

IBM® DB2® 通用数据库



# 管理指南：计划

版本 7



IBM® DB2® 通用数据库



# 管理指南：计划

版本 7

在使用本资料及其支持的产品之前，请阅读第257页的『附录F. 声明』中的一般信息。

本文档包含 IBM 的专利信息。它在许可证协议下提供，并受版权法保护。本出版物包含的信息不包括任何产品保证，且本手册提供的任何声明不应作如此解释。

通过您当地的 IBM 代表或 IBM 分部可订购出版物，或者，通过致电 1-800-879-2755（在美国）或 1-800-IBM-4YOU（在加拿大）来订购出版物。

当您发送信息给 IBM 后，即授予 IBM 非专有权，IBM 对于您所提供的任何信息，有权利以任何它认为适当的方式使用或分发，而不必对您负任何责任。

© Copyright International Business Machines Corporation 1993, 2001. All rights reserved.

# 目录

关于本书 . . . . .	vii	节点组和数据分区 . . . . .	31
谁应使用本书 . . . . .	viii	并行性的类型 . . . . .	33
本书的结构 . . . . .	viii	I/O 并行性 . . . . .	33
“管理指南”的其他卷的简要概述 . . . . .	ix	查询并行性 . . . . .	33
管理指南: 实现 . . . . .	ix	实用程序并行性 . . . . .	36
管理指南: 性能 . . . . .	x	硬件环境 . . . . .	36
		单处理器上的单分区 . . . . .	37
<b>第1部分 “DB2 通用数据库” 的世界 1</b>		具有多个处理器的单分区 . . . . .	38
<b>第1章 管理 DB2 通用数据库 . . . . .</b>	<b>3</b>	多分区配置 . . . . .	39
		最适合每种硬件环境的并行性的摘要 . . . . .	43
<b>第2部分 数据库概念 . . . . .</b>	<b>5</b>	<b>第5章 关于数据入库 . . . . .</b>	<b>45</b>
<b>第2章 基本关系数据库概念 . . . . .</b>	<b>7</b>	什么是数据入库? . . . . .	45
数据库对象概述 . . . . .	7	主题区 . . . . .	45
实例 . . . . .	8	仓库源 . . . . .	46
数据库 . . . . .	9	仓库目标 . . . . .	46
节点组 . . . . .	9	仓库代理程序和代理站点 . . . . .	46
表 . . . . .	9	步骤和过程 . . . . .	46
视图 . . . . .	9	入库任务 . . . . .	48
索引 . . . . .	10	<b>第6章 关于 Spatial Extender . . . . .</b>	<b>51</b>
模式 . . . . .	11	Spatial Extender 的用途 . . . . .	51
系统目录表 . . . . .	11	表示地理面貌的数据 . . . . .	51
存储器对象概述 . . . . .	12	数据如何表示地理面貌 . . . . .	52
表空间 . . . . .	12	空间数据的性质 . . . . .	53
容器 . . . . .	15	空间数据从哪里来 . . . . .	53
缓冲池 . . . . .	16	<b>第3部分 数据库设计 . . . . .</b>	<b>57</b>
系统对象概述 . . . . .	17	<b>第7章 逻辑数据库设计 . . . . .</b>	<b>59</b>
配置参数 . . . . .	17	决定要记录在数据库中的数据 . . . . .	59
数据的商业规则 . . . . .	19	为每种类型的关系定义表 . . . . .	61
恢复数据库 . . . . .	22	一对多和多对一关系 . . . . .	61
重组数据库中的表 . . . . .	23	多对多关系 . . . . .	62
DB2 安全性概述 . . . . .	23	一对一关系 . . . . .	63
认证 . . . . .	23	为所有表提供列定义 . . . . .	63
授权 . . . . .	24	标识一列或多列作为主键 . . . . .	65
联合数据库认证和授权概述 . . . . .	25	标识入选键列 . . . . .	66
<b>第3章 联合系统 . . . . .</b>	<b>27</b>	定义标识列 . . . . .	67
启用联合系统 . . . . .	29	确保相等的值表示相同的实体 . . . . .	68
<b>第4章 并行数据库系统 . . . . .</b>	<b>31</b>	考虑使表规范化 . . . . .	68

第一种范式 . . . . .	69	应用程序 (AP) . . . . .	134
第二种范式 . . . . .	69	事务管理器 (TM) . . . . .	135
第三种范式 . . . . .	71	资源管理器 (RM) . . . . .	136
第四种范式 . . . . .	72	将数据库设置为资源管理器 . . . . .	137
制定约束实施计划 . . . . .	73	xa_open 和 xa_close 字符串的用法 . . . . .	137
唯一约束 . . . . .	73	DB2 版本 7 的新 xa_open 字符串格式 . . . . .	137
引用完整性 . . . . .	74	TPM 和 TP_MON_NAME 值 . . . . .	139
表检查约束 . . . . .	78	较早版本的 DB2 的 xa_open 字符串格式 . . . . .	142
触发器 . . . . .	78	更新主机或 AS/400 数据库服务器 . . . . .	142
数据库设计的其他注意事项 . . . . .	78	数据库连接注意事项 . . . . .	142
<b>第8章 物理数据库设计 . . . . .</b>	<b>81</b>	作试探性决定 . . . . .	143
数据库目录 . . . . .	81	安全性注意事项 . . . . .	145
数据库文件 . . . . .	82	配置注意事项 . . . . .	146
估计表的空间需求 . . . . .	83	受支持的 XA 函数 . . . . .	147
系统目录表 . . . . .	84	XA 接口的问题确定 . . . . .	149
用户表数据 . . . . .	84	配置 XA 事务管理器以使用 DB2 UDB . . . . .	150
长整数字段数据 . . . . .	86	配置 IBM TXSeries CICS . . . . .	150
大对象 (LOB) 数据 . . . . .	86	配置 IBM TXSeries Encina . . . . .	150
索引空间 . . . . .	87	配置 BEA Tuxedo . . . . .	152
附加空间需求 . . . . .	90	配置 Microsoft 事务处理服务器 . . . . .	154
日志文件空间 . . . . .	90	<b>第11章 介绍高可用性和故障恢复支持 . . . . .</b>	<b>161</b>
临时工作空间 . . . . .	91	高可用性 . . . . .	161
设计节点组 . . . . .	91	基于联机分割镜像和暂挂 I/O 支持的高可用性 . . . . .	163
节点组设计注意事项 . . . . .	92	生成克隆数据库 . . . . .	164
设计和选择表空间 . . . . .	97	将分割镜像用作备用数据库 . . . . .	164
系统管理空间 . . . . .	100	将分割镜像用作备份映像 . . . . .	165
数据库管理空间的表空间 . . . . .	104	<hr/>	
表空间的设计注意事项 . . . . .	106	<b>第4部分 附录 . . . . .</b>	<b>167</b>
联合数据库设计注意事项 . . . . .	115	<b>附录A. 命名规则 . . . . .</b>	<b>169</b>
<b>第9章 设计分布式数据库 . . . . .</b>	<b>117</b>	一般命名规则 . . . . .	169
在一个事务中使用单个数据库 . . . . .	117	对象命名规则 . . . . .	169
在单个事务中使用多个数据库 . . . . .	118	关于模式名的附加信息 . . . . .	170
更新单个数据库 . . . . .	118	关于密码的附加信息 . . . . .	172
更新多个数据库 . . . . .	120	在对象名中使用定界标识符 . . . . .	172
其他配置注意事项 . . . . .	123	如何在联合系统中保留区分大小写的值 . . . . .	172
在多站点更新中访问基于 LAN 的 DB2 通 用数据库服务器的主机或 AS/400 应用程序 . . . . .	125	<b>附录B. 数据库迁移计划 . . . . .</b>	<b>175</b>
了解两阶段提交过程 . . . . .	126	迁移注意事项 . . . . .	175
在两阶段提交期间校正问题 . . . . .	129	迁移限制 . . . . .	175
若 AUTORESTART=OFF, 则再同步不确 定事务 . . . . .	130	安全性与授权 . . . . .	176
<b>第10章 针对事务管理器的设计 . . . . .</b>	<b>133</b>	存储器需求 . . . . .	176
X/Open 分布式事务处理模型 . . . . .	134	发行版间的不兼容性 . . . . .	176
		迁移数据库 . . . . .	177

<b>附录C. 发行版间的不兼容性</b> . . . . .	<b>179</b>	字符集 . . . . .	220
“DB2 通用数据库”计划的不兼容性 . . . . .	180	标识符的字符集 . . . . .	220
“DB2 通用数据库”未来版本中的只读视图 . . . . .	180	编码 SQL 语句 . . . . .	221
“DB2 通用数据库”未来版本中的 PK_COLNAMES 和 FK_COLNAMES . . . . .	180	双向 CCSID 支持 . . . . .	221
在“DB2 通用数据库”未来版本中 COLNAMES 不再可用 . . . . .	180	整理顺序 . . . . .	225
“DB2 通用数据库”版本 7 的不兼容性 . . . . .	181	日期时间值 . . . . .	226
应用程序编程 . . . . .	181	DB2 UDB 中的 Unicode 支持 . . . . .	232
SQL . . . . .	183	介绍 . . . . .	232
实用程序和工具 . . . . .	184	DB2 UDB 中的 Unicode 实现 . . . . .	234
连通性和共存 . . . . .	185	<b>附录E. 使用 DB2 资料库</b> . . . . .	<b>241</b>
“DB2 通用数据库版本 6”的不兼容性 . . . . .	185	DB2 PDF 文件和打印的书籍 . . . . .	241
系统目录视图 . . . . .	185	DB2 信息 . . . . .	241
应用程序编程 . . . . .	191	打印 PDF 书籍 . . . . .	249
SQL . . . . .	195	订购打印书籍 . . . . .	250
数据库安全性和协调 . . . . .	196	DB2 联机文档 . . . . .	251
实用程序和工具 . . . . .	197	访问联机帮助 . . . . .	251
连通性和共存 . . . . .	198	查看联机信息 . . . . .	252
配置参数 . . . . .	198	使用 DB2 向导 . . . . .	254
<b>附录D. 本地语言支持 (NLS)</b> . . . . .	<b>199</b>	设置文档服务器 . . . . .	255
本地语言版本 . . . . .	199	搜索联机信息 . . . . .	256
国家 / 地区代码和代码页支持 . . . . .	199	<b>附录F. 声明</b> . . . . .	<b>257</b>
DB2 管理服务器的语言环境设置 . . . . .	215	商标 . . . . .	259
基于 UNIX 的平台所支持的翻译语言环境 . . . . .	215	<b>索引</b> . . . . .	<b>261</b>
控制中心和文档文件集 . . . . .	218	<b>与 IBM 联系</b> . . . . .	<b>267</b>
派生代码页值 . . . . .	219	产品信息 . . . . .	267





---

## 关于本书

本管理指南在它的三卷中提供了必要的参考信息，以便使用和管理 2000 年就绪的 DB2\* 关系数据库管理系统 (RDBMS) 产品，包括：

- 关于数据库设计的信息（可在《管理指南：计划》中找到）
- 关于实现和管理数据库的信息（可在《管理指南：实现》中找到）
- 关于配置和调整数据库环境以改进性能的信息（可在《管理指南：性能》中找到）。

本书中描述的许多任务可以使用不同的接口来执行：

- **命令行处理器**，它允许您从图形界面访问和操纵数据库。从此接口，您也可执行 SQL 语句和 DB2 实用程序功能。本书中的大多数示例说明的是此接口的使用。有关使用命令行处理器的更多信息，参见 *Command Reference*。
- **应用程序编程接口**，它允许您在应用程序内执行 DB2 实用程序功能。有关使用应用程序编程接口的更多信息，参见 *Administrative API Reference* 手册。
- **控制中心**，它允许您以图形方式执行管理任务，如配置系统、管理目录、备份和恢复系统、调度作业以及管理媒体。控制中心也包含复制管理，以便以图形方式设置系统之间数据的复制。此外，“控制中心”使您能通过图形用户界面执行 DB2 实用程序功能。根据平台的不同，调用“控制中心”的方法也不同。例如，在命令行上使用 db2cc 命令（在 OS/2 上）、从 DB2 文件夹选择控制中心图标，或在 Windows 平台上使用开始屏面。要获得介绍性帮助，从控制中心窗口的帮助下拉菜单中选择入门。**Visual Explain** 和**性能监视器**工具是从控制中心中调用的。

还可使用其他工具来执行管理任务。它们包括：

- **脚本中心**，用于存储称为脚本的小应用程序。这些脚本可包含 SQL 语句、DB2 命令以及操作系统命令。
- **警告中心**，用于监视其他 DB2 操作产生的信息。
- **工具设置**，用于更改“控制中心”、“警告中心”和“复制”的设置。
- **日志**，用于调度要以无人照管方式运行的作业。
- **数据仓库中心**，用于管理仓库对象。

---

## 谁应使用本书

本书主要面向数据库管理员、系统管理员、安全性管理员和系统操作员，他们需要设计、实施和维护本地或远程客户机要访问的数据库。需要了解 DB2 关系数据库管理系统的管理和操作的程序员和其他用户也可使用本书。

---

## 本书的结构

本书包含关于下列主要主题的信息：

### DB2 通用数据库的世界

- 第1章 管理 DB2 通用数据库，提供了“DB2 通用数据库”的介绍和概述。

### 数据库概念

- 第2章 基本关系数据库概念，提供了数据库对象（包括存储器对象和系统对象）的概述。
- 第3章 联合系统，讨论联合系统，联合系统是一个数据库管理系统 (DBMS)，它支持提交特定 SQL 语句的应用程序和用户，这些 SQL 语句在单条语句中引用两个或多个 DBMS 或数据库。
- 第4章 并行数据库系统，提供有关 DB2 可使用的并行性类型的介绍。
- 第5章 关于数据入库，提供数据入库和数据入库任务的概述。
- 第6章 关于 Spatial Extender，通过说明 Spatial Extender 的用途以及讨论它所处理的数据对 Spatial Extender 作了介绍。

### 数据库设计

- 第7章 逻辑数据库设计，讨论逻辑数据库设计的概念和准则。
- 第8章 物理数据库设计，讨论物理数据库设计的准则，包括与数据存储相关的注意事项。
- 第9章 设计分布式数据库，讨论如何在单个事务中访问多个数据库。
- 第10章 针对事务管理器的设计，讨论如何在分布式事务处理环境（如 CICS）中使用数据库。
- 第11章 介绍高可用性和故障恢复支持，提供 DB2 提供的高可用性故障恢复支持的概述。

### 附录

- 附录A. 命名规则，介绍命名数据库和对象时需遵循的规则。
- 附录B. 数据库迁移计划，提供关于将数据库迁移至版本 7 的信息。

- 附录C. 发行版间的不兼容性, 介绍不同发行版(最多到版本 7)之间的不兼容性。
- 附录D. 本地语言支持 (NLS), 介绍“DB2 本地语言支持”, 包括关于国家或地区、语言和代码页的信息。
- 附录E. 使用 DB2 资料库, 提供关于 DB2 库的结构资料, 包括向导、联机帮助、信息和书籍。

---

## “管理指南”的其他卷的简要概述

### 管理指南: 实现

《管理指南: 实现》主要讨论数据库设计的实现。下面简要描述该卷中的特定章节和附录:

#### 使用“控制中心”进行管理

- “使用 GUI 工具管理 DB2”描述用来管理数据库的图形用户界面 (GUI) 工具。

#### 实现设计

- “在创建数据库之前”描述创建数据库之前的前提条件。
- “创建数据库”描述那些与数据库及相关数据库对象的创建相关联的任务。
- “改变数据库”讨论在改变数据库之前必须执行的工作, 并讨论与修改和删除数据库或相关数据库对象相关联的任务。

#### 数据库安全性

- “控制数据库访问”描述如何控制对数据库资源的访问。
- “审核 DB2 活动”描述如何检测和监视对数据的不需要的访问或不期望的访问。

#### 移动数据

- “移动数据实用程序”是一个只有一页纸的介绍, 它说明移动数据的多种方法并指导您参考 *Data Movement Utilities Guide and Reference* 一书。

#### 恢复

- “恢复数据库”是对数据库备份、复原和前滚概念的简介。更广泛的信息可以在 *Data Recovery and High Availability Guide and Reference* 中找到。

#### 附录

- “使用分布式计算环境 (DCE) 目录服务”讨论有关如何使用“DCE 目录服务”的信息。

- “用于数据库恢复的用户出口” 讨论如何结合数据库日志文件来使用用户出口程序，并描述一些样本用户出口程序。
- “向多数据库分区服务器发出命令” 讨论如何使用 *db2\_all* 和 *rah* 外壳脚本来将命令发送至一个分区数据库环境中的所有分区。
- “DB2 Windows NT 版如何使用 Windows NT 的安全性” 描述 DB2 如何使用 Windows NT 安全性。
- “使用 Windows NT 性能监视器” 提供有关向“Windows NT 性能监视器”注册 DB2 以及使用性能信息的信息。
- “使用 Windows NT 或 Windows 2000 数据库分区服务器” 提供有关可用来使用 Windows NT 或 Windows 2000 上的数据库分区服务器的实用程序。
- “配置多个逻辑节点” 描述如何在分区数据库环境中配置多个逻辑节点。
- “高速节点间通信” 描述如何启用“虚拟接口体系结构”以用于“DB2 通用数据库”。
- “轻量级目录访问协议 (LDAP) 目录服务” 提供关于如何使用“LDAP 目录服务”的信息。
- “扩展控制中心” 提供有关如何通过添加包含新操作的新工具栏按钮、添加新对象定义及添加新操作定义来扩展控制中心的信息。

## 管理指南: 性能

《管理指南: 性能》主要讨论性能问题，即，那些主题和论点讨论建立、测试和改进应用程序及“DB2 通用数据库”产品本身的性能。下面简要描述该卷中的特定章节和附录：

### 性能介绍

- “性能元素” 介绍有关管理和改进 DB2 UDB 性能的概念和注意事项。
- “体系结构和处理概述” 介绍下层“DB2 通用数据库”体系结构和处理。

### 调整应用程序性能

- “应用程序注意事项” 描述在设计应用程序时用于改善数据库性能的一些技术。
- “环境注意事项” 描述在设置数据库环境时用于改善数据库性能的一些技术。
- “系统目录统计信息” 描述如何收集并使用数据统计信息以确保最优性能。
- “了解 SQL 编译程序” 描述使用 SQL 编译程序编译 SQL 语句的过程。
- “SQL 解释设施” 描述“解释”设施，它允许您检查 SQL 编译程序为访问数据所做的选择。

### 调整和配置系统

- “运行性能”概述数据库管理器如何使用内存以及影响运行时性能的其他注意事项。
- “使用控制器”介绍如何使用控制器来控制数据库管理的某些方面。
- “调整配置”描述与增加数据库系统的大小相关的一些注意事项和任务。
- “将数据重新分布到各分区中”讨论在分区数据库环境中将数据重新分布在各分区中所需的任务。
- “基准测试”提供有关基准测试和如何执行基准测试的概述。
- “配置 DB2”讨论数据库管理器、数据库配置文件和配置参数的值。

## 附录

- “DB2 注册表和环境变量”描述简要表的注册表值和环境变量。
- “解释表和定义”描述“DB2 解释”设施使用的表以及如何创建那些表。
- “SQL 解释工具”描述如何使用 DB2 解释工具: db2expln 和 dynexpln。
- “db2exfmt — 解释表格式工具”描述如何使用 DB2 解释工具来格式化解释表数据。



---

# 第1部分 “DB2 通用数据库” 的世界





---

## 第1章 管理 DB2 通用数据库

DB2 很灵活，它允许您运行各种硬件配置。它允许您选择如何使用特定的 DB2 产品配置来最好地满足您的硬件需求和应用程序需求。

由于 DB2 支持的数据库环境的复杂程度有所不同，因此针对每个环境有着特定的注意事项和任务。这些在《管理指南》和 DB2 库中的其他书籍中有详细的讨论（参见第241页的『附录E. 使用 DB2 资料库』）。在某些情况下，这些书籍的全部章节只适合于某个特定的环境。在阅读本书的前言（“关于本书”）之后，您将了解本书以及其他卷的《管理指南》（《管理指南：实现》和《管理指南：性能》）中的哪些章节适合于您的商业需要。

如果您不熟悉关系数据库管理系统 (RDBMS) 或 DB2，您将会发现标题为“基本关系数据库概念”的那一节对您很有帮助。如果您熟悉这些概念，或者不需要进行回顾，则可跳过本节，并直接转到描述更高级主题的那些节中，如：

- 联合系统。本节讨论数据库管理系统 (DBMS)，该系统支持应用程序和用户单条语句中提交引用两个或多个 DBMS 或数据库的 SQL 语句。
- 并行数据库系统。本节提供有关 DB2 可使用的并行性类型的介绍。任务（如数据库查询）的各个部分可并行运行以大幅提高性能。
- 分布式事务处理。本节讨论如何在单一事务中访问多个数据库，以及如何在分布式事务处理环境中使用数据库。

DB2 可以满足您最特殊的数据管理需要，如：

- 复制，它允许您定期将数据复制至多个远程数据库。若需要将主数据库中的更新自动复制至其他数据库，则可使用 DB2 的复制功能部件来指定应复制哪些数据，应将数据复制到哪些数据库表，以及应以何种频率复制更新。若要使用 DB2 的复制功能部件，请参考 *Replication Guide and Reference*。该书介绍 DB2 数据复制的概念，并描述如何规划、配置和管理复制环境。
- 数据入库，在此操作中，您可以存储“资料式数据”，即从运行数据中抽取出来然后进行变换以供最终用户进行决策的数据。例如，数据入库工具可以从运行数据库复制所有销售数据，执行计算以对该数据进行摘要，并将摘要数据写至单独数据库中的目标。您可以查询单独的数据库（仓库），而不会对运行数据库有所影响。有关数据入库的详细信息，参考《数据仓库中心管理指南》。
- 地理信息系统 (GIS)，它可以通过 Spatial Extender 创建。GIS 是对象、数据和应用程序的复合体，它允许您生成和分析关于地理面貌的空间信息。在 Spatial

Extender 中，地理面貌可以由表或视图中的一行表示，也可以由这样的行的一部分表示。有关使用 Spatial Extender 的详细信息，参考《*Spatial Extender 用户指南和参考*》。

《*管理指南：计划*》还讨论了数据库设计，包括 DB2 的逻辑数据库设计和物理数据库设计注意事项。还讨论了其他计划问题，如计划数据库迁移、标识可能会对应用程序有所影响的不兼容性（不兼容性是 DB2 通用数据库的一部分，它的工作方式与它在 DB2 先前发行版中的工作方式不同；若在现有的应用程序中使用，则会生成预期不到的结果，或者需要更改应用程序，或者降低性能）以及利用本地语言支持 (NLS)。

《*管理指南：实现*》阐述了与实现数据库设计有关的详细信息。本书包括有关创建和改变数据库、数据库安全性和使用“控制中心”（这是一个 DB2 图形用户界面）管理 DB2 的主题。

《*管理指南：性能*》阐述了与建立、测试以及改进应用程序及 DB2 本身的性能相关的主题和问题。

---

## 第2部分 数据库概念



---

## 第2章 基本关系数据库概念

本节包括下列主题:

- 『数据库对象概述』
- 第12页的『存储器对象概述』
- 第17页的『系统对象概述』
- 第19页的『数据的商业规则』
- 第22页的『恢复数据库』
- 第23页的『重组数据库中的表』
- 第23页的『DB2 安全性概述』

---

### 数据库对象概述

本节提供了下列关键数据库对象的概述:

- 实例
- 数据库
- 节点组
- 表
- 视图
- 索引
- 模式
- 系统目录表

第8页的图1举例说明了这些对象的某一些之间的关系。它还显示了表、索引和长整数数据存储在表空间中的情况。

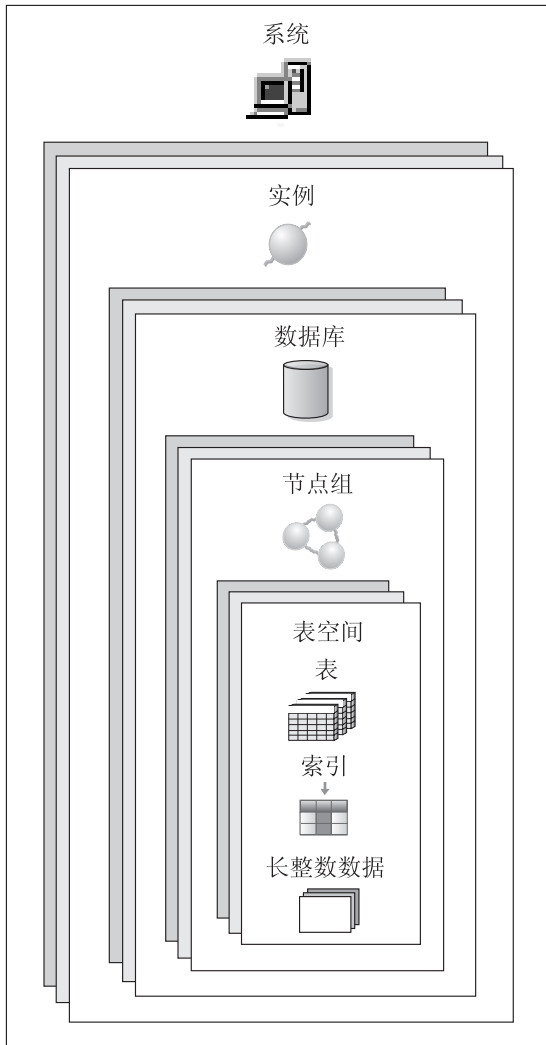


图 1. 一些数据库对象之间的关系

## 实例

实例（有时称为数据库管理器）是管理数据的 DB2 代码。它控制可对数据执行的操作，并管理分配给它的系统资源。每一个实例都是一个完整的环境。它包含为一个给定的并行数据库系统定义的所有数据库分区（参见第31页的『第4章 并行数

数据库系统』)。一个实例有它自己的数据库（其他实例不能访问它），并且它的全部数据库分区共享相同的系统目录。它还有独立的安全性，不受同一机器（系统）上其他实例的影响。

## 数据库

关系数据库将数据表示成表的集合。表由数目已定的列和任意数目的行组成。每个数据库都包括一组描述数据的逻辑和物理结构的系统目录表，一个包含为该数据库分配的参数的配置文件以及一个带有正在进行的事务和可归档事务的恢复日志。

## 节点组

节点组是一个或多个数据库分区组成的集合。当您想为数据库创建表时，首先创建节点组以存储表空间，然后创建表空间以存储表。有关节点组的更多信息，参见第31页的『节点组和数据分区』。有关数据库分区的定义，参见第31页的『第4章 并行数据库系统』。有关表空间的更多信息，参见第12页的『表空间』。

## 表

关系数据库将数据表示成表的集合。表由逻辑排列的行和列数据组成。所有数据库和表数据都被存储在表空间中。有关表空间的更多信息，参见第12页的『表空间』。表中的数据在逻辑上是相关的，且可以定义表与表之间的关系。根据数学规则和关系运算来查看和操纵数据。

表数据通过“结构化查询语言”（SQL，参见 *SQL Reference*）访问，SQL 是一种标准化语言，用于定义和操纵关系数据库中的数据。应用程序或用户使用查询从数据库中检索数据。查询使用 SQL 来创建如下格式的语句

```
SELECT <data_name> FROM <table_name>
```

## 视图

视图是高效率的数据呈示方法（无需维护数据）。视图不是实际的表，不需要永久性存储器。创建并使用一个“虚拟表”。

视图可以包括它所基于的表中的所有或某些列或行。例如，可以在视图中连接一个部门表和一个雇员表，以便可以列出特定部门中的所有雇员。

第10页的图2显示了表与视图之间的关系。

## 数据库

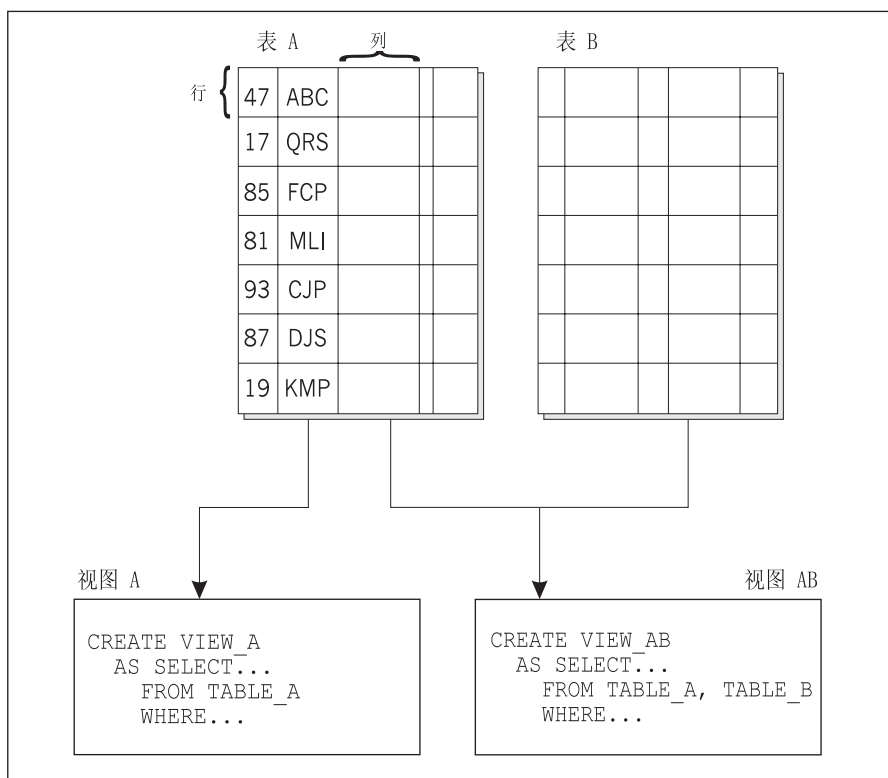


图2. 表与视图之间的关系

## 索引

索引是一组键，其每一个都指向一些表行。例如，在第11页的图3中，表 A 的一个索引基于表中的雇员号码。此键值提供指向表行的指针：雇员号码 19 指向雇员 KMP。通过使用指针创建指向数据的直接路径，索引使更有效率地访问表行成为可能。

SQL 优化器自动选择最有效率的访问表中数据的方法。当确定最快速的数据访问路径时，优化器会将索引考虑在内。

可创建唯一索引以确保索引键的唯一性。索引键是定义了一个索引的一个列或一些列的有序集合。使用唯一索引将确保在编入索引的列中，每个索引键的值都是唯一的。第19页的『数据的商业规则』更详细地描述了键和索引。



图3显示了索引与表之间的关系。

## 数据库

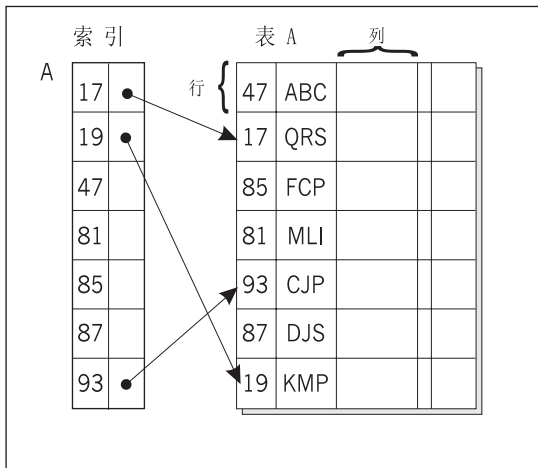


图 3. 索引与表之间的关系

## 模式

模式是一个标识符，如用户标识，它帮助分组表和其他数据库对象。模式可以归个人拥有，所有者可以控制对数据以及其中的对象的访问。

模式也可以是数据库中的对象。它可以在创建模式中的第一个对象时自动创建。这样的对象可以是任何可以由模式名限定的对象，如表、索引、视图、软件包、单值类型、函数或触发器。若要自动创建模式，则您必须拥有 `IMPLICIT_SCHEMA` 权限，也可以隐式地创建模式。

模式名用作两部分对象名的第一部分。创建一个对象时，可将其分配给特定模式。若不指定模式，则它被分配给缺省模式，缺省模式通常是创建该对象的人的用户标识。名称的第二部分是对象名。例如，名为 `Smith` 的用户可以有一个名为 `SMITH.PAYROLL` 的表。

## 系统目录表

每个数据库都包括一组描述数据的逻辑和物理结构的系统目录表。DB2 为每个数据库维护一大组系统目录表。这些表包含有关数据库对象（例如，用户表、视图和索引）的定义的信息，以及用户对这些对象所拥有的权限的安全性信息。它们在

数据库创建时被创建，并在常规操作期间得到更新。不能显式地创建或删除它们，但是可以使用目录视图查询和查看它们的内容。

---

## 存储器对象概述

下列数据库对象用来定义在系统上存储数据的方式以及改进（与访问数据相关的）性能的方法：

- 表空间
- 容器
- 缓冲池

### 表空间

数据库由称为表空间的部件组成。表空间是用来存储表的位置。当创建表时，您可以决定将特定对象（如索引和大对象（LOB））数据与其余表数据分开存放。表空间也可以分布在一个或多个物理存储设备上。下图显示了在表空间之间分布数据时具有的一些灵活性。

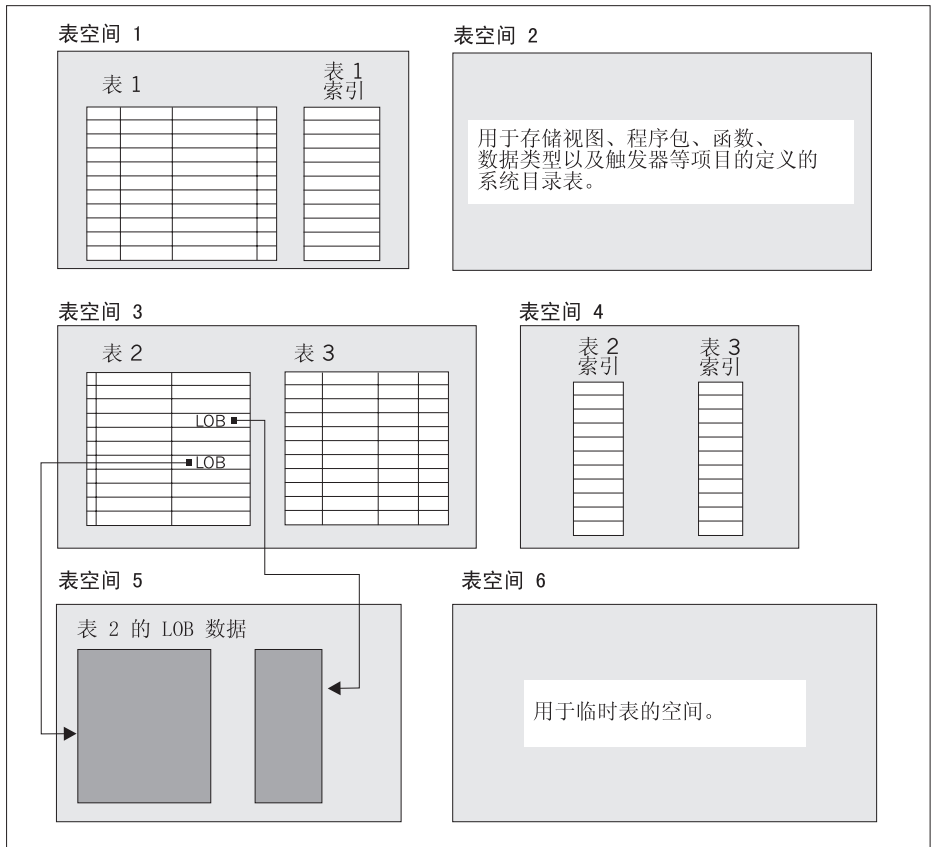


图 4. 表空间

表空间驻留在节点组中（参见第9页的『节点组』）。表空间定义和属性记录在数据库系统目录中（参见第11页的『系统目录表』）。

将容器分配给表空间。容器是分配的物理存储器（如文件或设备）。

表空间可以是系统管理空间 (SMS) 或数据库管理空间 (DMS)。对于 SMS 表空间，每个容器都是操作系统的文件空间中的一个目录，由操作系统的文件管理器控制存储空间。对于 DMS 表空间，每个容器或者是固定大小的预分配文件，或者是物理设备如磁盘，由数据库管理器控制存储空间。

第14页的图5举例说明了表、表空间以及两种类型的空间之间的关系。它还显示了表、索引和长整数数据存储表空间中的情况。

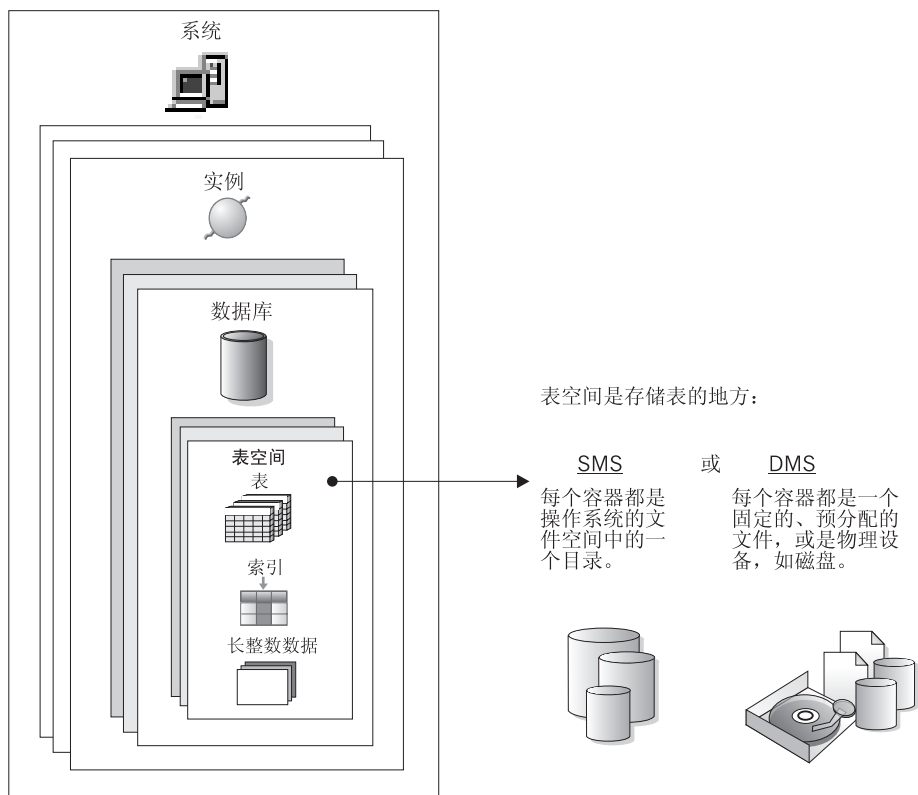


图 5. 表空间和表

第15页的图6显示了三种表空间类型：规则、临时和长整数。

包含用户数据的表存放在规则表空间中。缺省用户表空间名为 `USERSPACE1`。索引也存储在规则表空间中。系统目录表存放在规则表空间中。缺省系统目录表空间名为 `SYSCATSPACE`。

包含长整数字段数据或长整数对象数据（如多媒体对象）的表存放在长整数表空间中。

临时表空间分为系统临时表空间或用户临时表空间。系统临时表空间用来存储 SQL 操作（如排序、重组表、创建索引和连接表）期间所需的内部临时数据。虽然可以创建任意数目个系统临时表空间，但建议您只使用大多数表所使用的页大小创建一个。缺省系统临时表空间名为 `TEMPSPACE1`。用户临时表空间用来存储已说

明全局临时表（已说明全局临时表存储的是应用程序临时数据）。用户临时表空间不是在数据库创建时缺省创建的。

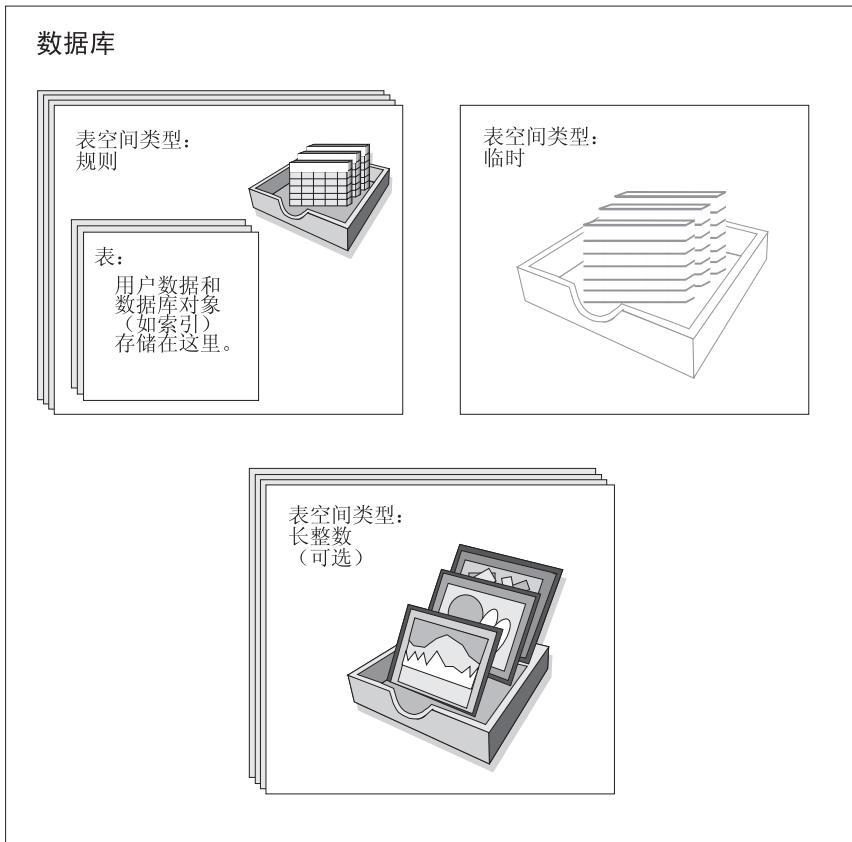


图 6. 三种表空间类型

## 容器

容器是物理存储设备。它可以由目录名、设备名或文件名标识。

将为表空间分配容器。单个表空间可以横跨多个容器，但每个容器只能属于一个表空间。

第16页的图7举例说明了表与数据库中的表空间、相关联的容器和磁盘之间的关系。

## 数据库

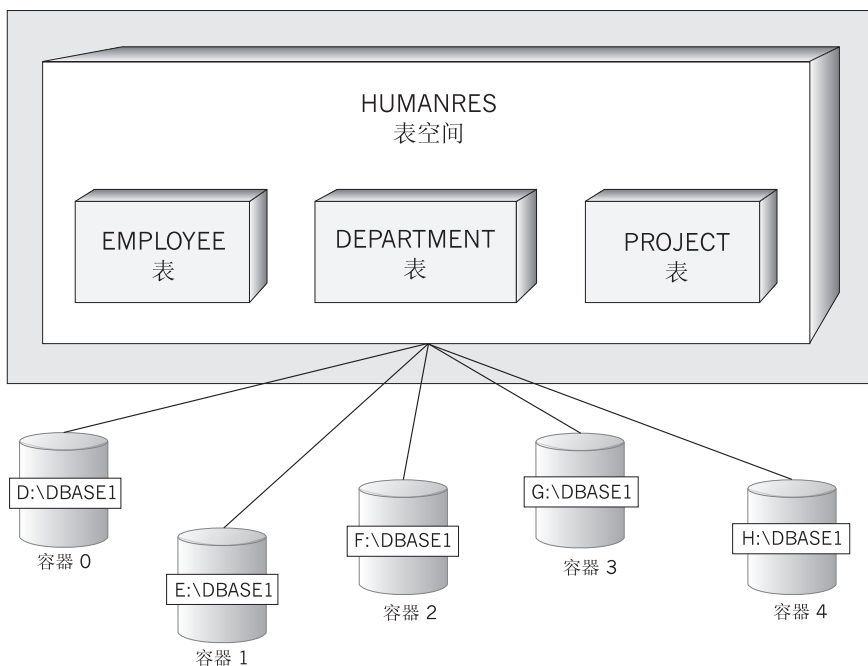


图 7. 数据库中的表空间和表

EMPLOYEE、DEPARTMENT 和 PROJECT 表在 HUMANRES 表空间中，该表空间横跨容器 0、1、2、3 和 4。此示例显示每个容器存在于不同的磁盘上。

任何表的数据都以循环方式存储在表空间中的所有容器中。这能在属于给定表空间的容器之间平衡数据。数据库管理器在使用另一个容器之前写入一个容器的页数称为数据块大小。

## 缓冲池

缓冲池指的是从磁盘读取高速缓存表和索引数据页时或修改它们时分配给它们的主内存。缓冲池的目的是改进系统性能。从内存访问数据要比从磁盘访问数据快得多；因此，数据库管理器需要读写磁盘 (I/O) 的次数越少，性能也越好。（可以创建多个缓冲池，虽然在大多数情况下只需要一个。）

因为可以缩短慢速 I/O 所造成的延迟，所以缓冲池的配置是最为重要的调整项目。

第17页的图8举例说明了缓冲池与容器之间的关系。

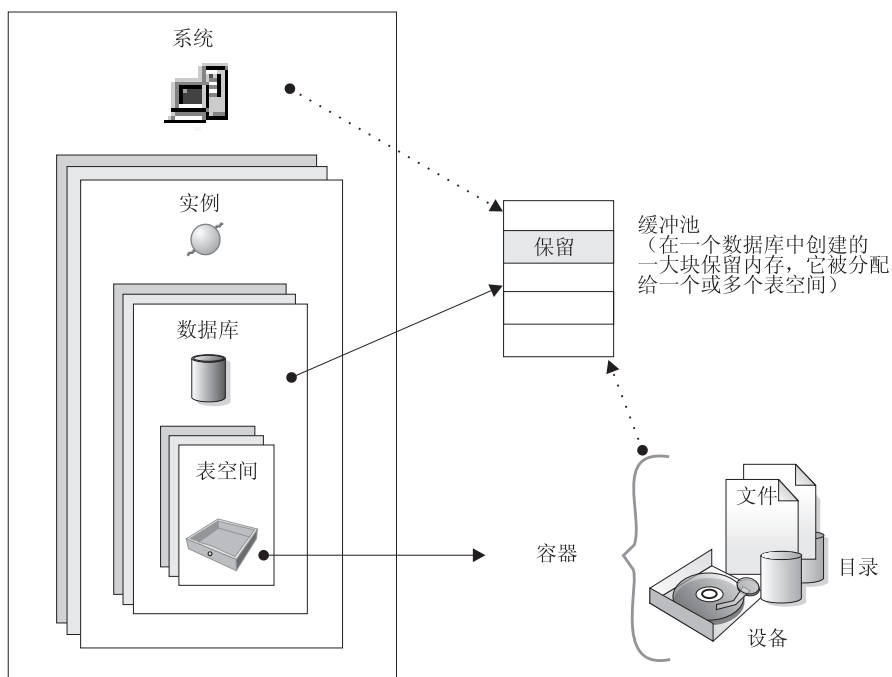


图 8. 缓冲池和容器

## 系统对象概述

当创建 DB2 实例或数据库时，将使用缺省参数值创建相对应的配置文件。可修改这些参数值以改进性能。

### 配置参数

配置文件包含特定的参数，这些参数定义诸如分配给 DB2 产品和个别数据库的资源以及诊断级别之类的值。有两种类型的配置文件：用于每个 DB2 实例的数据库管理器配置文件，以及用于每个个别数据库的数据库配置文件（参见第18页的图 9）。

数据库管理器配置文件是在创建 DB2 实例时创建的。它包含的参数在实例级影响系统资源，不受任何一个作为该实例一部分的数据库的影响。根据系统的配置，可将这些参数中的许多参数的值更改为非系统缺省值，以改进性能或增加容量。

每个客户机安装也有一个数据库管理器配置文件。此文件包含关于特定工作站的客户机使能器的信息。在可用于服务器的参数中，有一个子集可用于客户机。

数据库配置文件是在创建数据库时创建的，它驻留在数据库所驻留的地方。每个数据库都有一个配置文件。其参数指定要分配给该数据库的资源量以及其他事项。可更改许多参数的值，以改进性能或增加容量。根据特定数据库中活动类型的不同，可能需要进行不同的更改。

数据库对象 / 概念 等价的物理对象

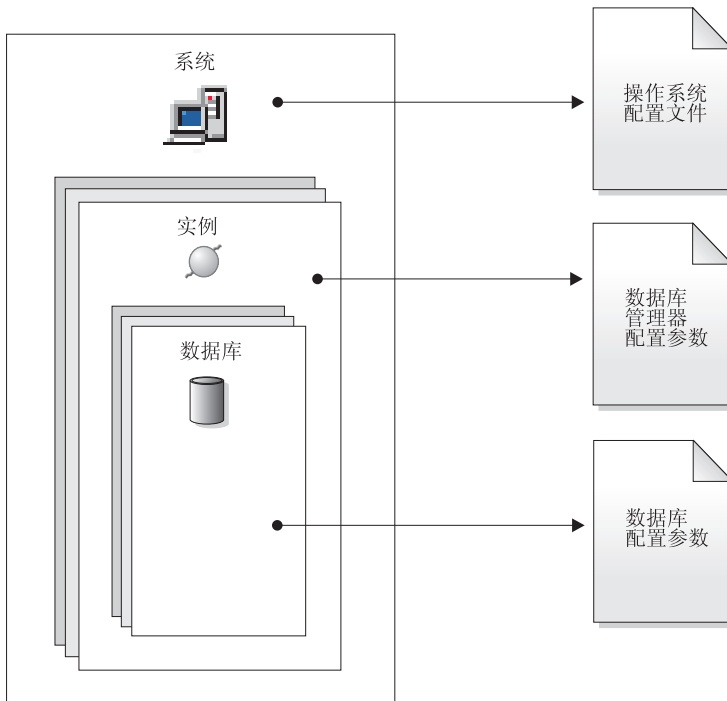


图 9. 配置参数文件



## 数据的商业规则

在任何商业中，数据通常必须符合特定限制或规则。例如，雇员号码必须是唯一的。DB2 提供了约束作为实施这样的规则的方法。

DB2 提供了下列约束类型：

- NOT NULL 约束
- 唯一约束
- 主键约束
- 外键约束
- 检查约束

### NOT NULL 约束

NOT NULL 约束防止空值进入一个列。

### 唯一约束

唯一约束确保一组列中的值对于表中的所有行都是唯一的，且不为空。例如，DEPARTMENT 表中的典型唯一约束可以是：部门号是唯一的，且不为空。

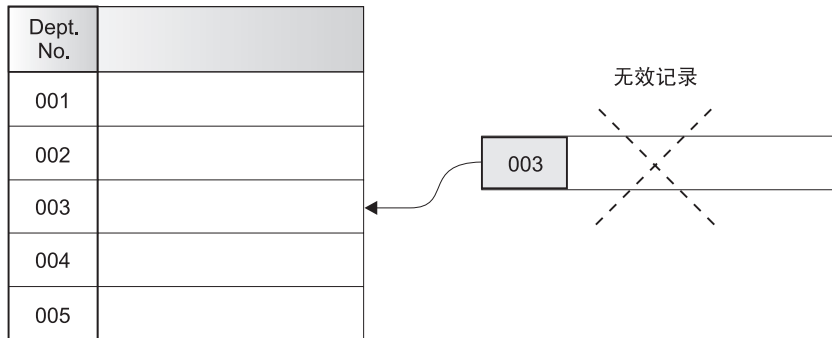


图 10. 唯一约束防止数据重复

数据库管理器在插入和更新操作期间强制执行此约束，以确保数据完整性。

### 主键约束

每个表都可以有一个主键。主键是与唯一约束具有相同特性的一个列或列的组合。可使用主键和外键约束来定义表之间的关系。

因为主键用来标识表中的一行，所以它应该是唯一的，并且只进行非常少的添加或删除。一个表不能有多个主键，但可以有多唯一键。主键是可

选的，可以在创建或改变表时定义。当导出或重组数据时，主键可以对数据进行排序，所以它们也是有益的。

在下面的表中，DEPTNO 和 EMPNO 是 DEPARTMENT 表和 EMPLOYEE 表的主键。

表 1. DEPARTMENT 表

DEPTNO (主键)	DEPTNAME	MGRNO
A00	Spiffy Computer Service Division	000010
B01	Planning	000020
C01	Information Center	000030
D11	Manufacturing Systems	000060

表 2. EMPLOYEE 表

EMPNO (主 键)	FIRSTNAME	LASTNAME	WORKDEPT (外键)	PHONENO
000010	Christine	Haas	A00	3978
000030	Sally	Kwan	C01	4738
000060	Irving	Stern	D11	6423
000120	Sean	O'Connell	A00	2167
000140	Heather	Nicholls	C01	1793
000170	Masatoshi	Yoshimura	D11	2890

## 外键约束

外键约束（也称为引用完整性约束）使您能够定义表间以及表内必需的关系。

例如，典型的外键约束可能规定 EMPLOYEE 表中的每个雇员必须是一个现有部门的成员，该部门在 DEPARTMENT 表中定义。

要建立此关系，应将 EMPLOYEE 表中的部门号定义成外键，并将 DEPARTMENT 表中的部门号定义成主键。

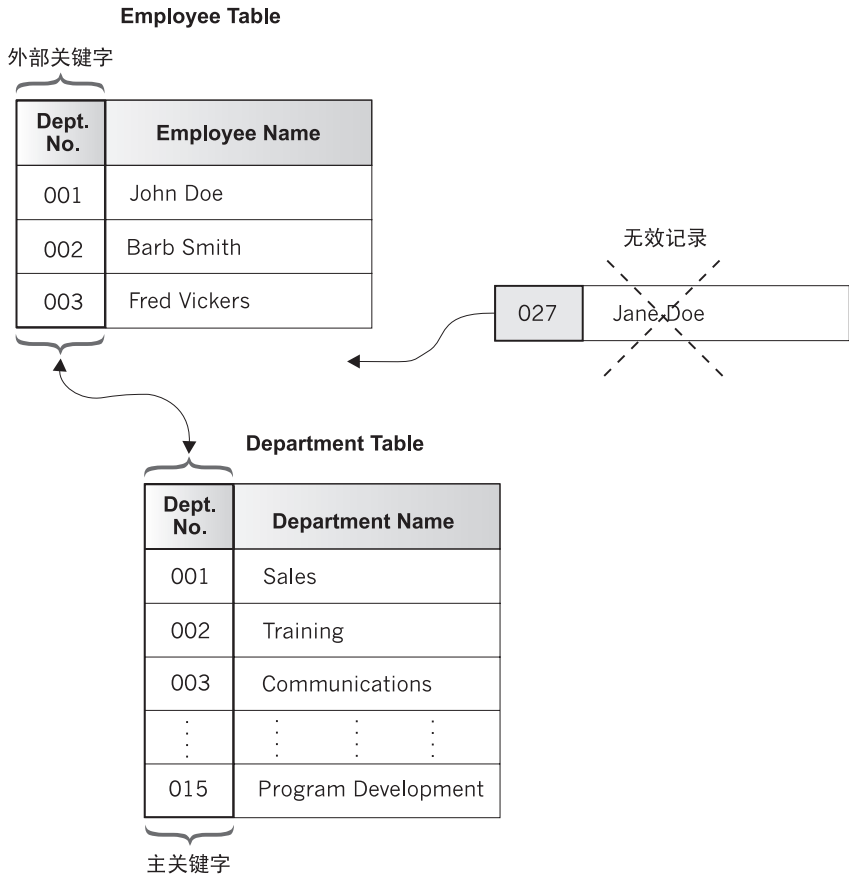


图 11. 外键约束和主键约束定义关键并保护数据

### 检查约束

检查约束是指定每个表行的一列或多列所允许的值的数据库规则。

例如，在 **EMPLOYEE** 表中，可将“工作类型”列定义为“推销员”、“经理”或“职员”。有了此约束，在“工作类型”列中具有不同的值的任何记录都是无效的，将被拒绝，并实施关于表中允许的数据类型的规则。

还可在数据库中使用触发器。与约束相比，触发器更为复杂，且潜在功能更加强大。它们定义一组操作，这组操作与用于指定基表的 **INSERT**、**UPDATE** 或 **DELETE** 子句一起执行或由这些子句触发。可使用触发器支持一般格式的完整性或商业规则。例如，在接受订单之前，可使用触发器来检查客户的信用额度，也可以在银行业务应用程序中使用触发器，以便在帐户提款未符合客户的标准提款模式时提高警惕。有关触发器的更多信息，参考 *Application Development Guide*。

---

## 恢复数据库

数据库可能由于硬件和 / 或软件故障而变得不能使用，不同的故障情形可能需要不同的恢复操作。应该有一个适当的预演策略，以防止数据库可能出现故障。

可以在 *Data Recovery and High Availability Guide and Reference* 中找到关于备份、恢复和如何开发良好的备份与恢复策略的深入信息。

数据库备份的概念与其他任何数据备份的概念一样：即，复制一份数据，然后将它存储在另一媒体上，以防原始媒体发生故障或毁坏。最简单的备份案例涉及关闭数据库（以确保不发生更多的事务），然后简单地对其进行备份。

如果数据库损坏或毁坏，可以从备份映像重建它。数据库的重建称为恢复。

若在执行交易时数据库崩溃，当重新启动数据库时将执行称为崩溃恢复的进程。崩溃恢复是将数据库移动回一致并可用状态的进程。这是通过回滚未完成的交易，并完成在发生崩溃时仍在内存中的已提交交易来完成的。

若数据库损坏或毁坏，有两种方式来恢复它：版本恢复和前滚恢复。

版本恢复指的是使用备份操作期间创建的映像来复原数据库的先前版本。数据库备份使您可以将数据库复原到与进行备份时完全相同的状态。但是，从备份时间到故障时间之间的每个工作单元都将丢失。

若想要在过了执行备份时的时间点能够复原数据库，必须使用前滚恢复。要使用前滚恢复方法，必须已建立了数据库的备份，并且已将日志归档（通过启用 *logretain* 和 / 或 *userexit* 数据库配置参数）。

每个数据库都包括恢复日志，它们用来从应用程序或系统错误中恢复。与数据库备份一起，它们可用来将数据库正好恢复到出错时的时间点。日志文件是在创建数据库时自动创建的。不能直接修改日志文件。

另一个重要的恢复对象是恢复历史文件。恢复历史文件包含特定备份信息的摘要，当必须将整个数据库或数据库的一部分恢复至给定时间点时，可以使用这些信息。恢复历史文件用来跟踪与恢复相关的事件，如备份、复原和装入操作。

**注：***Data Recovery and High Availability Guide and Reference* 中已合并 *Command Reference* 和 *API Reference* 中所有关于备份与恢复的信息。*Data Recovery and High Availability Guide and Reference* 是有关备份与恢复的主要的、单一信息来源。

---

## 重组数据库中的表

在许多次的更新之后，表会成为碎片状，这导致性能严重下降。若收集了统计信息，但看不出有明显的性能改进，则重组表数据可能会有所帮助。重组表数据时，根据指定的索引重新安排数据的物理顺序，并除去碎片数据中的自由空间。这使该数据可以更快速地被访问，从而改进性能。

在重组表之前，建议您调用 `REORGCHK` 命令，并收集该表的统计信息。运行此命令将帮助您确定重组表数据是否适当。关于 `REORGCHK` 命令的信息，参考 *Command Reference*。

---

## DB2 安全性概述

要保护与一个数据库服务器相关的数据和资源，DB2 同时使用外部安全性服务与内部访问控制信息。要访问一个数据库服务器，必须在被允许访问数据库数据或资源之前通过一些安全性检查。数据库安全性中的第一步称为认证，在此步骤，您必须证明您的身份是真实的。第二步称为授权，在此步骤，数据库管理器决定是否允许已经验证的用户执行请求的操作或访问请求的数据。

### 认证

使用 DB2 之外的安全性设施来完成用户认证。此安全性设施可以是操作系统的一部分，可以是一个独立的产品，或者，在特定情况下，可能根本不存在。例如，在基于 UNIX 的系统上，安全性设施包含在操作系统之内。“DCE 安全性服务”是一个独立的产品，它为分布式环境提供安全性设施。而在 Windows 95 或 Windows 3.1 操作系统上没有安全性设施。

安全性设施需要两个项目来认证用户：用户标识和密码。用户标识向安全性设施标识用户。通过提供正确的密码（只有该用户和安全性设施才知道的信息）来验证该用户的身份（与该用户标识相对应）。

一旦认证：

- 就必须使用一个 SQL 授权名或 `authid` 来向 DB2 标识该用户。此名称可以与用户标识或一个映射值相同。例如，在基于 UNIX 的系统上，可将符合 DB2 命名约定的一个 UNIX 用户标识转换为大写字体，来获得一个 DB2 `authid`。在“DCE 安全性服务”产品中，DB2 `authid` 包含在 DCE 注册表中，一旦认证成功完成，则可从注册表中将它提取出来。
- 获取用户所属的组的列表。在授权用户时，可使用组成员资格。组是安全性设施实体，它们也必须映射至 DB2 授权名。执行此映射的方法与映射用户标识的方法类似。

DB2 将获得最多 64 组的组列表。若一个用户所属的组超过 64 组，只将映射至有效 DB2 授权名的前 64 组添加到 DB2 组列表。不会返回任何错误，而且 DB2 将忽略前 64 组之后的所有组。

DB2 通过下列两种方式之一，使用安全性设施来认证用户：

- DB2 使用成功的安全性系统登录作为身份证明，并允许：
  - 使用本地命令访问本地数据
  - 使用远程连接，在这种情况下服务器信赖客户机认证。
- DB2 接受用户标识和密码的组合。它使用安全性设施对此组合的成功验证作为身份证明，并允许：
  - 使用远程连接，在这种情况下服务器需要认证的证明
  - 使用操作，在这种情况下用户希望以某个不同于注册时所用的身份来运行命令。

DB2 管理员可允许其他人在 AIX 和 Windows NT EEE 系统上，通过简要表注册表变量 DB2CHGPWD\_EEE 来更改密码。此变量的缺省值是 NOT SET（禁用）。DB2CHGPWD\_EEE 接受由其他 DB2 简要表变量使用的标准布尔值。

DB2 管理员负责在 Windows NT 上使用“Windows NT 域控制器”或在 AIX 上使用 NIS，来确保集中维护所有节点的密码。

**注：**若没有集中维护密码，启用 DB2CHGPWD\_EEE 变量可能造成密码在所有节点上不一致。即，若使用“更改密码”功能部件，则您的密码仅在您连接的节点上被更改。

AIX 上的 DB2 UDB 可记录对操作系统的失败密码尝试次数，并检测客户机何时超过允许的注册尝试次数，该值由 LOGINRETRIES 参数指定。

有关系统进入验证检查的其他信息（若您有访问数据库的远程客户机，该信息就更有参考价值），参见《管理指南：实现》中的“为服务器选择认证方法”。

## 授权

授权是一个过程，通过这个过程 DB2 获取有关一个已认证的 DB2 用户的信息，此信息指示该用户可执行的数据库操作，以及可以访问的数据对象。执行每个用户请求时，根据所涉及的对象和操作，可能有多个授权检查。

使用 DB2 设施来执行授权。DB2 表和配置文件用于记录与每个授权名相关的许可权。将一个已认证的用户的授权名和此用户所属的那些组的授权名，与记录的许可权比较。根据这个比较，DB2 决定是否允许执行请求的访问。

DB2 记录了两种类型的许可权：特权和权限级。特权为授权名定义单个许可权，它使用户能够创建或访问数据库资源。特权存储在数据库目录中。权限级提供将特权分组的一个方法，并控制更高级别的数据库管理器的维护和实用程序操作。数据库特定的权限存储在数据库目录中；系统权限与组成员资格相关联，并存储在一个给定实例的数据库管理器配置文件中。

组提供了一个简便的方法，来对一组用户执行授权，而不必单独对每个用户授予或撤消特权。除非另有指定，否则，组授权名可以用在为了授权而使用授权名的任何地方。通常，对于动态 SQL 和非数据库对象授权（如实例级命令和实用程序），考虑使用组成员资格，但对于静态 SQL，则不考虑使用它。这种一般情况的例外是，当授予 PUBLIC 特权时：在处理静态 SQL 时要考虑它们。DB2 文档中的适当地方提到了组成员资格不适用的特殊情况。

有关更多信息，参见《管理指南：实现》中的“特权、权限和授权”。

## 联合数据库认证和授权概述

因为 DB2 联合数据库系统可访问多个数据库管理系统中的信息，所以可能需要附加的步骤来保密数据。

当计划认证方法时，应考虑用户可能需要通过数据源和 DB2 的认证检查这一事实。在联合系统中，认证可在 DB2 客户机工作站、DB2 服务器、数据源（DB2、DB2 OS/390 版、其他 DRDA 服务器、Oracle）或在 DB2（客户机或 DB2 服务器）和数据源的组合中进行。甚至在 DCE 环境中，若数据源需要用户标识和密码，也可能需要特定的步骤。有关更多信息，参见《管理指南：实现》中的“联合数据库认证处理”。

类似地，用户必须通过数据源和 DB2 的授权检查。每个数据源（DB2、Oracle、DB2 OS/390 版等）在其控制下维护对象的安全性。当用户根据别名执行操作时，用户必须通过该别名引用的表或视图的授权检查。





---

## 第3章 联合系统

联合数据库系统或联合系统是一个数据库管理系统 (DBMS)，它支持提交特定 SQL 语句的应用程序或用户，这些 SQL 语句在单条语句中引用两个或更多个 DBMS 或数据库。一个示例是两个不同的 DB2 数据库中的表之间的连接。这种类型的语句称为分布式请求。

“DB2 通用数据库”联合系统支持跨数据库和 DBMS 的分布式请求。例如，可在 DB2 表和 Oracle 视图之间执行一个 UNION 操作。受支持的 DBMS 包括 DB2、DB2 系列的成员（如 DB2 OS/390 版和 DB2 AS/400 版）和 Oracle。

DB2 联合系统为数据库对象提供位置透明性。若（表和视图中的）信息被移动，则不必对请求该信息的应用程序进行任何更改，便可更新对该信息的引用（称为别名）。DB2 联合系统还弥补了不支持所有 DB2 SQL 语言或特定优化能力的 DBMS 的不足。不能在这样的 DBMS 下执行的操作（如递归 SQL）可以在 DB2 下运行。

DB2 联合系统以一种半自动方式运行：当 Oracle 应用程序访问同一个服务器时，可提交引用 Oracle 对象的 DB2 查询。DB2 联合系统不独占或限制对 Oracle 或其他 DBMS 对象的访问权（不包括完整性 / 锁定约束）。

一个 DB2 联合系统由一个 DB2 UDB 实例、一个充当联合数据库的数据库以及一个或多个数据源组成。联合数据库包含标识数据源及其特征的目录项。数据源由一个 DBMS 和数据组成。应用程序与联合数据库连接就如同与任何其他 DB2 数据库连接一样。有关联合数据库环境的形象化表示，参见第28页的图12。

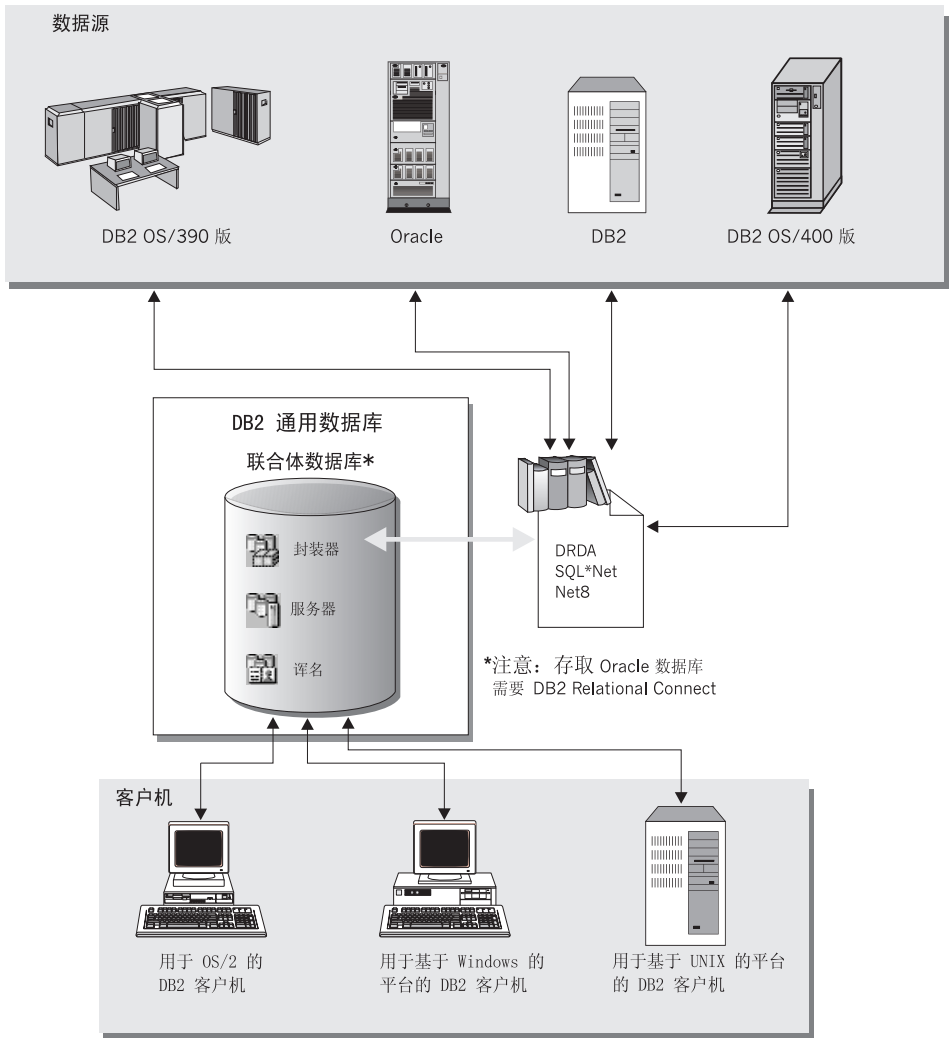


图 12. 联合数据库系统

DB2 联合数据库目录项包含有关数据源对象的信息：它们的名称、所包含的信息以及使用它们的条件。因此 DB2 目录将有关对象的信息存储在多个 DBMS 中，因此它称为全局目录。也将对象属性存储在该目录中。引用的实际 DBMS、用于与数据源通信的模块以及将访问的 DBMS 数据对象（如表）在数据库之外。（一个例外是：联合数据库可以是联合系统的一个数据源。）可使用控制中心或 SQL DDL 语句创建联合对象。必需的联合数据库对象包括：

**包装器** 标识访问特定类或类别的数据源所用的模块（DLL、库等）。

**服务器** 定义数据源。服务器数据包括包装器名、服务器名、服务器类型、服务器版本、授权信息和服务器选项。

**别名** 存储在联合数据库中的标识符，它们引用特定的数据源对象（表、别名、视图）。应用程序在查询中引用别名就象引用表和视图一样。

可创建附加对象，这取决于您的特定需要：

- 用户映射，解决认证问题
- 数据类型映射，定制数据源类型和 DB2 类型之间的关系
- 函数映射，将本地函数映射为数据源函数
- 索引规范，改进性能。

设置联合系统后，可以象在一个大型数据库中一样访问数据源中的信息。用户和应用程序将查询发送到一个联合数据库，然后该数据库根据需从 DB2 系列和 Oracle 系统中检索数据。用户和应用程序在查询中指定别名；这些别名提供对数据源中的表和视图的引用。从最终用户的角度来看，别名与别名是相似的。

有许多因素影响联合系统性能。最关键的因素是确保将关于数据源的准确的、最新的信息存储在联合数据库的全局目录中。DB2 优化器使用此信息，该信息可影响是否下推操作以便在数据源求值的决定。有关联合系统性能的其他信息，参考《管理指南：性能》。

DB2 联合系统运行时受某些限制的约束。分布式请求只限于只读操作。此外，不能对别名执行实用程序操作（LOAD、REORG、REORGCHK、IMPORT、RUNSTATS 等）。

但可使用通过设施并使用与数据源相关的 SQL 语言将 DDL 和 DML 语句直接提交给数据库管理器。

联合系统支持并行环境。将联合数据库按语义细分为本地对象（表、视图）引用和别名引用的范围限制了性能改善的程度。对别名数据的请求按顺序处理；本地对象可并行处理。例如，假设有查询 `SELECT * FROM A, B, C, D`，其中 A 和 B 是本地表，而 C 和 D 是引用位于 Oracle 数据源的表的别名，一个可能的方案是将表 A 和 B 并行连接。结果是按顺序与别名 C 和 D 连接。

---

## 启用联合系统

“DB2 企业版” (EE) 和 “DB2 扩充企业版” (EEE) 可以支持联合数据库。要启用联合系统：

1. 在安装期间，选择 DB2 EE 或 EEE 的 DB2 数据库的分布式连接安装选项。

2. 若联合系统中包括 Oracle 数据库，则安装 DB2 Relational Connect。有关更多信息，参考《安装和配置补遗》。
3. 将数据库管理器配置参数 *federated* 设置为 "YES"。
4. 创建包装器、服务器和别名（有关更多信息，参见《管理指南：实现》中的“创建数据库”）。
5. 按要求创建附加的对象或设置选项（有关更多信息，参见《管理指南：实现》中的“实现设计”）。

---

## 第4章 并行数据库系统

DB2 将数据库管理器扩展到了并行多节点环境中。数据库分区是数据库的一部分，它由自己的数据、索引、配置文件和事务日志组成。数据库分区有时称为节点或数据库节点。（“节点”是在“DB2 并行版 AIX 版的版本 1”产品中使用的术语。）

单分区数据库是只有一个数据库分区的数据库。该数据库中的所有数据都存储在此分区中。对于这种情况，节点组（若存在的话）不提供任何附加功能（参见第9页的『节点组』）。

分区数据库是有两个或更多个数据库分区的数据库。表可位于一个或多个数据库分区中。当一个表在由多个分区组成的节点组中时，它的某些行存储在一个分区中，而其他行存储在其他分区中。

通常，每个物理节点都存在单个数据库分区，而在每个数据库分区中，数据库管理器使用每个系统上的处理器来管理该数据库中的全部数据中属于该分区的那一部分。

由于数据分布在各数据库分区上，所以可在多个物理节点上使用多个处理器的能力来满足对信息的请求。数据检索和更新请求被自动分解成子请求，并在适用的数据库分区中并行执行。数据库分布在各数据库分区中的这个事实对于发出 SQL 语句的用户是透明的。

用户交互作用通过一个数据库分区发生，该数据库分区称为该用户的协调程序节点。协调程序与应用程序在相同的数据库分区上运行，或者对于远程应用程序，协调程序在该应用程序连接的数据库分区上运行。任何数据库分区都可用作协调程序节点。

---

### 节点组和数据分区

可以在数据库中定义一个或多个数据库分区组成的命名子集。您定义的每个子集称为节点组。包含多个数据库分区的每个子集称为多分区节点组。多分区节点组只能用属于相同实例的数据库分区来定义。

第32页的图13给出了一个含五个分区的数据库示例，在这个示例中：

- 一个节点组横跨除一个数据库分区外的所有其他分区（节点组 1）。
- 一个节点组包含一个数据库分区（节点组 2）。

- 一个节点组包含两个数据库分区。(节点组 3)。
- 节点组 2 中的数据库分区与节点组 1 共享并与之相交。
- 节点组 3 中只有一个数据库分区与节点组 1 共享并与之相交。

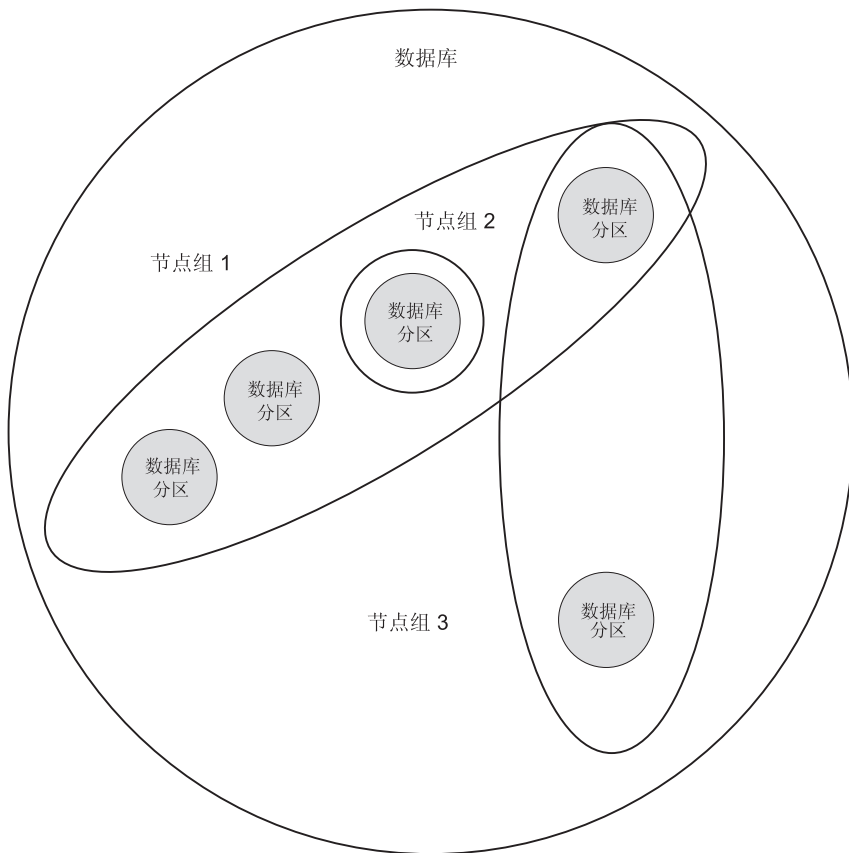


图 13. 数据库中的节点组

使用 `CREATE NODEGROUP` 语句可以创建一个新节点组。有关更多信息，参考 *SQL Reference*。数据分布在一个节点组中的所有分区上。若使用多分区节点组，则必须注意几个节点组设计注意事项。有关更多信息，参见第91页的『设计节点组』。

---

## 并行性的类型

任务（如数据库查询）的各个部分可并行运行以大幅增强性能。任务的性质、数据库配置和硬件环境确定了 DB2 将如何以并行方式执行任务。这些注意事项是相关的，在决定数据库的物理和逻辑设计方案时，应当一起考虑它们。本节描述受 DB2 支持的下列类型的并行性：

- I/O
- 查询
- 实用程序

### I/O 并行性

当一个表空间有多个容器时，数据库管理器可以利用并行 I/O。并行 I/O 指的是同时处理对两个或多个 I/O 设备的写入或读取；这能够显著地改进吞吐量。

I/O 并行性是第36页的『硬件环境』中描述的每个硬件环境的一个组成部分。第43页的表3列出了最适合 I/O 并行性的硬件环境。

### 查询并行性

有两种查询并行性：查询间并行性和查询内并行性。

**查询间并行性**是指多个应用程序同时查询一个数据库的能力。每个查询都独立于其他查询执行，但是 DB2 同时执行所有查询。DB2 始终支持这种类型的并行性。

**查询内并行性**是指使用分区内并行性和 / 或分区间并行性来同时处理单个查询的各部分。

术语**查询并行性**将在全书中使用。

#### 分区内并行性

**分区内并行性**是指将一个查询分为多个部分的能力。（有些实用程序也采用这种类型的并行性。查看第36页的『实用程序并行性』）。

分区内并行性将通常认为的单个数据库操作（如索引创建、数据库装入或 SQL 查询）细分成多个部分，其中的大部分或全部操作可以在单个数据库分区内以并行方式运行。

第34页的图14显示了一个查询，它被分为可并行运行的四个部分，返回结果的速度比按串行方式运行该查询的速度快得多。这几部分互为副本。要使用分区内并行性，必须适当地配置数据库。您可以选择并行度，或由系统为您选择。并行度表示一个查询中并行运行的部分数。

第43页的表3列出了最适合分区内并行性的硬件环境。

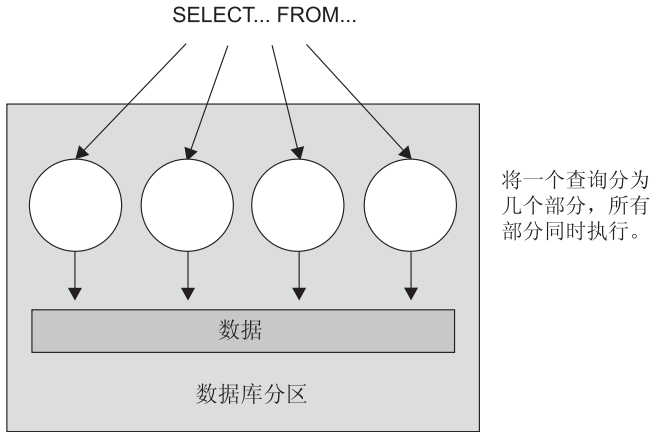


图 14. 分区内并行性

### 分区间并行性

分区间并行性是指将一个查询分为多个部分并将这几部分置于一个分区数据库的多个分区（位于一台或多台机器上）上的能力。该查询以并行方式运行。（有些实用程序也采用这种类型的并行性。查看第36页的『实用程序并行性』）。

分区间并行性将通常认为的单个数据库操作（如索引创建、数据库装入或 SQL 查询）细分成多个部分，其中的大部分或全部操作可以在一台或多台机器上的一个分区数据库的多个分区中以并行方式运行。

第35页的图15显示了一个查询，它被分为可并行运行的四个部分，返回结果的速度比在单个分区上按串行方式运行该查询的速度快得多。

并行度在很大程度上取决于您创建的分区数和您定义节点组的方式。

第43页的表3列出了最适合分区间并行性的硬件环境。



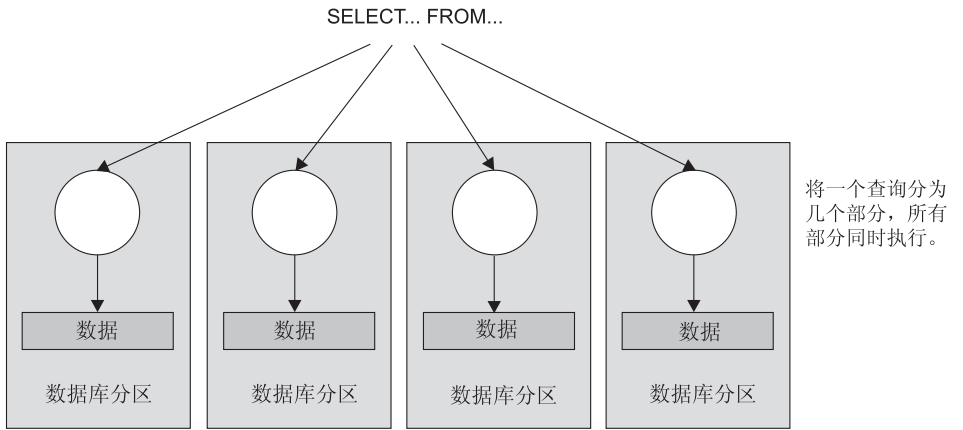


图 15. 分区间并行性

### 同时使用分区内和分区间并行性

可以同时使用分区内并行性和分区间并行性。此组合提供两种并行性程度，这也使处理查询的速度显著加快。

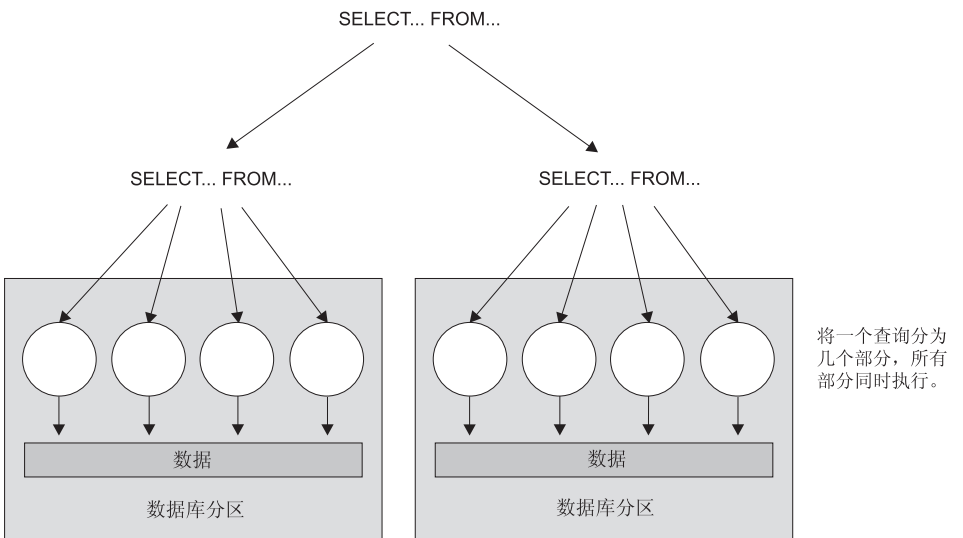


图 16. 同时使用分区间和分区内并行性

## 实用程序并行性

DB2 实用程序可以利用分区内并行性。它们还可以利用分区间并行性；前提是存在多个数据库分区，而实用程序在每个分区内并行执行。

装入实用程序可以利用分区内并行性和 I/O 并行性。装入数据是一个大量使用 CPU 的任务。装入实用程序利用多个处理器来执行如语法分析和格式化数据这类任务。它也可使用并行 I/O 服务器来以并行方式将数据写入容器中。有关如何对装入实用程序启用并行性的信息，参考 *Data Movement Utilities Guide and Reference*。

在分区数据库环境中，“自动装入”实用程序通过在该表驻留的每个数据库分区上并行调用 LOAD 命令，来利用分区内、分区间或 I/O 并行性。有关“自动装入”实用程序的更多信息，参考 *Data Movement Utilities Guide and Reference*。

在创建索引期间，可并行执行数据的扫描和后续排序。在创建索引时，DB2 既利用 I/O 并行性又利用分区内并行性。这有助于在发出 CREATE INDEX 语句时、重新启动期间（若一个索引标记为无效）及数据重组期间加快索引创建的速度。

备份和复原数据任务会涉及到繁重的涉及 I/O 的任务。当执行备份和复原操作时，DB2 既利用 I/O 并行性又利用分区内并行性。备份操作通过以并行方式读取多个表空间容器并以并行方式异步写入多备份媒体，来利用 I/O 并行性。有关如何对 BACKUP DATABASE 命令和 RESTORE DATABASE 命令启用并行性的信息，参考 *Command Reference* 中对这些实用程序的介绍。

---

## 硬件环境

本节对下列硬件环境提供了一个概述：

- 单处理器上的单分区
- 具有多个处理器的单分区 (SMP)
- 多分区配置
  - 具有一个处理器的分区 (MPP)
  - 具有多个处理器的分区 (SMP 群集)
  - 逻辑数据库分区 (在“DB2 并行版 AIX 版，版本 1”中又称为“多逻辑节点”，或 MLN)

下面讨论每种环境的容量和可伸缩性。容量是指能访问数据库的用户数和应用程序数。这很大程度上取决于内存、代理程序数、锁定数、I/O 和存储器管理。可伸缩性是指一个数据库随着其增长而继续显示出相同的操作特征和响应时间的能力。

## 单处理器上的单分区

此环境由内存和磁盘组成，但仅包含一个 CPU（参见图17）。可使用许多不同的名称来称呼它，包括独立数据库、客户机 / 服务器数据库、串行数据库、单处理器系统和单节点或非并行环境。

此环境中的数据库可满足一个部门或小办公室的需要，其中的数据和系统资源（包括一个单处理器或 CPU）由单数据库管理器来管理。

第43页的表3列出了此硬件配置最适合的并行性类型。

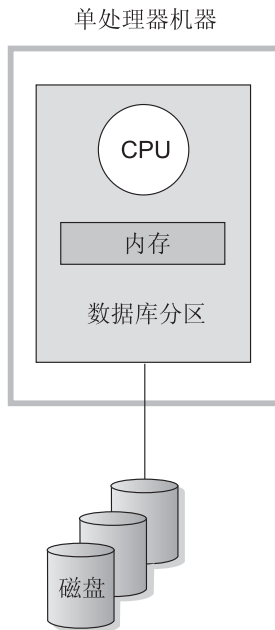


图 17. 单处理器上的单分区

### 容量和可伸缩性

在此环境中可以添加更多的磁盘。如果让每个磁盘拥有一个或多个 I/O 服务器，则多个 I/O 操作可同时执行。

单处理器系统受处理器可处理的磁盘空间容量的限制。无论可以添加的其他组件（如内存或磁盘）如何，随着工作负荷的增加，单个 CPU 可能无法更快地处理用户请求。若已达到最大容量或最大可伸缩性，则可考虑移到一个有多个处理器的单分区系统中。

## 具有多个处理器的单分区

此环境通常由同一台机器中几个能力相等的处理器组成（参见图18），称为对称多处理机（SMP）系统。象磁盘空间和内存这类资源是共享的。

利用可用的多个处理器，可以更快地完成不同的数据库操作。DB2 还可把单个查询的工作分布在可用的处理器中，以提高处理速度。其他数据库操作，如装入数据、备份和复原表空间以及在现存数据上创建索引，都可以利用多个处理器。

第43页的表3列出了此硬件配置最适合的并行性类型。

### SMP 机器

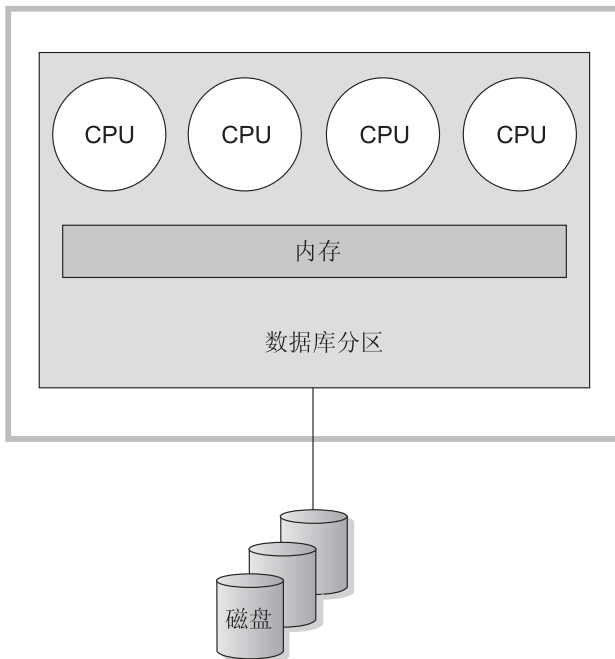


图 18. 单分区数据库对称多处理机系统

### 容量和可伸缩性

在此环境中可以添加更多的处理器。然而，由于不同的处理器可能会试图访问同一数据，所以随着您公司业务量的增加，此环境也可能遇到限制。利用共享内存和共享磁盘，可有效地共享所有数据库数据。

可以通过增加磁盘数来增加与处理器相关的数据库分区的 I/O 容量。可以建立 I/O 服务器以专门处理 I/O 请求。如果让每个磁盘拥有一个或多个 I/O 服务器，则多个 I/O 操作可同时执行。

若已达到最大容量或最大可伸缩性，可考虑移到有多个分区的系统中。

## 多分区配置

可以将一个数据库分布在多个分区中，每个分区在它自己的机器上。可将有多个数据库分区的多台机器编组在一起。本节描述下列分区配置：

- 具有一个处理器的系统上的分区
- 具有多个处理器的系统上的分区
- 逻辑数据库分区

### 具有一个处理器的分区

在此环境中，有许多数据库分区。每个分区都驻留在它自己的机器上，而且它有自己的处理器、内存和磁盘（第40页的图19）。所有机器通过通信设施连接在一起。可使用许多不同的名称来称呼此环境，包括群集、单处理器群集、大规模并行处理（MPP）环境和不共享配置。最后一个名称准确地反映了此环境中的资源安排。与 SMP 环境不同，MPP 环境没有共享的内存或磁盘。MPP 环境消除了由共享内存和磁盘带来的限制。

一个分区数据库环境允许一个数据库保持逻辑上的整体性，但它实际上分布于多个分区中。数据是分区的这一事实对大多数用户是透明的。可以在数据库管理器之间划分工作；每个分区中的每个数据库管理器都只为该数据库中它自己的那部分工作。

第43页的表3列出了此硬件配置最适合的并行性类型。

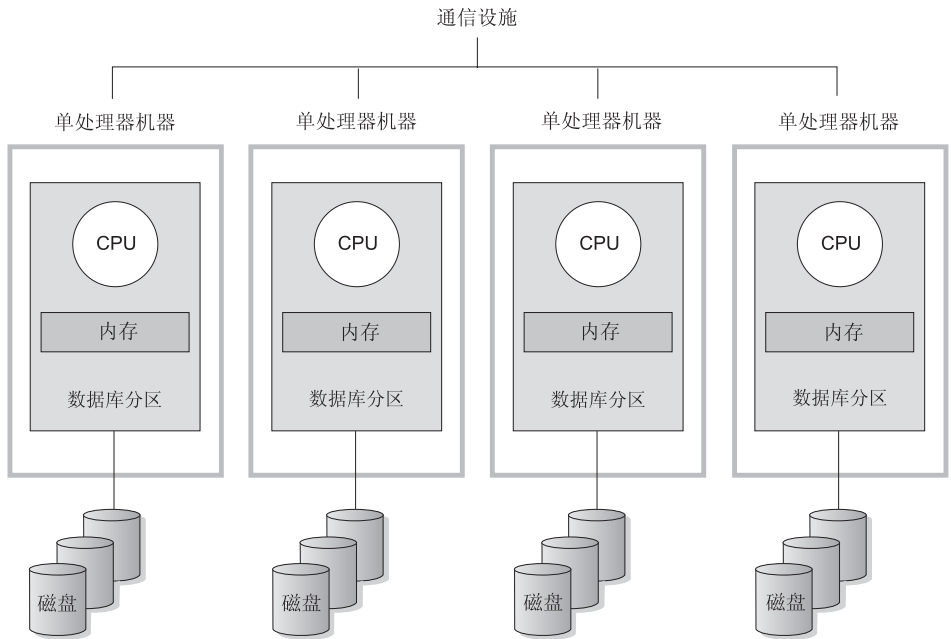


图 19. 大规模并行处理系统

**容量和可伸缩性:** 在此环境中可以向您的配置添加更多的数据库分区（节点）。在某些平台上，例如 RS/6000 SP，最大的节点数为 512。然而，在管理很多机器和实例时，可能存在实际的限制。

若已达到最大容量或最大可伸缩性，可考虑移到每个分区都有多个处理器的系统中。

### 具有多个处理器的分区

每个分区具有单处理器的配置的替代配置是，一个分区具有多个处理器的配置。这称为 *SMP* 群集（第41页的图20）。

此配置结合了 *SMP* 和 *MPP* 并行性的优点。这表示一个查询可以在跨多个处理器的单个分区中执行。亦即一个查询可以用并行方式在多个分区中执行。

第43页的表3列出了此硬件配置最适合的并行性类型。

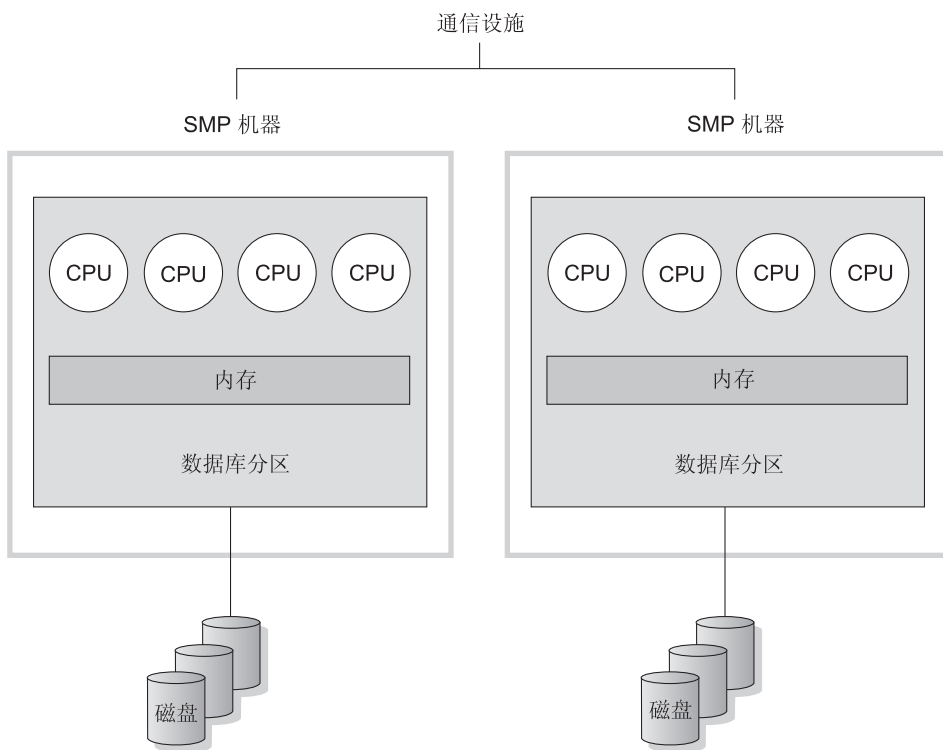


图 20. SMP 群集

**容量和可伸缩性:** 在此环境中，可以添加更多的数据库分区，并可以向现存数据库分区添加更多的处理器。

### 逻辑数据库分区

逻辑数据库分区与物理分区不同之处在于逻辑分区未被授予对整台机器的控制权。虽然机器已共享资源，但是数据库分区不共享资源。处理器是共享的，但磁盘和内存却不共享。

逻辑数据库分区提供了可伸缩性。在多个逻辑分区上运行的多个数据库管理器可以比一个单数据库管理器更能充分利用可用的资源。第42页的图21举例说明通过添加更多的分区可以在一台 SMP 机器上获得更大的可伸缩性；特别是对那些具有许多处理器的机器而言更是如此。通过对数据库分区，可以分别对每个分区进行管理和恢复。

## 大的 SMP 机器

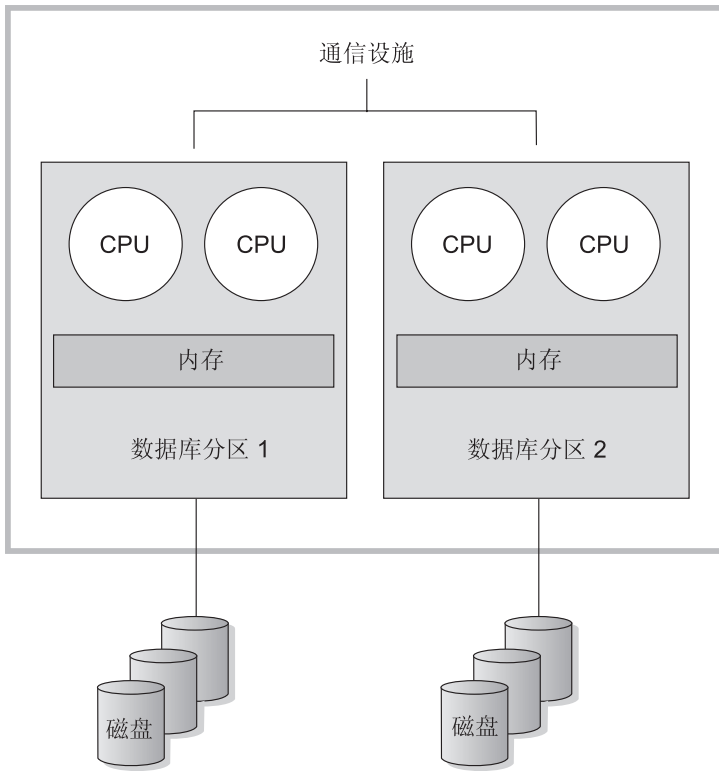


图 21. 分区数据库对称多处理机系统

第43页的图22举例说明可以扩大在图21中显示的配置以增强处理能力的事实。



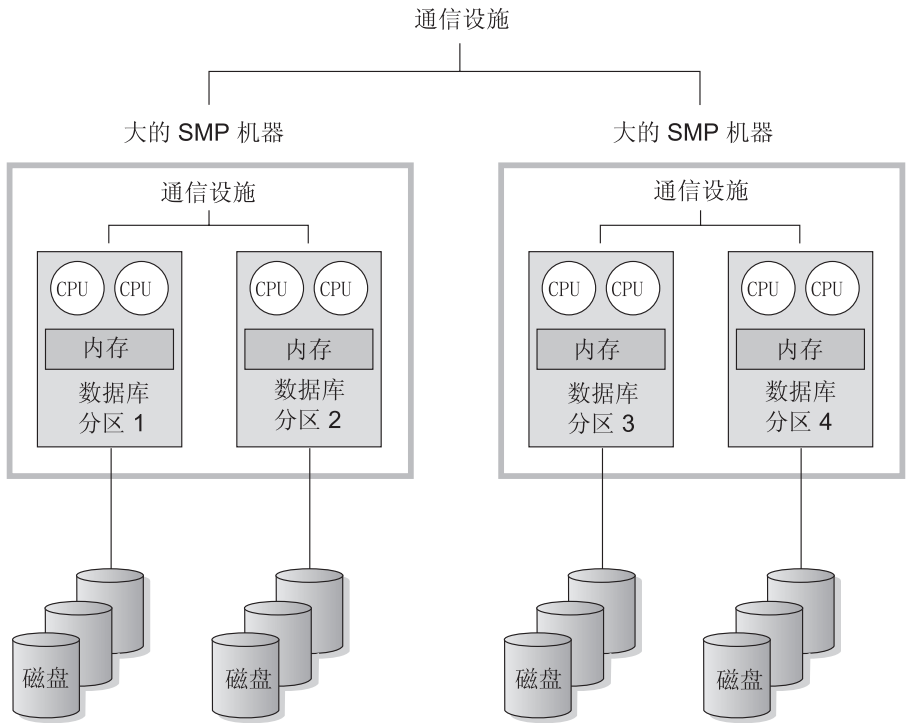


图 22. 分区数据库对称多处理机系统群集在一起

表3列出了此硬件环境最适合的并行性类型。

注：两个以上的分区同时存在于同一台机器上的能力（不考虑处理器的数量），使得可以更灵活地设计高可用性配置和故障恢复策略。机器发生故障之后，可以将一个数据库分区自动移至已包含同一数据库的另一个分区的另一台机器上，然后重新启动该分区。有关更多信息，参见第161页的『第11章 介绍高可用性和故障恢复支持』。

### 最适合每种硬件环境的并行性的摘要

下表总结了最适合于利用各种硬件环境的并行性的类型。

表 3. 每种硬件环境中的可能并行性类型

硬件环境	I/O 并行性	查询内并行性	
		分区内并行性	分区间并行性
单分区，单处理器	是	否 (1)	否
单分区，多个处理器 (SMP)	是	是	否
多分区，一个处理器 (MPP)	是	否 (1)	是

表 3. 每种硬件环境中的可能并行性类型 (续)

硬件环境	I/O 并行性	查询内并行性	
		分区内并行性	分区间并行性
多个分区, 多个处理器 (SMP 群集)	是	是	是
逻辑数据库分区	是	是	是

注: (1) 甚至是在单处理器系统中, 使用一个配置参数将并行度设置为大于 1 的值也可能获得好处, 特别是在执行的查询并未充分利用 CPU 时 (例如, 若它们受 I/O 约束)。

---

## 第5章 关于数据入库

“DB2 通用数据库”提供了“数据仓库中心”，这是一个使数据仓库处理自动化的组件。可使用“数据仓库中心”来定义要包括在仓库中的数据。然后，可使用“数据仓库中心”来自动调度仓库中数据的刷新。

本节提供数据入库和数据入库任务的概述。有关入库的更多详细信息，以及有关使用“数据仓库中心”的信息，参考《数据仓库中心管理指南》和“数据仓库中心”联机帮助。

---

### 什么是数据入库？

包含运行数据 — 运作业务的日常事务的数据 — 的系统包含对业务分析员非常有用的信息。例如，分析员可使用有关特定产品在一年中的特定时间在特定地区的销售信息来了解异常现象或预测将来的销售量。

然而，直接访问运行数据的分析员会遇到几个问题：

- 他们可能没有查询运行数据库的技术。例如，查询 IMS 数据库要求应用程序使用特殊类型的数据操作语言。通常，具有查询运行数据库技术的那些程序员的专职工作是维护数据库及其应用程序。
- 对于许多运行数据库（如银行的数据库），性能非常关键。系统不能处理进行特殊查询的用户。
- 通常，运行数据未处于适合业务分析员使用的最佳格式。例如，与原始数据相比，按产品、地区和季节归纳的销售数据对分析员更有用。

数据入库解决了这些问题。在数据入库中，将存储资料式数据 — 从运行数据中抽取的数据，经过变换之后供最终用户进行决策时使用。例如，数据入库工具能从运行数据库复制所有销售数据，执行计算来对数据进行摘要，然后将摘要数据从运行数据写至单独数据库中的一个目标。最终用户可以查询这个单独的数据库（仓库），而不会影响运行数据库。

下面各节描述您创建和维护数据仓库时将会使用到的对象（主题区、仓库源、仓库目标、代理程序、代理站点、步骤和过程）。

### 主题区

主题区标识与商业的逻辑领域相关的过程，并对其进行分组。例如，若是在构建市场营销和销售数据的仓库，则可以定义一个“销售”主题区和一个“市场营

销”主题区。然后，可以在“销售”主题区下面添加与销售相关的过程。同样，可以在“市场营销”主题区下面添加与市场营销数据相关的定义。

## 仓库源

仓库源标识将要向仓库提供数据的表和文件。“数据仓库中心”使用仓库源中的规范来访问和选择数据。这些源几乎可以是任何与仓库相连接的关系或非关系源（表、视图或文件）。

## 仓库目标

仓库目标是包含特定数据的数据库表或文件，这些数据已经过变换，最终用户可以使用。与仓库源相同，仓库目标也可以向“数据仓库中心”步骤提供数据。

## 仓库代理程序和代理站点

“数据仓库中心”代理程序管理数据源与目标仓库之间的数据流。在 Windows NT、AIX、OS/2、OS/390、OS/400 和 SUN Solaris 操作系统上有代理程序可用。代理程序使用“开放式数据库链接”(ODBC) 驱动程序或 DB2 CLI 来与不同数据库通信。

有好几个代理程序可以处理源与目标仓库之间的数据传送。您使用的代理程序的数目取决于现存连通性配置以及您计划通过仓库移动的数据量。若同时运行多个需要同一代理程序的过程，则可以生成一个代理程序的附加实例。

代理程序可以是本地的或远程的。本地仓库代理程序是与仓库服务器安装在同一台机器上的代理程序。远程仓库代理程序是安装在与仓库服务器相连的另一台机器上的代理程序。

代理站点是安装有代理程序软件的工作站的逻辑名。代理站点名称与 TCP/IP 主机名不同。单一物理机器只能有一个 TCP/IP 主机名。然而，可以在单一机器上定义多个代理站点。每个代理站点都由一个逻辑名标识。

缺省代理站点（名为“缺省 VW 代理站点”）是“数据仓库中心”在初始化仓库控制数据库期间在 Windows NT 上定义的一个本地代理程序。

## 步骤和过程

步骤是“数据仓库中心”中的逻辑实体，它定义：

- 输出表或文件的结构。
- 用于填充输出表或文件的机制（SQL 或程序）。
- 用来填充输出表或文件的调度。

步骤通过使用 SQL 语句或通过调用程序来移动数据和变换数据。运行步骤时，将在仓库源与仓库目标之间传送数据，并对该数据执行任何变换。

过程包含执行变换和移动任务的一系列步骤。通常，一个过程通过从一个或多个仓库源（这可以是数据库表或文件）抽取数据来填充仓库数据库中的仓库目标。然而，也可定义用来启动程序，而未指定任何仓库源或目标的过程。

可以按需要运行步骤，也可以调度步骤，以在设置好的时间运行。可以将步骤调度为只运行一次，也可以将其调度为重复运行，如在每个星期五运行。还可以将步骤调度为按顺序运行，以便当一个步骤完成运行时，下一个步骤开始运行。可以将步骤调度为在另一个步骤（成功或不成功）完成后运行。若调度过程，则该过程中的第一个步骤在调度的时间运行。

当一个步骤或过程运行时，它可以使用下列方法保存数据：

- 用新数据替换仓库目标中的所有数据。
- 将新数据追加至现存数据。
- 追加数据的单独版本。

假设您想要让“数据仓库中心”执行下列任务：

1. 从不同的数据库抽取数据。
2. 将数据转换为单一格式。
3. 将数据写至数据仓库中的一个表。

您应创建一个包含个别步骤的过程。每个步骤执行单独的任务，如从数据库抽取数据，或将其转换为正确的格式。然后，应使用另一步骤来填充目标表，该表包含变换后的数据。

下列各节描述您可以在“数据仓库中心”中找到的各种步骤类型。有关步骤的更多信息，参考《数据仓库中心管理指南》。

## SQL 步骤

SQL 步骤使用 SQL SELECT 语句来从仓库源抽取数据，并生成 INSERT 语句来将该数据插入到仓库目标表中。

## 程序步骤

有许多种类型的程序步骤：“DB2 AS/400 版程序”、“DB2 OS/390 版程序”、“DB2 UDB 版程序”、“Visual Warehouse 5.2 DB2 程序”、“OLAP 服务器程序”、“文件程序”和“复制”。这些步骤运行预定义的程序和实用程序。

## 变换器步骤

变换器步骤是特定的存储过程和用户定义函数，它们指定可以用来变换数据的统计或仓库变换器。可使用变换器来清除、反转和旋转数据；生成主键和周期表；以及计算各种统计信息。

在变换器步骤中，应指定其中一个统计或仓库变换器。运行过程时，该变换器步骤将数据写至一个或多个仓库目标。

## 用户定义程序步骤

用户定义程序步骤是“数据仓库中心”中的一个逻辑实体，它表示您想要让“数据仓库中心”启动的应用程序。仓库代理程序可以启动用户定义程序步骤：

- 在填充仓库目标期间启动。
- 在填充仓库目标之后启动。
- 自行启动。

例如，您可以编写一个执行下列过程的用户定义程序：

1. 从一个表导出数据。
2. 操作该数据。
3. 将该数据写至过渡输出资源或仓库目标。

---

## 入库任务

创建数据仓库包括下列任务：

- 定义一个主题区，它标识仓库中将要使用的过程并对它们进行分组。
- 检测源数据（或运行数据），并定义仓库源。
- 创建要用作仓库的数据库，并定义仓库目标。
- 通过定义一个过程，指定如何移动源数据以及将其变换成适合仓库数据库使用的格式。
- 测试您定义的步骤，并调度它们，以自动运行。
- 通过定义安全性，并监视数据库的使用情况来管理仓库。
- 如果有“DB2 仓库管理器”软件包，则创建仓库中数据的信息目录。信息目录是一个包含商业元数据的数据库。元数据帮助用户标识和找出机构中可供他们使用的数据和信息。仓库的最终用户可搜索目录，以确定要查询哪些表。
- 为仓库中的数据定义星形模式模型。星形模式是一种具有专门用途的设计，它由多个维度表（描述商业的各个方面）和一个事实表（包含关于商业的事实）组成。例如，如果您生产软包装饮料，则一些维度表是产品、市场和时间。事实表包含有关每个地区每个季节订购的产品的事务信息。

- 连接事实表和维表，以将维表中的详细信息与订单信息组合在一起。例如，可以连接产品维与事实表，以将有关如何包装每一种产品的信息添加至订单。

通过使用商业智能教程、查看 *DB2 通用数据库快速导览* 或阅读《数据仓库中心管理指南》，您可以了解更多关于这些任务以及其他任务的详情。





---

## 第6章 关于 Spatial Extender

本节通过说明 Spatial Extender 的用途以及讨论它所处理的数据对它作了介绍。有关使用 Spatial Extender 的详细信息，参考《*Spatial Extender 用户指南和参考*》。

---

### Spatial Extender 的用途

使用 Spatial Extender 来创建地理信息系统 (GIS): 对象、数据和应用程序的一个复合体，它允许您生成和分析关于地理面貌的空间信息。地理面貌包括构成地球表面的物体和占据地球表面的物体。它们组成自然环境（例如，河流、森林、山峰和沙漠）和文化环境（城市、住宅区、办公大楼、路标，等等）。

空间信息包括下列各项:

- 地理面貌相对于其环境的位置（例如，一座城市中医院和门诊部所在的点或城市的居住区与当地地震带的距离）
- 地理面貌相互相关的方式（例如，有关特定河流系统包含在特定地区内的信息，或该地区中的特定桥梁横跨该河流系统的支流的信息）
- 适用于一个或多个地理面貌的度量单位（例如，办公大楼与其红线之间的距离或鸟类保护区周界的长度）

空间信息，无论是其本身还是与传统数据库输出的组合，都可以帮助您设计项目和进行商业和政策决策。例如，假设一个福利区域经理需要核实他所服务的地区中实际生活着多少福利申请人和受领人。Spatial Extender 可根据服务的地区的位置以及申请人和受领人的地址生成此信息。

或者，假设连锁饭店老板想在邻近的城市做生意。为了确定在哪里开新饭店，这位老板需要诸如这样的问题的答案：在这些城市中，将会经常光顾我的饭店的那些类型的人主要集中在什么位置？主高速公路的位置在哪里？哪里的犯罪率最低？我的竞争者的饭店的位置在哪里？Spatial Extender 可生成直观显示的空间信息来回答这样的问题，下层数据库管理系统可生成标签和文本来解释这些显示。

---

### 表示地理面貌的数据

本节提供为了获得空间信息而生成、存储和操作的数据的概述。包括下列主题:

- 数据如何表示地理面貌
- 空间数据的性质
- 生成空间数据的方法

## 数据如何表示地理面貌

在 Spatial Extender 中，地理面貌可以由一个表或视图中的一行表示，也可以由这样的行的一部分表示。例如，考虑下面这两个地理面貌：办公大楼和住宅区。在图23中，BRANCHES 表的每一行都表示一间银行的一个分部。不同的是，CUSTOMERS 表的每一行共同表示银行的一个客户；但每一行的一部分——明确的说，包含客户地址的单元——可被视为表示的是客户的住所。

这些表包含标识和描述银行分部和客户的数据。这样的数据被称为属性数据。

### BRANCHES

ID	NAME	ADDRESS	CITY	STATE	ZIP
937	Airzone-Multern	92467 Airzone Blvd	San Jose	CA	95141

### CUSTOMERS

ID	LAST NAME	FIRST NAME	ADDRESS	CITY	STATE	ZIP	CHECKING	SAVINGS
59-6396	Kriner	Endela	9 Concourt Circle	San Jose	CA	95141	A	A

图 23. 表示地理面貌的表行；其地址数据表示地理面貌的表行。BRANCHES 表中的数据行表示银行的分部。CUSTOMERS 表中地址数据的单元表示客户的住所。

可以将属性数据的子集（表示分部和客户地址的值）转换成用来生成空间信息的值。例如，如图所示，一个分部的地址是 92467 Airzone Blvd., San Jose CA 95141。一个客户的地址是 9 Concourt Circle, San Jose CA 95141。Spatial Extender 可以将这些地址转换成特定的值，这些值指示该分部以及客户的住址相对于其各自环境的位置。图24显示了具有包含这样的值的新列的 BRANCHES 和 CUSTOMERS 表。

### BRANCHES

ID	NAME	ADDRESS	CITY	STATE	ZIP	LOCATION
937	Airzone-Multern	92467 Airzone Blvd	San Jose	CA	95141	

### CUSTOMERS

ID	LAST NAME	FIRST NAME	ADDRESS	CITY	STATE	ZIP	LOCATION	CHECKING	SAVINGS
59-6396	Kriner	Endela	9 Concourt Circle	San Jose	CA	95141		A	A

图 24. 添加了空间列的表。在每个表中，LOCATION 列都将包含与地址相对应的坐标。

当将地址和类似的标识符用作空间信息的起始点时，它们被称为**源数据**。因为起源于它们的值生成空间信息，所以这些生成的值被称为**空间数据**。下一节描述空间数据并介绍其相关联的数据类型。

## 空间数据的性质

许多空间数据由坐标组成。坐标是一个数字，它指示相对于参考点的位置。例如，纬度是指示相对于赤道的位置的坐标。经度是指示相对于格林威治子午线的位置的坐标。因此，黄石国家公园的位置由其纬度（北纬 44.45 度）及其经度（西半球 110.40 度）定义。

纬度、经度、它们的参考点以及其他相关参数共同称为**坐标系**。坐标系基于除纬度和经度外同时存在的值。这些坐标系有它们自己的位置量度标准、参考点和其他相异参数。

最简单的空间数据项由两个坐标组成，这两个坐标定义单个地理面貌的位置。术语**数据项**是指在关系表中占用一个单元的一个或多个值。更大的空间数据项由数个定义线性路径（如道路或河流）的坐标组成。第三种类型的空间数据项由定义区域周界（如小岛或水淹地带的边界）的坐标组成。

每个空间数据项都是空间数据类型的一个实例。标记一个位置的两个坐标的数据类型是 `ST_Point`；定义线性路径的坐标的数据类型是 `ST_LineString`；定义周界的坐标的数据类型是 `ST_Polygon`。这些类型与用于空间数据的其他数据类型一起，是属于单一层次结构的结构化类型。

## 空间数据从哪里来

可以通过三种方法获取空间数据：

- 根据属性数据生成
- 根据其他空间数据生成
- 导入

### 使用属性数据作为源数据

根据属性数据（如地址）生成空间数据称为**地理编码**。第52页的图24显示了用于空间数据的两个列，一个在 `BRANCHES` 表中，另一个在 `CUSTOMERS` 表中。想象 `Spatial Extender` 对这些表中的地址进行地理编码，并将生成的输出（与地址相对应的坐标）放到这两个列中。第54页的图25举例说明了此结果。

## BRANCHES

ID	NAME	ADDRESS	CITY	STATE	ZIP	LOCATION
937	Airzone-Multern	92467 Airzone Blvd	San Jose	CA	95141	1653 3094

## CUSTOMERS

ID	LAST NAME	FIRST NAME	ADDRESS	CITY	STATE	ZIP	LOCATION	CHECKING	SAVINGS
59-6396	Kriner	Endela	9 Concourc Circle	San Jose	CA	95141	953 1527	A	A

图 25. 包括从源数据生成的空间数据的表. CUSTOMERS 表中的 LOCATION 列包含地理编码器根据 ADDRESS、CITY、STATE 和 ZIP 列中的地址生成的坐标。通常，BRANCHES 表中的 LOCATION 列包含地理编码器根据此表的 ADDRESS、CITY、STATE 和 ZIP 列中的地址生成的坐标。

Spatial Extender 使用一个名为 *geocoder* 的函数来对属性数据进行地理编码，并将生成的空间数据放入列中。

### 使用其他空间数据作为源数据

空间数据不仅可以从属性数据生成，还可以从其他空间数据生成。例如，假定在 BRANCHES 表中定义了其分部的银行想知道每个分部附近 5 英里内有多少客户。在银行可以从数据库中获取此信息之前，它必须向数据库提供一个地带的定义，该地带位于每个分部半径 5 英里之内。Spatial Extender 函数 *ST\_Buffer* 可以创建这样的定义。通过使用每个分部的坐标作为输入，此函数可以生成一系列坐标，这些坐标划分出了期望的地带的周界。图26显示了带有 *ST\_Buffer* 提供的信息的 BRANCHES 表。

## BRANCHES

ID	NAME	ADDRESS	CITY	STATE	ZIP	LOCATION	SALES_AREA
937	Airzone-Multern	92467 Airzone Blvd	San Jose	CA	95141	1653 3094	1002 2001, 1192 3564, 2502 3415, 1915 3394, 1002 2001

图 26. 包括从现存空间数据生成的新空间数据的表. SALES\_AREA 列中的坐标是由 *ST\_Buffer* 函数根据 LOCATION 列中的坐标生成的。

除 *ST\_Buffer* 外，Spatial Extender 还提供了几种其他功能，它们可根据现存空间数据生成新的空间数据。

### 导入空间数据

获取空间数据的第三种方法是从 Spatial Extender 支持的其中一种格式的文件中调用它。这些文件包含通常适用于地图的数据：人口普查追踪、水淹地带、地震断

层，等等。通过将这样的数据与您生成的空间数据组合使用，可以增加您可用的地理信息。例如，若公共建设工程部需要确定一个住宅社区的危险程度，可使用 `ST_Buffer` 来在社区周围定义一个地带，然后导入有关水淹地带和地震断层的数据来了解这些问题区与该地带的重叠情况。



---

## 第3部分 数据库设计





---

## 第7章 逻辑数据库设计

本节描述数据库设计的步骤:

- 『决定要记录在数据库中的数据』
- 第61页的『为每种类型的关系定义表』
- 第63页的『为所有表提供列定义』
- 第65页的『标识一列或多列作为主键』
- 第68页的『确保相等的值表示相同的实体』
- 第68页的『考虑使表规范化』
- 第73页的『制定约束实施计划』
- 第78页的『数据库设计的其他注意事项』.

设计数据库的目标是为您的环境建立一种表示, 它易于理解且可以扩充。此外, 您还希望数据库设计有助于维护数据的一致性和完整性。要达到此目的, 可以设计一个减少冗余度并消除在更新数据库期间可能发生的异常的数据。

上述步骤是逻辑数据库设计的一部分。数据库设计不是一个线性过程; 进行该设计时, 可能还会重新执行那些步骤。

数据库设计的物理实现在第81页的『第8章 物理数据库设计』和《管理指南: 实现》中的“实现设计”中有描述。

---

### 决定要记录在数据库中的数据

数据库设计的第一步是标识要存储在数据库表中的数据的类型。一个数据库包括一个机构或企业中的实体及其他它们之间的关系的信息。在关系数据库中, 将实体表示为表。

一个实体是您想要存储其信息的一个人、一个对象或一个概念。样本表中描述的某些实体是雇员、部门和项目。(有关样本数据库的描述, 参考 *SQL Reference*。)

在 *Employee* 样本表中, 实体“雇员”具有属性或特性, 如雇员号、姓名、工作部门和薪水额。那些特性显示为 EMPNO、FIRSTNAME、LASTNAME、WORKDEPT 和 SALARY 列。

实体“雇员”的具体值由一个雇员的所有列中的值组成。每个雇员有一个唯一的雇员号 (EMPNO), 该号码可用来标识实体“雇员”的具体值。一个表中的每一行

表示一个实体或关系的一个具体值。例如，在下表中，第一行中的值描述名为 Haas 的雇员。

表 4. 雇员实体及其属性的具体值

EMPNO	FIRSTNAME	LASTNAME	WORKDEPT	JOB
000010	Christine	Haas	A00	President
000020	Michael	Thompson	B01	Manager
000120	Sean	O'Connell	A00	Clerk
000130	Dolores	Quintana	C01	Analyst
000030	Sally	Kwan	C01	Manager
000140	Heather	Nicholls	C01	Analyst
000170	Masatoshi	Yoshimura	D11	Designer

随着支持非传统数据库应用程序（如多媒体）的需要日益增长。可能要考虑一些属性以便支持多媒体对象，如文档、视频或混合媒体、图像和语音。

在一个表中，某一行的每一列都以某种方式与该行的所有其他列相关。样本表中表达的一些关系如下：

- 雇员被分配至部门
  - Dolores Quintana 被分配至部门 C01
- 雇员从事一个工作
  - Dolores 从事分析员的工作
- 部门要向其他部门报告
  - 部门 C01 向部门 A00 报告
  - 部门 B01 向部门 A00 报告
- 雇员从事某个项目的工作
  - Dolores 和 Heather 都在从事项目 IF1000 的工作
- 雇员管理部门
  - Sally 管理部门 C01。

“雇员”和“部门”是实体；Sally Kwan 是“雇员”的一个具体值的一部分，而 C01 是“部门”的一个具体值的一部分。相同的关系应用于一个表的每一行中的相同列。例如，一个表的某行表示的关系是 Sally Kwan 管理部门 C01；而另一行表示的关系是 Sean O'Connell 是部门 A00 中的职员。

一个表中包含的信息取决于要表示的关系、需要的灵活度和期望的数据检索速度。

除了标识企业中的实体关系外，还需标识其他类型的信息，如应用于该数据的商业规则。

## 为每种类型的关系定义表

可以在数据库中定义几种类型的关系。考虑雇员和部门之间的可能关系。一个雇员只能在一个部门工作；对于雇员，此关系是相异值的。另一方面，一个部门可有许多雇员；对于部门，此关系是多值的。雇员（相异值的）和部门（多值的）之间的关系是一对多的关系。本节讨论下列类型的关系：

- 『一对多和多对一关系』
- 第62页的『多对多关系』
- 第63页的『一对一关系』

### 一对多和多对一关系

要为每个一对多和每个多对一关系定义表：

1. 将关系的“多”方是相同实体的所有关系分组。
2. 为该组中的所有关系定义单个表。

在以下示例中，第一个和第二个关系的“多”方是“雇员”，因此我们定义一个雇员表 EMPLOYEE。

表 5. 多对一关系

实体	关系	实体
雇员	被分配到	部门
雇员	从事	工作
部门	报告给	(管理)部门

在第三个关系中，“部门”是“多”方，因此我们定义一个部门表 DEPARTMENT。

下面显示如何在表中表示这些关系：

EMPLOYEE 表:

EMPNO	WORKDEPT	JOB
000010	A00	President
000020	B01	Manager
000120	A00	Clerk
000130	C01	Analyst
000030	C01	Manager
000140	C01	Analyst
000170	D11	Designer

DEPARTMENT 表:

DEPTNO	ADMRDEPT
C01	A00
D01	A00
D11	D01

## 多对多关系

两个方向都是多值的关系是多对多关系。一个雇员可以开发多个项目，而一个项目可以有多个雇员。问题“Dolores Quintana 在做什么？”和“谁在开发项目 IF1000？”都得到多个答案。在一个表中，对每个实体（“雇员”和“项目”）使用一列，可以表示多对多关系，如下面示例所示。

下面显示如何在一个表中表示多对多关系（一个雇员可以参与多个项目，而一个项目可以有多个雇员在参与）：

雇员活动 (EMP\_ACT) 表:

EMPNO	PROJNO
000030	IF1000
000030	IF2000
000130	IF1000
000140	IF2000
000250	AD3112

## 一对一关系

一对一关系在两个方向都是相异值的。一个经理管理一个部门；一个部门只有一个经理。问题“谁是部门 C01 的经理？”和“Sally Kwan 管理哪个部门？”都有单个答案。可将该关系指定给 DEPARTMENT 表或 EMPLOYEE 表。因为所有部门都有经理，但不是所有雇员都是经理，因此将经理添加至 DEPARTMENT 表是最合乎逻辑的，如以下示例中所示。

下面显示如何在一个表中表示一对一关系：

DEPARTMENT 表：

DEPTNO	MGRNO
A00	000010
B01	000020
D11	000060

---

## 为所有表提供列定义

要在一个关系表中定义一列：

1. 为该列选择一个名称。

一个表的每一列都必须具有对于该表唯一的名称。第169页的『附录A. 命名规则』中详细地描述了如何选择列名。

2. 说明对该列有效的数据类型。

数据类型和长度指定对该列有效的数据类型和最大长度。可以从数据库管理器提供的那些类型中选择数据类型；也可以选择创建自己的用户定义类型。有关 DB2 提供的数据类型和用户定义类型的信息，参考 *SQL Reference*。

数据类型的示例有：数字、字符串、双字节（或图形）字符串、日期时间和二进制字符串。

大对象 (LOB) 数据类型支持多媒体对象，如文档、视频、图像和语音。这些对象是使用下列数据类型实现的：

- 二进制大对象 (BLOB) 字符串。BLOB 的示例是雇员的相片、语音和视频。
- 字符大对象 (CLOB) 字符串，它的字符序列可以是单字节字符或多字节字符，或这两者的组合。CLOB 的一个示例是雇员的简历。
- 双字节字符大对象 (DBCLOB) 字符串，它的字符序列是双字节字符。DBCLOB 的一个示例是日语简历。

要更好地了解大对象支持，参考 *SQL Reference*。

用户定义类型 (UDT)，是从现存类型派生的一种类型。您可能需要定义从现存类型派生并与现存类型共享特性，但仍被视为单独的和 incompatible 的类型。

结构化类型是一种用户定义类型，其结构在数据库中定义。它包含一系列属性，每个属性都有一个数据类型。可将结构化类型定义为称为超类型的另一种结构化类型的子类型。子类型继承它的超类型的所有属性，并可定义附加属性。与一个公共超类型相关的结构化类型集合称为类型层次结构，没有任何超类型的超类型称为该类型层次结构的根类型。

可以将结构化类型用作表或视图的类型。结构化类型的属性的名称和数据类型，与对象标识符一起，成为这个类型表或类型视图的列的名称和数据类型可以将类型表或类型视图的行当作结构化类型的实例。

不能将结构化类型用作表或视图的列的数据类型。也不支持将整个结构化类型的实例作为应用程序中的一个主变量来检索。

引用类型是结构化类型的辅助类型。类似于单值类型，引用类型是一个标量类型，它与一种内置数据类型共享一个公共的表示法。在类型层次结构中的所有类型都共享这同一个表示法。引用类型的表示法是在创建类型层次结构的根类型时定义的。当使用引用类型时，将结构化类型指定为该类型的参数。此参数称为该引用的目标类型。

引用的目标始终是类型表或视图中的行。当使用引用类型时，可能需要定义作用域。作用域标识一个表（称为目标表）或视图（称为目标视图），它包含引用值的目标行。目标表或视图的类型必须与引用类型的目标类型相同。定义了作用域的引用类型的一个实例会唯一地标识类型表或类型视图中的一行，该行称为它的目标行。

由于许多原因，可能会使用用户定义函数 (UDF)，包括调用那些允许在用户定义类型之间比较或转换的例程。UDF 扩展并增强了内部 SQL 函数提供的支持，并可在可使用内置函数的任何地方使用。有两种类型的 UDF:

- 外部函数，它是用一种编程语言编写的
- 有源函数，它将用于调用其他 UDF

例如，两个数字数据类型是“欧洲鞋码”和“美国鞋码”。这两种类型表示的都是鞋码，但是它们不兼容，因为度量基数不同且不能比较。可调用用户定义函数将一种鞋码转换为另一种。

要更好地了解用户定义类型、结构化类型、引用类型和用户定义函数，参考 *SQL Reference*。

### 3. 说明哪些列可能需要缺省值。

某些列不可能在所有行的值都是有意义的，因为：

- 列值不适用于该行。

例如，包含雇员中间姓名首字母的一列不适用于没有中间名首字母的雇员。

- 值适用，但尚未知道该值。

例如，MGRNO 列可能包含无效的经理号，因为该部门的上一个经理已调离，新经理尚未任命。

在这两种情况下，可以选择允许 NULL 值（指示列值是未知的或不适用的特殊值），或允许数据库管理器或应用程序指定的非 NULL 缺省值。

在 *SQL Reference* 中详细描述了 NULL 值和缺省值。

---

## 标识一列或多列作为主键

键是一组可用来标识或访问特定行的列。在表、索引或引用约束的描述中标识键。同一个列可以是多个键的一部分。

唯一键是约束为其任何两个值都不相等的键。唯一键的列不能包含 NULL 值。例如，可将雇员号码列定义为唯一键，因为该列中的每个值只标识唯一一个雇员。不能有相同雇员号码的两个雇员。

用来强制键唯一性的机制称为唯一索引。一个表的唯一索引是一列或若干列的有序集合，而每个值标识（在功能上确定）这些列的唯一行。唯一索引可以包含 NULL 值。

主键是在一个表上定义的唯一键中的一个，而且该键被选为最重要的键。一个表上只能有一个主键。

会自动为主键创建主索引。数据库管理器使用主索引来有效地访问表行，且主索引允许数据库管理器实现主键的唯一性。（也可以在非主键列上定义索引，以便在处理查询时高效率地访问数据。）

若一个表没有“自然”的唯一键，或者到达顺序是用于区分唯一行的方法，则使用时间戳记作为键的一部分可能有帮助。（另见第67页的『定义标识列』。）

某些样本表的主键是：

表	键列
<b>Employee 表</b>	EMPNO
<b>Department 表</b>	DEPTNO
<b>Project 表</b>	PROJNO

以下示例显示 PROJECT 表的一部分，包括其主键列。

表 6. PROJECT 表上的主键

PROJNO (主键)	PROJNAME	DEPTNO
MA2100	Weld Line Automation	D01
MA2110	Weld Line Programming	D11

若一个表中的每一列都包含重复的值，则不能只用一列来定义主键。具有多列的键是组合键。列值的组合应定义一个唯一的实体。若定义组合键不太容易，可以考虑创建具有唯一值的新列。

以下示例显示包含多列的一个主键（组合键）：

表 7. EMP\_ACT 表上的组合主键

EMPNO (主键)	PROJNO (主键)	ACTNO (主键)	EMPTIME	EMSTDATE (主键)
000250	AD3112	60	1.0	1982-01-01
000250	AD3112	60	.5	1982-02-01
000250	AD3112	70	.5	1982-02-01

## 标识入选键列

要标识入选键，选择定义唯一实体的最少数目的列。可能有多个人选键。在第20页的表2中，有多个人选键。EMPNO、PHONENO 和 LASTNAME 这三列都唯一地标识该雇员。

从一组入选键中选择一个主键的标准应是持久性、唯一性和稳定性。

- 持久性表示总是存在每一行的主键值。
- 唯一性表示每一行的键值与所有其他行不同。
- 稳定性表示主键值始终不变。

在该示例的三个人选键中，只有 EMPNO 满足全部这些标准。一个雇员在进入一家公司时，可能没有电话号码。姓可能改变，尽管它们在某个时候可能是唯一的，但不能保证始终如此。雇员号码列是主键的最佳选择。一个雇员只被赋予一次一个唯一的号码，而且只要该雇员在该公司内任职，通常不会更新该号码。因为每个雇员必须有一个号码，因此雇员号码列中的值是持久的。



## 定义标识列

标识列向 DB2 提供了一种为一个表中的每一行自动生成唯一数值的方法。一个表只能有一个定义了标识属性的列。标识列的示例包括订单号码、雇员号码、证券号码和事故号码。

标识列的值可以“始终生成”或“在缺省情况下生成”。

- DB2 保证定义为始终生成的标识列是唯一的。它的值总是由 DB2 生成；不允许应用程序提供显式的值。
- 定义为在缺省情况下生成的标识列向应用程序提供了显式地为标识列提供一个值的方法。若未给出任何值，则 DB2 生成一个，但在此情况下，不能保证该值的唯一性。DB2 只对它生成的那些值保证唯一行。“在缺省情况下生成”用于数据传播（复制现存表的内容），或用于一个表的卸装和重新装入。

标识列特别适合于生成唯一主键值这一任务。应用程序可使用标识列来避免当一个应用程序在数据库外部生成它自己的唯一计数器时可能会导致的并行性和性能问题。例如，一种常见的应用程序级实现是维护一个只有一行的表，它包含一个计数器。每个事务都锁定此表，增大该数字，然后提交；即，每次只有一个事务可以增大计数器。相反，若通过标识列维护该计数器，因为事务不锁定该计数器，所以可以获得更高级别的并行性。一个未提交的已增大该计数器的事务不会阻止后续事务也增大该计数器。

标识列的计数器的增大（或减小）独立于事务。若给定的事务两次增大一个标识计数器，则该事务可能会在生成的两个数字之间看到一个间隙，因为当前可能有其他事务正在增大同一标识计数器（即，将行插入同一个表中）。若一个应用程序必须要有连续范围的数字，则该应用程序应对带有标识列的表进行互斥锁定。因为会造成丢失并行性，所以必须对此决定作权衡。此外，有可能因为生成标识列的的事务已回滚，或因为存放值系列的数据库在指定所有存放的值之前被停用，从而导致给定的标识列出现为数字之间生成有间隙。

标识列生成的序号具有下列附加特性：

- 值可以是任何小数位为零的精确数字数据类型；即，小数位为零的 SMALLINT、INTEGER、BIGINT 或 DECIMAL。（单精度和双精度浮点类型被认为是适当的数字数据类型。）
- 连续值之间可以有任何指定的整数增量。缺省增量是 1。
- 标识列的计数器值是可恢复的。若发生故障，则从日志重构计数器值，因此可以保证继续生成唯一的值。
- 可以将标识列值存入高速缓存，以获得更好的性能。

---

## 确保相等的值表示相同的实体

您可以有多个表，它们描述同一组实体的属性。例如，EMPLOYEE 表显示雇员被分配至的部门的编号，DEPARTMENT 表显示分配到每个部门的经理。要同时检索这两组属性，可用匹配的列将这两个表连接在一起，如以下示例中所示。WORKDEPT 和 DEPTNO 中的值表示相同的实体，并表示 DEPARTMENT 和 EMPLOYEE 表之间的连接路径。

DEPARTMENT 表:

DEPTNO	DEPTNAME	MGRNO	ADMRDEPT
D21	Administration Support	000070	D01

EMPLOYEE 表:

EMPNO	FIRSTNAME	LASTNAME	WORKDEPT	JOB
000250	Daniel	Smith	D21	Clerk

当从多个表检索一个实体的信息时，确保相等的值表示相同的实体。连接列可有不同的名称（象上一个示例中的 WORKDEPT 和 DEPTNO），也可以有相同的名称（象 Department 表和 Project 表中的 DEPTNO 列）。

---

## 考虑使表规范化

规范化帮助消除表数据中的冗余和不一致。它是将表精简为一组列的过程，在这组列中，所有非键列都依赖于主键列。若不是这样，则在更新期间该数据可能变得不一致。

本节简要回顾第一种、第二种、第三种和第四种范式的规则。表的第五种范式在有关数据库设计的许多书籍中都有说明，在此就不再赘述。

### 格式 描述

**第一种** 表中的每一个行位置和每一个列位置均有一个值，永远不会是一组值（参见第69页的『第一种范式』）。

**第二种** 每个不是键一部分的列都依赖于键（参见第69页的『第二种范式』）。

**第三种** 每个非键列都独立于其他非键列，并依赖于键（参见第71页的『第三种范式』）。

第四种 没有行包含有关一个实体的两个或更多个独立的多值事实（参见第72页的『第四种范式』）。

## 第一种范式

若每个单元中都只有一个值，永远不会有一组值，则该表使用的是第一种范式。使用第一种范式的表不必满足更高的范式的标准。

例如，下表违反了第一种范式，因为对于 PART 的每个具体值，WAREHOUSE 列都包含了几个值。

表 8. 违反第一种范式的表

PART (主键)	WAREHOUSE
P0010	Warehouse A, Warehouse B, Warehouse C
P0020	Warehouse B, Warehouse D

以下示例显示使用第一种范式的同一个表。

表 9. 符合第一种范式的表

PART (主键)	WAREHOUSE (主键)	QUANTITY
P0010	Warehouse A	400
P0010	Warehouse B	543
P0010	Warehouse C	329
P0020	Warehouse B	200
P0020	Warehouse D	278

## 第二种范式

若每个不是键一部分的列都依赖于整个键，则该表使用的是第二种范式。

当一个非键列依赖于组合键的一部分时，违反了第二种范式，如以下示例所示：

表 10. 违反第二种范式的表

PART (主键)	WAREHOUSE (主键)	QUANTITY	WAREHOUSE_ADDRESS (主键)
P0010	Warehouse A	400	1608 New Field Road
P0010	Warehouse B	543	4141 Greenway Drive
P0010	Warehouse C	329	171 Pine Lane
P0020	Warehouse B	200	4141 Greenway Drive
P0020	Warehouse D	278	800 Massey Street

主键是一个组合键，它由 PART 和 WAREHOUSE 列组成。因为 WAREHOUSE\_ADDRESS 列只依赖于 WAREHOUSE 的值，因此该表违反了第二种范式的规则。

此设计存在下列问题：

- 对于该仓库中存储的一个部件，在每个记录中重复了该仓库地址。
- 若一个仓库的地址变更，则必须更新引用存储在该仓库中的一个部件的每一行。
- 由于存在这种冗余，该数据可能变得不一致，表现为不同的记录对相同的仓库显示不同的地址。
- 若某个时候一个仓库中没有存储部件，就可能没有哪一行记录了该仓库地址。

解决方案是将该表分割成下面两个表：

表 11. 符合第二种范式的 PART\_STOCK 表

PART (主键)	WAREHOUSE (主键)	QUANTITY
P0010	Warehouse A	400
P0010	Warehouse B	543
P0010	Warehouse C	329
P0020	Warehouse B	200
P0020	Warehouse D	278

表 12. 符合第二种范式的 WAREHOUSE 表

WAREHOUSE (主键)	WAREHOUSE_ADDRESS
Warehouse A	1608 New Field Road
Warehouse B	4141 Greenway Drive
Warehouse C	171 Pine Lane
Warehouse D	800 Massey Street

若有两个表使用第二种范式，就需要考虑性能。根据部件位置生成报告的应用程序必须将这两个表连接起来以检索相关信息。

要更好地了解性能注意事项，参考《管理指南：性能》中的“调整应用程序性能”。

### 第三种范式

若每个非键列都独立于其他非键列，且只依赖于键，则该表使用的是第三种范式。

以下示例中的第一个表包含 EMPNO 列和 WORKDEPT 列。假定添加了 DEPTNAME 列（参见表14）。这个新列依赖于 WORKDEPT，但主键是 EMPNO。该表现在违反了第三种范式。更改雇员 John Parker 的 DEPTNAME 不会更改该部门中其他雇员的部门名。注意，现在部门号 E11 中有两个不同的部门名。产生的不一致显示在该表的更新版本中。

表 13. 更新前的不规范 EMPLOYEE\_DEPARTMENT 表

EMPNO (主键)	FIRSTNAME	LASTNAME	WORKDEPT	DEPTNAME
000290	John	Parker	E11	Operations
000320	Ramlal	Mehta	E21	Software Support
000310	Maude	Setright	E11	Operations

表 14. 更新后的不规范 EMPLOYEE\_DEPARTMENT 表. 表中的信息变得不一致。

EMPNO (主键)	FIRSTNAME	LASTNAME	WORKDEPT	DEPTNAME
000290	John	Parker	E11	Installation Mgmt
000320	Ramlal	Mehta	E21	Software Support
000310	Maude	Setright	E11	Operations

可创建一个含有 WORKDEPT 列和 DEPTNAME 列的新表，将该表规范化。诸如更改部门名之类的更新现在更加简单；只需要更新这个新表。

返回部门名和雇员名的 SQL 查询编写起来更复杂，因为它需要连接两个表。它运行的时间可能比对单个表查询的时间长。因为 WORKDEPT 列必须出现在两个表中，所以需要附加的存储空间。

下面的表被定义为规范化的结果。

表 15. 将 EMPLOYEE\_DEPARTMENT 表规范化之后的 EMPLOYEE 表

EMPNO (主键)	FIRSTNAME	LASTNAME	WORKDEPT
000290	John	Parker	E11

表 15. 将 *EMPLOYEE\_DEPARTMENT* 表规范化之后的 *EMPLOYEE* 表 (续)

EMPNO (主键)	FIRSTNAME	LASTNAME	WORKDEPT
000320	Ramlal	Mehta	E21
000310	Maude	Setright	E11

表 16. 将 *EMPLOYEE\_DEPARTMENT* 表规范化之后的 *DEPARTMENT* 表

DEPTNO (主键)	DEPTNAME
E11	Operations
E21	Software Support

## 第四种范式

若没有一行包含有关一个实体的两个或更多个独立的多值事实，则该表使用的是第四种范式。

考虑下列实体：雇员、技能和语言。一个雇员可有几种技能，并通晓几种语言。有两个关系，一个关系是雇员和技能之间的关系，另一个关系是雇员和语言之间的关系。若一个表表示了这两种关系，则它未使用第四种范式，如以下示例所示：

表 17. 违反第四种范式的表

EMPNO (主键)	SKILL (主键)	LANGUAGE (主键)
000130	Data Modelling	English
000130	Database Design	English
000130	Application Design	English
000130	Data Modelling	Spanish
000130	Database Design	Spanish
000130	Application Design	Spanish

应改用两个表表示这种关系：

表 18. 符合第四种范式的 *EMPLOYEE\_SKILL* 表

EMPNO (主键)	SKILL (主键)
000130	Data Modelling
000130	Database Design
000130	Application Design

表 19. 符合第四种范式的 EMPLOYEE\_LANGUAGE 表

EMPNO (主键)	LANGUAGE (主键)
000130	English
000130	Spanish

但是，若这些属性是独立的（即，雇员只将特定的语言应用于特定的技能），则不应将该表分割。

设计数据库的一个良好策略是将所有数据安排在使用第四种范式的表中，然后决定该结果是否提供了一个可接受的性能级别。如果没有，则可将数据重新安排在使用第三种范式的表中，然后重新评价性能。

---

## 制定约束实施计划

约束是数据库管理器实施的规则。本节讨论四种类型的约束处理：

唯一约束	确保表中的键值是唯一的。检查对组成主键的列的任何更改，以保证唯一性。
引用完整性	在插入、更新和删除操作上实现引用约束。所有外键的所有值都有效才是一个数据库的正确状态。
表检查约束	验证更改后的数据有无违反当创建或更改表时指定的条件。
触发器	定义要执行的一组操作，当对指定的表执行更新、删除或插入操作时要调用这组操作。

### 唯一约束

唯一约束是一个规则，它确保键值在表中是唯一的。在唯一约束中组成该键的每一列必须定义为 NOT NULL。在使用 PRIMARY KEY 子句或 UNIQUE 子句的 CREATE TABLE 或 ALTER TABLE 语句中定义唯一约束。

一个表可有任意多个唯一约束；但是，只能定义一个唯一约束作为一个表的主键。此外，一个表在相同的一组列上不能有多个唯一约束。

当定义一个唯一约束时，数据库管理器就会创建一个唯一索引（如果需要的话），并将它指定为系统必需的主索引或唯一索引。此约束通过唯一索引来实现。一旦在某列上建立了唯一约束，则在多行更新期间对唯一性的检查将延迟到更新结束后进行。

唯一约束也可用作引用约束中的父键。

## 引用完整性

数据库管理器通过引用约束维护引用完整性，引用完整性要求给定属性或表列的所有值也存在于某些其他表或列中。例如，引用约束可能要求 EMPLOYEE 表中的每个雇员属于 DEPARTMENT 表中的一个部门。不能有雇员属于不存在的部门。

可在一个数据库中构建引用约束，以确保维持引用完整性，并允许优化器通过了解这些特殊关系使查询的执行更有效率。当制定引用完整性计划时，标识数据库表之间的所有关系。可定义一个主键和引用约束来标识一个关系。

考虑相关的下列表：

表 20. DEPARTMENT 表

DEPTNO (主键)	DEPTNAME	MGRNO
A00	Spiffy Computer Service Div.	000010
B01	Planning	000020
C01	Information Center	000030
D11	Manufacturing Systems	000060

表 21. EMPLOYEE 表

EMPNO (主键)	FIRSTNAME	LASTNAME	WORKDEPT (外键)	PHONENO
000010	Christine	Haas	A00	3978
000030	Sally	Kwan	C01	4738
000060	Irving	Stern	D11	6423
000120	Sean	O'Connell	A00	2167
000140	Heather	Nicholls	C01	1793
000170	Masatoshi	Yoshimura	D11	2890

下面许多概念的讨论都涉及到这些表（这些概念对于您理解引用完整性很有帮助）。

唯一键是值在所有行中都不重复的一个列或一组列。可将一个唯一键定义为表的主键。当唯一键被一个外键引用时，也称为父键。

主键是属于表定义一部分的唯一键。每个表只能有一个主键。在前面的表中，DEPTNO 和 EMPNO 分别是 DEPARTMENT 表和 EMPLOYEE 表的主键。



外键是一个表中的某一列或一组列，它们引用同一个表或另一个表的唯一键或主键。外键用于建立与唯一键或主键的关系，以便在所有表中实现引用完整性。EMPLOYEE 表中的 WORKDEPT 列是外键，因为它引用 DEPARTMENT 表中的 DEPTNO 主键。

组合键是有多列的键。主键和外键均可以是组合键。例如，若部门是由分部号和部门号的组合来唯一地标识，则需要两个列来创建 DEPARTMENT 表的键。

父键是一个引用约束的主键或唯一键。主键约束是未指定一组父键列时引用约束的缺省父键。

父表是包含一个父键的表，该父键与同一个表或另一个表中的至少一个外键相关。一个表可以是任意多个关系中的父代。例如，具有主键 DEPTNO 的 DEPARTMENT 表是包含外键 WORKDEPT 的 EMPLOYEE 表的父代。

父行是父表中的一行，该表的父键值匹配一个从属表中的至少一个外键值。一个父表中的行不必是父行。DEPARTMENT 表中的第四行 (D11) 是 EMPLOYEE 表中的第三行和第六行的父行。DEPARTMENT 表的第二行 (B01) 不是任何其他行的父行。

从属表是包含一个或多个外键的一个表。从属表也可以是父表。一个表可以是任意多个关系中的一个从属表。EMPLOYEE 表包含外键 WORKDEPT，该表从属于主键为 DEPTNO 的 DEPARTMENT 表。

从属行是一个从属表中的一行，该表有一个与父键值相匹配的非 NULL 外键值。该外键值表示从属行对父行的引用。因为外键可以接受 NULL 值，因此一个从属表中的一行不一定是从属行。

若一个表是从属表，或者它是从属表的子代，则该表就是一个表的子代。子表包含可以追溯到某个表的父键的外键。

参考循环是将一个表与它自身连接的一个路径。当一个表与它自己直接连接时，它是自参考表。若 EMPLOYEE 表有另一列 MGRID，它包含每个雇员的经理的 EMPNO，则 EMPLOYEE 表就是一个自参考表。MGRID 就是 EMPLOYEE 表的外键。

自参考表在同一个关系中既是父表，又是从属表。自参考行既是它本身的父行，又是它本身的从属行。在这种情况下存在的约束称为自引用约束。例如，若在一个自参考表某行中的外键值与该行中唯一键的值匹配，则该行是自参考行。

引用约束是指一个指定的外键的非 NULL 值仅当作为指定表的唯一键值出现时才有效。引用约束的目的是保证数据库关系得以维持，并遵守数据输入规则。

## SQL 操作的意义

对于取决于该表是父表还是从属表的某些 SQL 操作，引用约束的实现有特殊的意义。本节描述维护引用完整性对 INSERT、DELETE、UPDATE 和 DROP 等 SQL 操作的影响。

DB2 不会自动在整个系统中实施引用约束。因此，若您想要在整个系统中实施引用约束，应用程序必须包含必要的逻辑。

讨论下列主题:

- 『INSERT 规则』
- 『DELETE 规则』
- 第77页的『UPDATE 规则』。

**INSERT 规则:** 可以随时在父表中插入一行，而不必在从属表中执行任何操作。但是，除非外键值是 NULL，否则只能在从属表中插入一行（若在父表中有一行的父键值等于要插入的行的外键值）。若一个组合外键值的所有组件都为 NULL，则该值也是 NULL。

当指定一个外键时，此规则是隐式的。

若尝试将一行插入有引用约束的表中，且在父键中不存在任何非 NULL 外键值，则不允许 INSERT 操作。若在试图插入多行期间某行的 INSERT 操作失败，则不插入任何行。

**DELETE 规则:** 当从一个父表中删除一行时，DB2 检查在从属表中是否有任何从属行具有匹配的外键值。若找到这样的从属行，则可执行几项操作。当创建从属表时，通过指定删除规则，可确定将执行哪些操作。

当删除主键时从属表（包含该外键的表）的删除规则是:

- |                  |  |
|------------------|--|
| <b>RESTRICT</b>  | 若找到任何从属行，则阻止删除父表中的任何行。若需要除去父行和从属行，则首先删除从属行。首先删除父行会违反引用约束，这是不允许的。 |
| <b>NO ACTION</b> | 当应用了所有引用约束后，对于每个子行强制使其父行存在。                                      |
| <b>CASCADE</b>   | 表示删除父表中的一行将自动删除从属表中的任何相关行。若从属表中的一行在父表中没有父行存在时它本身也没有意义，则可使用此规则。   |

首先删除父行将自动删除引用主键的从属行。无需首先删除从属行。若某些从属行有它们自己的从属行，则对那些关系应用删除规则。DB2 管理级联删除。

## SET NULL

确保删除父表中某行时将任何从属行中外键的值设置为 NULL。该行的其他部分保持不变。

若当创建表时没有明确定义删除规则，则应用 NO ACTION 规则。

在一个删除操作中可能涉及的任何表称为删除连接的表。下列限制应用于删除连接的关系。

- 在多个表组成的一个参考循环中，一个表不能与它自己是删除连接关系。
- 当一个表通过多个从属关系与另一个表形成删除连接关系时，这些关系必须有相同的删除规则（CASCADE 或 NO ACTION）。
- 当一个自参考表是 CASCADE 关系中另一个表的从属表时，自参考关系的删除规则也必须是 CASCADE。

可以随时从一个从属表中删除行，而不用对父表执行任何操作。例如，在部门-雇员关系中，一个雇员可能退休，要从雇员表中删除表示他的那一行，但不影响部门表。（在雇员 - 部门的反向关系中，部门经理标识是引用雇员表的父键的外键。若一个经理退休，则会影响部门表。）

**UPDATE 规则：** DB2 不允许更新父行的唯一键。当更新一个从属表中的外键，而该外键不为 NULL 时，它必须与该关系中的父表的父键的某个值相匹配。若 UPDATE 操作违反任何引用约束，则发生错误，且不更新任何行。

当更新父键某列中的值时：

- 若从属表中的任何行与该键的原始值匹配，当更新规则是 RESTRICT 时拒绝该更新。
- 若更新语句完成时（触发后的情况除外）从属表中的任何行没有对应的父键，当更新规则是 NO ACTION 时拒绝该更新。

要更新父行中父键的值，必须通过下列操作首先除去与从属表中任何子行的关系：

- 删除子行；或，
- 更新从属表中的外键，以包括另一个有效的键值。

若与该行中的键值没有从属关系，该行不再是参考关系中的父行，且可更新。

若一个外键的一部分要更新，而该外键值的任何一部分都不为 NULL，则该外键的新值必须显示为父表中的唯一键值。若没有外键从属于给定的唯一键；即，包含该唯一键的行不是父行，则可更新该唯一键的某部分。但是，在这种情况下，不能选择多行来进行更新，因为使用的是唯一键，不允许有重复行。

## 表检查约束

在设计中给出的商业规则可通过表检查约束来实施。表检查约束指定应用于一个表的每一行的搜索条件。当对表应用更新或插入语句时，自动激活这些约束。它们是通过 CREATE TABLE 或 ALTER TABLE 语句定义的。

可使用表检查约束来验证。例如，一个部门号的值必须在范围 10 至 100 之间；一个雇员的职务只能是 "Sales"、"Manager" 或 "Clerk"；或者，在该公司工作超过八年的雇员的收入必须超过 \$40,500。

有关表检查约束对 IMPORT 和 LOAD 命令的影响的信息，参考 *Data Movement Utilities Guide and Reference*。

## 触发器

触发器是每当对指定的表执行删除、插入或更新操作时执行的一组定义的操作。可以定义触发器来帮助支持商业规则。还可使用触发器来自动更新摘要或审计数据。因为触发器存储在数据库中，所以不必在每个应用程序中编写操作码。触发器只编码一次并存储在数据库中，当应用程序使用该数据库时，会在需要时由 DB2 自动调用它。这确保始终实施与该数据相关的商业规则。若商业规则更改，则只需修改触发器。

可在一个已触发的 SQL 语句内调用用户定义函数 (UDF)。这样，当该触发器被触发时，被触发的操作执行非 SQL 操作。例如，可将电子邮件作为警告机制发送。有关触发器的更多信息，参见《管理指南：实现》中的“创建触发器”并参考 *Application Development Guide*。

---

## 数据库设计的其他注意事项

当设计一个数据库时，重要的是考虑用户应该能够访问哪些表。通过授权，授予或撤消对表的访问权。最高权限级别是系统管理权限 (SYSADM)。具有 SYSADM 权限的用户可指定其他授权，包括数据库管理员权限 (DBADM)。

在您的设计中，您可能要考虑其他问题，如审计活动、历史数据、摘要表、安全性、确定数据的类型和并行处理能力。

为了进行审计，可能必须在指定的时间段内记录每次对数据所做的更新。例如，您可能希望在每次雇员工资改变时更新和审计表。若定义了适当的触发器，则可

自动更新此表。也可通过 DB2 审计设施执行审计活动。有关更多信息，参见《管理指南：实现》中的“审计 DB2 活动”。

考虑到性能方面的原因，可能只想访问选择的数据量，同时将基本数据作为历史维护。应在设计中加入对维护此历史数据的要求，如数据可以保留几个月或几年才可以清除。

您可能还想使用摘要信息。例如，您可能有一个表，它具有所有雇员的信息。但是，您希望将此信息按分部或部门划分为单独的表。在这种情况下，以原表中的数据为基础的每个分部或部门的摘要表将对您有帮助。有关摘要表的更多信息，参见《管理指南：实现》中的“创建摘要表”。

在设计中还应标识隐含的安全性考虑事项。例如，您可能决定通过安全性表支持用户访问特定类型的数据。可定义各种数据的访问级别以及谁可访问此数据。机密数据，如雇员和工资单数据，将具有严格的安全性限制。有关安全性和权限的更多信息，参见《管理指南：实现》中的“控制数据库访问”。

可以创建具有相关的结构化类型的表。利用这种类型表，可以使用表之间定义的关系建立一个层次结构，称为类型层次结构。该类型层次结构由单个的根类型、超类型和子类型组成。

引用类型表示法是在创建类型层次结构的根类型时定义的。引用的目标始终是类型表或视图中的行。

有关实现包括类型行和表的设计的更多信息，参见《管理指南：实现》中的“实现设计”。有关在处于层次结构的类型表之间移动数据的信息，参考 *Data Movement Utilities Guide and Reference*。

随着业务扩展，可能需要提高“DB2 扩充企业版”提供的容量和性能。在此环境中，数据库分区分布在几台机器或几个系统中，分别负责整个数据库的部分存储和检索。每个分区（或节点）并行工作以处理 SQL 或实用程序操作。

本书包括与并行操作相关的问题和注意事项。



---

## 第8章 物理数据库设计

在完成逻辑数据库设计（参见第59页的『第7章 逻辑数据库设计』）之后，应当考虑关于数据库和表将驻留其中的物理环境的诸多因素。这包括了解为支持和管理数据库而要创建的文件、了解存储数据需要多少空间，以及确定如何使用存储数据所需的表空间。

包括下列主题：

- 『数据库目录』
- 第83页的『估计表的空间需求』
- 第90页的『附加空间需求』
- 第91页的『设计节点组』
- 第97页的『设计和选择表空间』
- 第115页的『联合数据库设计注意事项』

---

### 数据库目录

当创建一个数据库时，DB2 会创建一个单独的子目录来存储控制文件（如日志头文件）并将容器分配给缺省表空间。与该数据库相关的对象并非始终存储在该数据库目录中；它们可以存储在不同的位置（包括设备）中。

该数据库是在 DB2INSTANCE 环境变量定义的实例中创建的，或是在已显式连接（使用 ATTACH 命令）的实例中创建的。有关实例的介绍，参见《管理指南：实现》中的“使用数据库管理器的多个实例”。

在基于 UNIX 的系统上使用的命名模式是：

```
specified_path/$DB2INSTANCE/NODEnnnn/SQL00001
```

在 OS/2 和 Windows 操作系统上使用的命名模式：

```
D:\$DB2INSTANCE\NODEnnnn\SQL00001
```

其中

- specified\_path 是用来安装该实例的可选的、用户指定的位置。
- NODEnnnn 是分区数据库环境中的节点标识符。第一个节点是 NODE0000。
- "D:" 是标识根目录所在的卷的“盘符”。

SQL00001 包含与创建的第一个数据库相关的对象，并给予后续数据库更大的数字：SQL00002，依此类推。

子目录在与数据库管理器实例同名的目录中创建，该实例是在创建数据库时连接的。（在 OS/2 和 Windows 操作系统上，子目录在由“盘符”标识的卷的根目录下创建。）这些实例和数据库子目录在 CREATE DATABASE 命令指定的路径中创建，且数据库管理器自动维护它们。每个实例可能由一个实例所有者拥有（这取决于您使用的平台），该所有者对属于该实例的数据库具有系统管理员 (SYSADM) 权限。

要避免潜在的问题，不要创建使用相同命名模式的目录，不要操纵已经由数据库管理器创建的目录。

## 数据库文件

与数据库相关的文件包括：

文件名	描述
-----	----

<b>SQLDBCON</b>	此文件存储数据库的调整参数和标志。有关更改数据库配置参数的信息，参考《管理指南：性能》。
-----------------	--

<b>SQLOGCTL.LFH</b>	此文件有助于跟踪和控制所有数据库日志文件。
---------------------	-----------------------

<b>Syyyyyyy.LOG</b>	数据库日志文件，编号从 0000000 至 9999999。这些文件的编号由 <i>logprimary</i> 和 <i>logsecond</i> 数据库配置参数控制。个别文件的大小由 <i>logfilsiz</i> 数据库配置参数控制。
---------------------	--

通过使用循环记录，可重新使用这些文件，且保持相同的编号。通过使用归档记录，当将日志归档并分配新日志时，文件编号将依次增加。当达到 9999999 时，该编号绕回。

缺省情况下，这些日志文件存储在 SQLOGDIR 目录中。SQLOGDIR 在 SQLnnnnn 子目录中。

<b>SQLINSLK</b>	此文件帮助确保数据库只由数据库管理器的一个实例使用。
-----------------	----------------------------

<b>SQLTMLPK</b>	此文件帮助确保数据库只由数据库管理器的一个实例使用。
-----------------	----------------------------

<b>SQLSPCS.1</b>	此文件包含该数据库中所有表空间的定义和当前状态。
------------------	--------------------------

<b>SQLSPCS.2</b>	此文件是 SQLSPCS.1 的副本。若没有这两个文件中的任何一个，则不能访问您的数据库。
------------------	---

<b>SQLBP.1</b>	此文件包含数据库中所使用的所有缓冲池的定义。
----------------	------------------------



**SQLBP.2** 此文件是 SQLBP.1 的副本。若没有这两个文件中的任何一个，则不能访问您的数据库。

### **DB2RHIST.ASC**

此文件是数据库历史文件。它保存对数据库执行的管理操作的历史记录，如备份和复原操作。

### **DB2RHIST.BAK**

此文件是 DB2RHIST.ASC 的副本。

注:

1. 不要直接更改这些文件。只能使用记载的 API 和实现那些 API 的工具（包括命令行处理器和 控制中心）来间接访问它们。
2. 不要移动这些文件。
3. 不要除去这些文件。
4. 备份数据库或表空间唯一受支持的方法是通过 **sqlubkp**（备份数据库）API 来进行，包括该 API 的命令行处理器和 控制中心实现。

---

## 估计表的空间需求

数据库对象的大小估计不可能做到很精确。磁盘碎片、自由空间以及使用变长列所造成的开销都使大小估计变得十分困难，因为可能的列类型和列长度的范围实在是太广了。在最初估计数据库大小后，创建一个测试数据库，并用有代表性的数据填充它。

在“控制中心”中，可访问许多专门辅助您确定各种数据库对象的大小需求的实用程序:

- 可选择一个对象，然后使用“估计大小”实用程序。此实用程序可告诉您现存对象（例如，表）的当前大小。在更改该对象之后，该实用程序可以对该对象计算出新的估计值。该实用程序可以帮助您估计存储器需求（将未来的增长考虑在内）。它不仅仅是只能估计对象的大小。它还提供对象可能的大小范围：最小大小（基于当前值）和可能的最大大小。
- 可通过使用“显示相关的”窗口来确定对象之间的关系。
- 可选择实例中的任何一个数据库对象，然后请求“生成 DLL”。此功能使用 **db2look** 实用程序来生成数据库的数据定义语句。有关此实用程序的信息，参考 *Command Reference*。

在这两种情况下，可使用“显示 SQL”或“显示命令”按钮。也可将生成的 SQL 语句或命令作为脚本文件保存，以便稍后使用。所有这些实用程序都有联机帮助可向您提供帮助。

在制订物理数据库需求计划时，应考虑这些实用程序。

估计数据库的大小时，必须考虑下列各项的影响：

- 『系统目录表』
- 『用户表数据』
- 第86页的『长整数字段数据』
- 第86页的『大对象 (LOB) 数据』
- 第87页的『索引空间』

不讨论与下列各项相关的空间需求：

- 本地数据库目录文件
- 系统数据库目录文件
- 操作系统所需的文件管理开销，包括：
  - 文件块大小
  - 目录控制空间

## 系统目录表

创建数据库时，会创建系统目录表。当将数据库对象和特权添加至该数据库时，这些系统表将增大。最初，这些系统表使用大约 3.5 MB 的磁盘空间。

为目录表分配的空间容量取决于表空间的类型和包含这些目录表的表空间的数据块大小。例如，若使用数据块大小为 32 的 DMS 表空间，则目录表空间最初将被分配 20MB 的空间。有关更多信息，参见第97页的『设计和选择表空间』。

**注：**对于含多个分区的数据库，目录表只驻留在发出 CREATE DATABASE 命令的分区上。目录表的磁盘空间仅对于该分区才是必需的。

## 用户表数据

在缺省情况下，表数据存储在 4KB 页上。每一页（不管页大小如何）都包含 76 个字节的数据库管理器 开销。这留下了 4020 字节来存放用户数据（即行），虽然 4 KB 页上的行的长度不能超过 4005 字节。一行不会横跨多页。当使用 4KB 的页大小时，最多可有 500 列。

表数据页不包含用 LONG VARCHAR、LONG VARGRAPHIC、BLOB、CLOB 或 DBCLOB 数据类型定义的列的数据。但是，一个表数据页中的行的确包含这些列的描述符。（有关估计确实包含这些数据类型的表对象的空间所需的信息，参见第86页的『长整数字段数据』和第86页的『大对象 (LOB) 数据』。）

通常，以“首个合适”次序将行插入到表中。（使用自由空间映射）搜索文件，查找大小足以存放该新行的第一个可用空间。当更新一行时，除非该页上所剩的空间不足以包含它，否则在原位置进行更新。若所剩空间不足以包含新行，则在原始行位置创建一个记录，以指向更新后的行在表文件中的新位置。

若调用 `ALTER TABLE APPEND ON` 语句，则将一直追加数据，且不保留关于数据页上任何自由空间的信息。有关此语句的更多信息，参考 *SQL Reference*。

对于数据库中的每个用户表，可以通过如下计算公式来估计 4KB 页数：

```
ROUND DOWN(4020/(平均行大小 + 10)) = records_per_page
```

然后，将结果插入：

```
(number_of_records/records_per_page) * 1.1 = number_of_pages
```

其中，平均行大小是平均列大小的总和（有关每一列的大小的信息，参考 *SQL Reference* 中的 `CREATE TABLE` 语句。），而因子“1.1”表示开销。

**注：**此公式只是提供一个估计值。若记录长度因碎片和溢出记录而改变，则估计的准确性将降低。

也可以选择创建具有 8 KB、16 KB 或 32 KB 页大小的缓冲池或表空间。在特定大小的表空间中创建的所有表都将具有匹配的页大小。假设使用 32 KB 的页大小，则单个表或索引对象的最大大小可达 512 GB。当使用 8 KB、16 KB 或 32 KB 的页大小时，最多可有 1012 列。对于 4 KB 的页大小，最大列数为 500。最大行宽也随页大小的不同而不同：

- 当页大小是 4 KB 时，行长度最大可为 4005 字节。
- 当页大小是 8 KB 时，行长度最大可为 8101 字节。
- 当页大小是 16 KB 时，行长度最大可为 16 293 字节。
- 当页大小是 32 KB 时，行长度最大可为 32 677 字节。

拥有更大的页大小有助于减小任何索引中的级别数。若使用执行随机行读写的 OLTP（联机事务处理）应用程序，则小一点的页大小会更好，这样，因意外的行而浪费的缓冲区空间更少。若使用一次访问大量连续行的 DSS（决策支持系统）应用程序，则大一点的页大小会更好，这样可以减少读取特定行数所需的 I/O 请求数。当行大小小于页大小除以 255 的值时，会发生异常。在这种情况下，每个页上都有浪费的空间。（每页最多只能有 255 行。）为减少浪费的空间，小一点的页大小可能更合适。

不能将备份复原为另一种页大小。

不能导入超过 755 列的 IXF 数据文件。有关将数据导入表和 IXF 数据文件的更多信息，参考 *Data Movement Utilities Guide and Reference*。

已说明临时表只能在它们自己的“用户临时”表空间类型的表空间中创建。没有缺省用户临时表空间。临时表不能含有 LONG 数据。当应用程序与数据库断开连接时，该表被隐式删除，估计它们的空间需求时应将这一点考虑在内。

## 长整数字段数据

长整数字段数据存储在一个单独的表对象中，它的结构与其他数据类型不同（参见第84页的『用户表数据』和『大对象 (LOB) 数据』）。

数据存储在一些大小为 32 KB 的区域中，这些区域被分成大小为 512 字节的“2 的幂”倍的段。（因此，这些段可以是 512 字节、1024 字节、2048 字节，以此类推，直至 32 768 字节。）

长整数字段数据类型（LONG VARCHAR 或 LONG VARGRAPHIC）以使自由空间易于收回的方式存储。有关分配和自由空间的信息存储在 4KB 分配页中，它在整个对象中不经常出现。

对象中未使用的空间量取决于长整数字段数据的大小以及此大小是否在该数据的所有出现之处都是不变的。对于大于 255 字节的数据项，这个未使用的空间最大可为该长整数字段数据大小的 50%。

若该字符数据的长度小于页大小，且它适合含有该数据其余部分的记录，则应该使用 CHAR、GRAPHIC、VARCHAR 或 VARGRAPHIC 数据类型，而不要使用 LONG VARCHAR 或 LONG VARGRAPHIC。

## 大对象 (LOB) 数据

大对象 (LOB) 数据存储在一个单独的表对象中，这两个对象的结构与其他数据类型不同（参见第84页的『用户表数据』和『长整数字段数据』）。

要估计 LOB 数据所需的空间，需要考虑用来存储使用这些数据类型定义的数据的两个表对象：

- **LOB 数据对象**

数据存储在一些大小为 64 MB 的区域中，这些区域被分成大小为 1024 字节的“2 的幂”倍的段。（因此，这些段可以是 1024 字节、2048 字节、4096 字节，以此类推，直至 64MB。）

要减少 LOB 数据所用的磁盘空间量，可在 CREATE TABLE 和 ALTER TABLE 语句上的 LOB 选项子句上使用 COMPACT 参数。COMPACT 选项将所需的磁

盘空间量减至最小，方法是将 LOB 数据分成更小的段。此过程不涉及数据压缩，只是使用最接近 1 KB 边界的最小空间量。使用 COMPACT 选项可能导致在追加 LOB 值时性能下降。

包含在 LOB 数据对象中的自由空间量将受到更新和删除活动量，以及要插入的 LOB 值的大小的影响。

- **LOB 分配对象**

有关分配和自由空间的信息存储在与实际数据分离的 4 KB 分配页中。这些 4KB 页的数目取决于数据量，包括为大对象数据分配的未使用空间。开销的计算如下：每 64 GB 一个 4 KB 页加上每 8 MB 一个 4KB 页。

若该字符数据的长度小于页大小，且它适合含有该数据其余部分的记录，则应该使用 CHAR、GRAPHIC、VARCHAR 或 VARGRAPHIC 数据类型，而不要使用 BLOB、CLOB 或 DBCLOB。

## 索引空间

对于每个索引，可以按如下公式估计所需的空间：

$$(\text{平均索引键大小} + 8) * \text{行数} * 2$$

其中：

- “平均索引键大小”是索引键中每列的字节计数。有关如何计算不同数据类型的列的字节计数的信息，参考 *SQL Reference* 中的 CREATE TABLE 语句。（估计 VARCHAR 和 VARGRAPHIC 列的平均列大小时，使用当前数据大小的平均值加一个字节。不要使用最大说明大小。）
- 因子 “2” 表示开销，如非叶子页和自由空间。

**注：**对于允许 NULL 的每个列，添加一个额外的字节以表示该空值指示符。

创建索引时，临时空间是必需的。在创建索引期间所需的最大临时空间可以按如下公式估计：

$$(\text{平均索引键大小} + 8) * \text{行数} * 3.2$$

其中，因子 “3.2” 表示索引开销，以及索引创建期间进行排序所需的空间。

**注：**对于非唯一索引，存储重复的键项只需四个字节。上面显示的估计是假定没有重复项。因此以上公式可能会过多地估计存储索引所需的空间。

可使用下面两个公式来估计叶子页的数目（第二个公式提供更准确的估计）。这些估计的准确度很大程度上取决于平均值反映实际数据的准确程度。

注: 对于 SMS 表空间, 所需的最小空间为 12 KB。对于 DMS 表空间, 最小值是一个数据块。

- 每个叶子页的平均键数的粗略估计是:

$$\frac{(.9 * (U - (M*2))) * (D + 1)}{K + 6 + (4 * D)}$$

其中:

- U (一页上的可用空间) 大约等于页大小减 100。对于 4096 的页大小, U 是 3996。
- M = U / (8 + 最小键大小)
- D = 每个键值的平均重复项数
- K = 平均键大小

记住, 最小键大小和平均键大小必须有一个额外字节, 表示每个可空键部分; 还必须有一个额外的字节, 表示每个变长键部分的长度。

若存在包含列, 则在计算最小键大小和平均键大小时应将它们考虑在内。

若在创建索引期间指定了非缺省值 10% 的一个空闲百分比, 则可以使用任何 (100 - pctfree)/100 替换 .9。

- 每个叶子页的平均键数的更准确估计是:

$$L = \text{叶子页数} = X / (\text{叶子页上的平均键数})$$

其中, X 是表中的总行数。

可按如下方法估算索引的原始大小:

$$(L + 2L/(\text{叶子页上的平均键数})) * \text{页大小}$$

对于 DMS 表空间, 将一个表上所有索引的大小加在一起, 然后四舍五入为该索引所在表空间的数据块大小的一个倍数。

应该为 INSERT/UPDATE 活动所引起的索引增长提供附加空间, 这种增长可能导致分页。

使用下列计算方法来获得更精确的原始索引大小的估算值, 以及该索引中级别数的估算值。(若索引定义中使用包含列, 可能要引起特别注意。) 每个非叶子页的平均键数大约是:

$$\frac{(.9 * (U - (M*2))) * (D + 1)}{K + 12 + (8 * D)}$$

其中:

- $U$  (一页上的可用空间) 大约等于页大小减 100。对于 4096 的页大小,  $U$  是 3996。
- $D$  是在非叶子页上每个键值的重复值的平均数目 (这将在叶子页上的小很多, 您可能想将该值设置为 0 以便简化计算)。
- $M = U / (8 + \text{非叶子页的最小键大小})$
- $K = \text{非叶子页的平均键大小}$

只要没有包含列, 非叶子页与叶子页的最小键大小和平均键大小将是相同的。包含列不存储在非叶子页上。

除非  $(100 - \text{pctfree})/100$  大于 .9, 否则不应使用它来替换 .9, 因为在创建索引期间会在非叶子页上留下 10% 的自由空间。

可用如下所示的方法估算非叶子页数:

```
if L > 1 then {P++; Z++;}
while (Y > 1)
{
  P = P + Y
  Y = Y / N
  Z++
}
```

其中:

- $P$  是页数 (最初为 0)。
- $L$  是叶子页数。
- $N$  是每个非叶子页的键数。
- $Y = L / N$
- $Z$  是索引树中的级别数 (最初为 1)。

总页数是:

$$T = (L + P + 2) * 1.0002$$

附加的 0.02% 表示开销, 包括空间映射页。

创建索引所需的容量估算为:

$$T * \text{页大小}$$

---

## 附加空间需求

以下两项需要附加空间:

- 『日志文件空间』
- 第91页的『临时工作空间』

### 日志文件空间

日志文件所需的空间量（以字节数计），其范围可以为:

$$( \logprimary * ( \logfilsiz + 2 ) * 4096 ) + 8192$$

至:

$$( ( \logprimary + \logsecond ) * ( \logfilsiz + 2 ) * 4096 ) + 8192$$

其中:

- *logprimary* 是在数据库配置文件中定义的主日志文件数
- *logsecond* 是在数据库配置文件中定义的辅助日志文件数
- *logfilsiz* 是在数据库配置文件中定义的每个日志文件中的页数
- 2 是每个日志文件所需的标题页的数目
- 4096 是一页中的字节数
- 8192 是日志控制文件的大小（以字节计）。

有关 *logprimary*、*logsecond* 和 *logfilsiz* 配置参数的更多信息，参考 《管理指南: 性能》。

**注:** 总的活动日志空间不能超过 32 GB。

日志文件空间的上限取决于数据库管理器在运行时所需要的辅助日志文件的实际数目。此上限可能永远不需要，或可能只在很偶然的大量活动期间才需要。

若允许对该数据库执行前滚恢复，应该考虑特殊的日志空间需求:

- 当启用 *logretain* 配置参数时，日志文件将被归档在日志路径目录中。除非将日志文件移至另一个位置，否则，联机磁盘空间最终将会填满。
- 当启用 *userexit* 配置参数时，用户出口程序会将归档的日志文件移至另一个位置。要允许下列情况，附加的日志空间仍是必需的：
  - 等待用户出口程序移动的联机归档日志
  - 要格式化以供将来使用的新日志文件。



## 临时工作空间

某些 SQL 语句需要临时表来进行处理（如使用一个工作文件来进行不能在内存中执行的排序）。这些临时表需要磁盘空间；所需的空间量取决于查询以及返回的表的大小，无法估计。

可以使用数据库系统监控器和查询表空间 API 来跟踪在正常操作期间所用的工作空间量。

---

## 设计节点组

节点组是定义为属于某个数据库的一个或多个节点的命名集合。作为数据库系统配置一部分的每个数据库分区，必须已在称为 *db2nodes.cfg* 的分区配置文件中定义。一个节点组可以只包含一个数据库分区，也可以包含为该数据库系统定义的所有数据库分区。

使用 CREATE NODEGROUP 语句可创建新的节点组，可使用 ALTER NODEGROUP 语句修改它。可以在一个节点组中添加或删除一个或多个数据库分区。在修改节点组之前，数据库分区必须已在 *db2nodes.cfg* 文件中定义。表空间驻留在节点组中。表驻留在表空间内。

当创建或修改一个节点组时，会有一个分区映射与它相关。分区映射，连同分区键和散列算法，被数据库管理员用于确定该节点组中哪个数据库分区将存储给定行的数据。有关分区映射的更多信息，参见第93页的『分区映射』。有关分区键的更多信息，参见第94页的『分区键』。

使用非分区数据库时，不需要任何分区键或分区映射。若使用非分区数据库，则不存在节点组设计注意事项。数据库分区是数据库的一部分，与用户数据、索引、配置文件和事务日志组成。创建数据库时创建的缺省节点组由数据库管理员使用。IBMCATGROUP 是包含系统目录的表空间的缺省节点组。IBMTEMPGROUP 是系统临时表空间的缺省节点组。IBMDEFAULTGROUP 是包含用户定义表的表空间的缺省节点组，可选择将那些表置于该处。已说明临时表的用户临时表空间可在 IBMDEFAULTGROUP 或用户创建的任何节点组中创建，但不能在 IBMTEMPGROUP 中创建。

若使用的是多分区节点组，应考虑下列设计要点：

- 在多分区节点组中，若索引是该分区键的超集，则只能创建唯一的索引。
- 根据该数据库中的数据库分区数，可能有一个或多个单分区节点组，且存在一个或多个多分区节点组。
- 必须给每个数据库分区指定唯一的分区号。在一个或多个节点组中可能会发现相同的数据库分区。

- 要确保快速恢复包含系统目录表的数据库分区，则须避免将用户表放在同一个数据库分区上。这可通过将用户表放在不包括 IBMCATGROUP 节点组中的那些数据库分区的节点组中来实现。

除非要与一个更大的表并置，否则应该将小表放在单分区节点组中。并置是将不同表中包含相关数据的行放置在同一个数据库分区中。并置的表允许 DB2 利用更有效的连接策略。并置的表可以驻留在单个分区节点组中。若表驻留在多分区节点组中，且在分区键中具有相同数目的列，对应列的数据类型是分区兼容的，则将这些表视为是并置的。在并置的表中具有相同分区键值的行被放置在同一个数据库分区上。这些表可以位于同一个节点组中单独的表空间中，且仍被认为是并置的。

应该避免将中等大小的表扩展到太多数据库分区上。例如，100 MB 的表在 16 个分区的节点组中可能比在 32 个分区的节点组中执行得更好。

可以使用节点组将联机事务处理 (OLTP) 表与决策支持 (DSS) 表分离，以确保 OLTP 事务的性能不受负面影响。

## 节点组设计注意事项

逻辑数据库设计和要处理的数据量将决定数据库是否需要分区。本节包括下列与数据库分区相关的主题：

- 『数据分区』
- 第93页的『分区映射』
- 第94页的『分区键』
- 第96页的『表并置』
- 第96页的『分区兼容性』
- 第97页的『复制的摘要表』

## 数据分区

DB2 支持分区存储模型，这允许您将数据存储于数据库的多个数据库分区中。这意味着该数据是以物理方式存储在多个数据库分区中，同时仍可以看作数据是位于同一个位置来进行访问。访问一个分区数据库中的数据的应用程序和用户不需要知道该数据的物理位置。

该数据在物理上是分离的，但在逻辑上将它作为一个整体来使用和管理。用户可以选择如何通过说明分区键来将数据分区。用户还可以通过选择应存储数据的表空间和相关的节点组，以确定表数据可以分布在哪些数据库分区以及多少个数据库分区上。另外，将可更新的分区映射与散列算法一起使用，来指定分区键值至数据库分区的映射，以确定每行数据的位置和检索。因此，可以将大表的工作负

荷分布在整个分区数据库上，并将较小的表存储在一个或多个数据库分区上。每个数据库分区具有它存储的数据的本地索引，这可以提高本地数据访问的性能。

并不是必须将所有表都分布到数据库中的所有数据库分区上。DB2 支持部分撤消群集，这意味着可以将表及其表空间分布到系统（即，节点组）中的数据库分区的一个子集上。

当您想要将表放在每个数据库分区上时，可考虑的替代方法是使用摘要表，然后复制这些表。可创建包含所需信息的摘要表，然后将它复制至每个节点。有关更多信息，参见第97页的『复制的摘要表』。

### 分区映射

在分区数据库环境中，数据库管理员必须设法弄清一个表的哪些行存储哪些数据库分区上。数据库管理器必须知道到哪里去查找所需的数据，并使用一个称为分区映射的映射来查找数据。

分区映射是一个内部生成的数组，对于多分区节点组，它包含 4 096 项，对于单分区节点组，只包含一项。对于单分区节点组，分区映射只有一个项，该项包含用于存储一个数据库表所有行的数据库分区的分区号。对于多分区节点组，以循环方式指定节点组的分区号。正如使用网格将城市地图划分为区一样，数据库管理员使用分区键来确定存储数据的位置（数据库分区）。

例如，假定您将一个数据库创建在四个数据库分区（编号为 0-3）上。此数据库的 IBMDEFAULTGROUP 节点组的分区映射将是：

```
0 1 2 3 0 1 2 ...
```

若已使用数据库分区 1 和 2 在该数据库中创建了一个节点组，则该节点组的分区映射将是：

```
1 2 1 2 1 2 1 ...
```

若要装入数据库的一个表的分区键是一个可能在范围 1 至 500 000 之间的整数，则会将分区键散列至 0 至 4 095 之间的一个分区号。将该编号用作分区映射中的索引，以选择用于该行的数据库分区。

第94页的图27显示如何将具有分区键值 (c1, c2, c3) 的行映射至分区 2，然后引用数据库分区 n5。

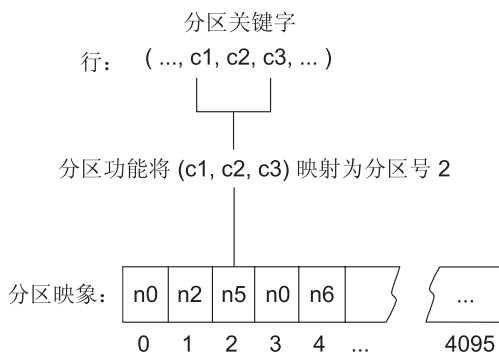


图 27. 使用分区映射的数据分布

分区映射用于灵活控制将数据存储在一个分区数据库中的位置。若将来需要更改数据库中各数据库分区上的数据分布，可以使用数据重新分布实用程序。此实用程序允许重新协调或调整数据分布。有关此实用程序的更多信息，参考《管理指南：性能》中的“跨数据库分区重新分布数据”。

可以使用“获取表分区信息” (**sqlugtpi**) API，来获取您可以查看的分区映射的副本。有关此 API 的更多信息，参考 *Administrative API Reference*。

## 分区键

分区键是一列（或一组列），用于确定将某行数据存储到什么分区。分区键是使用 **CREATE TABLE** 语句在表上定义的。若没有为一个表空间（该表空间被划分为一个节点组中的多个数据库分区）中的表定义分区键，缺省情况下会根据主键的第一列创建一个。若未指定主键，则缺省分区键是在该表中定义的第一个非长整数字段列。（长整数包括所有长整数数据类型和所有大对象 (LOB) 数据类型）。若正在与单个分区节点组相关的表空间中创建一个表，且您希望有一个分区键，则必须明确地定义该分区键。缺省情况下，不会创建它。

若没有列满足缺省分区键的要求，则会不带键创建该表。仅在单个分区节点组中才允许不带分区键的表。以后可以使用 **ALTER TABLE** 语句添加或删除分区键。只能对表空间与单分区节点组相关的表更改分区键。

选择好的分区键很重要。应考虑下列各项：

- 如何访问表
- 查询工作负荷的性质
- 数据库系统所使用的连接策略。

若并置不是主要的注意事项，则一个好的表分区键就是要考虑的主要事项，它可以将数据均匀地分布在节点组中的所有数据库分区上。与一个节点组相关的表空间中每个表的分区键确定这些表是否已并置。下列情况中，表被认为是并置的：

- 表被置于同一个节点组中的表空间内
- 每个表中的分区键具有相同数量的列
- 对应列的数据类型是分区兼容的。

这些特征确保并置的表中具有相同分区键值的那些行位于同一个分区上。有关分区兼容性的更多信息，参见第96页的『分区兼容性』。有关表并置的更多信息，参见第96页的『表并置』。

不适当的分区键会导致数据分布不均匀。数据分布不均匀的列和含有少数相异值的列不应选作分区键。相异值的数目必须足够大，以确保将行均匀地分布在节点组中的所有数据库分区上。应用分区散列算法的成本与分区键的大小是成正比的。分区键不能超过 16 列，而且列越少，性能越好。不应将不需要的列包括在分区键中。

当定义分区键时，应该考虑下列几点：

- 不支持创建只包含长整数数据类型（LONG VARCHAR、LONG VARGRAPHIC、BLOB、CLOB 或 DBCLOB）的多分区表。
- 不能改变分区键定义。
- 分区键应该包括最频繁连接的列。
- 分区键应该由经常参与 GROUP BY 子句的列组成。
- 任何唯一键或主键必须包含所有分区键列。
- 在联机事务处理 (OLTP) 环境中，分区键中的所有列都应该使用带常量或主变量的等于 (=) 谓词来参与该事务。例如，假定有一个在事务中经常使用的雇员号 *emp\_no*，如：

```
UPDATE emp_table SET ... WHERE  
emp_no = host-variable
```

在此情况下，EMP\_NO 列对于 EMP\_TABLE 而言是一个不错的单列分区键。

若 DB2\_UPDATE\_PART\_KEY 注册表变量设置为 NO，则不能更新表中某一行的分区键列值，只能删除或插入分区键值。

散列分区是确定分区表中每一行的位置的方法。该方法的原理如下：

1. 将散列算法应用于分区键的值，并生成介于 0 与 4095 之间的分区号。
2. 在创建节点组时创建分区映射。每个分区号依次以循环方式重复，以填写该分区映射。有关分区映射的更多信息，参见第93页的『分区映射』。

3. 该分区号用作分区映射中的一个索引。分区映射中该位置处的编号是存储该行的数据库分区的编号。

### 表并置

您可能会发现，作为对特定查询的响应，两个或多个表频繁地提供数据。在此情况下，您会希望这样的表中的相关数据的位置尽可能地靠近。在数据库被物理地划分为两个或多个数据库分区的环境中，必须有一种方法可将划分的表的相关碎片尽可能地靠近。完成此过程的功能称为表并置。

当表存储在同一个节点组中且它们的分区键兼容时，这些表是已并置的。将两个表置于同一个节点组中，确保存在一个公共的分区映射。这些表可能位于不同的表空间，但是这些表空间必须与同一个节点组相关。每个分区键中对列的数据类型必须是分区兼容的。有关分区兼容性的信息，参见『分区兼容性』。

当访问用于连接或子查询的多个表时，DB2 能够识别要连接的数据是否位于同一个数据库分区上。当发生这种情况时，DB2 可以选择在存储该数据的数据库分区上执行连接或子查询，而不必在数据库分区之间移动数据。这种在数据库分区上执行连接或子查询的能力具有显著的性能优点。有关更多信息，参考《管理指南：性能》中的“并置连接”。

### 分区兼容性

对分区键的对应列的基本数据类型进行比较，并可将它们说明为是分区兼容的。分区兼容的数据类型具有如下特性：具有相同值但有不同类型的两个变量会按相同的分区算法映射至同一个分区号。

分区兼容性具有下列特征：

- 基本数据类型与另一个相同的基本数据类型兼容。
- 内部格式用于 DATE、TIME 和 TIMESTAMP 数据类型。它们彼此都不兼容，且都不与 CHAR 兼容。
- 分区兼容性不受带有 NOT NULL 或 FOR BIT DATA 定义的列的影响。
- 对兼容数据类型的空值的处理是完全相同的；对不兼容数据类型的空值的处理可能不相同。
- 用户定义类型的基本数据类型用于分析分区兼容性。
- 对分区键中值相同的小数的处理是完全相同的，即使它们的标度和精度不同。
- 字符串中（CHAR、VARCHAR GRAPHIC 或 VARGRAPHIC）的尾部空格会被散列算法忽略。
- BIGINT、SMALLINT 和 INTEGER 是兼容的数据类型。
- REAL 和 FLOAT 是兼容的数据类型。

- 不同长度的 CHAR 和 VARCHAR 是兼容的数据类型。
- GRAPHIC 和 VARGRAPHIC 是兼容的数据类型。
- 分区兼容性不适用于 LONG VARCHAR、LONG VARGRAPHIC、CLOB、DBCLOB 和 BLOB 数据类型，因为不支持它们作为分区键。

### 复制的摘要表

摘要表是由查询定义的表，也用于确定表中的数据。摘要表可用于改进查询的性能。若 DB2 确定查询的一部分可用摘要表来解决，则数据库管理器可重写该查询以使用摘要表。

在分区数据库环境中，可复制摘要表。可使用复制的摘要表来改进查询性能。复制的摘要表基于一个可能已在单分区节点组中创建，但您想要在节点组中的所有数据库分区中进行复制的表。要创建复制的摘要表，调用带 REPLICATED 关键字的 CREATE TABLE 语句。

有关摘要表的更多信息，参见《管理指南：实现》中的“创建摘要表”。

通过使用复制的摘要表，可将通常未并置的表并置。对于一个大的事实表和一个小的维表的连接，复制的摘要表特别有用。要将所需的额外存储器以及必须更新每个副本所带来的影响降至最小，要复制的表应较小，且更新不频繁。

**注：**还应考虑复制那些不常更新的更大的表：复制的一次性成本可通过并置获得的性能效益来抵消。

通过在定义复制表所用的子查询子句中指定适当的谓词，可以复制选择的列和/或选择的行。

有关复制的摘要表的更多信息，参考 *SQL Reference* 中的 CREATE TABLE 语句。有关并置连接的更多信息，参考《管理指南：实现》中的“并置连接”。

---

## 设计和选择表空间

表空间是一种存储模型，它提供一个数据库和存储在该数据库内的表之间的一个间接层。表空间驻留在节点组中。它们允许将数据库和表数据的位置直接指定到容器上。（容器可以是目录名、设备名或文件名。）这可提供改善的性能、更灵活的配置和更好的完整性。

有关创建或改变表空间的信息，参见“创建表空间”或《管理指南：实现》中的“改变表空间”。

因为表空间驻留在节点组中，所以选择保存表的表空间时要定义如何将该表的数据分布至节点组中的数据库分区上。单个表空间可跨多个容器。在同一个物理磁盘（即驱动器）上创建多个容器（一个或多个表空间）是可能的。为提高性能，每个容器应使用不同的磁盘。图28举例说明了一个数据库内的表和表空间与该数据库相关的容器之间的关系。

数据库

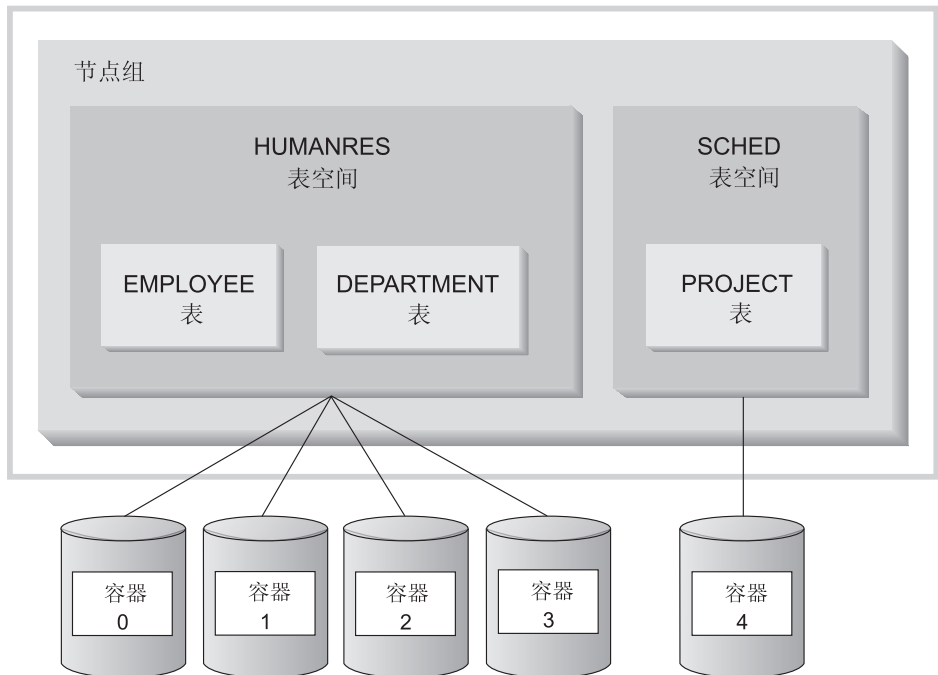


图 28. 数据库中的表空间和表

EMPLOYEE 和 DEPARTMENT 表在 HUMANRES 表空间中，该表空间跨越容器 0、1、2 和 3。PROJECT 表在容器 4 的 SCHED 表空间中。此示例显示每个容器存在于不同的磁盘上。

数据库管理器会尝试平衡分布在所有容器中的数据负荷。因此，所有容器都将用于存储数据。数据库管理器在使用另一个容器之前写入一个容器的页数称为数据块大小。数据库管理器并非始终从第一个容器开始存储表数据。

第99页的图29显示具有两个 4KB 页数据块大小的 HUMANRES 表空间，它有四个容器，每个容器有少量已分配的数据块。DEPARTMENT 和 EMPLOYEE 表都有 7 页，且横跨全部四个容器。



## HUMANRES 表空间

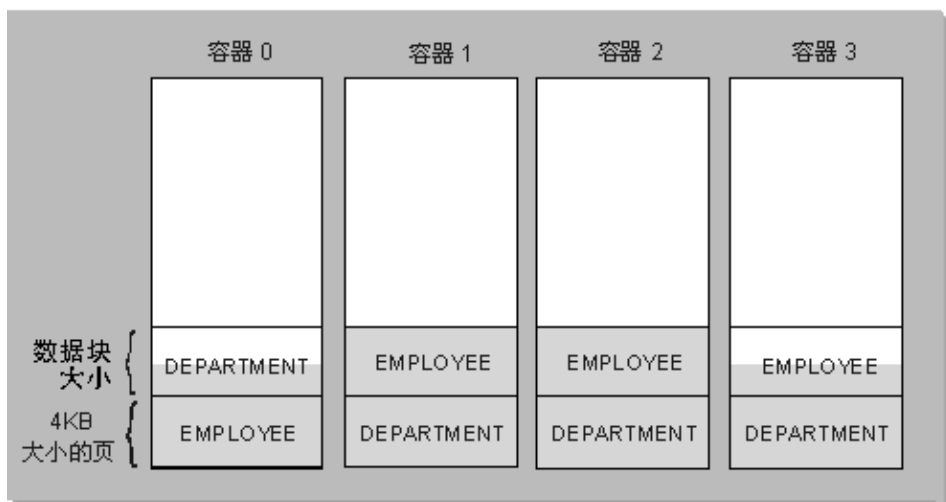


图 29. 容器和数据块

一个数据库至少必须包含三个表空间:

- 一个目录表空间，它包含该数据库的所有系统目录表。此表空间称为 `SYSCATSPACE`，它不能被删除。`IBMCATGROUP` 是此表空间的缺省节点组。
- 一个或多个用户表空间，它们包含所有用户定义表。缺省情况下，会创建一个表空间 `USERSPACE1`。`IBMDEFAULTGROUP` 是此表空间的缺省节点组。

当创建一个表时，应指定表空间名，否则，结果可能不是您所期望的。若不指定表空间名，则根据下列规则放置该表：若存在用户创建的表空间，则选择页大小最小，但对此表而言足够大的表空间。否则，若 `USERSPACE1` 的页大小对此表而言足够大，则使用它。若不存在具有足够大页大小的表空间，则不创建该表。

表的页大小或者由行大小确定，或者由列数确定。一行中允许的最大长度取决于创建此表所在的表空间的页大小。可能的页大小的值为 4 KB（缺省值）、8 KB、16 KB 和 32 KB。可使用具有一种页大小的表空间作为基表，而使用具有另一种页大小的另一个表空间来存储长整数或 LOB 数据。（记住，SMS 不支持跨表空间的表，而 DMS 却支持。）若列数或行大小超过表空间页大小的限制，则返回一个错误 (SQLSTATE 42997)。

- 一个或多个临时表空间，它们包含临时表。临时表空间可以是系统临时表空间或用户临时表空间。数据库必须有至少一个系统临时表空间；在缺省情况下，创建数据库时会创建一个名为 `TEMPSPACE1` 的系统临时表空间。`IBMTEMPGROUP` 是此表空间的缺省节点组。用户临时表空间不是在数据库创建时缺省创建的。

若一个数据库使用多个临时表空间，并且需要新的临时对象，则优化器将为此对象选择适当的页大小。然后将把该对象分配到具有相应页大小的临时表空间中。若存在多个临时表空间具有该页大小，则将以循环方式选择表空间。

如果对用 4 KB（缺省值）以上的页大小定义的表空间中的表运行查询（例如，对 1012 列使用 ORDER BY），则某些查询可能会失败。若没有使用更大页大小定义的临时表空间，则会发生这种情况。可能需要创建具有更大页大小（8 KB、16 KB 或 32 KB）的临时表空间。若没有与用户表空间中的最大页大小具有相同页大小的临时表空间，任何一种 DML（数据操作语言）语句都可能失败。

应定义一个 SMS 临时表空间，使它的页大小等于大多数用户表空间所使用的页大小。这对于典型的环境和工作负荷应已足够。另见第110页的『对临时表空间的建议』。

在分区数据库环境中，目录节点将包含全部三个缺省表空间，而其他每个数据库分区将只包含 TEMPSPACE1 和 USERSPACE1。

有两种类型的表空间，它们都可以在单个数据库中使用：

- 『系统管理空间』：操作系统的文件管理器控制该存储空间。
- 第104页的『数据库管理空间的表空间』：数据库管理器控制该存储空间。

## 系统管理空间

在 SMS（系统管理空间）表空间中，操作系统的文件系统管理器分配和管理用于存储表的空间。该存储模型通常由存储在文件系统空间中的多个文件组成，这些文件表示表对象。用户决定这些文件的位置，DB2 控制它们的名称，而文件系统负责管理它们。通过控制写入每个文件的数据量，数据库管理器均匀地将数据分布在所有表空间容器中。SMS 表空间是缺省表空间。

每个表至少有一个与它相关的 SMS 物理文件。有关这些文件的列表及其内容的描述，参见第103页的『SMS 物理文件』。

在 SMS 表空间中，当对象增大时，文件每次扩展一页。若需要提高插入性能，可以考虑启用多页文件分配。这允许系统一次为文件分配或扩展多页。必须运行 **db2empfa**，才可启用多页文件分配。在分区数据库环境中，必须对每个数据库分区运行此实用程序。一旦启用了多页文件分配，就不能禁用它。有关 **db2empfa** 的更多信息，参考 *Command Reference*。

应该在 CREATE DATABASE 命令上或 CREATE TABLESPACE 语句上使用 MANAGED BY SYSTEM 选项来显式地定义 SMS 表空间。当设计 SMS 表空间时，必须考虑两个关键要素：

- 表空间的容器。

必须指定要用于表空间的容器数。标识要使用的所有容器是非常重要的，因为您不能在创建了 SMS 表空间之后添加或删除容器。在分区数据库环境中，当将一个新的分区添加至 SMS 表空间的节点组时，可以使用 ALTER TABLESPACE 语句为新的分区添加容器。

用于一个 SMS 表空间的每个容器都标识一个绝对或相对的目录名。其中每一个目录都可以位于不同的文件系统（物理磁盘）上。表空间的最大大小可以按以下方法估计：

容器数 \* (操作系统支持的最大文件系统大小)

此公式假定有一个唯一的文件系统映射至每个容器，且每个文件系统都具有大量的可用空间。实际上，情况可能不是这样，表空间最大大小可能小得多。

**注：**定义容器时必须很小心。若容器上已有文件或目录，将返回一个错误 (SQL0298N)。

- 表空间的数据块大小。

数据块大小只能在创建表空间时指定。因为以后不能更改它，因此为数据块大小选择一个适当的值就很重要。有关更多信息，参见第109页的『选择数据块大小』。

当创建表空间时，若不指定数据块大小，则数据库管理器将使用缺省数据块大小来创建表空间，该缺省大小由 *dft\_extent\_sz* 数据库配置参数定义（有关此参数的更多信息，参考《管理指南：性能》）。此配置参数最初是根据创建该数据库时提供的信息设置的。若未在 CREATE DATABASE 命令上指定 *dft\_extent\_sz* 参数，则会将缺省数据块大小设置为 32。

要为表空间的容器数和数据块大小选择适当的值，必须了解：

- 操作系统对逻辑文件系统的大小施加的限制。

例如，某些操作系统有 2GB 的限制。因此，若想要一个 64GB 的表对象，则在此类型的系统上将需要至少 32 个容器。

当创建该表空间时，可以指定驻留在不同文件系统上的容器，以便增加可以存储在该数据库中的数据量。

- 数据库管理器如何管理与一个表空间相关的数据文件和容器。

在为该表空间指定的第一个容器中创建第一个表数据文件 (SQL00001.DAT)，并允许此文件增大至该数据块大小。当它达到此大小之后，数据库管理器将数据写入下一个容器中的 SQL00001.DAT。此过程会继续，直到所有容器都包含 SQL00001.DAT 文件为止，在那时，数据库管理器会返回至第一个容器。此过程（称为条形化分布）会继续在容器目录中运行，直到一个容器装满为止

(SQL0289N) 或操作系统中已无空间可分配为止 ( 磁盘已满错误 )。条纹化也用于索引 (SQLnnnnn.INX)、长整数字段 (SQLnnnnn.LF) 和 LOB (SQLnnnnn.LB 和 SQLnnnnn.LBA) 文件。

**注：**只要任何一个容器一满，SMS 表空间就满了。因此，为每个容器分配相同容量的空间是很重要的。

为了有助于将数据更加均匀地分布至这些容器中，数据库管理器根据将表标识符 ( 以上示例中为 1 ) 除以容器数所得的模，来确定首先使用的容器。容器从 0 开始依次编号。

有关在 SMS 表空间中使用的文件的更多信息，参见第103页的『SMS 物理文件』。

## SMS 物理文件

下列文件可在 SMS 表空间目录容器中找到:

文件名	描述
-----	----

### SQLTAG.NAM

每个容器的子目录中都存在这样一个文件，而且当连接数据库以验证该数据库是完整和一致时，数据库管理器要使用它们。

### SQLxxxxx.DAT

表文件。它存储所有表行，但 LONG VARCHAR、LONG VARCHARIC、BLOB、CLOB 或 DBCLOB 数据除外。

**SQLxxxxx.LF** 包含 LONG VARCHAR 或 LONG VARCHARIC 数据（也称为“长整数字段数据”）的文件。仅当表中存在 LONG VARCHAR 或 LONG VARCHARIC 列时，才创建此文件。

### SQLxxxxx.LB

包含 BLOB、CLOB 或 DBCLOB 数据（也称为“LOB 数据”）的文件。仅当表中存在 BLOB、CLOB 或 DBCLOB 列时，才创建这些文件。

### SQLxxxxx.LBA

包含有关 SQLxxxxx.LB 文件的分配和自由空间信息的文件。

### SQLxxxxx.INX

表的索引文件。对应表的所有索引都存储在此单个文件中。仅当定义了索引时，才创建该文件。

**注:** 当删除一个索引时，索引 (.INX) 文件并未实际释放该空间，除非删除该索引文件。若表上的所有索引都被删除（和被提交），或若该表被重组，则会删除该索引文件。若未删除该索引文件，则一旦提交了删除操作，就会将该空间标记为空闲的并将重新使用它，以便将来创建索引或维护索引。

### SQLxxxxx.DTR

用于重组 DAT 文件的临时数据文件。当重组一个表时，REORG 实用程序（通过 REORG TABLE 命令调用）会在其中一个系统临时表空间中创建一个表。可以定义这些临时表空间使用与用户定义表所用的不同的容器。

### SQLxxxxx.LFR

用于重组 LF 文件的临时数据文件。当重组一个表时，REORG 实用程序（通过 REORG TABLE 命令调用）会在其中一个系统临时表空间中创建一个表。可以定义这些临时表空间使用与用户定义表所用的不同的容器。

## SQLxxxxx.RLB

用于重组 LB 文件的临时数据文件。当重组一个表时，REORG 实用程序（通过 REORG TABLE 命令调用）会在其中一个系统临时表空间中创建一个表。可以定义这些临时表空间使用与用户定义表所用的不同的容器。

## SQLxxxxx.RBA

用于重组 LBA 文件的临时数据文件。当重组一个表时，REORG 实用程序（通过 REORG TABLE 命令调用）会在其中一个系统临时表空间中创建一个表。可以定义这些临时表空间使用与用户定义表所用的不同的容器。

### 注:

1. 不要直接更改这些文件。只能使用记载的 API 和实现那些 API 的工具（包括命令行处理器和 控制中心）来间接访问它们。
2. 不要移动这些文件。
3. 不要除去这些文件。
4. 备份数据库或表空间唯一受支持的方法是通过 **sqlubkp**（备份数据库）API 来进行，包括该 API 的命令行处理器和 控制中心实现。

## 数据库管理空间的表空间

在 DMS（数据库管理空间）表空间中，数据库管理器控制存储空间。该存储模型由有限数量的设备组成，这些设备的空间由 DB2 来管理。管理员决定要使用哪个设备，而 DB2 管理那些设备上的空间。此表空间实质上是为了最好地满足数据库管理器的需要而设计的特殊目的的文件系统。该表空间定义包括一些设备或文件的列表，它们属于可以用来存储数据的表空间。

包含用户定义的表和数据的 DMS 表空间可以定义为:

- 存储正常表和索引数据的规则表空间
- 存储长整数字段或 LOB 数据的长整数表空间。

在设计 DMS 表空间和容器时，应该考虑下列事项:

- 数据库管理器使用“拆开”来确保数据均匀地分布在所有容器上。
- 对于 4 KB 页，规则表空间的最大大小是 64 GB；对于 8 KB 页，是 128 GB；对于 16 KB 页，是 256 GB；对于 32 KB 页，是 512 GB。长整数表空间的最大大小是 2 TB。
- 与 SMS 表空间不同，组成一个 DMS 表空间的容器不必大小相同；然而，通常不建议这样做，因为会导致在容器间不均匀地进行条纹化，并会降低性能。若任何容器已满，DMS 表空间会使用其他容器中的可用自由空间。

- 因为空间是预分配的，所以在能够创建表空间之前，该空间必须是可用的。当使用设备容器时，该设备也必须足够的空间以便存储容器定义。每个设备上只能定义一个容器。为避免浪费空间，设备的大小应该等于容器的大小。例如，若分配给该设备 5 000 页，而将设备容器定义为分配 3 000 页，则设备上的 2 000 页将是不可用的。
- 每个容器上有一页是为开销保留的，而其余页将一次使用一个数据块。只使用整个数据块，因此为了对空间进行最优管理，可以使用如下公式来帮助您确定当分配容器时要使用的适当大小：

$$(\text{extent\_size} * n) + 1$$

其中，*extent\_size* 是表空间中每个数据块的大小，而 *n* 是您要在该容器中存储的数据块数。

- 表空间中有三个数据块是保留给开销使用的。
- 要存储任何用户表数据，至少需要两个数据块。（这些数据块是一个表的规则数据所必需的，但不是任何索引、长整数字段或大对象数据所需的，它们需要自己的数据块。）
- 设备容器必须使用带“字符型特殊接口”的逻辑卷，而不是物理卷。
- 在 DMS 表空间可以使用文件以代替设备。文件和设备之间不存操作性区别；但是，由于存在与文件系统相关的运行时开销，文件可能效率较低。在下列情况下，适合使用文件：
  - 设备不直接受支持
  - 设备不可用
  - 不需要最大性能
  - 不想设置设备。
- 若工作负荷涉及 LOB 或 LONG VARCHAR 数据，则可通过文件系统高速缓存改进性能。注意，DB2 的缓冲池不缓冲 LOB 和 LONG VARCHAR。
- 某些操作系统允许拥有大小超过 2GB 的物理设备。应该考虑将该物理设备分区为多个逻辑设备，以使任何容器都不超过操作系统允许的大小。

### 将容器添加至 DMS 表空间

ALTER TABLESPACE 语句用来向一个现存表空间添加容器，以增加它的存储容量。然后，该表空间的内容要在所有容器中重新平衡。在重新平衡期间，不限制对该表空间的访问。若需要添加多个容器，应该在一个 ALTER TABLESPACE 语句中或在同一个事务中同时添加它们，以免数据库管理器不得不多次重新平衡这些容器。

应使用 LIST TABLESPACE CONTAINERS 或 LIST TABLESPACES 命令来检查一个表空间的容器的满载程度。添加新容器应在现存容器几乎满载或完全满载之前进行。要在重新平衡完成之后，所有容器上的新空间才可用。

添加比现存容器小的容器会导致数据分布不均匀。这可能导致并行 I/O 操作，如预取数据，它们的执行效率比在大小相同的容器上执行的效率低。

## 表空间的设计注意事项

本节包括下列主题：

- 『表空间输入和输出 (I/O) 的注意事项』
- 第107页的『将表空间映射至缓冲池』
- 第108页的『将表空间映射至节点组』
- 第108页的『将表映射至表空间』
- 第109页的『选择数据块大小』
- 第110页的『对临时表空间的建议』
- 第111页的『对目录表空间的建议』
- 第112页的『工作负荷注意事项』
- 第113页的『选择 SMS 或 DMS 表空间』
- 第114页的『当将数据存放在 RAID 设备中时优化性能』。

### 表空间输入和输出 (I/O) 的注意事项

表空间的类型和设计决定了对该表空间执行的 I/O 的效率。在进一步考虑关于表空间的设计和使用的问题之前，您应了解下列概念。

**大块读取** 在单个请求中检索多页（通常为一个数据块）的一种读取。一次读取几页比分别读取每页更有效。

**预取** 在一个查询引用页之前，对那些页的读取。总的目的是为缩短响应时间。若页的预取可以与查询的执行异步发生，就能够达到此目的。当一个或多个 CPU 或 I/O 子系统以最大能力运行时，就达到最佳响应时间。

**页清除** 当读取和修改页时，它们会累积在数据库缓冲池中。当读入一页时，便将其读入到缓冲池页中。若该缓冲池已充满修改的页，则必须将这些修改的页之一写出至磁盘，然后才能再读入新的页。为避免缓冲池变满，页清除程序代理程序的任务就是写出修改的页，以保证缓冲池页可用于将来的读取请求。



无论何时，只要是有利的，DB2 就会执行大块读取。当检索顺序排列或本质上是部分顺序排列的数据时，通常会发生这种情况。在一个读取操作中读取的数据量取决于数据块大小 — 数据块大小越大，一次可以读取的页就越多。

数据块存储在磁盘上的方式影响 I/O 效率。在使用设备容器的 DMS 表空间中，数据往往在磁盘上是连续的，且可以在最短的搜索时间和磁盘延迟内进行读取。但是，若使用的是文件，则数据可能被文件系统分散，并存储在磁盘上的多个位置中。当使用一次将文件扩展一页的 SMS 表空间时（这使产生碎片的概率更高），最可能发生此情况。为了供 DMS 表空间使用而预分配的大文件往往在磁盘上是连续的，尤其当该文件分配在一个干净的文件空间中时更是这样。

可以通过在 CREATE TABLESPACE 语句上调整 PREFETCHSIZE 参数，来控制预取的程度。（该数据库中所有表空间的缺省值由 *dft\_prefetch\_sz* 数据库配置参数设置。）PREFETCHSIZE 参数告诉 DB2 在触发一个预取时要读取的页数。通过在 CREATE TABLESPACE 语句上将 PREFETCHSIZE 设置为 EXTENTSIZE 参数的倍数，可以并行读取多个数据块。（该数据库中所有表空间的缺省值由 *dft\_extent\_sz* 数据库配置参数设置。）EXTENTSIZE 参数指定在跳至下一个容器之前将写入一个容器的 4 KB 大小的页数。

例如，假定有一个表空间使用三个设备。若将 PREFETCHSIZE 设置为 EXTENTSIZE 的三倍，则 DB2 可以用并行方式从每个设备中执行大块读取，从而显著增加 I/O 吞吐量。此情况假定每个设备是一个单独的物理设备，且控制器具有足够的带宽来处理来自每个设备的数据流。注意，DB2 可能需要根据查询速度、缓冲池的利用情况和其他因素，在运行时动态调整预取参数。

某些文件系统使用它们自己的预取方法（如，AIX 上的“日志文件系统”）。在某些情况中，文件系统的预取设置得比 DB2 的预取更主动。这可能导致使用文件容器的 SMS 和 DMS 表空间的预取似乎比使用设备的 DMS 表空间的预取执行效率更高。但这是误导，因为它可能是在文件系统中发生的附加级别的预取的结果。DMS 表空间应该能够比任何等效配置的执行效率高。

为提高预取效率（甚至读取效率），必须存在足够数量的干净缓冲池页。例如，可能有一个并行预取请求要从一个表空间读取三个数据块，对于正在读取的每一页，从缓冲池写出经过修改的一页。该预取请求可能被拖慢，导致它跟不上查询的进展。应配置足够数量的页清除程序，以满足预取请求。对于数据库所用的每个实际磁盘，应定义至少一个页清除程序。有关这些主题的更多信息，参考《管理指南：性能》。

### 将表空间映射至缓冲池

每个表空间都与一个特定的缓冲池相关。缺省缓冲池是 IBMDEFAULTBP。若另一个缓冲池要与一个表空间相关，则该缓冲池必须存在（用 CREATE BUFFERPOOL

语句定义它），且当创建该表空间（使用 CREATE TABLESPACE 语句）时定义关联。表空间与缓冲池之间的关联可以使用 ALTER TABLESPACE 语句来更改。

若拥有多个缓冲池，则可以配置数据库使用的内存，以改善整体性能。对于一个或多个用户可任意访问的大表的表空间，缓冲池的大小可能受到限制，因为高速缓存该数据页可能没有好处。用于联机事务应用程序的表空间可以与一个较大的缓冲池相关联，以便可以更长时间地高速缓存应用程序所使用的数据页，导致响应时间更快。配置新缓冲池时必须小心。有关本主题的更多信息，参考《管理指南：性能》中的“管理数据库缓冲池”。

**注：**若确定数据库需要 8 KB、16 KB 或 32 KB 的页大小，必须将使用其中一种页大小的每个表空间映射为具有相同页大小的缓冲池。

当启动数据库时，所有缓冲池必需的存储器对于数据库管理器必须是可用的。若 DB2 无法获取必需的存储器，则数据库管理器将使用缺省缓冲池（具有 4 KB、8 KB、16 KB 和 32 KB 页大小的缓冲池各一个）启动，并发出警告。

在分区数据库环境中，可以为该数据库中的所有分区创建大小相同的一个缓冲池。也可以在不同的分区上创建不同大小的缓冲池。有关 CREATE BUFFERPOOL 语句的更多信息，参考 *SQL Reference*。

### 将表空间映射至节点组

在分区数据库环境中，每个表空间都与一个特定的节点组相关。这允许将表空间的特征应用于该节点组中的每个节点。该节点组必须存在（用 CREATE NODEGROUP 语句定义它），且当使用 CREATE TABLESPACE 语句创建表空间时定义表空间与该节点组之间的关联。

不能使用 ALTER TABLESPACE 语句来更改表空间与节点组之间的关联。只能为节点组内的个别分区更改表空间规范。在单分区环境中，每个表空间都与一个缺省节点组相关。除非定义了一个系统临时表空间（这将导致使用 IBMTEMPGROUP），否则，当定义一个表空间时，缺省节点组是 IBMDEFAULTGROUP。有关 CREATE NODEGROUP 语句的更多信息，参考 *SQL Reference*。有关节点组和物理数据库设计的更多信息，参见第91页的『设计节点组』。

### 将表映射至表空间

当确定如何将表映射至表空间时，应考虑：

- 表的分区。  
至少，应该确保选择的表空间位于含有您所要的分区的节点组中。
- 表中的数据量。

若计划在一个表空间中存储许多小表，则考虑使用 SMS 充当该表空间。对于小表，DMS 表现在 I/O 和空间管理效率方面的优点就没有那么重要。一次分配一页空间的 SMS 的优点（且仅在需要时使用）却对小表更具吸引力。若一个表较大或者您需要更快地访问表中的数据，应考虑具有较小数据块大小的 DMS 表空间。

您可能希望对每个非常大的表都使用单独的表空间，而将所有的小表组合在单个表空间中。这种分隔还允许您根据表空间的使用选择适当的数据块大小。（有关其他信息，参见『选择数据块大小』。

- 表数据的类型。

例如，有的表可能包含不经常使用的历史数据；最终用户可能愿意接受较长的响应时间，来等待对此数据执行的查询。在这种情况下，您可能会为历史表使用另一个表空间，并将此表空间分配给访问速率较低的较便宜的物理设备。

或者，您也许能够标识某些表，这些表对于使数据快速可用以及您需要快速响应时间是必不可少的。可能要将这些表置于分配给一个快速物理设备的表空间中，这样将有助于支持这些重要的数据需要。

通过使用 DMS 表空间，还可以将表数据分布在三个不同的表空间中：一个存储索引数据；一个存储 LOB 和长整数字段数据；一个存储规则表数据。这允许您选择表空间特征和支持最适合该数据的那些表空间的物理设备。例如，可能会将索引数据置于可找到的最快的设备上，这样性能可显著提高。若将一个表分布在各 DMS 表空间中，如果启用前滚恢复，应考虑一起备份和复原那些表空间。SMS 表空间不支持以此方式将数据分布在所有表空间中。

- 管理问题。

某些管理功能可以在表空间级执行，但不能在数据库或表级执行。例如，备份表空间（而不是数据库）可以帮助您更好地利用时间和资源。它允许频繁地备份带有大量更改的表空间，同时仅偶尔地备份带有少量更改的表空间。

可以复原数据库或表空间。若不相关的表不共享表空间，就可以选择复原数据库一个较小的部分，以降低成本。

一种好方法是将相关的表编组在一组表空间中。这些表可以通过引用约束相关，也可以通过定义的其他商业约束相关。

若经常需要删除并重新定义特定表，则应在它自己的表空间中定义该表，因为删除一个 DMS 表空间比删除一个表更有效率。

### 选择数据块大小

表空间的数据块大小表示在将数据写入下一个容器之前，将写入当前容器的表数据的页数。当选择数据块大小时，应考虑：

- 表空间中表的大小和类型。

将 DMS 表空间中的空间分配给一个表，一次一个数据块。当填充该表而一个数据块变满时，会分配新的数据块。

一个表由下列单独的表对象组成：

- 数据对象。它是存储规则列数据的地方。
- 索引对象。在表上定义的所有索引都存储在这里。
- 长整数字段对象。若表有一个或多个 LONG 列，则长整数字段数据存储在此处。
- 两个 LOB 对象。若表有一个或多个 LOB 列，则它们都存储在这两个表对象中：
  - 一个表对象用于存储 LOB 数据
  - 第二个表对象用于存储描述 LOB 数据的元数据。

每个表对象都是单独存储的，每个对象按需要分配新的数据块。每个表对象还与称为数据块映射的一个元数据对象配成一对，该元数据对象描述该表空间中属于该表对象的所有数据块。用于数据块映射的空间也是以一次一个数据块的方式分配。

因此，一个表的空间的初始分配是每个表对象两个数据块。若您在一个表空间中有多个小表，您可能要分配相对大的空间来存储相对少量的数据。在这种情况下，应该指定小的数据块大小或使用一次分配一页的 SMS 表空间。

另一方面，若您有一个增长速率高的非常大的表，且您使用具有较小数据块大小的 DMS 表空间，您可能会有与附加数据块的频繁分配相关的不需要的开销。

- 对这些表访问的类型。

若对表的访问包括许多查询或处理大量数据的事务，则从表中预取数据可以显著改善性能。（有关数据预取以及它与数据块大小的关系的信息，参考《管理指南：性能》。）

- 必需的最小数据块数。

若容器中没有足够的空间，以供表空间的五个数据块使用，则将无法创建表空间。

### 对临时表空间的建议

建议定义一个 SMS 临时表空间，使它的页大小等于大多数规则表空间所使用的页大小。这应该适用于典型的环境和工作负荷。但最好用不同的临时表空间配置和工作负荷进行实验。应该考虑下列几点：

- 在大多数情况下，临时表空间是按顺序成批访问的。也就是说，插入一组行或顺序读取一组行。因此，当需要更少的逻辑或物理页 I/O 请求来读取给定的数据量时，页大小越大，通常所获得的性能越优。当临时表的平均行大小小于页大

小除以 255 的值时，情况就不会总是这样。无论是那种页大小，任何一页上最多可有 255 行。例如，若一个查询需要 15 字节一行的临时表，则用 4 KB 页大小的临时表空间更合适，因为 255 行可以全部包含在一个 4 KB 页中。8 KB（或更大的）页大小将在临时表的每一页上浪费至少 4 KB（或更多）字节的空空间，最好不要减少需要的 I/O 请求数。

- 若在数据库中有超过一半的规则表空间使用了相同的页大小，建议您定义具有相同页大小的临时表空间。这样做的原因是这种安排可以使临时表空间与大多数或全部的规则表空间共享同一个缓冲池空间。这样可简化缓冲池的调整。
- 当使用临时表空间重组表时，该临时表空间的页大小必须与该表的页大小匹配。由于这个原因，您应确保为现存表使用的每种不同的页大小定义了临时表空间，这样才可使用临时表空间重组这些表。

也可不用临时表空间来执行重组，即“在应有的位置”重组该表；或者说，直接在目标表空间中重组。当然，这种“在应有的位置”重组要求在目标表空间中有额外的空间来完成重组过程。有关表重组的其他信息，参考《管理指南：性能》。

- 一般而言，当存在具有不同页大小的临时表空间时，优化器常常会选择具有最大缓冲池的临时表空间。在这种情况下，比较聪明的做法是给一个临时表空间分配一个足够大的缓冲池，而给其余临时表空间分配较小的缓冲池。这种缓冲池分配将有助于保证有效利用主内存。例如，若目录表空间使用 4 KB 页，而其余表空间使用 8 KB 页，则最佳临时表空间配置可能是：具有一个大缓冲池的一个 8 KB 临时表空间；和一个具有小一些的缓冲池的一个 4 KB 表空间。

**注：**目录表空间只能使用 4 KB 页大小。因此，数据库管理器始终强制保证 4 KB 临时表空间的存在，以允许目录表重组。

- 一般情况下，定义具有相同页大小的多个临时表空间没有什么好处。
- 对于临时表空间而言，SMS 几乎总是比 DMS 更合适，因为：
  - 在 SMS 中，磁盘空间是按需要分配的；而在 DMS 中必须预先分配它。预先分配可能比较困难：临时表空间保存着瞬时数据，这些数据在某个时候可能需要很大的存储器，但在正常情况下存储器需求却低得多。使用 DMS 时，必须预先分配存储器的峰值需求量；而使用 SMS 时，在高峰期间可以使用额外的磁盘空间来完成其他任务。
  - 数据库管理器尝试将临时表页保存在内存中，而不是将它们写出至磁盘。因此，DMS 的性能优点就没有那么突出。
  - SMS 容器可以利用文件系统缓冲；DMS 容器则不能。

### 对目录表空间的建议

建议使用 SMS 表空间来存储数据库目录，原因如下：

- 该数据库目录由许多大小不同的表组成。当使用 DMS 表空间时，对于每个表对象，分配最少两个数据块。根据选择的数据块大小，可能产生大量已分配而未使用的空间。当使用 DMS 表空间时，应选择小的数据块大小（二至四页）；否则，应使用 SMS 表空间。
- 目录表中存在大对象 (LOB) 列。LOB 数据不与其他数据一起保留在缓冲池中，而是每次需要时从磁盘中读取。从磁盘读取 LOB 降低了性能。因为一个文件系统常常有它自己存储（或高速缓存）数据的地方，所以使用 SMS 表空间或在文件容器上构建的 DMS 表空间，使得避免 I/O 成为可能（若先前引用了该 LOB 的话）。

将这些因素考虑在内，SMS 表空间更适合用作目录。

另一个要考虑的因素是，将来是否需要扩大目录表空间。虽然某些平台支持扩大 SMS 容器的基本存储器，虽然可以使用重定向复原来扩大 SMS 表空间，但使用 DMS 表空间能简化添加新容器的工作。

### 工作负荷注意事项

在您的环境中由 DB2 管理的主要工作负荷类型，会影响您选择要使用的表空间类型以及要指定的页大小。联机事务处理 (OLTP) 工作负荷的特征是，事务需要对数据进行随机存取，并常常返回小的数据集。因为存取是随机的，并且是存取一页或几页，所以预取是不可能的。

使用设备容器的 DMS 表空间在这种情况下表现得最好。若不需要最大性能，使用文件容器的 DMS 表空间或 SMS 表空间也适用于 OLTP 工作负荷。若期望很少的顺序 I/O 或不期望它，则 CREATE TABLESPACE 语句上的 EXTENTSIZE 和 PREFETCHSIZE 参数的设置对于 I/O 的效率就显得不重要。

查询工作负荷的特征是，事务需要对数据进行顺序访问或部分顺序访问，并常常返回大的数据集。使用多个设备容器（每个容器都在单独的磁盘上）的 DMS 表空间，最有可能提供有效的并行预取。应该将 CREATE TABLESPACE 语句上的 PREFETCHSIZE 参数的值设置为 EXTENTSIZE 参数的值乘以设备容器数之积。这允许 DB2 以并行方式从所有的容器中预取。

查询工作负荷的一个合理的替代方法是使用文件（若文件系统有自己的预取的话）。这些文件可以是使用文件容器的 DMS 类型或 SMS 类型。注意，若使用 SMS，则需要将目录容器映射至单独的物理磁盘，以实现 I/O 并行性。

混合工作负荷的目标是：对于 OLTP 工作负荷，使单个 I/O 请求尽可能有效；而对于查询工作负荷，最大程度地提高并行 I/O 的效率。

确定表空间页大小的注意事项如下所示：

- 对于执行随机行读写操作的 OLTP 应用程序，通常优先考虑小一点的页大小，因为它为存储不需要的行而浪费的缓冲池空间较少。
- 对于一次访问大量连续行的 DSS 应用程序，大一点的页大小会更好，因为它将减少读取特定行数所需的输入 / 输出请求数。但是，也有例外。若行大小小于：

页大小 / 255

则每页上都会有浪费的空间（每页最多有 255 行）。在这种情况下，更小一点的页大小可能更合适。

- 更大一点的页大小可允许减少索引中的级别数。
- 越大的页，支持的行越长。
- 在缺省的 4 KB 页上，一个表只能有 500 列，而更大的页大小（8 KB、16 KB 和 32 KB）支持 1012 列。
- 表空间的最大大小与表空间的页大小成正比。在 *SQL Reference* 中讲述了这些限制。

### 选择 SMS 或 DMS 表空间

当确定应使用哪种类型的表空间存储数据时，有一些问题需要权衡一下。

#### SMS 表空间的优点：

- 直到需要时，系统才分配空间。
- 由于不必预定义容器，所以创建数据库需要的初始工作较少。

#### DMS 表空间的优点：

- 通过使用 ALTER TABLESPACE 语句，可添加容器来增加表空间的大小。现存数据自动在新的容器集合中重新平衡，以保持最优 I/O 效率。
- 根据存储的数据的类型，一个表可以分布在多个表空间中：
  - 长整数字段和 LOB 数据
  - 索引
  - 规则表数据

可能出于性能方面的考虑而想要分隔表数据，或想要增加存储的表数据量。例如，您可能拥有具有 64GB 规则表数据、64GB 索引数据和 2TB 长整数数据的一个表。若使用的是 8 KB 页，则表数据和索引数据可达 128 GB。若使用的是 16 KB 页，则表数据和索引数据可达 256 GB。若使用的是 32 KB 页，则表数据和索引数据可达 512 GB。

- 可控制数据在磁盘上的位置（若操作系统允许的话）。

- 若所有的表数据都位于单个表空间中，可以使用比删除和重新定义一个表更少的开销来删除和重新定义一个表空间。
- 通常，精心调整的一组 DMS 表空间的性能将优于 SMS 表空间。

**注：**在 Solaris 和 PTX (IBM NUMA-Q) 上，强烈建议对性能非常关键的工作负荷使用具有原始设备的 DMS 表空间。

通常，小型个人数据库用 SMS 表空间管理最容易。另一方面，对于大的、增长中的数据库，您可能只希望使用 SMS 表空间用作临时表空间和目录表空间，而使用具有多个容器的单独的 DMS 表空间用于每个表。另外，您可能想将长整数字段数据和索引存储在它们自己的表空间中。

若选择使用带设备容器的 DMS 表空间，您必须乐意调整和管理您的环境。有关更多信息，参考《管理指南：性能》中的“DMS 设备的性能注意事项”。

### 当将数据存放在 RAID 设备中时优化性能

本节描述当将数据存放在“磁盘冗余阵列”(RAID)设备中时如何优化性能。一般而言，应对使用 RAID 设备的每个表空间执行下列操作：

- 为使用 RAID 设备的表空间定义一个容器。
- 使该表空间的 EXTENTSIZE 等于或数倍于 RAID 条大小。
- 确保该表空间的 PREFETCHSIZE 为：
  - RAID 条大小与 RAID 并行设备数之积（或此积的整数倍），并且是
  - EXTENTSIZE 的一个倍数。
- 使用 DB2\_PARALLEL\_IO 注册表变量（参见 DB2\_PARALLEL\_IO）对表空间启用并行 I/O。
- 使用 DB2\_STRIPED\_CONTAINERS 注册表变量（参见第 115 页的 DB2\_STRIPED\_CONTAINERS）保证该表空间中的数据块边界是对齐的。

### DB2\_PARALLEL\_IO

当对表空间容器读写数据时，若数据库中的容器数大于 1，DB2 可以使用并行 I/O。但对于某些情况，对单个容器表空间启用并行 I/O 也有好处。例如，若容器是在由多个物理磁盘组成的单个 RAID 设备上创建的，可发出并行读写调用。

要强制对只有一个容器的表空间执行并行 I/O，可使用 DB2\_PARALLEL\_IO 注册表变量。可将此变量设置为 "\*"（星号），表示每个表空间，或者可将它设置为由逗号分开的表空间标识的列表。例如：

```
db2set DB2_PARALLEL_IO=*      {turn parallel I/O on for all table spaces}
db2set DB2_PARALLEL_IO=1,2,4,8 {turn parallel I/O on for table spaces 1, 2,
                                4, and 8}
```



在设置了注册表变量后，必须停止 DB2 (**db2stop**)，然后重新启动它 (**db2start**)，以使更改生效。

## DB2\_STRIPED\_CONTAINERS

目前，当创建 DMS 表空间容器（设备或文件）时，会在该容器开始处存储一个单页标记。其余页可由 DB2 存储数据，并分组成按数据块大小确定的数据块。

当使用 RAID 设备作为表空间容器时，建议用等于或数倍于 RAID 条大小的数据块大小创建表空间。但是，由于存在单页容器标记，因此数据块不会按 RAID 条排列，并且在执行 I/O 请求时可能需要访问比最优情况更多的物理磁盘。

可以创建 DMS 表空间容器，以便该标记在容器自己的（整个）数据块中存在。这可避免上述问题，但它需要在该容器中占用额外的数据块。要以这种方式创建容器，必须将 DB2 注册表变量 DB2\_STRIPED\_CONTAINERS 设置为 "ON"，然后停止并重新启动实例：

```
db2set DB2_STRIPED_CONTAINERS=ON
db2stop
db2start
```

（使用 CREATE TABLESPACE 或 ALTER TABLESPACE 语句）创建的任何 DMS 容器的标记都占用一个完整的数据块。现有容器保持不变。

要停止创建具有此属性的容器，复位该变量（缺省值为 NULL），然后停止并重新启动实例：

```
db2set DB2_STRIPED_CONTAINERS=
db2stop
db2start
```

“控制中心”和 LIST TABLESPACE CONTAINERS 命令不显示一个容器是否是作为条纹化容器创建的。它们使用标号“文件”或“设备”，这取决于该容器是如何创建的。要验证一个容器是否是作为条纹化容器创建的，可以使用 DB2DART 的 /DTSF 选项来转储表空间和容器信息，然后查看要了解的容器的类型字段。也可使用查询容器 API (**sqlbftcq** 和 **sqlbtcq**) 来创建一个显示该类型的简单应用程序。

---

## 联合数据库设计注意事项

当设计联合数据库时，考虑下列设计主题：

- 空间要求
- 确定网络优先级。

通常，可从一个联合数据库访问的数据不存储在该数据库中。对数据源表和视图的引用存储在系统目录中，但实际数据位于数据源中。这样，联合数据库所需的存储空间可能比常规数据库少。若由于整理系统差别或数据源中缺少某个函数，而必须在本地执行查询，则此一般规则就不适用。在这种情况下，应在 DB2 实现表以进行处理。

由于大多数联合系统数据通常位于分布在一个网络中的一个或多个数据源中，因此可考虑改变分配给 DB2 和该网络系统的资源。在将多一点的资源分配给 DB2 系统所在的网络，而将少一点的资源分配给数据库管理器本身之后，性能可能会提高。

---

## 第9章 设计分布式数据库

在 DB2 中，一个事务通常称为一个工作单元。工作单元是应用程序进程内可恢复的操作序列。数据库管理器使用它来确保数据库处于一致状态。对数据库的任何读取或写入都在一个工作单元内执行。

例如，银行事务可能涉及将存款从储蓄帐户转至支票帐户。当应用程序从储蓄帐户中减去一定金额之后，这种不一致状态将一致保持，直到将该金额加入支票帐户为止。在两个步骤都完成后，便到达一致点。可以提交更改，并可使其可用于其他应用程序。

当对数据库发出第一条 SQL 语句时，就启动了一个工作单元。应用程序必须发出 COMMIT 或 ROLLBACK 语句来结束该工作单元。COMMIT 语句使在工作单元内所做的所有更改成为永久性的。ROLLBACK 语句则从该数据库中消除这些更改。若应用程序正常结束，而未显式发出这两条语句中的任何一个，则自动提交该工作单元。若它在执行一个工作单元时异常结束，则自动回滚该工作单元。一旦发出 COMMIT 或 ROLLBACK，就不能停止它们。对于某些多线程应用程序或某些操作系统（如 Windows），若应用程序正常结束，而未显式发出这两条语句中的任何一个，则自动回滚工作单元。建议应用程序始终显式地提交或回滚，以完成工作单元。若部分工作单元未成功完成，则回滚更新，使参与的表保持该事务开始前的状态。这确保了请求既不会丢失，也不会重复。

下列主题提供其他信息：

- 『在一个事务中使用单个数据库』
- 第118页的『在单个事务中使用多个数据库』
- 第123页的『其他配置注意事项』
- 第126页的『了解两阶段提交过程』
- 第129页的『在两阶段提交期间校正问题』。

有关创建使用分布式数据库的应用程序的信息，参考 *Application Development Guide* 和 *CLI Guide and Reference*。

---

### 在一个事务中使用单个数据库

事务的最简单形式是在单个工作单元中只对一个数据库读取和写入。此类数据库访问称为远程工作单元。

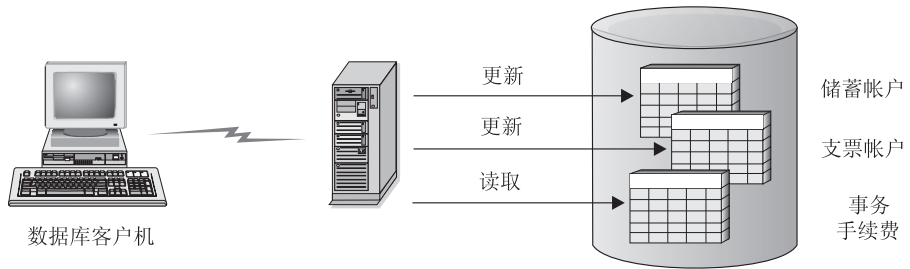


图 30. 在一个事务中使用单个数据库

图30显示了一个运行转帐应用程序的数据库客户机，该应用程序访问包含支票帐户和储蓄帐户表以及银行手续费表的数据库。该应用程序必须：

- 接受从用户界面指定的转帐金额
- 从储蓄帐户中减去该金额，然后确定新的余额
- 读取手续费表，以确定含有给定余额的储蓄帐户的事务手续费
- 从储蓄帐户中减去事务手续费
- 将转帐的金额添加至支票帐户
- 提交该事务（工作单元）。

要设置这样的应用程序，必须：

1. 在同一个数据库中为储蓄帐户、支票帐户和银行业务手续费表创建表（参见《管理指南：实现》中的“实现设计”）
2. 若实际位置相距较远，则设置数据库服务器以使用适当的通信协议，如《安装和配置补遗》所述
3. 若实际位置相距较远，则编目节点和数据库，以便在数据库服务器上标识该数据库，如《快速入门》书籍所述
4. 预编译您的应用程序以指定类型 1 连接；即在 `PRECOMPILE PROGRAM` 命令上指定 `CONNECT 1`（缺省值），如 *Application Development Guide* 所述。

## 在单个事务中使用多个数据库

当在单个事务中使用多个数据库时，对设置和管理环境的要求是不同的，这取决于在该事务中要更新的数据库的数量。

### 更新单个数据库

若您的数据分布在多个数据库上，您可能希望在从一个或多个数据库读取的同时更新一个数据库。可以在单个工作单元（事务）中执行此类访问。

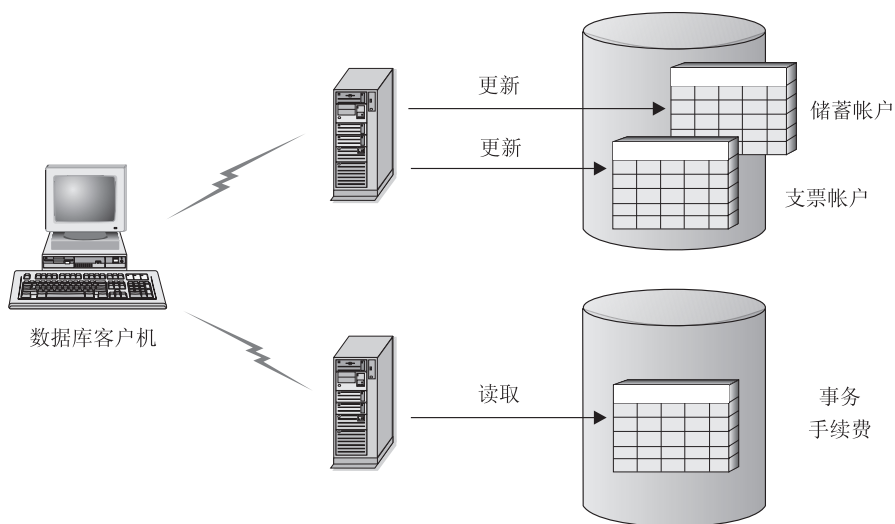


图 31. 在单个事务中使用多个数据库

图31显示了一个运行转帐应用程序的数据库客户机，该应用程序访问两个数据库服务器：一个包含支票和储蓄帐户，另一个包含银行业务手续费表。除了数据库数量和表的位置不同之外，此示例与第118页的图30中显示的示例相似。

要为此环境设置转帐应用程序，必须：

1. 在适当的数据库中创建必需的表（参见《管理指南：实现》中的“实现设计”
2. 若实际位置相距较远，则设置数据库服务器以使用适当的通信协议，如《安装和配置补遗》所述
3. 若实际位置相距较远，则编目节点和数据库，以便在数据库服务器上标识该数据库，如《快速入门》书籍所述
4. 预编译您的应用程序以指定类型 2 连接（即在 PRECOMPILE PROGRAM 命令上指定 CONNECT 2）和一阶段提交（即在 PRECOMPILE PROGRAM 命令上指定 SYNCPOINT ONEPHASE），如 *Application Development Guide* 所述。

若数据库位于主机或 AS/400 数据库服务器上，则需要 DB2 Connect 来与这些服务器连接。有关设置的信息，参考其中一本 DB2 Connect 《快速入门》书籍。有关使用 DB2 Connect 的信息，参考《DB2 Connect 用户指南》。

## 更新多个数据库

若您的数据分布在多个数据库上，则您可能想在单个事务中读取和更新多个数据库。此类数据库访问称为多站点更新。

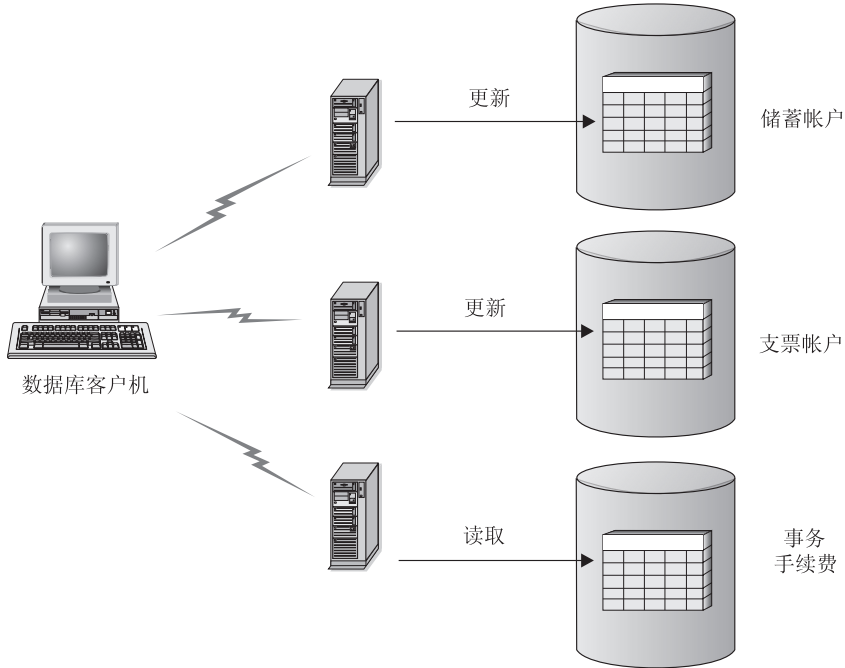


图 32. 在单个事务中更新多个数据库

图32显示了一个运行转帐应用程序的数据库客户机，该应用程序访问三个数据库服务器：一个包含支票帐户，另一个包含储蓄帐户，第三个包含银行业务手续费表。

要为此环境设置转帐应用程序，您有两种选择：

1. 使用事务管理器 (TM):

- a. 在适当的数据库中创建必需的表（参见《管理指南：实现》中的“实现设计”）。
- b. 若实际位置相距较远，则设置数据库服务器以使用适当的通信协议，如《安装和配置补遗》所述。
- c. 若实际位置相距较远，则编目节点和数据库，以便在数据库服务器上标识该数据库，如《快速入门》书籍所述。

- d. 预编译您的应用程序以指定类型 2 连接（即在 PRECOMPILE PROGRAM 命令上指定 CONNECT 2）和两阶段提交（即在 PRECOMPILE PROGRAM 命令上指定 SYNCPOINT TWOPHASE），如 *Application Development Guide* 所述。
  - e. 配置 DB2 事务管理器 (TM)，如『使用 DB2 事务管理器』中所述
2. 不使用事务管理器:
- a. 在适当的数据库中创建必需的表（参见《管理指南: 实现》中的“实现设计”。
  - b. 若实际位置相距较远，则设置数据库服务器以使用适当的通信协议，如《安装和配置补遗》所述。
  - c. 若实际位置相距较远，则编目节点和数据库，以便在数据库服务器上标识该数据库，如《快速入门》书籍所述。
  - d. 预编译您的应用程序以指定类型 2 连接（即在 PRECOMPILE PROGRAM 命令上指定 CONNECT 2）和一阶段提交（即在 PRECOMPILE PROGRAM 命令上指定 SYNCPOINT ONEPHASE），如 *Application Development Guide* 所述。

### 使用 DB2 事务管理器

数据库管理器提供事务管理器功能，这些功能可用于协调在单个工作单元内几个数据库的更新。数据库客户机自动协调工作单元，并使用事务管理器数据库来注册每个事务并跟踪其完成状态。

若您使用的是 XA 兼容的事务管理器，如 IBM TXSeries、BEA Tuxedo 或“Microsoft 事务处理服务器”，参见第133页的『第10章 针对事务管理器的设计』以获取完整的说明。

当使用用于基于 UNIX 的操作系统、Windows 操作系统或 OS/2 的 DB2 UDB 来协调事务时，必须满足特定配置需求。若您只使用 TCP/IP 来进行通信，且 DB2 UDB 和 DB2 OS/390 版是参与该事务的唯一数据库服务器，则配置就会简化。

**DB2 UDB 和 DB2 OS/390 版使用 TCP/IP 连通性:** 若以下所有陈述都适合于您的环境，则多站点更新的配置步骤就会简化。

- 与远程数据库服务器（包括 DB2 UDB OS/390 版）的所有通信都只使用 TCP/IP。
- 用于基于 UNIX 的系统、Windows 操作系统、OS/2 或 OS/390 的 DB2 UDB 是参与该事务的唯一数据库服务器。
- 未配置 DB2 Connect 同步点管理器 (SPM)。

DB2 Connect 同步点管理器是在创建 DB2 实例时自动配置的，它在下列情况下是必需的：

- 主机或 AS/400 数据库服务器使用 SNA 连通性，来进行多站点更新。
- XA 兼容的事务管理器（如 IBM TXSeries CICS）正在协调两阶段提交。

这适用于主机或 AS/400 数据库服务器使用 SNA 和 TCP/IP 连通性的情况。有关详细信息，参见第133页的『第10章 针对事务管理器的设计』。若您的环境不需要 DB2 Connect 同步点管理器，则可以通过在 DB2 Connect 服务器上发出 `db2 update dbm cfg using spm_name NULL` 命令来将其关闭。然后停止并重新启动 DB2。

将用作事务管理器数据库的数据库由数据库客户机的数据库管理器配置参数 *tm\_database* 确定。有关此配置参数的更多信息，参见《管理指南：性能》中的“配置 DB2”。当设置此配置参数时，考虑下列因素：

- 事务管理器数据库可以是：
  - 用于基于 UNIX 的系统、Windows 操作系统或 OS/2 的 DB2 UDB 的数据库
  - DB2 OS/390 版的版本 5 或更高版本的数据库。
- 若对 *tm\_database* 配置参数指定了值 1ST\_CONN，则与应用程序连接的第一个数据库被用作事务管理器数据库。

当使用 1ST\_CONN 时，必须小心。仅当能够比较容易地确保所有参与的数据库都正确编目时，才应使用此配置，即，如果：

- 启动该事务的数据库客户机位于包含参与数据库（包括事务管理器数据库）的同一个实例中。
- 您正在使用 DCE 目录服务来编目和管理对数据库的访问。

注意，若您的应用程序试图与正用作事务管理器数据库的数据库断开，则将接收到警告信息，且该连接将被挂起，直到提交该工作单元为止。

**其他环境：** 在您的环境中，如果：

- 与远程数据库服务器的通信不只使用 TCP/IP（例如，还使用 NETBIOS）
- 访问 DB2 MVS 版的版本 3 或版本 4、DB2 AS/400 版或 DB2 VM 版和 VSE 版
- 使用 SNA 访问 DB2 OS/390 版
- 使用 DB2 Connect 同步点管理器访问主机或 AS/400 数据库服务器



多站点更新的配置步骤更为复杂。

将用作事务管理器数据库的数据库由数据库客户机的数据库管理器配置参数 *tm\_database* 确定。有关此配置参数的更多信息，参见《管理指南：性能》中的“配置 DB2”。当设置此配置参数时，考虑下列因素：

- 事务管理器数据库可以是用于基于 UNIX 的系统、Windows 操作系统或 OS/2 的 DB2 UDB 的数据库。
- 若对 *tm\_database* 配置参数指定了值 1ST\_CONN，则与应用程序连接的第一个数据库被用作事务管理器数据库。

当使用 1ST\_CONN 时，必须小心。仅当能够比较容易地确保所有参与的数据库都正确编目时，才应使用此配置，即，如果：

- 启动该事务的数据库客户机位于包含参与数据库（包括事务管理器数据库）的同一个实例中。
- 您正在使用 DCE 目录服务来编目和管理对数据库的访问。

注意，若您的应用程序试图与正用作事务管理器数据库的数据库断开，则将接收到警告信息，且该连接将被挂起，直到提交该工作单元为止。

---

## 其他配置注意事项

当您设置环境时，应该考虑下列配置参数。有关设置这些参数的其他信息，参考《DB2 Connect 用户指南》。

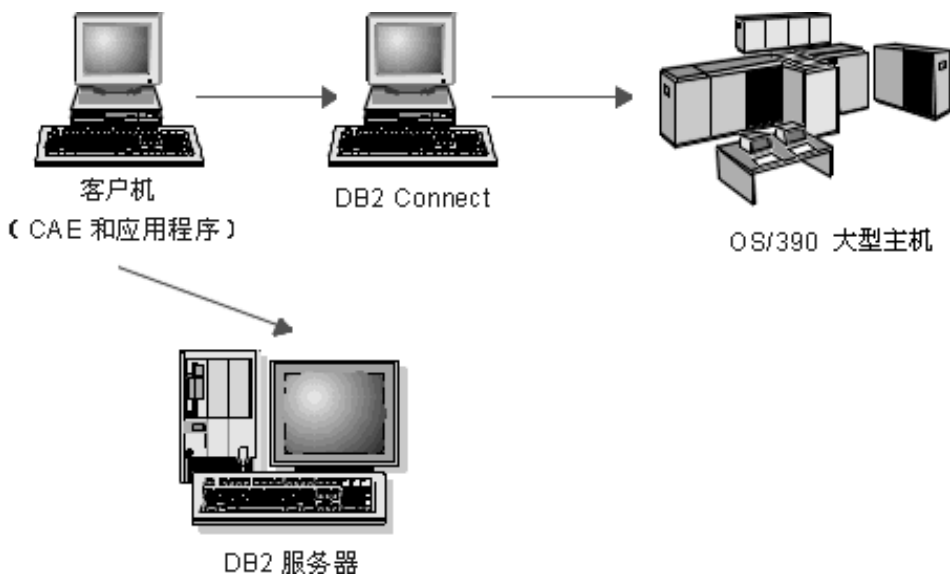


图 33. 配置注意事项

### 数据库管理器配置参数

- *tm\_database*  
此参数标识每个 DB2 实例的“事务管理器” (TM) 数据库的名称。
- *spm\_name*  
此参数向数据库管理器标识 DB2 Connect 同步点管理器实例的名称。为了使再同步成功，该名称必须在您的网络中是唯一的。
- *resync\_interval*  
此参数标识一个时间间隔（以秒计），在该时间间隔之后，“DB2 事务管理器”、DB2 服务器数据库管理器以及 DB2 Connect 同步点管理器或 DB2 UDB 同步点管理器应该重新尝试恢复任何未完成的不确定事务。
- *spm\_log\_file\_sz*  
此参数指定 SPM 日志文件的大小（以 4 KB 页计）。
- *spm\_max\_resync*  
此参数标识可同时执行再同步操作的代理程序数。
- *spm\_log\_path*  
此参数标识 SPM 日志文件的日志路径。

### 数据库配置参数

- *maxappls*

此参数指定允许的最大活动应用程序数。它的值必须等于或大于连接的应用程序数，加上可能正在同时完成一个两阶段提交或回滚的相同应用程序的数目，加上预期任何时候可能存在的不确定事务的数目之和。有关不确定事务的更多信息，参见第129页的『在两阶段提交期间校正问题』。

- *autorestart*

此数据库配置参数指定在需要时是否自动调用 RESTART DATABASE 例程。缺省值是 YES（即启用）。

包含不确定事务的数据库需要执行重新启动数据库操作来启动。若未启用 *autorestart*，则当断开与该数据库的最后一个连接时，下一个连接将失败并需要显式的 RESTART DATABASE 调用。这种情况将一直存在，直到事务管理器的再同步操作或管理员启动的试探性操作除去了不确定事务为止。当发出 RESTART DATABASE 命令时，若该数据库中有任何不确定事务，则将返回一条信息。管理员就可以使用 LIST INDOUBT TRANSACTIONS 命令和其他的命令行处理器命令来获取有关那些不确定事务的信息。

有关这些配置参数的更多信息，参考《管理指南：性能》。

## 在多站点更新中访问基于 LAN 的 DB2 通用数据库服务器的主机或 AS/400 应用程序

“DB2 通用数据库”不支持使用 TCP/IP 连通性从主机或 AS/400 数据库客户机进行多站点更新。在此情况下，只支持 SNA（系统网络体系结构）连通性。多站点更新需要 DB2 同步点管理器。在此方案中，不使用 DB2 Connect。

从主机或 AS/400 数据库客户机访问的数据库服务器不必在带有 DB2 同步点管理器的本地工作站上。主机或 AS/400 数据库客户机可以将 DB2 同步点管理器工作站用作一个中间网关来与 DB2 UDB 服务器连接。这允许您将 DB2 同步点管理器工作站隔离在一个安全的环境中，而实际的 DB2 UDB 服务器位于您公司网络中的远程位置。这也允许 DB2 公共服务器版本 2 数据库参与从主机或 AS/400 数据库客户机发出的多站点更新。

具体步骤如下所示：

- 在主机或 AS/400 应用程序将要直接访问的工作站上：
  1. 安装“DB2 通用数据库企业版”或“DB2 通用数据库扩充企业版”，以便向主机或 AS/400 数据库客户机提供多站点更新支持。
  2. 在同一个系统上创建数据库实例。例如，可以使用缺省实例 DB2，或使用如下命令来创建新实例：

```
db2icrt myinstance
```
  3. 按要求提供许可信息。

4. 确保注册表的值 DB2COMM 包括值 APPC。
  5. 按要求配置 SNA 通信。当使用受支持的 IBM SNA 产品时，将根据 *spm\_name* 数据库管理器配置参数的值自动创建 DB2 同步点管理器所需的 SNA 简要表。任何其他受支持的 SNA 堆栈都需要手工配置。有关详细信息，参考《安装和配置补遗》。
  6. 确定要对 *spm\_name* 数据库管理器配置参数指定的值。此参数是在创建 DB2 实例时预先配置的，并且是从机器的 TCP/IP 主机名派生的。若它在您的环境中可接受且是唯一的，则不要更改它。
  7. 必要时，使用 UPDATE DATABASE MANAGER CONFIGURATION 命令来在“DB2 通用数据库”服务器上更新 *spm\_name*。
  8. 配置此 DB2 工作站连接远程 DB2 UDB 服务器所需的通信（若有的话）。
  9. 配置远程 DB2 UDB 服务器连接此 DB2 服务器所需的通信。
  10. 当首次启动 SPM 时，停止“DB2 通用数据库”服务器上的数据库管理器，再重新启动它。

应该能够从此 DB2 UDB 工作站与远程 DB2 UDB 服务器连接。
- 在主机或 AS/400 数据库客户机将要访问的每个远程 DB2 UDB 服务器上：
    1. 配置带有 DB2 同步点管理器的远程 DB2 UDB 工作站与此 DB2 UDB 服务器连接所需的通信。
    2. 停止并重新启动数据库管理器。

---

## 了解两阶段提交过程

第127页的图34举例说明了多站点更新包括的步骤。若两阶段提交过程期间发生错误，则了解一个事务是如何管理的将有助于您解决问题。

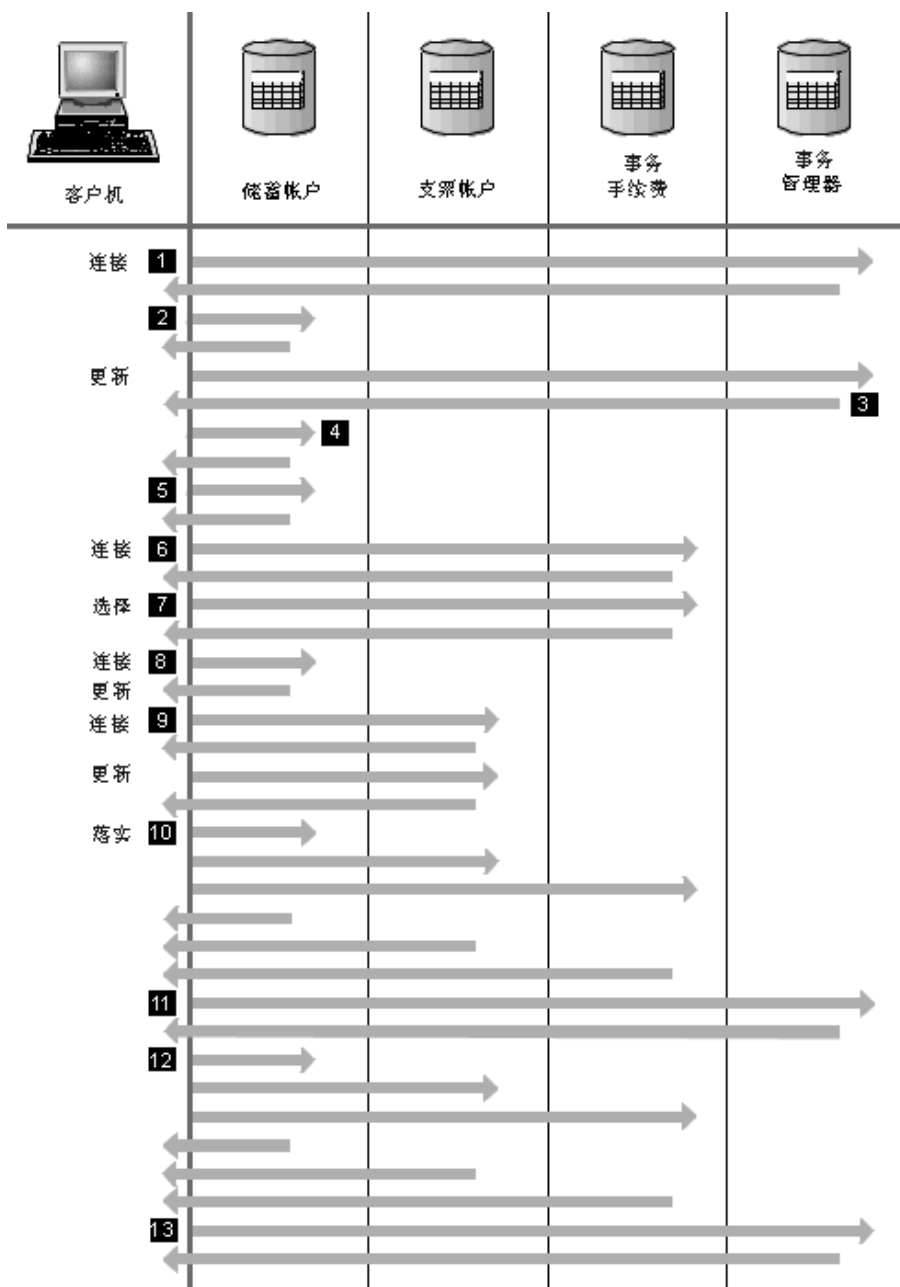


图 34. 更新多个数据库

**0** 为两阶段提交准备该应用程序。此过程可以通过预编译选项来完成（有关

详细信息，参考 *Application Development Guide* )。此过程也可以通过 DB2 CLI (调用层接口) 配置来完成 (有关详细信息，参考 *CLI Guide and Reference* )。

- 1** 当数据库客户机要与 SAVINGS\_DB 数据库连接时，它首先在内部与事务管理器 (TM) 数据库连接。该 TM 数据库将确认返回至数据库客户机。若将数据库管理器配置参数 *tm\_database* 设置为 *1ST\_CONN*，则 SAVINGS\_DB 在此应用程序实例期间成为事务管理器数据库。
- 2** 建立与 SAVINGS\_DB 数据库的连接，并确认该连接。
- 3** 数据库客户机开始更新 SAVINGS\_ACCOUNT 表。这开始工作单元。TM 数据库提供该工作单元的一个事务标识，以应答数据库客户机。注意，当运行该工作单元中的第一条 SQL 语句时，而不是在建立连接期间，要对该工作单元进行注册。
- 4** 当接收到该事务标识之后，数据库客户机向包含 SAVINGS\_ACCOUNT 表的数据库注册该工作单元。将一个应答发送回客户机，以指示该工作单元已成功注册。
- 5** 以正常方式处理对 SAVINGS\_DB 数据库发出的 SQL 语句。当使用在一个程序中嵌入的 SQL 语句时，对每条语句的应答都会在 SQLCA 中返回。(在 *Application Development Guide* 和 *SQL Reference* 中描述了 SQLCA。)
- 6** 当在工作单元内第一次访问该数据库期间，在包含 TRANSACTION\_FEE 表的该 FEE\_DB 数据库中注册该事务标识。
- 7** 以正常方式处理对 FEE\_DB 数据库发出的任何 SQL 语句。
- 8** 可以适当设置该连接，来对 SAVINGS\_DB 数据库运行附加的 SQL 语句。由于已向 SAVINGS\_DB 数据库注册了该工作单元 **4**，因此数据库客户机不必再次执行注册步骤。
- 9** 连接 CHECKING\_DB 数据库，并遵循 **6** 和 **7** 中描述的规则来使用它。
- 10** 当数据库客户机请求提交工作单元时，会将一条准备信息发送至参与该工作单元的所有数据库。每个数据库都将一个“PREPARED”记录写至它的日志文件，并答复数据库客户机。
- 11** 在数据库客户机接收到来自所有数据库的肯定响应之后，它会向事务管理器数据库发送一条信息，以通知数据库现在已准备提交工作单元 (PREPARED)。该事务管理器数据库则将一个“PREPARED”记录写入它的日志文件，并发送答复，以通知该客户机可以启动提交过程的第二阶段。
- 12** 在提交过程的第二阶段期间，数据库客户机会向所有参与数据库发送一条信息，告知它们要提交。每个数据库则将一个“COMMITTED”记录写入它

的日志文件，并释放为此工作单元挂起的锁定。当该数据库完成对更改的提交时，它会将一个答复发送至客户机。

- 13** 在数据库客户机接收到来自所有参与数据库的肯定应答之后，它会向事务管理器数据库发送一条信息，以通知数据库该工作单元已完成。于是，事务管理器数据库将一个“COMMITTED”记录写至其日志文件，指示该工作单元已完成，并答复该客户机，指示它已完成。

---

## 在两阶段提交期间校正问题

从错误状态恢复是一个与应用程序编程、系统管理、数据库管理和系统操作相关的正常任务。将数据库分布在几个远程服务器上会增加由于网络或通信故障而导致错误的可能性。为确保数据完整性，数据库管理器提供了两阶段提交过程，这在第126页的『了解两阶段提交过程』中已有说明（第 **10**、**11** 和 **12** 点）。下面解释数据库管理器如何在两阶段提交过程期间处理错误：

- **第一阶段的错误**

若一个数据库应答“它准备提交工作单元失败”，则数据库客户机将在提交过程的第二阶段期间回滚该工作单元。在这种情况下，不会向事务管理器数据库发送准备信息。

在第二个阶段期间，客户机会向所有在第一阶段成功地准备了提交的参与数据库发送一条回滚信息。于是，每个数据库将一个“ABORT”记录写入它的日志文件，并释放为此工作单元挂起的锁定。

## • 第二阶段的错误

此阶段中的错误处理取决于第二阶段将提交还是回滚该事务。若第一阶段遇到错误，则第二阶段将只回滚该事务。

若其中一个参与数据库提交工作单元失败（可能由于通信故障），则事务管理器数据库将对失败的数据库重试提交。然而，将通过 SQLCA 通知应用程序提交成功。DB2 将确保提交数据库服务器中尚未提交的事务。数据库管理器配置参数 *resync\_interval*（参见《管理指南：性能》中的“配置 DB2”）用于指定事务管理器数据库在试图提交工作单元期间要等待的时间。在数据库服务器上挂起所有锁定，直到提交工作单元为止。

若事务管理器数据库失败，它将在重新启动数据库时使该工作单元再同步。再同步过程将试图完成所有不确定事务；即，已完成第一阶段但未完成第二阶段提交过程的那些事务。与事务管理器数据库相关联的数据库管理器通过下列各项执行再同步：

1. 与指示在提交过程的第一阶段期间“已准备” (PREPARED) 提交的数据库连接。
2. 试图提交那些数据库中的不确定事务。（若找不到不确定事务，则数据库管理器假定在提交过程的第二阶段期间该数据库已成功提交该事务。）
3. 当在参与数据库中提交了所有不确定事务之后，再提交事务管理器数据库中的不确定事务。

若其中一个参与数据库失败并重新启动，则此数据库的数据库管理器将查询事务管理器数据库，以获知此事务的状态，来确定是否应该回滚该事务。若在日志中找不到该事务，则数据库管理器假定该事务已被回滚，且该事务将回滚此数据库中的不确定事务。否则，该数据库等待事务管理器数据库发出提交请求。

若该事务已由事务处理监视器（XA 兼容的事务管理器）协调，则数据库将始终依靠 TP 监视器来启动该再同步。

若由于某种原因您不能等待事务管理器自动解决不确定的事务，则您可以执行一些操作，以手工方式解决它们。此手工过程有时称为“作试探性决定”。有关手工恢复不确定事务的更多信息，参见第143页的『作试探性决定』。

## 若 **AUTORESTART=OFF**，则再同步不确定事务

有关“DB2 通用数据库”两阶段提交环境中的配置注意事项，参见第123页的『其他配置注意事项』。

尤其是，若 *autorestart* 数据库配置参数设置为 OFF，且 TM 或 RM 数据库中存在不确定事务，则需要执行 **RESTART DATABASE** 命令，以启动再同步过程。当



从命令行处理器发出 `RESTART DATABASE` 命令时，将使用不同的会话。若您从同一个会话中重新启动另一个数据库，则先前重新启动数据库命令所建立的连接将被断开，并且必须再次重新启动。在 `LIST INDOUBT TRANSACTIONS` 命令不再返回更多不确定事务之后，发出 `TERMINATE` 命令来断开该连接。



---

## 第10章 针对事务管理器的设计

若有除 DB2 数据库以外的资源，且希望它参与两阶段提交事务，您可能要在 XA 兼容的事务管理器中使用您的数据库。若您的事务只访问 DB2 数据库，则应该使用 DB2 事务管理器，如第120页的『更新多个数据库』所述。

下列主题可以辅助您将数据库管理器与 XA 兼容的事务管理器（如 IBM TXSeries CICS、IBM TXSeries Encina、BEA Tuxedo 或“Microsoft 事务处理服务器”）结合使用：

- 第134页的『X/Open 分布式事务处理模型』
- 第137页的『将数据库设置为资源管理器』
- 第137页的『xa\_open 和 xa\_close 字符串的用法』
- 第137页的『DB2 版本 7 的新 xa\_open 字符串格式』
- 第139页的『TPM 和 TP\_MON\_NAME 值』
- 第142页的『较早版本的 DB2 的 xa\_open 字符串格式』
- 第142页的『更新主机或 AS/400 数据库服务器』
- 第142页的『数据库连接注意事项』
- 第143页的『作试探性决定』
- 第145页的『安全性注意事项』
- 第146页的『配置注意事项』
- 第147页的『受支持的 XA 函数』
- 第149页的『XA 接口的问题确定』
- 第150页的『配置 XA 事务管理器以使用 DB2 UDB』
- 第154页的『配置 Microsoft 事务处理服务器』。

若您使用的是 XA 兼容的事务管理器，或正在实现该程序，请从我们的技术支持 Web 站点获取更多信息：

<http://www.ibm.com/software/data/db2/library/>

到达那里之后，选择“DB2 Universal Database”，然后使用关键字“XA”搜索该 Web 站点，以获取有关 XA 兼容的事务管理器的最新可用信息。

## X/Open 分布式事务处理模型

“X/Open 分布式事务处理” (DTP) 模型包括三个相关的组件:

- 『应用程序 (AP)』
- 第135页的『事务管理器 (TM)』
- 第136页的『资源管理器 (RM)』.

图35举例说明了此模型, 并显示了这些组件之间的关系。

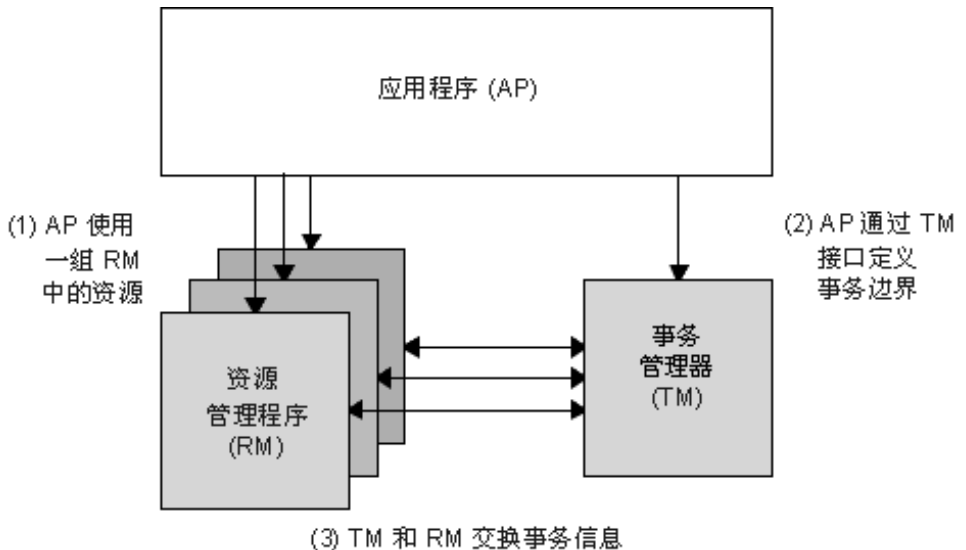


图 35. X/Open 分布式事务处理 (DTP) 模型

### 应用程序 (AP)

应用程序 (AP) 定义事务边界, 并定义构成该事务的应用程序特定的操作。

例如, CICS\* 应用程序也许要访问资源管理器 (RM), 如数据库和 “CICS 瞬时数据队列”, 并使用编程逻辑来操纵数据。通过针对该资源管理器的函数调用, 将每个访问请求传送至适当的 RM。对于 DB2, 这些可能是由 DB2 预编译程序为每条 SQL 语句生成的函数调用, 或是直接由程序员使用该 API 来编码的数据库调用。

事务管理器 (TM) 产品通常包括用于运行用户应用程序的事务处理 (TP) 监视器。TP 监视器提供 API 以允许应用程序启动和结束事务, 在希望运行该应用程序的多

个用户之间执行应用程序的调度和负载平衡。分布式事务处理 (DTP) 环境中的应用程序实际上是用户应用程序和 TP 监视器的组合。

要实现有效的联机事务处理 (OLTP) 环境, TP 监视器会在启动时预先分配大量服务器进程, 然后在多个用户事务之间调度和再利用它们。这样通过允许使用更少的服务器进程及其对应的 RM 进程来支持更多的并行用户, 节省了系统资源。再利用这些进程还避免了为每个用户的事务或程序在 TM 和 RM 中启动一个进程的开销。(一个程序调用一个或多个事务。) 这还意味着这些服务器进程对于 TM 和 RM 是真正的“用户进程”。它隐含安全性管理和应用程序编程。有关详细信息, 参见第145页的『安全性注意事项』。

从 TP 监视器可进行下列类型的事务:

- 非 XA 事务

这些事务涉及由于未定义给 TM 而未在 TM 的两阶段提交协议下协调的 RM。若该应用程序需要访问不支持 XA 接口的 RM, 这可能是需要的。TP 监视器只是提供有效的应用程序调度和负载平衡。由于 TM 不明确“打开”RM 来进行 XA 处理, 所以 RM 将此应用程序作为在非 DTP 环境中运行的任何其他应用程序来处理。

- 全局事务

这些事务涉及定义给 TM 并在 TM 的两阶段提交控制下的 RM。全局事务是可能涉及一个或多个 RM 的工作单元。事务分支是 TM 和 RM 之间支持全局事务的工作的一部分。当通过一个或多个由 TM 协调的应用程序进程来访问多个 RM 时, 一个全局事务可能有多个事务分支。

当多个应用程序进程中的每一个都访问 RM 时, 它们就象在单独的全局事务中一样, 存在松散耦合的全局事务, 但是那些应用程序处于 TM 的协调下。每个应用程序进程在一个 RM 内将有它自己的事务分支。当 AP、TM 或 RM 中任何一个请求提交或回滚时, 会一起完成事务分支。该应用程序负责确保在这些分支之间不发生资源死锁。(注意, DB2 事务管理器为使用 SYNCPOINT(TWOPHASE) 选项准备的应用程序执行的事务协调大致等效于这些松散耦合的全局事务。参见第120页的『更新多个数据库』。

当多个应用程序进程轮流在一个 RM 中的相同事务分支下工作时, 就会存在紧密耦合的全局事务。对于 RM, 这两个应用程序进程是一个单独的实体。RM 必须确保在事务分支内不发生资源死锁。

## 事务管理器 (TM)

事务管理器 (TM) 向事务指定标识符, 监视它们的进程, 并负责处理事务的完成和失败。事务分支标识符 (称为 XID) 由 TM 指定, 以标识一个 RM 内的全局事务和特定分支。它是一个 TM 中的日志与一个 RM 中的日志之间的相关标记。两阶

段提交或回滚需要 XID，以便在系统启动时执行再同步操作（也称为再同步（*resync*）），或在需要时允许管理员执行试探操作（也称为手工干预）。

在 TP 监视器启动后，它请求 TM 打开一组应用程序服务器已定义的所有 RM。TM 将 **xa\_open** 调用传送至 RM，以便可以将它们初始化以进行 DTP 处理。作为此启动过程的一部分，TM 执行再同步以恢复所有的不确定事务。不确定事务是处于未确定状态的全局事务。当成功完成两阶段提交协议的第一阶段（即，准备阶段）之后 TM（或至少一个 RM）成为不可用时，就会出现此情况。在 TM 可以使自己的日志与再次变成可用的 RM 的日志相符合之前，RM 不会知道要提交还是回滚它的事务分支。要执行再同步操作，TM 会一次或多次地将 **xa\_recover** 调用发出至每个 RM，以标识所有不确定事务。TM 将这些回答与它自己的日志中的信息作比较，以确定应该通知 RM **xa\_commit** 那些事务，还是 **xa\_rollback** 那些事务。若通过管理员执行试探操作，RM 已经提交或回滚了它的不确定事务的分支，则 TM 向该 RM 发出 **xa\_forget** 调用来完成再同步操作。

当用户应用程序请求提交或回滚时，它必须使用 TP 监视器或 TM 提供的 API，以便 TM 可以在所有参与的 RM 之间协调提交和回滚。例如，当 CICS 应用程序发出 CICS SYNCPOINT 请求以提交一个事务时，CICS XA TM（在“Encina 服务器”中实现）将发出 XA 调用，如 **xa\_end**、**xa\_prepare**、**xa\_commit** 或 **xa\_rollback**，以请求 RM 提交或回滚该事务。若只涉及一个 RM，或者若 RM 回答它的分支是只读的，则 TM 可选择使用单阶段提交以替代两阶段提交。

## 资源管理器 (RM)

资源管理器 (RM) 提供对共享资源（如数据库）的访问。

作为数据库的资源管理器的 DB2 可以参与由 XA 兼容的 TM 协调的全局事务。根据 XA 接口的要求，数据库管理器提供 **xa\_switch\_t** 类型的 **db2xa\_switch** 外部 C 变量，以将 XA 开关结构返回给 TM。此数据结构包含由该 TM 调用的各种 XA 例程的地址和 RM 的操作特征。有关数据库管理器支持的 XA 函数的更多信息，参见第147页的『受支持的 XA 函数』。

RM 可使用两种方法来对它参与每个全局事务进行注册：静态注册和动态注册：

- 静态注册要求 TM（为每个事务）将 **xa\_start**、**xa\_end** 和 **xa\_prepare** 等一系列调用发出至为服务器应用程序定义的所有 RM，而不管给定的 RM 是否正由该事务使用。若并非每个事务都包括每个 RM，则这是低效的，并且低效的程度与已定义的 RM 的数目成比例。
- 动态注册（由 DB2 使用）灵活且高效。仅当 RM 接收到对其资源的请求时，该 RM 才使用 **ax\_reg** 调用向 TM 注册。注意，即使只定义了一个 RM 时，或当每个事务都使用每个 RM 时，此方法也不存在性能方面的缺点，因为 **ax\_reg** 和 **xa\_start** 调用在 TM 中具有相似的路径。

XA 接口提供 TM 和 RM 之间的双向通信。它是两个 DTP 软件组件之间的系统级接口，而不是应用程序开发人员编码的普通应用程序接口。但是，应用程序开发人员应该熟悉该 DTP 软件组件所利用的编程限制。有关 X/Open XA 接口编程注意事项的信息，参考 *Application Development Guide*。

虽然 XA 接口是不变的，但是每个 XA 兼容的 TM 可能具有集成 RM 的产品特定方式。有关将 DB2 产品作为资源管理器与特定的事务管理器集成的信息，参见适当的 TM 产品文档。第150页的『配置 XA 事务管理器以使用 DB2 UDB』中提供了有关最流行的 TP 监视器的综合信息。

---

## 将数据库设置为资源管理器

将每个数据库定义为事务管理器 (TM) 的一个单独的资源管理器 (RM)，且必须用 `xa_open` 字符串标识该数据库。有关 DB2 的 `xa_open` 字符串格式的描述，参见『`xa_open` 和 `xa_close` 字符串的用法』。

### `xa_open` 和 `xa_close` 字符串的用法

数据库管理器 `xa_open` 字符串接受两种格式。一种格式是 DB2 版本 7 新增的。第二种格式由较早版本的 DB2 使用，为后面级兼容性而保留。新的实现应使用新格式，在可能的時候，应将旧实现迁移为新格式。DB2 的将来版本可能不支持旧的 `xa_open` 字符串格式。有关原始 `xa_open` 字符串格式的信息，参见第142页的『较早版本的 DB2 的 `xa_open` 字符串格式』。

当将一个数据库设置为资源管理器时，不需要 `xa_close` 字符串。若提供了此字符串，数据库管理器会忽略它。

### DB2 版本 7 的新 `xa_open` 字符串格式

下列 `xa_open` 字符串格式是 DB2 版本 7 新增的：

```
parm_id1 = <parm value>,parm_id2 = <parm value>, ...
```

以何次序指定这些参数并不重要。下表描述了 `parm_id` 的有效值。

表 22. `parm_id` 的有效值

参数名	值	必要的?	区分大小写?	缺省值
DB	数据库别名	是	否	无
应用程序用来访问数据库的数据库别名。				
UID	用户标识	否	是	无
有权与数据库连接的用户标识。若指定密码，则是必需的。				
PWD	密码	否	是	无

表 22. *parm\_id* 的有效值 (续)

参数名	值	必要的?	区分大小写?	缺省值
	与用户标识相关联的密码。若指定用户标识, 则是必需的。			
TPM	事务处理监视器名	否	否	无
	正在使用的 TP 监视器的名称。有关受支持的值, 参见第139页的『TPM 和 TP_MON_NAME 值』。可指定此参数以允许多个 TP 监视器使用单一 DB2 实例。指定的值将覆盖 <i>tp_mon_name</i> 数据库管理器配置参数中指定的值。			
AXLIB	包含 TP 监视器的 <b>ax_reg</b> 和 <b>ax_unreg</b> 函数的库。	否	是	无
	DB2 使用此值来获取必需的 <b>ax_reg</b> 和 <b>ax_unreg</b> 函数的地址。可使用它来覆盖根据 TPM 参数假定的值, 它也可以由未出现在 TPM 列表中的 TP 监视器使用。			
CHAIN_END	<i>xa_end</i> 链接标志。有效值是 T、F 或没有值。	否	否	F
	XA_END 链接是 DB2 可用来减少网络流的优化。若 TP 监视器环境可以保证在调用 <b>xa_end</b> 之后, 将立即在同一线程或进程中调用 <b>xa_prepare</b> , 且 CHAIN_END 为开, 则 <i>xa_end</i> 标志将与 <b>xa_prepare</b> 命令链接, 从而消去一个网络流。值 T 表示 CHAIN_END 为开; 值 F 表示 CHAIN_END 为关; 未指定值表示 CHAIN_END 为开。可使用此参数来覆盖从指定的 TPM 值派生的设置。			
SUSPEND_CURSOR	指定当挂起事务控制线程时是否保留游标。有效值是 T、F 或没有值。	否	否	F
	挂起事务分支的 TP 监视器可以将挂起的线程或进程重用于其他事务。在这些情况下, 必须关闭游标, 以使新事务不会继承它们。当继续执行挂起的事务时, 应用程序必须再次获取游标。若 SUSPEND_CURSOR 为开, 则不关闭任何打开的游标, 但不能将线程或进程重用于其他事务。只允许继续执行挂起的事务。值 T 表示 SUSPEND_CURSOR 为开; 值 F 表示 SUSPEND_CURSOR 为关; 未指定值表示 SUSPEND_CURSOR 为开。可使用此参数来覆盖从指定的 TPM 值派生的设置。			
HOLD_CURSOR	指定是否跨事务提交保持游标。有效值是 T、F 或没有值。	否	否	F



表 22. *parm\_id* 的有效值 (续)

参数名	值	必要的?	区分大小写?	缺省值
	通常, TP 监视器将线程或进程重用于多个应用程序。要确保新装入的应用程序不会继承由先前应用程序打开的游标, 游标在提交之后被关闭。若 HOLD_CURSOR 为开, 则跨事务提交保持游标。值 T 表示 HOLD_CURSOR 为开; 值 F 表示 HOLD_CURSOR 为关; 未指定值表示 HOLD_CURSOR 为开。可使用此参数来覆盖从指定的 TPM 值派生的设置。			

## TPM 和 TP\_MON\_NAME 值

*xa\_open* 字符串 TPM 参数和 *tp\_mon\_name* 数据库管理器配置参数用来向 DB2 指示正在使用的 TP 监视器。*tp\_mon\_name* 值适用于整个 DB2 实例。TPM 参数仅适用于特定 XA 资源管理器。TPM 值覆盖 *tp\_mon\_name* 参数。TPM 和 *tp\_mon\_name* 参数的有效值如下所示:

表 23. TPM 和 *tp\_mon\_name* 的有效值

TPM 值	TP 监视器产品	内部设置
CICS	IBM TxSeries CICS	AXLIB=libEncServer (对于 Windows) =/usr/lpp/encina/lib/libEncServer (对于基于 UNIX 的系统) HOLD_CURSOR=T CHAIN_END=T SUSPEND_CURSOR=F
ENCINA	IBM TxSeries Encina 监视器	AXLIB=libEncServer (对于 Windows) =/usr/lpp/encina/lib/libEncServer (对于基于 UNIX 的系统) HOLD_CURSOR=F CHAIN_END=T SUSPEND_CURSOR=F
MQ	IBM MQSeries	AXLIB=mqmax (对于 Windows) =/usr/mqm/lib/libmqmax.a (对于 AIX) =/opt/mqm/lib/libmqmax.a (对于 Solaris) HOLD_CURSOR=F CHAIN_END=F SUSPEND_CURSOR=F
CB	IBM Component Broker	AXLIB=somtrx1i (对于 Windows) =libsomtrx1 (对于基于 UNIX 的系统) HOLD_CURSOR=F CHAIN_END=T SUSPEND_CURSOR=F

表 23. TPM 和 *tp\_mon\_name* 的有效值 (续)

TPM 值	TP 监视器产品	内部设置
SF	IBM San Francisco	AXLIB=ibmsfDB2 HOLD_CURSOR=F CHAIN_END=T SUSPEND_CURSOR=F
TUXEDO	BEA Tuxedo	AXLIB=libtux HOLD_CURSOR=F CHAIN_END=F SUSPEND_CURSOR=F
MTS	Microsoft 事务处理服务器	并不一定要为 MTS 配置 DB2。DB2 的 ODBC 驱动程序自动检测 MTS。
JTA	Java 事务 API	并不一定要为“企业 Java 服务器”(EJS) (例如 IBM WebSphere) 配置 DB2。DB2 的 JDBC 驱动程序自动检测此环境。

## 示例

- 您正在 Windows NT 上使用 IBM TxSeries CICS。TxSeries 文档指示您需要使用值 `libEncServer:C` 来配置 *tp\_mon\_name*。这仍是可接受的格式；然而，对于对于 DB2 UDB 或 DB2 Connect 版本 7，您可以选择：
  - 指定值为 CICS 的 *tp\_mon\_name* (建议用于此方案)：
 

```
db2 update dbm cfg using tp_mon_name CICS
```

对于向 CICS 定义的每个数据库，在“区域” → “资源” → “产品” → XAD → “资源管理器初始化字符串”中，指定：

```
db=dbalias,uid=userid,pwd=password
```
  - 对于向 CICS 定义的每个数据库，在“区域” → “资源” → “产品” → XAD → “资源管理器初始化字符串”中，指定：
 

```
db=dbalias,uid=userid,pwd=password,tpm=cics
```
- 您正在 Windows NT 上使用 IBM MQSeries。MQSeries 文档指示您需要使用值 `mqmax` 来配置 *tp\_mon\_name*。这仍是可接受的格式；然而，对于对于 DB2 UDB 或 DB2 Connect 版本 7，您可以选择：
  - 指定值为 MQ 的 *tp\_mon\_name* (建议用于此方案)：

```
db2 update dbm cfg using tp_mon_name MQ
```

对于向 CICS 定义的每个数据库，在“区域” → “资源” → “产品” → XAD → “资源管理器初始化字符串”中，指定：

```
uid=userid,db=dbalias,pwd=password
```

- 对于向 CICS 定义的每个数据库，在“区域” → “资源” → “产品” → XAD → “资源管理器初始化字符串”中，指定：

```
uid=userid,db=dbalias,pwd=password,tpm=mq
```

3. 您正在 Windows NT 上同时使用 IBM TxSeries CICS 和 IBM MQSeries。正在使用单个 DB2 实例。在此方案中，应按如下方法进行配置：

- a. 对于向 CICS 定义的每个数据库，在“区域” → “资源” → “产品” → XAD → “资源管理器初始化字符串”中，指定：

```
pwd=password,uid=userid,tpm=cics,db=dbalias
```

- b. 对于在队列管理器特性中定义成资源的每个数据库，将 XaOpenString 指定为：

```
db=dbalias,uid=userid,pwd=password,tpm=mq
```

4. 您正在 Windows NT 上开发您自己的 XA 兼容的事务管理器 (XA TM)，您想要告知 DB2 库 "myaxlib" 有必需的函数 **ax\_reg** 和 **ax\_unreg**。库 "myaxlib" 在 PATH 语句中指定的目录中。您可以选择：

- 指定值为 myaxlib 的 *tp\_mon\_name*：

```
db2 update dbm cfg using tp_mon_name myaxlib
```

并且，对于向 XA TM 定义的每个数据库，指定 xa\_open 字符串：

```
db=dbalias,uid=userid,pwd=password
```

- 对于向 XA TM 定义的每个数据库，指定 xa\_open 字符串：

```
db=dbalias,uid=userid,pwd=password,axlib=myaxlib
```

5. 您正在 Windows NT 上开发您自己的 XA 兼容的事务管理器 (XA TM)，您想要告知 DB2 库 "myaxlib" 有必需的函数 **ax\_reg** 和 **ax\_unreg**。库 "myaxlib" 在 PATH 语句中指定的目录中。您还想启用 XA END 链接。您可以选择：

- 对于向 XA TM 定义的每个数据库，指定 xa\_open 字符串：

```
db=dbalias,uid=userid,pwd=password,axlib=myaxlib,chain_end=T
```

- 对于向 XA TM 定义的每个数据库，指定 xa\_open 字符串：

```
db=dbalias,uid=userid,pwd=password,axlib=myaxlib,chain_end
```

## 较早版本的 DB2 的 xa\_open 字符串格式

较早版本的 DB2 使用这里描述的 xa\_open 字符串格式。为了进行兼容，仍支持此格式。在可能的時候，应将应用程序迁移至新格式（参见第137页的『DB2 版本 7 的新 xa\_open 字符串格式』）。

将每个数据库定义为事务管理器 (TM) 的一个单独的资源管理器 (RM)，且必须用 xa\_open 字符串标识该数据库，该字符串的语法如下：

```
"database_alias<,userid,password>"
```

*database\_alias* 用于指定数据库的别名。除非您在创建数据库之后明确地编目了一个别名，否则，别名与数据库名相同。用户名和密码是可选的（这取决于认证方法），它们将用于向数据库提供认证信息。

当将一个数据库设置为资源管理器时，不需要 xa\_close 字符串。若提供了此字符串，数据库管理器会忽略它。

## 更新主机或 AS/400 数据库服务器

主机和 AS/400 数据库服务器可能是可以更新的，这取决于“XA 事务管理器”的体系结构。要支持来自不同进程的提交序列，必须启用 DB2 Connect 集中器。要启用 DB2 Connect EE 集中器，将数据库管理器配置参数 *max\_logicagents* 设置为大于 *maxagents* 的值。注意，DB2 Connect EE 集中器需要 DB2 版本 7.1 客户机才能支持来自不同进程的 XA 提交序列。有关此环境中允许的 SQL 语句的信息，参考 *Application Development Guide*。有关集中器的信息，参考《DB2 Connect 用户指南》。

若您要更新主机或 AS/400 数据库服务器，将需要配置了 DB2 同步点管理器 (SPM) 的 DB2 Connect。参考其中一本《快速入门》书籍以获取指导。

## 数据库连接注意事项

本节包括下列主题：

- 『RELEASE 语句』
- 第143页的『访问分区数据库的事务』

### RELEASE 语句

若使用 RELEASE 语句来释放与一个数据库的连接，则应使用 CONNECT 语句而不是 SET CONNECTION 来重新连接该数据库。

## 访问分区数据库的事务

在分区数据库环境中，可将用户数据分布在各数据库分区中。访问该数据库的应用程序与其中一个数据库分区（协调程序节点）连接，并向其发送请求。不同的应用程序可以与不同的数据库分区连接，而同一个应用程序可以为不同的连接选择不同的数据库分区。

对于对分区数据库环境中的某个数据库执行的事务，所有的访问都必须通过同一个数据库分区。即，从启动该事务起，直到提交该事务为止（包括提交时的那一刻），都必须使用同一个数据库分区。

对分区数据库执行的任何事务都必须在断开之前提交。

## 作试探性决定

XA 兼容的事务管理器（事务处理监视器）使用一个两阶段提交过程，该过程类似于 DB2 事务管理器所使用的那个过程，如第126页的『了解两阶段提交过程』所述。两个环境之间的主要区别是，TP 监视器提供记录和控制该事务的功能，而不是 DB2 事务管理器和事务管理器数据库来提供此功能。

当使用 XA 兼容的事务管理器时，可能发生类似于对 DB2 事务管理器讨论的那些错误（参见第129页的『在两阶段提交期间校正问题』）。类似于 DB2 事务管理器，XA 兼容的事务管理器将试图使不确定事务再同步。

若由于某种原因您不能等待事务管理器自动解决不确定事务，则您可以执行一些操作，以手工方式解决它们。此手工过程有时称为“作试探性决定”。

LIST INDOUBT TRANSACTIONS 命令（使用 WITH PROMPTING 选项）或相关的一组 API 允许您查询、提交和回滚不确定事务。另外，它还允许您除去运行记录并释放日志空间，以“忘记”通过试探方式提交或回滚的事务。要从在基于 UNIX 的系统、Windows 操作系统或 OS/2 上的 DB2 UDB 中获取不确定事务信息，与数据库连接，并发出 LIST INDOUBT TRANSACTIONS WITH PROMPTING 命令或等价的 API。有关此命令或相关的管理 API 的信息，参考 *Command Reference* 或 *Administrative API Reference*。

对于与主机或 AS/400 数据库服务器相关的不确定事务信息，您有两个选择：

- 可以直接从主机或 AS/400 服务器获取不确定信息。  
要直接从 DB2 OS/390 版获取不确定信息，调用 DISPLAY THREAD TYPE(INDOUBT) 命令。使用 RECOVER 命令来作试探性决定。要直接从 DB2 OS/400 版获取不确定信息，调用 **wrkcmdfn** 命令。
- 可以从用来访问主机或 AS/400 数据库服务器的 DB2 Connect 服务器获取不确定信息。

要从 DB2 Connect 服务器获取不确定信息，首先通过连接由 *spm\_name* 数据库管理器配置参数的值所表示的 DB2 实例来与 DB2 同步点管理器连接。然后发出 LIST DRDA INDOUBT TRANSACTIONS WITH PROMPTING 命令来显示不确定事务并作试探性决定。

务请极为谨慎地使用这些命令（或相关的 API），并且只将其作为最后的手段。最好的策略是等待事务管理器驱动再同步过程。若您在其中一个参与数据库中以手工方式提交或回滚一个事务，并对另一个参与的数据库执行相反的操作，可能会遇到数据完整性问题。校正数据完整性问题要求您了解该应用程序的逻辑、标识已更改或回滚的数据，然后执行数据库的时间点恢复，或以手工方式撤消或重新执行更改。

若您不能等待事务管理器启动再同步过程，且您必须释放不确定事务占用的资源，则必须执行试探性操作。若事务管理器无法在延长期内执行再同步，且该不确定事务占用着急需的资源，则可能发生这种情况。不确定事务占用着事务管理器或资源管理器成为不可用之前与此事务相关的资源。对于数据库管理器，这些资源包括对该事务占用的表和索引、日志空间和存储器的锁定。每个不确定事务还会递减（减一）数据库可以处理的并行事务的最大数目。

尽管没有简单明了的方法来执行试探性操作，但下面还是提供了一些一般准则：

1. 与您要完成其所有事务的数据库连接。
2. 使用 LIST INDOUBT TRANSACTION 命令显示不确定事务。*xid* 表示全局事务标识，它与事务管理器和参与事务的其他资源管理器使用的 *xid* 完全相同。
3. 对于每个不确定事务，使用您掌握的有关该应用程序和操作环境的知识，来确定其他参与的资源管理器。
4. 确定事务管理器是否可用：
  - 若该事务管理器是可用的，且资源管理器中的不确定事务是由于在第二个提交阶段期间或更早的再同步过程期间资源管理器不可用导致的，则您应该检查该事务管理器的日志，以确定对其他资源管理器执行过什么操作。然后您应该对该数据库执行相同的操作；即使用 LIST INDOUBT TRANSACTIONS 命令，试探性提交或试探性回滚该事务。
  - 若事务管理器不可用，您将需要使用其他参与资源管理器中该事务的状态，以确定您应该执行什么操作：
    - 若其他资源管理器中至少有一个已经提交该事务，则您应该在所有资源管理器中试探性提交该事务。
    - 若其他资源管理器中至少有一个已经回滚该事务，则您应该试探性回滚该事务。

- 若该事务在所有参与资源管理器中都处于“已准备好”（未确定）状态，则您应该试探性回滚该事务。
- 若其他资源管理器中有一个或多个不可用，则您应该试探性回滚该事务。

除非试探性提交或回滚的事务导致日志满情况（在 `LIST INDOUBT TRANSACTIONS` 命令的输出中指示），否则不要执行试探性忘记功能。试探性忘记功能释放由不确定事务占用的日志空间。隐含情况为，若一个事务管理器最终对此不确定事务执行了再同步操作，则可能会作出错误的决定来提交或回滚其他资源管理器，因为在此资源管理器中没有该事务的运行记录。通常，“丢失”运行记录意味着该资源管理器已经回滚该事务。

## 安全性注意事项

TP 监视器预分配一组服务器进程，并由在这些服务器进程标识下的不同用户运行事务。对于该数据库，每个服务器进程表现为一个有许多工作单元的大应用程序，所有工作单元都在与该服务器进程相关的相同标识下运行。

例如，在使用 CICS 的 AIX 环境中，当启动 TXSeries CICS 区域时，该区域与用于定义它的 AIX 用户名相关。所有“CICS 应用程序服务器”进程也都在 TXSeries CICS 主标识下运行，该标识通常被定义为“cics”。CICS 用户可以在他们的 DCE 登录标识下调用 CICS 事务，且在 CICS 中时，他们还可以使用 CESN 注册事务来更改他们的标识。在任何一种情况中，RM 都无法使用最终用户的标识。因此，一个“CICS 应用程序进程”可能正在代表许多用户运行事务，但是，对于 RM 来说，它就象是一个带有许多工作单元的单一程序，而这些工作单元具有相同的“cics” ID。您可以选择在 `xa_open` 字符串中指定用户标识和密码，并将使用该用户标识来代替“cics” ID 以与数据库连接。

对于静态 SQL 语句，没有很大影响，因为是使用绑定程序的特权而非最终用户的特权来访问数据库。但是，这意味着必须将数据库软件包的 EXECUTE 特权授予服务器标识，而不是最终用户标识。

对于动态语句，它们在运行时执行访问认证，必须将数据库对象的访问特权授予服务器标识，而非那些对象的实际用户。此时必须依靠 TP 监视器系统来确定允许用户运行的程序，而不是依靠数据库来控制特定用户的访问。必须授予服务器标识它的 SQL 用户需要的所有特权。

要确定谁访问了数据库表或视图，您可以执行下列步骤：

1. 从 `SYSCAT.PACKAGEDEP` 目录视图，获取取决于该表或视图的所有软件包的列表。

2. 通过安装期间使用的命名约定，确定与这些软件包对应的服务器程序（例如，CICS 程序）的名称。
3. 确定可能调用这些程序的客户机程序（例如，CICS 事务标识），然后使用 TP 监视器的日志（例如，CICS 日志）来确定谁运行了这些事务或程序以及是何时运行的。

## 配置注意事项

当您设置 TP 监视器环境时，应该考虑下列配置参数：

- *tp\_mon\_name*

此数据库管理器配置参数标识正在使用的 TP 监视器产品的名称（例如，“CICS”或“ENCINA”）。

- *tpname*

此数据库管理器配置参数标识远程事务程序的名称，当数据库客户机使用 APPC 通信协议向数据库服务器发出分配请求时，必须使用该事务程序。此值是在服务器的配置文件中设置的，且必须与在 SNA 事务程序中配置的事务处理器 (TP) 的名称相同。有关更多信息，参考《快速入门》手册。

- *tm\_database*

因为 DB2 不协调 XA 环境中的事务，所以不对 XA 协调的事务使用此数据库管理器配置参数。

- *maxappls*

此数据库配置参数指定允许的活动应用程序的最大数目。此参数的值必须等于或大于连接的应用程序数，加上可能正在同时完成一个两阶段提交或回滚的这些应用程序的数目之和。此和随后应加上在任何时刻可能存在的预期不确定事务数。有关不确定事务的更多信息，参见第129页的『在两阶段提交期间校正问题』。

对于 TP 监视器环境（例如，TXSeries CICS），可能需要增加 *maxappls* 参数的值。这有助于确保可以处理所有 TP 监视器进程。



- *autorestart*

此数据库配置参数指定在需要时是否自动调用 RESTART DATABASE 例程。缺省值是 YES (即, 启用)。

需要一个重新启动数据库操作来启动包含不确定事务的数据库。若断开与该数据库的最后一个连接时未启用 *autorestart*, 则下一个连接将失败并需要显式的 RESTART DATABASE 调用。这种情况将一直存在, 直到事务管理器的再同步操作或管理员启动的试探性操作除去了不确定事务为止。当发出 RESTART DATABASE 命令时, 若该数据库中有任何不确定事务, 将返回一条信息。管理员就可以使用 LIST INDOUBT TRANSACTIONS 命令和其他的命令行处理器命令来获取有关那些不确定事务的信息。

## 受支持的 XA 函数

“DB2 通用数据库”支持在 *X/Open CAE Specification Distributed Transaction Processing: The XA Specification* 中定义的 XA91 规范, 但是下列情况除外:

- 异步服务

XA 规范允许该接口使用异步服务, 这样可以在稍后的某个时间检查请求的结果。数据库管理器要求以同步方式调用请求。

- 静态注册

XA 接口允许以两种方式来注册 RM: 静态注册和动态注册。“DB2 通用数据库”只支持动态注册, 此方式更高级且更有效率。有关这两种方法的更多信息, 参见第136页的『资源管理器 (RM)』。

- 关联迁移

DB2 通用数据库不支持控制的线程之间的事务迁移。

有关 `xa_open` 和 `xa_close` 字符串用法的信息, 参见第137页的『`xa_open` 和 `xa_close` 字符串的用法』。

## XA 开关的用法和位置

根据 XA 接口的要求, 数据库管理器提供 `xa_switch_t` 类型的 `db2xa_switch` 外部 C 变量, 以将 XA 开关结构返回至该 TM。不是返回各种 XA 函数的地址, 而是返回下列字段:

字段	值
----	---

名称	数据库管理器的产品名。例如, DB2 AIX 版。
----	---------------------------

标志	TMREGISTER   TMNOMIGRATE
----	--------------------------

明确声明“DB2 通用数据库”使用动态注册, 且 TM 不应该使用关联迁移。明确声明不支持异步操作。

版本 必须为零。

### 使用“DB2 通用数据库”的 XA 开关

XA 体系结构要求“资源管理器”(RM)提供开关,该开关使“XA 事务管理器”(TM)可以访问 RM 的 **xa** 例程。RM 开关使用称为 `xa_switch_t` 的结构。该开关包含 RM 的名称、指向 RM 的 XA 入口点的非 NULL 指针、标志和版本号。

**基于 UNIX 的系统和 OS/2:** DB2 UDB 的开关可以通过下面两种方法之一获得:

- 通过更高一级的间接方法。在 C 程序中,可在使用 `db2xa_switch` 之前定义以下的宏来完成:

```
#define db2xa_switch (*db2xa_switch)
```

- 通过调用 **db2xacic**

DB2 UDB 提供了此 API,它返回 `db2xa_switch` 结构的地址。此函数的原型为:

```
struct xa_switch_t * SQL_API_FN db2xacic( )
```

无论使用哪一种方法,都必须将应用程序与 `libdb2` (在基于 UNIX 的系统上)或 `db2api.lib` (在 OS/2 上)链接。

**Windows NT:** 指向 `xa_switch` 结构的指针 `db2xa_switch` 被作为 DLL 数据导出。这意味着使用此结构的 Windows NT 应用程序必须以下列三种方式之一来引用它:

- 通过更高一级的间接方法。在 C 程序中,可在使用 `db2xa_switch` 之前定义以下的宏来完成:

```
#define db2xa_switch (*db2xa_switch)
```

- 若使用 Microsoft Visual C++ 编译器,则可以将 `db2xa_switch` 定义为:

```
extern __declspec(dllimport) struct xa_switch_t db2xa_switch
```

- 通过调用 **db2xacic**

DB2 UDB 提供了此 API,它返回 `db2xa_switch` 结构的地址。此函数的原型为:

```
struct xa_switch_t * SQL_API_FN db2xacic( )
```

使用这些方法中任何一种,必须将应用程序与 `db2api.lib` 链接。

**C 代码示例:** 下列代码举例说明在任何 DB2 UDB 平台上通过 C 程序访问 `db2xa_switch` 的不同方法。务必将应用程序与适当的库连接。

```
#include <stdio.h>
#include <xa.h>

struct xa_switch_t * SQL_API_FN db2xacic( );
```

```

#ifdef DECLSPEC_DEFN
extern __declspec(dllimport) struct xa_switch_t db2xa_switch;
#else
#define db2xa_switch (*db2xa_switch)
extern struct xa_switch_t db2xa_switch;
#endif

main( )
{
    struct xa_switch_t *foo;
    printf ( "%s \n", db2xa_switch.name );
    foo = db2xacic();
    printf ( "%s \n", foo->name );
    return ;
}

```

## XA 接口的问题确定

若从 TM 发出 XA 请求期间检测到错误，则应用程序可能无法从 TM 获取错误码。若您的程序异常终止，或从 TP 监视器或 TM 获得加密的返回码，则应该检查“首次故障服务日志”，当诊断级别 3 或更高级别生效时，它将报告 XA 错误信息。有关“首次故障服务日志”的更多信息，参考 *Troubleshooting Guide*。

您还应该参考控制台信息、TM 错误文件或关于所用的外部事务处理软件的其他产品特定信息。

数据库管理器将所有 XA 特定错误写入“首次故障服务日志”，并使用 SQLCODE -998（事务或试探性错误）和适当的原因码。以下是一些更为常见的错误：

- xa\_open 字符串中的语法无效。
- 由于下列原因之一，未能与打开字符串中指定的数据库连接：
  - 数据库未编目。
  - 数据库未启动。
  - 服务器应用程序的用户名或密码无权与该数据库连接。
- 通信错误。

下面是（由于丢失 xa\_open 字符串）在 AIX 上生成的 xa\_open 错误的错误日志的一个示例：

```

Tue Apr 4 15:59:08 1995
toop pid(83378) process (xatest) XA DTP Support      sqlxa_open Probe:101
DIA4701E Database "" could not be opened for distributed transaction
processing.
String Title : XA Interface SQLCA pid(83378)
SQLCODE = -998 REASON CODE: 4 SUBCODE: 1
Dump File : /u/toop/diagnostics/83378.dmp Data : SQLCA

```

---

## 配置 XA 事务管理器以使用 DB2 UDB

下列几节描述如何配置特定的产品，以将 DB2 用作资源管理器。您可以使用下列任何一种方式：

- 『配置 IBM TXSeries CICS』
- 『配置 IBM TXSeries Encina』
- 第152页的『配置 BEA Tuxedo』
- 第154页的『配置 Microsoft 事务处理服务器』。

### 配置 IBM TXSeries CICS

有关如何配置 IBM TXSeries CICS 以将 DB2 用作资源管理器的信息，参考 *IBM TXSeries CICS Administration Guide*。TXSeries 文档可以通过访问如下站点联机查看：

[http://www.transarc.com/Library/documentation/websphere/WAS-EE/en\\_US/html/](http://www.transarc.com/Library/documentation/websphere/WAS-EE/en_US/html/)。

主机和 AS/400 数据库服务器可以参与 CICS 协调的事务。

### 配置 IBM TXSeries Encina

以下是当通过 DB2 Connect 访问时，将“Encina 监视器”与“DB2 通用数据库”服务器或 DB2 MVS 版、DB2 OS/390 版、DB2 AS/400 版或 DB2 VSE 版和 VM 版集成，所需的各种 API 和配置参数。TXSeries 文档可以通过访问如下站点联机查看：

[http://www.transarc.com/Library/documentation/websphere/WAS-EE/en\\_US/html/](http://www.transarc.com/Library/documentation/websphere/WAS-EE/en_US/html/)。

主机和 AS/400 数据库服务器可以参与 Encina 协调的事务。

### 配置 DB2

要配置 DB2:

1. 必须在 DB2 数据库目录中定义每个数据库的名称。若该数据库是远程数据库，则还必须定义节点目录项。您可以使用“客户机配置辅助程序”(CCA)或 DB2 命令行处理器 (CLP)来执行该配置。例如：

```
DB2 CATALOG DATABASE inventdb AS inventdb AT NODE host1 AUTH SERVER
DB2 CATALOG TCPIP NODE host1 REMOTE hostname1 SERVER svcname1
```

2. 若 DB2 客户机知道它处理的是 Encina，则它可以针对 Encina 优化它的内部处理。您可以将 *tp\_mon\_name* 数据库管理器配置参数设置为 ENCINA 来指定此选项。缺省行为是不进行特殊的优化。若设置了 *tp\_mon\_name*，则该应用程序必须确保执行该工作单元的线程在结束之后也立即提交该工作。不能启动任何

其他工作单元。若您的环境不是这样的，则确保 `tp_mon_name` 值是 `NONE`（或者，通过 `CLP` 将此值设置为 `NULL`）。可以通过“控制中心”或 `CLP` 更新此参数。`CLP` 命令是：

```
db2 update dbm cfg using tp_mon_name ENCINA
```

### 为每个资源管理器配置 Encina

要为每个资源管理器 (RM) 配置 Encina，管理员必须为作为资源管理器的每个 DB2 数据库定义“打开字符串”、“关闭字符串”和“控制线程协议”，然后才能为应用程序中的事务注册该资源管理器。可以使用 `Enconcole` 全屏幕界面或 `Encina` 命令行界面来执行该配置。例如：

```
monadmin create rm inventdb -open "db=inventdb,uid=user1,pwd=password1"
```

每个 DB2 数据库都有一个资源管理器配置，且每个资源管理器配置必须具有 `rm` 名称（“逻辑 RM 名称”）。为了简化情况，您应该使它与数据库名相同。

`xa_open` 字符串包含建立与数据库的连接所需的信息。该字符串的内容是 `RM` 所独有的。DB2 UDB 的 `xa_open` 字符串包含要打开的数据库的别名，以及与该连接相关的可选的用户标识和密码。注意，此处定义的数据库名也必须在所有数据库访问都必需的规则数据库目录中编目。有关 DB2 的 `xa_open` 字符串的信息，参见第 137 页的『将数据库设置为资源管理器』。

DB2 不使用 `xa_close` 字符串。

“控制线程协议”确定一个应用程序的代理程序线程是否一次可以处理多个事务。DB2 UDB 支持 `TMXA_SERIALIZE_ALL_OPERATIONS` 的缺省值，并且只有当一个事务完成之后才能再次使用一个线程。

若您正访问 DB2 OS/390 版、DB2 MVS 版、DB2 AS/400 版或 DB2 VSE 版和 VM 版，则必须使用 DB2 同步点管理器。有关配置指示信息，参考 *DB2 Connect Enterprise Edition for OS/2 and Windows Quick Beginnings* 手册。

### 从 Encina 应用程序中引用 DB2 数据库

要从 Encina 应用程序中引用 DB2 数据库：

1. 使用“Encina 调度策略 API”，来指定可以从单个 TP 监视器的应用程序进程中运行多少个应用程序的代理程序。例如：

```
rc = mon_SetSchedulingPolicy (MON_EXCLUSIVE)
```

对于 DB2（DB2 通用数据库、主机或 AS/400 数据库服务器），您应该使用 `MON_EXCLUSIVE` 的缺省设置。这确保：

- 应用程序进程在事务的寿命期间被锁定。
- 应用程序用作单线程。

注：若您使用的是 ODBC 或 DB2 调用层接口，则必须禁用多线程支持。为此，您可以设置 CLI 配置参数 `DISABLEMULTITHREAD = 1`（禁用多线程）。DB2 通用数据库的缺省值为 `DISABLEMULTITHREAD = 0`（启用多线程）。有关更多信息，参考 *CLI Guide and Reference*。

2. 使用“Encina RM 注册 API”来提供 XA 开关和逻辑 RM 名称，以供 Encina 在应用程序进程中引用 RM 时使用。例如：

```
rc = mon_RegisterRmi ( &db2xa_switch, /* xa switch */
                    "inventdb", /* logical RM name */
                    &rmiId ); /* internal RM ID */
```

XA 开关包含 RM 中 TM 可以调用的 XA 例程的地址，它还指定由 RM 提供的功能。DB2 通用数据库的 XA 开关是 `db2xa_switch`，它驻留在 DB2 客户机库（在 Windows 操作系统和 OS/2 上，是 `db2app.dll`，在基于 UNIX 的系统上，是 `libdb2`）中。

逻辑 RM 名是 Encina 使用的那个名称，而不是在 Encina 下运行的 SQL 应用程序使用的实际数据库名。实际数据库名是在“Encina RM 注册表”API 的 `xa_open` 字符串中指定的。逻辑 RM 名设置为与本示例中的数据库名相同。

第三个参数返回 TM 引用此连接所用的内部标识符或句柄。

- 注：当通过 TM-XA 接口使用 Encina 来进行 DB2 的事务处理时，注意 Encina 嵌套的事务当前不受 DB2 XA 接口的支持。如果有可能的话，请避免使用这些事务。若无法避免，则确保仅在 Encina 事务系列的一个成员中执行 SQL 工作。

## 配置 BEA Tuxedo

要配置 Tuxedo 以将 DB2 用作资源管理器，须执行下列步骤：

1. 按该产品的文档中指定的步骤，安装 Tuxedo。确保执行所有基本的 Tuxedo 配置，包括日志文件和环境变量。

您还需要一个编译程序和 DB2 应用程序开发客户机。需要时安装它们。

2. 在 Tuxedo 服务器标识中，设置 `DB2INSTANCE` 环境变量，以引用包含您希望 Tuxedo 使用的数据库的实例。将 `PATH` 变量设置为包括 DB2 程序目录。确认 Tuxedo 服务器标识可以与 DB2 数据库连接。
3. 使用值 `TUXEDO` 来更新 `tp_mon_name` 数据库管理器配置参数。
4. 将 DB2 的定义添加至 Tuxedo 资源管理器的定义文件。在下面的示例中，`UDB_XA` 是为 DB2 在本地定义的 Tuxedo 资源管理器名，而 `db2xa_switch` 是类型为 `xa_switch_t` 的一个结构的 DB2 定义的名称：

- 对于 AIX。在文件 `${TUXDIR}/udataobj/RM` 中，添加定义：

```
# DB2 UDB
UDB_XA:db2xa_switch:-L${DB2DIR} /lib -ldb2
```

其中，{TUXDIR} 是安装了 Tuxedo 的目录，而 {DB2DIR} 是 DB2 实例目录。

- 对于 Windows NT。在文件 %TUXDIR%\udataobj\rm 中，添加定义：

```
# DB2 UDB
UDB_XA;db2xa_switch;%DB2DIR%\lib\db2api.lib
```

其中，%TUXDIR% 是安装了 Tuxedo 的目录，而 %DB2DIR% 是 DB2 实例目录。

#### 5. 为 DB2 构建 Tuxedo 事务监视器服务器程序：

- 对于 AIX:

```
${TUXDIR}/bin/buildtms -r UDB_XA -o ${TUXDIR}/bin/TMS_UBD
```

其中，{TUXDIR} 是安装了 Tuxedo 的目录。

- 对于 Windows NT:

```
%TUXDIR%\bin\buildtms -r UDB_XA -o %TUXDIR%\bin\TMS_UBD
```

#### 6. 构建应用程序服务器。在下面的示例中，-r 选项指定资源管理器名，-f 选项（使用了一次或多次）指定包含应用程序服务的文件，-s 选项指定此服务器的应用程序服务名，而 -o 选项指定输出服务器文件名：

- 对于 AIX:

```
${TUXDIR}/bin/buildserver -r UDB_XA -f svcfile.o -s SVC1,SVC2
-o UDBserver
```

其中，{TUXDIR} 是安装了 Tuxedo 的目录。

- 对于 Windows NT:

```
%TUXDIR%\bin\buildserver -r UDB_XA -f svcfile.o -s SVC1,SVC2
-o UDBserver
```

其中，%TUXDIR% 是安装了 Tuxedo 的目录。

#### 7. 设置 Tuxedo 配置文件以引用 DB2 服务器。在 UDBCONFIG 文件的 \*GROUPS 小节中，添加类似如下的项：

```
UDB_GRP LMID=simp GRPNO=3
TMSNAME=TMS_UBD TMSCOUNT=2
OPENINFO="UDB_XA:db=sample,uid=db2_user,pwd=db2_user_pwd"
```

其中，TMSNAME 参数指定您先前构建的事务监视器的服务器程序，而 OPENINFO 参数指定资源管理器名。其后是数据库名和 DB2 用户和密码（用于认证）。

您先前构建的应用程序服务器在 Tuxedo 配置文件的 \*SERVERS 节中被引用。

8. 若应用程序正在访问驻留在 DB2 OS/390 版、DB2 OS/400 版或 DB2 VM 版和 DB2 VSE 版上的数据，则将需要 DB2 Connect XA 集中器。有关配置详细信息和限制，参考《DB2 Connect 用户指南》。

9. 启动 Tuxedo:

```
tmboot -y
```

在该命令完成之后，Tuxedo 信息应指示服务器已启动。另外，若您发出 DB2 命令 LIST APPLICATIONS ALL，您应该看到两个连接（在这种情况下），它们由 Tuxedo 配置文件 UDBCONFIG 中的 UDB 组的 TMSCOUNT 参数指定。

## 配置 Microsoft 事务处理服务器

DB2 UDB V5.2 和更新版本可以完全与 Microsoft 事务处理服务器 (MTS) 版本 2.0 集成在一起。在 Windows 32 位操作系统上的 MTS 下运行的应用程序，可以使用 MTS 来协调多个 DB2 UDB、主机和 AS/400 数据库服务器以及其他 MTS 兼容的资源管理器的两阶段提交。

**注：**IBM Web 站点上可能提供了其他技术信息，以帮助您安装和配置 DB2 MTS 支持。将 URL 设置为 <http://www.ibm.com/software/data/db2/library/>，并使用关键字 "MTS" 来搜索 DB2 通用数据库“技术注释”。

### 在 DB2 中启用 MTS 支持

自动启用“Microsoft 事务处理服务器”支持。可将 *tp\_mon\_name* 数据库管理器配置参数设置为 MTS，但这没有必要，而且将忽略它。

### MTS 软件的先决条件

MTS 支持需要 DB2 客户机应用程序使能器 (CAE) 版本 5.2 或更新版本，且 MTS 必须是带有最新修订 0772 的版本 2.0 或更高版本。

在 Windows 32 位操作系统上安装 DB2 ODBC 驱动程序将自动把一个新关键字添加至该注册表:

```
HKEY_LOCAL_MACHINE\software\ODBC\odbcinit.ini\IBM DB2 ODBC Driver:  
Keyword Value Name: CTimeout  
Data Type: REG_SZ  
Value: 60
```

### 安装和配置

以下是 MTS 的安装和配置注意事项的摘要。要使用 DB2 的 MTS 支持，必须:

1. 在运行 MTS 应用程序的机器上安装 MTS 和 DB2 客户机。
2. 若主机或 AS/400 数据库服务器要参与多站点更新:
  - a. 在本地机器或远程机器上安装“DB2 Connect 企业版”(EE)。DB2 Connect EE 允许主机或 AS/400 数据库服务器参与多站点更新事务。



- b. 确保允许 DB2 Connect EE 服务器参与多站点更新。有关允许 DB2 Connect 参与多站点更新的信息，参考该平台的“DB2 Connect 企业版快速入门”手册。

当运行 DB2 CLI/ODBC 应用程序时，下列配置关键字（与 db2cli.ini 文件中设置的相同）必须为缺省值，不得更改：

- CONNECTYPE 关键字（缺省值为 1）
- MULTICONNECT 关键字（缺省值为 1）
- DISABLEMULTITHREAD 关键字（缺省值为 0）
- CONNECTIONPOOLING 关键字（缺省值为 0）
- KEEPCONNECTION 关键字（缺省值为 0）

为利用 MTS 支持而编写的 DB2 CLI 应用程序不能更改与上述关键字对应的属性值。而且，应用程序不能更改下列属性的缺省值：

- SQL\_ATTR\_CONNECT\_TYPE 属性（缺省值 SQL\_CONCURRENT\_TRANS）
- SQL\_ATTR\_CONNECTON\_POOLING 属性（缺省值 SQL\_CP\_OFF）

### 验证安装

1. 配置 DB2 客户机和 DB2 Connect EE 以访问您的 DB2 UDB、主机或 AS/400 服务器。
2. 验证从 DB2 CAE 机器至 DB2 UDB 数据库服务器的连接。
3. 使用 DB2 CLP 来验证从 DB2 Connect 机器至您的主机或 AS/400 数据库服务器的连接，并发出一些查询。
4. 验证从 DB2 CAE 机器经由 DB2 Connect 网关至您的主机或 AS/400 数据库服务器的连接，并发出一些查询。

### 受支持的 DB2 数据库服务器

支持下列服务器使用 MTS 协调的事务来进行多站点更新：

- DB2 通用数据库企业版的版本 6 和更高版本
- DB2 通用数据库扩充企业版的版本 6 和更高版本
- DB2 OS/390 版
- DB2 MVS 版
- DB2 AS/400 版
- DB2 VM 版和 VSE 版

### MTS 事务超时和 DB2 连接特性

您可以在 MTS Explorer 工具中设置事务超时值。有关更多信息，请参考联机 *MTS 管理员指南*。

若一个事务占用的时间超过该事务超时值（缺省值为 60 秒），则 MTS 将以异步方式对所有参与的“资源管理器”发出异常终止，然后将整个事务异常终止。

对于与 DB2 服务器的连接，该异常终止会被转换成 DB2 回滚请求。象任何其他数据库请求一样，将在该连接上连续执行回滚请求，以保证数据库服务器上的数据的完整性。

其结果为：

- 若该连接是空闲的，则立即执行回滚。
- 如果正在处理长时间运行的 SQL 语句，则回滚请求等待，直到该 SQL 语句完成为止。

### 连接缓冲

连接缓冲允许一个应用程序使用一个连接缓冲池中的连接，而不必在每次使用时重新建立连接。一旦创建了一个连接并将其置于缓冲池中，则应用程序可以再次使用该连接，而不用执行完整的连接过程。当应用程序与 ODBC 数据源断开时将缓冲该连接，并将它给予一个属性相同的新连接。

连接缓冲一直是“ODBC 驱动程序管理器 2.x”的一个功能部件。在与 MTS 一起交付的最新 ODBC 驱动程序管理器（版本 3.5）中，对于事务性 MTS COM 对象的 ODBC 连接，连接缓冲有一些配置变化和新的行为（参见第157页的『在参与同一个事务的 COM 对象之间再次使用 ODBC 连接』）。

“ODBC 驱动程序管理器 3.5”要求 ODBC 驱动程序在注册表中注册一个新关键字，才允许将连接缓冲激活。该关键字为：

```
Key Name: SOFTWARE\ODBC\ODBCINST.INI\IBM DB2 ODBC DRIVER
Name: CTimeout
Type: REG_SZ
Data: 60
```

32 位操作系统的 DB2 ODBC 驱动程序版本 6 和更高版本完全支持连接缓冲；因此注册了此关键字。版本 5.2 客户机必须安装修订包 3 (WR09024) 或更高版本。

缺省值 60 意味着在连接被断开之前将缓冲 60 秒。

在一个繁忙的环境中，最好是将 CTimeout 的值该为一个较大的数（Microsoft 有时建议对于某些环境使用 10 秒），以阻止过多的物理连接和断开，因为这些操作会占用大量系统资源，包括系统内存和通信堆栈资源。

另外，要确保在多处理器机器中的同一事务中的对象之间使用同一连接，必须关闭“每个处理器有多个缓冲池”支持。为此，将下列注册表设置复制到名为

odbcpool.reg 的文件中，将其作为无格式文本文件保存，并发出命令 **odbcpool.reg**。Windows 操作系统将导入此注册表设置。

```
REGEDIT4
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBCINST.INI\ODBC Connection Pooling]  
"NumberOfPools"="1"
```

如果不将此关键字设置为 1，则 MTS 可能在不同的缓冲中缓冲连接，导致不重新使用同一连接。

### 使用 ADO 2.1 和更新版本的 MTS 连接缓冲

如果 MTS COM 对象使用 ADO 来访问数据库，则必须关闭 OLEDB 资源缓冲，以便 ODBC 的 Microsoft OLEDB 提供器 (MSDASQL) 不会干扰 ODBC 连接缓冲。在 ADO 2.0 中，此功能被初始为 OFF，但在 ADO 2.1 中，被初始化为 ON。要关闭 OLEDB 资源缓冲，将下列各行复制到名为 oledb.reg 的文件中，并将其作为无格式文本文件保存，然后发出命令 **oledb.reg**。Windows 操作系统将导入这些注册表设置。

```
REGEDIT4
```

```
[HKEY_CLASSES_ROOT\CLSID\{c8b522cb-5cf3-11ce-ade5-00aa0044773d}]  
@="MSDASQL"  
"OLEDB_SERVICES"=dword:ffffffff
```

### 在参与同一个事务的 COM 对象之间再次使用 ODBC 连接

MTS COM 对象中的 ODBC 连接会自动打开连接缓冲（不管该 COM 对象是否是事务性的）。

对于参与同一个事务的多个 MTS COM 对象，可以使用如下方式在两个或多个 COM 对象之间再次使用该连接。

假定有两个 COM 对象 COM1 和 COM2，它们与同一个 ODBC 数据源连接并参与同一个事务。

在 COM1 连接并完成它的工作之后，它就会断开，并缓冲该连接。但是，此连接将被保留，以备同一个事务的其他 COM 对象使用。只有在当前事务结束之后，它才可供其他事务使用。

当在同一个事务中调用 COM2 时，会给予它缓冲的连接。MTS 将确保只将该连接分配给参与同一个事务的 COM 对象。

另一方面，若 COM1 未显式断开连接，则在该事务结束之后它仍维持着该连接。当在同一个事务中调用 COM2 时，会请求另一个连接。这样，此事务同时维持着两个连接，而不是一个。

由于下列原因，最好对参与同一个事务的 COM 对象使用再次使用连接的功能：

- 它在客户机和服务器中都使用较少的资源。只需要一个连接。
- 它消除了参与同一个事务的两个连接（访问相同的数据库服务器和访问相同的数据）相互锁定的可能性，因为 DB2 服务器将来自 MTS COM 对象的不同连接当作单独的事务来处理。

### 调整 TCP/IP 通信

若在同时进行太多的物理连接和断开的具有高工作负荷环境中，使用很小的 CTimeout 值，则 TCP/IP 堆栈可能会遇到资源约束。

要消除此问题，使用“TCP/IP 注册表项”。有关信息在 *Windows NT Resource Guide* 的第一卷中有描述。注册表键值位于 HKEY\_LOCAL\_MACHINE—> SYSTEM—> CurrentControlSet—> Services—> TCPIP—> 参数中。

其缺省值和推荐的设置如下所示：

名称	缺省值	建议的值
KeepAlive 时间	7200000 (2 小时)	相同
KeepAlive 时间间隔	1000 (1 秒)	10000 (10 秒)
TcpKeepCnt	120 (2 分钟)	240 (4 分钟)
TcpKeepTries	20 (20 次重试)	相同
TcpMaxConnectAttempts	3	6
TcpMaxConnectRetransmission	3	6
TcpMaxDataRetransmission	5	8
TcpMaxRetransmissionAttempts	7	10
若未定义注册表的值，则创建它。		

### 使用 MTS “BANK” 样本应用程序测试 DB2

可以使用 MTS 所附带的“BANK”样本程序来测试客户机产品和 MTS 的设置。

执行下列步骤：

1. 更改文件 \Program Files\Common Files\ODBC\Data Sources\MTSSamples.dsn, 使它类似于如下所示：

```
[ODBC]
DRIVER=IBM DB2 ODBC DRIVER
UID=your_user_id
PWD=your_password
DSN=your_database_alias
Description=MTS Samples
```

其中:

- *your\_user\_id* 和 *your\_password* 是用于连接主机的用户标识和密码。
  - *your\_database\_alias* 是用于连接数据库服务器的数据库别名。
2. 转至“控制面板”中的“ODBC 管理”，选择 **系统 DSN** 标签并添加数据源:
    - a. 选择“IBM ODBC 驱动程序”，然后选择**完成**。
    - b. 当显示数据库别名的列表时，选择先前指定的那一个。
    - c. 选择**确定**。
  3. 以 ID *your\_user\_id* 的名义，使用 DB2 CLP 与 DB2 数据库连接，如上所述。

- a. 绑定 db2cli.lst 文件:

```
db2 bind @C:\sql1lib\bnd\db2cli.lst blocking all grant public
```

- b. 绑定这些实用程序。

若服务器是 **DRDA** 主机服务器，则根据您连接的主机 (OS/390、AS/400、VSE&VM)，绑定 ddcsmvs.lst、ddcs400.lst 或 ddcsvm.lst。例如:

```
db2 bind @C:\sql1lib\bnd\@ddcsmvs.lst blocking all grant public
```

否则，绑定 db2ubind.lst 文件:

```
db2 bind @C:\sql1lib\bnd\@db2ubind.lst blocking all grant public
```

- c. 为 MTS 样本应用程序创建样本表和数据，如下所示:

```
db2 create table account (accountno int, balance int)
db2 insert into account values(1, 1)
```

4. 在 DB2 客户机上，确保数据库管理器配置参数 *tp\_mon\_name* 设置为 MTS。
5. 运行“BANK”应用程序。选择**帐户**按钮和 **Visual C++** 选项，然后提交该请求。其他选项可使用专用于 SQL 服务器的 SQL，但可能不起作用。



---

## 第11章 介绍高可用性和故障恢复支持

成功的电子商务取决于事务处理系统连续的可用性，而这些系统由数据库管理系统（如 DB2）所驱动，它们必须是每天 24 小时且每周 7 天都可用（“24 x 7”）。

---

### 高可用性

高可用性 (HA) 是一个术语，用来描述在某种程度上始终保持运行且对用户可用的系统。要做到这一点：

- 在处理的高峰期，必须有效地处理事务，而没有明显的性能降低（或者甚至没有可用性损失）。在分区数据库环境中，DB2 可以利用分区内并行性和分区间并行性来有效地处理事务。在 SMP 环境中，可以使用分区内并行性来同时处理复杂 SQL 语句的各个部分。另一方面，在分区数据库环境中，分区间并行性指的是在全部参与的节点上同时处理查询；每个节点处理表中行的一个子集。有关并行性的更多信息，参见第33页的『并行性的类型』。
- 当发生硬件或软件故障，或者发生灾难时，系统必须能够很快恢复。DB2 具有高级连续检查点系统和并行恢复能力，它允许进行非常快的崩溃恢复。

快速恢复能力还可取决于在适当的位置具有已证实有效的备份与恢复策略。有关恢复策略的更多信息，参见 *Data Recovery and High Availability Guide and Reference*。

- 驱动企业数据库的软件必须持续运行并且可用于事务处理。要保持数据库管理器运行，必须确保如果它失败，另一个数据库管理器可以接管其工作。这称为故障恢复。故障恢复能力允许在发生硬件故障时，自动将工作负荷从一个系统转移至另一个系统。

可通过在另一台机器上保存数据库的副本来实现故障恢复保护，该机器始终向前滚动日志文件。日志装运是将全部日志文件复制到备用机器的过程（或者从归档设备复制，或者通过对主数据库运行用户出口程序进行复制）。通过此方法，使用 DB2 复原实用程序或分割镜像功能将主数据库复原至备用机器。可以使用新的暂挂 I/O 支持来快速初始化新的数据库（参见第163页的『基于联机分割镜像和暂挂 I/O 支持的高可用性』）。备用机器上的辅助数据库持续向前滚动日志文件。如果主数据库发生故障，则将任何保留的日志文件全部复制到备用机器。在向前滚动至日志的末尾并停止操作之后，将所有客户机重新连接至备用机器上的辅助数据库。

还可通过特定于平台的软件提供故障恢复支持，可以将该软件添加至您的系统。例如：

- 高可用性群集多处理增强型可伸缩性 AIX 版。  
有关 HACMP/ES 的详细信息，参见 *Data Recovery and High Availability Guide and Reference*、或标题为 “IBM DB2 Universal Database Enterprise Edition for AIX and HACMP/ES” 的白皮书，可从 “DB2 UDB 和 DB2 Connect 联机支持” Web 站点 (<http://www.ibm.com/software/data/pubs/papers/>) 获得该书。
- Microsoft 群集服务器 Windows NT 版或 Windows 2000 版。  
有关 MSCS 的详细信息，参见 *Data Recovery and High Availability Guide and Reference*。
- Sun 群集或 VERITAS 群集服务器 Solaris 操作环境版。  
有关 Sun Cluster 2.x 的信息，参见 *Data Recovery and High Availability Guide and Reference*；有关 Sun Cluster 3.0 的信息，参见标题为 “DB2 and High Availability on Sun Cluster 3.0” 的白皮书，可从 “DB2 UDB 和 DB2 Connect 联机支持” Web 站点 (<http://www.ibm.com/software/data/pubs/papers/>) 获得该书。  
有关 VERITAS 群集服务器的信息，参见标题为 “DB2 and High Availability on VERITAS Cluster Server” 的白皮书，可从 “DB2 UDB 和 DB2 Connect 联机支持” Web 站点获得该书。
- Multi-Computer/ServiceGuard Hewlett-Packard 版。  
有关 HP MC/ServiceGuard 的详细信息，参见标题为 “IBM DB2 EE v.7.1 Implementation and Certification With Hewlett-Packard’s MC/ServiceGuard High Availability Software” 的白皮书，可从 “DB2 UDB 和 DB2 Connect 联机支持” Web 站点 (<http://www.ibm.com/software/data/pubs/papers/>) 获得该书。

故障恢复策略通常基于系统的群集。群集是一组已连接的系统，它们作为一个单一系统一起工作。每个处理器称为群集内的一个节点。在发生故障时，群集允许服务器通过接管失效服务器的工作负荷来互相替换。

IP 地址接管（或 IP 接管）是当一台服务器出现故障时，将服务器的 IP 地址从一台机器传送到另一台机器的能力；对于客户机应用程序来说，在不同时间出现的两台机器实际是同一台服务器。

故障恢复软件可以在系统之间使用波动信号监视或保持活动信息包来确认可用性。波动信号监视涉及维护群集中所有节点之间的持续通信的系统服务。如果未检测到波动信号，则将启动至备份系统的故障恢复。最终用户通常不会意识到系统已失效。

市场上两种最普通的故障恢复策略称为空闲备用和相互接管，尽管与这些术语相关联的配置还可能与依赖于供应商的不同术语相关联：

#### 空闲备用

在这种配置中，使用一个系统来运行 DB2 实例，第二个系统则“空闲”或



处于备用方式，随时准备在第一个系统发生操作系统故障或硬件故障时接管该实例。因为备用系统在需要它之前是空闲的，所以系统的整体性能不会受影响。

### 相互接管

在此配置中，每个系统是另一系统的指定备份。系统的整体性能可能会受影响，因为备份系统在故障恢复之后必须执行额外的工作：它必须执行其本身的工作，再加上发生故障的系统原来执行的工作。

故障恢复策略可以用来对实例、分区或多个逻辑节点进行故障恢复。

---

## 基于联机分割镜像和暂挂 I/O 支持的高可用性

**暂挂 I/O** 通过提供联机分割镜像处理（即分割镜像，而不关闭数据库）的完全实现来支持持续的系统可用性。分割镜像是数据库的“瞬时”副本，可通过对包含数据的磁盘进行镜像并在需要副本时分割此镜像来生成该副本。磁盘镜像是将所有数据写至两个独立硬盘的过程；两个硬盘互为镜像。分割镜像是生成该镜像的副本。

如果您不愿使用 DB2 备份实用程序备份大的数据库，则可以通过使用暂挂 I/O 和分割镜像功能从镜像生成副本。此方法还可以：

- 消除生产机器的备份操作开销
- 提供一种快捷方式来克隆系统
- 提供空闲备用故障恢复的快速实现。不需要执行初始复原操作，并且如果前滚操作已证明太慢或遇到错误，则重新初始化的速度很快。

**db2inidb** 命令初始化分割镜像，以便可使用它来：

- 生成克隆数据库  
例如，主数据库的只读克隆可用来创建报表。
- 作为备用数据库
- 作为备份映像

仅可以对分割镜像发出此命令，并且分割镜像必须首先运行 **db2inidb**，然后才能使用分割镜像（参见 *Data Recovery and High Availability Guide and Reference*）。

在分区数据库环境中，必须在每个分区上运行 **db2inidb** 命令，然后才能使用来自任何一个分区的分割映像。可以同时所有分区上运行此工具。

## 生成克隆数据库

数据库克隆可以提供主（活动的）数据库的脱机“备份”。然而，您不能备份克隆数据库，不能在原始系统上复原此映象，并且不可以前滚在原始系统上生成的日志文件。

要克隆数据库，执行下列步骤：

1. 暂挂主数据库上的 I/O:

```
db2 set write suspend for database
```

2. 使用适当的操作系统级命令来分割主数据库的镜像。

3. 恢复主数据库上的 I/O:

```
db2 set write resume for database
```

4. 从另一机器连接至镜像数据库。

5. 启动数据库实例:

```
db2start
```

6. 将镜像数据库初始化为为主数据库的克隆:

```
db2inidb database_alias as snapshot
```

**注：**此命令将回滚执行分割时正在进行的事务。

## 将分割镜像用作备用数据库

当镜像（备用）数据库不断向前滚动日志时，会不断地从主系统获取主数据库正在创建的新日志。要将分割镜像用作备用数据库，执行下列步骤：

1. 暂挂主数据库上的 I/O:

```
db2 set write suspend for database
```

2. 使用适当的操作系统级命令来分割主数据库的镜像。

3. 恢复主数据库上的 I/O:

```
db2 set write resume for database
```

4. 将镜像数据库连接至另一个实例。

5. 将镜像数据库置于前滚暂挂状态:

```
db2inidb database_alias as standby
```

如果您只具有 DMS 表空间（数据库管理空间），则可以采取完全数据库备份来卸载在生产数据库上进行备份的开销。

6. 设置用户出口程序来从主系统中检索日志文件。

7. 向前滚动数据库，直到日志的末尾。

8. 继续检索日志文件，并且向前滚动数据库至日志的末尾，直到主数据库关闭。

## 将分割镜像用作备份映像

要将分割镜像用作“备份映像”，执行下列步骤：

1. 暂挂主数据库上的 I/O:

```
db2 set write suspend for database
```

2. 使用适当的操作系统级命令来分割主数据库的镜像。

3. 恢复主数据库上的 I/O:

```
db2 set write resume for database
```

4. 主系统上发生故障，迫使从备份系统执行复原。

5. 使用操作系统级命令来将分割数据复制至主系统。不要复制分割日志，因为前滚恢复将需要主日志。

6. 启动主数据库实例:

```
db2start
```

7. 初始化主数据库:

```
db2inidb database_alias as mirror
```

8. 前滚主数据库，直到日志的末尾。



---

## 第4部分 附录



---

## 附录A. 命名规则

转至描述您需要有关其信息的命名规则的章节:

- 『一般命名规则』
- 『对象命名规则』
- 第172页的『如何在联合系统中保留区分大小写的值』

---

### 一般命名规则

除非另有指定, 所有名称可以包括下列字符:

- A 至 Z。在大多数名称中使用, 会将字符 A 至 Z 从小写转换为大写。
- 0 至 9
- @、#、\$ 和 \_ (下划线字符)

名称不能以数字或下划线字符开始。

不要使用 SQL 保留字来命名表、视图、列、索引或授权标识。有关 SQL 保留字的列表, 参见 *SQL Reference*。

取决于操作系统以及使用 DB2 的环境, 还有其它特殊字符可能起作用。然而, 虽然它们可能起作用, 但不能保证它们将一定起作用。建议在命名数据库中的对象时不要使用其他特殊字符。

---

### 对象命名规则

所有对象遵循“一般命名规则”。另外, 某些对象还具有下面所显示的附加限制。

表 24. 数据库、数据库别名和实例命名规则

对象	准则
<ul style="list-style-type: none"> <li>• 数据库</li> <li>• 数据库别名</li> <li>• 实例</li> </ul>	<ul style="list-style-type: none"> <li>• 数据库名在它们被编目的位置中必须是唯一的。在基于 UNIX 的 DB2 实现上，此位置是一个目录路径，而在 Windows 实现上，它是盘符。</li> <li>• 数据库别名在系统数据库目录中必须是唯一的。当创建新数据库时，该别名缺省为数据库名。因此，不能使用已作为数据库别名的名称来创建数据库，即使没有具有该名称的数据库。</li> <li>• 数据库、数据库别名和实例名最多只能包含 8 个字节。</li> <li>• 在 Windows NT 和 Windows 2000 系统上，任何实例都不能与服务名同名。</li> </ul> <p><b>注：</b> 为避免潜在的问题，若您打算在通信环境中使用数据库，则不要在数据库名称中使用特殊字符 @、# 和 \$。而且，因为不是所有键盘都有这些字符，因此若您计划以另一语言使用该数据库，则不要使用这些字符。</p>

表 25. 数据库对象命名规则

对象	准则
<ul style="list-style-type: none"> <li>• 别名</li> <li>• 缓冲池</li> <li>• 列</li> <li>• 事件监视器</li> <li>• 索引</li> <li>• 方法</li> <li>• 节点组</li> <li>• 模式</li> <li>• 存储过程</li> <li>• 表</li> <li>• 表空间</li> <li>• 触发器</li> <li>• UDF</li> <li>• UDT</li> <li>• 视图</li> </ul>	<p>最多可以包含 18 个字节，但下列名称除外：</p> <ul style="list-style-type: none"> <li>• 表名（包括视图名、摘要表名、别名和相关名），它们最多可以包含 128 个字节</li> <li>• 列名，它们最多可以包含 30 个字节</li> <li>• 模式名，它们最多可以包含 30 个字节</li> <li>• 对象名也可以包含：             <ul style="list-style-type: none"> <li>- 有效的重音字符（如 ö）</li> <li>- 除多字节空格外的多字节字符（用于多字节环境）</li> </ul> </li> </ul>

### 关于模式名的附加信息

- 不能复制模式名超过 18 个字节的表。
- 用户定义类型 (UDT) 不能具有超过 8 个字节的模式名。



- 下列模式名是保留字，因此不能使用：SYSCAT、SYSFUN、SYSIBM 或 SYSSTAT。
- 要避免将来可能发生迁移问题，不要使用以 SYS 开头的模式名。数据库管理器将不允许您使用以 SYS 开头的模式名来创建触发器、用户定义类型或用户定义函数。
- 建议您不要将 SESSION 用作模式名。已说明临时表必须由 SESSION 限定。因此，应用程序有可能说明与持久表同名的临时表，在这种情况下，应用程序逻辑会变得过于复杂。除非是在处理已说明临时表，否则应避免使用模式 SESSION。

表 26. 用户、用户标识和组命名规则

对象	准则
<ul style="list-style-type: none"> <li>• 组名</li> <li>• 用户名</li> <li>• 用户标识</li> </ul>	<ul style="list-style-type: none"> <li>• 组名最多可以包含 8 个字节。</li> <li>• 基于 UNIX 的系统上的用户标识最多可以包含 8 个字符。</li> <li>• Windows 上的用户名最多可以包含 30 个字符。Windows NT 和 Windows 2000 当前的实际限制为 20 个字符。</li> <li>• 当使用 DCE 认证时，用户名的限制为 8 个字符。</li> <li>• 若不使用 DCE 或“客户机”认证，当显式指定了用户名和密码时，支持使用超过 8 个字符的用户名将非 Windows 32 位客户机连接至 Windows NT 和 Windows 2000。</li> <li>• 名称和标识不能：               <ul style="list-style-type: none"> <li>- 为 USERS、ADMINS、GUESTS、PUBLIC、LOCAL 或 <i>SQL Reference</i> 中列示的任何 SQL 保留字。</li> <li>- 以 IBM、SQL 或 SYS 开始。</li> <li>- 包含重音字符。</li> </ul> </li> </ul> <p>在基于 UNIX 的系统上，组和用户可以同名。对于 GRANT 语句，您必须指定您是引用组还是用户。对于 REVOKE 语句，指定用户或组取决于对于具有不同 GRANTEETYPE 值的 GRANTEE，在授权目录表中是否存在多行。</p> <p>在 Windows NT 上，本地组、全局组和用户不能有相同的名称。</p> <p>在 OS/2 上，组和用户不能有相同的名称。</p> <p>注：</p> <ol style="list-style-type: none"> <li>1. 有些操作系统允许区分大小写的用户标识和密码。您应检查操作系统文档，看看是否是这种情况。</li> <li>2. 从成功的 CONNECT 或 ATTACH 返回的授权标识截断为 8 个字符。省略号 (...) 附加到授权标识上，并且 SQLWARN 字段包含警告以指示发生了截断。有关更多信息，参见 <i>SQL Reference</i> 中的 CONNECT 语句。</li> </ol>

## 关于密码的附加信息

您可能需要执行密码维护任务。因为这类任务必须在服务器上进行，并且许多用户并不能或不能轻松地在服务器环境下工作，故执行这些任务可能是一个重大的挑战。DB2 UDB 提供了一个无需在服务器上更新和验证密码的方法。例如，DB2 OS/390 版的版本 5 支持这种更改用户密码的方法。若收到出错信息 SQL1404N “密码到期”，则使用 CONNECT 语句来更改密码，如下所示：

```
CONNECT TO <database> USER <userid> USING <password>  
NEW <new_password> CONFIRM <new_password>
```

也可使用 “DB2 客户机配置辅助程序” (CCA) 的 “密码更改” 对话来更改密码。有关这些密码更改方法的更多信息，参考 *SQL Reference* 和 CCA 联机帮助。

表 27. 联合数据库对象命名规则

对象	准则
<ul style="list-style-type: none"><li>• 函数映射</li><li>• 索引规范</li><li>• 别名</li><li>• 服务器</li><li>• 类型映射</li><li>• 用户映射</li><li>• 包装器</li></ul>	<ul style="list-style-type: none"><li>• 别名、映射、索引规范、服务器和包装器名不能超过 128 个字节。</li><li>• 服务器和别名选项以及选项设置限制为 255 个字节。</li><li>• 联合数据库对象的名称也可以包含：<ul style="list-style-type: none"><li>– 有效的重音字母（如 ö）</li><li>– 除多字节空格外的多字节字符（用于多字节环境）</li></ul></li></ul>

---

## 在对象名中使用定界标识符

可以使用关键字。若关键字用在一个它也可能被解释为 SQL 关键字的上下文中，则必须将它指定为一个定界标识符。

然而，使用定界标识符，有可能创建一个违反这些命名规则的对象，以后使用该对象可能会导致错误。例如，若创建在名称中带有 + 或 - 号的列，并且后来在索引中使用了该列，则当您尝试重组该表时将遇到问题。有关定界标识符的信息，参见 *SQL Reference* 的 “SQL Identifiers” 一节。

---

## 如何在联合系统中保留区分大小写的值

对于分布式请求，有时需要指定在数据源区分大小写的标识符和密码。为了确保将它们传送到数据源时大小写正确，请遵循下列准则：

- 用要求的大小写指定标识符和密码并用双引号引起来。

- 若指定的是用户标识，应将数据源的 *fold\_id* 服务器选项设置为 "n"（否，不更改大小写）。若指定的是密码，应将数据源的 *fold\_pw* 服务器选项设置为 "n"。

用户标识和密码有替代项。若数据源要求用户标识为小写，可以任何大小写指定它，并可将其 *fold\_id* 服务器选项设置为 "l"（以小写形式将此标识发送至数据源）。若数据源要求此标识为大写，可以任何大小写指定它，并将其 *fold\_id* 服务器选项设置为 "u"（以大写形式将此标识发送至数据源）。同样，若数据源要求密码为小写或大写，可将 *fold\_pw* 服务器选项设置为 "l" 或 "u" 来满足此要求。

有关服务器选项的更多信息，参见《管理指南：实现》。

- 若在操作系统命令提示符处用双引号括起区分大小写的标识符或密码，则必须确保系统可对双引号进行正确的语法分析。为此：
  - 在基于 UNIX 的操作系统上，用单引号括起该语句。
  - 在 Windows NT 操作系统上，在每个引号前加一个反斜杠。

例如，DB2 数据源中的许多定界标识符是区分大小写的。假设您希望为驻留在数据源 NORBASE 中的 DB2 for CS 视图 "my\_schema"."wkly\_sal" 创建别名 NICK1。

在基于 UNIX 的系统的命令提示符处，应输入：

```
db2 'create nickname nick1 for norbase."my_schema"."wkly_sal"'
```

在 Windows NT 命令提示符处，应输入：

```
db2 create nickname nick1 for norbase.\."my_schema\".\."wkly_sal\"
```

若从 DB2 命令提示符处输入语句（交互方式），或者若在应用程序中指定它，则不需要单引号或反斜杠。例如，在基于 UNIX 的系统或 Windows NT 上的 DB2 命令提示符处，应输入：

```
create nickname nick1 for norbase."my_schema"."wkly_sal"
```



---

## 附录B. 数据库迁移计划

本节提供迁移过程的概述。注意，您不需要将 DB2 UDB 版本 6 数据库迁移至版本 7。有关迁移 DB2 UDB 版本 5.x 数据库的详细信息可在操作系统的《快速入门》手册中找到。

在迁移数据库时：

- 迁移下列数据库实体：
  - 数据库配置文件
  - 数据库系统目录表
  - 数据库目录
  - 数据库日志文件头
- 更改系统目录表，如下所示：
  - 添加新列。
  - 创建新表。
  - 在 SYSCAT 模式中迁移一组目录视图，并创建一组新目录视图。
  - 在 SYSSTAT 模式中创建一组可更新的目录视图。
  - 在 SYSFUN 模式中保留一组通用的标量函数，并创建一组新的通用标量函数。在数据库迁移期间，只删除并重建 SYSFUN.DIFFERENCE 标量函数。
- 在数据库目录中创建一个数据库历史文件和它的影子。此文件包含备份信息的摘要，若必须复原一个数据库，则可使用该信息，而且无论何时对此数据库执行特定操作，都会更新该文件。也保存备份信息的摘要，以用于表空间的备份和复原操作。

---

### 迁移注意事项

要成功迁移用数据库管理器的先前版本创建的数据库，必须考虑以下各项：

- 『迁移限制』
- 第176页的『安全性与授权』
- 第176页的『存储器需求』
- 第176页的『发行版间的不兼容性』

### 迁移限制

在尝试将数据库迁移至版本 7 之前，应当注意一些特定的先决条件或限制：

- 仅支持自 V5.x 或 V6 的迁移。不支持自 DB2 V1.2 并行版的迁移。在迁移到 V7 之前，必须将较早版本的 DB2（数据库管理器）迁移到 V5.x 或 V6。
- 支持从 V7 客户机发出迁移命令以将数据库迁移至 V7 服务器；然而，不支持从较旧版本的 DB2 客户机发出迁移命令来将数据库迁移至 V7 服务器。
- 不支持平台之间的迁移。
- 数据库中的用户对象不能将 V7 保留的模式名用作对象限定符。这些保留的模式名包括：SYSCAT、SYSSTAT 和 SYSFUN。
- 在迁移数据库之前，必须重命名使用名称 BIGINT、REAL、DATALINK 或 REFERENCE 的用户定义单值类型。
- 您不能迁移处于下列状态之一的数据库：
  - 备份暂挂
  - 前滚暂挂
  - 一个或多个表空间未处于正常状态
  - 事务不一致
- 支持低级别（V5.x 或 V6）的数据库备份的复原，但不支持低级别日志的前滚。

## 安全性与授权

您需要 SYSADM 权限，才可迁移数据库。

## 存储器需求

迁移期间，旧目录和新目录都需要空间。所需的磁盘空间量将随数据库的复杂性以及数据库对象的数目和大小的不同而不同。这些对象包括所有的表和视图。您应当提供至少相当于该数据库目录当前所占空间两倍的磁盘空间。若无足够的磁盘空间，迁移会失败。

若 SYSCAT 表空间是 SMS 类型的表空间，则还应考虑更新与日志文件相关联的数据库配置参数。应增大 *logfilesiz*、*logprimary* 和 *logsecond* 的值，以避免用于这些日志文件的空间用完（SQL1704N，原因码为 3）。若发生这种情况，则须增加日志空间参数并重新发出 MIGRATE DATABASE 命令。

## 发行版间的不兼容性

当计划迁移数据库时，应考虑该产品的两个版本之间的不兼容性的影响。

要利用版本 7 的增强功能，在迁移数据库后，应当调整数据库和数据库管理器配置。要简化此调整，可记录和比较迁移前后的配置参数值。（有关 GET DATABASE CONFIGURATION 命令和 GET DATABASE MANAGER CONFIGURATION 命令的描述，参考 *Command Reference*。）

## 迁移数据库

以下是为迁移数据库而必须执行的步骤。必须启动数据库管理器，才能开始迁移。

### 在迁移之前:

**注:** 必须对上一发行版（在迁移至或安装新发行版前，即为当前发行版）执行预迁移步骤。

1. 验证是否已没有与第175页的『迁移限制』相关的未解决的问题。
2. 断开所有应用程序和最终用户与正在迁移的每个数据库的连接（根据需要使用 LIST APPLICATIONS 命令或 FORCE APPLICATIONS 命令）。
3. 使用 DB2CKMIG 预迁移实用程序来确定是否可迁移该数据库（有关使用此实用程序的详细信息，参阅您的平台的《快速入门》一书）。注意，在 Windows NT 或 OS/2 上，在安装期间会提示您运行此工具，但在基于 UNIX 的系统上，实例迁移期间会自动调用此工具。
4. 备份数据库。

迁移不是可恢复的过程。若在更改版本 6 保留的模式名前备份数据库，则无法使用 DB2 UDB 版本 7 复原该数据库。要复原该数据库，必须使用先前版本的数据库管理器。

**注意！** 若没有数据库的备份，而迁移又失败了，将无法使用 DB2 UDB 版本 7 或数据库管理器的先前版本来复原该数据库。

您还应知道，从执行备份开始到完成升级至版本 7 为止这段时间内所执行的任何数据库事务都是不能恢复的。即，若在安装和迁移至版本 7 完成后的某个时间，需要复原该数据库（至版本 7 级别），则版本 7 安装之前所写的日志就不可用于前滚复原。

### 迁移:

5. 使用以下方法之一迁移数据库：
  - MIGRATE DATABASE 命令
  - RESTORE DATABASE 命令，用于复原数据库的完整备份
  - sqllemgdb - 迁移数据库 API。

**在 OS/2 上:** 在“配置 / 安装 / 分发” (CID) 体系结构环境中工作的 DB2CIDMG 迁移实用程序只在 DB2 OS/2 版上可用。它允许在基于 LAN 的工作站上进行远程无人照管安装和配置。您必须在 LAN 上装有 NetView DM/2，才可使用 CID 迁移。

**在基于 UNIX 的系统上：** 关于您的平台的《快速入门》一书描述了当您不想迁移一个给定实例中的所有数据库时应如何做。

#### 迁移之后:

6. 您可选择使用 DB2UIDDL 实用程序按您自己的计划来简化唯一索引的分阶段迁移管理。(版本 5 中创建的 DB2 版本 5 数据库不需要此工具来利用延迟唯一性检查, 因为版本 5 中创建的所有唯一索引已具有这些语义。然而, 对于先前迁移至版本 5 的数据库, 除非使用 DB2UIDDL 实用程序来更改唯一索引, 否则这些语义不是自动的。)此实用程序为用户表上的唯一索引生成 CREATE UNIQUE INDEX 语句, 并将它们写至一个文件。将此文件作为 DB2 CLP 命令文件运行, 将导致把唯一索引转换为版本 7 的语义。有关使用此实用程序的详细信息, 参考《快速入门》书籍之一。
7. 可选择对那些强烈影响 SQL 查询性能的表发出 RUNSTATS 命令。旧统计信息保留在迁移后的数据库中, 除非调用 RUNSTATS 命令, 否则不更新它们。
8. 您可选择使用 DB2RBIND 实用程序来重新验证所有软件包, 或者允许在第一次使用软件包时以隐含方式允许软件包重新验证。
9. 若计划在版本 7 中使用解释表, 则可选择迁移它们。有关更多信息, 参见《管理指南: 性能》中的“SQL 解释设施”。
10. 调整数据库和数据库管理器配置参数以利用版本 7 的增强功能。



---

## 附录C. 发行版间的不兼容性

本节标识存在于 DB2 通用数据库和 DB2 的先前发行版之间的不兼容性。

不兼容性是指 DB2 通用数据库中工作方式不同于先前发行版 DB2 的那一部分。若在现存应用程序中使用，将生成预期不到的结果，使更改应用程序成为必要，或降低性能。在本上下文中，“应用程序”是指：

- 应用程序代码
- 第三方实用程序
- 交互式 SQL 查询
- 命令或 API 调用。

描述了“DB2 通用数据库”版本 6 和版本 7 引入的不兼容性。它们根据下列类别分组：

- 系统目录视图
- 应用程序编程
- SQL
- 数据库安全性和调整
- 实用程序和工具
- 连通性和共存
- 配置参数。

每个不兼容性小节都包括该不兼容性的描述、该不兼容性的症状或效果，以及可能的解决方案。每个不兼容性描述的开头还有一个指示符，它标识该不兼容性适用的操作系统：

**WIN** DB2 支持的 Microsoft Windows 平台

**UNIX** DB2 支持的基于 UNIX 的平台

**OS/2** OS/2

**注：**对于“DB2 通用数据库”版本 6，不再支持版本 1.x 和版本 2.x 客户机，包括“DB2 并行版”版本 1.2 服务器附带封装的客户机。

---

## “DB2 通用数据库”计划的不兼容性

本节描述“DB2 通用数据库”用户在编码新应用程序或修改现存应用程序时应该记住的将来的不兼容性。这将简化面向将来版本的 DB2 UDB 的迁移。

### “DB2 通用数据库”未来版本中的只读视图

WIN	UNIX	OS/2
-----	------	------

#### 更改

系统目录视图将是只读视图。SYSSTAT 视图将继续是可更新的。

#### 症状

应用于 SYSCAT 视图中的列的 UPDATE 语句将失败。

#### 说明

编写的工具或应用程序通过更新 SYSCAT 视图中定义的列来更改目录中的值。

#### 解决方案

更改该工具或应用程序，以便通过更新 SYSCAT 视图中定义的列来更改目录。

### “DB2 通用数据库”未来版本中的 PK\_COLNAMES 和 FK\_COLNAMES

WIN	UNIX	OS/2
-----	------	------

#### 更改

SYSCAT.REFERENCES 的列 PK\_COLNAMES 和 FK\_COLNAMES 不再可用。

#### 症状

列不存在并返回错误。

#### 说明

编写的工具或应用程序使用过时的 PK\_COLNAMES 和 FK\_COLNAMES 列。

#### 解决方案

更改该工具或应用程序，以改用 SYSCAT.KEYCOLUSE 视图。

### 在“DB2 通用数据库”未来版本中 COLNAMES 不再可用

WIN	UNIX	OS/2
-----	------	------

### 更改

SYSCAT.INDEXES 的列 COLNAMES 不再可用。

### 症状

列不存在并返回错误。

### 说明

编写的工具或应用程序使用过时的 COLNAMES 列。

### 解决方案

更改该工具或应用程序，以改用 SYSCAT.INDEXCOLUSE 视图。

---

## “DB2 通用数据库”版本 7 的不兼容性

本节标识“DB2 通用数据库”版本 7 引入的不兼容性。

### 应用程序编程

#### Query Patroller 通用客户机

WIN	UNIX	OS/2
-----	------	------

**更改:** 因为有新的存储过程，所以这个新版本的客户机应用程序使能器 (CAE) 只能使用“Query Patroller 服务器”版本 7。CAE 是 DB2 的应用程序接口，所有应用程序最终必须通过它来访问数据库。

**症状:** 若对后面级服务器运行此 CAE，则返回信息 SQL29001。

#### 对象变换函数和结构化类型

WIN	UNIX	OS/2
-----	------	------

**更改:** 在版本 7 之前的客户机与版本 7 服务器之间有一个次要的且可能性极小的不兼容性，此不兼容性与对 SQLDA 所作的更改相关。如 *Application Development Guide* 所述，除了值 X'00' 和 X'01' 之外，第二个 SQLVAR 的字节 8 的值现在还可以是 X'12'。不期望新值的应用程序可能会受此扩展影响。

**解决方案:** 因为将来的发行版可能会对此字段进行其他扩展，所以建议开发者仅测试显式定义的值。

## JVM 使用的类和 Jar 文件的版本

WIN	UNIX	OS/2
-----	------	------

**更改:** 以前, 在 Java 存储过程或用户定义函数 (UDF) 启动后, “Java 虚拟机” (JVM) 便锁定 CLASSPATH 中给出的所有文件 (包括 sqllib/function 中的那些文件)。在数据库管理器停止之前, JVM 使用这些文件。根据运行存储过程或 UDF 的环境的不同 (即, 根据 *keepdari* 数据库管理器配置参数的值, 以及该存储过程是否受保护), 刷新类允许您替换类和 jar 文件, 而不会停止数据库管理器。这与先前行为不同。

“安装 Jar”、“替换 Jar”和“除去 Jar”命令更改了功能

WIN	UNIX	OS/2
-----	------	------

**更改:** 以前, 安装 jar 会导致清除所有 DARI (数据库应用程序远程接口) 进程。这样, 保证在下次调用时挑选新的存储过程类。目前, jar 命令不清除 DARI 进程。要确保挑选新安装或替换的 jar 中的类, 必须显式发出 SQLEJ.REFRESH\_CLASSES 命令。

由于不清除 DARI 进程而引入的另一不兼容性是这样的事实: 对于受保护存储过程, 当 *keepdari* 数据库管理器配置参数的值设置为 “YES” 时, 客户机可能会获得不同版本的 jar 文件。考虑以下方案:

1. 用户 A 替换一个 jar, 但未刷新类。
2. 然后, 用户 A 从该 jar 调用一个存储过程。假定此调用使用同一 DARI 进程, 则用户 A 将得到旧版本的 jar 文件。
3. 用户 B 调用同一存储过程。此调用使用一个新的 DARI, 这表示新创建的类装入程序将挑选新版本的 jar 文件。

换言之, 若在 jar 操作后不刷新类, 则可能会调用不同版本的 jar 的存储过程, 这取决于正在使用的 DARI 进程。这与先前行为不同, 先前行为 (通过清除 DARI 进程) 确保始终使用新类。

### 32 位应用程序不兼容性

	UNIX	
--	------	--

**更改:** 32 位可执行文件 (DB2 应用程序) 不能对新的 64 位数据库引擎运行。

**症状:** 应用程序连接失败。当尝试对 64 位 DB2 应用程序库连接 32 位目标时, 显示操作系统连接程序错误信息。

**解决方案:** 必须将该应用程序重新编译为 64 位可执行程序，并对新的 64 位 DB2 库重新连接。

### 更改暂存区的长度字段

WIN	UNIX	OS/2
-----	------	------

**更改:** 现在，任何更改传送给用户定义函数 (UDF) 的暂存区的长度字段的 UDF 都将接收到 SQLCODE -450。

**症状:** 更改暂存区的长度字段的 UDF 失败。调用语句接收到 SQLCODE -450，其中填入了模式和特定函数名。

**解决方案:** 重新编写 UDF 主体，不更改暂存区的长度字段。

## SQL

### 使用由模式 **SESSION** 限定的规则表的应用程序

WIN	UNIX	OS/2
-----	------	------

**更改:** 临时表只允许 **SESSION** 模式，现在，DB2 使用此模式来指示 **SESSION** 限定的表可引用临时表。然而，**SESSION** 不是一个为临时表保留的关键词，可以用作规则基本表的模式。因此，应用程序会发现同时存在一个 **SESSION.T1** 实表和一个 **SESSION.T1** 已说明临时表。当绑定软件包时，若遇到包含由 "SESSION"（显式或隐式）限定的表引用的静态语句，则既不将此语句的一节也不将此语句的从属存储在目录中。而是，需要在运行时以增量方式绑定此节。这将把该节的一个副本放在动态 SQL 高速缓存中，高速缓存的副本供该应用程序的唯一实例专用。在运行时，若存在与该表名相匹配的已说明临时表，则即使存在同名的持久基本表，也使用该已说明临时表。

**症状:** 在版本 6（和更早版本）中，任何带有包含由 **SESSION** 限定的表的静态语句的软件包都始终引用持久基本表。当绑定该软件包时，将该语句的一节以及相关从属记录保存在目录中。在版本 7 中，这些语句不在绑定时绑定，可以在运行时将它们解析为同名的已说明临时表。因此，会发生下列情况：

- 从版本 5 迁移。若版本 5 中存在这样的软件包，则将在版本 6 中再次绑定它，静态语句现在将以增量方式绑定。这会影响性能，因为除了不能在其他应用程序（甚至是同一应用程序可执行程序的不同实例）之间共享高速缓存的动态节外，这些以增量方式绑定的节的行为与高速缓存的动态 SQL 相同。
- 从版本 6 迁移至版本 7。即使版本 6 中存在这样的软件包，也无需在版本 7 中再次绑定它。而是，那些语句仍作为常规静态 SQL 执行，使用在原始绑定时保

存在目录中的节。然而，若重新绑定此软件包（无论是隐式绑定还是显式绑定），则不再存储软件包中具有 SESSION 限定的表引用的语句，它们将需要增量绑定。这会降低性能。

总而言之，在版本 7 中绑定的具有引用 SESSION 限定表的静态语句的任何软件包，都不再象静态 SQL 那样执行，因为它们需要增量绑定。事实上，若应用程序进程向与现存 SESSION 限定表、视图或别名同名的一个表发出 DECLARE GLOBAL TEMPORARY TABLE 语句，则对那些对象的引用始终被改为引用已说明临时表。

**解决方案：** 有可能的话，更改持久表的模式名，使它们不是 "SESSION"。否则，没有其他方法，性能会受到影响，并可能与已说明临时表发生冲突。

可使用以下查询来标识在应用程序使用临时表时可能会受影响的表、视图和别名：

```
select tabschema, tablename from SYSCAT.TABLES where tabschema = 'SESSION'
```

可使用以下查询来标识在目录中存储了静态节、并且在重新绑定软件包时其行为可能会更改的版本 7 绑定软件包（仅当从版本 6 移至版本 7 时才相关）：

```
select pkgschema, pkgname, bschema, bname from syscat.packagedep
where bschema = 'SESSION' and btype in ('T', 'V', 'I')
```

## 实用程序和工具

### Solaris 上的 DataLinks 文件管理器和文件系统过滤器

	UNIX	
--	------	--

**更改：** “DataLinks 文件管理器”和“文件系统过滤器”在 Solaris OS 2.5.1 上不受支持。

### AIX 和 Solaris 上的 db2set

	UNIX	
--	------	--

**更改：** 命令 "db2set -ul (user level)" 及其相关函数未移植至 AIX 或 Solaris。

### DataLinks 文件系统和 Norton 实用程序\*\*

WIN		
-----	--	--

**更改：** Windows NT DataLinks 文件系统与 Norton 实用程序不兼容。

**症状:** 当从 DLFS 所控制的驱动器中删除文件时, 内核异常导致: 错误 0x1E (未处理内核方式异常)。该异常为 0xC0000005 (访问冲突)。

**说明:** 发生此访问冲突的原因是因为在装入 DLFS 过滤器驱动程序之后装入 “Norton 实用程序” 驱动程序。

**解决方案:** 可以通过在装入 “Norton 实用程序” 驱动程序之后再装入 DLFS 驱动程序来解决此问题。通过单击 “开始”, 然后选择 “设置—> 控制面板—> 设备—> DLFS” 并将它设置为手工, 将 DLFS 驱动程序启动更改为手工。

可以创建在系统启动上装入 DLFS 驱动程序和 “DLFM 服务” 的批处理文件。批处理文件的内容如下:

```
net start dlfsd
net start "dlfm service"
```

将此批处理文件命名为 `start_dlfs.bat`, 并将它复制到 **WINNT\Profiles\Administrator\Start Menu\Programs\Startup** 目录中。只有管理员才有特权装入 DLFS 过滤器和 DLFM 服务。

## 连通性和共存

### 32 位客户机不兼容性

WIN	UNIX	OS/2
-----	------	------

**更改:** 32 位客户机不能连接实例, 也不能与 64 位服务器上的数据库连接。

**症状:** 若客户机和服务器都在运行版本 7 代码, 则返回 SQL1434N; 否则, 连接失败, 并返回 SQLCODE -30081。

**解决方案:** 使用 64 位客户机。

---

## “DB2 通用数据库版本 6” 的不兼容性

本节标识 “DB2 通用数据库” 版本 6 引入的不兼容性。

### 系统目录视图

#### “DB2 通用数据库版本 6” 中的系统目录视图

WIN	UNIX	OS/2
-----	------	------

**更改:** 在系统目录视图中引入了新的代码: "U" 表示类型表, "W" 表示类型视图。

**症状:** 使用类型码 "T" 表示表, "V" 表示视图, 在系统目录中搜索表和视图的查询再不会查找到类型表和类型视图。

**说明:** 几个系统目录, 包含称为 TABLES、PACKAGEDEP、TRIGDEP 和 VIEWDEP 的系统目录视图, 都有称为 TYPE 或 BTYPE 的一列, 它们包含单字母类型码。在版本 5.2 中, 类型码 "T" 用来表示所有的表, 类型码 "V" 用来表示所有的视图。在版本 6 中, 继续用类型码 "T" 表示非类型表, 而用新的类型码 "U" 表示类型表。同样, 继续用类型码 "V" 表示非类型视图, 而用新的类型码 "W" 表示类型视图。另外, 在系统目录表中将出现一种新的表, 它称为层次结构表, 用类型码 "H" 表示, 用户不能直接创建这种类型表, 而系统可使用它来实现表的层次结构。

**解决方案:** 更改该工具或应用程序, 以便可识别类型表和类型视图的代码。若该工具或应用程序需要表的逻辑视图, 应使用类型码 "T"、"U"、"V" 和 "W"。若该工具或应用程序需要表的物理视图, 包括层次结构表, 应使用类型码 "T" 和 "H"。

#### “DB2 通用数据库版本 6”中的主键列名和外键列名

WIN	UNIX	OS/2
-----	------	------

**更改:** SYSCAT.REFERENCES 的两列 PK\_COLNAMES 和 FK\_COLNAMES 的数据类型, 由 VARCHAR(320) 更改为 VARCHAR(640)。

**症状:** 主键或外键的列名被截断、不正确或丢失。

**说明:** 当主键或外键的列名长度超过 18 字节时, 则列名的列表不能以相同的格式保存在这两列中。在长度 ( $n$ ) 超过 18 个字节的列后面出现的、以 20 字节空白定界的列名, 将向右移  $n-18$  个字节。另外, 若列名的列表超过 640 字节, 该列将包含空字符串。

**解决方案:** SYSCAT.KEYCOLUSE 视图包含构成一个主键、外键和唯一键的那些列的列表, 并应该用来替代 SYSCAT.REFERENCES 中的那些列。或者, 用户可以限制列名的长度为 18 字节, 或限制列的列表总长度为 640 字节。

#### “DB2 通用数据库版本 6”中的 SYSCAT.VIEWS 列 TEXT

WIN	UNIX	OS/2
-----	------	------



**更改:** SYSCAT.VIEWS 列 TEXT 中的视图文本将不再分成多行。该数据类型已由 VARCHAR(3600) 更改为 CLOB(64K)。

**症状:** 工具或应用程序没有给出完整的视图文本。

**说明:** 工具或应用程序被编写为一次从 TEXT 列返回的视图文本不会超过 3600（或 3900）字节，这样的话，则不能处理此字段增加的大小。不再需要使用 SEQNO 字段检索多行并重构视图文本的机制。SEQNO 值将始终是 1。

**解决方案:** 更改该工具或应用程序，以便能够处理来自 TEXT 列的大于 3600 字节的值。或者，重写视图 TEXT 以便不超过 3600 字节。

#### “DB2 通用数据库版本 6”中的 SYSCAT.STATEMENTS 列 TEXT

WIN	UNIX	OS/2
-----	------	------

**更改:** SYSCAT.STATEMENTS 列 TEXT 中的语句文本将不再分成多行。该数据类型已由 VARCHAR(3600) 更改为 CLOB(64K)。

**症状:** 工具或在应用程序没有给出完整的语句文本。

**说明:** 工具或应用程序被编写为一次从 TEXT 列返回的视图文本不会超过 3600（或 3900）字节，这样的话，则不能处理此字段增加的大小。不再需要使用 SEQNO 字段检索多行并重构语句文本的机制。SEQNO 值将始终是 1。

**解决方案:** 更改该工具或应用程序，以便能够处理来自 TEXT 列的大于 3600 字节的值。或者，重写语句 TEXT 以便不超过 3600 字节。

#### “DB2 通用数据库版本 6”中的 SYSCAT.INDEXES 列 COLNAMES

WIN	UNIX	OS/2
-----	------	------

**更改:** SYSCAT.INDEXES 列 COLNAMES 的数据类型由 VARCHAR(320) 更改为 VARCHAR(640)。

**症状:** 索引中的列名丢失。

**说明:** 编写的工具或应用程序只能从数据类型为 VARCHAR(320) 的列检索数据，不能处理此字段增加的字节。

**解决方案:** SYSCAT.INDEXCOLUSE 视图包含构成一个索引的那些列的列表，应用来代替 COLNAMES 列。或者，从索引中除去一列，或减小列名大小，以便列名（前缀为 + 或 -）的列表不超过 320 字节。

### “DB2 通用数据库版本 6” 中的 SYSCAT.CHECKS 列 TEXT

WIN	UNIX	OS/2
-----	------	------

**更改:** CHECKS 列 TEXT 的数据类型由 CLOB(32K) 更改为 CLOB(64K)。

**症状:** 检查约束子句不完整。

**说明:** 编写的工具或应用程序只能从数据类型为 CLOB(32K) 的列检索数据，不能处理此字段增加的字节。

**解决方案:** 更改该工具或应用程序，以便能够处理来自 TEXT 列的长于 32 KB 的值。另外，重写检查约束子句以便使用更少的字符，使它不超过 32 KB。

### “DB2 通用数据库版本 6” 中列数据类型更改为 BIGINT

WIN	UNIX	OS/2
-----	------	------

**更改:** 几个系统目录视图已将其列数据类型由 INTEGER 更改为 BIGINT。

**症状:** 值比期望的要小得多（或大得多），特别是统计信息。

**说明:** 编写的工具或应用程序只能从数据类型为 INTEGER 的列检索数据，不能处理此字段增加的字节。

**解决方案:** 更改该工具或应用程序，以便能够处理比可存储在 INTEGER 字段中的最大值更大或比最小值更小的值。另外，更改导致该值落在可在 INTEGER 字段中表示的值外部的的基本结构或 SQL 代码。

### “DB2 通用数据库版本 6” 中的列不匹配

WIN	UNIX	OS/2
-----	------	------

**更改:** 在 SYSCAT 视图定义中，没有在视图结尾插入新列。

**症状:** 由于几个列不匹配或列数据类型不匹配，重新预处理失败。

**说明:** 将新列插入系统目录视图，并将其放在对特定查询环境有利的位置；确切地说，将较短的列放在很长的列之前，REMARKS 列始终是最后一列。

**解决方案:** 在 select 列表中显式命名那些列，而不是编码 “SELECT \*”。

## “DB2 通用数据库版本 6”中的 SYSCAT.COLUMNS 和 SYSCAT.ATTRIBUTES

WIN	UNIX	OS/2
-----	------	------

**更改:** SYSCAT.COLUMNS 和 SYSCAT.ATTRIBUTES 现在包含继承的列和属性的项。

**症状:** 若查询的主题是子表、子视图或子类型，则对 SYSCAT.COLUMNS 的查询（检索类型表或类型视图的列）和对 SYSCAT.ATTRIBUTES 的查询（检索结构化类型的属性）在版本 6 中返回的行数比在版本 5.2 中返回的多。

**说明:** 在版本 5.2 中，对于给定的表、视图或结构化类型，COLUMNS 和 ATTRIBUTES 目录只包含通过该表、视图或类型引进的列和属性的项。从超表或超类型继承的列和属性不出现在目录中。然而，在版本 6 中，COLUMNS 和 ATTRIBUTES 目录将包含继承的列和属性的项。

**解决方案:** 更改该工具或应用程序，以便可识别 COLUMNS 和 ATTRIBUTES 目录中的新项。

### “DB2 通用数据库版本 6”中不再支持 OBJCAT 视图

WIN	UNIX	OS/2
-----	------	------

**更改:** 版本 5.2 的 OBJCAT 模式中的递归目录视图不再是交付的“DB2 通用数据库”产品的一部分。

**症状:** 针对 OBJCAT 目录视图编写的查询将不再能够成功运行。

**解决方案:** OBJCAT 视图中大多数先前的信息已合并到规则的 SYSCAT 目录视图中。在大多数情况下，可以从系统目录视图中获取信息。若从版本 5.2 迁移，且 OBJCAT 目录视图存在，则应将它们删除。为此，可运行 CLP 脚本 objcatdp.db2，该脚本可在 sqllib 目录的 misc 子目录中找到。

也可创建一组自己的 OBJCAT 视图，它们与版本 5.2 中支持的目录视图等效。

在版本 5.2 中，*SQL Reference* 的附录 E 提醒用户 OBJCAT 目录视图只是临时的，在未来版本中不受支持。

### “DB2 通用数据库版本 6”中更改的关联代码

WIN	UNIX	OS/2
-----	------	------

**更改:** 在系统目录视图中, 以前由 "H" 表示的分层从属性现在由 "O" 表示。

**症状:** 在目录视图中搜索由代码 "H" 表示的分层从属性的查询将不再生效。

**说明:** 几个系统目录, 包含系统目录视图 PACKAGEDEP、TRIGDEP 和 VIEWDEP, 都有 BTYPE 列。在版本 5.2 中, OBJCAT 视图通过代码 "H" 表示分层从属性。在版本 6 中, 这些从属性由代码 "O" 表示。

**解决方案:** 修订这些查询, 以搜索代码 "O"。

### “DB2 通用数据库版本 6”中的 SYSIBM 基本目录表

WIN	UNIX	OS/2
-----	------	------

**更改:** 以下是对 SYSIBM 基本目录表的更改, 仍可使用它来代替 SYSCAT 视图:

- 删除了字段 (但仍在 SYSCAT 视图中):
  - SYSSTMT.SEQNO
  - SYSVIEWS.SEQNO
- 重命名了目录表: SYSTRIGDEP 更改为 SYSDEPENDENCIES。并且, 将列 BCREATOR 和 DCREATOR 分别重命名为 BSCHEMA 和 DSCHEMA。视图 SYSCAT.TRIGDEP 不变。
- 删除了字段 (未曾在 SYSCAT 视图中出现过):
  - SYSATTRIBUTES.DEFAULT\_VALUE
  - SYSATTRIBUTES.NULLS
  - SYSCOLUMNS.SERVERTYPE
  - SYSDATATYPES.REFREP\_TYPENAME
  - SYSDATATYPES.REFREP\_TYPESHEMA
  - SYSDATATYPES.REFREP\_LENGTH
  - SYSDATATYPES.REFREP\_SCALE
  - SYSDATATYPES.REFREP\_CODEPAGE
  - SYSINDEXES.TEXT  
(以前在视图中, 但保留下来仅供将来使用。)
  - SYSPLANDEP.PUBLICPRIV
  - SYSSECTION.SEQNO
  - SYSTABAUTH.UPDATE\_BY\_COLS
  - SYSTABAUTH.REF\_BY\_COLS

- SYSTABLES.MINPDLENGTH
- SYSTABLESPACES.READONLY
- SYSTABLESPACES.REMOVABLEMEDIA
- 数据类型更改:
  - SYSSECTION.SECTION, 由 VARCHAR(3600) 更改为 CLOB(10M)
  - SYSPLANDEP.COLUSAGE, 由 VARCHAR(3000) FOR BIT DATA 更改为 BLOB(5K)

## 应用程序编程

### “DB2 通用数据库版本 6”中的 VARCHAR 数据类型

WIN	UNIX	OS/2
-----	------	------

**更改:** 在版本 6 中, VARCHAR (VARGRAPHIC) 数据类型的最大可能大小已由 4000 字符 (2000 双字节字符) 增加到 32672 字符 (16336 双字节字符)。

**症状:** 对 VARCHAR (VARGRAPHIC) 数据类型使用 4000 字节定长缓冲区的应用程序, 若它将一个大于 4000 字节的 VARCHAR 字段读入一个很小的缓冲区中, 可能造成缓冲区被覆盖或截断。CLI 函数 - SQLGetTypeInfo() 现在将 VARCHAR 的大小返回为 32672。由于没有足够页大小的表空间可用, 在表 DLL 中使用此值的 CLI 应用程序可能收到错误信息。有关表空间页大小的更多信息, 参见第84页的『用户表数据』。

**解决方案:** 当编码应用程序时, 建议首先描述结果集的列 (使用 DESCRIBE 语句), 然后使用大小基于 DESCRIBE 语句返回的长度的缓冲区。

### 在“DB2 通用数据库版本 6”中通过 Java 编程定位的 UPDATE 和 DELETE

WIN	UNIX	OS/2
-----	------	------

**更改:** 在版本 6 中使用 Java 进行编程时, 定位的 UPDATE 和 DELETE 语句使用绑定游标软件包的那个人的缺省授权标识符。这不同于版本 5.2, 在版本 5.2 中使用运行软件包的那个人的授权标识符。

**症状:** 包含定位 UPDATE 和 DELETE 语句的软件包可能不能运行, 因为绑定软件包的那个人的授权标识符没有足够的权限。

**解决方案:** 必须给绑定软件包的那个人的授权标识符授予足够的权限, 以运行软件包中的定位 UPDATE 和 DELETE 语句。授予正确的特权并重新绑定软件包。

## “DB2 通用数据库版本 6” 中 FOR UPDATE 子句的语法更改

WIN	UNIX	OS/2
-----	------	------

**更改:** 在版本 5.2 中, 可在 SQLJ 程序中使用 SELECT 语句中的 FOR UPDATE 子句来标识可在后续的定位 UPDATE 语句中更新的列。在版本 6 中, 语法已更改。

**症状:** 若 SELECT 语句包含 FOR UPDATE 子句, 将接收到出错信息 SQJ0204E。

**解决方案:** 除去 SELECT 语句中的 FOR UPDATE 子句。通过迭代声明子句指定可更新的迭代器。例如:

```
#sql public iterator DelByName implements sqlj.runtime.ForUpdate(String EmpNo)
with updateColumns = (salary);
```

若想显式标识哪些列可更新, 通过 updateColumns 关键字以及 WITH 子句指定它们。

有关已定位迭代器声明的更多信息, 参考 *Application Development Guide*。

## “DB2 通用数据库版本 6” 中的字符名大小

WIN	UNIX	OS/2
-----	------	------

**更改:** “DB2 通用数据库版本 6” 支持 128 字节的表名、视图名和别名以及 30 字节的列名。先前版本中的上述每个实体名只能为 18 字节。

USER 和 CURRENT SCHEMA 专用寄存器以前是 CHAR(8), 而现在是 VARCHAR(128)。CURRENT EXPLAIN MODE 专用寄存器以前是 CHAR(8), 而现在是 VARCHAR(254)。TYPE\_SCHEMA 和 TABLE\_SCHEMA 内置函数的输出以前是 CHAR(8), 而现在是 VARCHAR(128)。

**症状:** 若对没有使用长名限制的版本 6 数据库运行早于版本 6 开发的应用程序, 则该应用程序的行为不会有任何更改。然而, 若对确实使用了长名的版本 6 数据库运行这些应用程序, 可能导致某些副作用, 这取决于这些应用程序是如何编码的。

下面是一些示例:

- 考虑一个现存的应用程序, 它将一个表名或列名 (一般来自目录视图) FETCH 进被定义为 18 字节长的主变量中。由于 18 字节是版本 6 之前的表名或列名的

大小限制，所以此应用程序在检查 SQLCA 的 sqlwarn1 位时可能不会遇到麻烦。将（不正确地）认为不会发生截断。

- 考虑一个应用程序，它将表名或列名（一般来自目录视图）FETCH 进 SQLDA 中，其中 sqldata 字段的大小根据 sqlllen 字段（由该 SELECT 的 DESCRIBE 确定）分配。即使表名或列名的大小可能已增加，应用程序也会得到正确（未被截断）的结果。若其他应用程序逻辑假定列名限长 18 字节，则可能会以意外的方式处理更长的名称；例如，更长的列名的显示可能在第 18 个字节后被截断。
- 由于 SQLCA 记号字段 (sqlerrmc) 限长 70 字节，尝试向表中插入一行的现存应用程序可能会受到影响。作为对错误 SQL0204N 的响应，这样的应用程序根据 SQLCA sqlerrmc 字段确定表名，然后根据该对象名执行一些操作。对于较早版本的 DB2，表或模式标识符限制保证整个表名可以包含在 SQLCA 中。在版本 6 中，情况并非如此。
- 使用后面级 API 的应用程序将只得到表名的前 18 个字节。
- 当与一个支持 18 字节名称的服务器连接时，使用模式函数（如 SQLTables()、SQLColumns() 等）的现存的 CLI 和 ODBC 应用程序将会受影响。尽管会出现截断警告，但应用程序可能会忽略此警告，使用截断名称继续运行。

**解决方案：** 解决这类问题的最好方法是重新编码应用程序，以便可处理更长的表名和列名。否则，确保不要对使用超过 18 字节的名称的版本 6 数据库运行这些应用程序。

#### “DB2 通用数据库版本 6”中的 PC/IXF 格式更改

WIN	UNIX	OS/2
-----	------	------

**更改：** “DB2 通用数据库版本 6”支持 128 字节的表名、视图名和别名以及 30 字节的列名。先前版本中的上述每个实体名只能为 18 字节。

**症状：** “DB2 通用数据库版本 5”客户机不能导入由“DB2 通用数据库版本 6”客户机导出的 PC/IXF 文件（错误 SQL3059N）。（从“DB2 通用数据库版本 6”客户机中导出的）PC/IXF 文件不能装入“DB2 通用数据库版本 5”数据库中（错误 SQL3059N）。

**解决方案：** 当导入或装入 PC/IXF 数据时，需使用“DB2 通用数据库”的兼容版本。

## “DB2 通用数据库版本 6”中非双精度 SQLVAR 中的 SQLNAME

WIN	UNIX	OS/2
-----	------	------

**更改:** “DB2 通用数据库版本 6”支持 30 字节的列名。先前版本只支持 18 字节的名称。在版本 5 中，所记载的行为是将 “0xFF” 放在非双精度 SQLVAR 的 SQLNAME 字段的第 30 个字节中；对于系统生成的名称以及用户在 “AS” 子句中指定的列名，还将 “0x00” 放在第 30 个字节中。

在版本 6 中，仅当名称是系统生成的时，才在第 30 个字节中返回 “0xFF”。

**症状:** 若用户指定的列名由 30 个字符组成，根据 SQLNAME 字段的第 30 个字节来确定是用户指定的列名还是系统生成的名称的任何应用程序，可能接收到意外的逻辑检查结果。这种情况几乎不会发生。

**解决方案:** 若 SQLNAME 字段的长度小于 30 字节，应修改这些应用程序应以便只检查该字段的第 30 个字节处的 “0xFF”。在这种情况下，名称是由用户生成的。

## “DB2 通用数据库版本 6”中过时的 DB2 CLI/ODBC 配置关键字

WIN		
-----	--	--

**更改:** 当迁移至新版本的 DB2 UDB 时，通过在 db2cli.ini 文件中指定一组可选关键字，可更改 DB2 CLI/ODBC 驱动程序的行为。

在版本 6 中，TRANSLATEDLL 和 TRANSLATEOPTION 关键字已过时。

**症状:** 若这些关键字仍存在，则会被忽略。可能注意到由于除去了这些设置而引起的行为更改。

**解决方案:** 将需要复查新的有效参数列表以决定适合于您的环境的关键字和设置。有关这些关键字的信息，参考 *CLI Guide and Reference*。

## “DB2 通用数据库版本 6”中的事件监视器输出流格式

WIN	UNIX	OS/2
-----	------	------

**更改:** 事件监视器输出流没有版本控制。因此，要增加对超过 18 字节的表名的支持，要求更改为一种输出流格式。

**症状:** 对事件监视器输出流进行语法分析的应用程序不再能够正确工作。



**解决方案:** 您有两个选择:

- 更新应用程序以便使用新的数据流。
- 设置注册表变量

```
DB20LDEVMON=evmonname1,evmonname2,...
```

其中, *evmonname* 是要以旧数据格式编写的事件监视器的名称。注意, 事件监视器中的任何新字段都不可以旧数据格式访问。

## SQL

### DB2 通用数据库版本 6 中的 DATALINK 列

	UNIX	
--	------	--

**更改:** 在“DB2 通用数据库”版本 6 下面插入的 DATALINK 值要求在列值描述符中多占四字节的空間。

**症状:** 当更新在版本 5.2 中创建的 DATALINK 列时, 需要在数据页中多占四字节以便存储新的列值。结果是, 数据页没有足够多的空间完成更新, 这样不得不将它移动到新的页中。这可能造成更新操作用完所有的空间。

**解决方案:** 需要在系统上添加更多空间以允许更新。

### “DB2 通用数据库版本 6”中的 SYSFUN 字符串函数特征

WIN	UNIX	OS/2
-----	------	------

**更改:** SYSFUN 模式中许多字符串函数现已在 SYSIBM 模式 (内置函数) 中定义了改进的版本。这些函数名为 LCASE、LTRIM、RTRIM 和 UCASE。

**症状:** 当准备语句或创建视图时, 从任何这些函数返回的数据类型都可能与在版本 6 中不同。造成这种情况的原因是, 内置函数 (在 SYSIBM 中定义的) 通常在 SYSFUN 模式中的函数解析之前被解析。

**解决方案:** 不需要任何操作。内置函数通常优先于 SYSFUN 模式中的函数。可通过转换 SQL 路径复原先前版本的行为 (以便 SYSFUN 在 SYSIBM 之前发生), 但性能会下降。也可用模式名 SYSFUN 限定函数名来调用先前版本的函数。

若没有执行显式操作, 如显式绑定软件包、重新创建视图、摘要表、触发器或约束, 则引用这些函数的已迁移的软件包、视图、摘要表、触发器和约束将继续使用 SYSFUN 模式中的版本。

## SET INTEGRITY 替换 SET CONSTRAINTS

WIN	UNIX	OS/2
-----	------	------

**更改:** SET INTEGRITY 语句已替换 SET CONSTRAINTS 语句。为了向下兼容, DB2 版本 6 和版本 7 接受这两种语句。

## DB2 通用数据库版本 6 中 SYSTABLE 列更改为具有新的完整性状态

WIN	UNIX	OS/2
-----	------	------

**更改:** 当运行 SET INTEGRITY ... OFF 语句时, SYSCAT.TABLES 的 CONST\_CHECKED 列中的 "U" 状态发生改变。

**症状:** 在版本 6 之前的版本中, 当运行 SET INTEGRITY ... OFF 语句时, CONST\_CHECKED 列中的 "U" 状态更改为 "N" 状态。"U" 状态现在更改为 "W" 状态。

**解决方案:** 不需要任何操作。CONST\_CHECKED 列中的新状态 "W" 用来指示约束类型以前被该用户检查过, 且表中的某些数据可能需要作完整性检查。

"N" 状态未澄清是否存在数据库管理器尚未验证过的任何旧数据。当后续 SET INTEGRITY ... IMMEDIATE CHECKED INCREMENTAL 语句运行时, 数据库管理器必须返回错误, 因为如果只检查新更改, 便不能保证数据完整性。另一方面, 可将 "W" 状态更改回 "U" 状态 (若指定 INCREMENTAL 选项的话), 以指示用户仍对表的数据完整性负责。若未指定 INCREMENTAL 选项, 数据库管理器将选择完全处理, 将 "W" 状态更改为 "Y" 状态, 并承担维护数据完整性的责任。

## 数据库安全性和协调

### 在 "DB2 通用数据库版本 6" 中使用客户机创建数据库

WIN	UNIX	OS/2
-----	------	------

**更改:** 客户机用来创建数据库的方法。

**症状:** 使用后面级客户机来创建数据库将导致错误。

**解决方案:** 当使用客户机来创建数据库时, 确保客户机和服务器运行同一级别的 DB2 代码。

## “DB2 通用数据库版本 6”中需要对层次结构拥有 SELECT 特权

WIN 32 位	UNIX	OS/2
----------	------	------

**更改:** ONLY 关键字的规范（对于表）现在要求用户对指定类型表的所有子表具有 SELECT 特权。同样，ONLY 关键字的规范（对于视图）现在要求用户对指定类型表的所有子视图具有 SELECT 特权。先前版本的 DB2 只需要指定的表或视图的 SELECT 特权。

**症状:** 此更改有两个可能的症状:

- 当重新绑定包含 SQL 语句（在 FROM 子句中指定 ONLY 关键字）的软件包时，若绑定软件包所用的授权标识缺少命名的类型表（或视图）的子表的 SELECT 特权，将发生授权错误（SQLCODE -551、SQLSTATE 42501）。
- 若一个视图或触发器的定义包含在 FROM 子句中的 ONLY 关键字，则该视图或触发器将继续正常工作。然而，除非创建者拥有命名的表（或视图）的全部子表的 SELECT 特权，否则不能再使用该视图或触发器的定义来创建新的视图或触发器。

**解决方案:** 应给需要重新绑定软件包或创建新视图或触发器的授权标识授予对 ONLY 关键字后指定的表（或视图）的全部子表（和子视图）的 SELECT 特权。

## “DB2 通用数据库版本 6”中过时的简要表注册表变量和环境变量

WIN	UNIX	OS/2
-----	------	------

**更改:** 下列简要表注册表变量或环境变量是过时的:

- DB2\_VECTOR

**解决方案:** 不再需要这些变量。

## 实用程序和工具

### “DB2 通用数据库版本 6”中当前的解释方式

WIN	UNIX	OS/2
-----	------	------

**更改:** “CURRENT EXPLAIN MODE”专用寄存器的类型已由 CHAR(8) 更改为 VARCHAR(254)。

**症状:** 若应用程序认为该类型仍为 CHAR(8)，该值可能由 254 字节截断为 8 字节。

**解决方案:** 将读取专用寄存器的所有主变量的类型由 CHAR(8) 重新定义为 VARCHAR(254)。

此更改一定要包含 "CURRENT EXPLAIN MODE" 专用寄存器的两个新值。这些新值是 "EVALUATE INDEXES" 和 "RECOMMEND INDEXES"。

#### “DB2 通用数据库版本 6”中的 USING 和 SORT BUFFER 参数

WIN	UNIX	OS/2
-----	------	------

**更改:** 在版本 6 中, 不再支持 LOAD 命令的 USING 和 SORT BUFFER 参数。这些参数被忽略。

**症状:** 返回一警告信息, 指出不再支持 USING 和 SORT BUFFER 参数, 装入实用程序将忽略这两个参数。

**解决方案:** 忽略该警告信息。有关其他信息, 参考 *Data Movement Utilities Guide and Reference*。

## 连通性和共存

### 在“DB2 通用数据库版本 6”中用 PCOMM 替换 RUMBA

WIN		
-----	--	--

**更改:** 在版本 6 中, RUMBA 在 Windows NT、Windows 98 和 Windows 95 (而不包括 Windows 3.1) 上被 PCOMM 替换。

**症状:** 无。

**解决方案:** 无。

## 配置参数

### 过时的数据库配置参数

WIN	UNIX	OS/2
-----	------	------

**更改:** 下列数据库配置参数已过时:

- DL\_NUM\_BACKUP (被 NUM\_DB\_BACKUP 数据库配置参数替换)

**解决方案:** 从应用程序中除去对这些参数的所有引用。

---

## 附录D. 本地语言支持 (NLS)

本节包含有关 DB2 提供的本地语言支持 (NLS) 的信息，包括有关国家 / 地区、语言和受支持的代码页（代码集）的信息，并介绍如何在您的数据库和应用程序中配置和使用 DB2 NLS 功能。

---

### 本地语言版本

DB2 版本 7 有英语、法语、德语、意大利语、西班牙语、巴西葡萄牙语、日语、韩语、简体中文、繁体中文、丹麦语、芬兰语、挪威语、瑞典语、捷克语、荷兰语、匈牙利语、波兰语、土耳其语、俄语、保加利亚语和斯洛文尼亚语等版本。

---

### 国家 / 地区代码和代码页支持

第200页的表28显示了受数据库服务器支持的语言和代码集，并显示了如何将这些值映射为数据库管理器使用的国家 / 地区代码或代码页值。

以下是该表每一列的说明：

- **代码页**显示 IBM 定义的代码页，它是从操作系统代码集映射而来的。
- **组**显示代码页是单字节 ("S") 还是双字节 ("D")。"-n" 是一个用于创建字母-数字组合的数。匹配组合显示 DB2 在何处允许连接和转换。例如，所有的 "S-1" 个组都可一起工作。
- **代码集**显示与受支持的语言相关的代码集。代码集被映射为 DB2 代码页。
- **Tr.** 显示领域标识符。
- **国家 / 地区代码**显示一个代码，数据库管理器内部使用该代码来提供特定于地区的支持。
- **语言环境**显示数据库管理器所支持的语言环境值。
- **OS** 显示支持此语言和代码集的操作系统。
- **国家 / 地区名**显示地区或国家的名称。

表 28. 受支持的语言和代码集

代码 页	组	代码集	Tr.	国家 / 地区 代码	语言环境	OS	国家 / 地区名
437	S-1	IBM-437	AL	355	-	OS2	阿尔巴尼亚
850	S-1	IBM-850	AL	355	-	OS2	阿尔巴尼亚
819	S-1	ISO8859-1	AL	355	sq_AL	AIX	阿尔巴尼亚
850	S-1	IBM-850	AL	355	Sq_AL	AIX	阿尔巴尼亚
819	S-1	iso88591	AL	355	-	HP	阿尔巴尼亚
1051	S-1	roman8	AL	355	-	HP	阿尔巴尼亚
819	S-1	ISO8859-1	AL	355	-	Sun	阿尔巴尼亚
1252	S-1	1252	AL	355	-	WIN	阿尔巴尼亚
1275	S-1	1275	AL	355	-	Mac	阿尔巴尼亚
37	S-1	IBM-37	AL	355	-	HOST	阿尔巴尼亚
1140	S-1	IBM-1140	AL	355	-	HOST	阿尔巴尼亚
864	S-6	IBM-864	AA	785	-	OS2	阿拉伯国家
1046	S-6	IBM-1046	AA	785	Ar_AA	AIX	阿拉伯国家
1089	S-6	ISO8859-6	AA	785	ar_AA	AIX	阿拉伯国家
1089	S-6	iso88596	AA	785	ar_SA.iso88596	HP	阿拉伯国家
1256	S-6	1256	AA	785	-	WIN	阿拉伯国家
420	S-6	IBM-420	AA	785	-	HOST	阿拉伯国家
437	S-1	IBM-437	AU	61	-	OS2	澳大利亚
850	S-1	IBM-850	AU	61	-	OS2	澳大利亚
819	S-1	ISO8859-1	AU	61	en_AU	AIX	澳大利亚
850	S-1	IBM-850	AU	61	En_AU	AIX	澳大利亚
819	S-1	iso88591	AU	61	-	HP	澳大利亚
1051	S-1	roman8	AU	61	-	HP	澳大利亚
819	S-1	ISO8859-1	AU	61	en_AU	Sun	澳大利亚
819	S-1	ISO8859-1	AU	61	en_AU	SCO	澳大利亚
1252	S-1	1252	AU	61	-	WIN	澳大利亚
1275	S-1	1275	AU	61	-	Mac	澳大利亚
37	S-1	IBM-37	AU	61	-	HOST	澳大利亚
1140	S-1	IBM-1140	AU	61	-	HOST	澳大利亚
437	S-1	IBM-437	AT	43	-	OS2	奥地利
850	S-1	IBM-850	AT	43	-	OS2	奥地利
819	S-1	ISO8859-1	AT	43	ge_AT	AIX	奥地利
850	S-1	IBM-850	AT	43	Ge_AT	AIX	奥地利
819	S-1	iso88591	AT	43	-	HP	奥地利
1051	S-1	roman8	AT	43	-	HP	奥地利
819	S-1	ISO8859-1	AT	43	de_AT	SCO	奥地利
819	S-1	ISO-8859-1	AT	43	de_AT	Linux	奥地利
819	S-1	ISO8859-1	AT	43	de_AT	Sun	奥地利
1252	S-1	1252	AT	43	-	WIN	奥地利
1275	S-1	1275	AT	43	-	Mac	奥地利
37	S-1	IBM-37	AT	43	-	HOST	奥地利
1140	S-1	IBM-1140	AT	43	-	HOST	奥地利

表 28. 受支持的语言和代码集 (续)

代码 页	组	代码集	Tr.	国家 / 地区 代码	语言环境	OS	国家 / 地区名
915	S-5	ISO8859-5	BY	375	-	OS2	白俄罗斯
915	S-5	ISO8859-5	BY	375	be_BY	AIX	白俄罗斯
1131	S-5	IBM-1131	BY	375	-	OS2	白俄罗斯
1251	S-5	1251	BY	375	-	WIN	白俄罗斯
1283	S-5	1283	BY	375	-	Mac	白俄罗斯
1025	S-5	IBM-1025	BY	375	-	HOST	白俄罗斯
274	S-1	IBM-274	BE	32	-	HOST	比利时
437	S-1	IBM-437	BE	32	-	OS2	比利时
850	S-1	IBM-850	BE	32	-	OS2	比利时
819	S-1	ISO8859-1	BE	32	n1_BE	AIX	比利时
850	S-1	IBM-850	BE	32	N1_BE	AIX	比利时
819	S-1	iso88591	BE	32	-	HP	比利时
819	S-1	ISO8859-1	BE	32	fr_BE	SCO	比利时
819	S-1	ISO8859-1	BE	32	n1_BE	SCO	比利时
819	S-1	ISO-8859-1	BE	32	n1_BE	Linux	比利时
819	S-1	ISO8859-1	BE	32	n1_BE	Sun	比利时
1252	S-1	1252	BE	32	-	WIN	比利时
1275	S-1	1275	BE	32	-	Mac	比利时
500	S-1	IBM-500	BE	32	-	HOST	比利时
1148	S-1	IBM-1148	BE	32	-	HOST	比利时
855	S-5	IBM-855	BG	359	-	OS2	保加利亚
915	S-5	ISO8859-5	BG	359	-	OS2	保加利亚
915	S-5	ISO8859-5	BG	359	bg_BG	AIX	保加利亚
915	S-5	iso88595	BG	359	bg_BG.iso88595	HP	保加利亚
1251	S-5	1251	BG	359	-	WIN	保加利亚
1283	S-5	1283	BG	359	-	Mac	保加利亚
1025	S-5	IBM-1025	BG	359	-	HOST	保加利亚
850	S-1	IBM-850	BR	55	-	OS2	巴西
850	S-1	IBM-850	BR	55	-	AIX	巴西
819	S-1	ISO8859-1	BR	55	pt_BR	AIX	巴西
819	S-1	ISO8859-1	BR	55	-	HP	巴西
819	S-1	ISO8859-1	BR	55	pt_BR	SCO	巴西
819	S-1	ISO8859-1	BR	55	pt_BR	Sun	巴西
819	S-1	ISO-8859-1	BR	55	pt_BR	Linux	巴西
1252	S-1	1252	BR	55	-	WIN	巴西
37	S-1	IBM-37	BR	55	-	HOST	巴西
1140	S-1	IBM-1140	BR	55	-	HOST	巴西

表 28. 受支持的语言和代码集 (续)

代码 页	组	代码集	Tr.	国家 / 地区 代码	语言环境	OS	国家 / 地区名
850	S-1	IBM-850	CA	1	-	OS2	加拿大
850	S-1	IBM-850	CA	1	En_CA	AIX	加拿大
819	S-1	ISO8859-1	CA	1	en_CA	AIX	加拿大
819	S-1	iso88591	CA	1	fr_CA.iso88591	HP	加拿大
1051	S-1	roman8	CA	1	fr_CA.roman8	HP	加拿大
819	S-1	ISO8859-1	CA	1	en_CA	SCO	加拿大
819	S-1	ISO8859-1	CA	1	fr_CA	SCO	加拿大
819	S-1	ISO8859-1	CA	1	en_CA	Sun	加拿大
819	S-1	ISO8859-1	CA	1	en_CA	Sun	加拿大
819	S-1	ISO-8859-1	CA	1	en_CA	Linux	加拿大
1252	S-1	1252	CA	1	-	WIN	加拿大
1275	S-1	1275	CA	1	-	Mac	加拿大
37	S-1	IBM-37	CA	1	-	HOST	加拿大
1140	S-1	IBM-1140	CA	1	-	HOST	加拿大
863	S-1	IBM-863	CA	2	-	OS2	加拿大 (法语)
1381	D-4	IBM-1381	CN	86	-	OS2	中国 (PRC)
1386	D-4	GBK	CN	86	-	OS2	中国 (PRC)
1383	D-4	IBM-eucCN	CN	86	zh_CN	AIX	中国 (PRC)
1386	D-4	GBK	CN	86	Zh_CN.GBK	AIX	中国 (PRC)
1383	D-4	hp15CN	CN	86	zh_CN.hp15CN	HP	中国 (PRC)
1383	D-4	eucCN	CN	86	zh_CN	SCO	中国 (PRC)
1383	D-4	eucCN	CN	86	zh_CN.eucCN	SCO	中国 (PRC)
1383	D-4	gb2312	CN	86	zh	Sun	中国 (PRC)
1381	D-4	IBM-1381	CN	86	-	WIN	中国 (PRC)
1386	D-4	GBK	CN	86	-	WIN	中国 (PRC)
935	D-4	IBM-935	CN	86	-	HOST	中国 (PRC)
1388	D-4	IBM-1388	CN	86	-	HOST	中国 (PRC)
5488**	D-4		CN	86	-		中国 (PRC)
<p>** 代码页 5488 仅能与 LOAD 或 IMPORT 实用程序一起使用来将数据从代码页 5488 移动到 DB2 Unicode 数据库, 或者从 DB2 Unicode 数据库导出至代码页 5488。有关更多信息, 参见版本 7.2 修订包 4 发行版注意事项的“数据移动实用程序指南和参考”部分。</p>							
852	S-2	IBM-852	HR	385	-	OS2	克罗地亚
912	S-2	ISO8859-2	HR	385	hr_HR	AIX	克罗地亚
912	S-2	iso88592	HR	385	hr_HR.iso88592	HP	克罗地亚
912	S-2	ISO8859-2	HR	385	hr_HR.ISO8859-2	SCO	克罗地亚
912	S-2	ISO-8859-2	HR	385	hr_HR	Linux	克罗地亚
1250	S-2	1250	HR	385	-	WIN	克罗地亚
1282	S-2	1282	HR	385	-	Mac	克罗地亚
870	S-2	IBM-870	HR	385	-	HOST	克罗地亚



表 28. 受支持的语言和代码集 (续)

代码 页	组	代码集	Tr.	国家 / 地区 代码	语言环境	OS	国家 / 地区名
852	S-2	IBM-852	CZ	421	-	OS2	捷克共和国
912	S-2	ISO8859-2	CZ	421	cs_CZ	AIX	捷克共和国
912	S-2	iso88592	CZ	421	cs_CZ.iso88592	HP	捷克共和国
912	S-2	ISO8859-2	CZ	421	cs_CZ.ISO8859-2	SCO	捷克共和国
912	S-2	ISO-8859-2	CZ	421	cs_CZ	Linux	捷克共和国
1250	S-2	1250	CZ	421	-	WIN	捷克共和国
1282	S-2	1282	CZ	421	-	Mac	捷克共和国
870	S-2	IBM-870	CZ	421	-	HOST	捷克共和国
850	S-1	IBM-850	DK	45	-	OS2	丹麦
819	S-1	ISO8859-1	DK	45	da_DK	AIX	丹麦
850	S-1	IBM-850	DK	45	Da_DK	AIX	丹麦
819	S-1	iso88591	DK	45	da_DK.iso88591	HP	丹麦
1051	S-1	roman8	DK	45	da_DK.roman8	HP	丹麦
819	S-1	ISO8859-1	DK	45	da	SCO	丹麦
819	S-1	ISO8859-1	DK	45	da_DA	SCO	丹麦
819	S-1	ISO8859-1	DK	45	da_DK	SCO	丹麦
819	S-1	ISO8859-1	DK	45	da	Sun	丹麦
819	S-1	ISO8859-1	DK	45	da	Sun	丹麦
819	S-1	ISO-8859-1	DK	45	da_DK	Linux	丹麦
1252	S-1	1252	DK	45	-	WIN	丹麦
1275	S-1	1275	DK	45	-	Mac	丹麦
277	S-1	IBM-277	DK	45	-	HOST	丹麦
1142	S-1	IBM-1142	DK	45	-	HOST	丹麦
922	S-10	IBM-922	EE	372	-	OS2	爱沙尼亚
922	S-10	IBM-922	EE	372	Et_EE	AIX	爱沙尼亚
1257	S-10	1257	EE	372	-	WIN	爱沙尼亚
1122	S-10	IBM-1122	EE	372	-	HOST	爱沙尼亚
437	S-1	IBM-437	FI	358	-	OS2	芬兰
850	S-1	IBM-850	FI	358	-	OS2	芬兰
819	S-1	ISO8859-1	FI	358	fi_FI	AIX	芬兰
850	S-1	IBM-850	FI	358	Fi_FI	AIX	芬兰
819	S-1	iso88591	FI	358	fi_FI.iso88591	HP	芬兰
819	S-1	ISO8859-1	FI	358	fi	SCO	芬兰
819	S-1	ISO8859-1	FI	358	fi_FI	SCO	芬兰
819	S-1	ISO8859-1	FI	358	sv_FI	SCO	芬兰
819	S-1	ISO8859-1	FI	358	-	Sun	芬兰
819	S-1	ISO-8859-1	FI	358	fi_FI	Linux	芬兰
1051	S-1	roman8	FI	358	-	HP	芬兰
1252	S-1	1252	FI	358	-	WIN	芬兰
1275	S-1	1275	FI	358	-	Mac	芬兰
278	S-1	IBM-278	FI	358	-	HOST	芬兰
1143	S-1	IBM-1143	FI	358	-	HOST	芬兰

表 28. 受支持的语言和代码集 (续)

代码 页	组	代码集	Tr.	国家 / 地区 代码	语言环境	OS	国家 / 地区名
855	S-5	IBM-855	MK	389	-	OS2	FYR 马其顿
915	S-5	ISO8859-5	MK	389	-	OS2	FYR 马其顿
915	S-5	ISO8859-5	MK	389	mk_MK	AIX	FYR 马其顿
915	S-5	iso88595	MK	389	-	HP	FYR 马其顿
1251	S-5	1251	MK	389	-	WIN	FYR 马其顿
1283	S-5	1283	MK	389	-	Mac	FYR 马其顿
1025	S-5	IBM-1025	MK	389	-	HOST	FYR 马其顿
437	S-1	IBM-437	FR	33	-	OS2	法国
850	S-1	IBM-850	FR	33	-	OS2	法国
819	S-1	ISO8859-1	FR	33	fr_FR	AIX	法国
850	S-1	IBM-850	FR	33	Fr_FR	AIX	法国
819	S-1	iso88591	FR	33	fr_FR.iso88591	HP	法国
1051	S-1	roman8	FR	33	fr_FR.roman8	HP	法国
819	S-1	ISO8859-1	FR	33	fr	Sun	法国
819	S-1	ISO8859-1	FR	33	fr	SCO	法国
819	S-1	ISO8859-1	FR	33	fr_FR	SCO	法国
819	S-1	ISO-8859-1	FR	33	fr_FR	Linux	法国
1252	S-1	1252	FR	33	-	WIN	法国
1275	S-1	1275	FR	33	-	Mac	法国
297	S-1	IBM-297	FR	33	-	HOST	法国
1147	S-1	IBM-1147	FR	33	-	HOST	法国
437	S-1	IBM-437	DE	49	-	OS2	德国
850	S-1	IBM-850	DE	49	-	OS2	德国
819	S-1	ISO8859-1	DE	49	de_DE	AIX	德国
850	S-1	IBM-850	DE	49	De_DE	AIX	德国
819	S-1	iso88591	DE	49	de_DE.iso88591	HP	德国
1051	S-1	roman8	DE	49	de_DE.roman8	HP	德国
819	S-1	ISO8859-1	DE	49	de	SCO	德国
819	S-1	ISO8859-1	DE	49	de_DE	SCO	德国
819	S-1	ISO8859-1	DE	49	de	Sun	德国
819	S-1	ISO-8859-1	DE	49	de_DE	Linux	德国
1252	S-1	1252	DE	49	-	WIN	德国
1275	S-1	1275	DE	49	-	Mac	德国
273	S-1	IBM-273	DE	49	-	HOST	德国
1141	S-1	IBM-1141	DE	49	-	HOST	德国
819	S-1	ISO8859-1	DE	49	De_DE.88591	SINIX	德国
819	S-1	ISO8859-1	DE	49	De_DE.6937	SINIX	德国

表 28. 受支持的语言和代码集 (续)

代码 页	组	代码集	Tr.	国家 / 地区 代码	语言环境	OS	国家 / 地区名
813	S-7	ISO8859-7	GR	30	-	OS2	希腊
869	S-7	IBM-869	GR	30	-	OS2	希腊
813	S-7	ISO8859-7	GR	30	e1_GR	AIX	希腊
813	S-7	iso88597	GR	30	e1_GR.iso88597	HP	希腊
813	S-7	ISO8859-7	GR	30	e1_GR.IS08859-7	SCO	希腊
813	S-7	ISO-8859-7	GR	30	gr_GR	Linux	希腊
737	S-7	737	GR	30	-	WIN	希腊
1253	S-7	1253	GR	30	-	WIN	希腊
1280	S-7	1280	GR	30	-	Mac	希腊
423	S-7	IBM-423	GR	30	-	HOST	希腊
875	S-7	IBM-875	GR	30	-	HOST	希腊
852	S-2	IBM-852	HU	36	-	OS2	匈牙利
912	S-2	ISO8859-2	HU	36	hu_HU	AIX	匈牙利
912	S-2	iso88592	HU	36	hu_HU.iso88592	HP	匈牙利
912	S-2	ISO8859-2	HU	36	hu_HU.IS08859-2	SCO	匈牙利
912	S-2	ISO-8859-2	HU	36	hu_HU	Linux	匈牙利
1250	S-2	1250	HU	36	-	WIN	匈牙利
1282	S-2	1282	HU	36	-	Mac	匈牙利
870	S-2	IBM-870	HU	36	-	HOST	匈牙利
850	S-1	IBM-850	IS	354	-	OS2	冰岛
819	S-1	ISO8859-1	IS	354	is_IS	AIX	冰岛
850	S-1	IBM-850	IS	354	Is_IS	AIX	冰岛
819	S-1	iso88591	IS	354	is_IS.iso88591	HP	冰岛
1051	S-1	roman8	IS	354	is_IS.roman8	HP	冰岛
819	S-1	ISO8859-1	IS	354	is	SCO	冰岛
819	S-1	ISO8859-1	IS	354	is_IS	SCO	冰岛
819	S-1	ISO8859-1	IS	354	-	Sun	冰岛
819	S-1	ISO-8859-1	IS	354	is_IS	Linux	冰岛
1252	S-1	1252	IS	354	-	WIN	冰岛
1275	S-1	1275	IS	354	-	Mac	冰岛
871	S-1	IBM-871	IS	354	-	HOST	冰岛
1149	S-1	IBM-1149	IS	354	-	HOST	冰岛

表 28. 受支持的语言和代码集 (续)

代码 页	组	代码集	Tr.	国家/ 地区 代码	语言环境	OS	国家/ 地区名
437	S-1	IBM-437	IE	353	-	OS2	爱尔兰
850	S-1	IBM-850	IE	353	-	OS2	爱尔兰
819	S-1	ISO8859-1	IE	353	en_IE	AIX	爱尔兰
850	S-1	IBM-850	IE	353	En_IE	AIX	爱尔兰
819	S-1	iso88591	IE	353	-	HP	爱尔兰
1051	S-1	roman8	IE	353	-	HP	爱尔兰
819	S-1	ISO8859-1	IE	353	en_IE	Sun	爱尔兰
819	S-1	ISO8859-1	IE	353	en_IE.ISO8859-1	SCO	爱尔兰
819	S-1	ISO-8859-1	IE	353	en_IE	Linux	爱尔兰
1252	S-1	1252	IE	353	-	WIN	爱尔兰
1275	S-1	1275	IE	353	-	Mac	爱尔兰
285	S-1	IBM-285	IE	353	-	HOST	爱尔兰
1146	S-1	IBM-1146	IE	353	-	HOST	爱尔兰
806	S-12	IBM-806	IN	91	hi_IN	-	印度
1137	S-12	IBM-1137	IN	91	-	HOST	印度
862	S-8	IBM-862	IL	972	-	OS2	以色列
916	S-8	ISO8859-8	IL	972	iw_IL	AIX	以色列
856	S-8	IBM-856	IL	972	Iw_IL	AIX	以色列
916	S-8	ISO-8859-8	IL	972	iw_IL	Linux	以色列
1255	S-8	1255	IL	972	-	WIN	以色列
424	S-8	IBM-424	IL	972	-	HOST	以色列
437	S-1	IBM-437	IT	39	-	OS2	意大利
850	S-1	IBM-850	IT	39	-	OS2	意大利
819	S-1	ISO8859-1	IT	39	it_IT	AIX	意大利
850	S-1	IBM-850	IT	39	It_IT	AIX	意大利
819	S-1	iso88591	IT	39	it_IT.iso88591	HP	意大利
1051	S-1	roman8	IT	39	it_IT.roman8	HP	意大利
819	S-1	ISO8859-1	IT	39	it	SCO	意大利
819	S-1	ISO8859-1	IT	39	it_IT	SCO	意大利
819	S-1	ISO8859-1	IT	39	it	Sun	意大利
819	S-1	ISO-8859-1	IT	39	it_IT	Linux	意大利
1252	S-1	1252	IT	39	-	WIN	意大利
1275	S-1	1275	IT	39	-	Mac	意大利
280	S-1	IBM-280	IT	39	-	HOST	意大利
1144	S-1	IBM-1144	IT	39	-	HOST	意大利

表 28. 受支持的语言和代码集 (续)

代码 页	组	代码集	Tr.	国家 / 地区 代码	语言环境	OS	国家 / 地区名
932	D-1	IBM-932	JP	81	-	OS2	日本
942	D-1	IBM-942	JP	81	-	OS2	日本
943	D-1	IBM-943	JP	81	-	OS2	日本
954	D-1	IBM-eucJP	JP	81	ja_JP	AIX	日本
943*	D-1	IBM-943	JP	81	Ja_JP	AIX	日本
954	D-1	eucJP	JP	81	ja_JP.eucJP	HP	日本
5039	D-1	SJIS	JP	81	ja_JP.SJIS	HP	日本
954	D-1	eucJP	JP	81	ja	SCO	日本
954	D-1	eucJP	JP	81	ja_JP	SCO	日本
954	D-1	eucJP	JP	81	ja_JP.EUC	SCO	日本
954	D-1	eucJP	JP	81	ja_JP.eucJP	SCO	日本
954	D-1	eucJP	JP	81	ja	Sun	日本
943	D-1	IBM-943	JP	81	ja_JP.PCK	Sun	日本
954	D-1	EUC-JP	JP	81	ja_JP	Linux	日本
943	D-1	IBM-943	JP	81	-	WIN	日本
930	D-1	IBM-930	JP	81	-	HOST	日本
939	D-1	IBM-939	JP	81	-	HOST	日本
5026	D-1	IBM-5026	JP	81	-	HOST	日本
5035	D-1	IBM-5035	JP	81	-	HOST	日本
1390	D-1		JP	81	-	HOST	日本
1399	D-1		JP	81	-	HOST	日本
1394**	D-1		JP	81	-		日本

\* 在 AIX 4.3 或更高版本中代码页是 943。如果您正在使用 AIX 4.2 或更低的版本，则代码页是 932。

\*\* 代码页 1394 仅可以与 LOAD 或 IMPORT 实用程序一起使用来将数据从代码页 1394 移动到 DB2 Unicode 数据库，或者从 DB2 Unicode 数据库导出至代码页 1394。有关更多信息，参见版本 7.2 修订包 4 发行版注意事项的“数据移动实用程序指南和参考”部分。

949	D-3	IBM-949	KR	82	-	OS2	南韩
970	D-3	IBM-eucKR	KR	82	ko_KR	AIX	南韩
970	D-3	eucKR	KR	82	ko_KR.eucKR	HP	南韩
970	D-3	eucKR	KR	82	ko_KR.eucKR	SGI	南韩
970	D-3	5601	KR	82	ko	Sun	南韩
1363	D-3	1363	KR	82	-	WIN	南韩
933	D-3	IBM-933	KR	82	-	HOST	南韩
1364	D-3	IBM-1364	KR	82	-	HOST	南韩

表 28. 受支持的语言和代码集 (续)

代码 页	组	代码集	Tr.	国家 / 地区 代码	语言环境	OS	国家 / 地区名
437	S-1	IBM-437	Lat	3	-	OS2	拉丁美洲
850	S-1	IBM-850	Lat	3	-	OS2	拉丁美洲
819	S-1	ISO8859-1	Lat	3	-	AIX	拉丁美洲
850	S-1	IBM-850	Lat	3	-	AIX	拉丁美洲
819	S-1	iso88591	Lat	3	-	HP	拉丁美洲
819	S-1	ISO8859-1	Lat	3	-	Sun	拉丁美洲
819	S-1	ISO-8859-1	Lat	3	-	Linux	拉丁美洲
1051	S-1	roman8	Lat	3	-	HP	拉丁美洲
1252	S-1	1252	Lat	3	-	WIN	拉丁美洲
1275	S-1	1275	Lat	3	-	Mac	拉丁美洲
284	S-1	IBM-284	Lat	3	-	HOST	拉丁美洲
1145	S-1	IBM-1145	Lat	3	-	HOST	拉丁美洲
921	S-10	IBM-921	LV	371	-	OS2	拉脱维亚
921	S-10	IBM-921	LV	371	Lv_LV	AIX	拉脱维亚
1257	S-10	1257	LV	371	-	WIN	拉脱维亚
1112	S-10	IBM-1112	LV	371	-	HOST	拉脱维亚
921	S-10	IBM-921	LT	370	-	OS2	立陶宛
921	S-10	IBM-921	LT	370	Lt_LT	AIX	立陶宛
1257	S-10	1257	LT	370	-	WIN	立陶宛
1112	S-10	IBM-1112	LT	370	-	HOST	立陶宛
437	S-1	IBM-437	NL	31	-	OS2	荷兰
850	S-1	IBM-850	NL	31	-	OS2	荷兰
819	S-1	ISO8859-1	NL	31	n1_NL	AIX	荷兰
850	S-1	IBM-850	NL	31	N1_NL	AIX	荷兰
819	S-1	iso88591	NL	31	n1_NL.iso88591	HP	荷兰
1051	S-1	roman8	NL	31	n1_NL.roman8	HP	荷兰
819	S-1	ISO8859-1	NL	31	n1	SCO	荷兰
819	S-1	ISO8859-1	NL	31	n1_NL	SCO	荷兰
819	S-1	ISO8859-1	NL	31	n1	Sun	荷兰
819	S-1	ISO-8859-1	NL	31	n1_NL	Linux	荷兰
1252	S-1	1252	NL	31	-	WIN	荷兰
1275	S-1	1275	NL	31	-	Mac	荷兰
37	S-1	IBM-37	NL	31	-	HOST	荷兰
1140	S-1	IBM-1140	NL	31	-	HOST	荷兰
850	S-1	IBM-850	NZ	64	-	OS2	新西兰
850	S-1	IBM-850	NZ	64	En_NZ	AIX	新西兰
819	S-1	ISO8859-1	NZ	64	en_NZ	AIX	新西兰
819	S-1	ISO8859-1	NZ	64	-	HP	新西兰
819	S-1	ISO8859-1	NZ	64	en_NZ	SCO	新西兰
819	S-1	ISO8859-1	NZ	64	en_NZ	Sun	新西兰
1252	S-1	1252	NZ	64	-	WIN	新西兰
37	S-1	IBM-37	NZ	64	-	HOST	新西兰
1140	S-1	IBM-1140	NZ	64	-	HOST	新西兰

表 28. 受支持的语言和代码集 (续)

代码 页	组	代码集	Tr.	国家 / 地区 代码	语言环境	OS	国家 / 地区名
850	S-1	IBM-850	NO	47	-	OS2	挪威
819	S-1	ISO8859-1	NO	47	no_NO	AIX	挪威
850	S-1	IBM-850	NO	47	No_NO	AIX	挪威
819	S-1	iso88591	NO	47	no_NO.iso88591	HP	挪威
1051	S-1	roman8	NO	47	no_NO.roman8	HP	挪威
819	S-1	ISO8859-1	NO	47	no	SCO	挪威
819	S-1	ISO8859-1	NO	47	no_NO	SCO	挪威
819	S-1	ISO8859-1	NO	47	no	Sun	挪威
819	S-1	ISO-8859-1	NO	47	no_NO	Linux	挪威
1252	S-1	1252	NO	47	-	WIN	挪威
1275	S-1	1275	NO	47	-	Mac	挪威
277	S-1	IBM-277	NO	47	-	HOST	挪威
1142	S-1	IBM-1142	NO	47	-	HOST	挪威
852	S-2	IBM-852	PL	48	-	OS2	波兰
912	S-2	ISO8859-2	PL	48	pl_PL	AIX	波兰
912	S-2	iso88592	PL	48	pl_PL.iso88592	HP	波兰
912	S-2	ISO8859-2	PL	48	pl_PL.ISO8859-2	SCO	波兰
912	S-2	ISO-8859-2	PL	48	pl_PL	Linux	波兰
1250	S-2	1250	PL	48	-	WIN	波兰
1282	S-2	1282	PL	48	-	Mac	波兰
870	S-2	IBM-870	PL	48	-	HOST	波兰
860	S-1	IBM-860	PT	351	-	OS2	葡萄牙
850	S-1	IBM-850	PT	351	-	OS2	葡萄牙
819	S-1	ISO8859-1	PT	351	pt_PT	AIX	葡萄牙
850	S-1	IBM-850	PT	351	Pt_PT	AIX	葡萄牙
819	S-1	iso88591	PT	351	pt_PT.iso88591	HP	葡萄牙
1051	S-1	roman8	PT	351	pt_PT.roman8	HP	葡萄牙
819	S-1	ISO8859-1	PT	351	pt	SCO	葡萄牙
819	S-1	ISO8859-1	PT	351	pt_PT	SCO	葡萄牙
819	S-1	ISO8859-1	PT	351	pt	Sun	葡萄牙
819	S-1	ISO-8859-1	PT	351	pt_PT	Linux	葡萄牙
1252	S-1	1252	PT	351	-	WIN	葡萄牙
1275	S-1	1275	PT	351	-	Mac	葡萄牙
37	S-1	IBM-37	PT	351	-	HOST	葡萄牙
1140	S-1	IBM-1140	PT	351	-	HOST	葡萄牙
852	S-2	IBM-852	RO	40	-	OS2	罗马尼亚
912	S-2	ISO8859-2	RO	40	ro_RO	AIX	罗马尼亚
912	S-2	iso88592	RO	40	ro_RO.iso88592	HP	罗马尼亚
912	S-2	ISO8859-2	RO	40	ro_RO.ISO8859-2	SCO	罗马尼亚
912	S-2	ISO-8859-2	RO	40	ro_RO	Linux	罗马尼亚
1250	S-2	1250	RO	40	-	WIN	罗马尼亚
1282	S-2	1282	RO	40	-	Mac	罗马尼亚
870	S-2	IBM-870	RO	40	-	HOST	罗马尼亚

表 28. 受支持的语言和代码集 (续)

代码 页	组	代码集	Tr.	国家 / 地区 代码	语言环境	OS	国家 / 地区名
866	S-5	IBM-866	RU	7	-	OS2	俄国
915	S-5	ISO8859-5	RU	7	-	OS2	俄国
915	S-5	ISO8859-5	RU	7	ru_RU	AIX	俄国
915	S-5	iso88595	RU	7	ru_RU.iso88595	HP	俄国
915	S-5	ISO8859-5	RU	7	ru_RU.ISO8859-5	SCO	俄国
915	S-5	ISO-8859-5	RU	7	ru_RU	Linux	俄国
1251	S-5	1251	RU	7	-	WIN	俄国
1283	S-5	1283	RU	7	-	Mac	俄国
1025	S-5	IBM-1025	RU	7	-	HOST	俄国
855	S-5	IBM-855	SP	381	-	OS2	塞尔维亚 / 蒙的内哥罗
915	S-5	ISO8859-5	SP	381	-	OS2	塞尔维亚 / 蒙的内哥罗
915	S-5	ISO8859-5	SP	381	sr_SP	AIX	塞尔维亚 / 蒙的内哥罗
915	S-5	iso88595	SP	381	-	HP	塞尔维亚 / 蒙的内哥罗
1251	S-5	1251	SP	381	-	WIN	塞尔维亚 / 蒙的内哥罗
1283	S-5	1283	SP	381	-	Mac	塞尔维亚 / 蒙的内哥罗
1025	S-5	IBM-1025	SP	381	-	HOST	塞尔维亚 / 蒙的内哥罗
852	S-2	IBM-852	SK	422	-	OS2	斯洛伐克
912	S-2	ISO8859-2	SK	422	sk_SK	AIX	斯洛伐克
912	S-2	iso88592	SK	422	sk_SK.iso88592	HP	斯洛伐克
912	S-2	ISO8859-2	SK	422	sk_SK.ISO8859-2	SCO	斯洛伐克
1250	S-2	1250	SK	422	-	WIN	斯洛伐克
1282	S-2	1282	SK	422	-	Mac	斯洛伐克
870	S-2	IBM-870	SK	422	-	HOST	斯洛伐克
852	S-2	IBM-852	SI	386	-	OS2	斯洛文尼亚
912	S-2	ISO8859-2	SI	386	sl_SI	AIX	斯洛文尼亚
912	S-2	iso88592	SI	386	sl_SI.iso88592	HP	斯洛文尼亚
912	S-2	ISO8859-2	SI	386	sl_SI.ISO8859-2	SCO	斯洛文尼亚
912	S-2	ISO-8859-2	SI	386	sl_SI	Linux	斯洛文尼亚
1250	S-2	1250	SI	386	-	WIN	斯洛文尼亚
1282	S-2	1282	SI	386	-	Mac	斯洛文尼亚
870	S-2	IBM-870	SI	386	-	HOST	斯洛文尼亚
437	S-1	IBM-437	ZA	27	-	OS2	南非
850	S-1	IBM-850	ZA	27	-	OS2	南非
819	S-1	ISO8859-1	ZA	27	en_ZA	AIX	南非
850	S-1	IBM-850	ZA	27	En_ZA	AIX	南非
819	S-1	iso88591	ZA	27	-	HP	南非
1051	S-1	roman8	ZA	27	-	HP	南非
819	S-1	ISO8859-1	ZA	27	-	Sun	南非
819	S-1	ISO8859-1	ZA	27	en_ZA.ISO8859-1	SCO	南非
1252	S-1	1252	ZA	27	-	WIN	南非
1275	S-1	1275	ZA	27	-	Mac	南非
285	S-1	IBM-285	ZA	27	-	HOST	南非
1146	S-1	IBM-1146	ZA	27	-	HOST	南非



表 28. 受支持的语言和代码集 (续)

代码 页	组	代码集	Tr.	国家 / 地区 代码	语言环境	OS	国家 / 地区名
437	S-1	IBM-437	ES	34	-	OS2	西班牙
850	S-1	IBM-850	ES	34	-	OS2	西班牙
819	S-1	ISO8859-1	ES	34	es_ES	AIX	西班牙
850	S-1	IBM-850	ES	34	Es_ES	AIX	西班牙
819	S-1	iso88591	ES	34	es_ES.iso88591	HP	西班牙
1051	S-1	roman8	ES	34	es_ES.roman8	HP	西班牙
819	S-1	ISO8859-1	ES	34	es	Sun	西班牙
819	S-1	ISO8859-1	ES	34	es	SCO	西班牙
819	S-1	ISO8859-1	ES	34	es_ES	SCO	西班牙
819	S-1	ISO-8859-1	ES	34	es_ES	Linux	西班牙
1252	S-1	1252	ES	34	-	WIN	西班牙
1275	S-1	1275	ES	34	-	Mac	西班牙
284	S-1	IBM-284	ES	34	-	HOST	西班牙
1145	S-1	IBM-1145	ES	34	-	HOST	西班牙
437	S-1	IBM-437	SE	46	-	OS2	瑞典
850	S-1	IBM-850	SE	46	-	OS2	瑞典
819	S-1	ISO8859-1	SE	46	sv_SE	AIX	瑞典
850	S-1	IBM-850	SE	46	Sv_SE	AIX	瑞典
819	S-1	iso88591	SE	46	sv_SE.iso88591	HP	瑞典
1051	S-1	roman8	SE	46	sv_SE.roman8	HP	瑞典
819	S-1	ISO8859-1	SE	46	sv	SCO	瑞典
819	S-1	ISO8859-1	SE	46	sv_SE	SCO	瑞典
819	S-1	ISO8859-1	SE	46	sv	Sun	瑞典
819	S-1	ISO-8859-1	SE	46	sv_SE	Linux	瑞典
1252	S-1	1252	SE	46	-	WIN	瑞典
1275	S-1	1275	SE	46	-	Mac	瑞典
278	S-1	IBM-278	SE	46	-	HOST	瑞典
1143	S-1	IBM-1143	SE	46	-	HOST	瑞典
437	S-1	IBM-437	CH	41	-	OS2	瑞士
850	S-1	IBM-850	CH	41	-	OS2	瑞士
819	S-1	ISO8859-1	CH	41	de_CH	AIX	瑞士
850	S-1	IBM-850	CH	41	De_CH	AIX	瑞士
819	S-1	iso88591	CH	41	-	HP	瑞士
1051	S-1	roman8	CH	41	-	HP	瑞士
819	S-1	ISO8859-1	CH	41	de_CH	SCO	瑞士
819	S-1	ISO8859-1	CH	41	fr_CH	SCO	瑞士
819	S-1	ISO8859-1	CH	41	it_CH	SCO	瑞士
819	S-1	ISO8859-1	CH	41	de_CH	Sun	瑞士
819	S-1	ISO-8859-1	CH	41	de_CH	Linux	瑞士
1252	S-1	1252	CH	41	-	WIN	瑞士
1275	S-1	1275	CH	41	-	Mac	瑞士
500	S-1	IBM-500	CH	41	-	HOST	瑞士
1148	S-1	IBM-1148	CH	41	-	HOST	瑞士

表 28. 受支持的语言和代码集 (续)

代码 页	组	代码集	Tr.	国家 / 地区 代码	语言环境	OS	国家 / 地区名
938	D-2	IBM-938	TW	88	-	OS2	台湾
948	D-2	IBM-948	TW	88	-	OS2	台湾
950	D-2	big5	TW	88	-	OS2	台湾
950	D-2	big5	TW	88	Zh_TW	AIX	台湾
964	D-2	IBM-eucTW	TW	88	zh_TW	AIX	台湾
950	D-2	big5	TW	88	zh_TW.big5	HP	台湾
964	D-2	eucTW	TW	88	zh_TW.eucTW	HP	台湾
950	D-2	big5	TW	88	big5	Sun	台湾
964	D-2	cns11643	TW	88	zh_TW	Sun	台湾
950	D-2	big5	TW	88	-	WIN	台湾
937	D-2	IBM-937	TW	88	-	HOST	台湾
874	S-20	TIS620-1	TH	66	-	OS2	泰国
874	S-20	TIS620-1	TH	66	Th_TH	AIX	泰国
874	S-20	tis620	TH	66	th_TH.tis620	HP	泰国
874	S-20	TIS620-1	TH	66	-	WIN	泰国
838	S-20	IBM-838	TH	66	-	HOST	泰国
857	S-9	IBM-857	TR	90	-	OS2	土耳其
920	S-9	ISO8859-9	TR	90	tr_TR	AIX	土耳其
920	S-9	iso88599	TR	90	tr_TR.iso88599	HP	土耳其
920	S-9	ISO8859-9	TR	90	tr_TR.ISO8859-9	SCO	土耳其
920	S-9	ISO-8859-9	TR	90	tr_TR	Linux	土耳其
1254	S-9	1254	TR	90	-	WIN	土耳其
1281	S-9	1281	TR	90	-	Mac	土耳其
1026	S-9	IBM-1026	TR	90	-	HOST	土耳其
437	S-1	IBM-437	GB	44	-	OS2	英国
850	S-1	IBM-850	GB	44	-	OS2	英国
819	S-1	ISO8859-1	GB	44	en_GB	AIX	英国
850	S-1	IBM-850	GB	44	En_GB	AIX	英国
819	S-1	iso88591	GB	44	en_GB.iso88591	HP	英国
1051	S-1	roman8	GB	44	en_GB.roman8	HP	英国
819	S-1	ISO8859-1	GB	44	en_UK	Sun	英国
819	S-1	ISO8859-1	GB	44	en_GB	SCO	英国
819	S-1	ISO8859-1	GB	44	en	SCO	英国
819	S-1	ISO-8859-1	GB	44	en_GB	Linux	英国
1252	S-1	1252	GB	44	-	WIN	英国
1275	S-1	1275	GB	44	-	Mac	英国
285	S-1	IBM-285	GB	44	-	HOST	英国
1146	S-1	IBM-1146	GB	44	-	HOST	英国
819	S-1	88591	GB	44	En_GB.88591	SINIX	英国
819	S-1	ISO8859-1	GB	44	En_GB.6937	SINIX	英国

表 28. 受支持的语言和代码集 (续)

代码 页	组	代码集	Tr.	国家 / 地区 代码	语言环境	OS	国家 / 地区名
1125	S-5	IBM-1125	UA	380	-	OS2	乌克兰
1124	S-5	IBM-1124	UA	380	uk_UA	AIX	乌克兰
1251	S-5	1251	UA	380	-	WIN	乌克兰
1123	S-5	IBM-1123	UA	380	-	HOST	乌克兰
437	S-1	IBM-437	US	1	-	OS2	美国
850	S-1	IBM-850	US	1	-	OS2	美国
819	S-1	ISO8859-1	US	1	en_US	AIX	美国
850	S-1	IBM-850	US	1	En_US	AIX	美国
819	S-1	iso88591	US	1	en_US.iso88591	HP	美国
1051	S-1	roman8	US	1	en_US.roman8	HP	美国
819	S-1	ISO8859-1	US	1	en_US	Sun	美国
819	S-1	ISO8859-1	US	1	en_US	SGI	美国
819	S-1	ISO8859-1	US	1	en_US	SCO	美国
819	S-1	ISO-8859-1	US	1	en_US	Linux	美国
1252	S-1	1252	US	1	-	WIN	美国
1275	S-1	1275	US	1	-	Mac	美国
37	S-1	IBM-37	US	1	-	HOST	美国
1140	S-1	IBM-1140	US	1	-	HOST	美国
1163	S-11	IBM-1163	VN	84	-	OS2	越南
1163	S-11	IBM-1163	VN	84	vi_VN	AIX	越南
1258	S-11	1258	VN	84	-	WIN	越南
1164	S-11	IBM-1164	VN	84	-	HOST	越南

以下映射至阿拉伯国家 (AA):

```

-----
/* 阿拉伯语 (沙特阿拉伯) */
/* 阿拉伯语 (伊拉克) */
/* 阿拉伯语 (埃及) */
/* 阿拉伯语 (利比亚) */
/* 阿拉伯语 (阿尔及利亚) */
/* 阿拉伯语 (摩洛哥) */
/* 阿拉伯语 (突尼斯) */
/* 阿拉伯语 (阿曼) */
/* 阿拉伯语 (也门) */
/* 阿拉伯语 (叙利亚) */
/* 阿拉伯语 (约旦) */
/* 阿拉伯语 (黎巴嫩) */
/* 阿拉伯语 (科威特) */
/* 阿拉伯语 (阿拉伯联合酋长国) */
/* 阿拉伯语 (巴林) */
/* 阿拉伯语 (卡塔尔) */

```

表 28. 受支持的语言和代码集 (续)

代码页	组	代码集	Tr.	国家 / 地区 语言环境	OS	国家 / 地区名
以下映射至英语 (US):						
-----						
/* 英语 (牙买加) */						
/* 英语 (加勒比地区) */						
-----						
以下映射至拉丁美洲 (Lat):						
-----						
/* 西班牙语 (墨西哥) */						
/* 西班牙语 (危地马拉) */						
/* 西班牙语 (哥斯达黎加) */						
/* 西班牙语 (巴拿马) */						
/* 西班牙语 (多米尼加共和国) */						
/* 西班牙语 (委内瑞拉) */						
/* 西班牙语 (哥伦比亚) */						
/* 西班牙语 (秘鲁) */						
/* 西班牙语 (阿根廷) */						
/* 西班牙语 (厄瓜多尔) */						
/* 西班牙语 (智利) */						
/* 西班牙语 (乌拉圭) */						
/* 西班牙语 (巴拉圭) */						
/* 西班牙语 (玻利维亚) */						

注:

1. Solaris 代码页 950 不支持 IBM 950 中的下列字符:

代码范围	描述	Sun Big-5	IBM Big-5
C6A1-C8FE	符号	保留区	符号
F9D6-F9FE	ETen 扩充	保留区	ETen 扩充
F286-F9A0	IBM 选择的字符	保留区	IBM 选择的

2. DB2 UDB 的此版本提供欧元符号的支持。根据最新的 Microsoft 定义修改 Microsoft Windows ANSI 代码页以在位置 0x80 包含欧元符号。此位置先前未定义。另外,更改代码页 850 的定义以用欧元符号来替换字符去点的 i (可在位置 0xD5 找到)。DB2 UDB 使用这些代码页的新定义为缺省值以提供欧元符号支持。此实现是给当前需要欧元符号支持的 DB2 UDB 客户的恰当的缺省值,且不应影响其他客户。但是,若您要继续使用这些代码页的先前的定义,则可在安装结束后拷贝下列文件:

- 12520850.cnv
- 08501252.cnv

- IBM00850.ucs
- IBM01252.ucs

从目录

sql1lib/conv/alt/

到目录

sql1lib/conv/

在安装完成之后。在拷贝非欧元版本覆盖之前，您可能希望备份当前的 IBM01252.usc 和 IBM00850.ucs 文件。拷贝文件之后，将没有 DB2 UDB 支持的欧元符号。

---

## DB2 管理服务器的语言环境设置

确保“DB2 管理服务器”实例的语言环境与 DB2 实例的语言环境兼容。否则，DB2 实例不能与“DB2 管理服务器”通信。

如果在“DB2 管理服务器”的用户简要表中未设置 LANG 环境变量，则将使用缺省的系统语言环境来启动“DB2 管理服务器”。如果未定义缺省的系统语言环境，则将使用代码页 819 来启动“DB2 管理服务器”。如果 DB2 实例使用 DBCS 语言环境之一，而使用代码页 819 启动“DB2 管理服务器”，则该实例将不能与“DB2 管理服务器”通信。“DB2 管理服务器”的语言环境与 DB2 实例的语言环境必须兼容。

例如，在简体中文 Linux 系统中，应该在“DB2 管理服务器”的用户简要表中设置 LANG=zh\_CN。

---

## 基于 UNIX 的平台所支持的翻译语言环境

在基于 UNIX 的平台上，可以几种不同的语言安装 DB2 产品消息和库。对于给出的平台（如下表所示），DB2 安装实用程序将消息目录文件集放在最常用的语言环境目录中。第216页的表29提供关于 AIX、HP-UX 和 Solaris 的信息。第217页的表30提供关于 Linux、Linux/390、SGI 和 Dynix 的信息。

如果您的系统使用的代码页与第216页的表29和第217页的表30中提供的相同，但使用的语言环境名称不同，则您仍可以通过创建至适当的消息目录的链接来查看翻译后的消息。

例如，如果您的 AIX 机器缺省语言环境是 `ja_JP.IBM-eucJP`，并且 `ja_JP.IBM-eucJP` 的代码页是 954，则您可以通过发出下列命令创建从 `/usr/lpp/db2_07_01/msg/ja_JP.IBM-eucJP` 至 `/usr/lpp/db2_07_01/msg/ja_JP` 的链接：

```
In -s /usr/lpp/db2_07_01/msg/ja_JP /usr/lpp/db2_07_01/msg/ja_JP.IBM-eucJP
```

在执行此命令之后，所有 DB2 消息都将用日语显示。

表 29. AIX、HP-UX、Solaris 的目录语言环境

	AIX		HP-UX		Solaris	
语言	语言环境	代码页	语言环境	代码页	语言环境	代码页
法语	fr_FR	819	fr_FR.iso88591	819	fr	819
	Fr_FR	850	fr_FR.roman8	1051		
德语	de_DE	819	de_DE.iso88591	819	de	819
	De_DE	850	de_DE.roman8	1051		
意大利语	it_IT	819	it_IT.iso88591	819	it	819
	It_IT	850	it_IT.roman8	1051		
西班牙语	es_ES	819	es_ES.iso88591	819	es	819
	Es_ES	850	es_ES.roman8	1051		
巴西葡萄牙语	pt_BR	819			pt_BR	819
日语	ja_JP	954	ja_JP.eucJP	954	ja	954
	Ja_JP	932				
韩国语	ko_KR	970	ko_KR.eucKR	970	ko	970
简体中文	zh_CN	1383	zh_CN.hp15CN	1383	zh	1383
	Zh_CN.GBK	1386				
繁体中文	zh_TW	964	zh_TW.eucTW	964	zh_TW	964
	Zh_TW	950	zh_TW.big5	950	zh_TW.BIG5	950
丹麦语	da_DK	819	da_DK.iso88591	819	da	819
	Da_DK	850	da_DK.roman8	1051		
芬兰语	fi_FI	819	fi_FI.iso88591	819	fi	819
	Fi_FI	850	fi_FI.roman8	1051		

表 29. AIX、HP-UX、Solaris 的目录语言环境 (续)

语言	AIX		HP-UX		Solaris	
	语言环境	代码页	语言环境	代码页	语言环境	代码页
挪威语	no_NO	819	no_NO.iso88591	819	no	819
	No_NO	850	no_NO.roman8	1051		
瑞典语	sv_SE	819	sv_SE.iso88591	819	sv	819
	Sv_SE	850	sv_SE.roman8	1051		
捷克语	cs_CZ	912				
匈牙利语	hu_HU	912				
波兰语	pl_PL	912				
荷兰语	nl_NL	819				
	Nl_NL	850				
土耳其语	tr_TR	920				
俄语	ru_RU	915				
保加利亚语	bg_BG	915	bg_BG.iso88595	915		
斯洛文尼亚语	sl_SI	912	sl_SI.iso88592	912	sl_SI	912

表 30. Linux、Linux/390、SGI、Dynix 的目录语言环境

语言	Linux		Linux/390		SGI		Dynix	
	语言环境	代码页	语言环境	代码页	语言环境	代码页	语言环境	代码页
法语	fr	819	fr	819			fr	819
德语	de	819	de	819			de	819
意大利语							es	819
西班牙语								
巴西葡萄牙语								
日语	ja_JP.ujis	954	ja_JP.ujis	954			ja_JP.EUC	954
韩国语	ko	970	ko	970	ko_KO.euc	970		
简体中文	zh_zh_CN.GBK	1386	zh_zh_CN.GBK	1386				
繁体中文	zh_TW.Big5	950	zh_TW.Big5	950				
丹麦语								
芬兰语								

表 30. Linux、Linux/390、SGI、Dynix 的目录语言环境 (续)

语言	Linux		Linux/390		SGI		Dynix	
	语言环境	代码页	语言环境	代码页	语言环境	代码页	语言环境	代码页
挪威语								
瑞典语								
捷克语								
匈牙利语								
波兰语								
荷兰语							nl	819
土耳其语								
俄语								
保加利亚语								
斯洛文尼亚语								

### 控制中心和文档文件集

将“控制中心”、“控制中心帮助”和文档文件集放置在目标工作站的下列目录中:

- DB2 AIX 版:
  - /usr/lpp/db2\_07\_01/cc/%L
  - /usr/lpp/db2\_07\_01/java/%L
  - /usr/lpp/db2\_07\_01/doc/%L
  - /usr/lpp/db2\_07\_01/qp/%L
  - /usr/lpp/db2\_07\_01/spb/%L
- DB2 HP-UX 版:
  - /opt/IBMDB2/V7.1/cc/%L
  - /opt/IBMDB2/V7.1/java/%L
  - /opt/IBMDB2/V7.1/doc/%L
- DB2 Linux 版:
  - /usr/IBMDB2/V7.1/cc/%L
  - /usr/IBMDB2/V7.1/java/%L
  - /usr/IBMDB2/V7.1/doc/%L
- DB2 Solaris 版:



- /opt/IBMd2/V7.1/cc/%L
- /usr/IBMd2/V7.1/java/%L
- /opt/IBMd2/V7.1/doc/%L

“控制中心”文件集在 Unicode 代码页中。文档和“控制中心”帮助文件集位于浏览器识别的代码集中。如果您的系统使用的语言环境名称与所提供的不同，您仍可以通过创建至适当的语言目录的链接来运行“控制中心”的翻译版本并查看“帮助”的翻译版本。

例如，如果您的 AIX 机器缺省语言环境是 ja\_JP.IBM-eucJP，可以通过发出下列命令创建从 /usr/lpp/db2\_07\_01/cc/ja\_JP.IBM-eucJP 至 /usr/lpp/db2\_07\_01/cc/ja\_JP 和从 /usr/lpp/db2\_07\_01/doc/ja\_JP.IBM-eucJP 至 /usr/lpp/db2\_07\_01/doc/ja\_JP 的链接：

- **ln -s /usr/lpp/db2\_07\_01/cc/ja\_JP /usr/lpp/db2\_07\_01/cc/ja\_JP.IBM-eucJP**
- **ln -s /usr/lpp/db2\_07\_01/doc/ja\_JP /usr/lpp/db2\_07\_01/doc/ja\_JP.IBM-eucJP**

在执行这些命令之后，“控制中心”和“帮助”以日语显示。

**注：**本来不支持“Web 控制中心”在 Linux/390 或 NUMA-Q 上运行。可使用它从客户机工作站来管理这些平台上的数据库。

---

## 派生代码页值

当进行数据库连接时，应用程序代码页由活动的环境派生而来。若设置了 DB2CODEPAGE 注册表变量，则将其值用作应用程序代码页。通常，不需要设置 DB2CODEPAGE 注册表变量，因为 DB2 会自动从操作系统派生该代码页信息。将 DB2CODEPAGE 注册表变量设置为不正确的值可能导致不可预测的结果。

在应用程序编码为使用 CLI 接口的情况下，CLI 代码层将使用某些情况下的语言环境设置，即使用户已设置 DB2CODEPAGE 注册表变量。

数据库代码页则从创建数据库时指定的值（显式或缺省）派生而来。例如，下面定义如何在不同的操作环境中确定活动环境：

- |                  |  |
|------------------|--|
| <b>UNIX</b>      | 在基于 UNIX 的操作系统上，活动环境是根据本国语言环境设置确定的，它包含有关语言、区域和代码集的信息。                      |
| <b>OS/2</b>      | 在 OS/2 上，在 CONFIG.SYS 文件中指定主代码页和次代码页。可以使用 <b>chcp</b> 命令来显示和动态更改给定会话中的代码页。 |
| <b>Macintosh</b> | 对于 Macintosh 操作系统，若未设置   |

DB2CODEPAGE 注册表变量，则 Macintosh 代码页由已安装脚本的“地区”版本代码派生而来。

## Windows

对于所有 Windows 32 位操作系统，若未设置 DB2CODEPAGE 注册表变量，则代码页由“注册表”中的 ANSI 代码页设置派生而得。

有关代码页值的环境映射的完整列表，参见第200页的表28。

---

## 字符集

一般而言，数据库管理器不限制应用程序可以使用的字符集。有关受 DB2 支持的多字节字符集 (MBCS) 的详细说明，参考 *Application Development Guide*。

### 标识符的字符集

可以在数据库名中使用的基本字符集包括单字节大写和小写拉丁字母 (A...Z, a...z)、阿拉伯数字 (0...9) 和下划线字符 (\_)。此列表增加了三个特殊字符 (#、@ 和 \$) 以提供与主机数据库产品的兼容性。在 NLS 环境中使用特殊字符 #、@ 和 \$ 应小心，因为它们不包括在 NLS 主机 (EBCDIC) 不变量字符集中。还可以使用来自扩展字符集的字符，这取决于所使用的代码页。如果是在一个多代码页环境中使用数据库，则您必须确保所有代码页都支持您计划使用的扩展字符集中的任何元素。

当命名数据库对象（如表和视图）、程序标号、主变量、游标时，也可使用扩充字符集（例如，带有相异标记的字母）中的元素。到底哪些字符可用取决于所用的代码页。若是在一个多代码页环境中使用该数据库，您必须确保所有代码页都支持您计划使用的扩充字符集中的任何元素。有关带有不属于扩充字符集的字符，但可以在 SQL 语句中使用的定界标识符的信息，参考 *SQL Reference*。

### DBCS 标识符的扩充字符集定义

在 DBCS 环境中，扩充字符集包含基本字符集中的所有字符以及下列各项：

- 除双字节空间外，每个 DBCS 代码页中的所有双字节字符都是有效的字母。
- 双字节空间是特殊字符。
- 在每个混合的代码页中可用的单字节字符被分配给各种类别，如下所示：

类别	在每个混合代码页内有效的代码点
数字	x30-39
字母	x23-24、x40-5A、x61-7A、xA6-DF (A6-DF 仅用于代码页 932 和 942)
特殊字符	所有其他有效的单字节字符代码点

## 编码 SQL 语句

SQL 语句的编码与语言无关。SQL 关键字可以以大写、小写或混合大小写输入。SQL 语句中数据库对象和主变量的名称以及程序标号不能包含第220页的『DBCS 标识符的扩充字符集定义』中所述的扩展字符集之外的字符。

## 双向 CCSID 支持

要正确处理不同平台上的双向数据，下列 BiDi 属性是必需的：

- 文本类型 (LOGICAL 或 VISUAL)
- 整形 (SHAPED 或 UNSHAPED)
- 方向 (RIGHT-TO-LEFT 或 LEFT-TO-RIGHT)
- 数字形状 (ARABIC 或 HINDI)
- 对称交换 (YES 或 NO)

因为不同平台上的缺省值不一样，因此在将 DB2 数据从一个平台移到另一个平台时会出现问题。例如，Windows 操作系统使用 LOGICAL UNSHAPED 数据，而 OS/390 通常使用 SHAPED VISUAL 数据。因而，若对双向属性没有任何支持，则从 DB2 通用数据库 OS/390 版发送至 Windows 32 位操作系统工作站上的 DB2 UDB 的数据将不能正确显示。

### 特定于双向的 CCSID

DB2 通过专门的双向“编码字符集标识符”(CCSID) 来支持双向数据属性。下列双向 CCSID 已定义并在 DB2 UDB 中实现：

CCSID (dec)	CCSID (hex)	代码 页	字符串 类型
00420	x'01A4'	420	4
00424	x'01A8'	424	4
08612	x'21A4'	420	5
08616	x'21A8'	424	10
00856	x'0358'	856	5
00862	x'035E'	862	4
00864	x'0360'	864	5
00916	x'0394'	916	5
01046	x'0416'	1046	5
01089	x'0441'	1089	5
01255	x'04E7'	1255	5
01256	x'04E8'	1256	5
62208	x'F300'	856	4
62209	x'F301'	862	10
62210	x'F302'	916	4
62211	x'F303'	424	5
62213	x'F305'	862	5
62215	x'F307'	1255	4
62218	x'F30A'	864	4
62220	x'F30C'	856	6
62221	x'F30D'	862	6

62222	x'F30E'	916	6
62223	x'F30F'	1255	6
62224	x'F310'	420	6
62225	x'F311'	864	6
62226	x'F312'	1046	6
62227	x'F313'	1089	6
62228	x'F314'	1256	6
62229	x'F315'	424	8
62230	x'F316'	856	8
62231	x'F317'	862	8
62232	x'F318'	916	8
62233	x'F319'	420	8
62234	x'F31A'	420	9
62235	x'F31B'	424	6
62236	x'F31C'	856	10
62237	x'F31D'	1255	8
62238	x'F31E'	916	10
62239	x'F31F'	1255	10
62240	x'F320'	424	11
62241	x'F321'	856	11
62242	x'F322'	862	11
62243	x'F323'	916	11
62244	x'F324'	1255	11
62245	x'F325'	424	10
62246	x'F326'	1046	8
62247	x'F327'	1046	9
62248	x'F328'	1046	4
62249	x'F329'	1046	12
62250	x'F32A'	420	12

其中“CDRA 字符串类型”定义为:

字符串类型	文本类型	数字形状	方向	整形	对称交换
4	可视的	Passthru	LTR	已整形	OFF
5	隐式	阿拉伯	LTR	未整形	ON
6	隐式	阿拉伯	RTL	未整形	ON
7(*)	可视的	阿拉伯	上下文(*)	未整形 Lig	OFF
8	可视的	阿拉伯	RTL	已整形	OFF
9	可视的	Passthru	RTL	已整形	ON
10	隐式	Passthru	上下文-L	未整形	ON
11	隐式	Passthru	上下文-R	未整形	ON
12	隐式	阿拉伯	RTL	已整形	ON

注: (\*) 当第一个字母字符为拉丁字符时, 字段方向为从左到右 (LTR); 当第一个字母字符为双向 (RTL) 字符时, 字段方向为从右到左 (RTL)。字符未整形, 但保留 LamAlef 连字, 而没有将其分开。

### 双向支持的 DB2 通用数据库实现

使用新的 CCSID 定义在 DB2 通用数据库中实现了双向格式转换。对于新的 BiDi 特定的 CCSID, 执行格式转换来代替代码页转换或在代码页转换之后再执行它。要

使用此支持，必须将 DB2BIDI 注册表变量设置为 YES。缺省情况下，不设置此变量。服务器对所有转换设置此变量，且只有在服务器启动后才能设置。因为存在附加检查和格式转换，将 DB2BIDI 设置为 YES 可能会影响性能。

要在非 DRDA 环境中指定特定的双向 CCSID，（从上表中）选择与您的客户机特性匹配的 CCSID，并将 DB2CODEPAGE 设置为该值。若您已有与该数据库的一个连接，您必须发出 TERMINATE 命令，并再次连接以使 DB2CODEPAGE 的新设置生效。若您选择对于您的客户机平台的代码页或字符串类型不适当的 CCSID，将得到意外的结果。若您选择了不兼容的 CCSID（例如，对于与阿拉伯语数据库的连接，选择了希伯来语 CCSID），或者若未为该服务器设置 DB2BIDI，当您尝试连接时会接收到出错信息。

对于 DRDA 环境，若 HOST EBCDIC 平台也支持这些双向的 CCSID，您只须设置 DB2CODEPAGE 值。然而，若 HOST 平台不支持这些 CCSID，则您还必须为将要连接的 HOST 数据库服务器指定一个 CCSID 覆盖。这是必需的，因为在 DRDA 环境中，代码页转换和格式转换由数据接收器执行。然而，若 HOST 服务器不支持这些双向 CCSID，它将不对从 DB2 UDB 接收的数据执行格式转换。若您使用 CCSID 覆盖，DB2 UDB 客户机一样对出站数据执行格式转换。有关设置 CCSID 覆盖的信息，参考 《DB2 Connect 用户指南》。

对于 HOST EBCDIC 平台是客户机，而 DB2 UDB 是服务器的情况，不支持 CCSID 覆盖。

### 双向支持的 DB2 Connect 实现

当在 DB2 Connect 和服务器上的数据库之间交换数据时，对入局数据执行转换的往往是接收器。除通常的代码页转换外，同样的约定通常也适用于双向格式转换。DB2 Connect 也能够对它准备发送到服务器数据库的数据和从服务器数据库接收到的数据执行双向格式转换。

为了 DB2 Connect 能够对服务器数据库的出局数据执行双向格式转换，必须替换服务器数据库的双向 CCSID。为此，可在服务器数据库的 DCS 数据库目录项的 PARMs 字段中使用 BIDI 参数。

**注：**若希望 DB2 Connect 对准备发送到 DB2 主机数据库的数据执行格式转换，即使不必替换其 CCSID，也必须将 BIDI 参数添加到 DCS 数据库目录的 PARMs 字段中。在这种情况下，应提供的 CCSID 是 DB2 主机数据库的缺省 CCSID。

将 BIDI 参数指定为 PARMs 字段中的第九个参数，同时指定双向 CCSID，希望替换缺省的服务器数据库双向 CCSID:

```
",,,,,,,,,BIDI=xyz"
```

其中 xyz 是 CCSID 覆盖

**注:** 要使 BIDI 参数生效, 必须将注册表变量 DB2BIDI 设置为 YES。

有关受支持的双向 CCSID 的列表和它们的字符串类型, 可在第221页的『特定于双向的 CCSID』中找到。

为了最好地描述如何使用此功能部件, 举例如下。

假设您有一个运行 CCSID 62213 (双向字符串类型 5) 的希伯来语 DB2 客户机, 而您又想访问运行 CCSID 00424 (双向字符串类型 4) 的 DB2 主机数据库。但您知道包含在 DB2 主机数据库中的数据是基于 CCSID 08616 (双向字符串类型 6)。

这样, 就出现两个问题: 第一个问题是, DB2 主机数据库不能区分 CCSID 00424 和 08616 的双向字符串类型。第二个问题是, DB2 主机数据库不能识别 DB2 客户机的 CCSID (62213)。它只支持 CCSID 00862, 它与 CCSID 62213 基于相同的代码页。

需要确保发送到 DB2 主机数据库的数据以双向字符串类型 6 格式开始, 并且要让 DB2 Connect 知道它必须对从 DB2 主机数据库接收到的数据执行双向转换。需要对 DB2 主机数据库使用以下编目命令:

```
db2 catalog dcs database nydb1 as telaviv parms ",,,,,,,,BIDI=08616"
```

此命令告诉 DB2 Connect 用 CCSID 08616 覆盖 DB2 主机数据库的 CCSID 00424。此覆盖包括下列处理:

1. 将 DB2 Connect 连接到使用 CCSID 00862 的 DB2 主机数据库。
2. DB2 Connect 对它准备发送给 DB2 主机数据库的数据执行双向格式转换即从 CCSID 62213 (双向字符串类型 5) 转换为 CCSID 62221 (双向字符串类型 6)。
3. DB2 Connect 对它从 DB2 主机数据库接收到的数据执行双向格式转换。即从 CCSID 08616 (双向字符串类型 6) 转换为 CCSID 62213 (双向字符串类型 5)。

**注:** 有些情况下, 使用双向 CCSID 可能造成 SQL 查询本身被修改, 结果使 DB2 服务器不能识别该 SQL 查询。特别是, 当可以使用另一种字符串类型时, 应避免使用 IMPLICIT CONTEXTUAL CCSID 和 IMPLICIT RIGHT-TO-LEFT CCSID。若 SQL 查询包括引用字符串, 则 CONTEXTUAL CCSID 可能产生不可预测的结果。避免在 SQL 语句中使用引用字符串; 尽可能使用主变量。

若某个特定的双向 CCSID 导致不能按照下列建议改正的问题, 则将 DB2BIDI 设置为 NO。

## 整理顺序

数据库管理器使用整理顺序来比较字符数据。这是对一组字符的排序，确定某个字符的 ASCII 码是大于、小于还是等于另一个字符的 ASCII 码。例如，可使用整理顺序来指示特定字符的小写与大写的 ASCII 码相等。

整理顺序在创建数据库时指定，且以后不能修改。

数据库管理器允许使用应用程序编程接口 (API) 来通过定制整理顺序创建数据库。有关实现定制整理顺序表的信息，参考 *Application Development Guide*。

**注：**用 FOR BIT DATA 属性定义的字符串数据和 BLOB 数据使用二进制排序顺序进行排序。

### 一般注意事项

一旦定义了整理顺序，就将用该整理顺序执行该数据库将来的所有字符比较。除了被定义为 FOR BIT DATA 或 BLOB 数据的字符数据以外，整理顺序将被用于所有的 SQL 比较和 ORDER BY 子句，以及建立索引和统计信息。要获取关于如何使用数据库整理顺序的更多信息，参见 *SQL Reference* 中的“字符串比较”。

在下列情况下可能会出现潜在的问题：

- 应用程序将数据库中已排序的数据与应用程序的数据合并在一起，应用程序数据是使用不同的整理顺序排序的。
- 应用程序将一个数据库中已排序的数据与另一个数据库中已排序的数据合并在一起，但这两个数据库的整理顺序不同。
- 应用程序对排序的数据所作的假定不符合相关的整理顺序。例如，数字排在字母之前对特定的整理顺序可能适用也可能不适用。

最后一点要记住的是，对字符代码点进行直接比较所得到的任何排序的结果将只与使用等同的整理顺序排序的查询结果匹配。

### 联合数据库注意事项

选择的数据库整理顺序可影响联合系统性能。若数据源与 DB2 联合数据库使用相同的整理顺序，DB2 可将涉及到字符数据的与次序有关的处理下推到数据源。若数据源整理顺序与 DB2 的整理顺序不匹配，检索数据并在本地执行涉及到字符数据的与次序有关的处理（这可能降低性能）。

要确定数据源和 DB2 是否有相同的整理顺序，考虑下列因素：

- 本地语言支持。  
整理顺序与服务器上受支持的语言相关。比较 DB2 NLS 信息和数据源 NLS 信息。

- 数据源特征。

某些数据源是用不区分大小写的整理顺序创建的，在与次序相关的操作中这可能产生不同的结果。

- 定制。

某些数据源为整理顺序提供多个选项，或允许定制整理顺序。

为一个 DB2 联合数据库选择整理顺序，以将从该数据库访问的混合数据源为根据。例如：

- 若 DB2 数据库主要访问与 DB2 有相同代码页 (NLS) 的 Oracle 数据库，则在数据库创建时指定标识顺序 (Oracle 数据库使用一个等效的整理顺序)。
- 若 DB2 数据库只访问 DB2 UDB 数据库，确保与整理顺序值匹配。

有关设置 MVS 整理顺序的信息，参考 *Administrative API Reference*，您可以在 **sqlcrea** (创建数据库 API) 的描述下面找到有关示例。这些样本包含 EBCDIC 500、37 和 5026/5035 代码页的整理表。

设置 DB2 数据库的整理顺序之后，确保为每个数据源服务器设置 *collating\_sequence* 服务器选项。此选项指定给定数据源服务器的整理顺序是否与 DB2 数据库的整理顺序匹配。

若整理顺序匹配，则将 *collating\_sequence* 选项设置为 "Y"。此设置可使 DB2 优化器考虑在数据源进行与次序有关的处理，这可改进性能。但是，若数据源整理顺序与 DB2 数据库整理顺序不同，将接收到不正确的结果。例如，若您的计划使用合并连接，DB2 优化器将把尽可能多的排序操作下推至数据源。若数据源整理顺序不同，则连接结果集可能不正确。

若整理顺序不匹配，则将 *collating\_sequence* 选项设置为 "N"。当数据源整理顺序与 DB2 不同时，或当数据源整理操作可能不区分大小写时，使用此值。例如，在有英语代码页的不区分大小写的数据源中，将认为 TOLLESON、ToLLeSoN 和 tolleson 全部相同。若无法确定数据源的整理顺序与 DB2 整理顺序是否完全相同，可将 *collating\_sequence* 选项设置为 "N"。

### 排序泰国字符

如果想要正确排序特殊元音（“前导元音”）、音调标记和其它特殊泰国字符，则必须在 CREATE DATABASE 命令中使用 COLLATE USING NLSCHAR 子句来创建数据库。有关语法，参见 *Command Reference*。

## 日期时间值

日期时间数据类型描述如下。尽管日期时间值可以在算术和字符串运算中使用，且与特定的字符串兼容，但它们既不是字符串也不是数字。



## 日期

日期值分为三个部分（年、月以及日）年部分的范围为 0001 至 9999。月部分的范围为 1 至 12。日部分的范围为 1 至  $x$ ，其中  $x$  取决于月。

日期的内部表示是 4 字节的字符串。每个字节由 2 个压缩十进制数字组成。前两个字节表示年，第三个字节表示月，最后一个字节表示日。

DATE 列的长度，如 SQLDA 中所描述的，是 10 字节，它是该值的字符串表示的适当长度。

## 时间

时间是一个三部分值（小时、分钟和秒），指定在 24 小时时钟下一天中的时间小时部分的范围是 0 至 24。其他部分的范围是 0 至 59。若小时是 24，则分钟和秒的规范为零。

时间的内部表示是 3 字节的字符串。每个字节由 2 个压缩十进制数字组成。第一个字节表示小时，第二个字节表示分钟，最后一个字节表示秒。

TIME 列的长度，如 SQLDA 中所描述的，是 8 字节，它是该值的字符串表示的适当长度。

## 时间戳记

时间戳记是一个七部分值（年、月、日、小时、分钟、秒和微秒），它指定以上定义的日期和时间，但不同的是这个时间包括微秒的规范。

时间戳记的内部表示是 10 字节的字符串。每个字节由 2 个压缩十进制数字组成。前 4 个字节表示日期，接着的 3 个字节表示时间，最后 3 个字节表示微秒。

TIMESTAMP 列的长度，如 SQLDA 中所描述的，是 26 字节，它是该值的字符串表示的适当长度。

## 日期时间值的字符串表示

其数据类型是 DATE、TIME 或 TIMESTAMP 的值是用对于 SQL 用户透明的一种内部格式表示的。但是，日期、时间和时间戳记也可用字符串表示，而这些表示直接涉及到 SQL 用户，因为没有其数据类型是 DATE、TIME 或 TIMESTAMP 的常数或变量。因此，要检索，必须将一个日期时间值分配给一个字符串变量。除非在对程序进行预编译或将其绑定至数据库时用 "F" 格式选项的规范覆盖了字符串表示，否则，字符串表示通常是与客户机的国家 / 地区代码相关的日期时间值的缺省格式。有关各种国家 / 地区代码的字符串格式的列表，参见第 230 页的表 33。

当在使用一个内部日期时间值的操作中使用日期时间值的有效字符串表示时，在执行该操作前，将该字符串表示转换为日期、时间或时间戳记的内部格式。下面几节定义日期时间值的有效字符串表示。

### 日期字符串

日期的字符串表示是从一个数字开始且长度至少为 8 个字符的一个字符串。可以包括尾部空格；可以从日期的月部分和日部分中略去前导零。

表31中列出了日期的有效字符串格式。每个格式由名称标识，并包括一个相关的缩写和有关其使用的示例。

表 31. 日期的字符串表示的格式

格式名	缩写	日期格式	示例
国际标准化组织	ISO	yyyy-mm-dd	1991-10-27
IBM 美国标准	USA	mm/dd/yyyy	10/27/1991
IBM 欧洲标准	EUR	dd.mm.yyyy	27.10.1991
日本工业标准公元	JIS	yyyy-mm-dd	1991-10-27
位置定义的（本地）	LOC	取决于数据库的 国家 / 地区代码	—

### 时间字符串

时间的字符串表示是从一个数字开始且长度至少为 4 个字符的一个字符串。可以包括尾部空格；可以从时间的小时部分中略去前导零，并可整个略去秒。若您选择略去秒，则认为是 0 秒的隐式规范。因此，13.30 等于 13.30.00。

表32中列出了时间的有效字符串格式每个格式由名称标识，并包括一个相关的缩写和有关其使用的示例。

表 32. 时间的字符串表示的格式

格式名	缩写	时间格式	示例
国际标准化组织	ISO	hh.mm.ss	13.30.05
IBM 美国标准	USA	hh:mm AM 或 PM	1:30 PM
IBM 欧洲标准	EUR	hh.mm.ss	13.30.05
日本工业标准公元	JIS	hh:mm:ss	13:30:05
位置定义的（本地）	LOC	取决于应用程序的 国家 / 地区代码	—

注:

1. 在 ISO、EUR 或 JIS 时间字符串格式中，.ss（或 :ss）是可选的。
2. 对于 USA 时间字符串格式，可以略去分钟规范，指示 00 分钟的隐式规范。因此，1 PM 等效于 1:00 PM。
3. 在 USA 时间字符串格式中，小时规范不能大于 12 且不能是 0，除非是 00:00 AM 这种特殊情况。使用 24 小时时钟的 ISO 格式，在 USA 格式和 24 小时时钟之间的对应性如下所示：
  - 12:01 AM 至 12:59 AM 对应于 00.01.00 至 00.59.00。
  - 01:00 AM 至 11:59 AM 对应于 01.00.00 至 11.59.00。
  - 12:00 PM（中午）至 11:59 PM 对应于 12.00.00 至 23.59.00。
  - 12:00 AM（午夜）对应于 24.00.00，而 00:00 AM（午夜）对应于 00.00.00。

### 时间戳记字符串

时间戳记的字符串表示是从一个数字开始且长度至少为 16 个字符的一个字符串。一个时间戳记的完整字符串表示的格式为 `yyyy-mm-dd-hh.mm.ss.nnnnnn`。可以包括尾部空格；可以从该时间戳记的月、日和小时部分略去前导零，并且可以截断微秒或整个略去它。若您选择略去微秒部分的任何数字，则认为是 0 的隐式规范。因此，1991-3-2-8.30.00 等于 1991-03-02-08.30.00.000000。

### 字符集注意事项

日期和时间戳记字符串只能包含单字节字符和数字。

### 日期和时间格式

用字符串表示的日期和时间格式是与应用程序的国家 / 地区代码相关的日期时间值的缺省格式。当预编译程序或将其绑定至数据库时，可指定 "F" 格式选项来覆盖此缺省格式。

以下是日期和时间的输入和输出格式的描述:

- 输入时间格式
  - 无缺省输入时间格式
  - 对于所有国家 / 地区代码，允许所有时间格式作为输入。
- 输出时间格式
  - 缺省输出时间格式等于本地时间格式。
- 输入日期格式
  - 无缺省输入日期格式
  - 在日期的本地格式与 ISO、JIS、EUR 或 USA 日期格式冲突之处，将把本地格式作为日期输入格式。例如，查看第230页的表33中的 UK 项。

- 输出日期格式
  - 在表33中显示了缺省输出日期格式

注：表33也显示了各个国家 / 地区代码的该字符串格式的列表。

表 33. 按国家 / 地区代码划分的日期和时间格式

国家 / 地区代码	本地日期格式	本地时间格式	缺省输出日期格式	输入日期格式
355 阿尔巴尼亚	yyyy-mm-dd	JIS	LOC	LOC, USA, EUR, ISO
785 阿拉伯	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
001 澳大利亚 (1)	mm-dd-yyyy	JIS	LOC	LOC, USA, EUR, ISO
061 澳大利亚	dd-mm-yyyy	JIS	LOC	LOC, USA, EUR, ISO
032 比利时	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
055 巴西	dd.mm.yyyy	JIS	LOC	LOC, EUR, ISO
359 保加利亚	dd.mm.yyyy	JIS	EUR	LOC, USA, EUR, ISO
001 加拿大	mm-dd-yyyy	JIS	USA	LOC, USA, EUR, ISO
002 加拿大 (法语区)	dd-mm-yyyy	ISO	ISO	LOC, USA, EUR, ISO
385 克罗地亚	yyyy-mm-dd	JIS	ISO	LOC, USA, EUR, ISO
042 捷克共和国	yyyy-mm-dd	JIS	ISO	LOC, USA, EUR, ISO
045 丹麦	dd-mm-yyyy	ISO	ISO	LOC, USA, EUR, ISO
358 芬兰	dd/mm/yyyy	ISO	EUR	LOC, EUR, ISO
389 FYR 马其顿	dd.mm.yyyy	JIS	EUR	LOC, USA, EUR, ISO
033 法国	dd/mm/yyyy	JIS	EUR	LOC, EUR, ISO
049 德国	dd/mm/yyyy	ISO	ISO	LOC, EUR, ISO
030 希腊	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
036 匈牙利	yyyy-mm-dd	JIS	ISO	LOC, USA, EUR, ISO

表 33. 按国家 / 地区代码划分的日期和时间格式 (续)

国家 / 地区代码	本地日期格式	本地时间格式	缺省输出日期格式	输入日期格式
354 冰岛	dd-mm-yyyy	JIS	LOC	LOC, USA, EUR, ISO
091 印度	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
972 以色列	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
039 意大利	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
081 日本	mm/dd/yyyy	JIS	ISO	LOC, USA, EUR, ISO
082 韩国	mm/dd/yyyy	JIS	ISO	LOC, USA, EUR, ISO
001 拉丁美洲 (1)	mm-dd-yyyy	JIS	LOC	LOC, USA, EUR, ISO
003 拉丁美洲	dd-mm-yyyy	JIS	LOC	LOC, EUR, ISO
031 荷兰	dd-mm-yyyy	JIS	LOC	LOC, USA, EUR, ISO
047 挪威	dd/mm/yyyy	ISO	EUR	LOC, EUR, ISO
048 波兰	yyyy-mm-dd	JIS	ISO	LOC, USA, EUR, ISO
351 葡萄牙	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
086 中国	mm/dd/yyyy	JIS	ISO	LOC, USA, EUR, ISO
040 罗马尼亚	yyyy-mm-dd	JIS	ISO	LOC, USA, EUR, ISO
007 俄罗斯	dd/mm/yyyy	ISO	LOC	LOC, EUR, ISO
381 塞尔维亚 / 蒙的内哥罗	yyyy-mm-dd	JIS	ISO	LOC, USA, EUR, ISO
042 斯洛伐克	yyyy-mm-dd	JIS	ISO	LOC, USA, EUR, ISO
386 斯洛文尼亚	yyyy-mm-dd	JIS	ISO	LOC, USA, EUR, ISO
034 西班牙	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
046 瑞典	dd/mm/yyyy	ISO	ISO	LOC, EUR, ISO
041 瑞士	dd/mm/yyyy	ISO	EUR	LOC, EUR, ISO
088 台湾	mm-dd-yyyy	JIS	ISO	LOC, USA, EUR, ISO

表 33. 按国家 / 地区代码划分的日期和时间格式 (续)

国家 / 地区代码	本地日期格式	本地时间格式	缺省输出日期格式	输入日期格式
066 泰国 (2)	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
090 土耳其	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
044 英国	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
001 美国	mm-dd-yyyy	JIS	USA	LOC, USA, EUR, ISO
084 越南	dd/mm/yyyy	JIS	LOC	LOC, EUR, ISO
注:				
1. 将国家 / 地区代码 001 指定给使用缺省 C 语言环境的国家 / 地区。				
2. 佛教纪元的 yyyy 等价于罗马教的 yyyy + 543 年 (限于泰国)。				

## DB2 UDB 中的 Unicode 支持

DB2 UDB 中 Unicode 支持的级别描述如下。

### 介绍

Unicode 字符编码标准是定长的字符编码模式，它包含了世界上几乎所有现用语言的字符。

Unicode 使用两种编码格式：8 位和 16 位。缺省编码格式是 16 位，即每个字符是 16 位（两个字节）宽，并且通常显示为 U+hhhh，其中 hhhh 是字符的十六进制代码点。生成的 65 000+ 代码元素足以用于编码世界上主要语言的大多数字符，Unicode 标准还提供了一种扩展机制，允许编码一百多万字符。扩展机制使用一对高位和低位代用字符来编码扩展的字符。第一个（或高位）代用字符具有 U+D800 和 U+DBFF 之间的代码值，而第二个（或低位）代用字符具有 U+DC00 和 U+DFFF 之间的代码值。

“国际标准化组织” (ISO) 和 “国际电工委员会” (IEC) 标准 10646 (ISO/IEC 10646) 指定了 “通用多八位元编码字符集” (UCS)，该编码字符集有一个 16 位（双字节）版本 (UCS-2) 和一个 32 位（四字节）版本 (UCS-4)。UCS-2 相当于没有代用字符的 Unicode 16 位格式。ISO/IEC 10646 还定义了一种扩展技术，以使用两个 UCS-2 字符来编码某些 UCS-4 字符。此扩展技术称为 UTF-16，它相当于有代用字符的 Unicode 16 位编码格式。总之，UTF-16 字符指令集由所有 UCS-2 字符和可通过代用字符对访问的附加的一百万字符构成。

当将 16 位 Unicode 字符串行化为字节时，某些处理器将最重要的字节放置在初始位置（称为大尾数法排序），而其它的处理器则首先放置最不重要的字节（称为小尾数法排序）。Unicode 的缺省字节排序是大尾数法。

有关 Unicode 的更多信息可在最新版本的“Unicode 标准”一书中找到，并且可从 Unicode Consortium Web 站点 ([www.unicode.org](http://www.unicode.org)) 找到。

### UTF-8

对于面向字节基于 ASCII 的应用程序和文件系统，16 位 Unicode 字符是引起问题的主要因素。例如，不明白 Unicode 的应用程序可能会将大写字符 'A' (U+0041) 的 8 个前导零位误解为单字节 ASCII NULL 字符。

UTF-8 (UCS 变换格式 8) 是一种算法变换，它将定长 Unicode 字符变换为变长 ASCII 安全的字节字符串。在 UTF-8 中，ASCII 和控制字符由通常的单字节代码表示，但其它字符变为双字节或更多字节。使用 UTF-8 格式的每个 UTF-16 字符的字节数可根据表 34 来确定。

表 34. UTF-8 的位分布

代码值 (二进制)	UTF-16 (二进制)	第一个字节 (二进制)	第二个字节 (二进制)	第三个字节 (二进制)	第四个字节 (二进制)
00000000 0xxxxxxx	00000000 0xxxxxxx	0xxxxxxx			
00000yyy yyxxxxxx	00000yyy yyxxxxxx	110yyyyy	10xxxxxx		
zzzzyyyy yyxxxxxx	zzzzyyyy yyxxxxxx	1110zzzz	10yyyyyy	10xxxxxx	
uuuuu zzzzyyyy yyxxxxxx	110110ww wwzzzzyy 110111yy yyxxxxxx	11110uuu  (where uuuu = wwww+1)	10uuzzzz	10yyyyyy	10xxxxxx

在以上每一项中，u、w、x、y 和 z 串都是字符的位表示。例如，U+0080 变换为二进制中的 11000010 10000000，而代用字符对 U+D800 U+DC00 变为二进制中的 11110000 10010000 10000000 10000000。

## DB2 UDB 中的 Unicode 实现

DB2 UDB 支持 UCS-2; 即无代用字符的 Unicode。

在版本 7.2 修订包 4 之前的 DB2 UDB 版本中, DB2 UDB 将代用字符对中的两个字符看作两个独立的 Unicode 字符。因此, 将该代用字符对从 UTF-16/UCS-2 变换为 UTF-8 会导致两个三字节序列 (参见第233页的表34中的倒数第二行)。从 DB2 UDB 版本 7.2 修订包 4 开始, DB2 UDB 在 UTF-16/UCS-2 和 UTF-8 之间变换时识别代用字符对, 因此一对 UTF-16 代用字符将会变为一个 UTF-8 四字节序列 (参见第233页的表34中的最后一行)。

DB2 UDB 将每个 Unicode 字符看作一个单独字符, 其中包括诸如 COMBINING ACUTE ACCENT 字符 (U+0301) 的那些 (非空格) 字符。因此, DB2 UDB 将认为字符 LATIN SMALL LETTER A WITH ACUTE (U+00E1) 照规范来讲与后跟字符 COMBINING ACUTE ACCENT (U+0301) 的字符 LATIN SMALL LETTER A (U+0061) 不是等价的。

### 代码页 / CCSID 号码

在 IBM 内部, 已将 UCS-2 代码页注册为代码页 1200, 它具有不断增加的字符集, 也就是说当向一个代码页添加新字符时, 代码页号码不变。代码页 1200 始终引用 Unicode 的当前版本。

UCS 标准的特定版本由 Unicode 2.0 和 ISO/IEC 10646-1 定义, 在 IBM 内部已将它注册为 CCSID 13488。此 CCSID 已在 DB2 UDB 内部使用, 用于存储 euc-Japan 和 euc-Taiwan 数据库中的字符串数据。CCSID 13488 和代码页 1200 都引用 UCS-2, 除了它们的“双字节”(DBCS) 空格的值不同外, 其处理方式是相同的:

CP/CCSID	单字节 (SBCS) 空间	双字节 (DBCS) 空间
1200	N/A	U+0020
13488	N/A	U+3000

注意: 在 UCS-2 数据库中, U+3000 没有任何特殊意义。

关于转换表, 由于代码页 1200 是 CCSID 13488 的超集, 所以对它们使用了相同的 (超集) 表。

在 IBM 中, 已将 UTF-8 注册为具有增长的字符集的 CCSID 1208 (有时也称为代码页 1208)。当向标准集中添加新字符, 此号码 (1208) 不会改变。

MBCS 代码页号码为 1208, 它是数据库代码页号码以及该数据库中字符串的代码页。UCS-2 的双字节代码页号码为 1200, 它是数据库中的图形字符串数据的代码页。当创建 Unicode 数据库时, CHAR、VARCHAR、LONG VARCHAR 和 CLOB



数据以 UTF-8 格式存储，而 GRAPHIC、VARGRAPHIC、LONG VARGRAPHIC 和 DBCLOB 数据以 UCS-2 格式存储。

### 创建 Unicode 数据库

缺省情况下，用创建数据库的应用程序的代码页创建数据库。因此，若从 Unicode (UTF-8) 客户机（例如，AIX 的 UNIVERSAL 语言环境或者若将客户机上的 DB2CODEPAGE 注册表变量设置为 1208）创建数据库，您的数据库将创建为 Unicode 数据库。或者，可显式地将“UTF-8”指定为 CODESET 名，并使用 DB2 UDB 支持的任何有效 TERRITORY 代码。

例如，要使用美国区域代码创建 Unicode 数据库，发出如下命令：

```
DB2 CREATE DATABASE dbname USING CODESET UTF-8 TERRITORY US
```

要使用 **sqlcrea** API 创建 Unicode 数据库，应相应地在 *sqldbcountryinfo* 中设置值。例如，将 SQLDBCODESET 设置为 UTF-8，将 SQLDBLOCALE 设置为任何有效的区域代码（例如 US）。

UCS-2 Unicode 数据库的缺省整理序列是 IDENTITY，它根据代码点对字符进行排序。因此，缺省情况下，将根据代码点来排序和比较所有 Unicode 字符。对于大多数 Unicode 字符，当以 UTF-8 和 UCS-2 编码时，它们的二进制整理次序是相同的。但是，如果具有需要一对代用字符来编码的任何扩展字符，则以 UTF-8 编码时将把字符向末尾整理，但是以 UCS-2 编码时将把同一字符向中间的某处整理，并且可以将它的两个代用字符分开。原因在于当以 UTF-8 编码时，扩展字符具有四字节二进制代码值 11110xxx 10xxxxxx 10xxxxxx 10xxxxxx，它大于 UTF-8 编码 U+FFFF。但是，在 UCS-2 中，将同一扩展字符编码为两个 UCS-2 代用字符，并且 DB2 UDB 独立地处理和整理每个代用字符。

所有与文化相关的参数，如日期或时间格式、小数分隔符等等，基于客户机的当前区域。

Unicode 数据库允许与 DB2 UDB 支持的所有代码页连接。客户机代码页和 UTF-8 之间的代码页字符转换由数据库管理器自动执行。图形字符串类型的数据通常使用 UCS-2，不必通过代码页转换。命令行处理器 (CLP) 环境是个例外。若从 CLP 选择图形字符串 (UCS-2) 数据，则返回的图形字符串数据将从 UCS-2 转换为客户机环境的代码页。

任何客户机均受字符编码系统、输入方法和它的环境所支持的字体限制，但是 UCS-2 数据库本身接受并存储所有 UCS-2 字符。因此，每个客户机通常使用 UCS-2 字符的一个子集，但数据库管理器允许使用 UCS-2 字符的整个编码系统。

当字符由本地代码页转换为 UTF-8 时，字节数可能增多。ASCII 字符不会增多，但其他 UCS-2 字符将增大两三倍。UTF-8 格式中的每个 UCS-2 字符的字节数可根据第233页的『UTF-8』中的表来确定。

### 数据类型

DB2 UDB 所支持的所有数据类型在 UCS-2 数据库中也受支持。特别是，UCS-2 数据库支持图形字符串数据，并以 UCS-2/Unicode 存储。每个客户机（包括 SBCS 客户机）与 UCS-2 数据库连接时，可使用 UCS-2/Unicode 格式的图形字符串数据类型。

UCS-2 数据库同任何 MBCS 数据库一样，其字符串数据以字节数计。当使用 UTF-8 格式的字符串数据时，不应当认为每个字符都是单字节。在多字节 UTF-8 编码中，每个 ASCII 字符都是单字节，但每个非 ASCII 字符为两个至四个字节。当定义 CHAR 字段时应考虑这点。取决于 ASCII 与非 ASCII 字符的比率，一个大小为  $n$  字节的 CHAR 字段可包含  $n/3$  到  $n$  之间的任意多个字符。

对图形字符串 UCS-2 数据类型使用字符串 UTF-8 编码也对总存储器需求有影响。对于大多数字符是 ASCII 但其间插入一些非 ASCII 字符的情况，存储 UTF-8 数据可能是比较好的替代方法，因为存储器需求接近每个字符一个字节。另一方面，在大多数字符是扩充为三字节或四字节 UTF-8 序列的非 ASCII 字符（例如，表意字符）的情况下，UCS-2 图形字符串格式可能是比较好的替代方法，因为每个三字节 UTF-8 序列变为一个 16 位 UCS-2 字符，而每个四字节 UTF-8 序列变为两个 16 位 UCS-2 字符。

在 MBCS 环境中处理字符串的 SQL 标量函数，如 LENGTH、SUBSTR、POSSTR、MAX 和 MIN 等，是基于“字节”数而不是“字符”数进行运算的。该行为在 UCS-2 数据库中也一样，但当为 UCS-2 数据库指定偏移量和长度时应格外小心，因为始终是在数据库代码页的上下文中定义这些值。也就是说，对于 UCS-2 数据库，应在 UTF-8 中定义这些值。因为一些单字节字符需要 UTF-8 格式的多字节，因此对单字节数据库有效的 SUBSTR 索引可能对 UCS-2 数据库无效。若指定了不正确的索引，则将返回 SQLCODE -191 (SQLSTATE 22504)。有关这些函数行为的描述，参考 *SQL Reference*。

在用户程序中，（C 语言中的）char 数据类型支持 SQL CHAR 数据类型。在用户程序中，sqlbchar 支持 SQL GRAPHIC 数据类型。注意，对于 UCS-2 数据库，sqlbchar 数据始终使用大尾结构（高字节在前）格式。当将应用程序与 UCS-2 数据库连接时，由 DB2 UDB 在应用程序代码页和 UTF-8 之间转换字符串数据，而图形字符串数据始终是 UTF-2 格式。

## 标识符

在 UCS-2 数据库中，所有标识符都使用多字节 UTF-8。因此，对于 DB2 UDB 允许在其中使用扩充字符集中的字符（例如，重音字符或多字节字符）的标识符，可以在该标识符中使用任何 UCS-2 字符。有关哪些标识符允许使用扩充字符的详细信息，参见第169页的『附录A. 命名规则』。

客户机可输入其环境支持的任何字符，并且标识符中的所有字符都由数据库管理器转换为 UTF-8。当在 UCS-2 数据库的标识符中指定本地语言字符时，必需考虑以下两点：

- 每个非 ASCII 字符需要两个或四个字节。因此，一个  $n$  字节标识符只能容纳  $n/3$  至  $n$  个字符，这取决于 ASCII 与非 ASCII 字符的比率。若仅有一个或两个非 ASCII 字符（例如，重音符），该限制接近  $n$  个字符，而对于全部由非 ASCII 字符组成的标识符（例如，日语），只能使用  $n/3$  个字符。
- 若要在不同的客户机环境中输入标识符，应使用这些客户机可用的公共字符子集来定义标识符。例如，若要从拉丁语系 1、阿拉伯语和日语环境中访问 UCS-2 数据库，则所有标识符应严格限制为 ASCII。

## Unicode 文字

可以用下列两种方式指定 Unicode 文字：

- 作为图标字符串常量指定，使用 *SQL Reference* 中“语言元素”一章的“图符字符串常量”一节所描述的 G'...' 或 N'....' 格式。用这种方式指定的任何文字将由数据库管理器从应用程序代码页转换为 16 位 Unicode。
- 作为 Unicode 十六进制字符串，使用 UX'....' 或 GX'....' 格式。在大尾数法排序中，在 UX 或 GX 后面的引号内指定的常量必须是 4 个十六进制数字的倍数。每个四位组表示一个 16 位 Unicode 代码点。注意代用字符始终成对出现，因此需要两个四位组来表示高位和低位代用字符。

当使用命令行处理器 (CLP) 时，若本地应用程序代码页中存在 UCS-2 字符，则第一种方法比较容易（例如，从使用代码页 850 的终端输入代码页 850 的任何字符）。对于在应用程序代码页编码系统之外的字符，应使用第二种方法（例如，从使用代码页 850 的终端指定日语字符）。

## UCS-2 数据库中的模式匹配

模式匹配是现存的 MBCS 数据库的行为与 UCS-2 数据库的行为稍微不同的一个范畴。

对于 DB2 UDB 中的 MBCS 数据库，当前行为如下所示：若匹配表达式包含 MBCS 数据，则模式可以包含 SBCS 和非 SBCS 字符。该模式中的特殊字符解释如下：

- SBCS 下划线字符是一个 SBCS 字符。

- DBCS 下划线字符是一个 MBCS 字符。
- 百分比字符（SBCS 或 DBCS）表示由零个或多个 SBCS 或非 SBCS 字符组成的字符串。

在 Unicode 数据库中，“单字节”与“双字节”字符之间没有任何真正的差别；每个 16 位 字符占用两个字节。尽管 UTF-8 格式是 Unicode 字符的“混合字节”编码，但 UTF-8 格式的 SBCS 和 MBCS 字符之间没有任何真正的区别。每个字符是一个 Unicode 字符，而与采用 UTF-8 格式的字节数无关。当指定字符串或图形字符串表达式时，下划线表示一个 Unicode 字符，而百分比符号表示由零个或多个 Unicode 字符组成的字符串。注意，需要两个下划线与一个代用字符相匹配。

在客户机端，字符串表达式使用客户机的代码页，将由数据库管理器转换为 UTF-8。SBCS 客户机代码页没有 DBCS 百分比字符或 DBCS 下划线字符，但每个受支持的代码页都包含一个单字节百分比字符（与 U+0025 对应）和一个单字节下划线字符（与 U+005F 对应）。UCS-2 数据库的特殊字符解释如下：

- SBCS 下划线字符（与 U+0025 对应）在图形字符串表达式中是一个 UCS-2 字符，而在字符串表达式中是一个 UTF-8 字符。
- SBCS 百分比符号（与 U+005F 对应）在图形字符串表达式中表示由零个或多个 UCS-2 字符组成的字符串，而在字符串表达式中表示由零个或多个 UTF-8 字符组成的字符串。

DBCS 代码页还支持 DBCS 百分比符号（与 U+FF05 对应）和 DBCS 下划线字符（与 U+FF3F 对应）。对于 UCS-2 数据库，这些字符没有任何特殊的意义。

对于可选的“转义表达式”，它指定一个用来修改下划线字符和百分比字符特殊意义的字符，仅支持 ASCII 字符或扩充为两字节 UTF-8 序列的字符。若指定扩充为三字节 UTF-8 值的转义字符，则将返回出错信息（错误 SQL0130N, SQLSTATE 22019）。

### 导入 / 导出 / 装入注意事项

正如本节中所述，UCS-2 数据库支持 DEL、ASC 和 PC/IXF 文件格式。不支持 WSF 格式。

当从 UCS-2 数据库导出到 ASCII 定界 (DEL) 文件时，所有字符数据都转换为应用程序代码页。字符串和图形字符串数据都转换为客户机的 SBCS 或 MBCS 代码页。这是导出任何数据库的期望行为，并且不得更改，因为整个 ASCII 定界文件只能有一个代码页。因此，若导出到 ASCII 定界文件中，只会保存应用程序代码页中存在的那些 UCS-2 字符。其他字符用该应用程序代码页的缺省替代字符替换。对于 UTF-8 客户机（代码页 1208），则不会丢失数据，因为 UTF-8 客户机支持所有 UCS-2 字符。

当从 ASCII 文件 (DEL 或 ASC) 导入到 UCS-2 数据库时, 字符串数据由应用程序代码页转换为 UTF-8, 而图形字符串数据由应用程序代码页转换为 UCS-2。不会丢失数据。若想要导入以另一种代码页保存的 ASCII 数据, 在发出 IMPORT 命令之前应更改数据文件代码页。一种方法是, 将 DB2CODEPAGE 设置为 ASCII 数据文件的代码页。

对于 SBCS 和 MBCS 客户机, 有效的 ASCII 定界符的范围就是那些客户机的 DB2 UDB 当前所能支持的范围。对于 UTF-8 客户机, 有效定界符的范围是 X'01' 至 X'7F', 具有一般限制。有关这些限制的完整列表, 参考 *Data Movement Utilities Guide and Reference* 中的“导出 / 导入 / 装入实用程序文件格式”附录。

当从 UCS-2 数据库导出到 PC/IXF 文件时, 字符串数据转换为客户机的 SBCS/MBCS 代码页。不转换图形字符串数据, 将以 UCS-2 (代码页 1200) 格式存储该数据。不会丢失数据。

当从 PC/IXF 文件导入到 UCS-2 数据库时, 假定字符串数据使用存储在 PC/IXF 文件头中的 SBCS/MBCS 代码页, 而图形字符串数据使用存储在 PC/IXF 文件头中的 DBCS 代码页。字符串数据由“导入”实用程序从 PC/IXF 文件头中指定的代码页转换为客户机的代码页, 然后由 INSERT 语句从客户机代码页转换为 UTF-8。图形字符串数据由“导入”实用程序从 PC/IXF 文件头中指定的 DBCS 代码页直接转换为 UCS-2 (代码页 1200)。

“装入”实用程序直接将数据装入数据库, 缺省情况下, 假设 ASC 或 DEL 文件中的数据使用数据库的代码页。因此, 缺省情况下, ASCII 文件不执行代码页转换。当 (使用 codepage 修饰词) 显式指定了数据文件的代码页时, “装入”实用程序在装入数据之前, 使用此信息来将指定的代码页转换为数据库代码页。对于 PC/IXF 文件, “装入”实用程序始终从 IXF 文件头中指定的代码页转换为数据库代码页 (对于 CHAR, 为 1208; 对于 GRAPHIC, 为 1200)。

对于 UCS-2, DBCLOB 文件的代码页总是 1200。CLOB 文件的代码页与要导入、装入或导出的数据文件的代码页相同。例如, 当使用 PC/IXF 格式装入或导入数据时, 将假定 CLOB 文件使用 PC/IXF 文件头指定的代码页。若 DBCLOB 文件处于 ASC 或 DEL 格式, 则“装入”实用程序假定 CLOB 数据使用的是数据库的代码页 (除非使用 codepage 修饰词另有所指), 而“导入”实用程序假定它使用的是客户机应用程序的代码页。

始终对 UCS-2 数据库指定 nochecklengths 修饰词, 因为:

- 可将任何 SBCS 与无 DBCS 代码页的数据库连接
- UTF-8 格式中的字符串长度通常与客户机代码页中的字符串长度不同。

有关“装入”、“导入”和“导出”实用程序的更多信息，参考 *Data Movement Utilities Guide and Reference*。

### 不兼容性

在与 UCS-2 数据库连接的应用程序中，图形字符串数据始终使用 UCS-2（代码页 1200）。对于与非 UCS-2 数据库连接的应用程序，图形字符串数据使用应用程序的 DBCS 代码页；但如果应用程序代码页是 SBCS，则不允许这样做。例如，当将一个 932 客户机与一个使用日语的非 UCS-2 数据库连接时，图形字符串数据使用代码页 301。对于与 UCS-2 数据库连接的 932 客户机应用程序，图形字符串数据使用 UCS-2 格式。

---

## 附录E. 使用 DB2 资料库

“DB2 通用数据库”资料库由联机帮助、书籍（PDF 和 HTML）和 HTML 格式的样本程序组成。本节描述所提供的信息以及如何访问这些信息。

要访问联机产品信息，可以使用“信息中心”。有关更多信息，参见第253页的『通过“信息中心”访问信息』。可以查看任务信息、DB2 书籍、故障诊断信息、样本程序和 Web 上的 DB2 信息。

---

### DB2 PDF 文件和打印的书籍

#### DB2 信息

下表将 DB2 书籍分为四个类别：

##### DB2 指南和参考信息

这些书籍包含所有平台的公共 DB2 信息。

##### DB2 安装和配置信息

这些书籍是针对特定平台上的 DB2 的。例如，有分别针对 OS/2 平台、Windows 平台和基于 UNIX 的平台上 DB2 的《快速入门》书籍。

##### HTML 格式的跨平台样本程序

这些样本是与“应用程序开发客户机”一起安装的样本程序的 HTML 版本。样本仅供参考，并不替代实际程序。

##### 发行说明

这些文件包含 DB2 书籍中未能包括的最新信息。

HTML 格式的安装手册、发行说明和教程可直接在产品 CD-ROM 上看到。大部分书籍在产品 CD-ROM 上都有 HTML 格式以便查看，而在 DB2 出版物 CD-ROM 上则有 Adobe Acrobat (PDF) 格式以便查看和打印。还可从 IBM 订购打印的副本；请参阅第250页的『订购打印书籍』。下表列示了可订购的书籍。

在 OS/2 和 Windows 平台上，可在 `sql1lib\doc\html` 目录下安装 HTML 文件。DB2 信息被翻译成各种语言；但是，并非所有的信息都有每一种语言的翻译版本。每当信息不能以某种特定语言表示出来时，就会提供英语信息。

在 UNIX 平台上，可在 `doc/%L/html`（其中 `%L` 表示语言环境）目录下安装多种语言版本的 HTML 文件。有关更多信息，参考适当的《快速入门》书籍。

您可以各种方法来获取 DB2 书籍并访问信息:

- 第252页的『查看联机信息』
- 第256页的『搜索联机信息』
- 第250页的『订购打印书籍』
- 第249页的『打印 PDF 书籍』

表 35. DB2 信息

名称	描述	书号	HTML 目录 PDF 文件名
<b>DB2 指南和参考信息</b>			
《管理指南》	《管理指南: 计划》提供数据库概念的概述、有关设计问题（如逻辑和物理数据库设计）的信息，以及高可用性的讨论。	SB84-0219	db2d0
	《管理指南: 实现》提供有关实现问题（如实现设计、访问数据库、审核、备份和恢复）的信息。	SB84-0218	db2d2x70
	《管理指南: 性能》提供有关数据库环境以及应用程序性能评估和调整的信息。	SB84-0243	db2d3x70
	在北美，可使用书号 SBOF-8934 来订购三卷英文版的《管理指南》。		
<i>Administrative API Reference</i>	描述 DB2 应用程序编程接口 (API) 以及您可以用来管理数据库的数据结构。此书还说明如何在应用程序中调用 API。	SC09-2947	db2b0
		db2b0x70	
《应用程序构建指南》	提供环境设置信息和关于如何在 Windows、OS/2 和基于 UNIX 的平台上编译、链接和运行 DB2 应用程序的循序渐进说明。	SB84-0220	db2ax
		db2axx70	
<i>APPC, CPI-C, and SNA Sense Codes</i>	提供关于使用“DB2 通用数据库”产品时可能遇到的 APPC、CPI-C 和 SNA 检测码的一般信息。	无书号	db2ap
	仅有 HTML 格式的版本。	db2apx70	



表 35. DB2 信息 (续)

名称	描述	书号	HTML 目录
		PDF 文件名	
<i>Application Development Guide</i>	说明如何开发使用嵌入式 SQL 或 Java (JDBC 和 SQLJ) 来访问 DB2 数据库的应用程序。讨论主题包括在分区环境或联合体系统中编写存储过程、编写用户定义函数、创建用户定义类型、使用触发器和开发应用程序。	SC09-2949 db2a0x70	db2a0
<i>CLI Guide and Reference</i>	说明如何开发使用“DB2 调用层接口”(一个与 Microsoft ODBC 规范兼容的可调用 SQL 接口)来访问 DB2 数据库的应用程序。	SC09-2950 db2l0x70	db2l0
<i>Command Reference</i>	说明如何使用“命令行处理器”，并描述可用来管理数据库的 DB2 命令。	SC09-2951 db2n0x70	db2n0
<i>Connectivity Supplement</i>	提供有关以下各项的设置和参考信息：如何将作为 DRDA 应用程序请求器的 DB2 AS/400 版、DB2 OS/390 版、DB2 MVS 版、DB2 VM 版与“DB2 通用数据库”服务器配合使用。此书还详述了如何将 DRDA 应用服务器与 DB2 Connect 应用程序请求器配合使用。  仅有 HTML 和 PDF 格式。	无书号 db2h1x70	db2h1
<i>Data Movement Utilities Guide and Reference</i>	说明如何使用 DB2 实用程序(如导入、导出、装入、自动装入程序和 DPROP)来使数据移动易于进行。	SC09-2955 db2dmx70	db2dm
《数据仓库中心管理指南》	提供有关如何使用“数据仓库中心”构建和维护数据仓库的信息。	SB84-0226 db2ddx70	db2dd
<i>Data Warehouse Center Application Integration Guide</i>	提供帮助程序员将应用程序与“数据仓库中心”和“信息目录管理器”集成的信息。	SC26-9994 db2adx70	db2ad
《DB2 Connect 用户指南》	提供 DB2 Connect 产品的概念、编程以及一般用法信息。	SB84-0221 db2c0x70	db2c0
<i>DB2 Query Patroller Administration Guide</i>	提供 DB2 Query Patroller 系统的操作概述、特定操作和管理信息以及管理图形用户界面实用程序的任务信息。	SC09-2958 db2dwx70	db2dw

表 35. DB2 信息 (续)

名称	描述	书号	HTML 目录
		PDF 文件名	
《DB2 Query Patroller 用户指南》	描述如何使用 DB2 Query Patroller 的工具和功能。	SB84-0222	db2ww
		db2wwx70	
《词汇表》	提供 DB2 及其组件中使用的术语的定义。  有 HTML 格式可用且在 <i>SQL Reference</i> 中。	无书号	db2t0
		db2t0x70	
《Image、Audio 和 Video Extenders 管理和编程》	提供有关 DB2 Extender 的一般信息，有关 Image、Audio 和 Video (IAV) Extender 的管理和配置的信息，以及有关使用 IAV Extender 进行编程的信息。它包括参考信息、诊断资料（带有消息）和样本。	SB84-0247	dmbu7
		dmbu7x70	
<i>Information Catalog Manager Administration Guide</i>	提供有关管理信息目录的指南。	SC26-9995	db2di
		db2dix70	
<i>Information Catalog Manager Programming Guide and Reference</i>	提供“信息目录管理器”的体系结构接口的定义。	SC26-9997	db2bi
		db2bix70	
《信息目录管理器用户指南》	提供有关使用“信息目录管理器”用户界面的信息。	SB84-0227	db2ai
		db2aix70	
《安装和配置补遗》	指导您了解计划、安装和设置特定于平台的 DB2 客户机。此补遗还包含关于联编、设置客户机和服务器通信、DB2 GUI 工具、DRDA AS、分布式安装、配置分布式请求和访问多机种数据源的信息。	GB84-0127	db2iy
		db2iyx70	
《消息参考》	列出由 DB2、“信息目录管理器”和“数据仓库中心”发出的消息和代码，并描述应执行的操作。  在北美，您可订购两卷英文版的《消息参考》（使用书号 SBOF-8932）。	第 1 卷 GB84-0216	db2m0
		db2m1x70	
		第 2 卷 GB84-0217	
		db2m2x70	
<i>OLAP Integration Server Administration Guide</i>	说明如何使用“OLAP 集成服务器”的“管理器”组件。	SC27-0782	n/a
		db2dpx70	

表 35. DB2 信息 (续)

名称	描述	书号	HTML 目录
		PDF 文件名	
<i>OLAP Integration Server Metaoutline User's Guide</i>	说明如何使用标准“OLAP 元轮廓”接口（而非通过使用“元轮廓辅助程序”）创建和植入 OLAP 元轮廓。	SC27-0784 db2upx70	n/a
<i>OLAP Integration Server Model User's Guide</i>	说明如何使用标准“OLAP 模型接口”（而非使用“模型辅助程序”）来创建 OLAP 模型。	SC27-0783 db2lpx70	n/a
《OLAP 安装与用户指南》	提供 OLAP Starter Kit 的配置和设置信息。	SA40-1755 db2ipx70	db2ip
《OLAP Spreadsheet Add-in 用户指南 Excel 版》	描述如何使用 Excel 电子表格程序来分析 OLAP 数据。	SA40-1756 db2epx70	db2ep
《OLAP Spreadsheet Add-in 用户指南 Lotus 1-2-3 版》	描述如何使用 Lotus 1-2-3 电子表格程序来分析 OLAP 数据。	SA40-1757 db2tpx70	db2tp
<i>Replication Guide and Reference</i>	提供随 DB2 提供的“IBM 复制”工具的计划、配置、管理和用法信息。	SC26-9920 db2e0x70	db2e0
《Spatial Extender 用户指南和参考》	提供关于 Spatial Extender 的安装、配置、管理、编程和故障诊断的信息。还提供对空间数据概念的重要描述，并提供 Spatial Extender 特定的参考资料（消息和 SQL）。	SB84-0249 db2sbx70	db2sb
《SQL 入门》	介绍 SQL 概念，并提供许多构造和任务的示例。	SB84-0223 db2y0x70	db2y0
<i>SQL Reference, 第 1 卷和第 2 卷</i>	描述 SQL 语法、语义和语言规则。此书还包括关于发行版间的不兼容性、产品限制和目录视图的信息。  在北美，可使用书号 SBOF-8933 来订购两卷英文版的 <i>SQL Reference</i> 。	第 1 卷 SC09-2974 db2s1x70  第 2 卷 SC09-2975 db2s2x70	db2s0   卷
<i>System Monitor Guide and Reference</i>	描述如何收集关于数据库和数据库管理程序的各种信息。此书说明如何利用信息来了解数据库活动、提高性能和确定问题的原因。	SC09-2956 db2f0x70	db2f0

表 35. DB2 信息 (续)

名称	描述	书号	HTML 目录
		PDF 文件名	
《Text Extender 管理和编程》	提供有关 DB2 Extender 的一般信息，有关 Text Extender 的管理和配置的信息，以及有关使用 Text Extender 进行编程的信息。它包括参考信息、诊断资料（带有消息）和样本。	SB84-0248	desu9
		desu9x70	
<i>Troubleshooting Guide</i>	帮助您确定错误源、从问题中恢复并向“DB2 客户服务”咨询以使用诊断工具。	GC09-2850	db2p0
		db2p0x70	
《新增内容》	描述“DB2 通用数据库的版本 7”中的新特性、函数和增强功能。	SB84-0224	db2q0
		db2q0x70	
<b>DB2 安装和配置信息</b>			
<i>DB2 Connect Enterprise Edition for OS/2 and Windows Quick Beginnings</i>	提供 OS/2 和 Windows 32 位操作系统上的 DB2 Connect 企业版的计划、迁移、安装和配置信息。此书还包含许多受支持的客户机的安装和设置信息。	GC09-2953	db2c6
		db2c6x70	
<i>DB2 Connect Enterprise Edition for UNIX Quick Beginnings</i>	提供基于 UNIX 的平台上的 DB2 Connect 企业版的计划、迁移、安装、配置和任务信息。此书还包含许多受支持的客户机的安装和设置信息。	GC09-2952	db2cy
		db2cyx70	
《DB2 Connect 个人版快速入门》	提供 OS/2 和 Windows 32 位操作系统上的 DB2 Connect 个人版的计划、迁移、安装、配置和任务信息。此书还包含所有受支持的客户机的安装和设置信息。	GB84-0212	db2c1
		db2c1x70	
<i>DB2 Connect Personal Edition Quick Beginnings for Linux</i>	在进行所有受支持的 Linux 分发时，提供“DB2 Connect 个人版”的计划、安装、迁移和配置信息。	GC09-2962	db2c4
		db2c4x70	
《DB2 Data Links Manager 快速入门》	提供“DB2 Data Links Manager AIX 版”和 Windows 32 位操作系统的计划、安装、配置和任务信息。	GB84-0211	db2z6
		db2z6x70	
《DB2 扩充企业版 UNIX 版快速入门》	提供在基于 UNIX 的平台上的 DB2 扩充企业版的计划、安装和配置信息。此书还包含许多受支持的客户机的安装和设置信息。	GB84-0209	db2v3
		db2v3x70	

表 35. DB2 信息 (续)

名称	描述	书号	HTML 目录
		<b>PDF 文件名</b>	
<i>DB2 Enterprise - Extended Edition for Windows Quick Beginnings</i>	提供 DB2 扩充企业版 Windows 32 位操作系统版的计划、安装和配置信息。此书还包含许多受支持的客户机的安装和设置信息。	GC09-2963 db2v6x70	db2v6
<i>DB2 for OS/2 Quick Beginnings</i>	提供 OS/2 操作系统上的“DB2 通用数据库”的计划、安装、迁移和配置信息。此书还包含许多受支持的客户机的安装和设置信息。	GC09-2968 db2i2x70	db2i2
《DB2 UNIX 版快速入门》	提供在基于 UNIX 的平台上的“DB2 通用数据库”的计划、安装、迁移和配置信息。此书还包含许多受支持的客户机的安装和设置信息。	GB84-0214 db2ixx70	db2ix
《DB2 Windows 版快速入门》	提供 Windows 32 位操作系统上的“DB2 通用数据库”的计划、安装、迁移和配置信息。此书还包含许多受支持的客户机的安装和设置信息。	GB84-0215 db2i6x70	db2i6
《DB2 个人版快速入门》	提供 OS/2 和 Windows 32 位操作系统上的“DB2 通用数据库个人版”的计划、安装、迁移和配置信息。	GB84-0213 db2i1x70	db2i1
<i>DB2 Personal Edition Quick Beginnings for Linux</i>	在进行所有受支持的 Linux 分发时，提供“DB2 通用数据库个人版”的计划、安装、迁移和配置信息。	GC09-2972 db2i4x70	db2i4
《DB2 Query Patroller 安装指南》	提供有关 DB2 Query Patroller 的安装信息。	GB84-0208 db2iwx70	db2iw
《DB2 仓库管理器安装指南》	提供仓库代理程序、仓库转换程序和“信息目录管理器”的安装信息。	GB84-0122 db2idx70	db2id
<b>HTML 格式的跨平台样本程序</b>			
HTML 格式的样本程序	为所有受 DB2 支持的平台上的编程语言提供 HTML 格式的样本程序。提供的样本程序仅供参考。并非所有样本都有所有编程语言的版本。HTML 样本仅当安装了“DB2 应用程序开发客户机”时才可用。  有关这些程序的更多信息，参考《应用程序构建指南》。	无书号	db2hs

表 35. DB2 信息 (续)

名称	描述	书号	HTML 目录
		PDF 文件名	
发行说明			
<i>DB2 Connect</i> 发行说明	提供 DB2 Connect 书籍中未能包括的最新信息。	参见注释 2。	db2cr
<i>DB2</i> 安装说明	提供 DB2 书籍中未能包括的最新安装特定信息。	仅在产品 CD-ROM 上提供。	
<i>DB2</i> 发行说明	提供 DB2 书籍中未能包括的、有关所有 DB2 产品和功能部件的最新信息。	参见注释 2。	db2ir

**注:**

1. 文件名第六个位置的字符 *x* 指示书籍的语言版本。例如，文件名 db2d0e70 标识英语版本的《管理指南》，而文件名 db2d0f70 标识同一本书的法语版本。下列字母用在文件名的第六个位置以指示语言版本：

语言	标识符
巴西葡萄牙语	b
保加利亚语	u
捷克语	x
丹麦语	d
荷兰语	q
英语	e
芬兰语	y
法语	f
德语	g
希腊语	a
匈牙利语	h
意大利语	i
日语	j
韩国语	k
挪威语	n
波兰语	p
葡萄牙语	v
俄语	r
简体中文	c
斯洛文尼亚语	l
西班牙语	z
瑞典语	s
繁体中文	t
土耳其语	m

2. DB2 书籍中未能包括的最新信息以 HTML 格式在“发行说明”中提供，或作为 ASCII 文件提供。在“信息中心”中和产品 CD-ROM 上都提供了 HTML 版本。要查看 ASCII 文件：
  - 在基于 UNIX 的平台上，请参阅 `Release.Notes` 文件。此文件位于 `DB2DIR/Readme/%L` 目录中，其中 `%L` 表示语言环境名，而 `DB2DIR` 表示：
    - 在 AIX 上，是 `/usr/lpp/db2_07_01`
    - 在 HP-UX、PTX、Solaris 和 Silicon Graphics IRIX 上，是 `/opt/IBMDB2/V7.1`
    - 在 Linux 上，是 `/usr/IBMDB2/V7.1`。
  - 在其它平台上，请参阅 `RELEASE.TXT` 文件。此文件在安装产品的目录中。在 OS/2 平台上，还可双击 **IBM DB2** 文件夹，然后双击**发行说明图符**。

## 打印 PDF 书籍

如果想要书籍的打印副本，则可打印 DB2 出版物 CD-ROM 上的 PDF 文件。使用 Adobe Acrobat Reader，可打印整本书籍或特定范围内的页。有关资料库中每本书的文件名，参见第242页的表35。

可从 Adobe Web 站点（网址 <http://www.adobe.com>）获取 Adobe Acrobat Reader 的最新版本。

这些 PDF 文件包括在 DB2 出版物 CD-ROM 上，文件扩展名为 PDF。要访问这些 PDF 文件：

1. 插入 DB2 出版物 CD-ROM。在基于 UNIX 的平台上，安装 DB2 出版物 CD-ROM。参考《快速入门》以了解安装过程。
2. 启动 Acrobat Reader。
3. 从下列位置之一打开期望的 PDF 文件：
  - 在 OS/2 和 Windows 平台上：  
`x:\doc\language` 目录，其中 `x` 表示 CD-ROM 驱动器而 `language` 表示两个字符的国家或地区代码，它表示您所用的语言（例如，EN 表示英语）。
  - 在基于 UNIX 的平台上：  
CD-ROM 上的 `/cdrom/doc/%L` 目录，其中 `/cdrom` 表示 CD-ROM 的安装点而 `%L` 表示期望的语言环境的名称。

还可从 CD-ROM 将 PDF 文件复制至本地或网络驱动器并从该处读取它们。

## 订购打印书籍

通过使用销售单 (SBOF) 书号, 可单独订购或成套订购已打印的 DB2 书籍 (仅限北美)。要订购书籍, 与 IBM 授权经销商或市场代表联系, 或致电 1-800-879-2755 (美国) 或 1-800-IBM-4YOU (加拿大)。还可从 Publications Web 页面 (网址为 <http://www.elink.ibm.com/pbl/pbl>) 订购这些书籍。

有两套书籍。SBOF-8935 提供了“DB2 仓库管理器”的参考和用法信息。SBOF-8931 提供了所有其他“DB2 通用数据库”产品和功能部件的参考和用法信息。每个 SBOF 的内容列示在下表中:

表 36. 订购打印书籍

SBOF 号	包括的书籍
SBOF-8931	<ul style="list-style-type: none"> <li>• 管理指南: 计划</li> <li>• 管理指南: 实现</li> <li>• 管理指南: 性能</li> <li>• Administrative API Reference</li> <li>• 应用程序构建指南</li> <li>• Application Development Guide</li> <li>• CLI Guide and Reference</li> <li>• Command Reference</li> <li>• Data Movement Utilities Guide and Reference</li> <li>• 数据仓库中心管理指南</li> <li>• Data Warehouse Center Application Integration Guide</li> <li>• DB2 Connect 用户指南</li> <li>• 安装和配置补遗</li> <li>• Image、Audio 和 Video Extenders 管理和编程</li> <li>• 消息参考, 第 1 卷和第 2 卷</li> <li>• OLAP Integration Server Administration Guide</li> <li>• OLAP Integration Server Metaoutline User's Guide</li> <li>• OLAP Integration Server Model User's Guide</li> <li>• OLAP Integration Server User's Guide</li> <li>• OLAP 安装和用户指南</li> <li>• OLAP Spreadsheet Add-in Excel 版用户指南</li> <li>• OLAP Spreadsheet Add-in Lotus 1-2-3 版用户指南</li> <li>• Replication Guide and Reference</li> <li>• Spatial Extender Administration and Programming Guide</li> <li>• SQL 入门</li> <li>• SQL Reference, 第 1 卷和第 2 卷</li> <li>• System Monitor Guide and Reference</li> <li>• Text Extender 管理和编程</li> <li>• Troubleshooting Guide</li> <li>• 新增内容</li> </ul>



表 36. 订购打印书籍 (续)

SBOF 号	包括的书籍
SBOF-8935	<ul style="list-style-type: none"> <li>• Information Catalog Manager Administration Guide</li> <li>• Query Patroller Administration Guide</li> <li>• 信息目录管理器用户指南</li> <li>• Query Patroller 用户指南</li> <li>• Information Catalog Manager Programming Guide and Reference</li> </ul>

## DB2 联机文档

### 访问联机帮助

随所有 DB2 组件都附带提供了联机帮助。下表描述了各种类型的联机帮助。

帮助类型	内容	如何访问...
命令帮助	说明命令行处理器中命令的语法。	<p>从命令行处理器，以交互方式输入： <code>? command</code></p> <p>其中 <i>command</i> 表示一个关键字或整个命令。</p> <p>例如，<code>? catalog</code> 显示所有 CATALOG 命令的帮助，而 <code>? catalog database</code> 显示 CATALOG DATABASE 命令的帮助。</p>
客户机配置辅助程序帮助	说明您可在窗口或笔记本中执行的任务。此帮助包括您需要知道的概述和先决条件信息，并描述如何使用窗口或笔记本控件。	从窗口或笔记本，单击 <b>帮助</b> 按钮或按 <b>F1</b> 键。
命令中心帮助		
控制中心帮助		
数据仓库中心帮助		
事件分析程序帮助		
信息目录管理器帮助		
卫星管理中心帮助		
脚本中心帮助		

帮助类型	内容	如何访问...
消息帮助	描述消息的起因以及您应该执行的任何操作。	<p>从命令行处理器，以交互方式输入：  <code>? XXXnnnnn</code></p> <p>其中 <code>XXXnnnnn</code> 表示有效的消息标识符。</p> <p>例如，<code>? SQL30081</code> 显示关于 <code>SQL30081</code> 消息的帮助。</p> <p>要每次查看一屏消息帮助，可输入：  <code>? XXXnnnnn   尚有</code></p> <p>要在文件中保存消息帮助，可输入：  <code>? XXXnnnnn &gt; filename.ext</code></p> <p>其中 <code>filename.ext</code> 表示想要保存消息帮助的文件。</p>
SQL 帮助	说明 SQL 语句的语法。	<p>从命令行处理器，以交互方式输入：  <code>help statement</code></p> <p>其中，<code>statement</code> 表示 SQL 语句。</p> <p>例如，<code>help SELECT</code> 显示有关 <code>SELECT</code> 语句的帮助。  <b>注：</b>在基于 UNIX 的平台上，SQL 帮助不可用。</p>
SQLSTATE 帮助	说明 SQL 状态及类代码。	<p>从命令行处理器，以交互方式输入：  <code>? sqlstate</code> 或 <code>? class code</code></p> <p>其中，<code>sqlstate</code> 表示有效的五位 SQL 状态，而 <code>class code</code> 表示该 SQL 状态的头两位。</p> <p>例如，<code>? 08003</code> 显示 <code>08003</code> SQL 状态的帮助，而 <code>? 08</code> 显示 <code>08</code> 类代码的帮助。</p>

## 查看联机信息

此产品中的书籍为超文本标记语言 (HTML) 软拷贝格式。软拷贝格式使您可搜索或浏览信息，并提供访问相关信息的超文本链接。它还使得在站点间共享资料库更容易。

可使用遵循 HTML 版本 3.2 规范的任何浏览器来查看联机书籍或样本程序。

要查看联机书籍或样本程序：

- 如果正在运行 DB2 管理工具，则使用“信息中心”。

- 从浏览器，单击**文件** → **打开页**。打开的页中包含 DB2 信息的描述和至 DB2 信息的链接：

- 在基于 UNIX 的平台上，打开以下页：

`INSTHOME/sql1lib/doc/%L/html/index.htm`

其中 %L 表示语言环境名称

- 在其它平台上，打开以下页：

`sql1lib\doc\html\index.htm`

该路径位于安装了 DB2 的驱动器上。

如果尚未安装“信息中心”，则可通过双击 **DB2 信息** 图符来打开该页。视您正在使用的系统不同，图符在主产品文件夹中或在“Windows 开始”菜单中。

### 安装 Netscape 浏览器

如果还未安装 Web 浏览器，则可从产品包装箱中的 Netscape CD-ROM 安装 Netscape。要获取如何安装它的详细指示信息，执行：

1. 插入 Netscape CD-ROM。
2. 安装 CD-ROM（仅限于在基于 UNIX 的平台上）。参考《快速入门》以了解安装过程。
3. 关于安装说明，可参考 CDNAV *nn.txt* 文件，其中 *nn* 表示两字符语言标识符。该文件位于 CD-ROM 的根目录下。

### 通过“信息中心”访问信息

“信息中心”提供对 DB2 产品信息的快速访问。在所有装有 DB2 管理工具的平台上，都提供了“信息中心”。

可通过双击“信息中心”图符来打开“信息中心”。视正在使用的系统的不同，该图符在主产品文件夹的“信息”文件夹中，或在 Windows 的**开始**菜单中。

还可通过使用工具栏和 DB2 Windows 平台上的**帮助**菜单来访问“信息中心”。

“信息中心”提供了六种类型的信息。单击适当的标签来查看提供给该类型的主题。

<b>任务</b>	可使用 DB2 执行的关键任务。
<b>参考</b>	DB2 参考信息，如关键字、命令以及 API。
<b>书籍</b>	DB2 书籍。
<b>故障诊断</b>	错误消息类别及其恢复操作。

**样本程序** 随“DB2 应用程序开发客户机”一起提供的样本程序。如果未安装“DB2 应用程序开发客户机”，则不显示此标签。

**Web** 万维网（WWW）上的 DB2 信息。要访问此信息，必须从系统连接至 Web。

当选择其中一个列表中的项时，“信息中心”启动一个查看器来显示信息。视所选择的信息种类的不同，查看器可能是系统帮助查看器、编辑器或 Web 浏览器。

“信息中心”提供了查找功能部件，因此您不用浏览这些列表就能查找特定主题。

对于全文本搜索，请遵循“信息中心”中指向**搜索 DB2 联机信息**搜索表格的超文本链接。

HTML 搜索服务器通常是自动启动的。如果 HTML 信息中的搜索不起作用，则可能必须使用下列其中一个方法来启动搜索服务器：

#### 在 Windows 上

单击开始并选择程序 → IBM DB2 → 信息 → 启动 HTML 搜索服务器。

#### 在 OS/2 上

双击 **DB2 OS/2** 版文件夹，然后双击**启动 HTML 搜索服务器**图符。

如果在搜索 HTML 信息时遇到任何其它问题，可参考发行说明。

**注：**搜索功能在 Linux、PTX 和 Silicon Graphics IRIX 环境中不可用。

## 使用 DB2 向导

向导通过让您一次一步地完成每一个任务来协助您完成特定管理任务。可通过“控制中心”和“客户机配置辅助程序”来获取向导。下表列出了这些向导并描述了它们的用途。

**注：**“创建数据库”、“创建索引”、“配置多站点更新”和“性能配置”向导对分区数据库环境可用。

向导	帮助您...	如何访问...
添加数据库	在客户机工作站上编目数据库。	从“客户机配置辅助程序”单击添加。
备份数据库	确定、创建并调度备份计划。	从“控制中心”，右键单击想要备份的数据库并选择 <b>备份 → 数据库</b> （使用向导）。

向导	帮助您...	如何访问...
配置多站点更新	配置多站点更新、分布式事务或两阶段落实。	从“控制中心”，右键单击数据库文件夹并选择多站点更新。
创建数据库	创建数据库并执行一些基本配置任务。	从“控制中心”，右键单击数据库文件夹，并选择创建 → 数据库（使用向导）。
创建表	选择基本数据类型并创建表的主键。	从“控制中心”，右键单击表图符，并选择创建 → 表（使用向导）。
创建表空间	创建新的表空间。	从“控制中心”，右键单击表空间图符，并选择创建 → 表空间（使用向导）。
创建索引	建议对于所有查询要创建和删除哪些索引。	从“控制中心”，右键单击索引图符，并选择创建 → 索引（使用向导）。
性能配置	通过更新配置参数来调整数据库性能以满足您的业务需求。	<p>从“控制中心”，右键单击想要调整的数据库并选择使用向导配置性能。</p> <p>对于分区数据库环境，从“数据库分区”视图，右键单击想要调整的首个数据库分区并选择使用向导配置性能。</p>
复原数据库	在故障之后恢复数据库。它帮助您了解要使用的备份及要重放的纪录。	从“控制中心”，右键单击想要复原的数据库并选择恢复 → 数据库（使用向导）。

## 设置文档服务器

在缺省情况下，DB2 信息安装在本地系统上。这表示需要访问 DB2 信息的每个人都必须安装相同的文件。要将 DB2 信息存储在单个位置中，执行下列步骤：

1. 将所有文件和子目录从本地系统上的 \sql1ib\doc\html 复制至 Web 服务器。每一本书都有其自己的子目录，该子目录包含构成该书的所有必需的 HTML 和 GIF 文件。确保目录结构仍相同。
2. 配置 Web 服务器以查找新位置中的文件。有关信息，可参考《安装和配置补遗》中的 NetQuestion 附录。
3. 如果正在使用“信息中心”的 Java 版本，可为所有 HTML 文件指定基本的 URL。您应将该 URL 用于书籍列表。
4. 当能够查看书籍文件时，可将经常查看的主题做成书签。您可能想把下列各页做成书签：
  - 书籍列表

- 经常使用的书籍的目录
- 经常引用的文章，如 ALTER TABLE 主题
- 搜索格式

有关如何从中央机器处理“DB2 通用数据库”联机文档文件的信息，参考《安装和配置补遗》中的 NetQuestion 附录。

## 搜索联机信息

要查找 HTML 文件中的信息，使用下列方法之一：

- 在顶部框中单击**搜索**。使用搜索格式来查找特定的主题。此功能在 Linux、PTX 和 Silicon Graphics IRIX 环境中不可用。
- 在顶部框中单击**索引**。使用索引来查找书中的特定主题。
- 显示帮助或 HTML 书籍的目录或索引，然后使用 Web 浏览器的查找功能查找书中的特定主题。
- 使用 Web 浏览器的书签功能来快速返回至特定的主题。
- 使用“信息中心”的搜索功能来查找特定的主题。请参见第253页的『通过“信息中心”访问信息』以获取详细信息。

---

## 附录F. 声明

IBM 可能未在所有国家或地区中提供本文档中讨论的产品、服务或功能部件。关于您所在区域目前可用的产品及服务的信息，请向当地的 IBM 代表咨询。任何对 IBM 产品、程序或服务的引用并不明示或默示只能使用 IBM 的产品、程序或服务。只要不侵犯 IBM 的知识产权，任何同等功能的产品、程序或服务，都可以用来代替 IBM 产品、程序或服务。然而，评估和验证任何非 IBM 产品、程序或服务均由用户自行负责。

IBM 可能已经申请或正在申请与本文档有关的各项专利权。提供本文档并不表示允许您使用这些专利。 您可以用书面方式将许可证查询寄往：

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

有关双字节 (DBCS) 信息的许可证查询，请与您的国家或地区的“IBM 知识产权部”联系，或用书面方式将查询寄往：

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106, Japan

以下段落不适用于联合王国或该条款与当地法律不一致的任何国家或地区：国际商业机器公司以“仅此状态”的基础提供此出版物，不附有任何形式的（无论是明示的还是默示的）保证，包括（但不限于）不侵犯、适销性或适用于某特定用途的默示保证或条件。一些国家或地区允许否认某些事务中的明示或默示保证，因此，此声明可能不适用于您。

本资料可能会包含技术错误或印刷错误。此处的信息会定期得到更改；这些更改将编入本出版物的新版本中。IBM 可能随时对此出版物中描述的产品和 / 或程序进行改进和 / 或更改，而不另行通知。

在此信息中对非 IBM Web 站点的任何引用仅是出于方便起见，不以任何方式提供对这些 Web 站点的保证。这些 Web 站点中的资料不是此 IBM 产品资料的一部分，使用这些 Web 站点时风险自负。

IBM 对于您以任何方式提供的任何信息，有权利以任何它认为适当的方式使用或分发，而不必对您负任何责任。

为了以下目的：(i) 允许在独立创建的程序和其他程序（包括本程序）之间进行信息交换 (ii) 允许对已经交换的信息进行相互使用，而希望获取本程序有关信息的被许可方请与以下地址联系：

IBM Canada Limited  
Office of the Lab Director  
1150 Eglinton Ave. East  
North York, Ontario  
M3C 1H7  
CANADA

只要遵守适当的条款和条件，包括某些情形下的一定数量的付款，都可获取这方面的信息。

本资料中描述的许可程序和它可用的全部许可证材料均由 IBM 根据“IBM 客户协议”、“IBM 国际程序许可证协议”或任何与客户之间的等效协议中的条款提供。

此处包含的所有性能数据都是在受控环境中确定的。因此，在其他操作环境中获得的结果可能与之相差很大。某些测量可能是在开发级的系统上进行的，不能保证这些测量方法在通用系统上同样可用。此外，某些测量方法可能是通过外推法归纳来估计的。实际结果可能会有所不同。此文档的用户应针对他们的特定环境验证数据是否适用。

关于非 IBM 产品的信息是从那些产品的供应商、他们发布的声明或其他公用来源获得的。IBM 未测试那些产品，不能确认与非 IBM 产品相关的性能、兼容性或任何其他声明的准确性。如有关于非 IBM 产品的功能的问题，应向那些产品的供应商提出。

所有关于 IBM 未来方向或意向的声明都可能随时更改或撤销，而不作任何通知，并且仅代表发展目标。

本资料中可能包含用于日常业务运作的的数据或报表的示例。为了尽可能完整地说明问题，这些示例可能包含个人、公司、品牌和产品的名称。所有这些名称都是虚构的，如与实际商业企业所使用的名称和地址相似，纯属巧合。

版权许可证：

本资料可能包含源语言的样本应用程序，它们举例说明各种操作平台上的编程技术。为了开发、使用、市场营销或分发符合编写这些样本程序所针对的操作系统的的应用程序编程接口的应用程序，您可以以任何形式复制、修改和分发这些样本



程序，而不必向 IBM 付款。尚未在所有条件下彻底测试这些示例。因此，IBM 不能保证或默示这些程序的可靠性、适用性或功能。

这些样本程序或任何派生产品的每个副本或任何部分都必须包括如下版权声明：

© (您的公司名) (年份)。本代码的某些部分是从“IBM 公司样本程序”派生的。

© Copyright IBM Corp. \_输入年份\_。 All rights reserved.

---

## 商标

以星号 (\*) 标出的下列各项是国际商业机器公司在美国和 / 或其他国家或地区的商标。

ACF/VTAM	IBM
AISPO	IMS
AIX	IMS/ESA
AIX/6000	LAN DistanceMVS
AIXwindows	MVS/ESA
AnyNet	MVS/XA
APPN	Net.Data
AS/400	OS/2
BookManager	OS/390
CICS	OS/400
C Set++	PowerPC
C/370	QBIC
DATABASE 2	QMF
DataHub	RACF
DataJoiner	RISC System/6000
DataPropagator	RS/6000
DataRefresher	S/370
DB2	SP
DB2 Connect	SQL/DS
DB2 Extender	SQL/400
DB2 OLAP Server	System/370
DB2 Universal Database	System/390
Distributed Relational	SystemView
Database Architecture	VisualAge
DRDA	VM/ESA
eNetwork	VSE/ESA
Extended Services	VTAM
FFST	WebExplorer
First Failure Support Technology	WIN-OS/2

下列各项是其他公司的商标或注册商标:

Microsoft、Windows 和 Windows NT 是 Microsoft Corporation 的商标或注册商标。

Java 或所有基于 Java 的商标和徽标以及 Solaris 是 Sun Microsystems, Inc. 在美国和 / 或其他国家或地区的商标。

Tivoli 和 NetView 是 Tivoli Systems Inc. 在美国和 / 或其他国家或地区的商标。

UNIX 是经 X/Open Company Limited 唯一许可的在美国和 / 或其他国家或地区的注册商标。

以双星号 (\*\*) 标出的其他公司、产品或服务名称, 可能是其他公司的商标或服务标记。

# 索引

## [ A ]

安全性 23  
    设计隐含 79  
安装  
    Netscape 浏览器 253

## [ B ]

版本恢复 22  
版本 6 的不兼容性  
    当前解释方式 197  
    对层次结构拥有 SELECT 特权 197  
    非双精度 SQLVAR 中的 SQLNAME 194  
    关联代码 189  
    过时的配置关键字 194  
    过时的数据库配置参数 198  
    事件监视器输出流格式 194  
    字符名大小 192  
    DATALINK 列 195  
    FOR UPDATE 语法 192  
    Java 编程 191  
    OBJCAT 视图 189  
    PC/IXF 格式更改 193  
    RUMBA 198  
    SYSFUN 字符串函数特征 195  
    SYSIBM 基本目录 190  
    SYSTABLE 列更改 196  
    USING 和 SORT BUFFER 198  
    VARCHAR 数据类型 191  
备份 22  
    备份数据库向导 254  
本地语言支持 (NLS)  
    日期时间值 226  
    双向 CCSID 支持 221  
    字符集 220  
崩溃恢复 22  
变换器步骤 48

标识列  
    概述 67  
标识入选键列 66  
表  
    并置 96  
    重组 23  
    概述 9  
    估计大小需求 83  
    规范化 68  
    检查约束 78  
    系统目录 84  
    映射至表空间 108  
    用户 84  
表空间  
    概述 12  
    工作负注意事项 112  
    临时 99, 110  
    目录 99, 111  
    设计 97, 106  
    输入和输出 (I/O) 注意事项 106  
    数据库管理空间 (DMS) 104  
    系统管理空间 (SMS) 100  
    选择 97  
    选择 SMS 或 DMS 113  
    映射至缓冲池 107  
    映射至节点组 108  
    用户 99  
    SYSCATSPACE 99  
    TEMPSPACE1 99  
    USERSPACE1 99  
并行处理能力  
    概述 79  
并行性  
    查询 33  
    分区间 34  
    分区内 33  
    概述 31  
    和不同的硬件环境 43  
    和索引创建 36  
    类型 33  
    实用程序 36

并行性 (续)  
    数据库备份和复原实用程序 36  
    “装入”实用程序 36  
    “自动装入”实用程序 36  
I/O 33  
并置  
    表 96  
不兼容性  
    版本 6 185  
    版本 7 181  
    创建数据库 196  
    计划的 180  
    列不匹配 188  
    列数据类型更改为 BIGINT 188  
    描述 179  
    外键列名 186  
    只读视图 (计划的) 180  
    主键列名 186  
    COLNAMES (计划的) 180  
    FK\_COLNAMES (计划的) 180  
    PK\_COLNAMES (计划的) 180  
    SYSCAT.CHECKS 列 TEXT 188  
    SYSCAT.INDEXES 列 COLNAMES 187  
    SYSCAT.STATEMENTS 列 TEXT 187  
    SYSCAT.VIEWS 列 TEXT 186  
不确定事务  
    恢复 130, 136  
    手工恢复 143  
    再同步 130  
部分撤消群集 93  
步骤 (在入库中) 46

## [ C ]

参考循环  
    定义 75  
仓库 45  
    代理程序 46  
    过程 47

- 仓库 (续)
  - 目标 46
  - 源 46
- 仓库步骤 46
  - 变换器 48
  - 程序 47
  - 用户定义程序 48
  - SQL 47
- 插入规则 76
- 查看
  - 联机信息 252
- 查询并行性 33
- 查询间并行性 33
- 查询内并行性 33
- 长整数字段数据
  - 估计大小需求 86
- 撤消群集
  - 部分 93
- 程序步骤 47
- 重组表 23
- 触发器 21
  - 概述 78
- 创建表空间向导 255
- 创建表向导 255
- 创建数据库向导 255
- 从属表 75
- 从属行 75
- 存储器对象
  - 表空间 12
  - 概述 12
  - 缓冲池 16
  - 容器 15

## [ D ]

- 打印 PDF 书籍 249
- 大对象 (LOB)
  - 列定义 63
- 大对象 (LOB) 数据
  - 估计大小需求 86
- 大小需求, 估计
  - 表 83
  - 临时工作空间 91
- 代理程序
  - 仓库 46
- 代理站点 46

- 代码页
  - 受支持的 Windows 代码页 219
  - DB2CODEPAGE 注册表变量 219
- 单处理器环境 37
- 单分区
  - 单处理器环境 37
  - 多处理器环境 38
- 第二种范式 69
- 地理编码 53
- 第三种范式 71
- 第四种范式 72
- 第一种范式 69
- 定义表列 63
- 多分区节点组 31
- 多分区配置 39
- 多媒体对象 60
- 多站点更新 118, 120
  - 访问 DB2 UDB 服务器的主机或 AS/400 应用程序 125

## [ F ]

- 发行版间的不兼容性
  - 描述 179
- 发行说明 249
- 分布式请求 27
- 分布式事务处理
  - 安全性注意事项 145
  - 错误处理 143
  - 对主机和 AS/400 数据库服务器的支持 142
  - 配置注意事项 146
  - 事务管理器 135
  - 数据库连接注意事项 142
  - 应用程序 134
  - 资源管理器 136
  - RELEASE 语句 142
- 分布式数据库 117
- 分割镜像
  - 作为备份映像 165
  - 作为备用数据库 164
- 分割镜像处理 163
- 分区
  - 键 94
  - 数据 92
  - 数据库 31

- 分区 (续)
  - 映射 93
- 分区间并行性 34
  - 配合分区内并行性使用 35
- 分区兼容性 96
- 分区内并行性 33
  - 配合分区间并行性使用 35
- 分区数据库 31
- 父代
  - 表 75
  - 行 75
  - 键 74, 75
- 复原向导 255
- 复制的摘要表 97

## [ G ]

- 高可用性 161
- 根类型 64
- 更新规则 77
- 工作单元 117
  - 远程 117
- 估计大小需求
  - 长整数字段数据 86
  - 大对象 (LOB) 数据 86
  - 日志文件空间 90
  - 索引空间 87
- 故障恢复支持 161
  - 空闲备用 162
  - 相互接管 163
- 关系
  - 多对多 62
  - 多对一 61
  - 一对多 61
  - 一对一 63
- 关系数据库概念
  - 概述 7

## [ H ]

- 缓冲池
  - 概述 16
  - IBMDEFAULTBP 107
- 恢复 22
  - 版本 22
  - 崩溃 22

恢复 (续)

前滚 22

## [ J ]

基于 UNIX 的平台

语言环境目录 215

兼容性

分区 96

检查约束 21

键

分区 94

概述 65

键列

标识 66

节点

数据位置, 确定 93

节点组 31

概述 9

设计 91

IBMCATGROUP 99

IBMDEFAULTGROUP 99

IBMTEMPGROUP 99

结构化类型 64, 79

进程 (在入库中) 47

具有多个处理器的分区 40

具有一个处理器的分区 39

## [ K ]

可伸缩性 36

克隆数据库

创建 164

空间

数据 53

信息 51

空值 65

控制中心

运行翻译版本 218

## [ L ]

类型表 79

概述 64

类型层次结构 64, 79

类型视图

概述 64

历史数据

概述 79

联合数据库

区分大小写的名称 172

认证 25

设计注意事项 115

授权 25

系统 27

整理顺序 225

联机帮助 251

联机信息

查看 252

搜索 256

连接缓冲, MTS 156

连接路径 68

两阶段提交 118, 120

错误处理 129

过程 126

列

为表定义 63

临时表空间 99

建议 110

临时工作空间

估计大小需求 91

逻辑数据库分区 41

逻辑数据库设计

定义表 61

关系 61

决定要记录的数据 59

## [ M ]

命名约定

一般 169

模式

概述 11

目标

表 64

行 64

类型 64

视图 64

目录

数据库 81

目录表空间 99

目录表空间 (续)

建议 111

## [ P ]

排序序列 225

配置

多分区 39

配置参数

概述 17

DB2 事务管理器注意事项 123

配置多站点更新向导 254

## [ Q ]

迁移

数据库 175

前滚恢复 22

区分大小写的名称, 联合数据库 172

权限级 24

缺省代理站点 46

## [ R ]

任务

入库 48

认证 23

联合数据库概述 25

日期

定义 227

格式 229

日期时间值

概述 226

字符串表示 227

日期字符串

定义 228

日志文件空间

估计大小需求 90

日志文件系统 161

容量 36

容器

概述 15

添加至 DMS 表空间 105

入库

对象 45

概述 45

- 入库 (续)
  - 任务 48
- 入库概述 45
  
- [ S ]**
- 删除规则 76
- 商业规则
  - 概述 19
- 商业元数据 48
- 设计
  - 表空间 106
  - 联合数据库 115
- 设计和选择表空间 97
- 设置文档服务器 255
- 审计活动
  - 概述 78
- 时间
  - 定义 227
  - 格式 229
- 时间戳记
  - 定义 227
- 时间戳记字符串
  - 定义 229
- 时间字符串
  - 定义 228
- 实例
  - 概述 8
- 实体
  - 在数据库中 59
- 实体的具体值 59
- 实用程序并行性 36
- 使表规范化 68
- 试探性操作 144
- 视图
  - 概述 9
- 事务
  - 访问分区数据库 143
  - 非 XA 135
  - 紧密耦合 135
  - 两阶段提交 135
  - 全局 135
  - 松散耦合 135
- 事务处理
  - 配置 XA 事务管理器 150
- 事务管理器 120, 121

- 事务管理器 (续)
  - BEA Tuxedo 152
  - IBM TXSeries CICS 150
  - IBM TXSeries Encina 150
    - 从 Encina 应用程序引用 DB2 数据库 151
    - 配置 DB2 150
    - 为每个资源管理器配置 Encina 151
  - Microsoft 事务处理服务器 154
- 授权 24
  - 概述 78
  - 联合数据库概述 25
- 受支持
  - 用于 MTS 协调的事物的 DB2 数据库服务器 155
- 书籍 241, 250
- 输入和输出 (I/O) 注意事项
  - 表空间 106
- 数据
  - 长整数字段 86
  - 大对象 (LOB) 86
  - 分区 92
  - 运行 45
  - 资料式 45
- 数据仓库中心 45
- 数据库
  - 分布式 117
  - 概述 9
  - 目录 81
  - 文件 82
  - 在单个事务中使用数个数据库 118
  - 主机系统上 119
- 数据库对象
  - 表 9
    - 概述 7
    - 节点组 9
    - 命名规则 220
    - 模式 11
    - 实例 8
    - 视图 9
  - 数据库 9
  - 索引 10
  - 系统目录表 11
- 数据库分区 31

- 数据库管理空间 (DMS) 12, 104
- 数据库管理器配置参数
  - 概述 17
- 数据库配置参数
  - 概述 18
- 数据库迁移 175
- 数据库设计
  - 逻辑 59
  - 物理 81
- 数据库系统
  - 联合 27
- 数据块大小 16
  - 定义 98
  - 选择 109
- 数据类型 79
- 数据源 27
- 属性 59
- 属性数据 52
- 双向 CCSID 支持
  - CCSID 表 221
  - DB2 Connect 实现 223
  - DB2 UDB 实现 222
- 搜索
  - 联机信息 254, 256
- 索引
  - 概述 10
  - 索引键 10
  - 索引空间
    - 估计大小需求 87
  - 索引向导 255

## **[ T ]**

- 泰国字符
  - 排序 226
- 特权 24
- 提交
  - 两阶段 126
  - 两阶段期间的错误 129
- 添加数据库向导 254, 255
- 同步点管理器 (SPM) 121
- 图形信息系统 (GIS) 51

## [ W ]

- 外键
  - 定义 74
- 外键约束 20
- 唯一的
  - 键 74
  - 索引 10
- 唯一键
  - 概述 65
- 唯一约束 19
  - 概述 73
- 文件系统
  - 日志 161
- 文件, 数据库 82
- 物理数据库设计 81
- 物理文件
  - SMS 103

## [ X ]

- 系统对象
  - 概述 17
  - 配置参数 17
- 系统管理空间 (SMS) 12, 100
- 系统记录设施
  - XA 接口示例 149
- 系统临时表空间 99
- 系统目录表
  - 概述 11
  - 估计初始大小 84
- 系统网络体系结构 (SNA) 125
- 向导
  - 备份数据库 254
  - 创建表 255
  - 创建表空间 255
  - 创建数据库 255
  - 复原数据库 255
  - 配置多站点更新 254
  - 索引 255
  - 添加数据库 254, 255
  - 完成任务 254
  - 性能配置 255
- 协调程序节点 31
- 信息目录 48
- 信息中心 253

- 星形模式 48
- 性能配置向导 255
- 选择数据块大小 109

## [ Y ]

- 样本程序
  - 跨平台 247
  - HTML 247
- 引用类型
  - 概述 64, 79
- 引用约束
  - 概述 74
  - 删除连接的关系 77
  - SQL 操作的意义 76
  - SQL DELETE 规则 76
  - SQL INSERT 规则 76
  - SQL UPDATE 规则 77
- 应用程序编程
  - 整理顺序, 准则 225
- 硬件环境 36
  - 并行性类型 43
  - 单分区, 单处理器 37
  - 单分区, 多个处理器 38
  - 具有多个处理器的分区 40
  - 具有一个处理器的分区 39
  - 逻辑数据库分区 41
- 映射
  - 表空间至缓冲池 107
  - 表空间至节点组 108
  - 表至表空间 108
  - 分区 93
- 用户表
  - 页限制 84
- 用户表空间 99
- 用户定义程序步骤 48
- 用户定义函数 (UDF)
  - 概述 64
- 用户定义类型 (UDT)
  - 列定义 63
- 用户临时表空间 99
- 语言标识符
  - 书籍 248
- 语言环境
  - 管理服务器和实例之间的兼容性 215

- 语言环境目录
  - 基于 UNIX 的平台 215
- 元数据 48
- 远程工作单元 117
- 约束
  - 概述 73
  - 检查 21
  - 外键 20
  - 唯一的 19
  - 主键 19
  - NOT NULL 19
- 运行数据 45

## [ Z ]

- 暂挂 I/O
  - 支持持续可用性 163
- 摘要表
  - 复制的 97
  - 概述 79
- 整理顺序
  - 联合数据库注意事项 225
  - 泰国字符 226
  - 一般注意事项 225
  - 整理顺序选项 226
- 主键
  - 定义 74
  - 概述 65
  - 生成唯一列 67
- 主键约束 19, 75
- 主索引 65
- 主题区 45
- 资料式数据 45
- 资源管理器
  - 将数据库设置成 137
- 子代表 75
- 子类型 64
- 自参考
  - 表 75
  - 行 75
  - 约束 75
- 组合键 66, 75
- 最新信息 249
- 作用域
  - 引用类型 64
- 坐标系 53

## [ 特别字符 ]

“首个合适”次序 84

## C

CCSID 支持, 双向  
    CCSID 表 221  
    DB2 Connect 实现 223  
    DB2 UDB 实现 222

## D

DB2 同步点管理器 (SPM) 125  
DB2 资料库  
    查看联机信息 252  
    打印 PDF 书籍 249  
    订购打印书籍 250  
    结构 241  
    联机帮助 251  
    设置文档服务器 255  
    书籍 241  
    书籍的语言标识符 248  
    搜索联机信息 256  
    向导 254  
    信息中心 253  
    最新信息 249  
DB2 Connect  
    用于, 用于多站点数据库更新  
        119  
DB2CODEPAGE 注册表变量 219  
db2imidb 工具 163  
DMS (数据库管理空间) 12, 104  
DMS 表空间  
    添加容器 105  
DTP (分布式事务处理) 134

## G

GIS (地理信息系统) 51

## H

HTML  
    样本程序 247

## I

IBMCATGROUP 99  
IBMDEFAULTGROUP 99  
IBMTEMPGROUP 99  
I/O 并行性 33

## L

LIST INDOUBT TRANSACTIONS 命  
    令 143  
LOB (大对象)  
    列定义 63  
LOB (大对象) 数据  
    估计大小需求 86

## M

Microsoft 事务处理服务器  
    安装和配置 154  
    调整 TCP/IP 通信 158  
    连接缓冲 156  
    软件的先决条件 154  
    使用样本应用程序测试 DB2 158  
    使用 ADO 2.1 和更新版本的连接  
        缓冲 157  
    事务超时和 DB2 连接特性 155  
    受支持的 DB2 数据库服务器  
        155  
    验证安装 155  
    再次使用 ODBC 连接 157  
    在 DB2 中启用支持 154  
MPP 环境 39  
MTS 协调的事务  
    受支持的 DB2 数据库服务器  
        155

## N

Netscape 浏览器  
    安装 253  
NOT NULL 约束 19

## P

PDF 249

## R

RAID 设备  
    优化性能 114

## S

SmartGuide  
    向导 254  
SMP 环境 38  
SMP 群集环境 40  
SMS (系统管理空间) 12, 100  
SMS 物理文件 103  
SNA (系统网络体系结构) 125  
Spatial Extender  
    概述 51  
SPM (同步点管理器) 121  
SQL 步骤 47  
SQL 优化器 10  
SYSCATSPACE 99

## T

TEMPSPACE1 99  
TPM 值 139  
TP\_MON\_NAME 值 139

## U

UDF (用户定义函数)  
    概述 64  
USERSPACE1 99

## W

Windows  
    受支持的代码页 219  
    DB2CODEPAGE 注册表变量 219

## X

XA 事务管理器  
    正在配置 150  
xa\_open 字符串 137  
X/Open 事务管理器接口 (XA)  
    分布式事务处理 (DTP) 134



---

## 与 IBM 联系

如果有技术问题，请在与“DB2 客户支持中心”联系之前复查并执行 *Troubleshooting Guide* 所建议的操作。本指南提出了一些建议，指导您收集一些信息从而帮助“DB2 客户支持中心”更好地为您服务。

要获取信息或订购任何“DB2 通用数据库”产品，与当地分支机构的 IBM 代表联系或与任何授权的 IBM 软件经销商联系。

您如果住在美国，请致电下列其中一个号码：

- 1-800-237-5511，可获得客户支持
- 1-888-426-4343，可了解所提供的服务项目

---

## 产品信息

您如果住在美国，请致电下列其中一个号码：

- 1-800-IBM-CALL (1-800-426-2255) 或 1-800-3IBM-OS2 (1-800-342-6672)，可订购产品或获取一般信息。
- 1-800-879-2755，可订购出版物。

**<http://www.ibm.com/software/data/>**

DB2 万维网网页提供关于新闻、产品描述和培训计划等等的当前 DB2 信息。

**<http://www.ibm.com/software/data/db2/library/>**

“DB2 产品和服务技术库”可供您访问常见问题、修正、书籍以及最新的 DB2 技术信息。

注：此信息可能只有英文版。

**<http://www.elink.ibm.com/pbl/pbl/>**

“国际出版物” Web 订购站点提供关于如何订购书籍的信息。

**<http://www.ibm.com/education/certify/>**

IBM Web 站点中的“专业证书程序”提供各种 IBM 产品（包括 DB2）的证书测试信息。

**<ftp://software.ibm.com>**

以匿名形式登录。可在目录 /ps/products/db2 中找到有关 DB2 和许多其他产品的演示、修正、信息和工具。

**comp.databases.ibm-db2, bit.listserv.db2-l**

这些因特网新闻组可供用户来讨论使用 DB2 产品的经验。

**On Compuserve: GO IBMDB2**

输入此命令来访问 IBM DB2 系列论坛。这些论坛支持所有的 DB2 产品。

有关如何在美国以外的地区与 IBM 联系的信息，请参阅 *IBM Software Support Handbook* 的附录 A。要访问此文档，访问以下 Web 页面：<http://www.ibm.com/support/>，然后选择该页面底部附近的 IBM Software Support Handbook 链接。

**注：**在某些国家或地区，IBM 授权的经销商应与他们的经销商支持机构联系，而不是与“IBM 支持中心”联系。





SB84-0219-01



Spine information:



IBM® DB2® 通用数据库 《管理指南: 计划》

版本 7