面与 DB2 沟通。图 1.5 是存取 DB2 的方式。

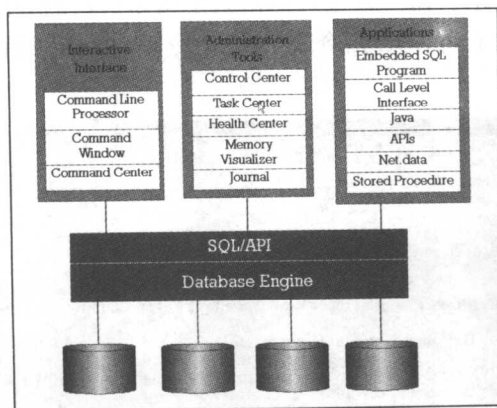


图 1.5

DB2 存取的界面主要可以分为 3 类：

- 交互式界面：包括命令行处理器、指令中心以及指令窗口。
- 管理者界面：包括控制中心、任务中心、Health Center、Memory Visualizer 以及日志等。
- 应用程序界面：包括嵌入式 SQL 程序、Call Level Interface、Java、Net.data 以及 API。

1.2.1 交互式界面

DB2 提供 3 种交互式界面，包括命令行处理器、指令窗口及指令中心。

1.2.1.1 命令行处理器

这是一个文字模式的界面，可以让你在此界面对 DB2 下达 SQL 指令以及 DB2 指令，如图 1.6 所示。

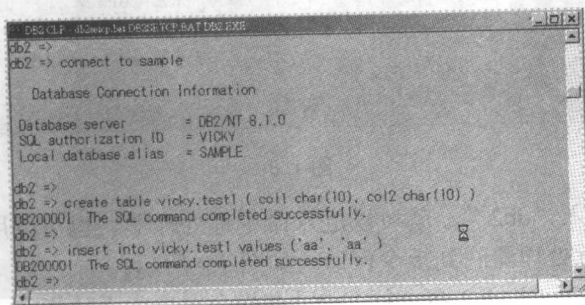


图 1.6

进入 DB2 命令行处理器时会得到一个提示符号“db2 =>”。在提示符号后，可以直接输入 SQL 指令(如 CREATE TABLE 或是 INSERT 等)或 DB2 指令(如 BACKUP DATABASE 或是 LOAD 等)。

1.2.1.2 指令窗口

指令窗口其实就是 DB2 命令行处理器，只是指令窗口的界面并没有提示符号“db2 =>”。所以在指令窗口中下达 SQL 指令或 DB2 指令时，都需在指令前带上 db2，如图 1.7 所示。

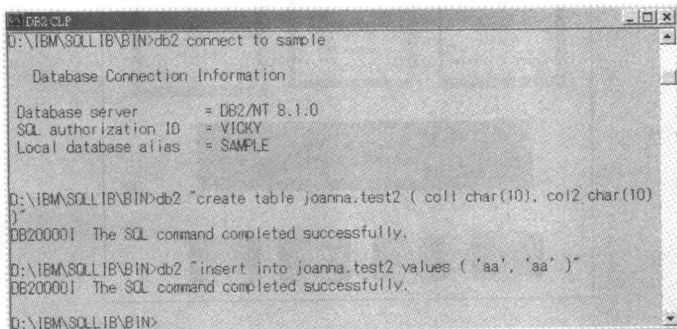


图 1.7

在指令窗口中下达指令，通常的习惯是需要指令的两端都带上双引号(“ ”)。在 Windows 的环境并不需要如此，但是在 UNIX 环境，这就是必须用法，否则在指令中的非文字符号(如*等)都会被 UNIX shell 先截走，造成所下达的指令无法执行。

指令窗口与命令行处理器是可以交互转换的。如图 1.8 所示。

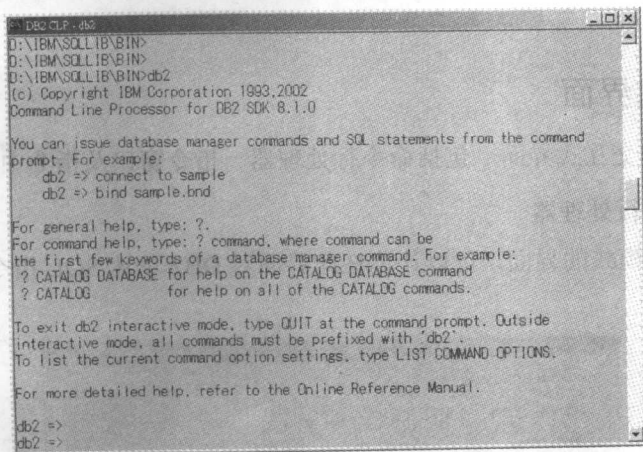


图 1.8

从指令窗口中输入“db2”，就可以进入命令行处理器。也可以用 TERMINATE 或 QUIT 指令离开命令行处理器回到指令窗口。如图 1.9 所示。

不论是从指令窗口输入 db2 命令行，或是在指令窗口中下达指令，都可以加上一些选项，以改变 DB2 执行指令时的行为。图 1.10 是 DB2 命令行处理器的选项。

图 1.9

Option	Description	Default Setting
-a	Display SQLCA	OFF
-c	Auto-commit	ON
-e	Display SQLCODE/SQLSTATE	OFF
-f	Read from input file	OFF
-l	Log commands in history file	OFF
-n	Remove new line character	OFF
-o	Display output	ON
-p	Display db2 interactive prompt	ON
-r	Save output report to file	OFF
-s	Stop execution on command error	OFF
-t	Set statement termination character	OFF
-v	Echo current command	OFF
-w	Display FETCH/SELECT warning messages	ON
-x	Suppress printing of column headings	OFF
-z	Save all output to output file	OFF

图 1.10

下面的例子，是从指令窗口中建立一个表格，但是不要自动确认：

```
db2 +c "CREATE TABLE vicky.test3 ( col1 CHAR(10), col2 CHAR(10) )"
```

注意： 一般从指令窗口与命令行处理器所下达的指令，DB2 都会自动在该指令完成后自动确认(Commit)，这可以从图 1.10 中 -c 选项的 Default Setting 是 ON 看出。如果不需要 DB2 有这样的行为，可以在指令前带上 +c 选项，就可以让 DB2 不自动确认该指令。你必须自己下达 COMMIT 指令。

另外一个常用这些选项的方式，是当有许多指令需要执行，但又不想从指令窗口下达这些指令时，可以先将这些指令存放在一个文件中。图 1.11 是 test1.sql 文件内容。

```
-- This is a testing DB2 Command Window

CONNECT TO sample;

CREATE TABLE joanna.table1 ( col1 CHAR(10), col2 CHAR(20) );

INSERT INTO joanna.table1 VALUES ( 'AA', 'AA' );

INSERT INTO joanna.table1 VALUES ( 'AA', 'AA' );

SELECT *FROM joanna.table1;
```

图 1.11

将文件准备好后,就可以从指令窗口中用下面的指令执行:

```
db2 -svtf test1.sql
```

这个指令中用了几个选项,从上面的选项说明可以看出它们的作用:

- -s: 任一个指令发生错误,后面的指令就不再继续执行下去。
- -v: 将所执行的指令显示在屏幕上。
- -t: 采用默认的命令结束符号(;).
- -f: 从其后所指定的文件(test1.sql)读入指令。

在 test1.sql 文件中,也可以在一行前面用 “—” 加入批注。

另外,在指令窗口或命令行处理器也可以查询 SQL 错误代码和 DB2 指令语法。例如从指令窗口中下达下面的指令,就可以查知 SQL0805N 错误码的意义、原因、错误码的意义、原因及解决方法:

```
db2 ? SQL0805N
```

下面的指令会显示出创建数据库的语法:

```
db2 ? CREATE DATABASE
```

1.2.1.3 指令中心

执行 SQL 指令、DB2 指令、或前面所述 test1.sql 指令文件的另一种方法,是使用 DB2 的指令中心。指令中心是一种图形化的界面。图 1.12 是 DB2 指令中心的外观。在 Interactive Panel 面板可以交互式地输入 DB2 指令或的 SQL 语句指令。在 Script Panel 面板可以选取并执行预先写好的指令文件(如前面的 test1.sql)。如果所执行的 SQL 语句是一个 SELECT 查询指令,其结果会显示在 Query Results 面板上。

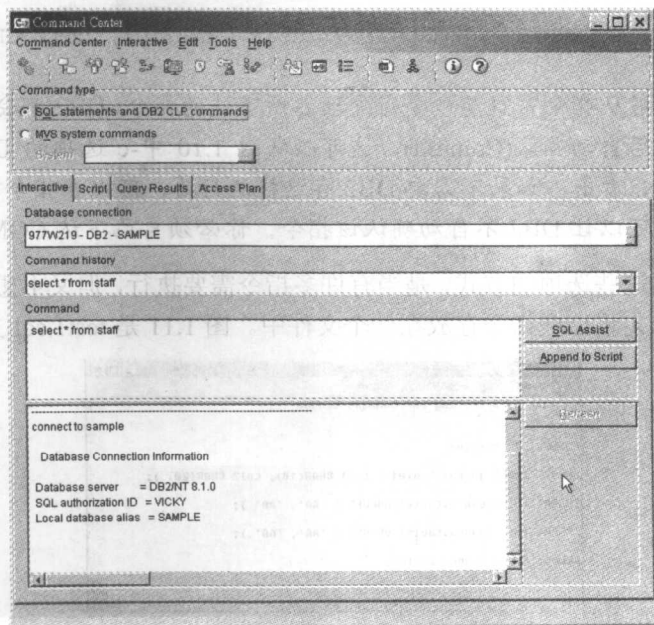


图 1.12

在 DB2 的交互式界面，可以执行下面几种工作：

- 执行 SQL 指令，如 CREATE TABLE、INSERT 或 SELECT 等。在本书的第 9 章和第 10 章会对 DB2 的 SQL 指令有进一步的讨论。
- 执行 DB2 指令，如 BACKUP DATABASE、LIST DB DIRECTORY、LOAD 等。本章的最后会介绍一些用户经常要用的 DB2 指令。
- 在 DB2 指令窗口中，可以执行 DB2 的系统指令(DB2 System Command)，如 db2move、db2look、db2start、db2stop 等。这些指令都只能从 DB2 指令窗口在操作系统提示符号后下达，所以称为 DB2 系统指令。本章最后也会介绍一些经常要用的 DB2 系统指令。

1.2.2 应用程序界面

应用程序开发者可以利用下面几种方式开发 DB2 的应用程序。

1.2.2.1 嵌入式 SQL 程序

所谓嵌入式 SQL 程序，是用 C、C++、Fortran、Cobol 和 Java 等语言编写的程序，而在程序中嵌入(Embed)SQL 指令。当你编写一段嵌入式的 SQL 程序后，需要执行一些特定的步骤，才可以让这个程序成为可执行文件。这些步骤如图 1.13 所示。

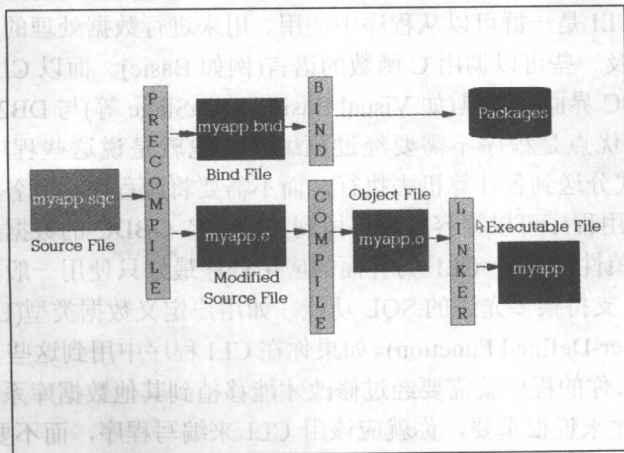


图 1.13

这个例子假设是用 C 语言编写的嵌入式 SQL 程序，源程序文件的名称是 myapp.sqc：

- (1) 预编译：这个步骤是将源程序文件 myapp.sqc 中的 SQL 指令转换成 DB2 进行最优化处理(Optimization)所需的格式，放置到一种称为绑定文件(Bind file)的文件中，文件的默认扩展名为.bnd(myapp.bnd)。而原本 myapp.sqc 中 SQL 指令的部分，则换成 DB2 调用函数，称为 Modified Source File。所以预编译的结果会产生两个文件，一个是绑定文件，另一个则是 Modified Source File。下面是执行预编译指令的例子：

```
PREP myapp.sqc BINDFILE
```