

<<循序渐进DB2>>

图书基本信息

书名：<<循序渐进DB2>>

13位ISBN编号：9787302201380

10位ISBN编号：7302201382

出版时间：2009-5

出版时间：清华大学

作者：牛新庄

页数：653

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<循序渐进DB2>>

内容概要

《循序渐进DB2：DBA系统管理、运维与应用案例》DB2数据库是IBM公司关系型数据库核心产品，在国内以及全球有着广泛的应用。

针对DB2初学者，本书循序渐进地把DB2所涉及的众多概念介绍给大家。

客户端连通性、实例、数据库、表空间和缓冲池、数据移动、备份恢复、故障诊断、锁与并发，以及数据库安全都是本书关注的重点。

在介绍这些DB2对象和概念的同时，作者尽可能从DBA日常工作的角度探究DB2数据库常规维护工作。

本书同时还就表、索引、序列、触发器等数据库对象从应用设计的角度进行了介绍。

本书适合DB2的初学者、DB2开发人员、准备参加DB2认证考试的读者以及DB2数据库管理人员学习和阅读。

<<循序渐进DB2>>

作者简介

牛新庄博士（数据库维护、优化和架构专家）拥有DB2 V5、V6、V7、V8和V9全部认证，同时他还拥有OCP、AIX、HP-UX、MQ、TSM和WebSphere等20多项国际认证。

牛新庄博士是IBM官方资深培训讲师(培训DB2、AIX、MQ、WebSphere、TSM和CICS)，是中信银行、山东农信、青岛海尔等公司的资深技术顾问，曾经帮助工农中建招交六大行、联想集团、青岛海尔、云南红塔、江苏电力公司等国内很多企业做过问题诊断、性能调优和技术支持，具有丰富的理论和实践结合经验。

2004年和2005年分别在上海北京成立咨询顾问公司，2008年以年薪217万被中国建设银行总行外聘为资深技术专家。

牛新庄博士拥有的奖项有：国内数据库领域最高荣誉的“2006年中国首届杰出数据库工程师”奖，首届IBM杰出软件专家奖，“2006年IT 168技术卓越”奖。

<<循序渐进DB2>>

书籍目录

- 第1章 DB2安装配置 11.1 DB2数据库概述 11.1.1 DB2发展历史 11.1.2 DB2版本和平台支持 71.1.3
DB2产品组件和功能 101.2 DB2数据库安装配置 131.2.1 DB2在Windows上的安装 141.2.2 DB2
在Linux/UNIX上的安装 211.3 DB2数据库体系结构 23第2章 创建实例和管理服务器 312.1 实例
312.1.1 实例概念 312.1.2 创建实例 322.1.3 实例目录 352.1.4 实例相关命令 392.1.5 DB2INSTANCE
变量介绍 432.1.6 删除实例 442.1.7 配置实例 452.2 管理服务器 452.2.1 管理服务器概念 452.2.2 创
建管理服务器 472.2.3 管理服务器相关命令 482.2.4 删除DB2管理服务器 492.2.5 配置管理服务器 49
第3章 创建数据库和表空间 513.1 创建数据库 513.1.1 DB2数据库存储模型 533.1.2 表空间管理类型
553.1.3 创建数据库 583.1.4 数据库目录 703.2 表空间设计 733.2.1 创建表空间 733.2.2 表空间维
护 763.2.3 表空间设计注意事项 833.2.4 prefetchsize大小选择 893.2.5 文件系统(CIO/DIO)和裸设备
903.2.6 OVERHEAD和TRANSFERRATE设置 933.2.7 优化RAID设备上表空间性能 933.2.8 合理设置
系统临时表空间 953.3 缓冲池 963.3.1 缓冲池的使用方法 973.3.2 缓冲池和表空间之间关系 973.3.3
缓冲池维护 983.3.4 缓冲池设计原则 1013.4 本章小结 104第4章 访问数据库 1054.1 访问DB2 1054.2
DB2图形化操作环境 1064.3 DB2 CLP处理程序 1154.3.1 DB2 CLP简介 1154.3.2 DB2 CLP设计
1154.3.3 DB2 CLP命令选项 1174.3.4 设置DB2_CLPPROMPT定制DB2 CLP 1214.4 配置DB2服务器的
TCP/IP通信 1264.4.1 在服务器上更新services文件 1274.4.2 在服务器上更新数据库管理器配置文件
1274.4.3 设置DB2服务器的通信协议 1284.4.4 查看服务器通信端口状态 1284.4.5 使用控制中心配
置DB2服务器通信 1294.5 配置客户机至服务器通信 1304.5.1 客户机至服务器通信概述 1304.5.2 使
用控制中心配置客户端通信 1304.5.3 使用CA配置客户机到服务器通信 1314.5.4 深入了解DB2节点目
录、数据库目录 1374.5.5 使用CLP配置客户机到服务器通信案例 1434.6 本章小结 147第5章 创建数
据库对象 1495.1 模式 1495.1.1 模式概念 1495.1.2 系统模式 1515.1.3 设置和获得当前模式 1515.1.4
模式和用户的区别 1525.2 表设计考虑 1535.2.1 选择合适的数据类型 1535.2.2 选择合适约束类
型 1565.2.3 使用not null with default 1595.2.4 生成列及应用案例 1595.2.5 自动编号和标识列应用案例
1605.2.6 使用not logged initially特性 1615.2.7 使用append on特性 1625.2.8 数据、索引和大对象分开
存放 1625.2.9 设置pctfree 1635.2.10 表的locksize 1635.2.11 表的volatile特性 1635.2.12 创建带XML列
的表 1645.2.13 表维护相关命令 1655.2.14 表设计高级选项 1695.3 索引设计 1735.3.1 索引优点
1735.3.2 索引类型 1745.3.3 索引结构 1775.3.4 理解索引访问机制 1805.3.5 创建集群索引 1825.3.6
创建双向索引 1835.3.7 完全索引访问(index access only) 1845.3.8 创建索引示例 1855.3.9 索引总结
1915.4 使用序列提高性能 1945.4.1 应用程序性能和序列 1945.4.2 设计序列原则 1955.4.3 序列维护
1965.4.4 比较序列与标识列 2005.5 视图 2025.5.1 视图类型 2025.5.2 创建with check option视图
2065.5.3 视图维护 2075.6 表表达式 2085.6.1 嵌套的表表达式 2085.6.2 公用表表达式 2095.7 触
发器设计 2105.7.1 触发器的类型 2105.7.2 创建触发器示例 2125.7.3 触发器设计总结 2145.8 本章小结
216第6章 数据移动 2176.1 数据移动格式 2176.1.1 定界ASCII文件格式 2186.1.2 非定界ASCII文件
格式 2186.1.3 PC/IXF文件格式 2196.1.4 工作表文件格式 2196.1.5 游标 2196.2 EXPORT 2206.2.1
EXPORT概述 2206.2.2 导出数据 2206.2.3 导出数据示例 2236.3 IMPORT 2246.3.1 IMPORT概述
2246.3.2 导入数据 2246.3.3 导入数据示例 2296.4 LOAD 2316.4.1 LOAD概述 2316.4.2 装入数据
2326.4.3 装入示例 2406.4.4 在线LOAD 2446.4.5 监控LOAD进度 2476.4.6 LOAD期间和之后的表空
间状态 2486.4.7 使用CURSOR文件类型来移动数据 2526.4.8 提高LOAD性能 2536.4.9 LOAD失败恢
复 2586.4.10 LOAD和IMPORT比较 2606.5 数据移动性能问题 2626.6 DB2MOVE和DB2LOOK
2636.6.1 数据库移动工具——DB2MOVE 2636.6.2 DB2 DDL提取工具(DB2LOOK) 2656.6.3 利
用DB2MOVE和DB2LOOK移动数据案例 2666.6.4 带COPY操作的DB2MOVE实用程序 2696.7 本章小
结 275第7章 数据库备份与恢复 2777.1 恢复概念 2777.1.1 崩溃恢复 2817.1.2 灾难恢复 2827.1.3 版
本恢复 2827.1.4 前滚恢复 2837.2 DB2日志 2857.2.1 日志文件的使用 2857.2.2 日志类型 2877.2.3 日
志相关配置参数 2907.2.4 数据库日志总结 2917.3 数据库和表空间备份 2937.3.1 数据库备份 2937.3.2
表空间备份 2957.3.3 增量备份 2957.3.4 检查备份完整性——db2ckbkp 2987.4 数据库和表空间恢
复 3007.4.1 数据库恢复 3007.4.2 表空间恢复 3027.4.3 增量恢复 3037.4.4 增量恢复检查——db2ckrst

<<循序渐进DB2>>

3047.4.5 重定向恢复 3057.4.6 恢复已drop的表 3097.5 数据库和表空间前滚 3127.5.1 数据库前滚
 3127.5.2 表空间前滚 3147.6 RECOVER实用程序 3177.7 恢复历史文件 3217.8 数据库重建 3247.8.1
 数据库重建概念 3247.8.2 使用表空间备份重建可恢复数据库 3247.8.3 只使用部分表空间备份重建
 可恢复数据库 3277.8.4 使用包含日志文件的在线备份重建数据库 3297.8.5 使用增量备份镜像重建可
 恢复数据库 3307.8.6 使用重定向选项重建可恢复数据库 3307.8.7 重建不可恢复数据库 3317.8.8 数据
 库重建的限制 3317.9 监控备份、复原和恢复进度 3327.10 备份、恢复和复原期间表空间状态 3337.11
 优化备份、复原和恢复性能 3337.12 备份恢复最佳实践 335第8章 DB2故障诊断 3378.1 DB2故障
 诊断机制 3378.1.1 故障诊断相关文件 3378.1.2 收集故障诊断信息 3428.1.3 设置故障诊断级别 3438.2
 深入讲解故障诊断文件 3458.2.1 解释管理通知日志文件条目 3458.2.2 解释诊断日志文件条目
 3468.3 故障诊断工具 3498.3.1 使用db2support收集环境信息 3498.3.2 db2ls和db2level 3508.3.3 使
 用db2diag分析db2diag.log文件 3518.3.4 db2pd 3548.3.5 DB2内部返回码 3558.4 故障诊断分析流程
 3568.4.1 故障诊断流程 3568.4.2 结合系统事件判断 3598.4.3 结合系统运行状况诊断 3608.5 本章小
 结 361第9章 DB2性能监控 3639.1 监控工具概述 3639.2 快照监视器 3659.2.1 快照监视器概述
 3659.2.2 利用表函数监控 3709.2.3 性能管理视图 3739.3 快照监视器案例 3749.3.1 监控案例1—动
 态SQL语句 3749.3.2 监控案例2—通过表函数监控 3769.3.3 编写快照监控脚本 3789.4 db2pd及监控
 案例 3809.5 事件监视器及监控案例 3909.6 db2mtrk及监控案例 3959.7 活动监视器 3979.8 DB2性能
 监控总结 398第10章 锁和并发 39910.1 锁的概念 39910.1.1 数据一致性 39910.1.2 事务和事务边界
 40010.1.3 锁的概念 40210.2 锁的属性、策略及模式 40710.2.1 锁的属性 40710.2.2 加锁策略
 40710.2.3 锁的模式 40810.2.4 如何获取锁 41010.2.5 锁的兼容性 41210.3 隔离级别(Isolation Levels)
 41310.3.1 可重复读(RR—Repeatable Read) 41310.3.2 读稳定性(RS—Read Stability) 41410.3.3 游标稳定
 性(CS—Cursor Stability) 41610.3.4 未提交读(UR—Uncommitted Read) 41710.3.5 隔离级别的摘要
 41910.4 锁转换、锁等待、锁升级和死锁 42110.4.1 锁转换及调整案例 42110.4.2 锁升级及调整案例
 42310.4.3 锁等待及调整案例 42610.4.4 死锁及调整案例 42910.5 锁相关的性能问题总结 43210.6 锁
 与应用程序设计 43410.7 锁监控工具 43710.8 最大化并发性 44110.8.1 选择合适的隔离级别 44110.8.2
 尽量避免锁等待、锁升级和死锁 44210.8.3 设置合理的注册变量 44210.9 锁和并发总结 450第11章
 数据库运行维护 45111.1 统计信息更新 45111.1.1 统计信息的重要性 45111.1.2 使用RUNSTATS收
 集统计信息的原则 45511.1.3 减小RUNSTATS对系统性能影响的策略 45711.1.4 DB2自动统计信息收
 集 45811.2 Runstats更新举例 46111.2.1 RUNSTATS更新示例 46111.2.2 收集分布式统计信息 46211.2.3
 包含频率和分位数统计信息的RUNSTATS 46311.2.4 包含列组统计信息的RUNSTATS 46511.2.5 包
 含LIKE STATISTICS的RUNSTATS 46511.2.6 包含统计信息配置文件的RUNSTATS 46611.2.7 带有抽样的
 的RUNSTATS 46711.2.8 带有系统页级抽样的RUNSTATS 46711.2.9 收集统计信息的其他可供选择的
 方法 46811.2.10 RUNSTATS总结 46911.3 表和索引碎片整理 47011.3.1 表重组(REORG) 47011.3.2
 索引重组 47811.3.3 确定何时重组表和索引 48011.3.4 重组表和索引的成本 48411.3.5 合理设计以减
 少碎片生成 48511.3.6 启用表和索引的自动重组 48611.4 碎片整理案例 48711.4.1 执行表、索引检查
 是否需要做REORG 48711.4.2 表和索引碎片整理 48811.5 案例：生成碎片检查、统计信息更新、碎
 片整理和REBIND脚本 48911.6 重新绑定程序包 49011.7 数据库运行维护总结 491第12章 数据库常
 用工具 49312.1 解释工具 49312.1.1 Visual Explain(可视化解释) 49312.1.2 db2expln 50112.1.3
 db2exfmt 50312.1.4 各种解释工具比较 50512.1.5 如何从解释信息中获取有价值的建议 50512.2 索
 引设计工具(db2advis) 50612.2.1 DB2 Design Advisor(db2advis) 50612.2.2 DB2 Design Advisor(db2advis)
 案例讲解 50812.3 基准测试工具db2batch 51012.3.1 db2batch 51012.3.2 db2batch基准程序测试分析例
 51212.4 数据一致性检查工具 51312.4.1 db2dart及案例 51312.4.2 inspect及案例 51412.5 db2look
 51612.5.1 db2look概述 51612.5.2 利用db2look构建模拟测试数据库 51712.6 其他工具 51912.6.1
 db2bfd 51912.6.2 db2_kill和db2nkill 52012.6.3 db2tbtst 52112.7 本章小结 521第13章 数据库安全
 52313.1 DB2安全机制概述 52413.2 认证(authentication) 52713.2.1 什么时候进行DB2身份认证
 52713.2.2 DB2身份认证类型 52813.3 权限(authorization) 53513.3.1 权限层次 53513.3.2 授予/撤销实
 例级权限 53913.3.3 授予/撤销数据库级权限 54213.4 特权 54313.4.1 特权层次结构 54313.4.2 授予特
 权 54713.4.3 撤销特权 54913.4.4 显式特权/隐式特权/间接特权 55113.4.5 静态和动态SQL特权考虑因

<<循序渐进DB2>>

素 55513.4.6 维护特权/权限 55713.5 某银行安全规划案例 56013.6 执行安全审计(db2audit) 56213.7
基于标签的访问控制(LBAC)及案例 56613.8 本章小结 573第14章 DBA日常维护 57514.1 DB2健康检
查 57514.1.1 查看是否有僵尸实例进程 57514.1.2 inspect数据库是否一致 57614.1.3 查找诊断日志判
断是否有异常 57614.1.4 检查数据库备份完整性、日志归档是否正常 57714.1.5 维护实例目录和数据
库目录权限 57914.1.6 查看磁盘空间 57914.2 数据库监控 58014.2.1 监控工具 58114.2.2 监控缓冲池
命中率 58214.2.3 监控执行成本最高的SQL语句 58214.2.4 监控运行最长的SQL语句 58214.2.5 监
控SQL准备和预编译时间最长的SQL语句 58314.2.6 监控执行次数最多的SQL语句 58314.2.7 监控排序
次数最多的SQL语句 58414.2.8 监控引起锁等待的SQL语句 58414.3 日常维护 58414.3.1 查找创建的
新对象 58414.3.2 查找无效对象 58514.3.3 检查表空间状态 58514.3.4 检查表状态 58514.3.5 查找需
要REORG的表和索引 58514.3.6 查找需要RUNSTATS的表和索引 58614.3.7 定期清理db2diag.log文件
58714.3.8 查找异常增长的表空间和表 587第15章 DB2常见问题总结 58915.1 实例常见问题和诊断案
例 58915.1.1 实例无法启动问题总结 58915.1.2 实例无法正常终止 59015.1.3 实例启动报SQL1042C错
误 59015.1.4 实例目录误删除 59115.1.5 实例崩溃问题 59215.2 数据库常见问题总结 59215.2.1 数据
库日志空间满——SQL0964C错误 59215.2.2 数据库时区和时间 59415.2.3 中文乱码和代码页转换
59415.2.4 通信错误——SQL30081N 59715.2.5 数据库备份、前滚暂挂 59715.2.6 数据库活动日志删
除 59815.2.7 数据库损坏(数据页、索引页)——SQL1043C 59815.2.8 索引重新构建问题 60015.2.9
DB2实用程序不可用 60115.2.10 快速清空表数据 60115.2.11 表和索引统计信息不一致 60215.3 表
空间状态 60315.3.1 Backup Pending 60415.3.2 脱机 60415.3.3 Quiesced Exclusive|Share|Update 60515.3.4
Restore Pending和Storage Must be Defined 60515.3.5 Roll Forward Pending 60615.3.6 表空间状态总结
60615.4 LOAD期间表状态总结 60715.4.1 Check Pending 60715.4.2 Load Pending 60815.4.3 Load in
Progress 60815.4.4 Not Load Restartable 60915.4.5 Read Access Only 60915.4.6 Unavailable 61015.5 锁相
关问题 61115.5.1 锁升级 61115.5.2 锁等待问题解决流程 61115.5.3 死锁 61115.6 内存常见问题
61215.6.1 bufferpool设置过大数据库无法启动 61215.6.2 排序溢出 61215.6.3 锁内存不足 61215.7 备
份恢复常见问题 61315.8 数据移动常见问题总结 61315.8.1 标识列 61415.8.2 生成列 61715.8.3 大对
象 62115.8.4 空值处理 62215.8.5 定界符注意问题 62515.8.6 PC/IXF注意问题 62815.8.7 代码页不同
注意事项 63015.8.8 日期格式 63115.8.9 XML问题 63315.9 安全常见问题总结 63515.9.1 从PUBLIC
撤销隐式的权限和特权 63615.9.2 保护系统编目视图 63815.9.3 创建实例用户显式指定组 63915.9.4
为SYSxxx_GROUP参数使用显式值 64015.9.5 跟踪隐式的特权 64015.9.6 不授予不必要的特权
64215.9.7 使用加密的AUTHENTICATION模式 64215.9.8 使用独立ID创建和拥有对象 64415.9.9 使
用视图控制数据访问 64515.9.10 使用存储过程控制数据访问 64615.9.11 使用LBAC控制数据访问
64715.9.12 对重要敏感数据加密 64815.10 SQL0805和SQL0818错误 650

<<循序渐进DB2>>

章节摘录

创建数据库对象 在数据库创建后，我们可以根据我们的业务需求来设计和创建数据库对象了。可以在DB2数据库中创建下列数据库对象：**模式 表 索引 序列 视图 触发器** 我们可以使用图形用户界面或通过显式执行SQL语句来创建这些数据库对象。

用于创建这些数据库对象的语句称为“数据定义语言(DDL)”，它们通常以关键字CREATE或ALTER作为前缀。

5.1 模式 5.1.1 模式概念 数据库中的大多数对象指定一个由两部分组成的唯一名称，如图5-1所示。

第一部分(最左边的)称为限定词或模式，而第二部分(最右边的)称为简单(或未限定)名称。

从句法上来说，这两部分并置成用句点分隔的单个字符串。

第一次创建可以由模式名限定的任何对象(例如表、索引、视图、用户定义的数据类型、用户定义的函数、昵称、程序包或触发器)时，会根据对象名称中的限定词将该对象指定给一个特定模式。

图5-1 数据库对象名的组成 例如，图5-2说明在创建表的过程中如何将表指定给一个特定模式。

DB2中的模式(schema)是一个已命名对象的集合，它提供一种方法来按逻辑分组这些对象。这些对象包括表、视图、索引、触发器、函数和包。

模式提供了数据库中对象的逻辑类别。

模式也是名称限定词；它提供一种方法来对几个对象使用相同名称，并防止对这些对象进行二义性引用。

例如，使用模式名“PROD”和“DEV”很容易区分两个不同的SALES表(PROD.SALES和DEV.SALES)。

模式名的最大长度为30字节，它用作分两部分的对象名的第一部分。

例如，名称CITIC.CUSTOMER。

在这个示例中，CUSTOMER表的完全限定名包含模式名：CITIC，这可以在系统编目中将它与其他名为CUSTOMER的表区分开。

可以把模式想象为特定对象的创建者、生成者和主人。

图5-2 创建表并将其指定给一个特定格式 如果创建对象而没有指定模式，那么对象使用您的用户名与一个隐式模式相关联(假设用户或组具有IMPLICIT_SCHEMA数据库权限

，IMPLICIT_SCHEMA权限简单来说就是假设一个用户创建一个对象的时候没有使用模式，那么数据库就隐含地创建一个和用户名一样的模式，关于这个权限在13章中有详细讲解)。

当SQL语句引用对象时，如果没有指定模式名，那么也会隐式地加上调用者的用户名。

5.1.2 系统模式 对于每个数据库，都创建和维护一组系统编目表。

这些表包含关于数据库对象(例如表、视图、索引和包)的定义的信息以及关于用户对这些对象的访问类型的安全信息。

这些表存储在SYSCATSPACE表空间中，并采用保留的系统模式名：**SYSIBM、SYSFUN**和**SYSPROC**：一组例程，包括函数和存储过程，其中的SYSIBM是基本系统编目的模式(不建议直接访问它)。

SYSCAT：一组只读的系统编目表视图，记录数据库对象的结构信息。

SYSSTAT：一组可更新的编目视图。

这些可更新的视图允许更新某些统计信息，从而模拟和测试数据库的性能，或者更新统计信息而不使用RUNSTATS实用程序。

SYSIBMADM：一组动态性能视图，可以从该组视图中获取数据库的性能运行信息。

在本书“第9章：DB2性能监控”中有关于性能视图的详细讲解和案例。

5.1.3 设置和获得当前模式 在客户端连接实例或数据库时，会话的特殊寄存器CURRENT

<<循序渐进DB2>>

SCHEMA包含默认的限定符，用于对特定DB2连接中发出的动态SQL语句所引用的未限定对象进行限定。

它的初始值等于特殊寄存器USER中的值(运行时用户)。

静态SQL语句(在默认情况下)由绑定应用程序的用户的授权ID进行限定。

用户可以使用SET CURRENT SCHEMA语句修改特殊寄存器CURRENT SCHEMA的值。

可以用VALUES CURRENTSCHEMA或SELECT CURRENT SCHEMA FROM SYSIBM.SYSDUMMY1命令获得当前的模式名。

下面我们看两个使用模式的示例： 例5-1 用户= HRUSER01，具有IMPLICIT_SCHEMA权限。

命令结果 CREATE TABLE TEST1(ID INT, NAME VARCHAR(25)) Table HRUSER01.TEST1 created
CREATE TABLE CITIC.TEST1(ID INT, NAME VARCHAR(25)) Table CITIC.TEST1 created

SET CURRENT SCHEMA=CITIC CURRENT SCHEMA special register set to CITIC INSERT INTO TEST1 VALUES(1,John Doe) Data inserted into table CITIC.TEST1 例5-2 用户= HRUSER01，没有IMPLICIT_SCHEMA权限。

命令结果 CREATE TABLE TEST1(ID INT, NAME VARCHAR(25)) SQL0552N "HRUSER01" does not have the privilege to perform operation "IMPLICIT CREATE SCHEMA". SQLSTATE=42502

CREATE TABLE HRUSER01.TEST1 (ID INT, NAME VARCHAR(25)) SQL0552N "HRUSER01" does not have the privilege to perform operation "IMPLICIT CREATE SCHEMA". SQLSTATE=42502 CREATE SCHEMA HRUSER01 AUTHORIZATION HRUSER01 Schema HRUSER01 created CREATE TABLE TEST1(ID INT, NAME VARCHAR(25)) Table HRUSER01.TEST1 created 5.1.4 模式和用户的区别

我们要把模式和用户区分开，默认情况下一个用户(用户拥有IMPLICIT_SCHEMA权限)有一个和它同名的模式，您也可以根据需要创建模式授权给某个用户。

模式创建有隐式创建和显式创建两种方式。

隐式创建 如果您具有IMPLICIT_SCHEMA权限，那么可以隐式创建模式。

只要具有此权限，无论您何时使用不存在的模式名创建对象，都会隐式创建一个模式。

只要创建对象的用户拥有 IMPLICIT_SCHEMA权限，通常会在第一次创建模式中的数据对象时隐式创建模式。

显式创建 使用CREATE SCHEMA语句来创建模式。

有关模式的信息保存在连接的数据库的系统目录表中。

要创建模式并让另一个用户成为该模式的所有者(后一个操作是可选的)，您需要 SYSADM或DBADM权限。

即使您不具有这两种权限中的任何一种，您也可以使用您自己的授权标识来创建模式。

作为CREATE SCHEMA语句的一部分创建的任何对象的定义者是模式所有者。

此所有者可以授予和撤销其他用户的模式特权。

要通过命令行来创建模式，请输入以下语句： CREATE SCHEMA [AUTHORIZATION]

其中是模式的名称。

此名称在目录中已记录的模式内必须唯一，并且不能以SYS开头。

如果指定了可选的AUTHORIZATION子句，那么将成为模式所有者。

如果未指定此子句，那么发出此命令的授权标识将成为模式所有者。

例5-3 下面的示例创建了agent模式并且把agent授权给db2inst1用户所有。

CREATE SCHEMA agent AUTHORIZATION db2inst1 删除模式 在删除模式之前，必须删除该模式中的所有对象或将它们移至另一个模式。

当尝试 DROP语句时，该模式名必须在语句中；否则会返回错误。

要使用命令行来删除模式，请输入： DROP SCHEMA RESTRICT 在以下示例中，删除了模式“agent”： DROP SCHEMA agent RESTRICT 5.2 表设计考虑 所有数据都存储在数据库的表中。

表由不同数据类型的一系列或多列组成。

数据存储在行(或称为记录)中。

<<循序渐进DB2>>

本节我不会过多地讲CREATE TABLE、ALTER TABLE或DROP TABLE之类的命令。

这些命令您可以查SQL参考手册，本节主要讲一些和表设计相关的考虑事项，因为很多时候如果我们在建表的时候没有注意到这些，一旦系统上线，后期的调整往往非常麻烦，所以在建表之前，我们要作好规划设计。

5.2.1 选择合适的数据类型 定义列时，需要对列进行命名，定义这些列中将包含的数据的类型(称为数据类型)，并定义要创建的表中每列的数据长度。

DB2提供了一套丰富且灵活的数据类型。

DB2附带 INTEGER、CHAR、DATE和大对象等基本数据类型。

它还提供了创建用户定义的数据类型(UDT)的工具，使用户能够创建复杂的非传统的数据类型，从而适应当今复杂的编程环境。

在给定的情况下，选用哪种数据类型取决于列中存储的信息的类型和范围。

内置的数据类型分为5类：数字、字符串、大对象、日期时间和XML。

用户定义的数据类型分为：单值类型、结构化类型和引用类型(一般不用)。

DB2内置的数据类型如图5-3所示。

图5-3 DB2内置的数据类型 其中，XML数据类型是DB2 V9以后版本提供的数据类型。

DB2提供了XML数据类型来存储格式良好的XML文档。

XML数据类型用于定义表中存储XML值的列，这些列中存储的所有XML值必须是结构良好的XML文档。

引入此本机XML数据类型能够将结构良好的XML文档以其本机分层格式存储在数据库中其他关系数据库旁边。

XML列中的值存储为与字符串数据类型不同的内部表示。

要在XML数据类型的列中存储XML数据，需要使用XMLPARSE函数对数据进行转换。

可以使用XMLSERIALIZE函数将XML数据类型的值转换为XML文档的串行化字符串值。

DB2还提供了许多其他的内置函数来操纵XML数据类型。

我们在创建表时为列选择数据类型时一定要注意下面几点： 要根据业务需求选择合适的
数据类型，避免出现数据类型转换。

例如，我曾经看到有的客户使用字符来存放日期、时间戳，最后我们还要在程序中使用日期转换函数to_date作数据类型转换，这会对应用程序带来性能影响。

根据需求选择合适的长度。

例如，用一个字段empno来存储员工号，用SMALL INT就可以满足，但是如果我们用INT就会造成两个字节的浪费。

如果某个字段的内容都是数字，建议大家选用整数而不要选用CHAR。

一个占用4字节的INT类型字段就可以表达达到4294967295，如果使用CHAR型则至少需要10个字节。

一个占用8字节LONG INT类型字段就可以表达达到18446744073709551615，如果使用CHAR型至少需要20个字节。

CHAR和VARCHAR的选择，如果一列的数据有变化，但是变化不大时，而我们又追求性能，建议使用CHAR类型，因为VARCHAR的读取性能要分两个步骤，先读长度再读数据比CHAR的性能要弱些。

LONG VARCHAR、BLOB、CLOB和CBLOB数据类型的选择，这些大对象数据类型的读取是不经过内存而直接读取的，所以可根据情况看是否能够用VARCHAR字段代替。

如果使用大对象数据类型，考虑是否对该大对象列记录日志NOT LOGGED。

考虑把大对象数据列单独存放在独立的表空间，和索引数据分隔存放。

下面让我们看看列在磁盘上是如何布局的。

如果您创建了一个只有定长列的表，将严格按照 CREATE 语句中指定的顺序安排它们，如图5-4所示。

CREATE TABLE TESTORD(COL1 INT, COL2 CHAR(5), COL3 DEC(10,2), COL4 FLOAT) 图5-4

定长列的表在磁盘上布局 如果表拥有变长列(如VARCHAR)，列仍然按照CREATE TABLE语句中指定的顺序排序，但可变数据本身在行的末尾，如图5-5所示。

<<循序渐进DB2>>

CREATE TABLE TESTORD(COL1 INT, COL2 VARCHAR(5), COL3 DEC(10,2)) 图5-5 拥有变长列的表在磁盘上的布局 如果表有长字段，它将不随每行直接插入。因为行的长度受页大小限制(4KB到32KB)，所以行只有一个指向长字段的指针，而将长字段与行分开放置在数据库页中，如图5-6所示。

CREATE TABLE TESTORD(COL1 INT, COL2 CLOB(100 K), COL3 DEC(10,2)) 图5-6 拥有长字段的表在磁盘上的布局 5.2.2 选择合适的约束类型 在任何业务中，数据通常必须符合特定限制或业务规则。

例如，职员编号、银行支票号必须是唯一的。数据库管理器提供了约束作为强制实施这种规则的方法。约束是用于业务需求的规则。

DB2提供了下列5种类型的约束： NOT NULL约束 NOT NULL约束防止在列中输入空值。NOT NULL约束是这样一种规则，它防止在表的一列或多列中输入空值。

数据库中使用空值来表示未知状态。

默认情况下，随数据库管理器一起提供的所有内置数据类型都支持空值的存在。

但是，一些业务规则可能要求必须始终提供值(例如，乘飞机时必须提供紧急联系人信息)。

NOT NULL约束用于确保决不会为给定表列指定空值。

为特定列定义NOT NULL约束后，尝试在该列中放入空值的任何插入或更新操作将失败。

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。在唯一约束中指定的列必须定义为NOT NULL。

唯一约束(也称为唯一键约束)是这样一种规则，它禁止表的一列或多列中出现重复值。

唯一键和主键是受支持的唯一约束。

例如，可对供应商表中的供应商标识定义唯一约束以确保不会对两个供应商指定同一供应商标识。

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。

在唯一约束中指定的列必须定义为NOT NULL。

数据库管理器使用唯一索引在对唯一约束的各列进行更改时强制键的唯一性。

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5-7显示了当表存在唯一约束时，阻止将重复的记录添加到该表。

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。

表可以有任意数目的唯一约束，但最多将一个唯一约束定义为主键。

对于同一组列，表不能有多个唯一约束。

<<循序渐进DB2>>

编辑推荐

本书主要特色： 本书有丰富的DB2运行维护实践案例，读者可以通过这些案例加深对DB2认识和理解。

面对众多DB2的特点、特性，作者精心选择了与DBA日常工作关系最密切的部分，由易入难逐步讲解。

本书中包含了大量的提示点等，这些都是作者对DB2维护工作总结后得到的经验。

本书总结了DB2常见问题总结和解决方法，对DBA的日常维护非常有参考价值。

读者对象 数据库管理人员、数据库开发人员、系统维护人员、数据库初学者及其他数据库从业人员，也可以作为各大中专院校相关专业师生的参考用书和相培训机构的培训教材。

<<循序渐进DB2>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>