

IBM DB2 Cube Views



# 指南和参考

版本 8.2



IBM DB2 Cube Views



# 指南和参考

版本 8.2

**注意**

注意：在使用本资料及其支持的产品之前，请阅读第 255 页的『声明』中的信息。

**第二版（2004 年 9 月）**

本文档包含 IBM 的专利信息。它根据许可证协议提供，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，且本手册中提供的任何声明不应作如此解释。

您可以在线方式或通过您当地的 IBM 代表订购 IBM 出版物。

- 要以在线方式订购出版物，可访问“IBM 出版物中心”（IBM Publications Center），网址为 [www.ibm.com/shop/publications/order](http://www.ibm.com/shop/publications/order)
- 要查找您当地的 IBM 代表，可访问“IBM 全球联系人目录”（IBM Directory of Worldwide Contacts），网址为 [www.ibm.com/planetwide](http://www.ibm.com/planetwide)

要从美国或加拿大的 DB2 营销机构订购 DB2 出版物，请拨打电话 1800-IBM-4YOU（425-4968）。

当您发送信息给 IBM 后，即授予 IBM 非专有权，IBM 可以它认为合适的任何方式使用或分发此信息，而无须对您承担任何责任。

# 目录

关于本书. . . . .	v	从立方体模型中除去维. . . . .	48
本书的适用对象. . . . .	v	从数据库中删除元数据对象. . . . .	49
本书中使用的语法定义. . . . .	v		
在线信息. . . . .	vi		
<b>第 1 章 安装、迁移和配置 DB2 Cube Views. . . . .</b>	<b>1</b>	<b>第 4 章 DB2 Cube Views 企业建模方案 51</b>	
DB2 Cube Views 增强功能. . . . .	1	随着时间的推移计算仓库存货的流和值. . . . .	51
DB2 Cube Views 简介. . . . .	3	使广告成本与销售相关. . . . .	56
安装需求. . . . .	4	计算商店的利润和边际利润. . . . .	61
安装 DB2 Cube Views. . . . .	5	对因特网订单数量进行计数. . . . .	64
将 DB2 Cube Views XML 文件从 V8.1 迁移至 V8.2. . . . .	6	将销售数字分等级. . . . .	66
为 DB2 Cube Views 设置数据库. . . . .	6	使用存储在事实表中的时间数据来创建 Time 维. . . . .	69
设置 CVSAMPLE 样本数据库. . . . .	7		
数据库连接故障诊断. . . . .	8		
<b>第 2 章 关于 DB2 Cube Views 元数据对象. . . . .</b>	<b>9</b>	<b>第 5 章 DB2 Cube Views 立方体模型优化. . . . .</b>	<b>73</b>
映射至关系表的元数据对象. . . . .	9	摘要表. . . . .	74
映射至关系表的立方体模型的示例. . . . .	12	带有功能相关性和约束的摘要表. . . . .	78
公共元数据对象属性. . . . .	17	优化过程概述. . . . .	80
立方体模型. . . . .	18	关于优化的元数据设计注意事项. . . . .	83
事实对象. . . . .	19	立方体的优化片. . . . .	84
维. . . . .	19	分析查询以获取候选优化片. . . . .	91
层次结构. . . . .	20	用于优化的约束定义. . . . .	93
层. . . . .	23	优化顾问程序的参数. . . . .	97
量度. . . . .	26	优化立方体模型. . . . .	98
属性. . . . .	28	用于创建摘要表的 SQL 脚本的示例. . . . .	99
属性关系. . . . .	29	测试查询结果. . . . .	102
连接. . . . .	30	对摘要表进行故障诊断. . . . .	102
立方体. . . . .	30	摘要表维护. . . . .	104
立方体事实对象. . . . .	31	删除摘要表. . . . .	104
立方体维. . . . .	31		
立方体层次结构. . . . .	32	<b>第 6 章 DB2 Cube Views 和联合数据源. . . . .</b>	<b>107</b>
立方体层. . . . .	32	联合系统概述. . . . .	107
元数据对象规则. . . . .	33	使用 DB2 Cube Views 优化远程数据源概述. . . . .	108
		对 DB2 Cube Views 启用联合系统. . . . .	109
		定义远程数据源. . . . .	109
		对 DB2 Cube Views 的远程表定义昵称. . . . .	110
		对 DB2 Cube Views 的昵称定义信息性约束. . . . .	110
		远程数据源的查询重新路由故障诊断. . . . .	111
<b>第 3 章 设计 DB2 Cube Views 元数据对象模型. . . . .</b>	<b>37</b>	<b>第 7 章 DB2 Cube Views API. . . . .</b>	<b>113</b>
启动和刷新 OLAP 中心. . . . .	37	DB2 Cube Views API 概述. . . . .	113
OLAP 中心和 API 版本兼容性. . . . .	37	DB2 Cube Views API: DB2 存储过程和 XML 语法分析. . . . .	114
DB2 Cube Views 的权限和特权. . . . .	38	DB2 Cube Views 存储过程. . . . .	115
创建 DB2 Cube Views 元数据对象. . . . .	40	DB2 Cube Views API 参数. . . . .	117
在 DB2 Cube Views 与 OLAP 工具之间交换元数据. . . . .	40	DB2 Cube Views API 元数据操作的参数. . . . .	117
使用快速启动向导创建立方体模型. . . . .	41	输入和输出参数. . . . .	117
创建完整的立方体模型. . . . .	42	DB2 Cube Views 元数据操作. . . . .	117
将现有维添加至立方体模型. . . . .	45	检索操作: 描述. . . . .	117
创建连接. . . . .	46	建议操作: 建议. . . . .	118
创建立方体. . . . .	47	修改操作: 改变、创建、删除、导入和重命名. . . . .	123

管理操作: 验证和转换 . . . . .	127
元数据操作中的样本输入和输出参数 . . . . .	129
操作参数 . . . . .	131
操作的操作数 . . . . .	134
消息结构 . . . . .	134
操作步骤的顺序 . . . . .	135
日志记录和跟踪 . . . . .	136
DB2 Cube Views API 的运行时跟踪 . . . . .	136
DB2 Cube Views API 的日志文件 . . . . .	136
错误日志记录 . . . . .	137
日志记录和跟踪方案 . . . . .	137
代码页支持 . . . . .	138
DB2 Cube Views 元数据表和 XML 模式文件 . . . . .	140
DB2 Cube Views 配置文件 . . . . .	141
元数据对象格式 . . . . .	141
<b>第 8 章 样本文件 . . . . .</b>	<b>143</b>
db2mdapiclient 实用程序概述 . . . . .	143
db2mdapiclient 命令: 处理元数据对象 . . . . .	144
样本数据库文件 . . . . .	145
API 样本文件 . . . . .	146

<b>附录. 消息 . . . . .</b>	<b>151</b>
SQLSTATE、API 和其它服务器消息 . . . . .	151
API SQL 状态 . . . . .	151
常规 . . . . .	152
外部 API 错误 . . . . .	172
优化 . . . . .	211
OLAP 中心消息 . . . . .	219
10000-10600 . . . . .	219
DB2 和 DB2 Cube Views 中的状态消息 . . . . .	253
<b>声明 . . . . .</b>	<b>255</b>
商标 . . . . .	256
<b>词汇表 . . . . .</b>	<b>259</b>
<b>索引 . . . . .</b>	<b>263</b>
<b>与 IBM 联系 . . . . .</b>	<b>267</b>
产品信息 . . . . .	267
对文档的意见 . . . . .	267

---

## 关于本书

本书提供与下列 DB2 Cube Views 主题有关的信息:

- 如何开始使用 DB2 Cube Views
- OLAP 中心 (图形用户界面), 您可以使用它来导入和导出元数据、创建立方体模型和立方体
- 优化, 帮助您提高 OLAP 查询的性能
- 元数据对象, 可以存储在 DB2 Universal Database™ (DB2®) 目录中
- 应用程序编程接口 (API), 通过它可以创建使用 SQL 访问数据的应用程序
- 如何通过元数据对象构建维和复杂量度的示例; 这些维和量度可用于建立典型商业方案的模型

---

## 本书的适用对象

通过 DB2 Cube Views, 您可以从 OLAP 和数据库工具中捕获多维元数据, 并将该元数据存储在 DB2 目录中。然后, 您可以使用此元数据来创建 OLAP (联机分析处理) 立方体模型和立方体。(立方体是立方体模型的子集)。

DB2 Cube Views 还提供了优化顾问程序, 它通过提供 SQL 脚本以构建摘要表来帮助您改进向立方体模型发出的查询的性能。

如果您是使用 OLAP 元数据和 DB2 通用数据库 (DB2 UDB) 的数据库管理员, 请阅读本书。您应熟悉:

- DB2 目录和摘要表
- OLAP 概念, 如立方体、维、层次结构和量度
- API 概念以及 CLI、ODBC、JDBC、XML 和 DB2 存储过程

---

## 本书中使用的语法定义

命令语法以正在运行的文本格式显示, 带有下列约定:

- 命令以小写粗体字体显示。
- 变量以斜体显示且紧跟在列表中的命令之后说明。
- 如果可以输入两个或更多备用参数中的一个参数, 则可用的参数显示时以竖条隔开, 缺省参数将加上下划线。
- 可选变量和参数将括在方括号中。

例如, **copy** 命令语法可以以下格式显示:

**copy** *filename* [*filetype*]

*filename* 是该文件的必需名称。

[*filetype*] 是可选文件类型。

---

## 在线信息

本节提供与此产品相关的 Web 地址。

### **www.ibm.com/redbooks**

IBM® Redbooks™ Web 站点

从 Redbooks Web 站点搜索、查看、下载或订购以下红皮书的硬拷贝 / CD 版本:

- *DB2 UDB's High Function Business Intelligence in e-business*, SG24-6546-00
- *Up and Running with DB2 UDB ESE Partitioning for Performance in an e-Business Intelligence World*, SG24-6917-00
- *Database Performance Tuning on AIX®*, SG24-5511-01
- *DB2 UDB V7.1 Performance Tuning Guide*, SG24-6012-00

### **www.ibm.com/software/data/db2/db2md**

IBM DB2 Cube Views Web 站点

### **www.ibm.com/software/data/**

IBM 数据管理 Web 站点

### **www.ibm.com/software/data/db2/udb/winos2unix/support/**

DB2 通用数据库和 DB2 Connect™ 在线支持 Web 站点

### **www.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/download.d2w/report**

DB2 维护 - DB2 UDB 修订包 Web 站点

### **www.ibm.com/software/data/developer**

DB2 开发者园地 Web 站点

### **www.ibm.com/software/data/db2/library**

DB2 产品和服务技术库 Web 站点

### **www.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/v8pubs.d2w/en\_main**

DB2 出版物 Web 站点



---

## 第 1 章 安装、迁移和配置 DB2 Cube Views

本节描述下列主题:

### DB2 Cube Views 增强功能

此发行版合并了自 DB2 Cube Views V8.1 以来对 DB2 Cube Views 元数据的若干更改。

### DB2 Cube Views 简介

DB2 Cube Views 是 DB2 Universal Database (DB2 通用数据库) 的一个加载功能部件, 用于改进 DB2 UDB 执行 OLAP 处理的能力。您可以使用 DB2 Cube Views 来简化 OLAP 解决方案的部署和管理, 以及改进 OLAP 工具和应用程序的性能。

### 安装需求

在安装 DB2 Cube Views 之前, 确保符合所有需求。

### 安装 DB2 Cube Views

在 Windows、AIX、Linux 和 Solaris Operating System 平台上使用安装启动板安装 DB2 Cube Views。

### 迁移 DB2 Cube Views

可通过手工运行脚本或通过使用 OLAP 中心将启用 DB2 Cube Views 的数据库从版本 8.1 迁移至版本 8.2。

### 为 DB2 Cube Views 设置数据库

可为 DB2 Cube Views 设置新的数据库以供使用。

### 设置 CVSAMPLE 样本数据库

DB2 Cube Views 提供可以用来创建名为 CVSAMPLE 的样本数据库的样本数据。

### 数据库连接故障诊断

如果使用 OLAP 中心不能连接至数据库, 则检查 DB2 Cube Views 的版本是否与 DB2 目录中的元数据表的版本相匹配。

---

## DB2 Cube Views 增强功能

此发行版合并了自 DB2 Cube Views V8.1 以来对 DB2 Cube Views 元数据的若干更改。

### 用于改善摘要表的约束和功能相关性信息

DB2 Cube Views 利用约束和功能相关性信息来产生较小的摘要表并提供更好的查询覆盖范围。

### 约束信息

在 DB2 Cube Views 中, 立方体模型约束是在事实表与维表之间 (或者在雪花维的表之间) 定义的。如果摘要表包含维表的主键, 则 DB2 UDB 可将查询重新路由至摘要表, 即使查询引用该维的其它列也是如此。

### 功能相关性信息

DB2 UDB 具有基于功能相关性信息的智能路由功能，用来解析 SQL 查询。功能相关性允许您指定功能上依赖于一个或多个其它列的一个或多个列（假定所有列都存在于同一个表中）。

如果 SQL 查询包括摘要表中的列和从中构造摘要表的表中的列并且两个列之间存在功能相关性，则 DB2 优化器可通过将两个表连接在一起解析查询以获取最终结果集。

DB2 Cube Views 会建议摘要表只包括选择维和层的键列以便生成的摘要表范围要窄一些。

### 用于改善摘要表的用户定义的优化片

此发行版的 DB2 Cube Views 现在包括一种机制以便您指定将使用的查询类型以及它们将引用立方体的什么区域。优化顾问程序可以使用此附加元数据来使优化集中在立方体的适当区域。任何类型的查询都将获益，但报表查询有可能会得到最明显的改善。

### 元数据 API 支持的多个 XML 模式版本

DB2 Cube Views 元数据 API 仅完全支持 XML 模式版本号 8.2.0.1.0（包括新的 DESCRIBE 和 TRANSLATE 操作）。

要求传递至元数据 API 或从元数据 API 传递来的所有 XML 文档都具有版本号。版本号使存储过程能够识别客户机正在使用哪个 XML 模式。

### 属性和量度可空性

已扩展元数据 API 属性和量度以包含称为 nullable 的只读属性，它是由 *yes*、*no* 或 *unknown* 值组成的值集。

#### *yes* 的可空性

属性或量度可包含空值。

#### *no* 的可空性

属性或量度永远也不能包含空值。

#### *unknown* 的可空性

不能由 API 确定可空性，或者当将属性或量度从 V8.1 迁移至 V8.2 时不能由迁移实用程序确定可空性。

### 通过使用层的层次结构建模

在 DB2 Cube Views 的前发行版中，层次结构是从属性的有序列表建模的，层次结构由属性和称为属性关系的对象组成。在发行版中，层次结构从层对象的有序列表建模，每层引用一个或多个属性。

此发行版包括新的“层向导和层属性”窗口，该窗口允许您创建和修改新的层对象。

### 层对象的功能相关性

当创建层时，可以请求 DB2 Cube Views 尝试创建功能相关性对象。

功能相关性指示层对象的缺省属性和相关属性在功能上由层的键属性确定。然后，优化顾问程序就可以使用功能相关性来尽量减小它建议的摘要表大小。

如果不能创建功能相关性，则会返回警告消息。例如，功能相关性不能指向跨多个表的列。

**限制：** 如果基础列不遵守在层对象的功能相关性中定义的关系，则查询结果可能会不正确。因为 DB2 UDB 不会对任何功能相关性检查数据的有效性，所以必须确保表列中的数据以指定的方式功能相关。

有关功能相关性的更多信息，请参阅第 23 页的『层』。

### 优化顾问程序的自动操作

此发行版包括新的 API ADVISE 操作。此操作建议应该构建这些摘要表来改进立方体模型的查询性能。此操作具有一些自变量，它们限制优化顾问程序可以运行的时间长短以及它可以用于摘要表的磁盘空间量。

### 称为 CVSAMPLE 的已更新样本数据库

提供了一个名为 CVSAMPLE 的新的已改善样本数据库。CVSAMPLE 数据库是更健壮的雪花模式。还提供了反映此发行版中的元数据库更改的 DB2 Cube Views 元数据。有关如何设置 CVSAMPLE 数据库的信息，请参阅第 7 页的『设置 CVSAMPLE 样本数据库』。

---

## DB2 Cube Views 简介

DB2 Cube Views 是 DB2 Universal Database (DB2 通用数据库) 的一个加载功能部件，用于改进 DB2 UDB 执行 OLAP 处理的能力。您可以使用 DB2 Cube Views 来简化 OLAP 解决方案的部署和管理，以及改进 OLAP 工具和应用程序的性能。

通过 DB2 Cube Views，您可以描述关系表的维结构并创建 OLAP 构造。您可以将结构信息和 OLAP 构造作为多维元数据存储于 DB2 数据库中。

DB2 UDB 中的新多维元数据具有以下两个主要优点：

### 改进商业智能工具和应用程序之间的多维元数据流动性

使用 OLAP 中心（附带的一种图形界面），仓储和商业智能工具的用户可以将多维元数据作为 DB2 数据库一部分来存储，并使其可用于所有工具和应用程序。

### 增强 OLAP 式的查询的性能

基于多维元数据，可以使用 OLAP 中心中的优化顾问程序的建议来创建 DB2 摘要表。摘要表包含映射到 OLAP 结构的预计算数据。从具有相同 OLAP 结构的仓储或商业智能应用程序生成的查询的性能将会有所提高。

DB2 Cube Views 使用 DB2 功能部件，如摘要表、不同的索引方案、OLAP 式运算符和聚集函数。提供了下列组件：

### 多维元数据对象

您可以创建一组元数据对象，从而以维的方式建立关系数据和 OLAP 结构的模型。DB2 Cube Views 将您创建的每个元数据对象存储在 DB2 目录中。

### OLAP 中心

通过 OLAP 中心，您可以创建、处理、导入或导出立方体模型、立方体以及在 OLAP 工具中使用的其它元数据对象。OLAP 中心提供了易于使用的向导和

窗口，用来帮助您处理元数据对象。例如，优化顾问程序分析元数据对象并就如何构建摘要表提出建议，这些摘要表为 OLAP 式 SQL 查询存储聚集数据并为聚集数据建立索引。要启动 OLAP 中心，请参阅启动 OLAP 中心。在启动 OLAP 中心之后，请参阅优化立方体模型以使用优化顾问程序向导。

### 多维服务

DB2 Cube Views 为 OLAP 工具和应用程序开发者提供了基于 SQL 和基于 XML 的应用程序编程接口 (API)。使用 CLI、ODBC 或 JDBC 连接或者通过使用 DB2 UDB 的嵌入式 SQL，应用程序和工具可以使用单个存储过程来创建、修改和检索元数据对象。

### 样本数据

样本应用程序和数据库可用于帮助您学习如何使用产品。

您还可以在 DB2 目录和 OLAP 工具之间交换元数据对象。为了能将元数据对象导入或导出 DB2 目录，对特定 OLAP 和数据库工具提供了名为“元数据桥接”的实用程序。请参阅特定 OLAP 或数据库工具的文档以确定是否提供元数据桥接。

---

## 安装需求

在安装 DB2 Cube Views 之前，确保符合所有需求。

### 系统需求

必须在想要连接至的每个 DB2 UDB 服务器和客户机上安装适当的组件。多维服务在服务器和客户机上是必需的。还可在客户机上安装 OLAP 中心。

**限制:** 必须对所有客户机和服务器组件安装相同版本的 DB2 Cube Views。OLAP 中心版本 8.1 客户机可连接至 DB2 UDB 版本 8.2 服务器，但不能执行任何“创建”、“改变”或“删除”操作。

必须具有下列服务器、客户机和硬件组件。

- 服务器组件:

#### **Microsoft® Windows®**

Windows NT® 4、Windows 2000® 32 位、Windows XP Professional 32 位、Windows Server 2003 32 位或 Windows Server 2003 64 位

**AIX** AIX V4.3.3 32 位、AIX 5L™ 32 位或 AIX 5L 64 位

#### **Linux:**

Linux Red Hat 8 (内核 2.4.18, glibc 2.2.93-5) 32 位、Linux SuSE 8.0 (内核 2.4.18, glibc 2.2.5) 32 位、Linux SLES 8 SP3 (内核 2.4.21, glibc 2.2.5) 32 位或 Linux RHEL 3 Update 2 (内核 2.4.21-9, glibc 2.3.2) 32 位。

有关 DB2 UDB 支持的分发和内核级别的最新信息，请访问 [www.ibm.com/db2/linux/validate](http://www.ibm.com/db2/linux/validate)

#### **On Sun Solaris™ Operating System**

Solaris 8 32 位、Solaris 8 64 位、Solaris 9 32 位或 Solaris 9 64 位

#### **在 HP-UX 上**

HP-UX 11i v2 64-bit for Intel™ Itanium

- 客户机组件: Windows NT 4、Windows 2000 32 位、Windows XP 32 位、Windows XP 64 位、Windows Server 2003 32 位或 Windows Server 2003 64 位
- 硬件组件:
  - 500 MB 磁盘空间
  - 256 MB RAM

### 安装 DB2 Cube Views 的先决条件

在安装 DB2 Cube Views 之前, 应安装下列组件。

#### DB2 信息中心

要访问 OLAP 中心的联机帮助和《DB2 Cube Views 指南和参考》的联机版本, 必须安装 DB2 信息中心。从 DB2 信息中心 CD 安装 DB2 信息中心。

#### DB2 通用数据库版本 8.2

在安装 DB2 Cube Views 之前, 必须安装 DB2 UDB 版本 8.2。

注: 在安装 DB2 UDB 版本 8.2 之前, 应除去所有先前版本的 DB2 UDB。

---

## 安装 DB2 Cube Views

在 Windows、AIX、Linux 和 Solaris Operating System 平台上使用安装启动板安装 DB2 Cube Views。

### 在 Windows 上安装 DB2 Cube Views

在安装 DB2 Cube Views 之前, 确保:

- 安装了 DB2 信息中心。
- 完成了 DB2 通用数据库版本 8.2 的干净安装。
- 符合所有安装需求。

要在 Windows 上安装 DB2 Cube Views:

1. 插入 DB2 Cube Views CD。安装程序将自动启动。
2. 在启动板上, 查看**发行说明**以了解最新安装、磁盘和内存需求信息。此外, 查看 DB2 Cube Views CD 的根目录中的 readme.txt 文件以获取任何附加指示信息。
3. 单击**安装产品**以开始安装并遵循提示。

### 在 AIX、Linux 或 Solaris Operating System 上安装 DB2 Cube Views

在安装 DB2 Cube Views 之前, 确保:

- 完成了 DB2 通用数据库版本 8.2 的干净安装。
- 符合所有安装需求。

要在 AIX、Linux 或 Solaris Operating System 上安装 DB2 Cube Views:

1. 插入 DB2 Cube Views CD。
2. 切换至 UNIX 操作系统上的相应目录并启动 db2setup 文件。
3. 在启动板上, 查看**发行说明**以了解最新安装、磁盘和内存需求信息。此外, 查看 DB2 Cube Views CD 的根目录中的 readme.txt 文件以获取任何附加指示信息。

4. 单击**安装产品**以开始安装并遵循提示。

---

## 将 DB2 Cube Views XML 文件从 V8.1 迁移至 V8.2

可通过手工运行脚本或通过使用 OLAP 中心将启用 DB2 Cube Views 的数据库从版本 8.1 迁移至版本 8.2。

### 使用 DB2 脚本迁移

**建议：**在运行此迁移脚本之前备份并导出版本 8.1 XML 元数据。

使用 `sqllib\misc` 目录中的 `db2mdmigrate.sql` 文件来创建新的元数据表以及将元数据从 V8.1 迁移至 V8.2 所需的其它 SQL。要使用此文件：

1. 通过输入以下命令连接至想要迁移的数据库：

```
db2 connect to database_name
```

2. 从 DB2 命令窗口输入以下命令以运行 `db2mdmigrate.sql` 文件。

```
db2 -tvf db2mdmigrate.sql
```

此脚本没有错误处理。此脚本运行一组 DDL 和 SQL 语句。如果任何语句失败或在脚本还未完成时放弃，则迁移将只完成一部分，而且 DB2 Cube Views 无法正常工作。

3. 如果遇到错误，执行下列任务。
  - a. 删除所有 DB2INFO.\* 表。
  - b. 删除 DB2INFO.MD\_MESSAGE 存储过程。
  - c. 使用 `sqllib\misc` 目录中的文件 `db2mdapi.sql` 来重新创建元数据表。

### 使用 OLAP 中心进行迁移

当 OLAP 中心连接至 DB2 UDB 数据库时，OLAP 中心将自动检测 DB2 UDB 的当前版本并确定是否需要迁移元数据表。如果 OLAP 中心确定需要迁移，它将显示一条错误消息以通知您。

如果接受 OLAP 中心的迁移建议，OLAP 中心将连接至该数据库并将元数据表迁移至 V8.2。如果拒绝迁移，OLAP 中心将不会连接至该数据库。

如果迁移过程中发生任何错误，OLAP 中心将回滚该事务，而数据库将不会迁移。

---

## 为 DB2 Cube Views 设置数据库

可为 DB2 Cube Views 设置新的数据库以供使用。

设置数据库包括：

- 向数据库注册 DB2 Cube Views 存储过程
- 为 DB2 Cube Views 创建元数据目录表

使用 OLAP 中心首次登录至尚未为 DB2 Cube Views 配置的数据库时，OLAP 中心将为您设置该数据库。另外，您还可以使用 `db2mdapi.sql` 文件来设置数据库。

**要点：**不要改变 `db2mdapi.sql` 文件，否则将获得不可预测的结果。

要使用 db2mdapi.sql 文件安装数据库:

1. 打开 DB2 命令窗口并连接至您的数据库。
2. 切换至 SQLLIB\misc 目录, 并在 DB2 命令窗口中输入以下命令:  

```
db2 -tvf db2mdapi.sql
```

对数据库仅运行一次 db2mdapi.sql 脚本。如果遇到错误, 解决问题然后执行下列任务。

- a. 删除所有 DB2INFO.\* 表。
- b. 删除 DB2INFO.MD\_MESSAGE 存储过程。
- c. 使用 sqllib\misc 目录中的文件 db2mdapi.sql 重新创建元数据表。

---

## 设置 CVSAMPLE 样本数据库

DB2 Cube Views 提供可以用来创建名为 CVSAMPLE 的样本数据库的样本数据。

该样本数据不仅包括一组表, 这些表中包含关于虚构的饮料销售公司的数据。而且包括一组描述样本数据表的元数据对象。提供的 CVSAMPLE 数据是版本 8.1 中提供的 MDSAMPLE 数据库的改进扩展版本。《DB2 Cube Views 指南和参考》中的大多数示例基于 CVSAMPLE 数据库和相应的立方体模型。

通过打开 DB2 命令窗口并输入以下命令来创建并填充样本 CVSAMPLE 数据库:

1. 创建名为 CVSAMPLE 的样本数据库:  

```
db2 create db cvsample
```
2. 连接至数据库:  

```
db2 connect to cvsample
```
3. 运行 db2mdapi.sql 脚本以便为 DB2 Cube Views 设置数据库。切换至 SQLLIB\misc 目录, 然后输入以下 DB2 命令:  

```
db2 -tvf db2mdapi.sql
```
4. 切换至 SQLLIB\samples\olap\cvsample 目录, 然后输入以下 DB2 命令以创建 CVSAMPLE 表:  

```
db2 -tvf CVSampleTables.sql
```

在创建 CVSAMPLE 数据库之后, 可通过从由商业智能应用程序导出的 XML 文件导入定义来创建 DB2 Cube Views 元数据对象。

例如, 以下过程使用 CVSAMPLE 数据库的完整描述来填充 DB2 Cube Views 目录表。

要导入 CVSAMPLE 元数据:

1. 启动 OLAP 中心并连接至 CVSAMPLE 数据库。
2. 单击 **OLAP 中心 -> 导入**。
3. 浏览以查找位于 SQLLIB/samples/olap/cvsample 目录的 CVSampleMetadata.xml 文件。  
单击下一步。

可在 OLAP 中心中浏览元数据对象。有关使用 OLAP 中心的信息, 请参阅第 40 页的『创建 DB2 Cube Views 元数据对象』。

---

## 数据库连接故障诊断

如果使用 OLAP 中心不能连接至数据库，则检查 DB2 Cube Views 的版本是否与 DB2 目录中的元数据表的版本相匹配。

下表显示当 DB2 Cube Views 的版本与 DB2 目录中的元数据表的版本不匹配时 OLAP 中心要采取的措施。

DB2 Cube Views 的版本	DB2 目录中的元数据表的版本	OLAP 中心的行为
未安装	不适用	连接失败，OLAP 中心显示错误消息
版本 8.1	无	连接失败，OLAP 中心显示错误消息
版本 8.1	版本 8.1	连接失败，OLAP 中心显示错误消息
版本 8.2	无	OLAP 中心可配置数据库以与 DB2 Cube Views 版本 8.2 配合使用
版本 8.2	版本 8.1	OLAP 中心可迁移数据库以与 DB2 Cube Views 版本 8.2 配合使用
版本 8.2	版本 8.2	连接成功。



---

## 第 2 章 关于 DB2 Cube Views 元数据对象

DB2 Cube Views 元数据对象将关系表描述为 OLAP 结构，但这些元数据对象与传统的 OLAP 对象不同。元数据对象存储有关基本表中数据的元数据，它们描述相关数据所在的位置，并描述基本数据内的关系。

DB2 Cube Views 将关于关系数据的信息存储在元数据对象中，这些元数据对象为您了解数据提供了新的角度。DB2 Cube Views 扩展了 DB2 目录，除了存储有关表和列的信息之外，DB2 目录还包含有关表和列如何与 OLAP 对象相关以及这些元数据对象之间的关系的消息。

某些元数据对象通过聚集数据或直接与关系表中的特定列对应来充当直接访问关系数据的基准。其它元数据对象描述基本元数据对象之间的关系，并将这些基本元数据对象链接在一起。可以依据元数据对象之间的相互关系将所有元数据对象组合称为立方体模型的元数据对象中。本质上，立方体模型表示关系表的特定分组方式和配置。

DB2 Cube Views 管理下列元数据对象，并将它们存储在 DB2 目录中：

- 立方体模型
- 事实对象
- 维
- 层次结构
- 层
- 量度
- 属性
- 属性关系
- 连接
- 立方体
- 立方体事实对象
- 立方体维
- 立方体层次结构
- 立方体层

---

### 映射至关系表的元数据对象

立方体模型可以用多种方式构造，但通常构建为表示关系星型模式或雪花模式。立方体模型包含描述关系数据中的关系的元数据对象。

星型模式在中心具有一个事实表以及连接至该事实表的一个或多个维表，而雪花模式是星型模式的扩展，这样，一个或多个维由多个表定义。基于简单星型模式的立方体模型是在中心事实对象周围构建的。事实对象包含描述如何从跨维的事实表中聚集数据的量度集。量度描述从关系表的列中进行的数据计算，并且连接在一起来创建事实对象。第 10 页的图 1 显示量度和事实对象如何与关系数据相关。

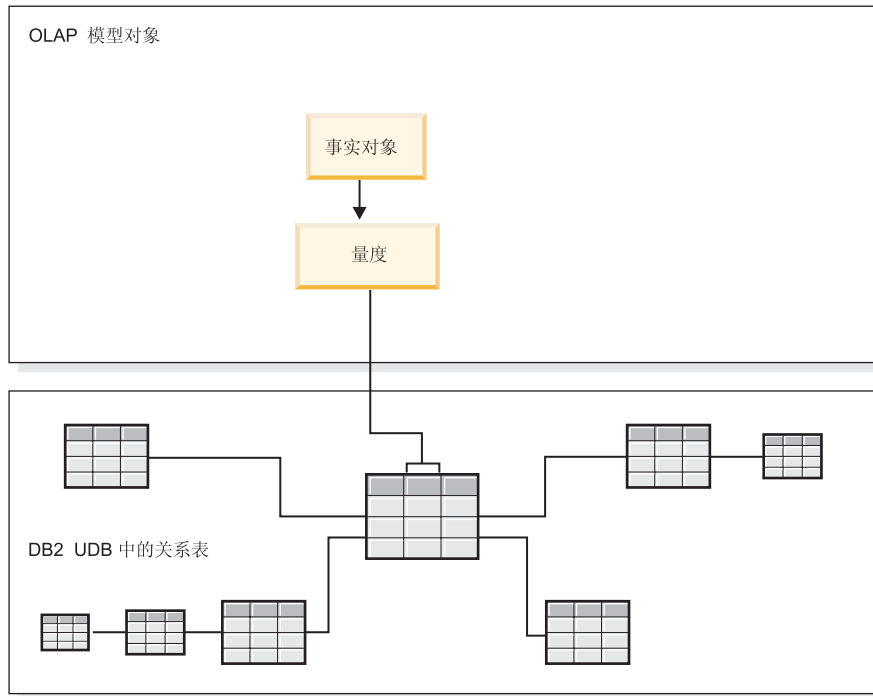


图 1. 事实对象. 事实对象和量度如何与关系数据相关

维连接至立方体模型中的事实对象，就象维表连接至星型模式中的事实表。关系表中的数据列由组织成维的属性表示。

第 11 页的图 2 显示如何从关系表构建维。层次结构存储有关维中的层如何彼此相关以及如何构造的信息。层次结构提供了计算和导航维的方式。每个维都有一个相应的层次结构，该层次结构包含带有相关属性的集合的层。在立方体模型中，每个维可以具有多个层次结构。

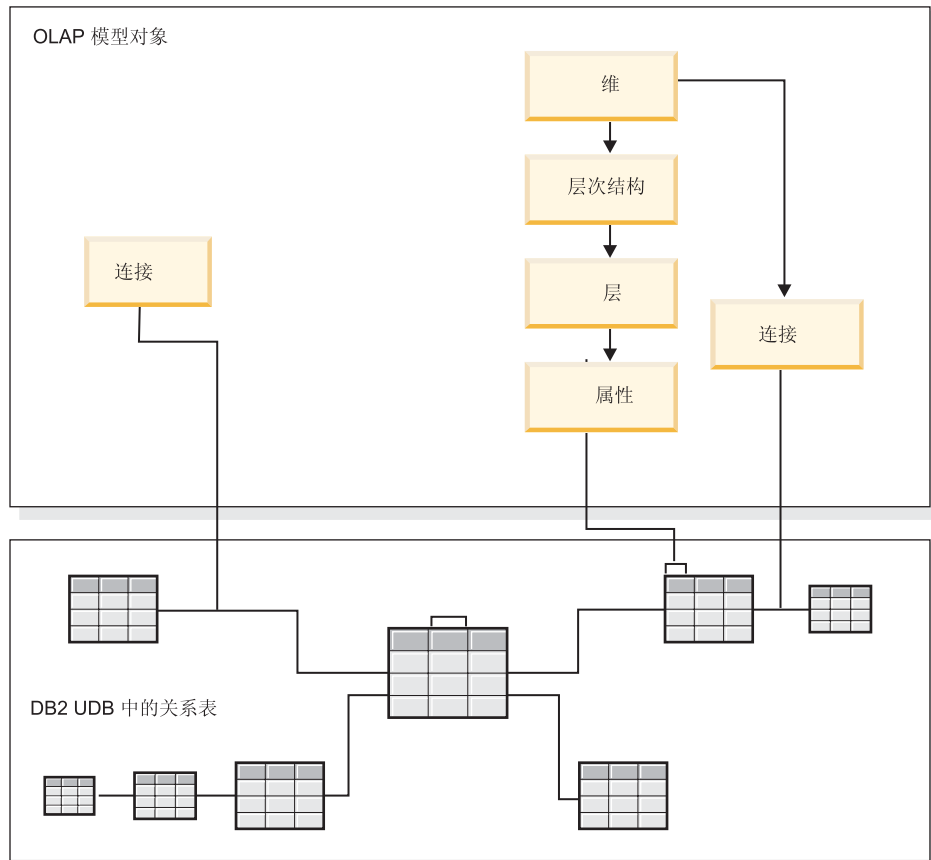


图 2. 维. 如何从关系表构建维

所有维都连接至立方体模型中基于星型模式或雪花模式的事实对象。“连接”（Join）可以连接表以创建事实对象或维。在立方体模型中，连接可以将事实对象连接至维。这些维引用其相应层次结构、层、属性和相关连接。事实对象引用其量度、属性和相关连接。第 12 页的图 3 显示元数据对象在立方体模型中如何组合在一起以及映射至关系雪花模式。

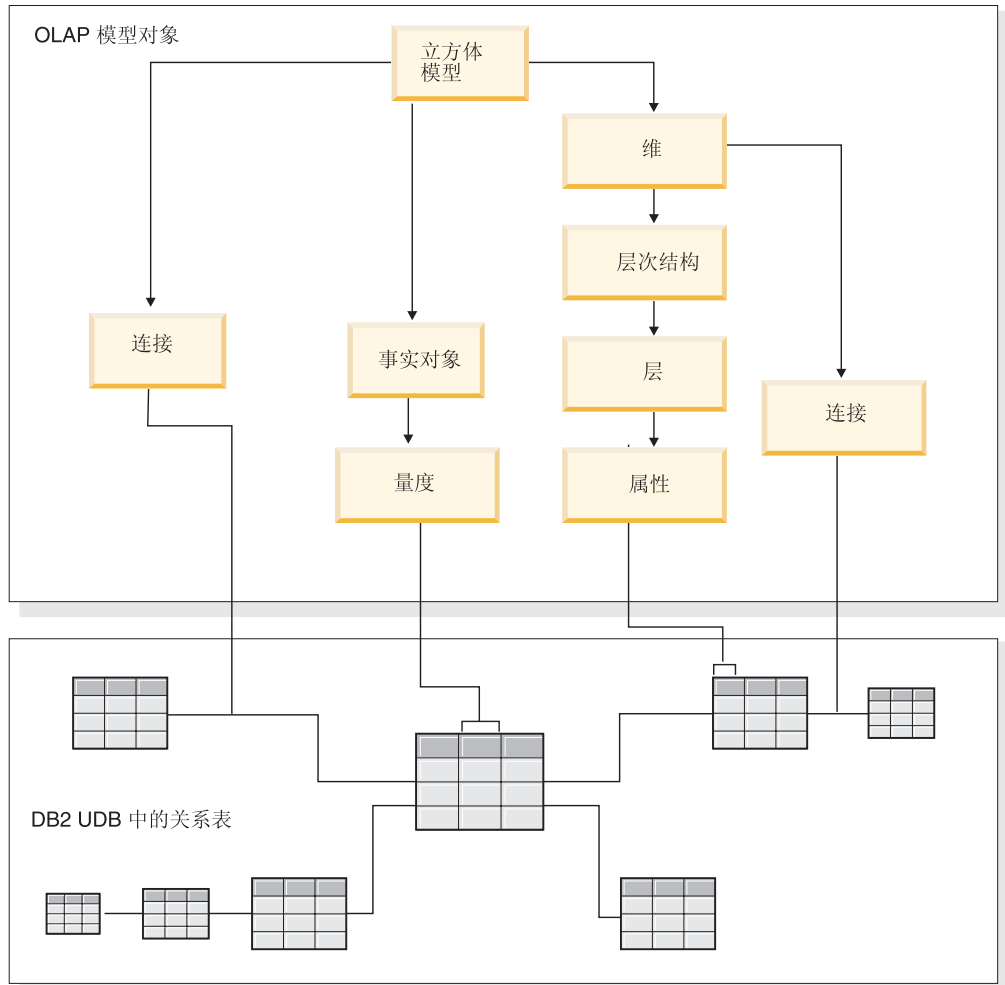


图 3. 立方体模型. 元数据对象如何组合在一起以及映射至关系雪花模式

您可以重新使用立方体模型的组件为特定应用程序创建更为精确的立方体。立方体是最精确的元数据对象，并且是最接近 OLAP 概念上的立方体的对象。立方体是立方体模型的特定实例或子集。立方体具有从父立方体模型派生的一组类似但限制性更多的特定元数据对象，包括立方体维、立方体层次结构、立方体层和立方体事实对象。立方体的每个立方体维只能定义一个立方体层次结构，但立方体模型的维则可以定义多个层次结构。由于立方体和立方体模型之间的这种结构差异，因此您可以使用单个 SQL 语句检索大多数立方体。

## 映射至关系表的立方体模型的示例

立方体模型及相应元数据集合是为基于雪花模式的 CVSAMPLE 数据库构建的。

第 13 页的图 4 显示包含 Sales 事实表以及 Store、Location、Time、Product、Line 和 Family 维表的雪花模式。Market 维具有两个维表，Store 作为主维表，而 Location 作为支架维表。Production 维具有三个维表，Product 作为主维表，而 Line 表和 Family 表作为支架维表。

每个主维表（Store、Time 和 Product）中的主键都连接至 Sales 事实表中的相应外键。例如，Store.StoreID = Sales.StoreID，Time.TimeID = Sales.TimeID，而 Product.ProductID = Sales.ProductID。

在雪花模式中，每个维表中的主键都连接至另一个维表中的相应外键。例如，Location.PostalcodeID = Store.PostalcodeID，Family.FamilyID = Line.FamilyID，而 Line.LineID = Product.LineID。

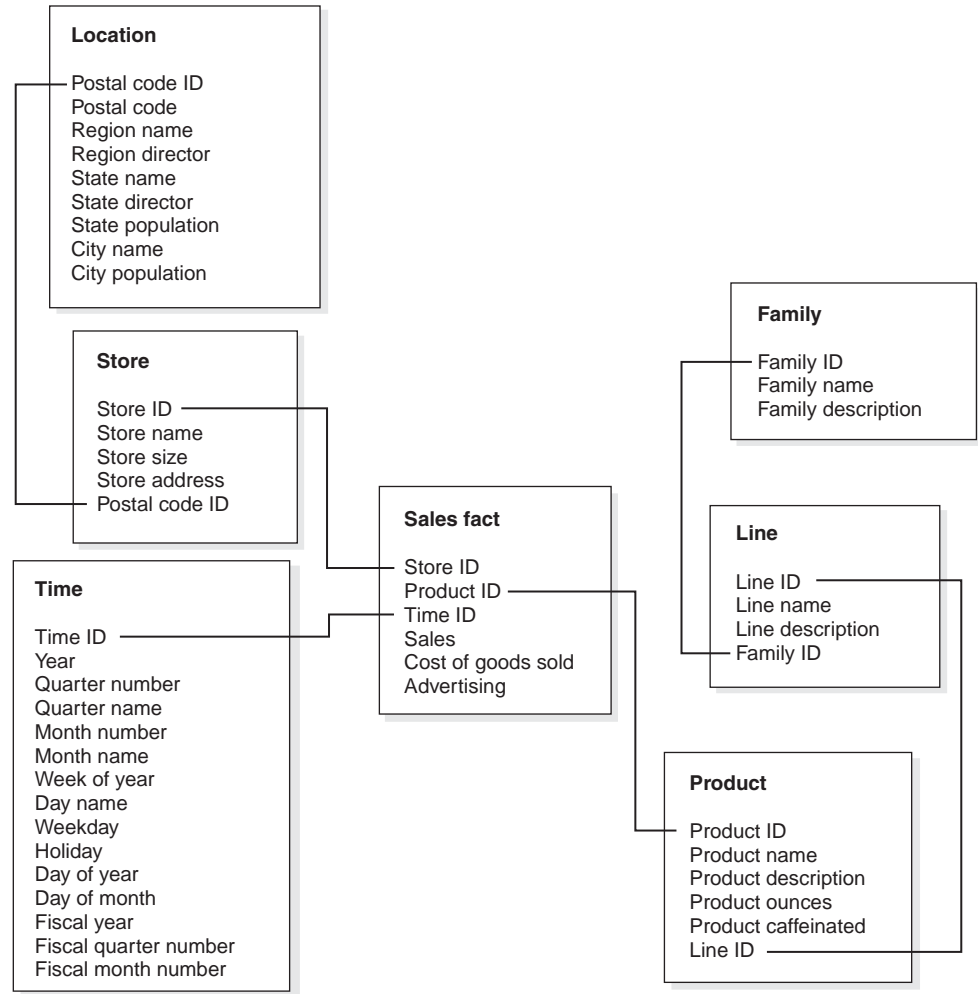


图 4. 雪花模式. 基于 CVSAMPLE 雪花模式的示例

基于 CVSAMPLE 雪花模式的立方体模型是根据描述从 Sales 事实表中聚集的关系数据的 Sales 事实对象构建的。量度描述如何从 Sales 表的列中计算数据。事实对象还包括与事实表中的外键相对应的属性，这些外键用于将维连接至事实对象。在此示例中，Sales 事实对象有七个量度：Sales、Cost of goods sold、Advertising、Total expense、Advertising-sales correlation、Profit 和 Profit margin。Sales 事实对象具有三个属性：Store ID (Salesfact)、TimeID (Salesfact) 和 ProductID (Salesfact)。

维连接至立方体模型中的事实对象，就象维表连接至星型模式中的事实表。关系表中的数据列由维引用的属性对象表示。

Product 维引用下列属性：

- Family ID
- Family name
- Family description
- Line ID
- Line name
- Line description
- Product ID
- Product name
- Product description
- Product ounces
- Product caffeinated

Time 维引用下列属性:

- Year
- Quarter name
- Quarter number
- Month name
- Month number
- Time ID
- Day of month
- Day name
- Day of week
- Holiday
- Weekday
- Fiscal year
- Fiscal quarter name
- Fiscal quarter number
- Fiscal month.

Product 维引用下列属性:

- Region name
- Region director
- State name
- State director
- State population
- City name
- City population
- Postal code ID
- Postal code
- Store ID
- Store name

- Store size
- Store address.

创建连接以将每个维连接至事实对象。本示例中的三个连接为 Product、Time 和 Store。图 5 显示了本示例中描述的立方体模型。

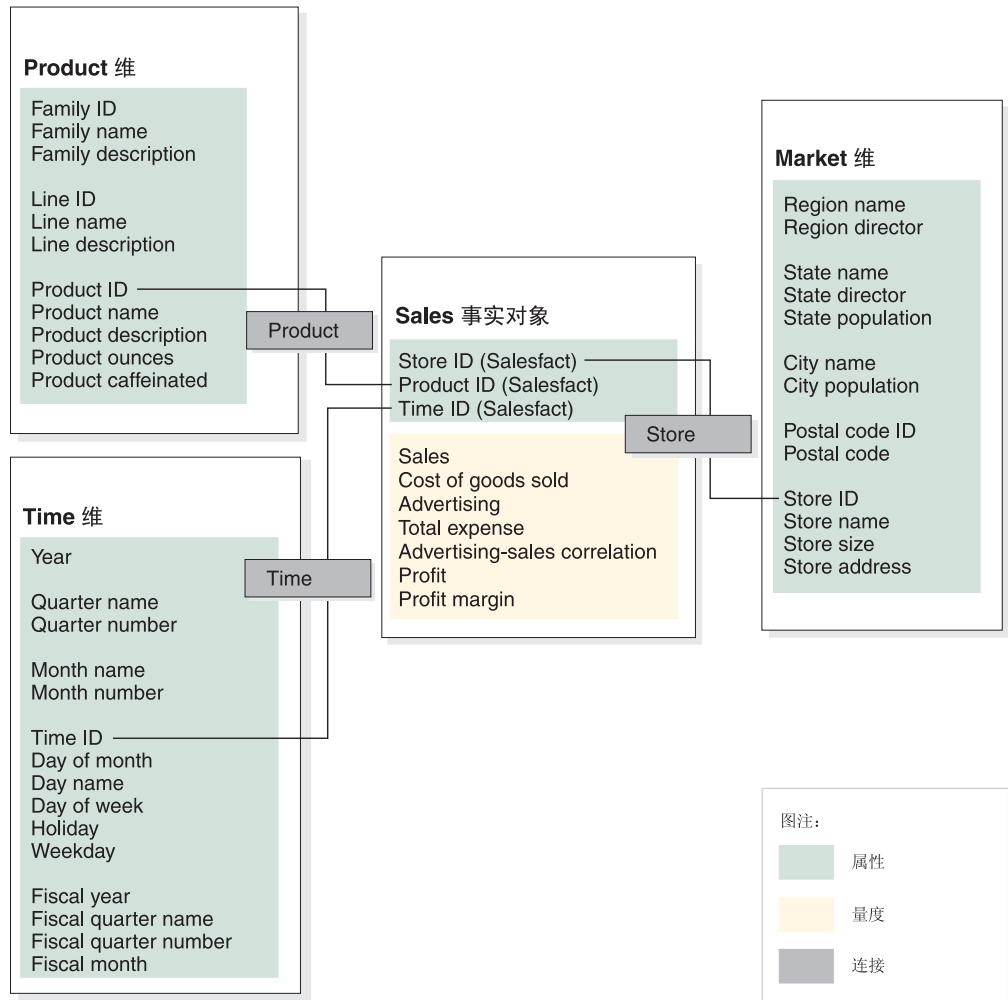


图 5. 立方体模型. 基于 CVSAMPLE 星型模式的示例立方体模型

层次结构存储有关某个维中的组合到层中的属性如何彼此相关以及如何构造的信息。作为元数据对象，层次结构提供了计算和导航维的方式。每个维都具有相应的层次结构（包含组合相关属性的层）。在立方体模型中，每个维可以具有多个层次结构。

Product 层次结构将所有属性都包括在 Product 维中（如第 16 页的图 6 中所示）。Product 维中的属性被分到三个层中。Family 层是 Product 层次结构的顶层。Family 层将 Family ID 包括为层键属性、将 Family name 包括为缺省属性并将 Family description 包括为相关属性。第二层（Line 层）将 Line ID 包括为层键属性、将 Line name 包括为缺省属性并将 Line description 包括为相关属性。底层（Product 层）将 Product ID 包括为层键属性、将 Product name 包括为缺省属性并将 Product description、Product ounces 和 Product caffeinated 包括为相关属性。

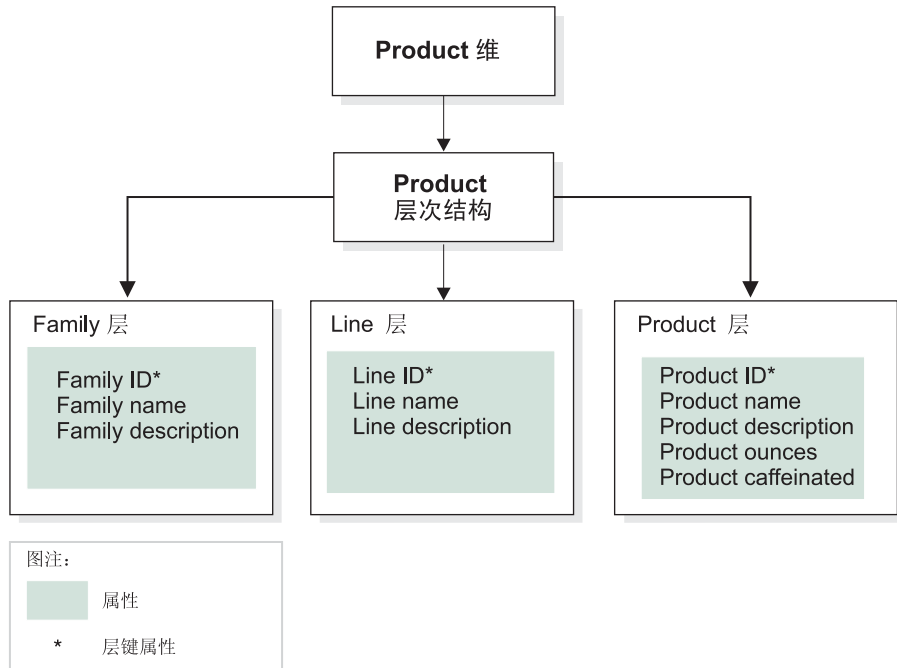


图 6. 维. 基于 CVSAMPLE 数据库的 Product 维的示例维

您还可以为立方体模型构建一个或多个立方体。CVSAMPLE 数据库有两个立方体，但是此处只描述 General sales 立方体。General sales 立方体显示在第 17 页的图 7 中。立方体事实对象从立方体模型事实对象引用量度的子集（Sales、Cost of goods sold、Advertising 和 Total expense）。该立方体具有三个立方体维，每个立方体维都引用立方体模型中的三个维的其中一个。Product 立方体维具有一个 Product 立方体层次结构，它包含引用 Family、Line 和 Product 层的立方体层。Market 立方体维具有 Market 立方体层次结构，该层次结构包含引用 Region、State、City、Postal code 和 Store 层的立方体层。Time 立方体维具有一个 Time 立方体层次结构，它包含引用包括 Year、Quarter 和 Month 的层的子集的立方体层。该立方体的每个立方体维只定义一个立方体层次结构。（立方体的每个立方体维只能具有一个立方体层次结构。）



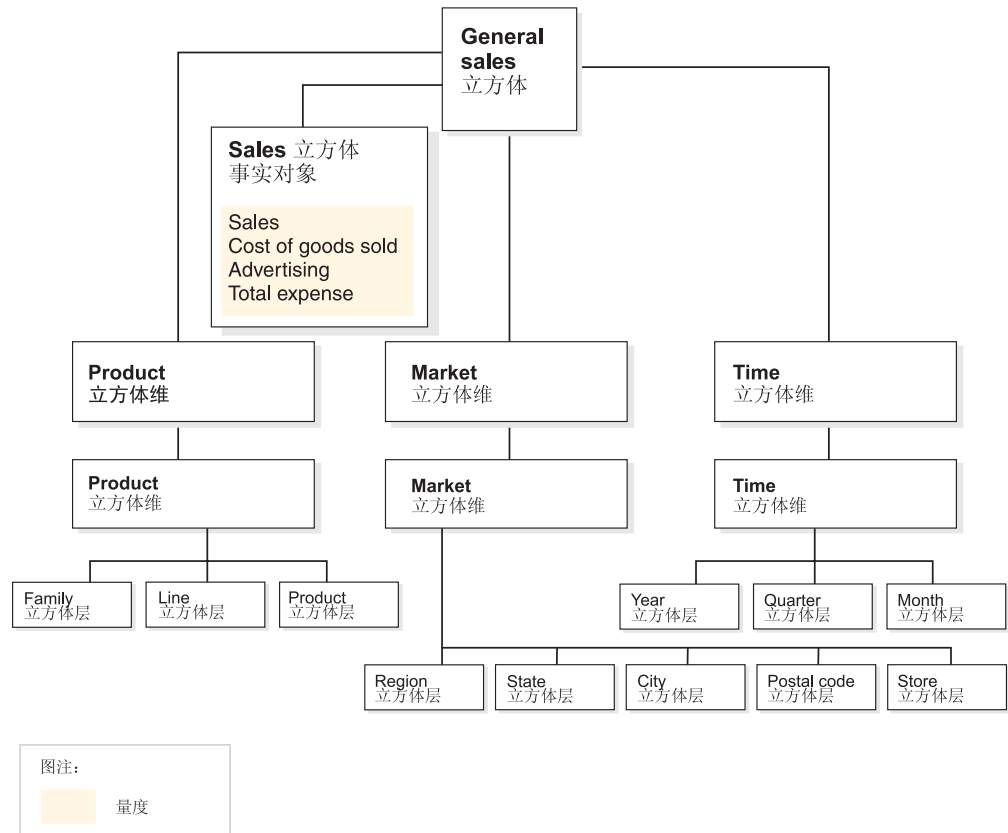


图 7. 立方体. 基于 CVSAMPLE 数据库中的 General sales 立方体的示例立方体

## 公共元数据对象属性

每个元数据对象都具有一组公共属性和特定于元数据对象的属性。公共属性用于标识元数据对象实例，描述元数据对象实例的用法或角色以及跟踪元数据对象实例更改。元数据对象是以与命名其它 DB2 对象相同的方式通过使用模式命名的。如果不想使用元数据对象的缺省用户名模式，则需要使元数据对象完全符合所需的模式名。

下表描述了所有元数据对象都存在的公共属性。

表 1. 公共元数据对象属性

属性	描述
名称	元数据对象的名称。
模式	拥有元数据对象的模式。
业务名	向用户显示的名称。此名称可在图形用户界面中作为对用户更有意义的名称使用。
注释	元数据对象性质或用法的文本描述或注释。
创建时间	创建元数据对象的时间。
创建者	定义元数据对象的用户（模式）。
修改时间	上一次修改元数据对象的时间。
修改者	执行修改的用户（模式）。

除了一组公共属性之外，每个元数据对象还具有一组特定属性。这些特定属性描述定义元数据对象的组件和性质。有关特定于每个元数据对象的属性的信息，请参阅该元数据对象的主题。

### 元数据对象命名约定

DB2 UDB 提供了两种不同的命名约定用于命名对象：常规和定界。对于元数据对象，定界约定在命名对象及引用 DB2 表和列时使用。定界约定允许在名称中使用混合大小写的字符、空格以及特殊字符（如本地语言字符）。完整的字符集由其中存储元数据对象的数据库的代码页确定。

以下约定适用于元数据对象：

表 2. 元数据对象的命名约定

对象	约定
模式	<ul style="list-style-type: none"><li>• 长度：1-30 个字节</li><li>• 限制名称：模式名称不能为 <i>SESSION</i> 或以 <i>SYS</i> 开头。只有大写名称才受限制。</li></ul>
元数据对象的名称	<ul style="list-style-type: none"><li>• 长度：1-128 个字节</li><li>• 无其它限制。</li></ul>
元数据对象的业务名	<ul style="list-style-type: none"><li>• 长度：1-128 个字节</li><li>• 无其它限制</li></ul>
元数据对象的注释	<ul style="list-style-type: none"><li>• 长度：0-254 个字节</li><li>• 无其它限制</li></ul>
引用列中使用的表模式	<ul style="list-style-type: none"><li>• 长度：1 至 128 个字节</li><li>• 无其它限制</li></ul>
引用列中使用的表名	<ul style="list-style-type: none"><li>• 长度：1 至 128 个字节</li><li>• 无其它限制</li></ul>
引用列中使用的列名	<ul style="list-style-type: none"><li>• 长度：1-128 个字节</li><li>• 无其它限制</li></ul>

---

## 立方体模型

DB2 Cube Views 立方体模型是逻辑星型模式或雪花模式的表示法，并将相关维对象组合在中央事实对象周围。

每个维可以具有多个层次结构。有关如何连接由事实对象和维使用的表的结构信息由立方体模型引用。同时存储在立方体模型中的还有足够用于构造 SQL 查询和检索 OLAP 数据的信息。了解立方体模型并能够显示特定维的多个视图的其它报表和 OLAP 工具可能受益于使用立方体模型。

立方体模型定义复杂的关系集合，并可用于选择性地应用程序显示相关事实对象和维。将维连接至中心事实对象的每个连接对象与相应的维作为一个集存储在一起。立方体模型组件的子集可由多个立方体用于不同的分析用途。

您可以通过使用立方体模型向导在 OLAP 中心中创建空立方体模型。空立方体模型不具有事实对象或任何维。使用 OLAP 中心中的向导，您可以通过创建事实对象和一个或多个维来完成立方体模型。您也可以使用快速启动向导创建完整的立方体模型。DB2 Cube Views 将在打开优化顾问程序时验证立方体模型。要优化立方体模型，该立方体模型必须包含下列必需组件：

- 事实对象
- 至少一个维
- 为至少一个维定义的层次结构
- 现有事实对象和维之间的连接
- 引用现有表列的属性

特定于立方体模型的属性在下表中描述。

表 3. 立方体模型属性

属性	描述
事实对象	立方体模型中使用的事实对象
(维和连接)集	立方体模型中使用的维以及它们对应的连接

---

## 事实对象

事实对象在立方体模型中用作星型模式的中心并组合与特定应用程序有关的相关量度。

事实对象引用在事实-到-维连接中使用的属性以及用于跨多个数据库表映射附加量度的属性和连接。可以在特定属性上连接多个关系事实表以映射附加相关量度。因此，除一组量度之外，事实对象还存储一组属性和一组连接。

您可以使用 OLAP 中心中的事实向导来创建事实对象。在事实向导中指定一个或多个事实表以及任何必需的连接、量度和量度的聚集。

事实对象的特定属性在下表中描述。

表 4. 事实对象属性

属性	描述
量度集	事实对象中所有相关量度的集合
属性集	事实对象中使用的所有属性的集合
连接集	连接所有指定量度和属性所需的所有连接的集合

---

## 维

维提供了对一起描述量度的某个特征的相关属性集进行分类的方法。维用于立方体模型中依据逻辑类别（如 Region、Product 或 Time）对事实对象中的数据进行组织。

维引用零个或多个层次结构。层次结构描述组合到层中的被引用属性的关系和结构，并提供遍历维的导航和计算方法。

相关属性和组合这些属性所需的连接都是在维的属性中定义的。

维还具有描述维是否面向时间的类型。例如，称为 Time 且包含诸如 Year、Quarter 和 Month 之类的层的维就是 Time 类型。称为 Market 且包含诸如 Region、State 和 City 的另一个维就是 Regular 类型。应用程序可以使用类型信息以相应的智能方式执行与时间相关的功能。

您可以使用 OLAP 中心中的维向导在立方体模型的上下文中或在不引用立方体模型的情况下创建新维。可通过使用“添加维”向导将现有维添加至立方体模型以在若干立方体模型间共享维。

维的特定属性在下表中描述。

表 5. 维属性

属性	描述
属性集	维中使用的所有属性的集合。
连接集	连接所有指定属性所需的所有连接的集合。此处仅指定连接维表所需的连接。
层次结构集	应用于维的层次结构的集合。
层集	维引用的层的集合。
类型	维类型，可以是 Regular 或 Time

## 层次结构

层次结构定义立方体模型的维中按层分组的一组属性之间的关系。这些关系提供了一种遍历维的导航和计算方法。可以为立方体模型的维定义多个层次结构。

层次结构类型描述层次结构内各层之间的关系。支持下列四种层次结构类型：

**平衡** 具有一致深度的包含有意义的层和分支的层次结构。每个层的逻辑父代都在其正上方的层中。平衡层次结构可以表示其中每个层（如 Year、Quarter 和 Month）的含义和深度均一致的时间。由于每个层表示相同类型的信息并且每个层在逻辑上均相等，因此它们是一致的。图 8 显示了一个平衡的时间层次结构的示例。

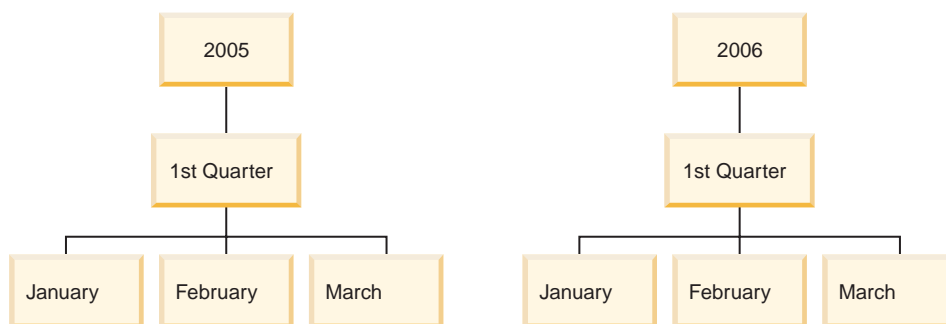


图 8. 平衡层次结构. 平衡层次结构的示例

**未平衡** 一种层次结构，具有一致的父子关系但又具有逻辑上不一致的层。同时，层次结构分支还可能具有不一致的深度。未平衡层次结构可以表示组织图表。例如，第 21 页的图 9 显示层次结构顶层的 Chief Executive Officer (CEO) 以及在下面可能分支出来的至少两个人员（包括 Chief Operating Officer 和 Executive

Secretary)。Chief Operating Officer 还具有更多的人员分支，但 Executive Secretary 则没有。该层次结构的两个分支上的父子关系一致。但是，两个分支的层在逻辑上不相等。例如，Executive Secretary 不是 Chief Operating Officer 的逻辑等价值。

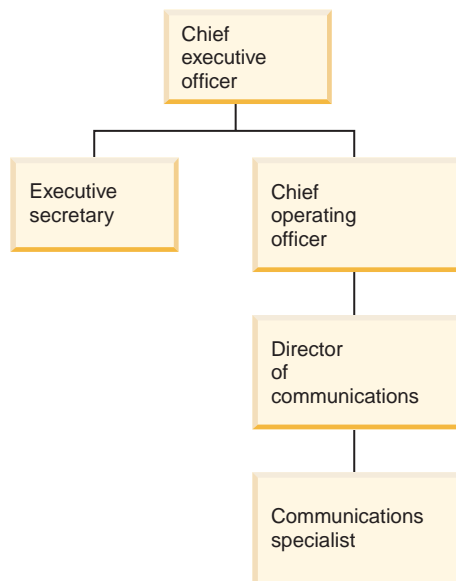


图 9. 不平衡层次结构. 不平衡层次结构的示例

**未对齐** 一种层次结构，其中每个层都具有一致的含义，但分支却由于分支层中的至少一个成员属性空缺而具有不一致的深度。未对齐层次结构可表示地理层次结构，其中每个层（如城市或国家或地区）的含义使用方式一致，但层次结构的深度不同。第 22 页的图 10 显示定义了 Continent、Country、Province/State 和 City 层的地理层次结构。一个分支将 North America 作为 Continent、将 United States 作为 Country、将 California 作为 Province/State 并将 San Francisco 作为 City。但是，当一个成员并非在所有层中均具有条目时，层次结构将变为未对齐。例如，另一分支将 Europe 作为 Continent、将 Greece 作为 Country 并将 Athens 作为 City，但因为对于此示例的业务模型 Province/State 层不适用于 Greece 而使该层没有条目。在此示例中，Greece 和 United States 分支下行至不同的深度，并创建未对齐层次结构。

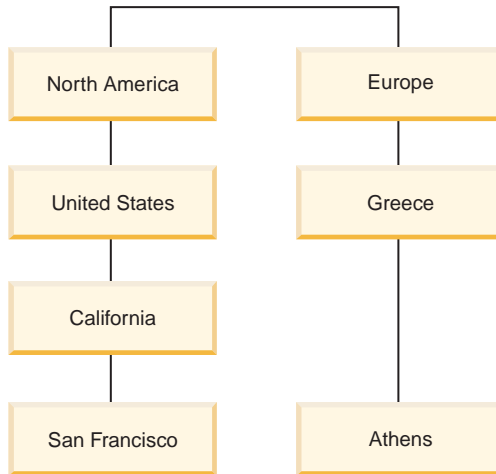


图 10. 未对齐层次结构. 未对齐层次结构的示例

**网络** 一种层次结构，其中未指定各层的顺序，但各层具有语义含义。例如，图 11 显示了描述产品属性（如 Color、Size 和 PackageType）的网络层次结构。由于层没有固有的父子关系，因此层的顺序并不重要。装饰品公司可能具有这样的成员条目，例如 White 表示 Color、Small 表示 Size 以及 Shrink wrap 表示 PackageType。第二个成员条目可能用 Red 表示 Color、Large 表示 Size 以及 Box 表示 PackageType。

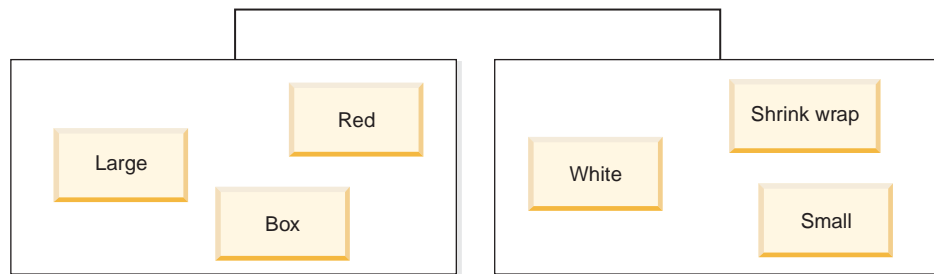


图 11. 网络层次结构. 网络层次结构的示例

层次结构还指定层次结构的部署机制。部署机制定义如何解释层次结构中各个层的属性。支持以下两种部署机制：

**标准** 使用层次结构的层定义，其中层次结构中的每个层都被用作部署中的一项。例如，Time 维的平衡层次结构将依据每个定义的层（包括 Year、Quarter 和 Month）进行组织。标准部署可用于全部四种层次结构类型。表 6 显示如何使用标准部署组织 Time 维的某些平衡层次结构属性

表 6. 标准部署. Time 维的平衡层次结构的标准部署

Year	Quarter	Month
2003	Qtr 1	Jan
2003	Qtr 1	Feb
2003	Qtr 1	Mar
2004	Qtr 1	Jan
2004	Qtr 1	Feb

表 6. 标准部署 (续). Time 维的平衡层次结构的标准部署

Year	Quarter	Month
2004	Qtr 1	Mar

**递归** 使用层次结构的层之间固有的父子关系。使用递归部署的未平衡层次结构表现为父子层对。例如，表 7 显示了未平衡层次结构的层对，该层次结构描述了第 21 页的图 9 中所示的组织图表。父子层对包括：Chief Executive Officer 和 Executive Secretary、Chief Executive Officer 和 Chief Operating Officer、Chief Operating Officer 和 Director of Communications 以及 Director of Communications 和 Communications Specialist。递归部署只能与未平衡层次结构配合使用。

表 7. 递归部署. Organization 维的未平衡层次结构的递归部署

父属性	子属性
Chief executive officer	Executive secretary
Chief executive officer	Chief operating officer
Chief operating officer	Director of communications
Director of communications	Communications specialist

您可以在 OLAP 中心中使用“层次结构”向导创建层次结构。在创建维之后，您可以定义维的层次结构。

层次结构对象的属性将在下表中描述。

表 8. 层次结构属性

属性	描述
层的已排序集合	层次结构中自顶向下的一个或多个层的已排序列表。
类型	层次结构类型可以是：平衡、未平衡、未对齐和网络
部署	层次结构部署可以是标准和递归

## 层

层由作为层次结构排序中的一个逻辑步骤一起工作的一组属性组成。层包含可在该层中的一个或多个角色中起作用的一个或多个相关属性。层中各个属性之间的关系通常是使用功能相关性定义的。

层可以通过下列三个方法的其中一个来使用属性：

### 层键属性

一个或多个属性的值唯一地标识该层的每个实例。例如，对于 City 层，定义层键属性最简单的方法是使用标识列（例如，City ID），该列被保证为唯一的。因为在若干省份和国家或地区中城市名称不一定是唯一的，国家或地区中城市名称不一定是唯一的。但是，因为 Country name、State name 和 City name 属性的组合可以唯一地定义城市，因此可以将这三个属性的集合作为层键属性包括。因为 DB2 Cube Views 会创建功能相关性来根据定义的那些会创建功能相关性来根据定义的那些层改进优化性能。

## 缺省属性

一个属性的值可以由报表应用程序显示以便为数据中的某个层的每个实例提供有意义的名称。缺省属性是必需的，而且功能必须由层键属性确定。例如，如果层键属性被定义为 City ID，则 City ID 列的值在报表中的作用不大。可以将 City name 定义为要在报表中显示的缺省属性以使显示的数据起作用。可以将层键属性集中的某个属性用作缺省属性。例如，如果将 City 层的层键属性定义为 Country name、State name 和 City name 的集合，可以将缺省属性定义为 City name。

## 相关属性

零个或多个属性的集合，提供有关定义为层键属性的那些层的实例的其它信息。所有相关属性的功能必须由层键属性确定。例如，城市层可能具有一些相关属性（如，市长、城市人口、城市位置和城市描述）。

可以使用 OLAP 中心中的层向导单独创建新层或在层次结构的上下文中创建新层。可以跨立方体模型共享维中的层。当创建层时，DB2 Cube Views 在缺省情况下为该层中的属性创建功能相关性，以便缺省属性和相关属性在功能上依赖于该层中的层键属性。优化顾问程序使用功能相关性来建议最佳摘要表，而 DB2 优化器则使用功能相关性来正确地对 SQL 查询进行路由。

有许多方法可用于对构建使用层的层次结构建立模型。无论遵循理想还是不理想的建模技巧，都应定义每个层以使层键属性按功能确定缺省属性和相关属性。层中的层键属性必须唯一地标识该层中的值。必须定义应在每个层的各个属性之间创建功能相关性。功能相关性可由优化顾问程序用来建议摘要表，也可由 DB2 优化器用来正确地对 SQL 查询进行路由。功能相关性使优化顾问程序向导能够建议较小的摘要表以便能够较快地返回查询结果。

## 理想建模

理想情况是，每个维的关系数据都存储在包含维中每个层的标识列的单个维表中，并且每个标识列都唯一地标识该层中的值。例如，对于 Region 维您可能有单个维表，该维表包含下列各列：

表 9. 维表的理想建模

### 理想 Region 维表中的列

---

City ID (主键)

---

City name

---

City mayor

---

State ID

---

State name

---

State governor

---

Country ID

---

Country name

---

可以创建包含 Country、State 和 City 层的层次结构。在每个层中，可以定义被定义为层键属性的标识列与任何相关属性之间的功能相关性（如第 25 页的表 10 中所示）。



表 10. 层次结构的理想建模型

层	层键属性	层相关属性
Country	Country ID	Country name
State	State ID	State name State governor
City	City ID	City name City mayor

功能相关性是在下列属性对之间创建的:

- Country ID -> Country name
- State ID -> State name 和 State governor
- City ID -> City name 和 City mayor

因为 City ID 是主键且应该已定义约束, 所以 DB2 Cube V相关性, 原因是 City ID 是主键且应该已定义约束。

### 非理想建模

如果层次结构中的每个层没有唯一标识数据列, 则在为每个层定义层键属性时必须更加仔细, 以便创建有效的功能相关性。例如, 对于 Region 维您可能有单个维表, 该维表包含下列各列:

表 11. 维表的非理想建模

非理想 Region 维表中的列
City ID (主键)
City name
City mayor
State name
State governor
Country name

可以创建包含 Country、State 和 City 层的层次结构 (如理想建模示例中所示)。但是, 必须更加仔细地定义层键属性以确保层中的每行都可以是唯一定义的。例如, City name 并未唯一定义 City 层, 原因是在美国和英国都有一个城市名为 Leicester。唯一地定义 City 层的仅有方法是使用 Country name、State name 和 City name 属性的组合 (如表 12中所示)。

表 12. 层次结构的非理想建模

层	层键属性	层相关属性
Country	Country name	
State	Country name State name	State governor
City	Country name State name City name	City mayor

功能相关性是在下列两个属性组合之间创建的:

- Country name、State name 和 City name → City mayor
- Country name 和 State name → State governor

层对象的属性将在下表中描述。

表 13. 层属性

属性	描述
层键属性集	组合在一起的一个或多个属性的集合，唯一地定义层。
一个缺省属性	一个必需的缺省属性，它功能上由层键属性确定并可由报表应用程序用来显示有意义的数据。
相关属性集	零个或多个属性的集合，包含有关该层的更多信息并且在功能上由层键属性确定。
功能相关性	指定层是否具有相应的 DB2 UDB 功能相关性的布尔值 (Yes 或 No)。
功能相关性的名称	如果功能相关性设置为“是”，则此属性包含 DB2 UDB 功能相关性的名称 (最大长度为 18 字节)。如果功能相关性设置为“否”，则此属性被忽略。

## 量度

量度定义量度实体并且在事实对象中使用。

量度在立方体模型中的一组维的上下文中变得有意义。例如，金额为 300 的收入在单独出现时并没有意义。在将 revenue 量度放在维 (如 Region 和 Time) 的上下文中时，量度将变得有意义：New York 在一月份的收入为 300。常见的量度示例为 Revenue、Cost 和 Profit。

量度是通过组合以下两个属性定义的：SQL 表达式列表和聚集列表。表列、属性和量度被映射至模板以构建 SQL 表达式。然后，生成的 SQL 表达式将用作量度第一个聚集函数的输入。如果量度具有多个聚集，则将按列示的顺序执行聚集函数，其中每个后续的聚集都将前一个聚集的结果作为其输入。如果量度的 SQL 表达式只引用其它量度，则聚集函数是可选的。由于引用的量度提供聚集，因此聚集函数是可选的。

量度的 SQL 表达式是通过组合以下两个属性创建的：模板以及列、属性和量度的列表。模板使用标记符号，其中 `$$$n` 为标记并且 n 从列表中引用特定列、属性或量度。列、属性和量度的列表已排序，并且列、属性或量度在列表中的位置与标记 n 的值相对应。

SQL 表达式用作第一个聚集的输入。每个聚集指定应用于对应维列表的函数。聚集函数可以是底层的数据库支持的任何聚集函数。DB2 Cube Views 中支持以下聚集函数：

- AVG
- CORRELATION
- COUNT
- COUNT\_BIG

- COVARIANCE
- MAX
- MIN
- REGRESSION 函数（全部 9 种类型）
- STTDEV
- SUM
- VARIANCE

每个维只能由量度对象聚集一次。量度必须具有一个包含空的维列表的聚集，而任何其它聚集必须每个都具有显式的维列表。空的维列表的聚集被应用至未由另一聚集特定使用的立方体模型中的所有维。

直接映射至列的简单量度的示例为 Revenue。可以为带有以下三个维的立方体模型创建 Revenue 量度：Product、Market 和 Time。Revenue 具有 SQL 表达式模板 `template = "${$1}"`，它表示至列、属性和量度的单项列表中指定的列的简单映射，其中 `list = "Column Fact.Rev"`。聚集列表为 (SUM, <NULL>)，其中 SUM 是聚集函数，而 <NULL> 是空的维列表。SQL 表达式用作 SUM 聚集函数的输入，它将生成 SQL 表达式：`SUM(Fact.Rev)`。

更复杂的量度 Profit 可能具有 SQL 表达式模板 `template = "${$1} - {$$2}"`，其中属性、列和量度的列表为 `list = "Measure Revenue, Column Fact.Cost"`。如果将标记替换为正确的引用，SQL 表达式将变为 "Revenue - Fact.Cost"。如果将 Revenue 量度引用扩展至其列引用，SQL 表达式将变为：`"Fact.Rev - Fact.Cost"`。Profit 量度的聚集列表为：(SUM, <NULL>)。profit SQL 表达式被用作 SUM 聚集函数的输入，所以 Profit 量度的 SQL 为 `SUM(Fact.Rev - Fact.Cost)`。

如果量度具有需要两个或更多参数的聚集函数（如 CORRELATION），则量度将具有两个或更多 SQL 表达式。

量度还具有基于 SQL 数据类型的数据类型。DB2 Cube Views 自动确定量度的数据类型。在由模式完全限定时，每个名称在量度和属性中必须是唯一的。

OLAP 中心隐藏了元数据对象定义的许多复杂性。在 OLAP 中心中，您不必显式定义量度的 SQL 表达式列表或聚集列表。如果要创建直接映射至列、属性或其它量度的量度，则在事实向导或“事实属性”窗口中创建量度时，您需要选择源。如果要创建计算的量度，则可以使用“SQL 表达式构建器”窗口来创建源表达式。“SQL 表达式构建器”提供了可用列、属性、量度、运算符以及函数和常量的列表。在“量度属性”窗口中，您可以查看量度源数据的数据类型以及聚集源数据之后量度的数据类型。

下表描述定义量度的特定属性。在创建量度时，OLAP 中心将定义所有这些属性。

表 14. 量度属性

属性	描述
SQL 表达式列表（模板、[（列、属性和量度的列表）]）	用作量度第一个聚集函数的输入的 SQL 表达式的列表。每个 SQL 表达式都具有模板以及列、属性和量度的已排序列表。
聚集列表（函数和维列表）	指定如何计算量度的聚集的列表。每个聚集都具有 SQL 聚集函数以及要对其应用函数的维的可选列表。

表 14. 量度属性 (续)

属性	描述
数据类型 (模式、名称、长度、小数位和可空)	确定量度的数据类型。数据类型基于 SQL 数据类型,并由数据类型模式、名称、长度、小数位和可空组成。只有在该模式是 SYSIBM 之外的模式时,OLAP 中心才会显示它。

## 属性

属性表示数据库表列的基本抽象概念。属性包含可能是至表列或较复杂表达式的简单映射的 SQL 表达式。这些较复杂的表达式可组合多个列或属性并可使用包括用户定义的函数的所有 SQL 函数 (如果需要的话)。

DB2 Cube Views 的 OLAP 中心隐藏了属性对象定义的许多复杂性。在 OLAP 中心中,您不必显式定义属性的表达式模板或参数列表。如果要创建直接映射至列的属性,则在维向导或“维属性”窗口中创建属性时,您需要选择源列。如果要创建计算属性,则可以使用“SQL 表达式构建器”窗口来创建源表达式。“SQL 表达式构建器”提供了可用属性、列、运算符、函数以及常量的列表。

如果要在不使用 OLAP 中心的情况下创建属性,则必须将属性的 SQL 表达式定义创建为两个属性的组合,包括模板以及列和属性的列表。模板使用标记注释,其中 `$$$n` 是标记而 `n` 引用列表中的特定列或属性。列和属性的列表已排序,并且列或属性在列表中的位置与标记 `n` 的值相对应。例如,模板 `template = "${$1} || ' ' || {$$2}` 可与相应列表 (如 `list = "Column CUSTOMER.FIRSTNAME, Attribute LastName"`) 配合使用以并置客户的姓和名并在姓与名之间加上空格。当模板标记被替换为正确的列表引用时,SQL 表达式为 `"Customer.FirstName || ' ' || LastName"`。属性引用被进一步扩展为列引用以形成 SQL 表达式: `"Customer.FirstName || ' ' || Customer.LastName"`。

在定义属性的 SQL 表达式中使用其它属性时,这些其它属性不能形成属性引用循环。例如,如果属性 A 引用属性 B,则属性 B 不能引用属性 A。

在由模式完全限定时,每个名称在数据库中的所有其它属性和量度的名称中必须是唯一的。

下表描述定义属性的具体特性。在创建属性对象时,OLAP 中心将定义所有这些属性。

表 15. 属性特性

属性	描述
SQL 表达式模板	定义属性的 SQL 表达式。模板通过使用 <code>\$\$\$n</code> 符号 (其中 <code>n</code> 是与列和属性的列表对应的序号) 引用列和属性。
SQL 表达式的列和属性的列表	组成属性的所有列和属性的已排序列表。这些列和属性将按 SQL 表达式模板中指定的方式应用。
数据类型 (模式、名称、长度、小数位和可空)	确定属性的数据类型。数据类型基于 SQL 数据类型,并由数据类型模式、名称、长度、小数位和可空组成。只有在该模式是 SYSIBM 之外的模式时,OLAP 中心才会显示它。

## 属性关系

属性关系概括性地描述属性的关系，但属性关系不是立方体模型的一部分。

关系由下列属性组成：

- 左边的属性和右边的属性
- 类型
- 基数
- 可能的功能相关性

类型描述右边属性的角色是什么（就左边属性而论）。存在两种可能的类型：“描述性的”和“关联的”。

### 描述性的

指定右边属性是左边属性的描述符。例如，右边属性 `ProductName` 描述左边属性 `ProductCode`。

**关联的** 指定右边属性与左边属性相关联，但不是左边属性的描述符。例如，右边属性 `CityPopulation` 与 `CityID` 相关联，但不是它的描述符。

基数描述左边的属性和右边的属性的实例如何相关。可以对属性关系使用以下基数：

- 1:1** 右边每个属性实例最多在左边存在一个属性实例，而左边每个属性实例最多在右边存在一个属性实例。
- 1:多** 右边每个属性实例最多在左边存在一个属性实例，而左边每个属性实例在右边存在任意数目的属性实例。
- 多:1** 右边每个属性实例在左边存在任何数目的属性实例，而左边每个属性实例最多在右边存在一个属性实例。
- 多:多** 右边每个属性实例在左边存在任意数目的属性实例，而左边每个属性实例在右边存在任意数目的属性实例。

功能相关性说明属性关系是否定义两个属性之间的功能关系。指定属性关系是功能相关性意味着保证左边属性的每个实例将确定右边属性的实例。DB2 Cube Views 将不在属性关系描述的属性之间创建功能相关性，不管您如何设置功能相关性属性都是如此。

可定义若干属性关系，它们指示 `CountryID` 和 `Country`、`StateID` 和 `State`、`CityID` 和 `City` 以及 `CityID` 和 `City_Population` 之间可能存在的功能相关性。

您可以在 OLAP 中心中显式创建属性关系对象。可使用“属性关系”向导以显式方式创建属性关系。从“关系对象”视图打开“属性关系”向导，并指定所有对象定义特性。

定义属性关系对象的特定属性在下表中描述。

表 16. 属性关系属性

属性	描述
左边属性	关系中使用的左边属性。
右边属性	关系中使用的右边属性。
类型	属性关系描述的关系类型。类型用于确定属性担任的角色：描述性的和关联的

表 16. 属性关系属性 (续)

属性	描述
基数	连接中应当使用的基数: 1:1、1:多、多:1 和多:多
功能相关性	确定属性关系是否也是功能相关性: 是和否。 <b>注:</b> DB2 Cube Views 将不为属性关系创建功能相关性, 不管您是否对此属性指定“是”或“否”都是如此。

## 连接

连接是描述两个关联表之间的连接的元数据对象。连接引用属性, 而这些属性引用正在连接的表中的列。

最简单的连接形式引用两个属性: 一个映射至第一个表中的列, 另一个映射至第二个表中的列。您也可以指定运算符以指示如何比较列。

连接还可用于构造组合连接的模型, 其中第一个表中的两个或更多列连接至第二个表中相同数目的列。组合连接使用属性对将相应的列映射在一起。每对属性都具有一个运算符, 它指示如何比较这些列对。

连接还可以具有类型和基数。连接类型映射至关系连接类型。连接可用于在维中连接维表或在立方体模型中将立方体模型的维连接至其事实对象, 或者在事实对象中将多个事实表连接在一起。您可以使用 OLAP 中心中的“连接”向导来创建连接。

定义连接的特定属性在下表中描述。

表 17. 连接属性

属性	描述
(左边属性、右边属性和运算符) 列表	左边属性: 处于连接左边属性。右边属性: 处于连接右边属性。运算符: 连接中应该使用的运算符 =、<、>、<>、>= 和 <=。
类型	应当使用的连接类型: 内连接、完整外连接、左外连接和右外连接
基数	连接中应当使用的基数: 1:1、1:多、多:1 和多:多

## 立方体

立方体是有时可使用单个 SQL 语句传输的 OLAP 立方体的精确定义。从立方体模型中派生的立方体包含基于立方体模型中的元数据对象的元数据对象的子集。

立方体事实对象和立方体维列表是位于引用的立方体模型中的立方体事实和立方体维列表的子集。由于立方体维仅允许每个立方体维具有一个立方体层次结构, 因此立方体适用于不使用多个层次结构的工具 and 应用程序。

可在优化立方体模型以指定立方体模型中最活跃最重要的区域时使用立方体。可指定用来定义立方体中查询最频繁的特定区域的优化片。

您可以使用 OLAP 中心中的立方体向导来创建立方体。必须具有完整的立方体模型才能创建关联的立方体。立方体的属性将在下表中描述。

表 18. 立方体属性

属性	描述
立方体模型	从中派生立方体的立方体模型。
立方体事实对象	立方体中使用的立方体事实对象。立方体事实对象是从立方体模型中的事实对象派生的。
立方体维列表	立方体中使用的立方体维的已排序列表。每个立方体维都是从立方体模型中的相应维派生的。一个立方体层次结构与每个立方体维关联。
优化片列表	零个或多个优化片的集合。每个优化片包括： <ul style="list-style-type: none"> <li>• 类型可以是下寻、报表、MOLAP 抽取、混合抽取或深入钻取。</li> <li>• 优化级别的集合，对于立方体中的每个立方体维，集合中都存在一个优化级别。优化级别引用：               <ul style="list-style-type: none"> <li>– 立方体维及其相应立方体层次结构</li> <li>– 立方体层，任何或全部</li> </ul> </li> </ul>

## 立方体事实对象

立方体事实对象在特定事实对象的排序列表中具有量度的子集。立方体事实对象允许立方体灵活地划分立方体模型的事实对象的作用域。结构信息（如连接和属性）是从父事实对象中引用的。

在 OLAP 中心中，使用立方体向导在立方体模型的上下文中创建立方体。您不必显式定义立方体事实对象，因为 OLAP 中心知道立方体事实对象是从关联立方体模型中的事实对象派生的。选择要在立方体中使用立方体模型事实对象中的哪些量度。

定义立方体事实对象的特定属性将在下表中描述。

表 19. 立方体事实对象属性

属性	描述
事实对象	从中派生立方体事实对象的事实对象。
量度列表	立方体中使用的量度的已排序列表。所有量度必须是从中派生立方体事实对象的事实对象的一部分。

## 立方体维

立方体维用于划分维的作用域以在立方体中使用。立方体维引用从中派生出来的维以及给定立方体的相关立方体层次结构。

只能将一个立方体层次结构应用于立方体维。应用于立方体维的连接和属性是从维定义中引用的。

在 OLAP 中心中，使用立方体向导在立方体模型的上下文中创建立方体。选择要将哪些立方体模型维包括在立方体中。对于作为立方体维包括的每个维，您可以选择将哪些层包括在立方体层次结构中。

定义立方体维对象的特定属性在下表中描述。

表 20. 立方体维属性

属性	描述
维	从中派生立方体维的维。
立方体层次结构	应用于立方体维的立方体层次结构。

## 立方体层次结构

立方体层次结构是层次结构的子集，并在立方体中使用。立方体层次结构引用从中派生它的层次结构（父层次结构），并可以具有充当父层次结构中的父层子集的立方体层的集合。

一个立方体维只能有一个立方体层次结构。通常，立方体层次结构具有与从中派生的层次结构相同的层次结构类型和部署机制。如果层次结构是网络类型，则立方体层次结构是平衡的（如果不缺少成员）或未对齐的（如果缺少成员）。

在 OLAP 中心中，通过使用立方体向导在立方体模型的上下文中创建立方体。选择要将哪些立方体模型维包括在立方体中。对于作为立方体维包括的每个维，您可以选择将哪些立方体层包括在立方体层次结构中。

定义立方体层次结构的特定属性在下表中描述。

表 21. 立方体层次结构属性

属性	描述
层次结构	从中派生立方体层次结构的层次结构。
立方体层的已排序集合	充当包括在父层次结构中的层的子集的一个或多个立方体层的已排序集合。立方体层的顺序应与父层次结构中顺序的相同。

## 立方体层

立方体层是层的子集并在立方体中使用。立方体层引用从中派生它的层（父层）并继承对父层定义的层键属性和缺省属性。

立方体层可具有充当父层中的相应属性的子集的属性的集合。

立方体层的属性将在下表中描述。

表 22. 立方体层属性

属性	描述
属性集	充当父层的相关属性的子集的零个或多个相关属性的集合。



---

## 元数据对象规则

有三种类型的规则适用于元数据对象：基本规则、立方体模型完整性规则和优化规则。这些规则确保每个对象在立方体模型上下文内外均有效，并确保可以编写和优化有效的 SQL 查询。

### 基本规则

基本规则定义元数据对象在其使用上下文之外的有效性。每个元数据对象都具有自己的规则集。如果元数据对象遵循所有基本规则，则该对象有效。

### 完整性规则

完整性规则仅适用于立方体模型并扩展基本规则以确保立方体模型适当地链接至其它元数据对象，并且可以编写有效的 SQL 查询。

### 优化规则

优化规则进一步扩展基本规则和立方体模型完整性规则。这些规则确保可以成功优化为元数据对象创建的 SQL 查询。

## 立方体模型规则

### 基本规则:

- 维-连接对必须同时具有维和连接。
- 维至事实连接一侧的所有属性必须存在于维的属性列表中，连接另一侧的所有属性必须存在于事实对象的属性列表中。
- 立方体模型必须引用立方体模型的事实对象中的量度聚集所引用的所有显式维。如果立方体模型具有维，包含空的维列表的聚集必须与立方体模型中至少一个维匹配。确保同一量度的其它聚集中未引用此维。但是，如果立方体模型没有任何维，所有量度只能具有包含空的维列表的聚集。

### 完整性规则:

- 立方体模型必须引用一个事实对象。
- 立方体模型必须至少引用一个维。

### 优化规则:

- 用于连接事实对象和维的连接必须具有基数 1:1 或多:1，并且必须将事实表连接至维的主表。
- 立方体模型中的至少一个维必须至少有一个层次结构。

## 事实对象规则

### 基本规则:

- 事实对象必须至少引用一个量度。
- 事实对象引用的所有属性和量度必须能够在事实对象中连接。仅考虑事实对象的连接。
- 事实对象中的任意两个表之间恰好可定义一个连接。
- 事实对象中不允许存在连接循环。
- 事实对象引用的连接必须引用事实对象的属性。

## 维规则

### 基本规则:

- 维必须至少引用一个属性。
- 所有由维引用的属性都必须能够在维内被连接。仅考虑维的连接。
- 维内不允许存在连接循环。
- 在维内的任何两个表之间恰好可定义一个连接。
- 维所引用的连接必须引用维的属性。
- 维所引用的层必须引用维的属性。
- 维所引用的层次结构必须引用维的层。

### 优化规则:

- 维必须具有一个主表，连接使用 1:1 或多:1 基数连接至该表。

## 层规则

### 基本规则:

- 层必须至少引用一个层键属性。
- 层必须至少引用一个缺省属性。

## 层次结构规则

### 基本规则:

- 层次结构必须至少引用一个层。
- 递归部署必须恰好存在两个层。
- 标准部署可用于所有类型的层次结构，而递归部署只能用于未平衡层次结构。

## 量度规则

### 基本规则:

- 每个 SQL 表达式模板都可以具有零个或更多以下参数：属性、列和量度。
- 用作 SQL 表达式模板的参数的属性和量度不能形成相关性循环。
- 量度的 SQL 模板不能为空字符串。
- SQL 模板不能使用聚集函数。
- 如果至少引用一个量度并且只引用量度，则定义聚集是可选的。
- SQL 模板的数量必须与第一个聚集函数的参数数量相匹配（如果聚集存在）。
- 具有多个 SQL 模板的量度必须至少在聚集脚本中定义一个步骤。
- 如果量度引用定义多个 SQL 模板的第二个量度，则引用量度不能具有聚集脚本。
- 多参数聚集函数只能用于第一个聚集中。
- 如果量度定义了一个或多个聚集，则一个聚集必须指定空的维列表。
- 无论在聚集内还是跨聚集，量度只能引用每个维一次。
- 在 SQL 模板中，标记指示符必须从 1 开始编号，并且必须持续地连续编号。
- 在 SQL 表达式内，每个列、属性和量度必须至少被引用一次。

## 属性规则

基本规则:

- 每个 SQL 模板都可以具有零个或更多以下参数属性和列。
- 用作 SQL 表达式模板的参数的属性不能形成相关性循环。
- 属性的 SQL 模板不能为空字符串。
- SQL 模板不能具有聚集函数。
- 在 SQL 模板中, 标记指示符必须从 1 开始编号, 并且必须持续地连续编号。
- 在 SQL 表达式内, 每个列和属性必须至少被引用一次。

## 属性关系规则

基本规则:

- 属性关系必须引用两个属性。
- 属性关系不能被定义为具有多:多基数的功能相关性。

## 连接规则

基本规则:

- 连接必须至少引用一个三元组 (左边属性、右边属性和运算符)。
- 必须为每个连接三元组定义有效的运算。左边的属性和右边的属性的数据类型应互相兼容, 并与指定的运算兼容。
- 所有左边属性必须解析为单个表的一个或多个列。
- 所有右边属性必须解析为单个表的一个或多个列。

优化规则:

- 您必须为参与连接的列定义约束。如果在等式的两边使用相同的列集, 则必须定义与列集相匹配的主键。如果等式两边使用的是不同的列集, 则一边必须具有匹配的主键, 而另一边必须具有匹配的外键并引用主键。
- 连接基数必须是 1:1、多:1 或 1:多。在等式两边有相同列集的连接中, 基数必须为 1:1。对于所有其它连接而言, 基数在定义了主键的一边必须为 1, 在定义了外键的一边为 N。如果外键的一边也定义了主键, 则必须为那一边的基数使用 1。
- 连接中使用的所有属性必须解析为不可空 SQL 表达式。
- 连接类型必须为内连接。

## 立方体规则

基本规则:

- 立方体必须引用一个立方体模型。
- 立方体必须引用一个立方体事实对象。
- 立方体必须至少引用一个立方体维。
- 立方体事实对象必须从被引用的立方体模型中使用的事实派生。
- 所有立方体维必须从被引用的立方体模型中使用的维派生。
- 立方体必须具有零个或多个优化片。
- 对于立方体中的每个立方体维, 优化片必须具有一个优化层。

- 优化片必须至少具有一个或多个优化级别。
- 优化级别必须具有一个立方体维引用和一个层次结构引用。
- 优化级别必须具有 **All**、**Any** 或立方体层引用。
- 在优化级别对象中，被引用立方体维和立方体层次结构必须从立方体对象中的对象派生。该立方体层次结构必须属于该立方体维。
- 在优化级别对象中，如果该立方体层引用不是 **All** 或 **Any**，则该立方体层必须属于该立方体层次结构。
- 优化级别中的立方体维或立方体层次结构引用在优化片中不应重复。
- 如果存在类型为混合抽取的优化片对象，则另一个类型为 MOLAP 抽取的优化片一定不能存在，反之亦然。
- 每个立方体最多具有一个类型为 MOLAP 抽取的优化片。
- 每个立方体最多具有一个类型为混合抽取的优化片。
- 如果存在类型为混合抽取的优化片，则另一个类型为深入钻取的优化片可以存在。

### 立方体事实对象规则

基本规则:

- 立方体事实对象必须引用一个事实对象。
- 立方体事实对象必须至少引用一个量度。
- 所有由立方体事实引用的量度也必须在相应事实对象中被引用。

### 立方体维规则

基本规则:

- 立方体维必须引用一个维。
- 立方体维必须引用一个立方体层次结构。
- 引用的立方体层次结构必须从在引用的维中使用的层次结构中派生。

### 立方体层规则

基本规则:

- 立方体层必须引用一个层。
- 立方体层必须至少引用一个相关属性。
- 所有被引用属性在该层中必须是相关属性。

### 立方体层次结构规则

基本规则:

- 立方体层次结构必须引用一个层次结构。
- 立方体层次结构必须至少引用一个立方体层。
- 立方体层次结构引用的立方体层必须从相应层次结构中的层派生。
- 立方体层次结构中的立方体层的列示顺序必须与被引用层次结构中的相应层的顺序相同。立方体层顺序不适用于网络层次结构。

---

## 第 3 章 设计 DB2 Cube Views 元数据对象模型

本节描述下列主题:

### 启动 OLAP 中心

启动 OLAP 中心, 使您可以管理元数据对象。

### OLAP 中心和 API 兼容性

对于 OLAP 中心和元数据 API, DB2 Cube Views V8.2 具有有限的向下兼容性支持。

### DB2 Cube Views 的权限和特权

要使用 OLAP 中心执行任务, 必须与 DB2 数据库管理员 (DBA) 联系以得到有关获取必需的操作系统权限和特权的帮助。

### 创建 DB2 Cube Views 元数据对象

可使用 OLAP 中心创建 DB2 Cube Views 元数据对象。

### 从立方体模型中除去维

如果不再需要某个维, 可从立方体模型中除去该维。如果某个维已被另一立方体模型使用, 可除去该维而不删除该维。

### 从数据库中删除元数据对象

如果不再使用此数据库中的立方体模型中的某个元数据对象, 可删除它。

---

## 启动和刷新 OLAP 中心

启动 OLAP 中心, 使您可以管理元数据对象。

要启动 OLAP 中心:

1. 单击开始 -> 程序 -> IBM DB2 -> 商业智能工具 -> OLAP 中心。此时将打开数据库连接窗口。
2. 在数据库连接窗口中, 登录至要管理元数据的数据库。

当元数据对象在数据库中时, OLAP 中心会显示快照。虽然 DB2 Cube Views 会始终确保它管理的元数据对象的完整性, 但当另一个 OLAP 用户或 API 应用程序在数据库中创建元数据对象时, OLAP 中心窗口的内容不会自动更新。如果另一个用户或 API 应用程序更改了元数据对象, 则您可以选择视图 -> 刷新以查看数据库的新状态。同时, 在处理同一个元数据对象的多个用户可能会遇到错误, 原因是他们在数据库中看到的可能不是最新的数据。多个用户不应该同时处理同一组元数据对象。

---

## OLAP 中心和 API 版本兼容性

对于 OLAP 中心和元数据 API, DB2 Cube Views V8.2 具有有限的向下兼容性支持。

DB2 Cube Views API 支持下列各项:

- 版本 8.2 API 支持来自版本 8.1 客户机的描述请求。描述是在前发行版中受支持的唯一版本 8.1 操作。

DB2 Cube Views OLAP 中心支持下列各项:

- 版本 8.2 OLAP 中心不支持版本 8.1 API 或版本 8.1 元数据表。
- 版本 8.1 OLAP 中心不支持版本 8.2 API 或版本 8.2 元数据表。
- 版本 8.1 OLAP 中心不能连接至版本 8.2 DB2 数据库。
- OLAP 中心支持版本 8.1 XML 的导入。导入版本 8.1 XML 文件时，在导入元数据之前，OLAP 中心使用转换操作将 XML 迁移至版本 8.2。
- OLAP 中心可导出版本 8.2 和版本 8.1 XML。

DB2 Cube Views db2mdapiclient 支持下列各项:

- 版本 8.1 db2mdapiclient 支持版本 8.2 API (如果使用版本 8.2 XML 的话)。
- 版本 8.2 db2mdapiclient 支持版本 8.1 API (如果使用版本 8.1 XML 的话)。

## DB2 Cube Views 的权限和特权

要使用 OLAP 中心执行任务，必须与 DB2 数据库管理员 (DBA) 联系以得到有关获取必需的操作系统权限和特权的帮助。

除非另外指定，否则下表中对 SELECT、INSERT、UPDATE 和 DELETE 特权的所有引用适用于安装了 DB2 Cube Views 的数据库的 DB2INFO 模式中的表。

### Windows 权限和特权

对于 Windows 操作系统，OLAP 中心任务的权限适用于用来连接至数据库的用户标识。

表 23. Windows 常规权限和特权

任务	权限和特权
所有 OLAP 中心任务	<ul style="list-style-type: none"> <li>• 针对安装了 DB2 Cube Views 的数据库的 SYSADM 或 DBADM</li> <li>• 针对安装了 DB2 Cube Views 的数据库中的存储过程 DB2INFO.MD_MESSAGE 的 EXECUTE (仅适用于 DB2 UDB V8)</li> <li>• 针对目标数据库的 CONNECT</li> </ul>
创建元数据对象	SELECT 和 INSERT
修改元数据对象	SELECT、INSERT、UPDATE 和 DELETE
删除元数据对象	SELECT 和 DELETE
将元数据对象导出至 XML 文件	SELECT
将 XML 文件中的元数据对象导入至 DB2 UDB	SELECT、INSERT 和 UPDATE
运行优化顾问程序	针对系统表和基本表的 SELECT
运行由优化顾问程序生成的 DB2 命令	<ul style="list-style-type: none"> <li>• 针对模式 DB2INFO 的 CREATEIN 和 DROPIN</li> <li>• 针对基本表的 SELECT 和 ALTER (或 CONTROL)</li> </ul>

一些设置和安装任务可使用 OLAP 中心完成。

表 24. Windows 设置和安装权限和特权

任务	权限和特权
所有设置和安装任务	<ul style="list-style-type: none"> <li>• 针对安装了 DB2 Cube Views 的数据库的 SYSADM, 用于创建元数据对象的模式</li> <li>• 针对要安装 DB2 Cube Views 的数据库的 CONNECT、CREATETAB 和 IMPLICIT_SCHEMA</li> <li>• 针对安装了 DB2 Cube Views 的数据库中的模式 DB2INFO 的 CREATEIN 和 DROPIN</li> <li>• 针对安装了 DB2 Cube Views 的数据库的 DB2INFO 模式中的所有表的 SELECT、INSERT、UPDATE、DELETE 和 CONTROL</li> </ul>

### AIX 权限和特权

在 AIX 上, 有两种不同类型的用户标识, 各自有不同的权限集合。一种类型的用户标识应设置为用于连接至数据库和运行 DB2INFO.MD\_MESSAGE 存储过程。此类型的用户标识在下文被称为连接用户标识。

另一种类型的用户标识应设置为用于运行所有其它 OLAP 中心任务。此类型的用户标识在下文被称为执行用户标识。此用户标识是在 AIX 上设置的特殊用户标识, 用于运行受防护存储过程。受防护存储过程在此用户标识下运行, 该用户标识被设置为 sqllib/adm 中的 .fenced 文件的所有者。此用户标识控制可供该存储过程使用的系统资源。

表 25. AIX 的连接用户标识的常规权限和特权

任务	权限和特权
所有 OLAP 中心任务	<ul style="list-style-type: none"> <li>• 针对安装了 DB2 Cube Views 的数据库的 SYSADM 或 DBADM</li> <li>• 针对安装了 DB2 Cube Views 的数据库中的存储过程 DB2INFO.MD_MESSAGE 的 EXECUTE (仅适用于 DB2 UDB V8)</li> <li>• 针对目标数据库的 CONNECT</li> </ul>

表 26. AIX 的执行用户标识的权限和特权

任务	权限和特权
创建元数据对象	SELECT 和 INSERT
修改元数据对象	SELECT、INSERT、UPDATE 和 DELETE
删除元数据对象	SELECT 和 DELETE
将元数据对象导出至 XML 文件	SELECT
将 XML 文件中的元数据对象导入至 DB2 UDB	SELECT、INSERT、UPDATE 和 DELETE
运行优化顾问程序	针对系统表和基本表的 SELECT

表 26. AIX 的执行用户标识的权限和特权 (续)

任务	权限和特权
运行由优化顾问程序生成的 DB2 命令	<ul style="list-style-type: none"> <li>• 针对模式 DB2INFO 的 CREATEIN 和 DROPIN</li> <li>• 针对基本表的 SELECT 和 ALTER (或 CONTROL)</li> </ul>

一些设置和安装任务可使用 OLAP 中心执行。这些任务需要连接用户标识的其它权限。

表 27. AIX 的连接用户标识的设置和安装权限和特权

任务	权限和特权
所有设置和安装任务	<ul style="list-style-type: none"> <li>• 针对安装了 DB2 Cube Views 的数据库的 SYSADM, 用于创建元数据对象的模式</li> <li>• 针对要安装 DB2 Cube Views 的数据库的 CONNECT、CREATETAB 和 IMPLICIT_SCHEMA</li> <li>• 针对安装了 DB2 Cube Views 的数据库中的模式 DB2INFO 的 CREATEIN 和 DROPIN</li> <li>• 针对安装了 DB2 Cube Views 的数据库的 DB2INFO 模式中的所有表的 SELECT、INSERT、UPDATE、DELETE 和 CONTROL</li> </ul>

## 创建 DB2 Cube Views 元数据对象

可使用 OLAP 中心创建 DB2 Cube Views 元数据对象。

### 在 DB2 Cube Views 与 OLAP 工具之间交换元数据

使用 OLAP 中心以连接至 DB2 UDB 数据库并导入或导出元数据对象。

#### 连接至 DB2 数据库

可连接至 DB2 数据库或与 DB2 数据库断开连接。

在使用 OLAP 中心连接至 DB2 数据库之前，必须设置数据库以便与 DB2 Cube Views 元数据配合使用。如果尝试连接至的数据库未设置以便与 DB2 Cube Views 元数据配合使用，在尝试连接时将显示一条消息。如果用来连接的用户标识具有适当的权限和特别，OLAP 中心可执行设置任务。

**注意：** 如果已经连接至某个数据库然后又连接至另一个数据库，将与原始数据库断开连接。原始数据库中的所有元数据对象都不会再显示在 OLAP 中心对象树中。

要连接至 DB2 数据库：

1. 通过单击 **OLAP 中心 -> 连接** 打开数据库连接窗口，然后指定必需的信息。
  - 在 **数据库名称** 字段中，选择要连接至的数据库。
  - 在 **用户名** 字段中，输入用于指定的数据库的用户标识。
  - 在 **密码** 字段中，输入用于指定的用户标识的密码。



2. 单击**确定**。已连接至的 DB2 数据库中的元数据对象将显示在 OLAP 中心对象树中。

要与 DB2 数据库断开连接:

从 OLAP 中心主窗口中单击 **OLAP 中心 -> 断开连接**。将从 OLAP 中心对象树中除去所有元数据对象。

## 导入元数据

可将元数据对象导入到 DB2 UDB 中, 使您可以使用 DB2 Cube Views 管理元数据对象。

将元数据对象导入到 DB2 UDB 中之前, 确保具有 DB2 Cube Views XML 文件。可通过使用至供应商应用程序的元数据桥、从现有 DB2 UDB 数据库中导出元数据或使用 XML 编辑器来创建 DB2 Cube Views XML 文件。

可以导入版本 8.1 和版本 8.2 DB2 Cube Views XML 文件。“导入”向导创建对象时, 可将版本 8.1 元数据对象转换为版本 8.2 元数据对象。

要导入元数据:

1. 通过单击 **OLAP 中心 -> 导入** 打开“导入”向导。“导入”向导将打开。
2. 在“源”页上, 指定具有想要导入的元数据信息的 XML 文件。
3. 在“导入选项”页上, 查看要导入的元数据对象列表并指定导入选项。您可以查看该对象当前是否存在, 否则将在导入对象后创建。
4. 在“总结”页上, 查看您对元数据导入所做的决定。
5. 单击**完成**以导入元数据对象。您导入的对象将出现在 OLAP 中心对象树中。

## 导出元数据

可导出 DB2 Cube Views 元数据对象, 使您可以在业务合作伙伴应用程序中使用元数据对象。

在将元数据对象导出至 DB2 Cube Views XML 文件之前, 必须创建想要导出的元数据对象。

要导出元数据:

1. 通过单击 **OLAP 中心 -> 导出来** 打开“导出”窗口。
2. 选择要导出的元数据的版本。如果要使用连接至供应商工具的桥, 检查供应商工具支持哪个版本的元数据。
3. 选择**所有元数据对象**以导出数据库中的所有元数据对象或选择**选择的元数据对象**以指定要导出的特定元数据对象集合。可选择一个立方体模型, 或选择具有同一父立方体模型的一个或多个立方体。如果选择一个或多个立方体, 父立方体模型将自动导出。
4. 指定要导出至的 XML 文件的名称。如果该文件已经存在, 它将被覆盖。
5. 单击**确定**。将创建包含有关您指定的元数据对象的信息的 DB2 Cube Views XML 文件。可将 DB2 Cube Views XML 文件与业务合作伙伴应用程序配合使用。

## 使用快速启动向导创建立方体模型

可以根据关系模式一次性创建立方体模型和对应的事实对象、量度、维、属性和连接。

必须对数据库中的表定义引用完整性约束才能创建立方体模型及其对应的元数据。

快速启动向导将创建它可通过模式推断出的元数据对象。您可以指定事实表和量度列，向导将检测对应的维、连接和属性。完成快速启动向导后，您可根据需要添加、删除和修改元数据对象。

要创建立方体模型及其对应元数据：

1. 通过右键单击数据库或 OLAP 中心对象树中的**立方体模型**文件夹并单击**创建立方体模型 - 快速启动**来打开快速启动向导。快速启动向导打开。如果对象树未包含**立方体模型**文件夹，则通过单击视图 -> **显示 OLAP 对象**来确保您在 OLAP 对象视图中。

“简介”页描述快速启动向导的功能。

2. 在“事实表”页上，选择立方体模型的模式和事实表。选择的事实表直接映射至立方体模型中的事实对象。在指定事实表中用作外键的列将映射至立方体模型中的属性对象。向导不能检测隐式维（其列存在于事实表中的维）。可在完成快速启动向导之后将隐式维添加至立方体模型。
3. 在“量度”页上，从指定的事实表中选择要直接映射至量度的列。主键和外键不能映射至量度且未列示出来。对于基于数字列的量度，缺省聚集函数为 SUM，而对于基于字符列的量度，缺省聚集函数为 COUNT。您不能在此向导中创建计算的量度；可在完成快速启动向导后创建计算的量度。
4. 在“总结”页上，查看将创建的元数据对象。单击**完成**以创建立方体模型和对应的元数据对象。单击**完成**时，快速启动向导将在 DB2 目录中创建下列元数据对象：
  - 包含所有其它元数据对象的立方体模型。
  - 与您指定的事实表相对应的事实对象。
  - 与您指定的事实表列相对应的量度。
  - 与连接至事实表的每个维表相对应的维。支架表是连接至维表的表，且包括在相应的维对象中。
  - 与维和支架表中的每个列相对应并与事实表中的任何外键相对应的属性。
  - 充当事实维连接和维表中连接维表及任何对应支架表的连接的对象。

可将层次结构、计算的量度和立方体添加至立方体模型。

## 创建完整的立方体模型

要创建完整的立方体模型，可创建空立方体模型，然后添加事实对象和维，并对立方体模型中的每个维添加层次结构和层。立方体模型定义星型模式或雪花模式中的关系数据之间的关系，以便您可优化向关系数据发出的查询。

可创建空立方体模型，或者遵循下面的步骤来创建完成立方体模型所需的所有对象。在完成立方体模型后，可创建一些立方体对象，它们可能包含立方体模型对象的属性的全部或子集。

**DB2 Cube Views** 立方体模型是逻辑星型模式或雪花模式的表示法。立方体模型是中心事实对象周围的相关维对象的分组。每个维可以有多个层次结构，这增加了立方体模型的灵活性。有关如何连接由事实和维对象使用的表的结构信息存储在立方体模型中。立方体模型还存储了用于检索 OLAP 数据的充足信息。了解立方体模型并能够支持特定维的多个视图的其它报表和 OLAP 工具可受益于为某个维定义的多个层次结构。

要创建完整的立方体模型:

1. 创建立方体模型。
2. 创建事实对象。
3. 创建维。
4. 为维创建层次结构。
5. 为每个层次结构创建层。

## 创建立方体模型

创建立方体模型以对中心事实对象周围的相关维对象进行分组。

有关如何连接由事实和维对象使用的表的结构信息存储在立方体模型中。可使用立方体模型中的对象子集创建立方体以有选择向应用程序显示相关事实和维。

立方体模型向导指导您完成创建空立方体模型的步骤。通过使用向导创建立方体模型之后,使用相应连接对象添加事实对象和维。可以在不同立方体模型之间共享维。

要创建立方体模型:

1. 通过右键单击数据库或 OLAP 中心对象树中的**立方体模型**文件夹并单击**创建立方体模型**来打开立方体模型向导。如果对象树不包含**立方体模型**文件夹,则通过单击**图 -> 显示 OLAP 对象**来确保您在 OLAP 对象视图中。
2. 在**名称**页上,指定该立方体模型的名称、业务名、模式和注释。
3. 单击**完成**以创建该立方体模型。

## 创建事实对象

创建事实对象作为现有立方体模型的一部分。事实对象组合涉及给定应用程序的相关量度。事实对象在立方体模型中作为星型模式的中心。可以创建简单的和计算的量度。

创建事实对象之前,必须创建立方体模型。

要创建事实对象:

1. 通过右键单击立方体模型并单击**创建事实**打开事实向导。如果**创建事实**选项不可用,则表示已对选择的立方体模型定义了事实对象。
2. 在“名称”页上,指定事实对象名称、业务名和模式。还可以输入有关事实对象的注释。在事实向导中创建的所有计算和非计算的量度都将具有对该事实对象指定的同一模式名称。
3. 在“表”页上,从可用表列表中选择一个或多个事实表。
4. 在“连接”页上,指定选择的各个表之间的连接。可选择现有的连接,也可单击**创建连接**以打开可在其中创建新连接的向导。所有选择的表必须以没有循环的方式连接。可在每对表之间仅指定一个连接。如果在“表”页上仅选择了一个表,则不需要指定连接。
5. 在“量度”页上,创建简单量度和/或计算的量度。

### 简单量度

直接映射至列的量度。将列移动至选择的量度列表将创建直接映射至该列的简单量度。

## 计算的量度

引用表达式且通过一个或多个列、属性或其它量度创建的量度。要创建计算的量度，单击**创建计算的量度**。

6. 在“聚集”页上，对每个量度指定简单聚集函数。对于具有数字数据类型的量度，缺省聚集函数为 **SUM**。对于具有字符数据类型的量度，缺省聚集函数为 **COUNT**。要更改聚集，在表中单击聚集并选择另一个函数或不选择任何内容。
7. 单击**完成**以创建事实对象。

## 创建维

可创建维作为立方体模型的一部分，或者可以创建维以便稍后使用。

在为立方体模型创建维之前，必须创建事实对象。

维对象定义一组相关属性并连接一起描述量度的某个方面的属性。当需要连接来组合属性时（例如，使用雪花模式），必需的连接和在这些连接中使用的属性列示在维定义中。称为 **Region** 的维可能包含诸如 **State**、**Country**、**City** 和 **Population** 之类的属性。维引用可用于驱动该维的导航和计算的层次结构。维具有描述维的特性的类型。

要创建维：

1. 通过右键单击立方体模型并单击**创建维**来打开维向导。
2. 在“名称”页上，指定该维的名称、业务名、模式和注释。在维向导中创建的所有属性将具有您对该维指定的相同模式名。
3. 在“表”页上，选择从中创建该维的源表。必须至少选择一个表。
4. 在“连接”页上，指定选择的各个表之间的连接。可选择现有连接，或者可单击**创建连接**以打开一个向导，使您可以创建连接。所有选择的表必须以没有循环的方式连接。在每对表之间只能指定一个连接。如果在**表**页上只选择了一个表，则不必指定连接。
5. 在“属性”页上，指定要包括在该维中的属性。必须至少指定一个属性。可从列表中选择属性，或者单击**创建计算的属性**以打开一个窗口，使您可以创建表达式。该对象树显示源表及其可用列和属性。如果数据库中已经存在映射至表的其中一列的属性，将显示该属性。如果列未映射至属性，则显示该列本身。如果已选择某列并且该列已移至**选择的属性**列表，将创建映射至该列的属性。选择想要在维的层次结构中使用的所有属性。OLAP 中心会将指定连接使用的所有标识属性自动添加至该维。
6. 在“类型”页上，指定想要创建的维的类型。选择 **Time** 以指定该维为 **Time** 维，或者选择 **Regular** 以指定该维并非 **Time** 维。
7. 在“事实-到-维”页上，指定连接对象以将该维连接至事实对象。还可使用“连接”向导创建连接。仅当要创建作为立方体模型的一部分的维时，才显示此页。
8. 单击**完成**以创建该维。

## 为维创建层次结构

层次结构定义立方体模型的给定维内的两个或多个层之间的关系。您还可定义仅使用一个层的层次结构。定义这些关系提供了遍历指定维的导航和计算方法。例如，**CalendarYear** 层次结构可能包含诸如 **Year**、**Quarter** 和 **Month** 之类的层。

在为维创建层次结构之前，必须创建维。

可对立方体模型中的维定义多个层次结构。各层之间的关系由层次结构类型确定。

要对维创建层次结构:

1. 通过展开维文件夹来打开“层次结构”向导以查看现有维。右键单击维并单击**创建层次结构**。
2. 在“名称”页上, 指定层次结构的名称、业务名、模式和注释。
3. 在“层”页上, 指定层次结构的层和类型:
  - a. 如果**可用层**列表中不存在任何层, 则单击**创建层**并使用“创建层”向导来创建层。
  - b. 选择想要的层。必须至少选择一个层。
  - c. 指定层次结构的类型和部署。如果指定**递归**层次结构类型, 则只能选择两个层。选择至少一个层后, 可单击**显示样本**以查看层次结构中的样本数据。
4. 单击**完成**以将新的层次结构添加至立方体模型。

### 为每个层次结构创建层

层定义一组相关属性之间的关系。DB2 Cube Views 会尽可能创建定义层属性之间的关系的功能相关性。

1. 通过展开维文件夹下的维来对 OLAP 中心对象树中的立方体模型打开“层”向导。右键单击层文件夹并单击**创建层**。
2. 在“名称”页上, 指定立方体模型的名称、业务名、模式和注释。
3. 在“层键属性”页, 选择一起唯一地定义该层的一个或多个属性。
4. 在“缺省属性”页上, 选择描述该层的数据的一个属性。
5. 在“相关属性”页上, 选择零个或多个属性以提供有关该层的更多信息。
6. 在“使用功能相关性进行优化”页上, 指定是否要在层键属性与缺省和相关属性之间创建适用的功能相关性。仅当您确定层键属性在功能上确定缺省属性和任何相关性时, 才应选择此复选框。
7. 单击**完成**以创建层。

## 将现有维添加至立方体模型

可将现有维添加至立方体模型。如果在另一个立方体模型中使用了某个维, 或者如果对另一个立方体模型使用了该维但稍后除去了它, 该维可能已经存在。

在添加维之前, 必须创建立方体模型和事实对象。还必须具有不属于立方体模型的现有维。

要添加现有维:

1. 通过右键单击立方体模型并单击**添加维**来打开“添加维”向导。
2. 在“维”页上, 选择要添加至立方体模型的一个或多个现有维。必须至少选择一个维。
3. 在“事实-到-维连接”页上, 指定连接对象以将被添加维添加至立方体模型的事实对象。

如果该维有一个相应的现有连接, 将对相应的维显示该连接。确保缺省连接有意义, 以便连接某一侧的属性引用相应维表中的列, 而连接另一侧的属性引用事实表中的列。

如果没有相应的现有连接或者有多个相应的现有连接，必须指定连接。要指定另一连接或为特定维创建连接，选择该维并单击**指定连接**。

4. 单击**完成**以将该维添加至立方体模型。

## 为现有维指定事实-到-维连接

可对现有维指定至立方体模型的事实-到-维连接。

- 要指定现有连接：
  1. 从候选连接列表中选择连接。
  2. 单击**确定**。
- 要创建连接：
  1. 单击**创建连接**。“连接”向导打开。
  2. 创建想要的连接。新连接将出现在现有候选连接列表中。
  3. 从候选连接列表中选择您创建的连接。
  4. 单击**确定**。

## 创建连接

可以创建连接以将维连接至事实对象以作为事实对象或维的一部分。

连接对象将两个关系表连接在一起。连接引用属性，而这些属性又引用要连接的表中的列。连接还具有类型和基数。

最简单的连接形式引用两个属性；一个映射至第一个表中的列，另一个映射至第二个表中的列，同时具有一个运算符以指示应如何比较列。

连接对象还可以用于对组合连接建模，其中第一个表中的两个或多个列连接至第二个表中相同数目的列。组合连接使用属性对将相应的列映射在一起。每对属性都具有一个运算符，它指示应如何比较这些列对。

连接对象主要用来在立方体模型中将立方体模型的维连接至其事实对象。连接还可以用来以雪花模式将维表连接在一起，或有时在事实对象中用来将多个事实表连接在一起。

要创建连接：

1. 打开“连接”向导。
2. 在“名称”页上，指定连接的名称、业务名、模式和注释。

**提示：**用一个名称来表示同时包括要连接的两个表的名称的连接。例如，如果您要连接 `SalesFact` 表和 `Product` 表，则将连接命名为：`SalesFact-Product`。

3. 在“连接”页上，添加一个或多个属性对并为该连接选择连接类型和基数。通过选择左边的属性和右边的属性然后单击**添加**来创建属性对。属性对出现在属性对表中。缺省连接运算符是 `=`。可通过在表中单击当前运算符并选择新运算符来更改运算符。缺省连接类型为 `Inner`，缺省基数为 `1:1`。
4. 单击**完成**以创建连接。

## 创建立方体

可创建立方体来指定重要的立方体模型区域。还可以使用立方体来对业务合作伙伴应用程序定义数据子集。

在创建立方体之前，必须创建或导入立方体模型。立方体模型必须具有事实对象和至少一个维，并且每个维必须具有层次结构。

立方体是可使用单个 SQL 语句查询的 OLAP 立方体的精确定义。立方体是从现有立方体模型派生的。立方体事实和立方体维列表是被引用立方体模型中的事实和维的子集。由于立方体维仅允许每个立方体维具有一个立方体层次结构，因此立方体适用于不使用多个层次结构的工具和应用程序。

要创建立方体：

1. 通过右键单击立方体模型并单击**创建立方体**来打开立方体向导。
2. 在“名称”页上，指定该立方体的名称、模式、业务名和注释。
3. 在“量度”页上，选择要包括在立方体中的量度。必须至少选择一个量度。
4. 在“维”页上，选择要包括在立方体中的立方体维。必须至少选择一个立方体维。选择立方体维，并单击选择的立方体维旁边的按钮以打开一个窗口，以便能够对立方体维指定层次结构信息。
5. 在“指定查询类型”页上，指定如何期望使用此立方体。如果选择**高级设置**，单击**指定**以对此立方体指定期望经常查询的特定优化片。
6. 单击**完成**以创建该立方体。

### 为立方体维指定立方体层次结构和立方体层

可通过对每个立方体维指定要包括在立方体层次结构中的立方体层和相关属性来为不同应用程序创建定制立方体。可指定要在立方体层次结构中引用的立方体层和相关属性的子集。

要在从立方体向导打开的“选择立方体层次结构的属性”窗口中指定立方体维的立方体层次结构：

1. 从**可能的层次结构**列表中选择立方体层次结构。
2. 在**层和属性**列表中，确保想要的立方体层或相关属性旁边的复选框已被选中。层键属性和缺省属性是在您选择层时包括的。
3. 单击**确定**以将立方体层和相关属性添加至立方体层次结构并返回至立方体向导。

### 指定立方体优化片

可选择对立方体指定优化片，不过它的功能非常强大，可以指导优化顾问程序提供重点放在立方体模型的最重要区域的摘要表。

优化片指示期望经常查询的立方体区域。优化片的类型指示期望如何查询该区域。例如，如果 50% 的查询包括 Month，可在 Any-Month-Any 片上定义优化片。优化顾问程序可使用提供的优化片来建议最贴近您的需要的摘要表。

要对立方体指定优化片：

1. 从立方体向导或“立方体属性”窗口中，打开“查询类型”页并选择**高级设置**然后单击**指定**。
2. 对立方体定义一个或多个优化片。

## 添加片

单击**新建**来添加片。新片出现在交互式图形并作为一行出现在**优化片的列表表**。在添加片之后，修改该片指定查询类型和每个立方体维中期望经常查询的立方体层。

## 修改片

可在交互式图形或该图形下面的**优化片的列表表**中修改某个片。

要在表中修改某个片，在该表中选择表示该片的行。单击想要修改的类型或立方体维并选择选项。

要在交互式图形中修改某个片，通过将立方体维中的节点拖至某层来更改层并通过右键单击该片并选择查询类型来更改类型。

可对每个立方体维选择下列其中一个选项：

- 如果该片并没有首选立方体层，选择 **Any**。优化顾问程序将确定要对其进行优化的立方体层。
- 如果经常在最高聚集级别查询数据（例如对所有产品或所有区域），则选择 **All**。
- 如果知道许多查询使用的立方体层，则选择对该立方体维定义的特定立方体层。例如，可在 **Any-Month-Any** 立方体层（在 **Market** 立方体维中选择 **Any**，在 **Time** 立方体维中选择 **Month** 并在 **Product** 立方体维中选择 **Any**）中定义优化片。

**建议：**要获取最佳效果，对每个立方体只定义几个（不超过三个）重点的片。

**除去片** 可在表中或在交互式图形中选择某个片并单击**除去**以删除该片。

---

## 从立方体模型中除去维

如果不再需要某个维，可从立方体模型中除去该维。如果某个维已被另一立方体模型使用，可除去该维而不删除该维。

从立方体模型中除去维时，下列操作适用：

- 仅从选择的立方体模型中除去该维，但它仍保留在正在引用它的其它任何立方体模型中。
- 从对应的立方体中除去该立方体模型中基于此维的所有立方体维。
- 不从数据库删除该维。
- 在**所有维**文件夹中提供此维。

要从立方体模型中除去维：

1. 在 **OLAP** 中心对象树中，展开**立方体模型**文件夹以查看现有立方体模型。如果对象树不包含**立方体模型**文件夹，通过单击视图 -> **显示 OLAP 对象**确保您在 **OLAP** 对象视图中。
2. 展开立方体模型节点以查看立方体模型中包含的对象类别。
3. 展开**维**文件夹以查看现有维。
4. 右键单击维，然后单击**除去**。



---

## 从数据库中删除元数据对象

如果不再使用此数据库中的立方体模型中的某个元数据对象，可删除它。

可从 OLAP 对象视图或关系对象视图中删除大多数对象。仅当对象未被其它对象引用时，才能删除它们。

要从数据库中删除元数据对象：

在 OLAP 中心对象树中选择一个或多个对象并右键单击选择的对象，然后单击删除。如果该菜单未包含删除菜单项，则不能删除选择的对象。在删除父对象时，将删除所有子对象。例如，删除维还会删除所有相应层次结构。



---

## 第 4 章 DB2 Cube Views 企业建模方案

本节描述下列主题:

### 计算随时间推移仓库中的存货的流动和价值

在将存货发送至特定商店进行销售之前，零售公司 XYZ Retail 的存货是放在仓库中的。XYZ Retail 一直维护有关仓库存货状态的数据，并想要分析此数据。

### 使广告成本与销售相关

一家汽车经销商正在考虑增加广告开销。为作出明智的决定，该经销商首先要分析广告开销与销售之间的历史关系。经销商想要确定广告等级的变化是否会影响销售，特别是广告的增加是否与销售量的增加是否有密切关联。

### 计算商店的利润和边际利润

玩具商店的总理想要能够分析各种因素（例如时间和产品类型）如何影响利润和边际利润。

### 对互联网订单计数

零售公司几年前已经通过增加互联网销售来扩展业务。现在该公司想要分析互联网销售的成果。该公司首先要计算的其中一项就是通过互联网完成的订单数量。

### 将销售数字分等级

办公用品连锁商店在过去的几年里得到飞速发展。业务经理在考虑关闭某些业绩最差的商店以削减成本和增加利润。

### 使用存储在事实表中的时间数据来创建 Time 维

零售公司 XYZ Retail 要在 DB2 Cube Views 中对其销售交易数据建模以便能够更有效地分析这些数据。但是，因为数据的交易特性，可用的唯一时间信息就是每笔交易的相应日期。

---

## 随着时间的推移计算仓库存货的流和值

在将存货发送至特定商店进行销售之前，零售公司 XYZ Retail 的存货是放在仓库中的。随着时间的推移，XYZ Retail 维护有关仓库中存货的状态的数据并想要分析此数据。

该公司特别想检查它的仓库的两个方面:

- 商品放入和取出仓库的流
- 给定时间仓库中商品的值

第一个方面，商品的流涉及查看随着时间的推移的数据。第二个方面，商品的值建立特定时刻仓库的快照。

### 方案详细信息

XYZ Retail 具有一个包含以下与仓库有关的列的事实表：QUANTITY\_IN、QUANTITY\_OUT、CURRENT\_QUANTITY、PRODUCT\_VALUE、PRODUCT\_ID 和 TIME\_ID。此数据是每周输入到表中的。该数据库还具有 Product 表和 Time 表。例如，在第 52 页的表 28 中显示了一组样本事实表数据。

表 28. 样本事实表数据

<b>PRODUCT_ID</b>	<b>TIME_ID</b>	<b>QUANTITY_IN</b>	<b>QUANTITY_OUT</b>	<b>CURRENT_QUANTITY</b>	<b>PRODUCT_VALUE</b>
1234	1	5	0	5	5
1234	2	20	10	15	5
1234	3	10	20	5	5

这三个样本数据条目的每一个的 **PRODUCT\_ID** 值相同，这是因为一种产品类型可多次移入或移出仓库。

XYZ Retail 的 DBA 必须创建三个不同的量度:

#### **Flow In**

对商品流入仓库的流建模。

#### **Flow Out**

对商品流出仓库的流建模。

#### **Current Value**

对给定时间商品的值建模。

为了创建前两种量度 (Flow In 和 Flow Out)，DBA 创建分别映射至 **QUANTITY\_IN** 和 **QUANTITY\_OUT** 列的量度并合计所有维中的数据。这也称为全加法量度，因为仅使用 **SUM** 函数来聚集所有维中的数据。例如，表 29 显示了 **PRODUCT\_ID** 为 1234 的商品的 **QUANTITY\_IN** 和 **QUANTITY\_OUT** 列三个月的样本数据。Flow In 和 Flow Out 量度合计每个月的价值以计算整个季度进入和取出仓库的总数量。

表 29. **PRODUCT\_ID** 1234 的 Flow In 和 Flow Out 全加法量度的样本数据的计算

	<b>January</b>	<b>February</b>	<b>March</b>	<b>Quarter 1</b>
<b>QUANTITY_IN</b>	5	20	10	35
<b>QUANTITY_OUT</b>	0	10	20	30

全加法量度是所创建的最简单最常用的量度，通常用作较复杂量度的构件。对于基于数字源数据的量度，缺省情况下，OLAP 中心会创建全加法量度。

为了创建第三个量度 (Current Value)，DBA 创建通过将 **PRODUCT\_VALUE** 乘以 **CURRENT\_QUANTITY** 计算值的计算的量度。例如，如果 **PRODUCT\_ID**=1234 的产品的值是 5，则该样本数据的 Current Value 量度显示在表 30 中。

表 30. **PRODUCT\_ID** 1234 的 Current Value 量度的样本数据的计算

	<b>January</b>	<b>February</b>	<b>March</b>
<b>CURRENT_QUANTITY</b>	5	10	20
<b>Current Value</b>	25	50	100

然后必须在各维中聚集此数据。但是，由于此量度要计算特定时间点的值，所以合计 Time 维中的值没有意义。取而代之，聚集将合计 Product 维中的数据并找出随着时间的推移数据的平均值。这也称为半加法量度，这是因为 **SUM** 函数只涉及部分聚集。

计算快照数据（表示特定时刻的数据，如月库存数据）的量度通常是半加法量度，这是因为将月份加到季度中没有任何意义。如果某产品整个季度都搁置在仓库中，则该产品会包括在该季度那三个月中每一个月的仓库库存 CURRENT\_QUANTITY 快照数据中。如果随着时间的推移合计 CURRENT\_QUANTITY 数据，则在仓库中搁置了三个月的产品会被计三次。如在表 31 中所示，Quarter 1 的值 25 没有对于仓库的活动没有任何意义。该表显示仓库中从未有过 25 个商品，因此计算得到 25 个产品的值没有任何意义。

表 31. 使用 SUM 函数对 PRODUCT\_ID 1234 Time 维 CURRENT\_QUANTITY 列的样本数据的计算

	January	February	March	Quarter 1
SUM(CURRENT_QUANTITY)	5	15	5	25

不是在所有维中使用 SUM 函数，而是可以对 time 维执行其它聚集函数（例如，AVG、MIN 和 MAX）。例如，对于 January、February 和 March 的同一组样本数据，可以对 time 维使用另一聚集函数来为 Quarter 创建有意义的值，如在表 32 中所示。Current Value 量度可表示整个季度存储在仓库中的商品的平均总值，或者在该季度中任意时间点的最大值或最小值。

表 32. 使用 AVG、MAX 和 MIN 函数对 PRODUCT\_ID 1234 time 维 CURRENT\_QUANTITY 列的样本数据的计算

	January	February	March	Quarter 1
AVG(CURRENT_QUANTITY)	5	15	5	8.3
MAX(CURRENT_QUANTITY)	5	15	5	15
MIN(CURRENT_QUANTITY)	5	15	5	5

### 创建量度的步骤

下列步骤说明可如何使用“OLAP 中心事实属性”窗口来在现有事实对象中创建 Flow In、Flow Out 和 Current Value 量度。

1. 要打开“事实属性”窗口，右键单击 OLAP 中心对象树中的事实对象并单击**编辑量度**。“事实属性”窗口打开。
2. 创建 Flow In 量度:
  - a. 在“量度”页上，单击**创建计算的量度**以创建 Flow In 量度。“SQL 表达式构建器”窗口打开。
  - b. 在“SQL 表达式构建器”窗口中，在**名称**字段中输入 FLOW IN。
  - c. 要创建 flow in 表达式，完成下列步骤:
    - 展开“数据”列表中的**列**文件夹和事实表。
    - 双击 **QUANTITY\_IN** 列以将它添加至表达式。
    - 单击**确定**以关闭“SQL 表达式构建器”窗口。不需要在“聚集”页上更改缺省聚集函数 SUM。SUM 函数是 Flow In 量度的缺省函数，这是因为数据源是数字的并且该量度引用列而不仅仅是现有量度。

3. 创建 Flow Out 量度:
  - a. 在“量度”页上，单击**创建计算的量度**以创建 Flow Out 量度。“SQL 表达式构建器”窗口打开。
  - b. 在“SQL 表达式构建器”窗口中，在**名称**字段中输入 FLOW OUT。
  - c. 要创建 flow out 表达式，完成下列步骤:
    - 展开**数据**列表中的**列**文件夹和事实表。
    - 双击 **QUANTITY\_OUT** 列。
  - d. 单击**确定**以关闭“SQL 表达式构建器”窗口。不需要在“聚集”页上更改缺省聚集函数 SUM。SUM 函数是 Flow Out 量度的缺省函数，这是因为数据源是数字的并且该量度引用列而不仅仅是现有量度。
4. 创建 Current Value 量度:
  - a. 在“量度”页上，单击**创建计算的量度**以创建 Current Value 量度。“SQL 表达式构建器”窗口打开。
  - b. 在“SQL 表达式构建器”窗口中，在**名称**字段中输入 CURRENT VALUE。
  - c. 要创建 Current Value 表达式，完成下列步骤:
    - 展开**数据**列表中的**列**文件夹和事实表。
    - 双击**数据**列表中的 **PRODUCT\_VALUE** 列。
    - 双击**运算符**列表中的 \* 运算符。
    - 双击**数据**列表中的 **CURRENT\_QUANTITY** 列。

第 55 页的图 12 显示了可以在“SQL 表达式构建器”窗口中创建的 Current Value 表达式。



图 12. 在 SQL 表达式构建器窗口中完成 *Current Value* 表达式


- d. 单击**确定**以关闭“SQL 表达式构建器”窗口。
- e. 在“聚集”页上，单击 *Current Value* 量度的聚集，然后从列表中单击**聚集脚本**。“聚集脚本构建器”窗口打开。缺省聚集脚本将 **SUM** 函数用于所有维。
- f. 必要时，通过选择 **Time** 并单击  下移按钮来下移 **Time** 维，以使它成为列示在脚本中的最后一个维。然后在选择了 **Time** 维的情况下，双击**列函数**列表中的 **AVG** 函数。聚集脚本（如在第 56 页的图 13 中所示）对 **Time** 维之外所有维的数据求和（数据是根据 **Time** 维求平均值的）。



图 13. Current Value 量度的聚集脚本

- g. 在“聚集脚本构建器”窗口中，单击**验证**以验证聚集脚本是否有效。单击**确定**以保存聚集脚本并关闭窗口。
- 5. 单击**确定**以保存对事实对象的更改并关闭“Facts 属性”窗口。

现在就有了三个针对仓库中存货的计算的量度。可以使用这些量度来分析产品流入和流出仓库的模式。

## 使广告成本与销售相关

一家汽车经销商正在考虑增加广告开销。为作出明智的决定，该经销商首先想要分析广告开销与销售之间的历史关系。经销商想要确定广告等级的变化是否会影响销售，特别是广告的增加是否与销售量的增加是否有密切关联。

### 方案详细信息

经销商的数据库具有一个包含 Sales 列和 Ad Costs 列的事实表。该数据库还具有若干个其它维表。DBA 可以创建使用 DB2 CORRELATION 函数来执行成本与销售之间的



关联计算的量度。CORRELATION 函数是需要两个输入参数的多参数函数。在此示例中，DBA 将使用 Sales 列和 Ad Costs 列作为两个输入参数。

DBA 必须首先应用聚集脚本中的多参数聚集函数。多参数函数可以应用于所有维，也可以先将它应用于 Time 维之外的所有维，然后再将另一个函数（例如，MAX 函数）应用于 Time 函数。DBA 为量度定义 SQL 表达式以便量度直接映射至 Ad Costs 列。SQL 表达式是在多参数函数中使用的两个参数中的第一个。DBA 将第二个参数定义为直接映射至 Sales 列的 SQL 表达式。CORRELATION 函数定义为唯一的聚集函数，以便量度能够计算所有维中广告成本与销售结果的统计关联。

### 创建量度的步骤

下列步骤说明可如何使用“OLAP 中心事实属性”窗口来在现有事实对象中创建 Advertising-Sales Correlation 量度。

1. 通过右键单击 OLAP 中心对象树中的事实对象并单击**编辑量度**来打开“事实属性”窗口。
2. 单击**创建计算的量度**按钮。“SQL 表达式构建器”窗口打开。
3. 在“SQL 表达式构建器”窗口中，在**名称**字段中输入 ADVERTISING-SALES CORRELATION。
4. 定义量度的表达式，该表达式还将用作聚集脚本中多参数 CORRELATION 函数的第一个参数。要定义该表达式，展开**数据**列表中的**量度**文件夹并单击 **AD COSTS** 量度以将它添加至**表达式**列表。第 58 页的图 14 显示了在“SQL 表达式构建器”窗口中创建的表达式。

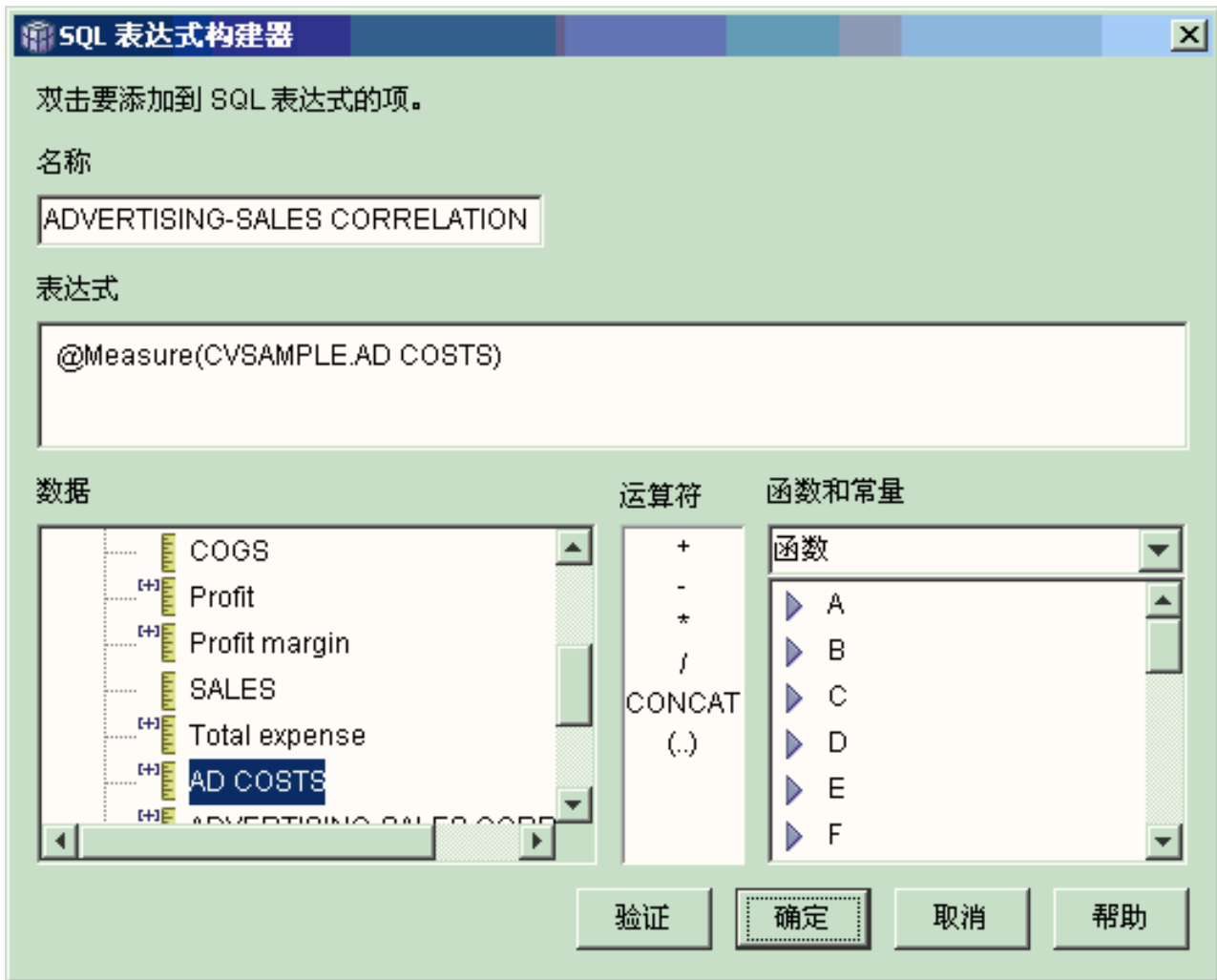


图 14. 在 SQL 表达式构建器窗口中完成 *advertising-sales correlation* 表达式

5. 在“聚集”页上，单击 **ADVERTISING-SALES CORRELATION** 量度的聚集函数，然后选择聚集脚本，如在第 59 页的图 15 中所示。“聚集脚本构建器”窗口打开。

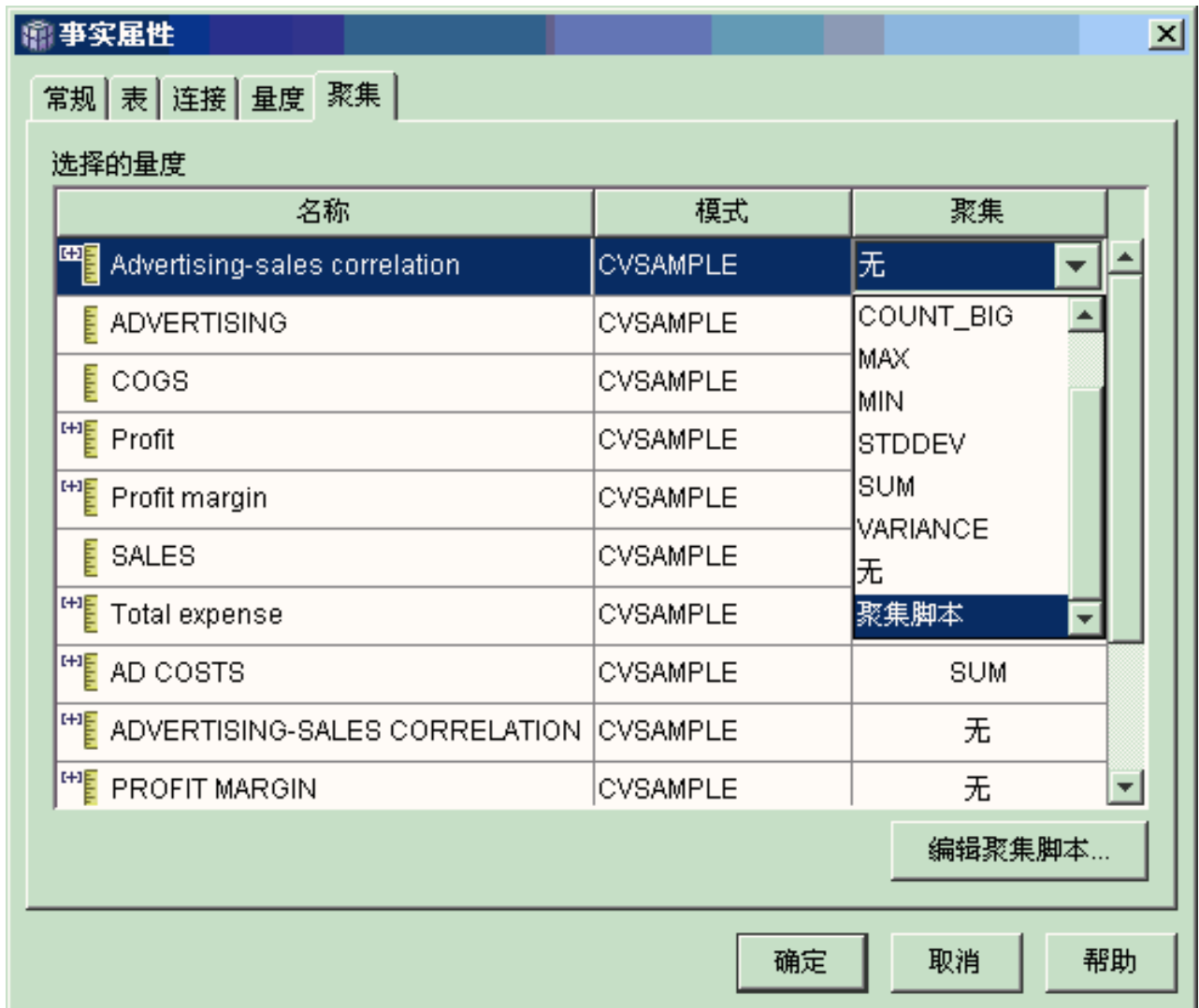


图 15. 事实属性窗口的聚集页

- 在列函数字段中，选择多参数函数。在多参数函数列表中，选择 **CORRELATION** 函数然后单击**添加至脚本**。“函数参数”窗口打开。
- 选择使用**现有量度**，然后从列表中选择 **SALES**，如在第 60 页的图 16 中所示。

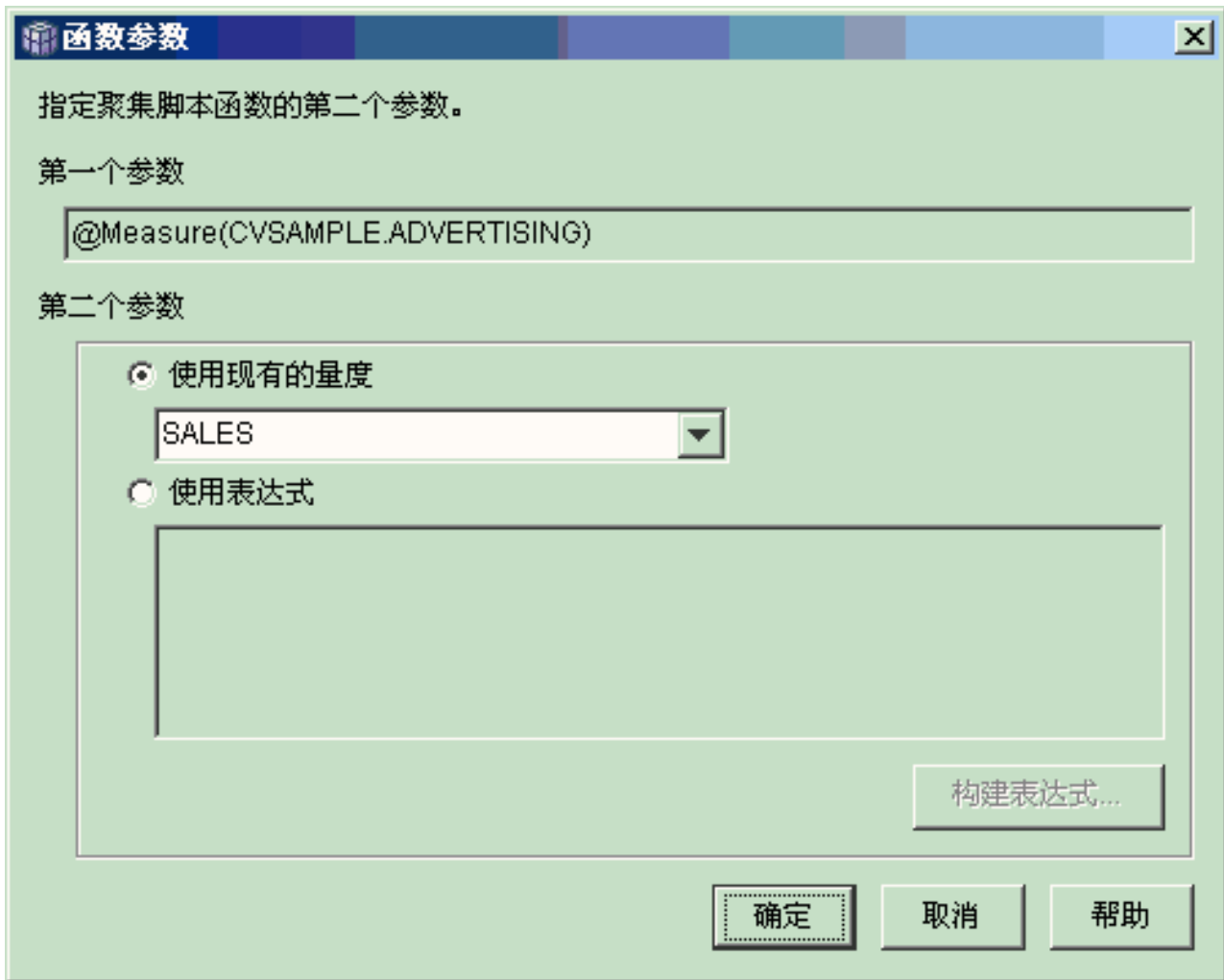


图 16. 在函数参数窗口中指定为第二个参数的 Sales 量度

8. 单击**确定**以保存选择并关闭“函数参数”窗口。
9. 第 61 页的图 17 显示: CORRELATION 函数位于脚本中维列表的顶部。

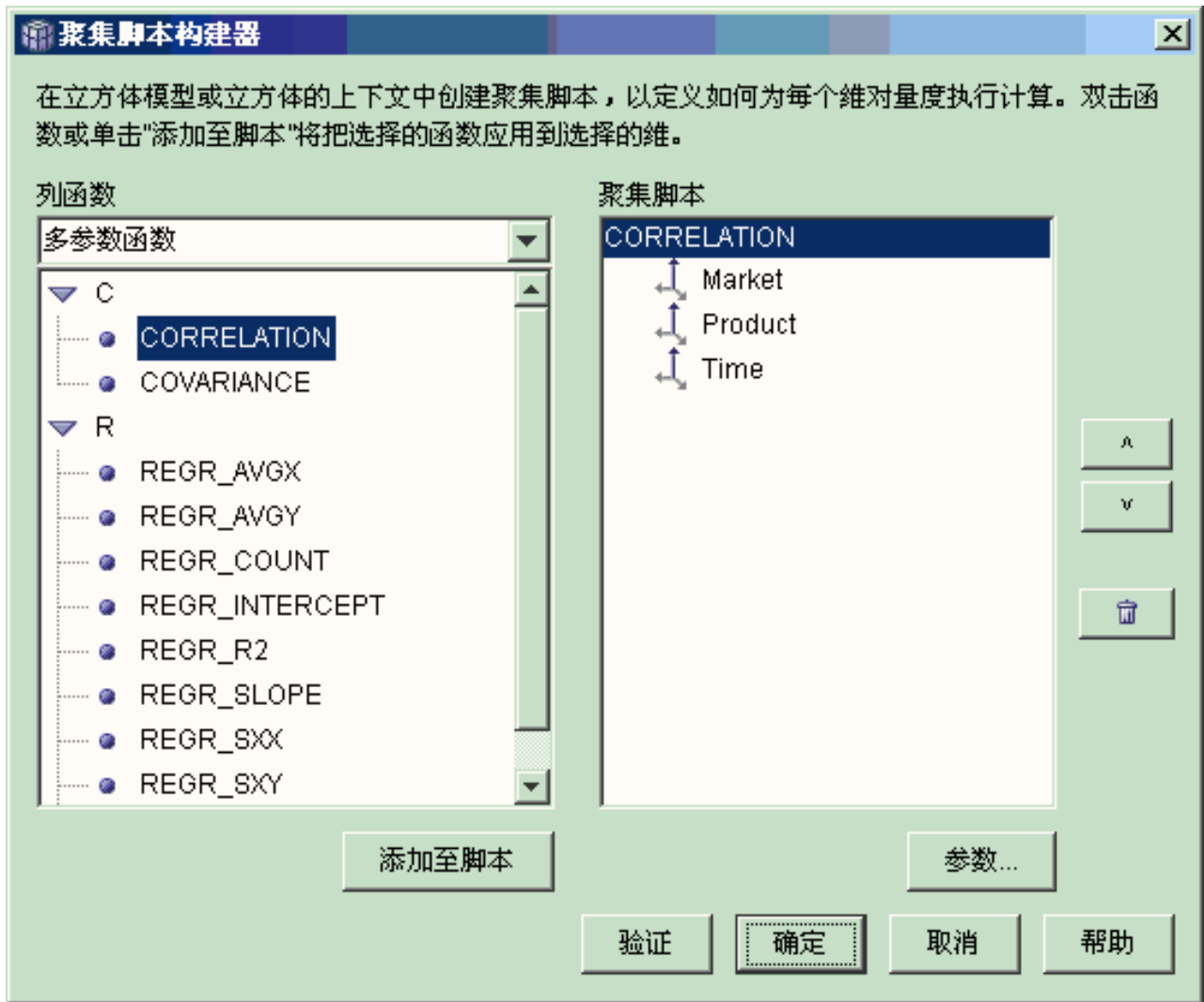


图 17. Advertising-Sales Correlation 量度的聚集脚本

10. 在“聚集脚本构建器”窗口中，单击**验证**按钮以验证聚集脚本是否有效。
11. 单击**确定**以保存聚集脚本并关闭窗口。
12. 单击**确定**以保存对事实对象的更改并关闭“Facts 属性”窗口。

现在，就具有了使数据库中的两种类型的数据相关联的量度。可以使用此量度来根据结果中的历史趋势来决定将来的广告开销。

## 计算商店的利润和边际利润

玩具商店的总理想要能够分析各种因素（例如时间和产品类型）如何影响利率和边际利润。

在可以完成更高级的分析之前，玩具商店的 DBA 必须首先创建利润和边际利润量度。接着，DBA 可以创建其它量度，这些量度使利润量度和边际利润量度的各种因素相关联以及对这些因素进行比较。

## 方案详细信息

除了具有几个维表中的每一个的相应外键列之外，玩具商店的数据库还具有一个事实表，这个表包括“售出商品成本”（COGS）列和 Expense 列。DBA 已经创建了分别映射至 Sales、COGS 和 Expense 列的 Sales、COGS 和 Expense 量度。完全可从这些现有量度创建 Profit 和 Profit Margin 量度。

为了创建 Profit 量度，DBA 创建一个用 SQL 表达式计算 SALES-(COGS+EXPENSE) 并合计所有维中计算出来的数据的量度。可通过引用现有量度、列或两者的组合创建 Profit 量度。

在创建了 Profit 量度之后，DBA 可以创建 Profit Margin 量度。Profit Margin 量度是两个现有量度的比率，用百分比表示，即， $(\text{Profit} / \text{Sales}) * 100$ ，此量度不需要它自己的聚集函数。不需要聚集函数的原因是，此量度只引用已聚集数据的其它量度。如果 DBA 使用组合量度（只引用其它量度的量度）来计算比率，则 DBA 不需要定义其它聚集。大部分聚集函数（如 SUM）对于比率都没有意义。例如，如果玩具商店四个连续季度的边际利润分别 40%、32%、28% 和 37%，随着时间的推移合计比率会得出该年的边际利润是 137%，此值没有任何意义。

## 创建量度的步骤

下列步骤说明可如何使用“OLAP 中心事实属性”窗口来在现有事实对象中创建 Profit 量度和 Profit Margin 量度。

1. 要打开“事实属性”窗口，右键单击 OLAP 中心对象树中的事实对象并单击**编辑量度**。“事实属性”窗口打开。
2. 创建 Profit 量度：
  - a. 在“量度”页上，单击**创建计算的量度**按钮。“SQL 表达式构建器”窗口打开。
  - b. 在“SQL 表达式构建器”窗口中，在**名称**字段中输入 PROFIT。
  - c. 要创建 Profit 表达式，展开**数据**列表中的**量度**文件夹并完成以下步骤：
    - 双击**数据**列表中的 **SALES** 量度以将它添加至表达式。
    - 双击**运算符**列表中的 - 运算符。
    - 双击**数据**列表中的 **COGS** 量度。
    - 双击**运算符**列表中的 + 运算符。
    - 双击**数据**列表中的 **EXPENSE** 量度。
    - 在**表达式**字段中，突出显示内容如下的表达式部分：  
`@Measure(CVSAMPLE.COGS)+@Measure(CVSAMPLE.EXPENSE)`，然后双击**运算符**列表中的 **(..)** 运算符以将表达式的选定部分括在圆括号中。

第 63 页的图 18 显示了可在“SQL 表达式构建器”窗口中创建。

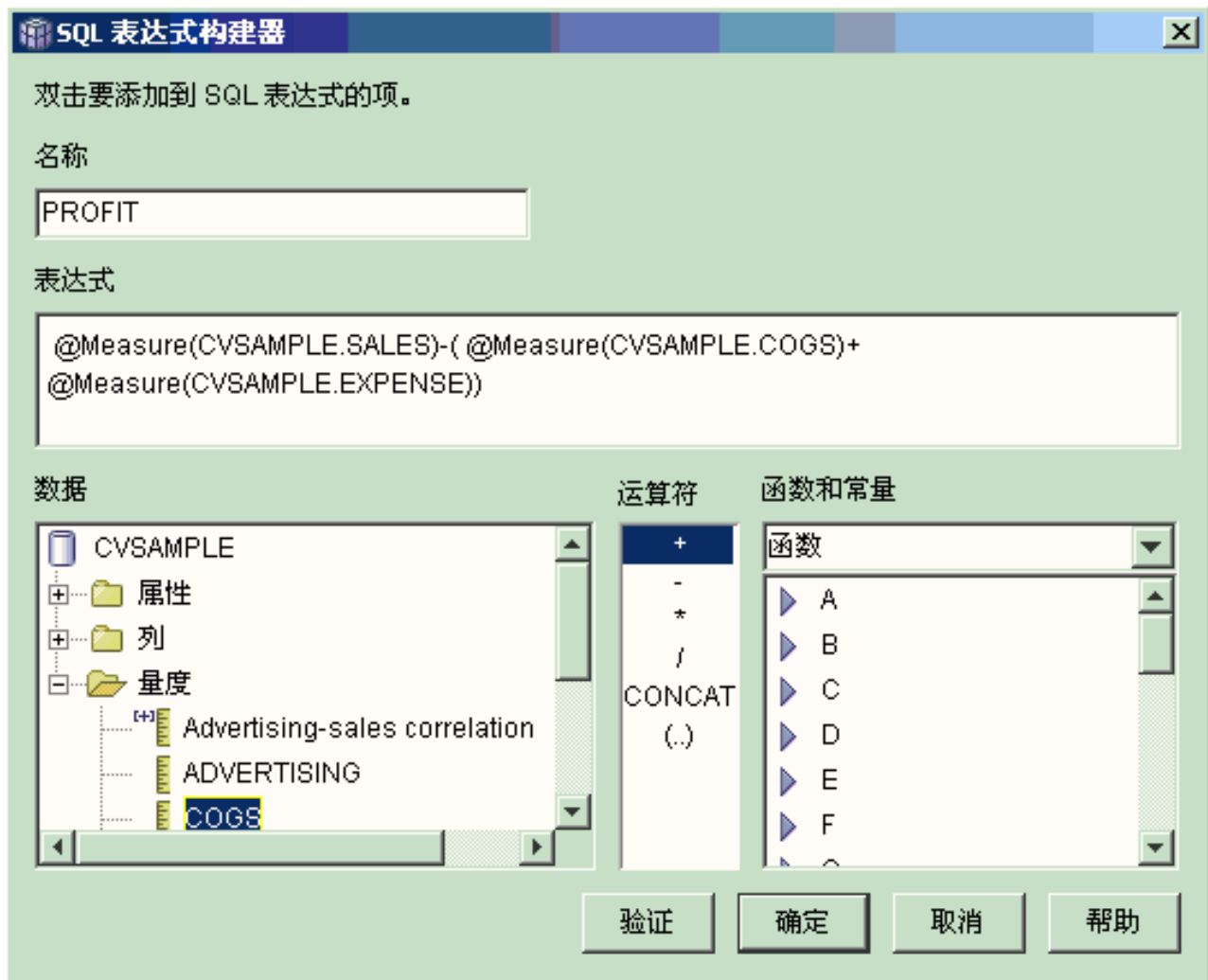


图 18. 在 SQL 表达式构建器窗口中完成 profit 表达式

- d. 单击**确定**以创建 Profit 量度并关闭“SQL 表达式构建器”窗口。
  - e. 在“聚集”页上，单击 Profit 量度的聚集，然后选择 SUM 函数。Profit 量度就完成了。
3. 创建 Profit Margin 量度:
    - a. 在“量度”页上，单击**创建计算的量度**。“SQL 表达式构建器”窗口打开。
    - b. 在“SQL 表达式构建器”窗口中，在名称字段中输入 PROFIT MARGIN。
    - c. 要创建 Profit Margin 表达式，展开数据列表中的量度文件夹并完成以下步骤:
      - 双击数据列表中的 **PROFIT** 量度以将它添加至表达式。
      - 双击运算符列表中的 / 运算符。
      - 双击数据列表中的 **SALES** 量度。
      - 通过在表达式字段中进行输入来将整个表达式括在圆括号中。
      - 将光标定位在表达式的末尾并双击运算符列表中的 \* 运算符。
      - 在表达式字段中，在表达式的末尾输入 100。

图 19显示了可在“SQL 表达式构建器”窗口中创建的 profit margin 表达式。

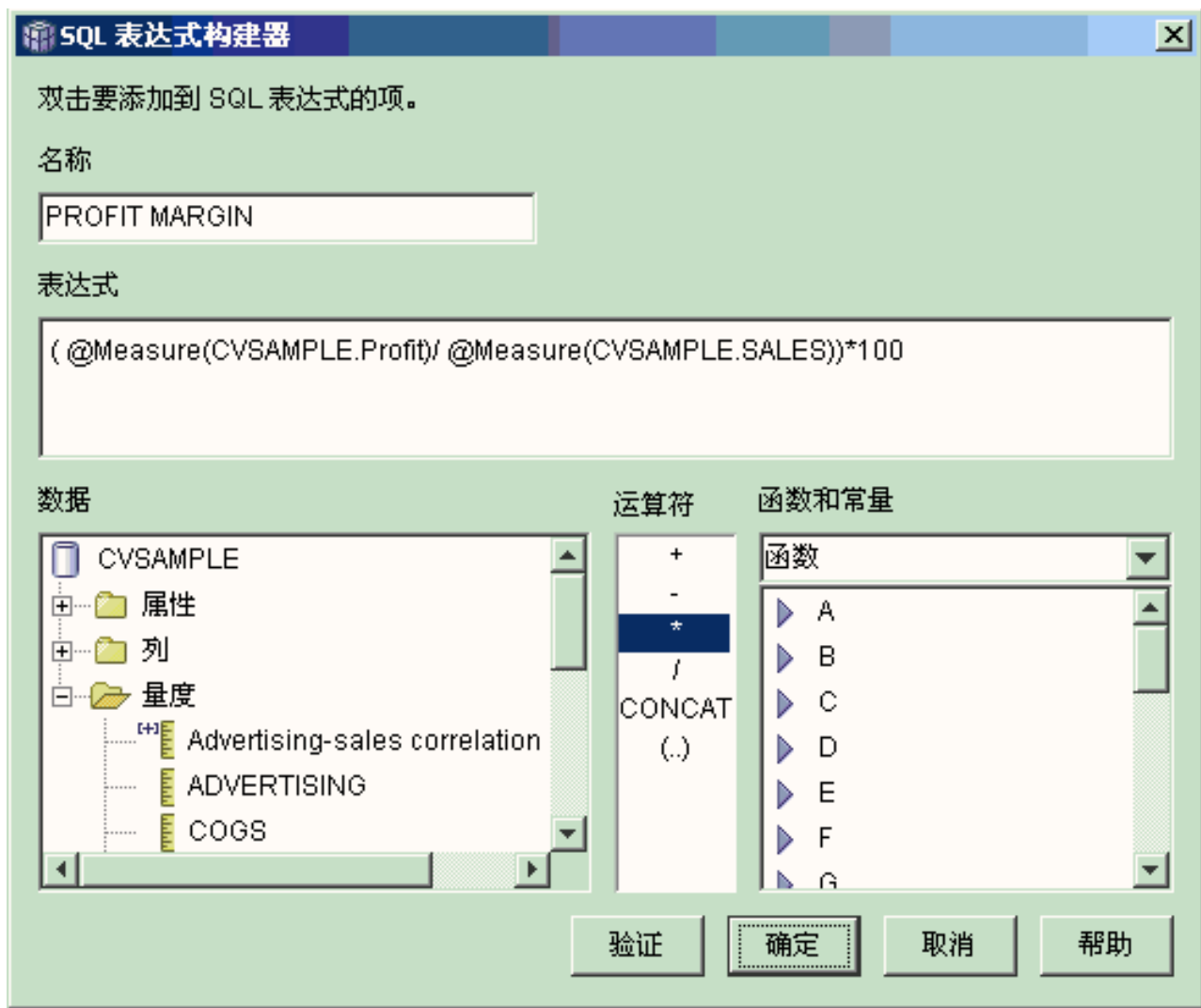


图 19. 在 SQL 表达式构建器窗口中完成 profit margin 表达式

- d. 单击**确定**以创建 Profit Margin 量度并关闭“SQL 表达式构建器”窗口。

在“聚集”页上，缺省情况下，OLAP 中心将聚集函数设置为 NONE 以使用组合量度，因此不需要更改聚集函数。

- 4. 单击**确定**以关闭“事实属性”窗口并保存添加至事实对象的两个新量度。

在 DBA 创建了这两个量度之后，就可以对这些重要的量度完成其它分析了。

## 对因特网订单数量进行计数

零售公司几年前已经通过增加互联网销售来扩展业务。现在该公司想要分析互联网销售成果。该公司首先要计算的其中一项就是通过互联网完成的订单数量。



## 方案详细信息

该公司的数据库具有因特网订单（包含 ORDER\_ID、PRODUCT\_ID、QUANTITY 和 TIME\_ID 列）的事实表。PRODUCT\_ID 列包含在相应的订单中售出的每种产品，而 QUANTITY 列存储订单中购买的产品的数量。具有多种产品的订单具有的条目数跟该订单中售出的唯一产品数一样。例如，表 33 显示了三个订单，其中订单 1 包含三个产品 A、一个产品 O 和一个产品 C。

表 33. 部分事实表内容

ORDER_ID	PRODUCT_ID	QUANTITY
1	A	3
1	O	1
1	G	1
2	L	1
2	Q	2
3	P	5

DBA 可以创建 Order Count 量度，它对 ORDER\_ID 列中的每个唯一条目进行计数。Order Count 量度是通过在 SQL 表达式中使用 DISTINCT 关键字并将 COUNT 函数用作所有维中的聚集来定义的。该量度的 SQL 表达式将创建一系列各不相同的订单，在聚集期间会对这些订单进行计数。由于该量度不涉及任何合计，所以称它为非加法量度。

当具有想要计数的字符数据或其它数据时，非加法量度也非常有用。例如，可使用非加法量度来将产品装运至的邮政编码数进行计数。

在此示例中，DBA 决定定义直接映射至 ORDER\_ID 列的 Order Count 量度。但是，您可以选择用同样的方法使用 ORDER\_ID 列。根据在 SQL 表达式中是使用列还是量度，缺省聚集会有所不同，但在任一情况下，都需要将缺省聚集更改为 COUNT 函数，如在『创建量度的步骤』中所述。

### 创建量度的步骤

下列步骤说明可如何使用“OLAP 中心事实属性”窗口来在现有事实对象中创建 Order Count 量度。

1. 要打开“事实属性”窗口，右键单击 OLAP 中心对象树中的事实对象并单击**编辑量度**。“事实属性”窗口打开。
2. 在“量度”页上，单击**创建计算的量度**按钮。“SQL 表达式构建器”窗口打开。
3. 在“SQL 表达式构建器”窗口中，在**名称**字段中输入 ORDER COUNT。
4. 要创建 order count 表达式，展开**数据**列表中的**量度**文件夹并完成以下步骤：
  - 在**函数和常量**字段中，选择**其它**。在其它函数和常量列表中，双击 **DISTINCT** 关键字。
  - 双击**数据**列表中的 **ORDER ID** 量度。

第 66 页的图 20 显示了可以在“SQL 表达式构建器”窗口中创建的 order count 表达式。

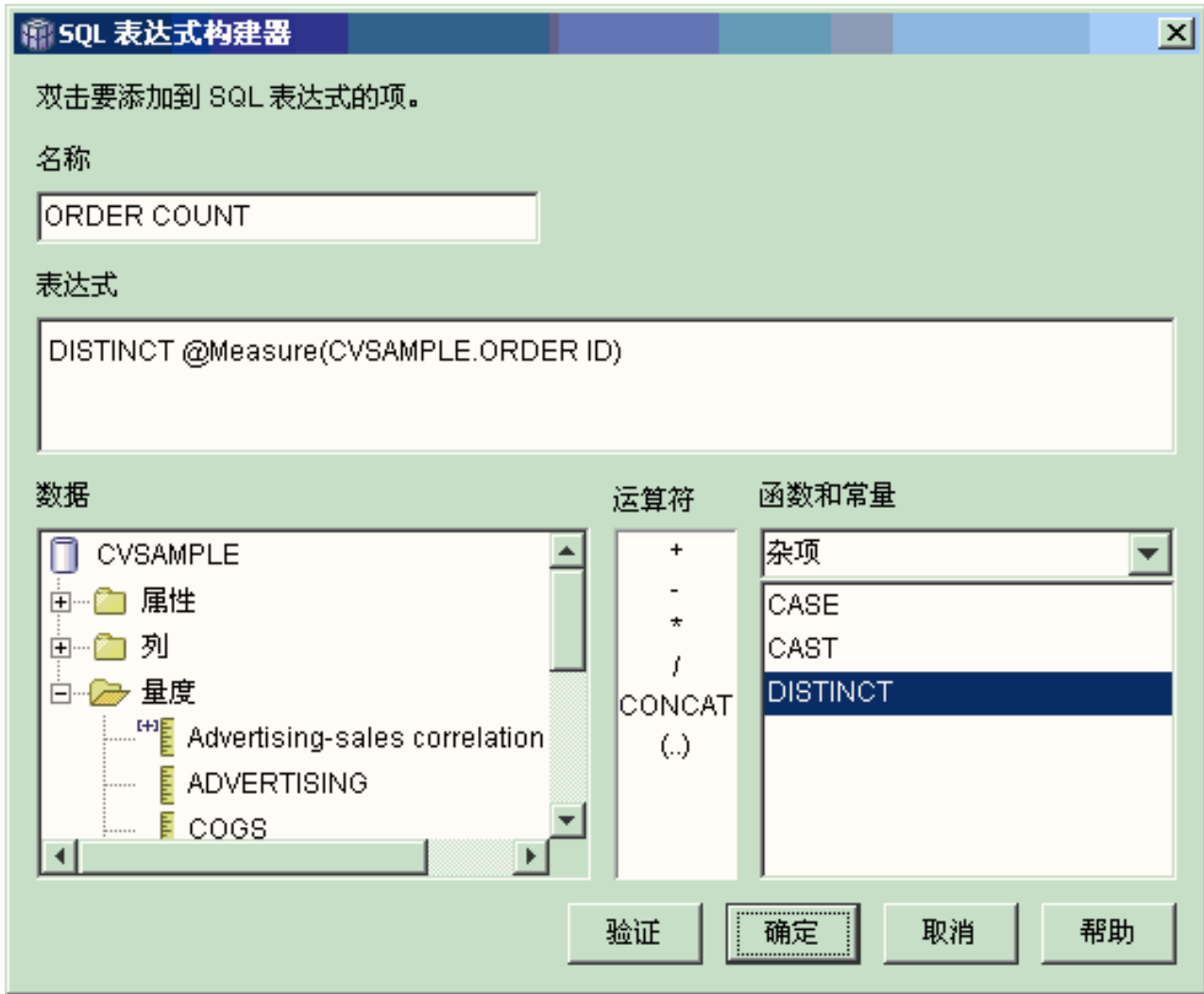


图 20. 在 SQL 表达式构建器窗口中完成 *order count* 表达式

5. 单击**确定**以关闭“SQL 表达式构建器”窗口。
6. 在“聚集”页上，单击 Order Count 量度的聚集，然后选择 **COUNT** 函数。
7. 单击**确定**以保存对事实对象的更改并关闭“Facts 属性”窗口。

现在就有了对各不相同的 Order ID 行值进行计数的量度。可以将此量度与其它量度配合使用以进一步分析数据。

## 将销售数字分等级

办公用品连锁商店在过去的几年里得到飞速发展。业务经理在考虑关闭某些业绩最差的商店以削减成本和增加利润。

商店的销售历史是决定是否关闭商店的重要因素。分析员必须能够对销售数字分等级并下寻各维以完成他们的分析。

## 方案详细信息

办公用品商店的数据库有一个包含 Sales 列和其它列的事实表。数据库还有若干个维表。DBA 可以创建使用 RANK 函数的 Sales Rank 量度，此函数是 DB2 通用数据库 (DB2 UDB) 提供的一个 OLAP 函数。

DB2 Cube Views 支持 DB2 UDB 提供的下列 OLAP 函数:

### RANK

对行进行排序并对每个行指定一个等级。该等级被定义为 1 加上前面的行数 (对于排序来说是唯一的)。如果两行或多行因为具有重复的行值而不能确定它们的相对顺序, 则指定同一个等级号。如果存在重复的行值, 则分等级的结果在编号中可能会有间隔。第 68 页的表 34 显示了一个示例, 该示例演示对于一组样本行值, RANK 函数产生什么分等级结果。

RANK 函数的典型语法是:

```
RANK ( ) OVER (ORDER BY sort-key-expression expression-order)
```

其中 *sort-key-expression* 是要分等级的一组数据, *expression-order* 是关键字 **ASC** 或 **DESC**, 它以升序顺序或降序顺序对 *sort-key-expression* 的值进行排序。DB2 Cube Views 要求 *sort-key-expression* 是现有量度而不是列或属性。另外, DB2 Cube Views 不支持 DB2 UDB 随此函数提供的 PARTITION BY 子句。有关 RANK 函数的更多信息在 DB2 UDB 信息中心中提供。

### DENSERANK

对行进行排序并对每个行指定一个等级。行的等级被定义为 1 加上严格在该行前面的行数。因此, 分等级的结果将是按顺序的, 等级编号中没有间隔。第 68 页的表 34 显示了一个示例, 该示例演示对于一组样本行值, DENSERANK 函数产生什么分等级结果。

DENSERANK 函数的典型语法是:

```
DENSERANK ( ) OVER (ORDER BY sort-key-expression expression-order)
```

其中 *sort-key-expression* 是要分等级的一组数据, *expression-order* 是关键字 **ASC** 或 **DESC**, 它以升序顺序或降序顺序对 *sort-key-expression* 的值进行排序。DB2 Cube Views 要求 *sort-key-expression* 是现有量度而不是列或属性。另外, DB2 Cube Views 不支持 DB2 UDB 随此函数提供的 PARTITION BY 子句。有关 DENSERANK 函数的更多信息在 DB2 UDB 信息中心中提供。

### ROWNUMBER

通过排序计算行的顺序行号, 第 1 行是 1。如果未指定 ORDER BY 子句, 则按任意顺序将行号指定给行。

ROWNUMBER 函数的典型语法是:

```
ROWNUMBER ( ) OVER ([ORDER BY sort-key-expression expression-order])
```

其中 *sort-key-expression* 是要分等级的一组数据, *expression-order* 是关键字 **ASC** 或 **DESC**, 它以升序顺序或降序顺序对 *sort-key-expression* 的值进行排序。DB2 Cube Views 要求将现有量度 (而不是列或属性) 用作此函数的数据源。另外, DB2 Cube Views 不支持 DB2 UDB 随此函数提供的 PARTITION BY 子句。有关 ROWNUMBER 函数的更多信息在 DB2 UDB 信息中心中提供。

这些 OLAP 函数未列在“SQL 表达式构建器函数”和常量列表中。

表 34. 使用 RANK 和 DENSERANK 函数对一组样本值分等级的结果

行值	排序	RANK 函数产生的分等级结果	DENSERANK 函数产生的分等级结果
100	1	1	1
35	2	2	2
23	3	3	3
8	4	4	4
8	4	4	5
6	5	6	6

### 创建量度的步骤

下列步骤说明可如何使用“OLAP 中心事实属性”窗口来在现有事实对象中创建 Sales Rank 量度。

1. 要打开“事实属性”窗口，右键单击 OLAP 中心对象树中的事实对象并单击**编辑量度**。“事实属性”窗口打开。
2. 在“量度”页上，单击**创建计算的量度**以创建 Sakes Rank 量度。“SQL 表达式构建器”窗口打开。
3. 在“SQL 表达式构建器”窗口中，在**名称**字段中输入 SALES RANK。
4. 要创建 Sales Rank 表达式，完成下列步骤：
  - 在“表达式”字段中输入以下函数语法：RANK ( ) OVER (ORDER BY measure DESC)。
  - 展开**数据**列表中的**量度**文件夹。
  - 突出显示函数语法中**表达式**字段中的词语 measure，然后双击 **SALES** 量度以将 SALES 量度添加至表达式。

最终表达式显示在第 69 页的图 21 中。

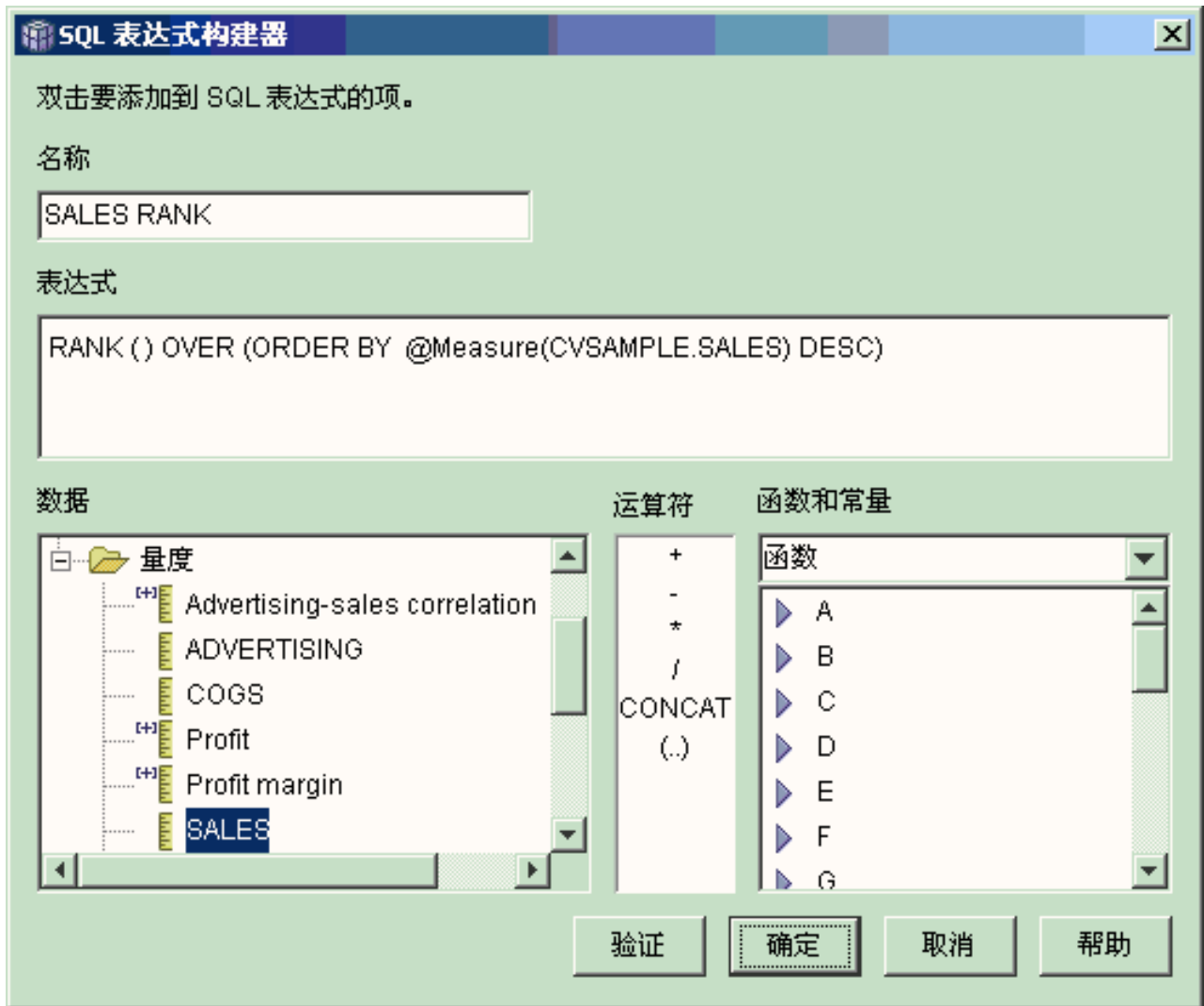


图 21. 在 SQL 表达式构建器窗口中完成 Sales Rank 表达式

- 单击**验证**以确保表达式有效。单击**确定**以关闭“SQL 表达式构建器”窗口。

不需要在“聚集”页上更改缺省聚集 None。None 选项是 Sales Rank 量度的缺省选项，这是因为数据源是数字的并且仅引用现有量度。

通过在量度表达式中使用 RANK 函数来按降序顺序对 Sales 列进行排序，分析员可以下寻其它维以确定特定产品或存储在数据库中任何其它维数据的去年销售历史最差的商店。

## 使用存储在事实表中的时间数据来创建 Time 维

零售公司 XYZ Retail 要在 DB2 Cube Views 中对其销售交易数据建模以便能够更有效地分析这些数据。但是，因为数据的交易特性，可用的唯一时间信息就是每笔交易的相应日期。

在 Time 维中建模的时间信息是将上下文添加至许多公共计算（如按季度分析销售趋势和计算每个星期的平均库存值）所需要的。

许多 DBA 都会避免将时间数据存储为交易的日期或时间戳记，这是因为如果在某一天没有交易，则数据中就会存在一个洞，这在准确聚集和显示数据时会产生问题。通常，最好对时间表中的时间数据建模。但是，XYZ Retail 的 DBA 确信每天都至少会有一笔交易，所以决定保留数据的当前结构。

### 方案详细信息

XYZ Retail 具有一个事实表，它包含有关每笔交易的可量度数据，包括 Sales、Costs、Quantity Sold 和 Date。另外，数据库包含 Region 维表和 Product 维表。问题是：时间数据包括在事实表中而不是存储在独立的维表中。DBA 必须根据事实对象中的日期数据创建维对象。

根据事实表中的日期数据的单个列创建 Time 维有两个独特需求：

- 由于有效立方体模型中的所有维对象都必须连接至事实对象，而且 time 维对象和事实对象都基于同一个事实表，所以必须通过使用将事实表连接至它本身的自连接将 time 维对象连接至事实对象。
- DBA 必须构建将日期数据聚集到更有意义的层（如 Week、Month、Quarter 和 Year）中的计算的属性。

自连接是将表连接至它本身的连接类型，在此示例中，表就是事实表。自连接应连接可一起唯一地标识事实表中的任何行的一个或多个列。最好使用主键。但是，如果未定义主键，则主键的优秀候选项是用来将事实表与维表连接在一起的一组列。要优化立方体模型，必须定义主键。连接基数必须是 1:1，连接类型必须是内连接。

第 71 页的图 22 显示了事实对象、基于事实表的维和事实-到-维连接可如何映射至同一个事实表。

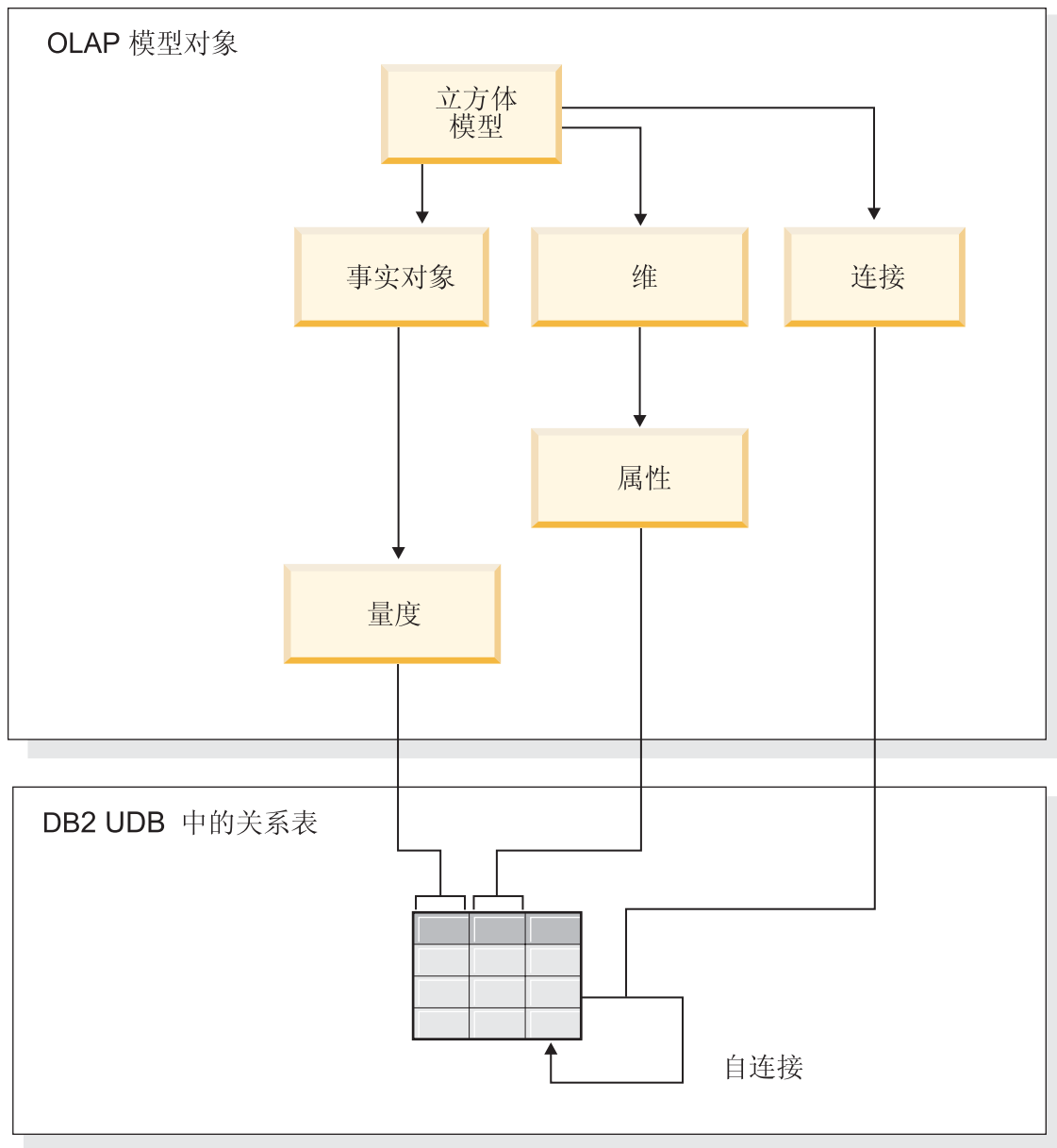


图 22. 自连接如何将表连接至它本身

### 创建属性和维的步骤

下列步骤说明可如何使用“OLAP 中心维”向导来根据事实表创建 Time 维和计算的属性:

1. 要打开“维”向导，右键单击 OLAP 中心对象树中的立方体模型，然后单击**创建维**。“维”向导打开。
2. 在“名称”页上，在**名称**字段中输入 Time。可以有选择地更改业务名以及输入注释。单击**下一步**。
3. 选择立方体模型的事实表。单击**下一步**。不需要指定维连接，原因是维中只有一个表。在“维连接”页上，单击**下一步**。
4. 在“维属性”页上，选择**时间戳**记列。
5. 可选：创建将时间戳记数据聚集到更大的块（如 Month、Quarter 和 Year）中的其它计算的属性。要创建计算的属性，单击**创建计算的属性**按钮以打开“SQL 表达式构

建器”并为将源时间戳记列计算到月、季度和年中的每个属性定义表达式。在定义了每个计算的属性之后，单击**验证**按钮以确保该表达式有效，然后单击**确定**以关闭“SQL 表达式构建器”并返回到“维”向导。在选择并创建了想要的所有属性之后，单击**下一步**。

6. 在“维类型”页上，选择 **Time**。单击**下一步**。
7. 在“事实-维连接”页上，单击**创建连接**。在打开的“连接”向导中，创建自连接。输入一个名称并单击**下一步**。为左边的属性和右边的属性选择唯一地定义事实表中的任何行一列或一组列（如主键）。选择一对时间并单击**添加**以将属性对添加至连接。选择内连接类型和 1:1 基数。在添加了必要的属性对之后，单击**完成**。“连接”向导就会关闭。
8. 在“事实-维连接”页上，单击**完成**。

在立方体模型中定义了 Time 维的情况下，XYZ Retail 就可以添加对于其数据分析有意义的新层。现在，他们可以执行与时间有关的分析（包括库存）。



---

## 第 5 章 DB2 Cube Views 立方体模型优化

本节描述下列主题:

**摘要表** DB2 Cube Views 使用 DB2 摘要表改进向立方体模型发出的查询的性能。摘要表是一种特殊类型的具体查询表 (MQT)，该表专门包括摘要数据。

### 带有功能相关性和约束的摘要表

优化顾问程序利用有关数据之前的关系的信息 (例如, 功能相关性和约束) 来建议包含聚集量度和层属性的摘要表, DB2 优化器需要这些摘要表来有效地应答查询。

### 优化过程概述

使用 DB2 Cube Views 优化星型模式或雪花模式可改进 OLAP 式的 SQL 查询的性能。优化过程包括创建、实现和维护优化顾问程序建议的摘要表。

### 关于优化的元数据设计注意事项

设计元数据对象 (包括层和层次结构、量度、立方体和优化片) 的方法会影响优化顾问程序向导建议的摘要表。

### 立方体的优化片

优化片是可选的, 但它的功能非常强大, 可以帮助指导优化顾问程序提供重点放在立方体模型的最重要区域的摘要表。

### 分析查询以获取优化片

优化片是一个功能非常强大的工具, 可以改进查询性能, 但仅当它们准确反映用户查询时才有效。

### 用于优化的约束定义

约束为优化顾问程序和 DB2 优化器提供重要信息。必须为星型模式或雪花模式中的外键和主键定义信息性约束或强约束。

### 优化顾问程序的参数

为优化顾问程序向导提供的有关每个参数的信息将影响向导建议的摘要表以及获得的性能改进。务必提供准确信息, 并在成本和性能需求之间作出小心谨慎的决定。

### 优化立方体模型

通过优化针对立方体模型执行的查询, 您可以提高发出 OLAP 式 SQL 查询的产品的性能。

### 用于创建摘要表的 SQL 脚本的示例

优化顾问程序向导提供 SQL 脚本来创建建议的摘要表。SQL 脚本包含构建一个或多个摘要表所需的 SQL 命令。

### 测试查询结果

使用优化顾问程序创建摘要表之前和之后, 您可以使用 DB2 通用数据库中的 db2batch 基准工具来为查询性能结果制定基准。

### 对摘要表进行故障诊断

如果在创建摘要表之后查询性能没有改进, 可使用 DB2EXPLAIN 设施来对查询重路由进行故障诊断。

### 摘要表维护

当基本表中的数据更改时，您需要更新摘要表。您可以通过两种不同方式更新摘要表：立即刷新或延迟刷新。

### 删除摘要表

当删除立方体模型时，DB2 Cube Views 不会删除关联的摘要表。如果未将摘要表用于任何其它用途，则可以删除表以释放磁盘空间。

---

## 摘要表

DB2 Cube Views 使用 DB2 摘要表改进向立方体模型和立方体发出的查询的性能。摘要表是一种特殊类型的具体查询表（MQT），该表专门包括摘要数据。

因为优化顾问程序始终建议使用带有摘要数据的 MQT，所以 DB2 Cube Views 文档中使用术语摘要表来描述建议的 MQT。

您可以提前为查询完成大量计算和连接，并将此数据存储于摘要表中。当运行可使用预计算数据的查询时，DB2 UDB 将查询重新路由至摘要表。查询不需要与预计算完全匹配。如果使用简单的分析（如 SUM 和 COUNT），DB2 UDB 可以动态地从预计算数据中聚集结果。一个摘要表即可满足多个不同的查询。使用摘要表可以显著地改进以下查询的查询性能：这些查询访问常用数据或涉及一个或多个维或表的聚集数据。

第 75 页的图 23 显示了一个基于雪花模式的立方体模型，它包含 Sales 事实对象以及 Time、Market 和 Product 维。事实对象具有量度和属性，而每个维都具有一组属性，每个维通过事实-到-维连接来连接至事实对象。

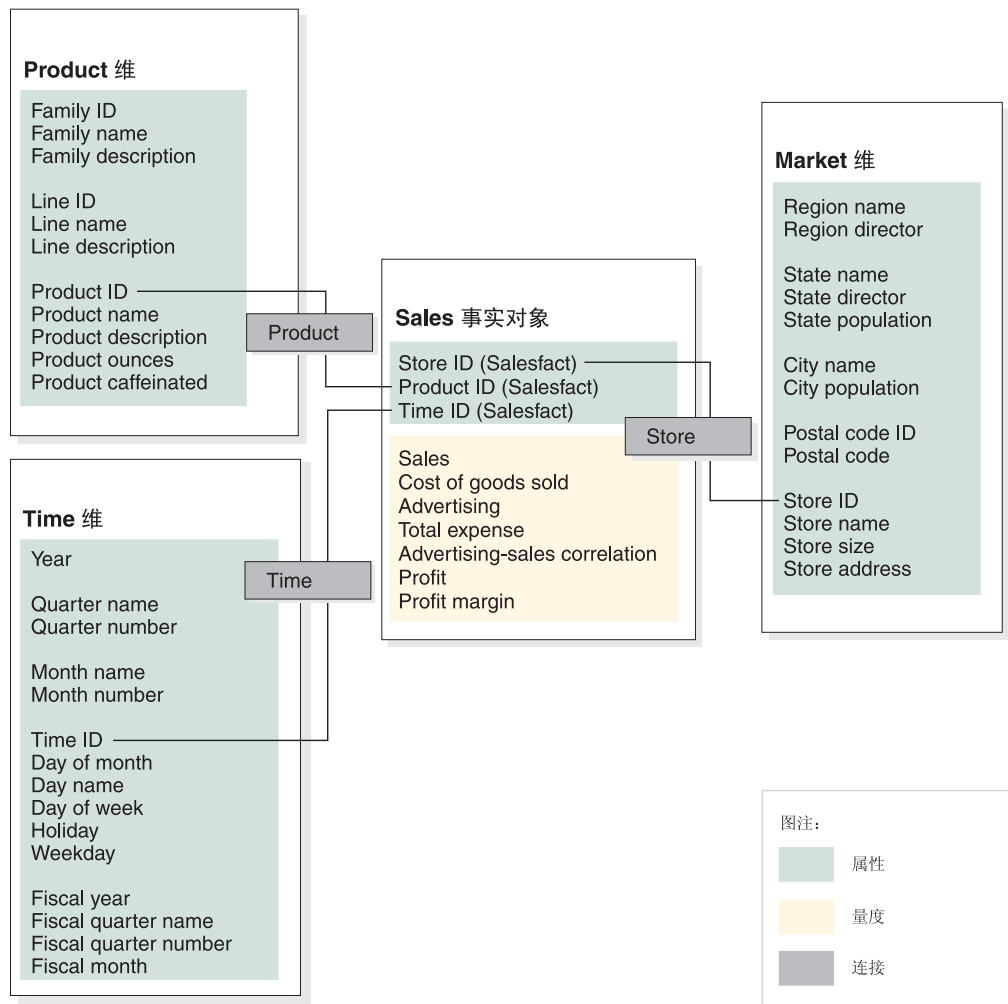


图 23. 立方体模型. 带有 Sales 事实对象以及 Time、Product 和 Market 维的立方体模型

第 76 页的图 24 中显示了立方体模型中每个维的层次结构。穿过层次结构底部的粗黑线连接的突出显示的框表示基本表中实际存在的数据。Sales 数据在 Day 层、Store 层和 Product 层进行存储。必须聚集层次结构中基本层以上的数据。如果您查询基本表以获取特定月份的销售数据，DB2 UDB 必须动态添加每天的销售数据以返回每月销售图。例如，可以使用以下查询来查看 2004 年每个月每个地区的每条产品线的销售数据：

```
SELECT LINE_ID, REGION_NAME, MONTH_NUMBER, SUM(SALES)
FROM TIME, STORE, LOCATION, PRODUCT, LINE, SALESFACT
WHERE SALESFACT.STOREID = STORE.STOREID
  AND STORE.POSTALCODEID = LOCATION.POSTALCODEID
  AND SALESFACT.PRODUCTID = PRODUCT.PRODUCTID
  AND PRODUCT.LINEID = LINE.LINEID
  AND SALESFACT.TIMEID = TIME.TIMEID
  AND YEAR = '2004'
GROUP BY LINEID, MONTH_NUMBER;
```

第 76 页的图 24 中连接 Line-Region-Month 片的细线表示查询存取的片。Line-Region-Month 是立方体模型的一个片，它包括每个层次结构的一个层。可定义摘要表以满足在特定片或之上的查询。可对该查询存取的 Line-Region-Month 片构建摘要表。可以通过带有某些附加聚集的摘要表来满足存取包括 All Time、Year、Quarter、All Markets、All Products 和 Family 的片或之上的数据的其它任何查询。但是，如果要查

询该片之下更详细的数据，如 Day 或 City，DB2 UDB 不能将摘要表用于这一更详细的查询。

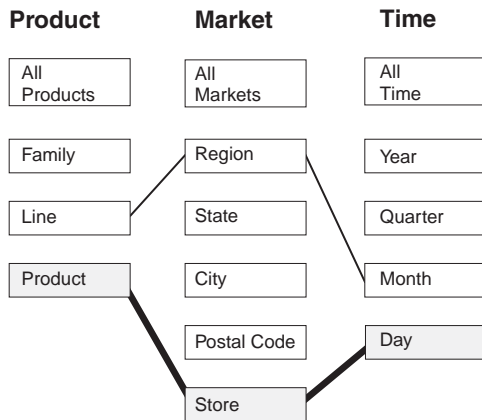


图 24. 查询片. Product、Market 和 Time 层次结构。显示 Line-Region-Month 片以及 Product-Store-Day 片中存在的基本数据。

在图 25中，虚线定义了 Line-State-Month 片。为 Line-State-Month 片构建的摘要表可以满足访问处于该片或该片之上的数据的任何查询。为 Line-State-Month 片构建的摘要表能满足的所有数据都包括在顶层那组突出显示的框中。

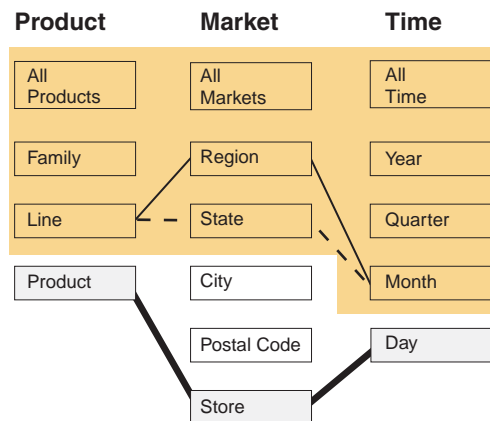


图 25. 摘要表片. Product、Market 和 Time 层次结构。突出显示的数据可由在 Line-State-Month 片构建的摘要表来满足。

DB2 SQL 编译器中的重写程序了解现有摘要表，因此可以自动重写查询来从摘要表（而不是基本表）读取数据。重写的查询通常更快，因为摘要表通常比基本表更小，并包含预先聚集的数据。用户继续编写对基本表的查询。DB2 UDB 将决定何时对特定查询使用摘要表，并将重写用户的查询以存取摘要表，如第 77 页的图 26中所示。重写查询存取包含预先聚集的数据的摘要表。通常摘要表比基本表要小得多（因此也快得多）并且返回的结果与基本表返回的结果相同。

您可以使用 DB2 EXPLAIN 工具查看是否重新路由了查询以及该查询重新路由至哪个表（如果适用的话）。

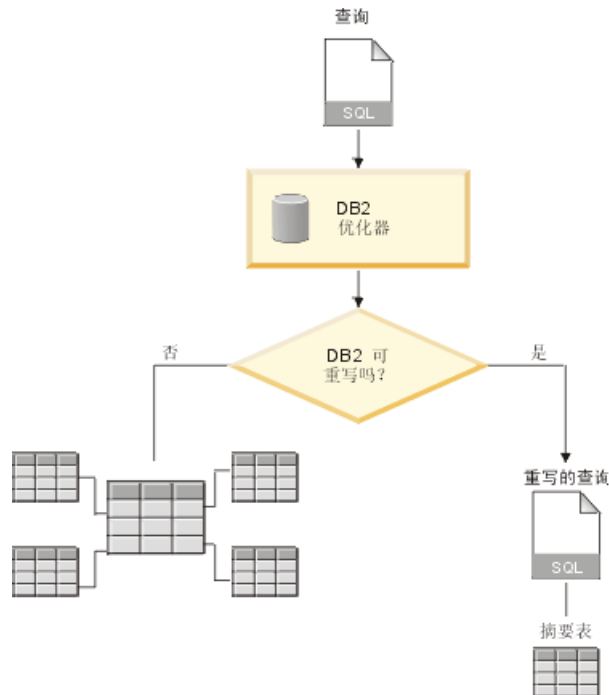


图 26. 查询重写. 用于重写查询的 DB2 UDB 过程

可重写用于查看 2004 年每个月每个地区的每条产品线的销售数据的查询以使用为 Line-Region-Month 片构建的摘要表: 原始查询为:

```
SELECT LINE_ID, REGION NAME, MONTH_NUMBER, SUM(SALES)
FROM TIME, STORE, LOCATION, PRODUCT, LINE, SALESFACT
WHERE SALESFACT.STOREID = STORE.STOREID
  AND STORE.POSTALCODEID = LOCATION.POSTALCODEID
  AND SALESFACT.PRODUCTID = PRODUCT.PRODUCTID
  AND PRODUCT.LINEID = LINE.LINEID
  AND SALESFACT.TIMEID = TIME.TIMEID
  AND YEAR = '2004'
GROUP BY LINEID, MONTH_NUMBER;
```

重写查询为:

```
SELECT LINE_ID, REGION_NAME, MONTH_NUMBER, SUM(SALES)
FROM SUMMARYTABLE1
WHERE YEAR = '2004'
GROUP BY LINE_ID, REGION_NAME, MONTH_NUMBER;
```

重写查询对于 DB2 UDB 来说更为简单且可以更快地完成, 因为数据已经预先聚集并且许多表连接已预先计算, 因此 DB2 UDB 只存取一个很小的表而不是存取六个表 (包括一个很大的事实表)。使用摘要表可以节省很多工作, 具有很大的事实表的模式尤其如此。例如, 带有十亿行的事实表可能已预先聚集至只有一百万行的摘要表, 而且此聚集涉及的计算只进行一次, 而不是每次发出查询时进行一次。存取只有事实表的千分之一的摘要表比存取大型的基本表要快得多。

在此示例中, 第 78 页的图 27 显示 Line-State-Month 片的摘要表。DB2 UDB 需要计算来自较高的 State 层 (而不是来自较低的 Store 层) 的 Region 的数据, 所以州比商店要少, 因此摘要表包含的行比基本表包含的行要少。DB2 UDB 不需要执行任何附加计算即可按 Month 和 Line 返回销售数据, 因为已在这些层聚集数据。此查询通过提前连

接在查询使用的表的摘要表中的数据得到完全的满足，因此不必在发出查询时执行这些连接。对于更复杂的查询，性能增益可能非常明显。

Region name	State name	Line ID	Year	Quarter number	Quarter name	Month number	Sales	Cost of goods	Advertising	Total expense	Profit
West	Idaho	054	2004	1	Qtr 1	2	9700	2500	700	3200	6500
East	Maine	102	2004	2	Qtr 2	5	3000	500	200	700	2300
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

图 27. 摘要表. 为 Line-Region-Month 片创建的摘要表的示例

在某些情况下，查询可能存取与包括在摘要表中的属性相关的属性。DB2 优化器可使用功能相关性和约束以将摘要表与相应的维表以动态方式连接在一起。

当优化顾问程序建议使用摘要表时，立方体模型中的所有量度均包括在内。在此示例中，SalesFacts 对象只有五个量度，包括 Sales、Cost of goods、Advertising、Total expense 和 Profit，它们全部包括在摘要表中。如果为立方体模型定义五十个量度，全部五十个量度都会包括在摘要表中。优化顾问程序不必在摘要表中包括对某层定义的所有相关属性，原因是 DB2 Cube Views 将某层的属性间定义功能相关性。

## 带有功能相关性和约束的摘要表

优化顾问程序利用有关数据之前的关系的信息（例如，功能相关性和约束）来建议包含聚集量度和层属性的摘要表，DB2 优化器需要这些摘要表来有效地应答查询。

DB2 Cube Views 会尽可能在层的属性之间定义功能相关性。当定义层时，会定义层键属性与层中其它属性（缺省属性和相关属性）之间的关系。该关系指示：层键属性可一起用来确定层中的其它属性。DB2 Cube Views 通过在属性之间定义功能相关性来说明层的属性之间的关系。DB2 UDB 和 DB2 Cube Views 可使用由功能相关性定义的关系来对数据执行智能优化。

必须确保层属性的底层数据以功能相关性所描述的方式功能相关。DB2 UDB 不会验证功能相关性的有效性。

如果层键属性与层的相关属性之间存在功能相关性，则优化顾问程序可以将没有相关属性的层键属性包括在摘要表中。仍可将针对层的相关属性的查询路由至摘要表，这是因为 DB2 优化器在发出查询来创建最终结果集时会将摘要表与维表连接起来。

例如，可以使用非常类似于在第 74 页的『摘要表』中描述的查询来按 2004 年中的每个月查看在所有区域中每条产品线的销售数据。以下查询的不同点在于它按 Line 名称（而不是 Line ID）和 Month 名称（而不是 Month 编号）对结果进行分组。对于存取数据而言，此结果集更有用，但依赖于功能相关性和约束。

```
SELECT LINE_NAME, REGION_NAME, MONTH_NAME, SUM(SALES)
FROM TIME, STORE, LOCATION, PRODUCT, LINE, SALESFACT
WHERE SALESFACT.STOREID = STORE.STOREID
AND STORE.POSTALCODEID = LOCATION.POSTALCODEID
```

```
AND SALESFACT.PRODUCTID = PRODUCT.PRODUCTID
AND PRODUCT.LINEID = LINE.LINEID
AND SALESFACT.TIMEID = TIME.TIMEID
AND YEAR = '2004'
GROUP BY LINE_NAME, REGION_NAME, MONTH_NAME;
```

DB2 优化器将重写查询以使它将摘要表连接至适当的维表，如在第 80 页的图 28 中所示。

Line 层包含以下属性:

- 作为层键属性的 Line ID
- 作为缺省属性的 Line 名称
- 作为相关属性的 Line 描述

Line ID 是 Line 表的主键，因此存在对 Line ID 列的约束。由于存在对 Line ID 列的约束，所以 DB2 Cube Views 不会为此层创建功能相关性。DB2 优化器使用该约束来将摘要表连接至 Line 表以及存取查询结果集的 Line 名称数据。

Month 层包含以下属性:

- 用作层键属性的 Year 和 Month 编号
- 作为缺省属性的 Month 名称

对于 Month 层，存在功能相关性，它指定 Month 名称对于 Year 和 Month 编号的组合功能相关。Year 列和 Month 数列都不是 Time 表的主键或外键的一部分，因此不存在对这些列的约束。DB2 优化器使用 Month 名称与 Year 和 Month 编号的组合之间的功能相关性来将摘要表连接至 Time 表以及存取查询结果集的 Month 名称数据。

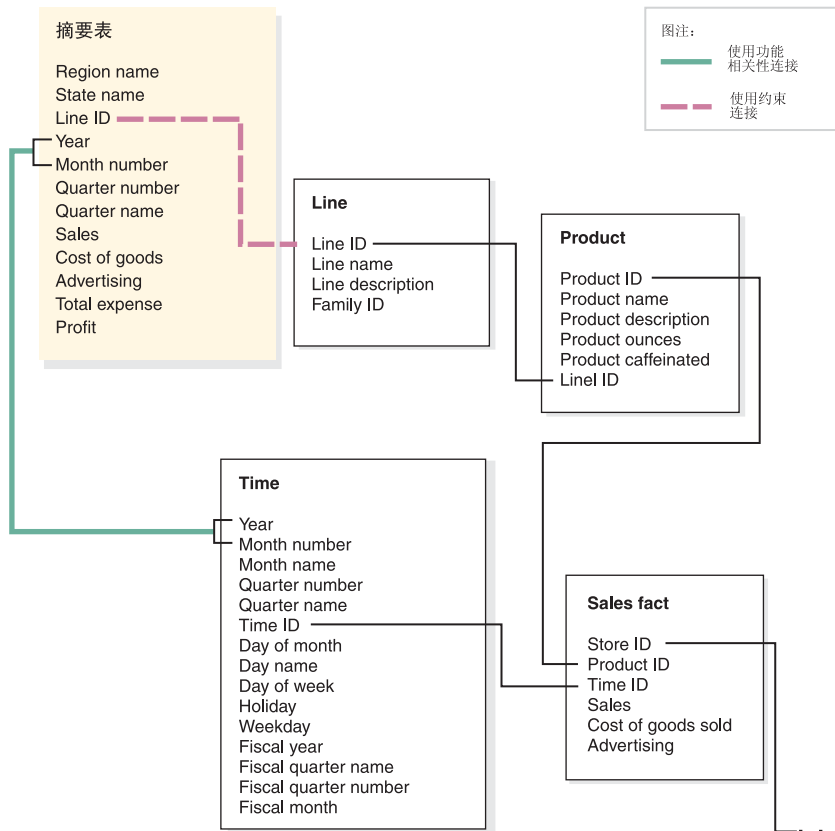


图 28. 连接. 当发出查询时, DB2 优化器动态地将摘要表与适当的维表连接在一起

DB2 优化器将查询重写为:

```
SELECT Q4.LINE_NAME, Q4.REGION_NAME, Q4.MONTH_NAME, SUM(Q4.SALES)
FROM (
  SELECT DISTINCT Q1.LINE_NAME, Q3.REGION_NAME, Q2.MONTH_NAME,
    Q3.SALES, Q2.YEAR, Q2.MONTH_NUMBER
  FROM LINE AS Q1, TIME AS Q2, SUMMARYTABLE1 AS Q3
  WHERE (Q3.YEAR=2004)
    AND Q3.LINEID=Q1.LINEID)
    AND (Q3.MONTH_NUMBER=Q2.MONTH_NUMBER)
    AND (2004=Q2.YEAR)
) AS Q4
GROUP BY Q4.LINE_NAME, Q4.REGION_NAME, Q4.MONTH_NAME
```

生成的查询只连接原查询中的三个表而不是六个表,更重要的是,重写后的查询不必存取通常比较大且很慢的大型事实表。重写后的查询会快很多,原因是摘要表已经包含预先聚集的度量数据。

## 优化过程概述

使用 DB2 Cube Views 优化星型模式或雪花模式可改进 OLAP 式的 SQL 查询的性能。优化过程包括创建、实现和维护优化顾问程序建议的摘要表。

优化顾问程序可以通过建议摘要表来帮助您优化立方体模型。由于 DB2 UDB 摘要表包含可在查询中使用的一个或多个表的预计算结果,因此它们能够提高查询性能。您可以在摘要表中预先计算和存储高成本的表连接和复杂计算,以便将来使用这些聚集的查询可以更快地运行。有关摘要表的信息,请参阅第 74 页的『摘要表』。



优化顾问程序将分析元数据和提供给向导的信息，并建议适当的摘要表。在运行优化顾问程序之后，将获得 SQL 文件，此文件可以构建一组建议的摘要表集。您可以选择修改 SQL，然后运行 SQL 来创建摘要表。

运行优化顾问程序只是优化过程中的一个步骤。在开始优化之前，需要考虑几个问题，包括但不限于：

- 如何在基本表上有效使用 DB2 约束
- 如何定义立方体模型以使其遵循优化验证规则
- 要对何种类型的查询进行优化
- 需要提供多少空间
- 如何维护摘要表以使它们包含的数据是最新的

在优化之前，必须在基本表上定义约束。有关所需约束的类型的信息，请参阅第 93 页的『用于优化的约束定义』。

优化过程的许多部分是重复的并且可能需要重复执行它们来微调并保持性能增益。图 29 概述了优化过程中的主要步骤。

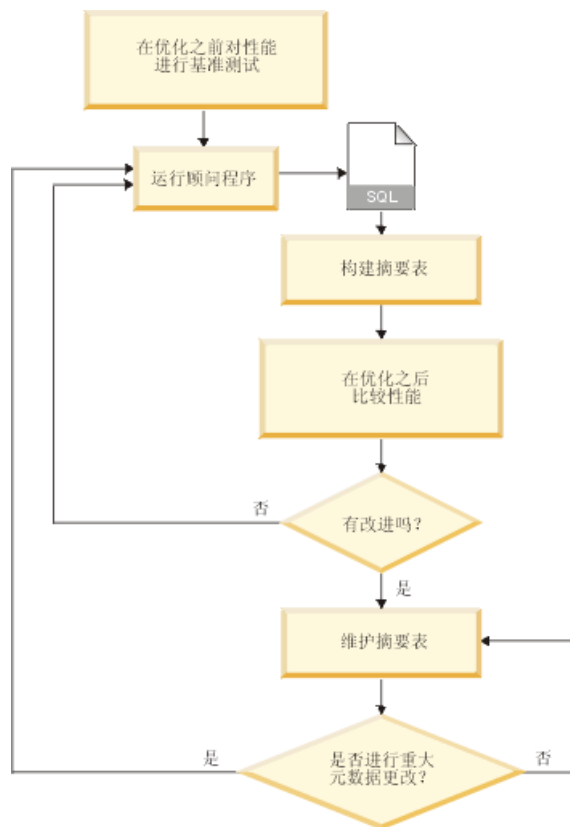


图 29. 优化过程. 优化过程中的主要步骤的概述

优化过程包括以下常规任务：

- 评估性能

在运行 DB2 Cube Views 优化顾问程序之前，应该为一组特定的典型查询评估当前性能。性能评估是可选的步骤，它提供了一个基准，以便您稍后可以分析优化是否

成功。您可以使用随 DB2 UDB 提供的 db2batch 基准工具创建基准。有关使用 db2batch 的信息，请参阅第 102 页的『测试查询结果』。您可运行样本查询来完成性能基准，但优化顾问程序不需要样本查询，因为它基于元数据并可在不了解将发出的特定查询的情况下提供建议。

- **运行优化顾问程序向导**

将几个重要的参数提供给向导，包括：要对其进行优化的查询类型、磁盘空间和时间局限性、更新方法以及表空间位置。有关这些参数选项的信息，请参阅第 97 页的『优化顾问程序的参数』。优化顾问程序根据您提供的信息、元数据、DB2 统计信息以及允许的任何数据采样来创建其建议。优化顾问程序会考虑指定的参数并生成两个 SQL 文件。一个 SQL 文件包含用来构建一组建议的摘要表的 SQL 命令。另一个 SQL 文件包含用来更新建议的摘要表的 SQL 命令。

- **创建摘要表**

您可以在完成向导之后立即创建摘要表，也可以将操作添加到正常的数据库维护调度表中。创建摘要表可能需要大量时间和处理资源。构建摘要表之后，验证对优化的立方体模型的查询性能是否有所改进。运行您在优化之前运行的同一组样本查询，并比较性能结果。如果未看到显著的性能增益，则可能需要再次运行优化顾问程序并分配更多磁盘空间和 / 或时间，或者可能需要更改其它设置。有关如何验证和分析性能结果的信息，请参阅第 102 页的『测试查询结果』。

- **维护摘要表**

在创建摘要表之后，需要定期维护这些表，以确保它们适当地与数据保持同步。当运行优化顾问程序时，选择立即刷新或延迟刷新更新选项。

#### **立即刷新选项**

如果选择立即更新选项，DB2 UDB 使基本表与摘要表保持同步，并在更改基础表时以增量形式更新摘要表。DB2 UDB 支持对简单聚集（如 SUM 和 COUNT）进行增量维护。对于其它聚集，不论您选择哪个刷新选项，优化顾问程序建议使用延迟刷新选项的摘要表。

#### **延迟刷新选项**

如果选择延迟选项，则需要重新构建摘要表才能更新它们。可以决定何时执行摘要表更新。如果对整个基本表作了重大更改，则延迟更新可能比增量更新更有效率。

在这些选项之间选择时，需要在可以分配给维护的资源 and 数据必须同步的精确度之间进行权衡。有关立即更新和延迟更新选项的更多信息，请参阅第 104 页的『摘要表维护』。

- **定期重新评估**

需要定期重新评估摘要表，以确保其继续满足您的需要。如果通过添加或更新立方体模型来对元数据进行大量更改，则可能需要再次运行优化顾问程序并构建一组新的摘要表：

- 如果添加元数据对象（如新维或量度），则访问新对象中数据的查询将无法使用现有摘要表。但是，不使用新对象的查询将继续使用摘要表。
- 如果更新元数据对象以包括先前没有对其进行优化的数据，则访问已更新对象的查询将无法使用摘要表。

- 如果删除一个或多个对象，则不会更改摘要表的有效性，但是会在不再使用的聚集上浪费磁盘空间。

除重大元数据更改之外，如果定期执行的查询类型更改，并且此类型不是对其进行优化的类型，则也可能需要再次运行向导。

每次运行向导和构建新表时，应再次完成整个优化过程，包括创建基准和分析摘要表的性能。

在删除立方体模型时，如果关联的摘要表不再用于其它任何用途，则您也可以将这些关联的摘要表删除。当删除关联的立方体模型时，DB2 Cube Views 不会删除摘要表。有关如何删除摘要表的信息，请参阅第 104 页的『删除摘要表』。

---

## 关于优化的元数据设计注意事项

设计元数据对象（包括层和层次结构、量度、立方体和优化片）的方法会影响优化顾问程序向导建议的摘要表。

通常您想要根据数据的结构定义事实对象、维和连接的基本结构。对立方体模型中的这些高级对象的基本结构提供的选择较少，因此您几乎不能通过改变这些对象来改进建议的摘要表。但是，在选择定义层和层次结构、量度、立方体和优化片的方式方面会有较大的灵活性。

### 层和层次结构

如果可能，使用第 23 页的『层』中描述的理想建模方法来定义层。通过使用理想建模方法，可减少建议的摘要表使用的磁盘空间量和用于刷新建议的摘要表的临时空间量。

### 量度

如果磁盘空间有限，可以选择在立方体中仅包括关键量度，并删除预计不常用或没有商业价值的所有量度，原因是在立方体中定义的量度越多，摘要表就越大。

包括在立方体中的量度的类型（描述性的或非描述性的）也会影响优化：

- 描述性量度总是从一层聚集至下一层。例如，SUM(Sales) for Quarter 可通过对每个月的销售数据求和计算出来。
- 非描述性量度（例如标准偏差）必须总是直接通过基本数据计算出来，不能从一层聚集至下一层。

总的来说，在立方体模型只包含描述性量度时，优化顾问程序和 DB2 优化器的灵活性和选择范围更大。可优化包含非描述性量度的立方体模型，但如果省略立方体中不需要的非描述性量度，可能会得到更佳优化结果。

### 立方体

设计满足业务需要的立方体。如果构建重点放在立方体模型数据的重要区域的立方体，在优化时立方体将得到最有效的使用。

如果知道用户经常发出的查询的一个或多个特征，可在优化顾问程序的优化片中指定该信息。优化顾问程序会建议改进发送至立方体的指定区域的查询性能的摘要表。优化片是由一组特定层或对每个立方体维指定的 **Any** 层以及在片中应该具有的查询类型（例如，下寻、报表、MOLAP 抽取、混合抽取或深入钻取）定义的。

在 OLAP 立方体中，可指定立方体通常用于一种查询类型（如下寻）。在对立方体指定一种查询类型时，OLAP 中心使用您指定的类型和用于每个立方体维的 **Any** 选项定义优化片。有关优化片的更多信息，请参阅『立方体的优化片』。

以下列表描述可对 OLAP 中心中的立方体指定的查询的类型：

### 下寻查询

下寻查询通常访问集中于立方体模型顶部的数据子集。查询可以转至立方体模型中的任何层。当用户深入一个维的内部时，它们通常停留在其它维的较高层。为下寻查询优化对停留在立方体模型较高层中的查询最为有利。关系 OLAP (ROLAP) 电子表格应用程序通常用于执行下寻查询。例如，电子表格应用程序用户可以通过访问 2004 年所有区域和所有产品的收入来着手。然后，用户可以通过查询所有区域和每个国家或地区的季度收入来进一步查看数据。

通常情况下，性能对于这些类型的查询非常重要，因为这些查询是由必须等待结果处理的用户实时发出的。

### 报表查询

报表查询同样可能会访问立方体模型的任何部分。报表查询通常成批发出。对于报表查询来说，查询性能通常不像下寻查询的查询性能那样重要，因为用户很少等待每个单独查询的即时响应。

### MOLAP 抽取查询

MOLAP 抽取查询访问立方体的基本层或对该立方体定义的优化片。该立方体用来将数据装入到 MOLAP 数据存储中。立方体优化片在逻辑上映射至抽取片以便将数据装入到 MOLAP 应用程序中以进行进一步的处理。

如果对 OLAP 中心中的立方体定义 MOLAP 抽取类型或者如果使用在所有立方体维中指定的 **Any** 选项指定优化片，优化顾问程序将创建要对在立方体的基本层抽取的数据优化的摘要表。确保立方体的基本层映射至在其中进行抽取的片。

### 高级设置

如果对 OLAP 中心中的立方体指定高级设置，可对立方体中频繁查询的特定区域指定优化片。

如果知道频繁发出的查询的一个或多个特征，可在优化顾问程序的优化片中指定该信息。优化顾问程序在建议摘要表时将考虑这些片。优化片是由一组特定层或对每个立方体维指定的任何层以及在片中应该具有的查询类型（例如，下寻、报表、MOLAP 抽取、混合抽取或深入钻取）定义的。有关优化片的信息，请参阅『立方体的优化片』。

---

## 立方体的优化片

优化片是可选的，但它的功能非常强大，可以帮助指导优化顾问程序提供重点放在立方体模型的最重要区域的摘要表。

立方体通常可具有许多立方体维。通过指定一个或多个优化片，可以指定立方体的哪个区域的查询活动最多。优化片是由一组一个或多个层以及在片上期望的查询类型

(如下寻、报表、MOLAP 抽取、混合抽取或深入钻取)定义的。任何类型的查询都可从定义优化片中获益,但报表查询有可能获益最多。

当创建优化片时,必须对每个立方体维指定一个选项。当为每个立方体维指定选项时,遵守以下准则:

- 如果您知道特定的层很重要或经常被查询,则在立方体维中指定该特定层(如 Time 立方体维中的 Month)。
- 如果立方体维的最高聚集很重要或经常被查询,则在立方体维中指定 **All**。
- 如果在某个立方体维中没有哪个层比其它层明显重要一些、该立方体维中的许多层都会被查询或者您不了解该立方体维中各层被查询的情况,则在立方体维中指定 **Any**。

例如,在具有 10 个立方体维的立方体中,可能只在两个立方体维中具有重要的层,则可在那两个立方体维中指定特定层次,而在其它每个立方体维中指定 **Any**。

下列各节描述每种查询类型的示例优化片以及优化顾问程序可能会建议的摘要表:

### 下寻优化片

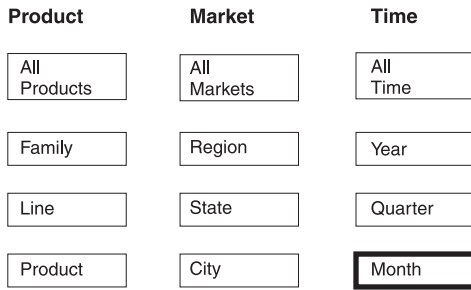
下寻优化片表示用户通常下寻到立方体维中定义的层。因此,优化顾问程序应将这些层包括在一个或多个建议的摘要表中。由于类型是下寻,所以优化顾问程序可能会对某些立方体维中的浅查询和深查询进行优化。

**Recommended:** 仅当知道特定层特别重要时才应对立方体维指定该层。一般情况下,对大部分立方体维指定 **Any** 选项,而仅当特定层涉及大量查询活动时才应选择该层。

第 86 页的图 30 显示了下寻优化片以及优化顾问程序可能会建议摘要表所处的片的示例。此优化片是在 Product 立方体维和 Market 立方体维中的 Any 层以及 Time 立方体维中的 Month 层定义的。此优化片表示用户通常下寻到 Month 层,但在 Product 或 Market 立方体维中并没有特定下寻模式。

可能的摘要表建议包括两个聚集级别。一个聚集级别在 Line-State-Month 片中定义,另一个聚集级别在 Family-Region-Month 片中定义。这两个聚集级别都包括 Time 维的 Month 层中的聚集,但在其它两个维中提供不同的聚集级别。因为这些查询可立即得到满足并且 DB2 UDB 不需要聚集,所以这些查询可立即得到满足并且 DB2 UDB 不需要聚集。越低的聚集级别对越深的下寻查询提供程度越大的性能改善。

## 下寻片



## 可能的建议

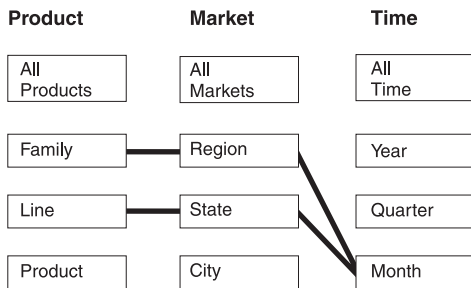


图 30. 下寻. 下寻优化片和可能的摘要表建议

## 报表优化片

报表优化片表示用户通常在立方体维中定义的层创建报表。因此，优化顾问程序应将这些层包括在一个或多个建议的摘要表中。

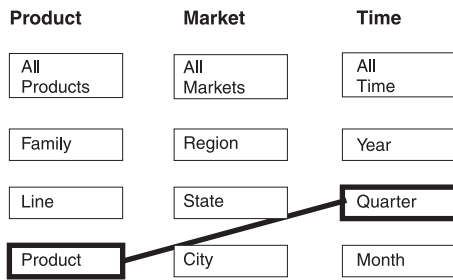
**Recommended:** 仅当知道特定层特别重要时才应对立方体维指定该层。一般情况下，对大部分立方体维指定 **Any** 选项，而仅当特定层涉及大量查询活动时才应选择该层。

第 87 页的图 31 显示了报表优化片以及优化顾问程序可能会建议摘要表所处的片的示例。

此优化片是在 Product 立方体维中的 Product 层、Market 立方体维中的 Any 层以及 Time 立方体维中的 Quarter 层定义的。此优化片表示用户创建这样的报表：包括 Product 层和 Quarter 层并且不包括 Market 立方体维、可能包括 Market 立方体维中的若干个层或者不知道用户包括 Market 区域中的那个层。例如，用户可能通常会创建显示上四个季度每个产品系列的 Sales 数据的报表，但这些报表在按 Region、State 或 City 显示 Sales 数据方面各不相同。

可能的摘要表建议包括两个聚集级别。一个聚集级别在 Product-City-Quarter 片中，另一个聚集级别在 Product-Region-Quarter 片中。这两个聚集级别都包括了在片中指定的 Product 层和 Quarter 层。根据数据采样和其它元数据，优化顾问程序决定创建包括两个聚集级别的摘要表，一个包括 Market 立方体维中的 City 层，另一个包括 Market 立方体维中的 Region 层。

## 报表片



## 可能的建议

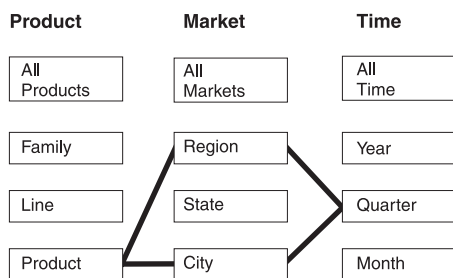


图 31. 报表. 报表优化片和可能的摘要表建议

## MOLAP 抽取优化片

MOLAP 抽取优化片表示通常将指定片中的数据抽取到供应商产品中的 MOLAP 立方体中。因此，优化顾问程序应建议摘要表确保对指定片发出的查询速度很快。

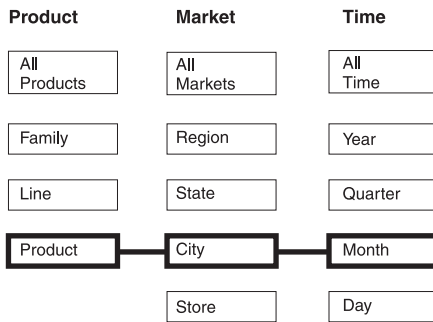
**Recommended:** 对每个立方体维指定特定层，以便优化片与抽取到 MOLAP 立方体中的数据的层相匹配。

对每个立方体只能定义一个 MOLAP 抽取优化片。不能在包含 MOLAP 抽取优化片的立方体中定义混合抽取优化片。

第 88 页的图 32 显示了 MOLAP 抽取优化片以及优化顾问程序可能会建议摘要表所处的片的示例。此优化片是在 Product 立方体维中的 Product 层、Market 立方体维中的 City 层以及 Time 立方体维中的 Month 层定义的。此优化片表示将 Product-City-Month 层中的数据抽取到 MOLAP 立方体中。

可能的建议包括直接满足 Product-City-Month 优化片指定的 MOLAP 抽取查询的摘要表。

## MOLAP 抽取片



### 可能的建议

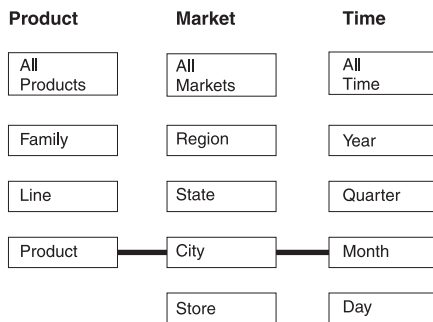


图 32. MOLAP 抽取. MOLAP 抽取优化片和可能的摘要表建议

## 混合抽取优化片

混合抽取优化片表示通常将指定片中的数据抽取到供应商产品中的混合 OLAP (HOLAP) 立方体中。因此，优化顾问程序应将指定的片包括在建议的摘要表中。

**Recommended:** 对每个立方体维指定特定层，以便优化片与抽取到 HOLAP 立方体中的数据的层相匹配。

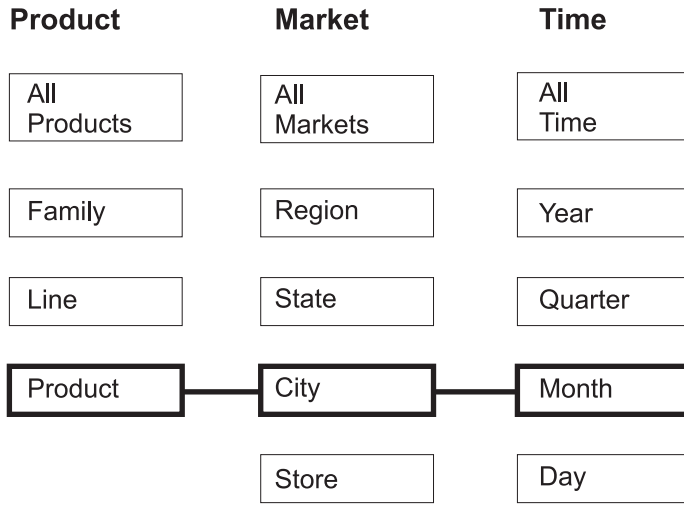
对每个立方体只能定义一个混合抽取优化片。不能在包含混合抽取优化片的立方体中定义 MOLAP 抽取优化片。可以在包含混合抽取优化片的那个立方体中定义零个或多个深入钻取优化片。优化顾问程序预计可能会在指定的混合抽取优化片下面具有深入钻取查询并尝试对指定片下面的深入钻取查询以及指定片中的混合抽取查询进行优化。

第 89 页的图 33 显示了混合抽取优化片以及优化顾问程序可能会建议摘要表所处的片的示例。此优化片是在 Product 立方体维中的 Product 层、Market 立方体维中的 City 层以及 Time 立方体维中的 Month 层定义的。此优化片表示将 Product-City-Month 层中的数据抽取到 HOLAP 立方体中。

可能的摘要表建议包括两个聚集级别。Product-City-Month 片中的聚集级别直接满足优化片指定的 HOLAP 抽取查询。Line-State-Day 片中的聚集级别包括 Day 层（它在混合抽取片下面）以满足 Time 立方体维中可能的深入钻取查询。优化顾问程序分析了其它元数据并执行了数据采样以开发此建议的摘要表。



## 混合抽取片



## 可能的建议

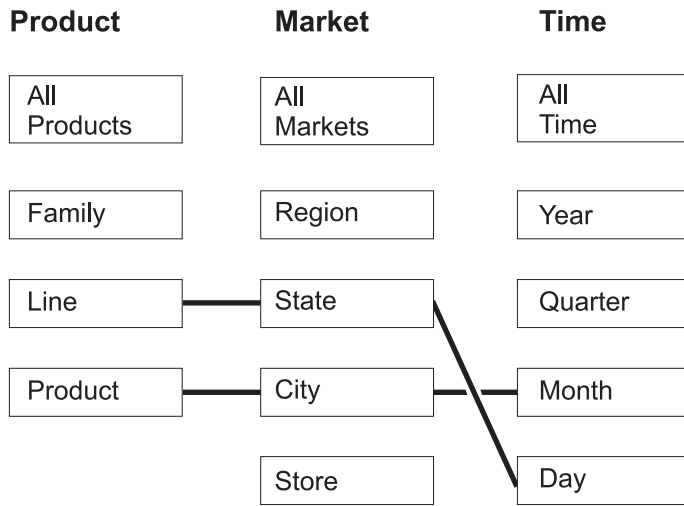


图 33. 混合抽取. 混合抽取优化片和可能的摘要表建议

## 深入钻取优化片

深入钻取优化片必须在立方体中定义有相应的混合抽取优化片。深入钻取优化片表示通常从供应商产品中的混合 OLAP (HOLAP) 立方体深入钻取到指定的片。因此，优化顾问程序应将指定层中或下面的片包括在建议的摘要表中。

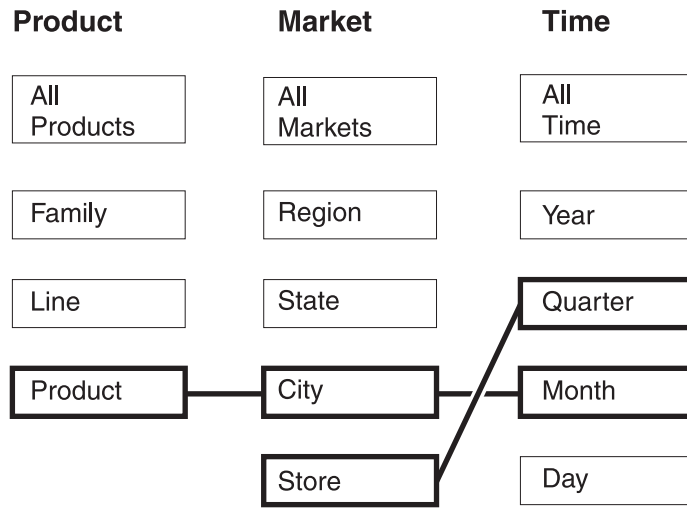
**Recommended:** 仅当知道特定层特别重要时才应对深入钻取类型的优化片指定立方体维中的该特定层。一般情况下，对于深入钻取优化片，对大部分立方体维指定 **Any** 选项，而仅当特定层涉及大量查询活动时才应选择该层。

第 91 页的图 34 显示了深入钻取优化片和相应的混合抽取优化片以及优化顾问程序可能会建议摘要表所处的片的示例。混合抽取优化片是在 Product 立方体

维中的 Product 层、Market 立方体维中的 City 层以及 Time 立方体维中的 Month 层定义的。此优化片表示将 Product-City-Month 层中的数据抽取到 HOLAP 立方体中。深入钻取优化片是在 Product 立方体维中的 Any 层、Market 立方体维中的 Store 层以及 Time 立方体维中的 Quarter 层定义的。此优化片表示 HOLAP 立方体中的深入钻取查询通常包括 Store 层和 Quarter 层以及深入钻取查询可能引用也可能不引用 Product 立方体维中的特定层。

可能的摘要表建议包括两个聚集级别。Product-City-Month 片中的摘要表直接满足优化片指定的 HOLAP 抽取查询。Family-Store-Quarter 片中的聚集级别包括 Store 层和 Quarter 层，这些层是由深入钻取优化片指定的。优化顾问程序分析了其它元数据并执行了数据采样以建议此摘要表同时包括 Product 立方体维中的 Family 层。

## 具有深入钻取的混合抽取片



## 可能的建议

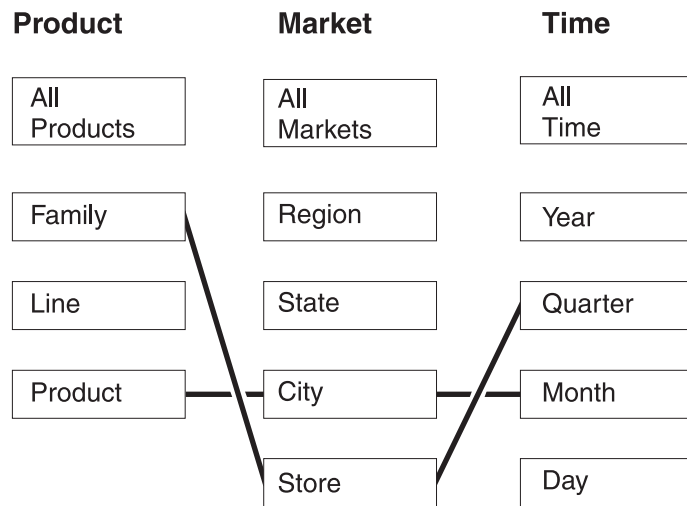


图 34. 深入钻取. 混合抽取优化片和深入钻取优化片以及可能的摘要表建议

## 分析查询以获取候选优化片

优化片是一个功能非常强大的工具，可以改进查询性能，但仅当它们准确反映发出的 SQL 查询时才有效。

当分析用户的查询模式时，考虑以下信息：

- 用户很可能发出的查询类型
- 查询访问频率很高的层次结构的各层
- 用户从中发出查询的产品

要确定您应该为其创建优化片的立方体区域，可查看用户的查询历史。您可能想要调查用户以确定他们的查询需求将来是否会发生转变。您要查找具有最高优先级的优化区域。

**建议：**不要对一个立方体指定三个以上的优化片。

### 对特定情况定义的优化片的示例

每个方案都是基于 CVSAMPLE 数据库中的 Daily sales 立方体的，该立方体具有三个立方体维，包括 Product、Market 和 Time，每个立方体维都具有下列立方体层次结构：

表 35. CVSAMPLE. CVSAMPLE 数据库中 Daily sales 立方体的立方体维和对应的立方体层次结构。

Product 立方体维	Market 立方体维	Time 立方体维
All products	All markets	All time
Family	Region	Year
Line	State	Quarter
Product	City	Month
	Postal code	Day
	Store	

下表中描述的情况是一些示例，这些示例说明可能会对基于 CVSAMPLE 数据库的特定查询装入定义哪些优化片。

表 36. 优化片示例

情况	方案	要定义的优化片
用户使用发出特定查询类型的查询产品	您知道该用户主要使用发出下寻查询的产品。您不知道立方体的哪些区域查询频率较高。	因为您不知道立方体的哪些区域对用户更重要，特定的优化片不能给您提供太大帮助。一般情况下，您可在立方体向导中创建立方体时指定整个立方体用于下寻查询，或在创建立方体之后修改立方体属性。
大多数查询都是同一种类型的且集中在一个层组上。	您知道用户主要发出报表查询而且几乎所有查询都引用 Market 维的 State 层。	您知道有关立方体的特别重要的区域的特定信息，因此优化片能提供很大帮助。您可以定义报表类型的 Any-State-Any 优化片。此片引用 Product 立方体维中的 Any 层、Market 立方体维中的 State 层和 Time 立方体维中的 Any 层。

表 36. 优化片示例 (续)

情况	方案	要定义的优化片
大多数查询都是同一类型的且集中在几个层组上。	您知道该用户主要发出报表查询。大约有一半查询引用 Market 立方体维的 State 层, 另一半查询则随机分布于立方体维的其它各层。	可以定义以下两个优化片以准确地表示此查询分布: <ul style="list-style-type: none"> <li>• 定义具有 Any-State-Any 层的报表优化片来表示引用 State 层的查询。此片引用 Product 立方体维中的 Any 层、Market 立方体维中的 State 层和 Time 立方体维中的 Any 层。</li> <li>• 定义具有 Any-Any-Any 层的报表优化片来表示引用任何立方体维中的任何层的其它查询。通过指定第二个片, 您可以清楚地指示有大量的查询访问 Market 维的其它层。</li> </ul>
大多数查询都是同一类型的且集中在许多层组上。	您知道该用户主要发出报表查询。查询活动不是随机的, 而是分布在立方体的大约 15 个区域中。	优化片的作用是指定查询频率较高的区域, 这种情况下, 查询过于分散因而无法指定一组特定优化片。可在立方体向导中创建立方体时对报表查询指定整个立方体, 或在创建立方体之后修改立方体属性, 而不指定优化片。

## 用于优化的约束定义

约束为优化顾问程序和 DB2 优化器提供重要信息。必须为星型模式或雪花模式中的外键和主键定义信息性约束或强约束。

在使用优化顾问程序之前, 必须先对基本表定义约束。约束需要支持第 33 页的『元数据对象规则』中的基本规则、立方体模型完整性规则和优化规则, 所以立方体模型对优化有效。规则主要定义如何将立方体模型的元数据对象连接在一起。

您可以对需要定义的外键约束使用信息性约束。信息性约束是 DB2 通用数据库版本 8 中提供了一种新型约束。信息性约束提供了一种无需增加维护成本即可提高查询性能的方法。DB2 SQL 编译器可以使用这些约束, 但是数据库管理器不能强制执行它们。此类型的约束使 DB2 UDB 可了解数据中的关系, 而不要求强制执行这些关系。对于主键约束, 必须使用随 DB2 UDB 提供的数据库强制执行的约束。

每个连接需要定义相应的约束。例如, 雪花模式中使用的**事实-到-维**连接和**维-到-维**连接涉及的列都需要约束。

要根据第 94 页的图 35 中显示的雪花模式优化立方体模型, 需要对每个**事实-到-维**连接定义约束。这三个**事实-到-维**连接为:

- Store.StoreID 与 Sales.StoreID 之间的连接
- Time.TimeID 与 Sales.TimeID 之间的连接

- Product.ProductID 与 Sales.ProductID 之间的连接

有几种规则适用于这些连接中的每个连接。您可以仅将信息性约束用于外键约束。

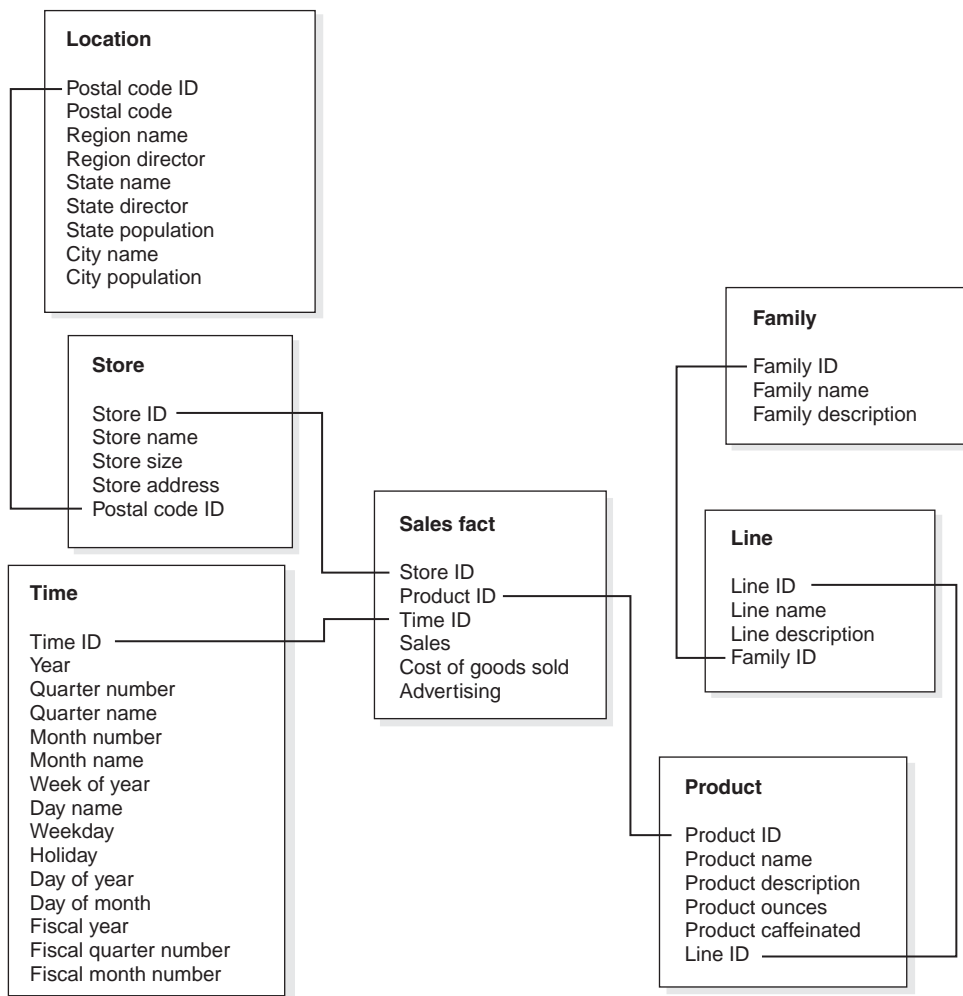


图 35. 雪花模式. CVSAMPLE 数据库中使用雪花模式的关系表。

对于 Store 和 Sales 表之间的连接，必须定义下列约束：

- StoreID 是 Store 表中的主键。
- Store.StoreID 和 Sales.StoreID 都是不可空的列。
- Sales.StoreID 是引用 Store.StoreID 的外键。可以将外键约束定义为信息性约束。
- 如果 Sales.StoreID 不是 Sales 表的主键，则连接基数为 1:多 (Store.StoreID : Sales.StoreID)。如果 Sales.StoreID 是 Sales 表的主键，则连接基数为 1:1。
- 连接类型为 INNER JOIN。

对于 Time 与 Sales 表之间的连接，必须定义下列约束：

- TimeID 是 Time 表中的主键。
- Time.TimeID 和 Sales.TimeID 均为不可空的列。
- Sales.TimeID 是引用 Time.TimeID 的外键。可以将外键约束定义为信息性约束。

- 如果 Sales.TimeID 不是 Sales 表的主键，连接基数为 1:多 (Time.TimeID : Sales.TimeID)。如果 Sales.TimeID 是 Sales 表的主键，则连接基数为 1:1。
- 连接类型为 INNER JOIN。

对于 Product 与 Sales 表之间的连接，必须定义下列约束：

- ProductID 是 Product 表中的主键。
- Product.ProductID 和 Sales.ProductID 均为不可空的列。
- Sales.ProductID 是引用 Product.ProductID 的外键。可以将外键约束定义为信息性约束。
- 如果 Sales.ProductID 不是 Sales 表的主键，则连接基数为 1:多 (Product.ProductID : Sales.ProductID)。如果 Sales.ProductID 是 Sales 表的主键，则连接基数为 1:1。
- 连接类型为 INNER JOIN。

在雪花模式中，每个维具有一个主维表，可以将一个或多个附加维连接至此主维表。主维表是唯一可以连接至事实表的表。直接连接至主表的每个支架表必须具有多:1（其中“多”位于主表一侧）或 1:1 连接基数。因为这些连接基数规则，主维表通常具有所有维表中信息的最详细级别。如果一组维表仅使用 1:1 连接基数，则所有表均具有相同级别的详细信息。

此立方体模型基于雪花模式，所以必须对维表之间的连接定义附加约束。这三个维-到-维连接为：

- Store 表与 Location 表之间的连接
- Product 表与 Line 表之间的连接
- Line 表与 Family 表之间的连接

有几种规则适用于这些连接中的每个连接。您可以仅将信息性约束用于外键约束。

对于 Store 与 Location 表之间的连接，必须定义下列约束：

- PostalcodeID 是 Location 表中的主键。
- Location.PostalCodeID 和 Store.PostalCodeID 都是不可空的列。
- Store.PostalCodeID 是引用 Location.PostalCodeID 的外键。可以将外键约束定义为信息性约束。
- 连接基数为 1:多 (Location.PostalCodeID : Store.PostalCodeID)，原因是 Store.PostalCodeID 不是 Store 表的主键或唯一键。
- 连接类型为 INNER JOIN。

对于 Product 与 Line 表之间的连接，必须定义下列约束：

- LineID 是 Line 表的主键。
- Line.LineID 与 Product.LineID 都是不可空的列。
- Product.LineID 是引用 Line.LineID 的外键。可以将外键约束定义为信息性约束。
- 连接基数为 1:多 (Line.LineID : Product.LineID)，原因是 Product.LineID 不是 Product 表的主键或唯一键。
- 连接类型为 INNER JOIN。

对于 Line 与 Family 表之间的连接，必须定义下列约束：

- FamilyID 是 Family 表的主键。
- Family.FamilyID 和 Line.FamilyID 都是不可空的列。

- Line.FamilyID 是引用 Family.FamilyID 的外键。可以将外键约束定义为信息性约束。
- 连接基数为 1:多 (Family.FamilyID : Line.FamilyID)，原因是 Line.FamilyID 不是 Line 表的主键或唯一键。
- 连接类型为 INNER JOIN。

图 36 显示雪花模式维中的一组有效维表。主维表是 Customer 表，具有三个附加支架表，包括直接连接至 Customer 的 City 和 CustomerGroup 以及连接至 City 的 CityInfo。连接基数在语义上是有效的，原因是个城市或客户组可能有多个客户，而每个城市只有一组城市信息。这是一个用于优化的有效维，因为它符合优化验证规则。维仅有一个主表，直接连接至主表的 City 和 CustomerGroup 表通过多:1 基数连接。CityInfo 表通过 1:1 基数（它也是有效的）连接。Customer 表具有四个维表中最详细的信息级别。

### 可以进行优化的维

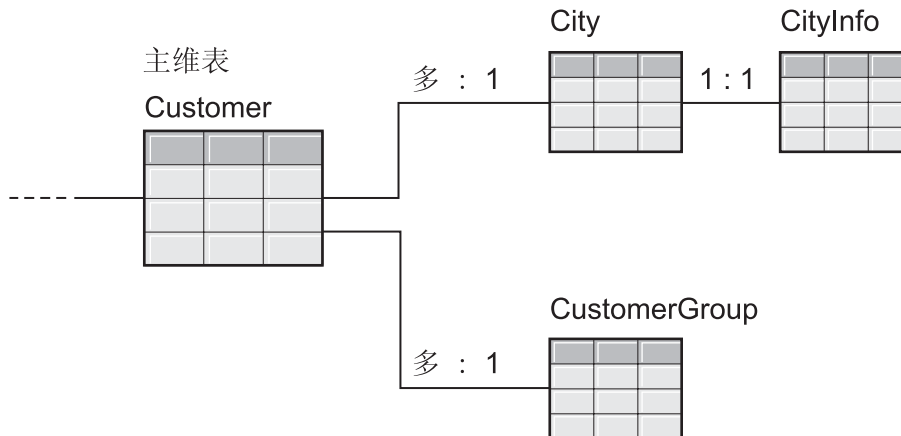


图 36. 有效维. 可以优化的一个维中使用的一组维表

第 97 页的图 37 显示雪花模式维中一组无效的维表。因为定义的基数关系，任何这些表均不可能成为将优化的立方体模型中的主维表。虽然基数在语义上是有效的，但是如果任何这些表作为主维表与事实表连接，事实表中的数据将加倍，从而导致所谓的扇形陷阱。

例如，如果 Customer 是主维表，则 Region 与 SalesRep 之间的 1:多连接基数使维的优化无效。如果每个区域具有五名销售代表，则当连接 SalesRep 和 Region 表时，每个区域即会有五个条目。当这些表与 City 和 Customer 表连接，并最终与事实表连接时，即会为 City、Customer 和事实表中的每个现有行另外添加五行。重复相同事实行五次会导致量度计算错误。维中的其它每个表均具有类似问题。因为 City 与 Customer 之间以及 Region 与 SalesRep 之间的 1:多连接，City 表不能为主维表。Region 表不能与事实表连接，因为维中的每个连接均为与 Region 表的 1:多连接。因为 Region 与 City 表之间以及 City 与 Customer 表之间的 1:多连接，SalesRep 表不能为主维表。



## 无法进行优化的维

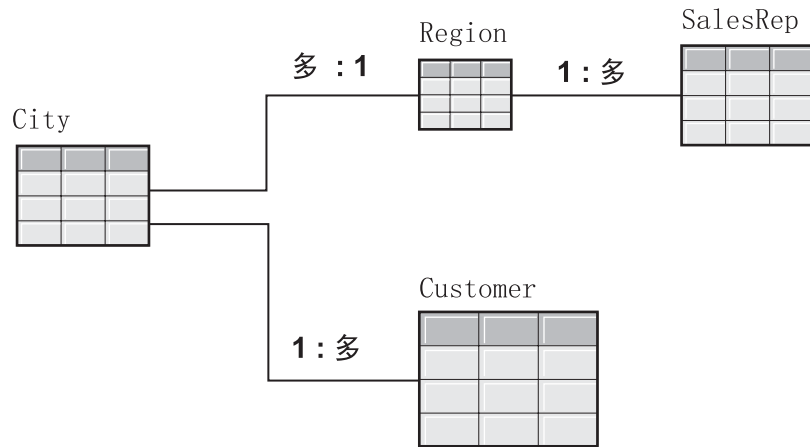


图 37. 无效维. 无法优化的一个维中使用的一组维表

## 优化顾问程序的参数

为优化顾问程序向导提供的有关每个参数的信息将影响向导建议的摘要表以及获得的性能改进。务必提供准确信息，并在成本和性能需求之间作出小心谨慎的决定。

### 期望的查询活动

指定要为其优化立方体模型中的每个立方体的查询的类型。查询类型描述通常在何时以及如何访问 DB2 关系数据。此信息帮助优化顾问程序了解最经常查询的立方体模型部分。如果知道有些特定片查询得极为频繁，还可对立方体指定优化片。

在创建立方体时对每个立方体定义查询类型。可使用优化顾问程序来复查您对每个立方体指定的内容并在必要时做出更改。

### 磁盘空间局限性

指定可以用于摘要表的近似磁盘空间量。优化顾问程序在摘要表构建之前无法知道它的确切大小，因此建议摘要表尽可能接近指定的磁盘空间量。构建的摘要表使用的空间可能多于或少于您指定的空间。

指定的磁盘空间量直接与优化结果相关。增加磁盘空间可以增加查询数（并且性能得到改进）和增大改进程度。在选择磁盘空间量时应考虑下列因素：

- 所需的查询性能级别
- 要对其进行优化的立方体模型数
- 每个立方体模型的重要程度
- 每个立方体模型的使用频率
- 磁盘空间的可用性及成本

一般来说，通过允许使用中等数量的磁盘空间，例如允许立方体模型引用的关系表当前使用 1% 至 10% 的空间，可以得到显著改进。表 37 显示用于摘要表的磁盘空间量与预期查询性能改进之间的关系。当决定提供的空间量时，考虑所有元数据和基本表上下文中的每个立方体模型。

表 37. 磁盘空间. 所用磁盘空间的百分比与相应的预期性能改进

用于摘要表的基本表磁盘空间百分比	相关查询的预期改进
小于 1%	低
5%	中
50%	高
无限	最高

### 时间局限性

您指定的时间是优化顾问程序可以用来确定建议的最大时间量。允许优化顾问程序运行的时间越多，结果越好。下表提供一些确定应向优化顾问程序提供的时间量的近似准则。性能结果会有所变化，您可能需要允许比表 38 中指定的更多时间。

表 38. 时间局限性. 确定为优化顾问程序提供的时间量的准则

数据库优化方案	近似时间限制
未执行数据采样	5 至 30 分钟
对小于 10 GB 大小的小型数据库执行数据采样	1 小时或更少
对大于 10 GB 大小的大型数据库执行数据采样	几个小时

### 数据采样

数据采样是优化顾问程序检查立方体模型中数据的一种方法。这为优化顾问程序提供更多信息，以便其可以创建最有效的建议集合。通过数据采样创建的将建议将与指定的磁盘空间更加准确地匹配。在不进行数据采样的情况下，优化顾问程序将仅分析元数据和 DB2 统计信息来确定建议。

## 优化立方体模型

通过优化对立方体模型执行的查询，您可以提高发出 OLAP 式的 SQL 查询的产品的性能。

**先决条件:** 必须为在立方体模型中使用的基本表指定 DB2 约束。必须在每个事实表和维表之间以及雪花模式中的每个维表之间指定约束。必须在非可空列上指定约束。有关定义约束的更多信息，请参阅第 93 页的『用于优化的约束定义』。

优化立方体模型时，优化顾问程序向导创建可以为立方体模型构建一组建议的摘要表的 SQL。摘要表聚集通常访问的数据以加速查询性能。

要优化立方体模型:

1. 通过右键单击 OLAP 中心对象树中的立方体模型，然后单击**优化顾问程序**来打开优化顾问程序向导。

2. 在“查询类型”页上，查看要对其优化每个立方体的查询类型。可以修改查询类型或对立方体指定优化片。立方体查询类型用于改进优化结果。有关优化片的信息，请参阅第 84 页的『立方体的优化片』。
3. 在“摘要表”页上，指定要立即还是延迟更新摘要表。有关更新选项的信息，请参阅第 104 页的『摘要表维护』。指定存储摘要表和摘要表索引的表空间。
4. 在“局限性”页上，指定将要构建的摘要表和索引可以使用的磁盘空间大小。指定是否允许数据采样。同时应指定允许优化顾问程序用于确定建议的最大时间量。指定的空间、信息和时间越多，性能结果的提高越显著。有关指定优化顾问程序向导的参数信息，请参阅第 97 页的『优化顾问程序的参数』。

这是优化顾问程序向导中的输入参数的最后一页。单击**下一步**以打开进度窗口，以便您能监视（必要的时候停止）优化顾问程序确定用于创建和刷新摘要表的建议的进度。可以修改对优化顾问程序的时间局限性。如果单击**停止**，优化顾问程序将返回在给定时间内确定的摘要表建议。

5. 在“SQL 脚本”页上，分别在**用来创建摘要表的 SQL 脚本**字段和**用来刷新摘要表的 SQL 脚本**字段中输入唯一的文件名。
6. 单击**完成**将建议的 SQL 脚本保存到指定文件名的文件中。
7. 运行 SQL 脚本。如果正在创建大型摘要表，则构建摘要表可能需要大量时间才能完成。可以使用“DB2 命令中心”或“命令窗口”来运行 SQL 脚本。要从“DB2 命令窗口”运行 SQL 脚本：
  - a. 切换至保存 SQL 脚本的目录。
  - b. 连接至已优化的立方体模型的数据库。例如，输入：db2 connect to CVSAMPLE.
  - c. 输入以下命令：

```
db2 -tvf filename
```

其中 *filename* 是创建摘要表 SQL 脚本的名称。

---

## 用于创建摘要表的 SQL 脚本的示例

优化顾问程序向导提供 SQL 脚本来创建建议的摘要表。SQL 脚本包含构建一个或多个摘要表所需的 SQL 命令。

第 100 页的图 38 显示创建一个摘要表的样本 SQL 脚本的部分以及元数据对象如何映射至 SQL。在样本 SQL 脚本中，摘要表被称为 DB2INFO.MQT0000000021T01，其中 21 是立方体模型标识，T01 是摘要表标识。立方体模型标识最长可达 10 位数。摘要表标识在立方体模型中标识摘要表。一个立方体模型中的摘要表标识允许最多达 99 个摘要表。不要更改优化顾问程序向导定义的摘要表名称。如果更改表名，则 DB2 Cube Views 无法标识它为立方体模型创建的摘要表。

```

DROP TABLE DB2INFO.MQT0000000021T01;
UPDATE COMMAND OPTIONS USING c OFF;
CREATE SUMMARY TABLE DB2INFO.MQT0000000021T01 AS
(
SELECT
SUM(T1."SALES") AS "SALES",
SUM(T1."COGS") AS "COGS",
SUM(T1."ADVERTISING") AS "ADVERTISING",
SUM(T1."COGS" + T1."ADVERTISING") AS "Total expense",
SUM(T1."SALES" - (T1."COGS" + T1."ADVERTISING")) AS "Profit",

T5."REGION_NAME" AS "REGION_NAME",
T5."STATE_NAME" AS "STATE_NAME",
T6."LINEID" AS "LINEID (LINE)",
T4."YEAR" AS "YEAR",
T4."QUARTER_NUMBER" AS "QUARTER_NUMBER",
Qtr ' CONCAT (cast (T4."QUARTER_NUMBER" AS CHAR(1))) AS "QUARTER_NAME",
T4."MONTH_NUMBER" AS "MONTH_NUMBER"

FROM "CVSAMPLE"."SALESFACT" AS T1,
"CVSAMPLE"."STORE" AS T2,
"CVSAMPLE"."PRODUCT" AS T3,
"CVSAMPLE"."TIME" AS T4,
"CVSAMPLE"."LOCATION" AS T5,
"CVSAMPLE"."LINE" AS T6

WHERE
T1."STOREID"=T2."STOREID" AND T1."PRODUCTID"=T3."PRODUCTID"
AND T1."TIMEID"=T4."TIMEID" AND T2."POSTALCODEID"=T5."POSTALCODEID"
AND T3."LINEID"=T6."LINEID"

GROUP BY
T5."REGION_NAME",
T5."STATE_NAME",
T6."LINEID",
T4."YEAR",
T4."QUARTER_NUMBER",
Qtr ' CONCAT (cast (T4."QUARTER_NUMBER" AS CHAR(1))),
T4."MONTH_NUMBER")

DATA INITIALLY DEFERRED
REFRESH DEFERRED
ENABLE QUERY OPTIMIZATION
MAINTAINED BY SYSTEM
NOT LOGGED INITIALLY;

COMMENT ON TABLE DB2INFO.MQT0000000021T01 IS 'AST created for cube model CVSAMPLE.Sales Model';
COMMIT;

ALTER TABLE DB2INFO.MQT0000000021T01 ACTIVATE NOT LOGGED INITIALLY;

REFRESH TABLE DB2INFO.MQT0000000021T01;

CREATE INDEX DB2INFO.IDX0000000021T0101 ON DB2INFO.MQT0000000021T01("STATE_NAME");
CREATE INDEX DB2INFO.IDX0000000021T0102 ON DB2INFO.MQT0000000021T01("MONTH_NUMBER");
CREATE INDEX DB2INFO.IDX0000000021T0103 ON DB2INFO.MQT0000000021T01("LINEID (LINE)") CLUSTER;

COMMIT;

REORG TABLE DB2INFO.MQT0000000021T01;
RUNSTATS ON TABLE DB2INFO.MQT0000000021T01 AND INDEXES ALL;
COMMIT;

```

映射到量度

映射到属性

事实和维表

连接事实表和维表。

如果适用的话，在雪花模式中将维连接到维表

映射到立方体模型中的片段

创建索引

图 38. 样本 SQL 脚本

如果对立立方体模型建议多个摘要表，对于每个摘要表，用来创建摘要表的 SQL 脚本将包括以下一组语句。

以下各节说明创建摘要表 SQL 脚本样本中的语句：

### DROP TABLE 语句

首先删除将要创建的每个摘要表，以确保不存在同名的表。在图 38 中，通过下列语句删除 DB2INFO.MQT0000000021T01 表：DROP TABLE DB2INFO.MQT0000000021T01；。

### CREATE TABLE 语句

脚本使用 CREATE TABLE 语句创建摘要表。此语句是脚本中最大的部分，包括带有 SELECT、FROM、WHERE 和 GROUP BY 子句的 SELECT 语句以及更新方法定义。摘要表是使用相应列（但没有任何数据）创建的。刷新摘要表的 SQL 脚本从事实表和维表中聚集数据并填充摘要表。

在 CREATE TABLE 语句的第一行定义表名：CREATE SUMMARY TABLE DB2INFO.MQT0000000021T01。

第 100 页的图 38 中显示的 SELECT 子句有 5 行以 SUM 开头。每个这些行映射至立方体模型的其中一个量度。例如，SUM(T1."SALES"-(T1."COGS"+T1."ADVERTISING')) AS "Profit" 使用聚集函数 SUM 映射至 Profit 计算的量度。正在为其创建摘要表的立方体模型具有下列量度：Sales、COGS、Advertising、Total expense 和 Profit。在不执行任何计算的情况下选择一列的下面七行映射至属性。例如，T5."REGION\_NAME" AS "REGION\_NAME" 映射至区域名属性。摘要表包括来自立方体模型的下列属性：Region name、State name、Line ID、Year、Quarter number、Quarter name 和 Month name。

FROM 子句中的表是立方体模型中使用的事实表和维表。此示例使用 SalesFact、Store、Product、Time、Location 和 Line 表。

WHERE 子句定义事实表和维表之间的连接，每个连接映射至立方体模型中的连接对象。要优化的立方体模型基于雪花模式，所以维-到-维连接也会包括在 WHERE 子句中。

GROUP BY 子句映射至为立方体模型定义的片。第 100 页的图 38 显示映射至特定片的一个分组集合。分组可以包括下列元数据类型来定义片：

- 片级别层次结构中的层键属性
- 片级别之上的层键属性
- 在功能上与层键属性无关的相关属性

这部分 SQL 脚本可能包含 GROUPING SETS，所以摘要表可能包含多个级别。如果立方体模型包含非描述性量度，则这部分 SQL 脚本可能包含 ROLLUP。

在显示的示例中，要优化的立方体模型具有下列层次结构：Market [Region, State, City, Postal code, Store]、Product [Family, Line, Product]、Time [Year, Quarter, Month, Day] 和 Fiscal Time [Fiscal year, Fiscal quarter, Fiscal month]。如果层次结构中的层未包括在分组集合中，则片处于最高层，例如 All Time、All Regions 或 All Products。GROUP BY 子句中的片是 State-Line-Month 片，并且包括 Region name、Year、Quarter number 和 Quarter name 属性。Region name 在 State 层之上，而 Year、Quarter number 和 Quarter name 在 Month 层之上。片中的每个层属性均是 SELECT 子句所映射的属性。

CREATE TABLE 语句的最后部分是更新方法定义。在第 100 页的图 38 中，CREATE TABLE 语句的最后三行将摘要表设置为延迟刷新：

```
DATA INITIALLY DEFERRED
REFRESH DEFERRED
ENABLE QUERY OPTIMIZATION
MAINTAINED BY SYSTEM
NOT LOGGED INITIALLY;
```

如果将摘要表定义为立即刷新，则语句将为：

```
DATA INITIALLY DEFERRED
REFRESH IMMEDIATE
ENABLE QUERY OPTIMIZATION
MAINTAINED BY SYSTEM
NOT LOGGED INITIALLY;
```

## CREATE INDEX 语句

优化顾问程序向导为摘要表建议一个或多个索引，创建摘要表之后将会创建索引。在第 100 页的图 38 中，创建群集和非群集索引。创建索引之后，REORG 语句可用于根据群集索引对表进行重组。在某些情况下，这可以改进对表的读取性能。

## RUNSTATS 语句

创建摘要表所有建议的特征之后，RUNSTATS 语句更新 DB2 优化器统计信息，DB2 优化器使用这些统计信息来考虑用于查询重新路由的摘要表和索引。

---

## 测试查询结果

使用优化顾问程序创建摘要表之前和之后，您可以使用 DB2 通用数据库中的 db2batch 基准工具来为查询性能结果制定基准。

要测试查询的性能：

1. 创建包含要测试的、由分号分隔的查询的输入文件。
2. 在命令行上输入以下命令：

```
db2batch -d dbname -f file_name -cli
```

其中，*dbname* 指定要对其运行查询的数据库，*file\_name* 指定包含 SQL 查询的输入文件，*-cli* 指定以 CLI 方式运行。db2batch 工具总结性能结果并提供算术平均和几何平均。对于语法和选项，在命令行上输入 db2batch -h。有关 db2batch 基准工具和创建基准测试的更多信息，请参阅“DB2 信息中心”。

创建建议的摘要表之后，如果对性能结果满意，则不需要执行任何额外的性能分析。

如果查询未能如您所期望的那样得到改善，则可以再次运行优化顾问程序向导并允许更多磁盘空间和时间，以及启用数据采样（如果在此前未启用的话）。允许更多磁盘空间极有可能对性能产生最大影响。为摘要表提供的空间越多，性能改进程度就越高。如果允许向导执行数据采样，向导可以提供更好的建议。同样，允许向导创建建议的时间越多，建议可能就越好。

如果由于查询没有任何提高或仅有一点提高，您对性能结果不满意，或者查询在一段时间内执行良好，而后性能下降，请参阅『对摘要表进行故障诊断』。

---

## 对摘要表进行故障诊断

如果在创建摘要表之后查询性能没有改进，可使用 DB2EXPLAIN 设施来对查询重路由进行故障诊断。

在使用 DB2EXPLAIN 来验证 DB2 UDB 是否正在使用摘要表之前，应该：

- 验证基本表和摘要表上的统计信息是否是最新的。
- 标识不能执行的查询（如果尚不清楚的话）。可以使用 DB2 SQL 快照监视器来捕获慢速查询。

要确定查询性能不能达到如期改进的原因：

1. 创建说明表。要为数据库设置说明表，连接至数据库并从 \SQLLIB\misc 目录运行以下命令：

```
db2 -tvf explain.dd1
```

2. 运行说明程序。当启用说明方式时，SQL 查询不会运行，仅处理说明命令的信息请求。运行下列 SQL 命令系列来打开说明方式，设置刷新期限，以便在摘要表被延迟刷新以及运行查询、关闭说明方式和查询说明表以查看是否已重新路由由查询时，DB2 UDB 考虑摘要表：

```

set current explain mode explain

set current refresh age any

SELECT SUM(SALES) FROM MDSAMPLE.SALESFACT

set current explain mode no

SELECT EXPLAIN_TIME, EXPLAIN_LEVEL AS "LEV",
       QUERYNO, STATEMENT_TEXT
FROM EXPLAIN_STATEMENT
WHERE STATEMENT_TEXT LIKE '%SALESFACT%'
ORDER BY EXPLAIN_TIME

```

3. 查看说明信息并检查重写的查询是否重新路由至摘要表。例如，您可能会看到与下列样本相似的报表：

```

2002-06-30-23.22.12.325002 O 11 SELECT SUM(SALES)
      FROM MDSAMPLE.SALESFACT
2002-06-30-23.22.12.325002 P 11 SELECT Q3.$C0
      FROM (SELECT SUM(Q2.$C0) FROM (SELECT Q1.SALESFACT_SALES
      FROM DB2INFO.MQT0000000021T01 AS Q1) AS Q2) AS Q3

```

查询的一次执行占用两行。标有 O 的行是发送给 DB2 UDB 的原始查询。标有 P 的行是由 DB2 优化器重写的查询。可从此示例的重写查询中看到 DB2 优化器从 DB2INFO.MQT0000000021T01 摘要表中选择数据。

如果查询重新路由至摘要表但是性能并未有如期改善，则可能需要通过使用不同的选项再次运行优化顾问程序向导。

如果查询未重新路由至摘要表，则确定原因并采取相应的措施。未将查询重新路由至摘要表的原因包括：

#### 摘要表不存在

首先，确保摘要表存在。如果它不存在，则运行优化顾问程序向导来生成创建摘要表 SQL 脚本。然后运行脚本来创建摘要表。

#### 延迟刷新摘要表已过期

如果摘要表存在并且将其设置为延迟刷新，则可能需要更新刷新期限。通过设置 (DFT\_REFRESH\_AGE) = 999999999999 (ANY) 可以将表的刷新期限设置得尽可能大且独立于会话。

#### 查询访问摘要表中未包括的数据

如果查询正在访问的数据未在摘要表中，DB2 优化器不会重新路由查询。如果在创建摘要表之后添加一个新的量度，则新的量度不会存在于摘要表中。如果试图查询新的量度，DB2 优化器无法将查询重新路由至摘要表，原因是摘要表未包含符合查询的所有数据。

另外，如果试图查询立方体模型片段（为此片段构建摘要表）之下的数据，则不能使用摘要表。例如，如果查询在请求 City 层聚集的数据，而摘要表包括在 State 层（在 City 层之上）聚集的数据，则查询不能使用该摘要表。

#### 查询包含不能重新路由的构造

DB2 优化器无法重新路由使用某些复杂查询构造的查询。某些禁止 DB2 优化器重新路由查询的复杂构造是与以下内容相似的递归和物理属性函数：

- NODENUMBER

- 外连接
- 并集
- XMLAGG
- Window 聚集函数，此函数是通过 OVER 子句指定的聚集函数

---

## 摘要表维护

当基本表中的数据更改时，您需要更新摘要表。您可以通过两种不同方式更新摘要表：立即刷新或延迟刷新。

在运行优化顾问程序向导时，请选择创建立即刷新摘要表还是延迟刷新摘要表。所做的选择会影响表和刷新摘要表 SQL 脚本的更新设置。对于这两种选项，您需要将刷新摘要表脚本作为正常数据库维护调度表的一部分运行。运行刷新脚本可能需要大量时间和处理资源。确保在维护批处理窗口中分配足够的时间以完成更新。

### 立即刷新

立即刷新摘要表与基本表紧密保持同步。DB2 UDB 跟踪基本表的更改，因此它可以通过仅更改与基本表更改的部分对应的摘要表部分来以增量形式更新摘要表。如果摘要表数据与基本表保持一致非常重要，则使用立即刷新选项。例如，如果基本表使用周销售数据更新并且用户填写反映更新销售数据的周报表，则立即刷新可能是个不错的选择。

如果经常在整个基本表的不同地方进行大量更改，则立即刷新可能并不是最佳选择，因为它可能需要 DB2 UDB 的大量开销来跟踪更改并单独执行更新语句以再次聚集更改。

如果使用常规 SQL 语句（如 INSERT、UPDATE 和 DELETE）来更新基本表，在更改基本表之后，DB2 UDB 将自动同步受影响的摘要表。但是，如果使用 DB2 LOAD 或 IMPORT 命令更新基本表，则需要在完成更新之后通过运行刷新脚本来手工触发同步。

立即更新并非在所有情况下均可用，优化顾问程序向导可能建议使用延迟选项（如有必要）。

### 延迟刷新

因为必须手工同步摘要表和基本表，延迟刷新摘要表的更新频率通常比立即刷新的低。摘要表以创建时的数据快照为基础。每次更新将依据当前数据重新创建摘要表，但并不知道自上次创建摘要表以来数据发生了怎样的更改。

在对整个相应的基本表进行较大更改时，或者如果更新数据的速度比需要访问数据的速度更快，延迟刷新是个不错的选择。例如，如果销售数据每周更新，但是您只需要按季度创建报表，则可以使用延迟刷新选项，并在每个季度运行报表之前重新构建摘要表。

---

## 删除摘要表

当删除立方体模型时，DB2 Cube Views 不会删除关联的摘要表。如果未将摘要表用于任何其它用途，则可以删除这些表以释放磁盘空间。

摘要表是一类表，可以在控制中心或命令行中使用通常的 DB2 过程来删除。任何关联的索引也随摘要表一同删除。



摘要表在 DB2INFO 模式中定义。摘要表名包括立方体模型标识。例如，可以将摘要表命名为 DB2INFO.MQT000000021T01，其中 21 是立方体模型标识，T01 唯一地标识立方体模型内的摘要表。立方体模型标识最长可达 10 位数。

要从命令行中删除摘要表，输入 `DROP TABLE table_name`。



---

## 第 6 章 DB2 Cube Views 和联合数据源

本节描述下列主题:

### 联合系统概述

可使用 IBM DB2 Information Integrator 作为信息集成的企业级解决方案。DB2 Information Integrator 是若干技术的集合，它将数据管理系统、联合系统以及几种其它技术组合成一个公共平台。

### 使用 DB2 Cube Views 优化远程数据源概述

可使用 DB2 Cube Views 优化联合星型模式或雪花模式以在查询性能方面得到显著的提高。

### 对 DB2 Cube Views 启用联合系统

要对 DB2 Cube Views 启用联合系统，必须定义远程数据源、对联合服务器定义昵称并对昵称定义信息性约束。

### 联合数据源的查询重新路由故障诊断

如果在进行优化之后向远程数据源发出的查询的性能并没有预期的改善，确保对于 DB2 Cube Views 该联合系统的设置是正确的。

---

## 联合系统概述

可使用 IBM DB2 Information Integrator 作为信息集成的企业级解决方案。DB2 Information Integrator 是若干技术的集合，它将数据管理系统、联合系统以及几种其它技术组合成一个公共平台。

DB2 联合系统是特殊类型的分布式数据库管理系统 (DBMS)。可以使用信息集成技术的联合系统方面来存取跨各种数据源分布的不同数据类型。联合系统由下列组件组成:

- 充当联合服务器的 DB2 实例
- 充当联合数据库的数据库
- 一个或多个数据源
- 存取数据库和数据源的客户机 (用户和应用程序)

借助联合系统，可以在单个 SQL 语句中将分布式请求发送至多个数据源。例如，可以在单个 SQL 语句中连接位于 DB2 通用数据库表、Oracle 表和 XML 标记文件中的数据。

### 联合服务器

联合系统中的 DB2 服务器称为联合服务器。可以配置任意数目的 DB2 实例来充当联合服务器。可以将现有 DB2 实例用作联合服务器，也可以专门为联合系统创建新的联合服务器。

管理联合系统的 DB2 实例称为服务器，因为它会响应来自最终用户和客户机应用程序的请求。联合服务器经常将它接收到的部分请求发送至数据源以便处理。下推操作是远程处理的操作。管理联合系统的 DB2 实例称为联合服务器，即使当它将请求下推至数据源时充当客户机也是如此。

联合服务器通过包装器与数据源交互。联合服务器使用存储在称为包装程序模块的库中的例程来实现包装器。这些例程允许联合服务器执行一些操作（例如，连接至数据源并重复地从数据源中检索数据）。通常，DB2 联合实例所有者使用 CREATE WRAPPER 语句在联合数据库中注册包装器。

### 联合数据库

对于最终用户和客户机应用程序，数据源作为单个集体的 DB2 UDB 数据库出现。用户和应用程序与联合服务器管理的联合数据库进行交互。联合数据库包含系统目录。联合数据库系统目录包含标识数据源及其特征的条目。联合服务器会参考存储在联合数据库系统目录和数据源包装器中的信息来确定处理 SQL 语句的最佳方案。

### 数据源

在联合系统中，数据源可以是关系 DBMS 实例（例如，DB2、Informix、Oracle 或 Sybase）或非关系数据源（例如，BLAST 搜索算法或 XML 标记文件）。

用来存取数据源的方法或协议取决于数据源的类型。例如，DRDA 用来存取 DB2 系列数据源（例如，DB2 z/OS 和 OS/390 版）。

**客户机** 客户机可以包括存取您安装的联合数据库及数据源的用户和应用程序。DB2 Cube Views 是可充当存取联合数据库和数据源的客户机的应用程序的示例。

---

## 使用 DB2 Cube Views 优化远程数据源概述

可使用 DB2 Cube Views 优化联合星型模式或雪花模式以在查询性能方面得到显著的提高。

优点包括：

- 具有集成 DB2 平台，可从中访问多个 IBM 和供应商产品
- 将 DB2 UDB Linux 版、UNIX 版、Windows 版的丰富功能扩展至 DB2 UDB z/OS 版和 DB2 UDB iSeries 版
- 改进对联合数据库和数据源的查询的性能

在联合系统中，事实表和维表可能在某个远程服务器上或在若干远程服务器中。必须在本地表示 DB2 Cube Views 的星型模式或雪花模式。可在本地使用引用远程表和远程表的复制副本的昵称或昵称与复制表的组合来表示星型模式或雪花模式。必须以本地方式在本地服务器上用某种方法表示星型模式或雪花模式中的所有事实表和维表。

在星型模式或雪花模式存在于本地服务器上之后，可以使用 DB2 Cube Views 来构建基于该星型模式或雪花模式的立方体模型并使用优化顾问程序来优化该立方体模型。如果在本地使用昵称表示其中一些表，则指定优化顾问程序向导可以使用数据采样会提供更好的摘要表建议，但需要更多时间来创建这些建议。优化顾问程序可能不能在每种情况下都使用数据采样，但是建议尽可能允许数据采样。

可以在本地服务器上创建建议的摘要表，并且可以使用下列三个方法中的其中一个来响应针对远程星型模式或雪花模式中的表的查询：

### 查询路由至本地摘要表

如果该查询可通过摘要表中的数据响应，则 DB2 优化器会将该查询直接路由至本地摘要表，于是根本不需要与远程表交互。此路由方案最大限度地增强了性能。

通常，查询不能仅通过摘要表响应，但可通过将摘要表与一个或多个维表相连接来响应。在此情况下，您可能考虑在联合服务器上组合一些维表或所有维表（而不是事实表）以便 DB2 优化器能够完成本地表上的连接。

#### 将查询下推至远程数据源

如果该查询不能通过摘要表及任何组合的维表中的数据来响应，则 DB2 优化器将尝试将查询下推至远程服务器。远程服务器会完成该查询并将结果集返回至本地服务器。

如果结果集很小，则性能仍然可以得到显著增强。如果远程星型模式或雪花模式中的所有表只存在于一个远程服务器上，则此方法是可行的。

#### 在本地对从远程数据源拉出的数据运行查询

如果该查询不能通过摘要表及任何组合的维表中的数据响应且不能下推至远程数据源，则满足该查询所需的那些表将被拉出并从远程服务器复制至本地服务器，于是查询将在本地完成。例如，如果星型模式位于远程 zSeries 服务器上且查询使用 DB2 UDB 中提供的但不在 DB2 z/OS 版中的函数，则该查询不能在远程服务器上完成。

如果响应查询所需的事实表和维表很大，则此路由方案可能不会增强性能。

---

## 对 DB2 Cube Views 启用联合系统

要对 DB2 Cube Views 启用联合系统，必须定义远程数据源、对联合服务器定义昵称并对昵称定义信息性约束。

**Prerequisites:** 确保启用了 DB2 Information Integrator 支持。要从命令行启用联合系统支持，输入：`db2 update dbm cfg using federated yes`。在启用联合系统支持后停止并启动 DB2 UDB。

要对 DB2 Cube Views 启用远程数据源：

1. 定义远程数据源。
2. 定义远程表的昵称。
3. 定义昵称的信息性约束。

### 定义远程数据源

要对 DB2 Cube Views 定义和配置联合服务器，必须向联合服务器提供有关想要存取的远程数据源和对象的信息。

要定义远程数据源：

1. 在联合服务器数据库目录中对远程数据库和相应节点进行编目以使联合服务器知道要连接至的远程数据源。可使用配置助手中的添加数据库向导来对远程数据库和相应节点进行编目。
2. 在联合服务器上连接至本地数据库。如果还没有本地数据库，则创建本地数据库然后连接至该数据库。
3. 通过发出 `CREATE WRAPPER` 命令来注册包装器。例如，如果远程数据源来自 DB2 系列（例如，z/OS），则发出以下命令：

```
CREATE WRAPPER drda LIBRARY 'libdb2drda.a'
```

必须知道数据源所需的库。

4. 对想要存取的远程数据源使用的每个服务器注册服务器定义。例如，为 DB2 系列服务器创建 DRDA 包装器。发出以下命令以注册远程 z/OS 服务器：

```
CREATE SERVER server_name
  TYPE DB2/ZOS
  VERSION 8.1
  WRAPPER DRDA
  AUTHORIZATION "userid" PASSWORD "password"
  OPTIONS (DBNAME 'database_name')
```

**提示：**可按 SQL 脚本文件中的描述运行此命令。要直接从 DB2 命令窗口发出此命令，输入命令时用双引号将命令引起来，并使用转义双引号（\）将您提供的用户标识和密码引起来：

```
DB2 "CREATE SERVER server_name
  TYPE DB2/ZOS
  VERSION 8.1
  WRAPPER drda
  AUTHORIZATION \"userid\" PASSWORD \"password\"
  OPTIONS (DBNAME 'database_name')"
```

5. 对于定义的每个服务器，将查询工作负载设置为总是从联合服务器下推至远程数据源。远程数据源上的数据量可能很大，您不希望联合服务器上的 DB2 UDB 尝试将这些数据从远程数据源复制至联合服务器。通过将服务器设置为尽量下推，DB2 UDB 将总是尝试先将查询下推至远程数据源以使只有查询结果集被复制至联合服务器。如果想要通过 DB2 Cube Views 优化得到性能改进，则需要此设置。例如：

```
CREATE SERVER OPTION DB2_MAXIMAL_PUSHDOWN
  FOR SERVER server_name
  SETTING 'Y'
```

6. 创建用户映射以使包装器可自动连接至服务器。例如：

```
CREATE USER MAPPING FOR USER
  SERVER server_name
  OPTIONS(REMOTE_AUTHID 'user_name', REMOTE_PASSWORD 'password')
```

## 对 DB2 Cube Views 的远程表定义昵称

对要在联合服务器上创建的星型模式或雪花模式中的每个远程表定义昵称。DB2 Cube Views 需要昵称来处理远程表。

要对远程数据源上的表定义昵称：

使用 CREATE NICKNAME 语句来对想要存取的每个远程表定义昵称。例如：

```
CREATE NICKNAME local_schema_name.local_nickname
  FOR remote_server_name.remote_schema_name.remote_table_name
```

**提示：**对本地昵称使用在远程服务器上定义的同一种模式名。例如：

```
CREATE NICKNAME CVSAMPLE.FAMILY FOR remote_server_name.CVSAMPLE.FAMILY
```

## 对 DB2 Cube Views 的昵称定义信息性约束

定义信息性约束以说明数据之间的关系可以改进性能。应对联合服务器上处于星型模式或雪花模式的表之间的外键定义信息性约束。

DB2 Cube Views 优化顾问程序向导需要信息性约束以建议用于改进性能的摘要表。DB2 优化器还会使用这些约束以便更有效率地处理查询并适当地将查询路由至现有摘要表。

DB2 Information Integrator 自动在联合服务器上定义与远程数据源上的现有主键约束相匹配的主键约束。必须为用于在联合服务器上构建星型模式或雪花模式的外键创建信息性约束。

要定义外键:

使用 ALTER NICKNAME 语句将信息性约束添加至昵称。 例如:

```
ALTER NICKNAME local_schema_name.local_nickname
ADD FOREIGN KEY(column_name)
REFERENCES local_schema_name.local_nickname (column_name)
ON DELETE RESTRICT
NOT ENFORCED
ENABLE QUERY OPTIMIZATION
```

您已完成对 DB2 Cube Views 启用远程数据源这一任务。

接下来, 使用 DB2 Cube Views 对联合服务器上的星型模式或雪花模式构建完整的立方体模型。在构建完整的立方体模型之后, 可使用优化顾问程序向导来优化该立方体模型。

---

## 远程数据源的查询重新路由故障诊断

如果在进行优化之后向远程数据源发出的查询的性能并没有预期的改善, 确保对于 DB2 Cube Views 该联合系统的设置是正确的。

要想向远程数据源发出的查询得到改善, 必须先完成下列步骤:

1. 对 DB2 Cube Views 启用联合系统。
2. 创建完整的立方体模型, 该模型符合第 33 页的『元数据对象规则』中描述的基本规则、立方体模型完整性规则和优化规则。
3. 优化立方体模型。
4. 如果查询性能没有改善, 检查第 102 页的『对摘要表进行故障诊断』中描述的步骤。

如果完成上述步骤后, 查询的性能仍未改善, 考虑下列问题:

- 确保定义了所有适用的约束。
- 确保将 DB2\_MAXIMAL\_PUSHDOWN 设置按第 109 页的『定义远程数据源』中描述的设置“是”。
- 考虑在联合服务器上合并查询涉及的维表。合并各维以使联合服务器上存在维表的复制副本可以改善性能。





## 第 7 章 DB2 Cube Views API

### DB2 Cube Views API 概述

DB2 Cube Views 提供了应用程序编程接口 (API)，该接口提供对存储在 DB2 Cube Views 中的元数据的计划性存取。使用 API，应用程序可以使用 DB2 Cube Views 的元数据对象与元数据交互，而不必与关系表和连接交互。

DB2 Cube Views API 提供了对存储在 DB2 数据库的系统目录表中的元数据的存取。使用 API 的应用程序可以创建和修改为数据仓库中的多维结构和 OLAP 结构建立模型的元数据对象。

图 39 显示如何通过 API 交换数据和元数据。

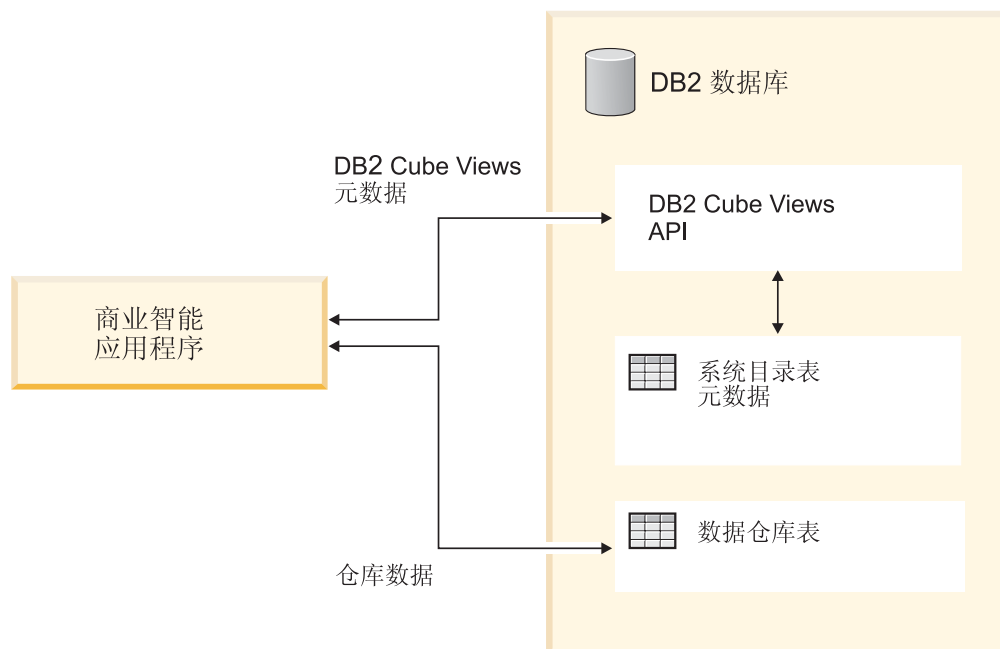


图 39. 通过 DB2 Cube Views API 进行的数据交换

API 是向 DB2 数据库注册的单个存储过程。此存储过程接受用来表达复杂元数据和元数据操作的输入和输出参数。API 参数格式由 XML 模式定义。

API 使用下列技术在 DB2 Cube Views 与商业智能应用程序之间交换元数据:

- 带有 ODBC 的 SQL
- DB2 CLI 和 JDBC
- XML

---

## DB2 Cube Views API: DB2 存储过程和 XML 语法分析

DB2 Cube Views API 通过 DB2 Cube Views 存储过程和 XML 语法分析在商业智能应用程序与 DB2 数据库之间交换元数据。

有关使用 DB2 存储过程编程的信息，请参阅 *DB2 Application Development Guide*。使用 DB2 Cube Views API 编程之前，应了解下列概念：

### 事务

DB2 UDB 支持事务性多用户元数据存取。（有关更多信息，请参阅 *DB2 Application Development Guide*。）使用 DB2 Cube Views API 执行的所有数据库操作都属于调用应用程序的数据库事务。因此，应用程序可以在 API 调用 `md_message` 存储过程之后执行 `COMMIT` 或 `ROLLBACK` 以管理数据库工作单元。

### 内存管理

参数在应用程序与 `md_message` 存储过程之间以 CLOB 结构的形式进行交换。调用 `md_message` 存储过程的应用程序必须预先分配 CLOB 参数结构，该结构的大小与用于对存储过程进行编目的结构相同。对于 CLOB，API 支持 DB2 UDB 最大大小（即 2 GB）。缺省 CLOB 大小为 1 MB。

### 系统配置

要支持大量参数的交换，您可能需要更改下列 DB2 UDB 设置：

- 可能需要使用更大的堆和堆栈大小链接调用 `md_message` 存储过程的数据库客户机应用程序。
- 可能需要使用 `query_heap_sz` 设置增加数据库的 DB2 查询堆大小。

### XML 语法分析

使用 API 的应用程序必须分析由 `md_message` 存储过程返回的输出参数。想要使用该 API 的开发者有多种 XML 解析器可供选择。

### 错误处理

API 以三种格式生成错误信息：

- 由存储过程返回至调用应用程序的 `SQLCODE` 和 `SQLSTATE` 信息
- 通过使用 **response** API 参数传递至调用应用程序的 XML 结构
- 位于运行 API 的数据库服务器上的错误和运行时日志文件

如果由于 XML 验证、语法分析或标记而出错，则 **response** 参数将返回至调用应用程序，并使用 `<error>` 标记代替操作标记。此 `<error>` XML 标记将包含 `<status>` 标记，并带有描述 API 遇到的问题的返回码和消息。

如果在与 XML 处理无关但与元数据操作的执行有关的 API 内出错，则返回 **response** 参数的内容。

以下示例显示 `<error>` 标记中信息的类型。在此示例中，参数结构的描述仅使用有限数目的 XML 标记。大多数参数具有的 XML 标记比此处所显示的要多，并将使用 XML 模式验证参数内容。

```

<olap:response xmlns:olap="http://www.ibm.com/olap"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" version="8.1.2.1.0">
<error>
<status id="3100" text="The system failed to parse XML for
  &quot;INPUT PARAMETER&quot; (line:&quot;3&quot;;
  char:&quot;26&quot;; message:&quot;Unknown element
  'dropa'&quot;)." type="error"/>
</error>
</olap:response>

```

当调用 DB2 Cube Views 存储过程时，无论实际上是否执行了该存储过程，DB2 UDB 都会将 SQLCODE 和 SQLSTATE 返回至调用应用程序。如果 DB2 Cube Views 存储过程能够执行，存储过程将返回一条状态消息作为发送至调用应用程序的 XML 数据的一部分。

## DB2 Cube Views 存储过程

该存储过程称为 `md_message`，它处理以 DB2 Cube Views 参数格式表达的参数。

过程从输入参数中抽取操作和元数据信息，然后执行请求的元数据操作。过程根据操作生成包含所请求操作的执行状态（成功或失败）和元数据信息的输出参数。

该存储过程作为在 UNIX 系统上的 `.fenced` 文件中指定的用户运行。受防护用户必须具有对 `db2md_config.xml` 文件中指定的日志文件的 `WRITE` 访问权。否则，存储过程不能记录任何内容。

DB2 Cube Views 存储过程是作为 DB2 UDB 存储过程实现的。它可由使用任何 DB2 UDB 编程接口的任何应用程序使用。存储过程的名称不区分大小写，而存储过程参数的名称和内容区分大小写。`md_message` 的语法和原型为：

```

语法:      call md_message (request, metadata, response)
原型:      md_message (request IN CLOB(1M),
                                     metadata INOUT CLOB(1M),
                                     response OUT CLOB(1M))

```

**request**、**metadata** 和 **response** 参数是字符大对象（CLOB）类型，是一种 DB2 UDB 数据类型。应用程序使用待执行操作的描述填充 **request** 参数，并可选择使用操作将依据的元数据填充 **metadata** 参数。在它实现输入参数之后，`md_message` 将在 **response** 参数中返回操作的状态，并在 **metadata** 参数中返回请求的元数据。**metadata** 参数用于元数据的输入和输出。DB2 UDB 在数据库服务器上进行商业智能应用程序与 `md_message` 存储过程之间的参数结构的传送。

CLOB 自变量的大小可以变化。缺省大小为 1 MB。CLOB 自变量大小由 `sqllib/misc/db2mdapi.sql` 脚本指定。您可以重新编制具有任意 CLOB 参数大小（最多 2 GB）的存储过程的目录。当增加参数的大小时，存储过程在运行时将使用更多的内存，原因是输出参数缓冲区在启动存储过程时被预分配为编目大小。如果大小太小，输入和输出参数中的数据会被截断。

要更改缺省 CLOB 大小，使用增加了的大小限制重新注册存储过程 API。

1. 编辑 `sqllib/misc/db2mdapi.sql` 文件并增加 `CREATE PROCEDURE` 文件的大小限制。

```

CREATE PROCEDURE
  DB2INFO.MD_MESSAGE(IN request CLOB(1M),
                    INOUT metadata CLOB(10M),
                    OUT response CLOB(10M))

```

2. 重新运行 db2mdapi.sql 文件以重新注册存储过程 API。在脚本尝试创建已经存在的表时忽略所有错误。

有关通过 C++ 调用存储过程 API 的信息，请参阅 `sqllib/samples/olap/client/db2mdapiclient.cpp` 文件中的 C++ 源代码。

以下示例显示如何从嵌入式 SQL 应用程序中调用 DB2 Cube Views 存储过程：

```
// Standard declarations
// ...

// Include the Communication Area to access error details
EXEC SQL INCLUDE SQLCA;

// SQL declarations of host variables that will be used for calling the
// DB2 Cube Views stored procedure
EXEC SQL BEGIN DECLARE SECTION;

// Allocate CLOB for the request parameter
SQL TYPE is CLOB(1M)      request;

// Allocate CLOB for the metadata parameter
SQL TYPE is CLOB(1M)      metadata;

// Allocate CLOB for the response parameter
SQL TYPE is CLOB(1M)      response;

EXEC SQL END DECLARE SECTION;

// Connect to database and other application initializations
// ...

// Populate the request parameter structure with the operation
strcpy(request.data, "<request><describe> ... </describe></request>");

// string length with end-of-string
request.length = strlen(request.data) + 1;

// Populate the metadata parameter structure with the metadata
strcpy(metadata.data, "");

// string length with EOS
metadata.length = strlen(metadata.data) + 1;

// Call DB2 Cube Views stored procedure
EXEC SQL CALL "DB2INFO.MD_MESSAGE"(:request,:metadata,:response);

// Check that the stored procedure has returned without errors
if (sqlca.sqlcode)
{
// error checking using sqlaintp()
}

// Process response parameter structure to determine success of operation
// ...

// Process metadata parameter structure to extract requested metadata
// ...

// Disconnect from database and other application terminations
// ...
```

## DB2 Cube Views API 参数

### DB2 Cube Views API 元数据操作的参数

DB2 Cube Views 的 API 提供了三种类型的元数据功能：检索、修改和管理。每种类型的功能包括一个或多个操作，而每个操作具有一组参数。

参数格式定义在商业智能应用程序与 DB2 Cube Views 之间表示和交换元数据操作与对象所依据的标准。参数格式使用 XML 来表示 DB2 Cube Views 元数据操作和对象。XML 模式定义参数格式。

### 输入和输出参数

存储过程 md\_message 中的每个操作有两个输入和两个输出参数。

输入参数包括**请求**参数和**元数据**参数。输出参数包括**响应**参数和**元数据**参数。

输入	参数内容
请求	包含请求存储过程的操作。该请求可能包括影响请求的行为和作用域的选项。
元数据	包含将与 <b>请求</b> 参数中描述的操作配合使用的元数据对象。  某些请求操作（例如，描述）不需要输入元数据。
输出	
响应	包含存储过程所执行操作的所有结果（除元数据对象之外）。
元数据	包含 <b>请求</b> 输入参数中所描述操作请求的元数据对象。  某些操作（例如，创建）不返回输出元数据。

图 40 显示请求应用程序中的输入参数如何通过 DB2 Cube Views API 传递至输出参数。API 创建 DB2 UDB 可读取的元数据对象的说明。

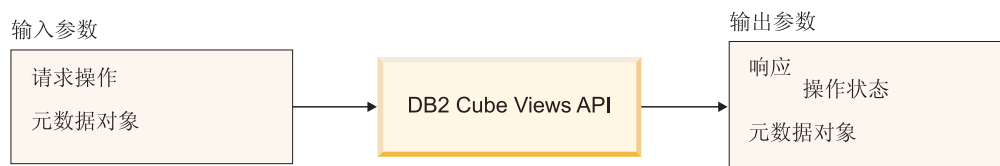


图 40. API 参数

## DB2 Cube Views 元数据操作

### 检索操作：描述

DB2 Cube Views 包括一个检索描述：描述。

描述操作支持 XML 版本 8.1.2.1.0 和 8.2.0.1.0。所有其它操作都需要 XML 版本 8.2.0.1.0。有关 XML 模式文件的更多信息，请参阅第 140 页的『DB2 Cube Views 元数据表和 XML 模式文件』。

## 描述

此操作检索元数据对象信息。此操作返回指定 *objectType* 的一个或多个元数据对象（例如，单一维对象、维对象集、包括所有对象类型的对象集）的信息。元数据对象在元数据参数中返回。用于表示检索出的元数据对象的格式在第 141 页的『元数据对象格式』中描述。在递归方式中，此操作还返回所有其它元数据对象的信息，这是因为它会遍历对象关联路径（从符合 *objectType* 规范的对象集开始）。

表 39 列示描述请求参数中的组件。表 40 列示描述响应参数的组件。

表 39. 描述请求参数

名称	类型	数据类型	有效值	描述
objectType	自变量	XML 字符串	all cubeModel cube cubeFacts cubeDimension cubeHierachy cubeLevel dimension facts hierarchy attributeRelationship level join measure attribute	可检索的 DB2 Cube Views 元数据对象的类型。可对此参数指定其中一个有效值。
restriction [可选]	自变量	XML 字符串	请参阅第 131 页的『操作参数』。	限制元数据检索的作用域。这与 SQL 中的谓词相似。
recurse	自变量	XML 字符串	yes no	启用或禁用元数据对象的递归检索。

表 40. 描述响应参数

名称	类型	数据类型	有效值	描述
status	状态消息	DB2 Cube Views 消息结构	请参阅第 134 页的『消息结构』小节。	指示描述操作的状态的消息。
object	检索结果	XML	请参阅第 134 页的『操作的操作数』。	请求的元数据数据对象。如果在操作过程中出错，则此值为空字符串。

## 建议操作: 建议

此操作建议应该构建这些摘要表来改进指定立方体模型的查询性能。此操作具有一些自变量，它们限制顾问程序可以运行的时间长短以及它可以用于摘要表的磁盘空间量。

表 41. 建议请求参数

名称	类型	数据类型	有效值	描述
cubeModelRef	自变量	XML 元素		要优化的立方体模型。
tablespaceName [可选]	自变量	XML 字符串		用于摘要表的表空间。如果没有提供表空间，则生成的 SQL 将不指定表空间。在这种情况下，DB2 UDB 将把表放置在缺省表空间中。
indexspaceName [可选]	自变量	XML 字符串		用于摘要表索引的表空间。如果没有提供表空间，则生成的 SQL 将不指定表空间。在这种情况下，DB2 UDB 将把索引放置在缺省表空间中。
diskspaceLimit [可选]	自变量	XML 非负整数		可用于摘要表及其索引的磁盘空间（以字节计）。指定 0 表示顾问程序决定要使用多少磁盘空间。为顾问程序提供的磁盘空间越多，它生成的建议也就越好。指定 0 意味着无限制。缺省值为 0。
timeLimit [可选]	自变量	XML 非负整数		应该用来建议查询的时间量（以秒计）。为顾问程序提供的磁盘空间越多，它生成的建议也就越好。指定 0 意味着无限制，且当顾问程序确定多花费时间并不能生成更好的建议时，它会停止。缺省值为 0。
sampling [可选]	自变量	XML 字符串	yes 或 no	指定是否应该完成立方体模型基本表的数据采样。使用采样通常会改进建议，但这会增加顾问程序运行的时间。如果不允许采样，则顾问程序仅根据数据库统计信息生成建议。借助数据采样，顾问程序可同时利用数据库统计信息和采样。如果具有小型的原始数据副本，则可以更改 DB2 UDB 统计信息以使这些表看起来与原始表一样大。如果没有指定采样，则顾问程序作出的建议与在没有采样的情况下根据原始表作出的建议相同。
refresh [可选]	自变量	XML 字符串	延迟或立即	指定当基本表更改时 DB2 UDB 是否应该尝试立即刷新摘要表以确保它们同步。延迟刷新意味着必须指定刷新摘要表的时间。对于立即刷新，DB2 将在更新基本表的同时更新摘要表。如果基本表是使用数据装入更新的，则必须手工指定刷新。立即刷新摘要表是增量更新的，而延迟刷新摘要表则是完全重新构建的。使用立即刷新时存在许多限制，而且即使指定立即刷新，顾问程序也可能选择使用延迟刷新。

以下示例显示“建议”操作:

```
<olap:request xmlns:olap="http://www.ibm.com/olap" ... >
<advise
  tablespaceName="TS_MQTTABLE"
  indexspaceName="TS_MQTINDEX"
  diskSpaceLimit="75000"
  timeLimit="300"
  sampling="yes"
  refresh="immediate">
<cubeModelRef name="SalesModel" schema="MDSAMPLE"/>
</advise>
</olap:request>
```

表 42. 建议响应参数

名称	类型	数据类型	有效值	描述
status	状态消息	DB2 Cube Views 消息结构		指示“建议”操作的状态的消息。
info	消息列表	消息结构列表		警告消息，指示遇到的任何非致命情况以及在“建议”请求中指定的内容进行的“建议”操作结果导致的可能更改。例如，由于建议的 MQT 中存在非分布式量度，所以“建议”操作可能覆盖为 <b>REFRESH IMMEDIATE</b> MQT 指定的内容。参考消息可能显示导致模型的一些部件未被优化的情况或显示有关哪些优化已完成的信息（包括选择聚集和索引的原因）。
recommendation		XML 元素		所有建议的摘要表和索引所需的总磁盘空间的估计。
sql		XML 元素		用于创建并填充摘要表以及为它创建索引的 SQL。将由您运行该 SQL。
refreshSql		XML 元素		用于刷新摘要表以使它们与已更新的基本表同步的 SQL。将由您运行该 SQL。对于立即刷新摘要表，DB2 UDB 有时会在更新基本表时自动更新这些摘要表。此操作取决于更改基本表的方式。

以下示例显示“建议”响应操作:

```
<olap:response xmlns:olap="http://www.ibm.com/olap" ... >
<advise>
<status id="0" text="Operation completed successfully.
No errors were encountered." type="informational"/>
<info>
<message id="7401" text="The DB2INFO.MQT0000000041T01 summary table
is recommended.
It is estimated to have 100 rows, 55KB table size and 5KB index size."/>
<message id="7401" text="The DB2INFO.MQT0000000041T02 summary table
is recommended.
It is estimated to have 8 rows, 4KB table size and 1KB index size."/>
<message id="7406" text="The PART dimension does not have any
hierarchies that can be optimized by the Optimization Advisor.
The recommendations will not optimize for any attributes
```



```

from this dimension."/>
</info>
<recommendation diskpace="65108"/>
<sql>
<![CDATA[
-- *****
-- * Script to create/refresh summary tables.
-- *
-- * Cube model schema: MDSAMPLE
-- * Cube model name: SalesModel
-- * Diskspace limit: 75000
-- * Time limit: 300
-- * Sampling: Yes
-- * Refresh type: Refresh immediate
-- * Tablespace name: TS_MQTTABLE
-- * Indexspace name: TS_MQTINDEX
-- *****

DROP TABLE DB2INFO.MQT0000000041T01;

DROP TABLE DB2INFO.MQT0000000041T02;

UPDATE COMMAND OPTIONS USING c OFF;

CREATE SUMMARY TABLE DB2INFO.MQT0000000041T01 AS
(SELECT
SUM(T2."COGS") AS "COGS",
SUM(T2."MARKETING"+T2."PAYROLL") AS "EXPENSE",
SUM(T2."MARKETING") AS "MARKETING",
SUM(T2."PAYROLL") AS "PAYROLL",
SUM(T2."SALES"-(T2."COGS"+(T2."MARKETING"+T2."PAYROLL"))) AS "PROFIT",
SUM(T2."SALES") AS "SALES",
COUNT(*) AS "COUNT",
T5."REGION" AS "REGION",
T5."DIRECTOR" AS "DIRECTOR",
T6."FAMILY" AS "FAMILY",
T6."FAMILYNAME" AS "FAMILYNAME",
T3."SKU" AS "SKU",
T3."CAFFEINATED" AS "CAFFEINATED",
T3."OUNCES" AS "OUNCES",
T3."PKGTYPE" AS "PKGTYPE",
T3."SKUNAME" AS "SKUNAME",
T4."YEAR" AS "YEAR"

FROM
"MDSAMPLE"."MARKET" AS T1,
"MDSAMPLE"."SALESFACT" AS T2,
"MDSAMPLE"."PRODUCT" AS T3,
"MDSAMPLE"."TIME" AS T4,
"MDSAMPLE"."REGION" AS T5,
"MDSAMPLE"."FAMILY" AS T6

WHERE
T1."STATEID"=T2."STATEID" AND
T3."PRODUCTID"=T2."PRODUCTID" AND
T4."TIMEID"=T2."TIMEID" AND
T1."REGIONID"=T5."REGIONID" AND
T3."FAMILYID"=T6."FAMILYID"

GROUP BY
T5."REGION",
T5."DIRECTOR",
T6."FAMILY",
T6."FAMILYNAME",
T3."SKU",
T3."CAFFEINATED",
T3."OUNCES",

```

```

T3."PKGTYPE",
T3."SKUNAME",
T4."YEAR")

DATA INITIALLY DEFERRED
REFRESH IMMEDIATE
IN "TS_MQTTABLE"
INDEX IN "TS_MQTINDEX"
NOT LOGGED INITIALLY;

COMMENT ON TABLE DB2INFO.MQT0000000041T01 IS 'AST created for
cube model MDSAMPLE.SalesModel';

REFRESH TABLE DB2INFO.MQT0000000041T01;

CREATE INDEX DB2INFO.IDX0000000041T0101 ON DB2INFO.MQT0000000041T01("FAMILY",
"SKU");
CREATE INDEX DB2INFO.IDX0000000041T0102 ON DB2INFO.MQT0000000041T01("REGION");

RUNSTATS ON TABLE DB2INFO.MQT0000000041T01 AND INDEXES ALL;

CREATE SUMMARY TABLE DB2INFO.MQT0000000041T02 AS
(SELECT
SUM(T2."COGS") AS "COGS",
SUM(T2."MARKETING"+T2."PAYROLL") AS "EXPENSE",
SUM(T2."MARKETING") AS "MARKETING",
SUM(T2."PAYROLL") AS "PAYROLL",
SUM(T2."SALES"-(T2."COGS"+(T2."MARKETING"+T2."PAYROLL"))) AS "PROFIT",
SUM(T2."SALES") AS "SALES",
COUNT(*) AS "COUNT",
T1."YEAR" AS "YEAR",
T1."QUARTER" AS "QUARTER"

FROM
"MDSAMPLE"."TIME" AS T1,
"MDSAMPLE"."SALESFACT" AS T2

WHERE
T1."TIMEID"=T2."TIMEID"

GROUP BY
T1."YEAR",
T1."QUARTER")

DATA INITIALLY DEFERRED
REFRESH IMMEDIATE
IN "TS_MQTTABLE"
INDEX IN "TS_MQTINDEX"
NOT LOGGED INITIALLY;

COMMENT ON TABLE DB2INFO.MQT0000000041T02 IS 'AST created for
cube model MDSAMPLE.SalesModel';

REFRESH TABLE DB2INFO.MQT0000000041T02;

CREATE INDEX DB2INFO.IDX0000000041T02C ON DB2INFO.MQT0000000041T02("QUARTER")
CLUSTER;

REORG TABLE DB2INFO.MQT0000000041T02;

RUNSTATS ON TABLE DB2INFO.MQT0000000041T02 AND INDEXES ALL;

COMMIT;

]]>
</sql>
<refreshSql>

```

```

<![CDATA[
-- *****
-- * Script to create/refresh summary tables.
-- *
-- * Cube model schema: MDSAMPLE
-- * Cube model name: SalesModel
-- * Diskspace limit: 75000
-- * Time limit: 300
-- * Sampling: Yes
-- * Refresh type: Refresh immediate
-- * Tablespace name: TS_MQTTABLE
-- * Indexspace name: TS_MQTINDEX
-- *****

UPDATE COMMAND OPTIONS USING c OFF;

REFRESH TABLE DB2INFO.MQT0000000041T01;
REFRESH TABLE DB2INFO.MQT0000000041T02;
COMMIT;

]]>
</refreshSql>
</advise>
</olap:response>

```

## 修改操作: 改变、创建、删除、导入和重命名

DB2 Cube Views API 包括五个修改操作:

- 改变
- 创建
- 删除
- 导入
- 重命名

当存储过程实现修改操作时, 它还会确保对象完整且有效。

### 创建

“创建”操作创建元数据对象。它接受一个或多个元数据对象操作数, 并按将它们传递至操作的顺序在 DB2 Cube Views 中创建这些对象。传递至此操作的对象序列可包括不同类型的对象。

传递至此操作的对象可以根据需要引用其它对象。如果对象之间存在引用, 它们必须反映在对象的排序中。例如, 如果对象 MyObject 引用对象 YourObject, 则在传递 MyObject 之前, YourObject 必须传递至该操作。有关元数据对象如何互相引用的更多信息, 请参阅第 141 页的『元数据对象格式』。

“创建”操作验证每个对象。如果要创建的对象已存在, 或者由要创建的对象引用的对象还不存在, 则会返回错误。如果输入对象指定不存在的模式, 该操作将创建该模式 (如果您在数据库中具有足够的权限)。

表 43. 创建请求参数

名称	类型	数据类型	有效值	描述
object	操作数	XML 元素	请参阅第 134 页的『操作的操作数』。	可被创建的对象。

表 44. 创建响应参数

名称	类型	数据类型	有效值	描述
status	状态消息	DB2 Cube Views 消息结构	请参阅第 134 页的『消息结构』一节。	指示“创建”操作的状态的消息。

## 改变

“改变”操作更新元数据对象信息。它接受一个或多个元数据对象操作数，并更新元数据目录表中的对应对象操作数。对象是按将它们传递至操作的顺序更新的。传递至此操作的对象序列可包括不同类型的对象。

传递至此操作的对象可以根据需要引用其它对象。如果对象之间存在引用，它们必须反映在对象的排序中。有关元数据对象如何互相引用的更多信息，请参阅第 141 页的『元数据对象格式』。

此操作无法更新对象的模式或名称。可使用“重命名”操作来更改对象名。

“改变”操作验证每个对象。如果要更新的对象不存在，则会返回错误。

表 45. 改变请求参数

名称	类型	数据类型	有效值	描述
object	操作数	XML 元素	请参阅第 134 页的『操作的操作数』。	可被更新的对象。

表 46. 改变响应参数

名称	类型	数据类型	有效值	描述
status	状态消息	DB2 Cube Views 消息结构	请参阅第 134 页的『消息结构』小节。	指示“改变”操作的状态的消息。

## 重命名

“重命名”操作重命名由其当前模式和名称标识的单个 DB2 Cube Views 元数据对象。只能更改对象的名称。不能更改对象的模式。即使对象当前正由其它元数据对象引用，“重命名”操作也可以重命名这些对象。

表 47. 重命名请求参数

名称	类型	数据类型	有效值	描述
objectType	自变量	XML 字符串	cubeModel cube cubeFacts cubeDimension cubeHierachy cubeLevel dimension facts hierarchy attributeRelationship level join measure attribute	已重命名的 DB2 Cube Views 元数据对象的类型。可对此参数指定其中一个有效值。
currentRef	操作数	DB2 Cube Views 元数据对象引用	请参阅第 134 页的『操作的操作数』。	要重命名的元数据对象的当前模式和名称。
newRef	操作数	DB2 Cube Views 元数据对象引用	请参阅第 134 页的『操作的操作数』。	已重命名的元数据对象的新模式和名称。

表 48. 重命名响应参数

名称	类型	数据类型	有效值	描述
status	状态消息	DB2 Cube Views 消息结构	请参阅第 134 页的『消息结构』小节。	指示“重命名”操作的状态的消息。

## 删除

“删除”操作从 DB2 Cube Views 中删除元数据对象。此操作根据指定的 *objectType* 和 *restriction* 组件删除一个或多个元数据对象。如果删除的对象当前正由另一元数据对象引用，则会返回错误。

表 49. 删除请求参数

名称	类型	数据类型	有效值	描述
objectType	自变量	XML 字符串	请参阅第 131 页的『操作参数』。	正在删除的元数据对象的类型。可对此参数指定其中一个有效值。
restriction [可选]	自变量	XML 字符串	请参阅第 131 页的『操作参数』。	限制元数据删除的作用域。这与 SQL 中的谓词相似。

表 50. 删除响应参数

名称	类型	数据类型	有效值	描述
status	状态消息	DB2 Cube Views 消息结构	请参阅第 134 页的『消息结构』小节。	指示“删除”操作的状态的消息。

## 导入

“导入”操作创建元数据对象或在元数据目录中报告元数据对象的存在。此操作的行为与“创建”操作类似，不过它处理预先存在的元数据对象的存在方式与“创建”操作不同。

可定义“导入”操作的可选方式。这些不同方式确定在尝试导入与目录中已经存在的对象同名的对象时要采取什么操作。

根据运行方式，如果要创建的对象已存在，或者要创建的对象引用的对象不存在，将返回错误。如果输入对象指定不存在的模式，“导入”操作将创建该模式（如果您对数据库有足够权限的话）。

“导入”操作验证每个对象。

有关各种操作方式的详细描述，请参阅第 131 页的『操作参数』。

表 51. 导入请求参数

名称	类型	数据类型	有效值	描述
mode	自变量	XML 字符串	create new - ignore collisions create new - replace collisions create new - abort on collision report new - report collisions	定义已导入的新的和现有的对象的操作。 请参阅第 131 页的『操作参数』中有关方式的描述。
object	操作数	XML 元素	请参阅第 134 页的『操作的操作数』。	正在导入的对象。

表 52. 导入响应参数

名称	类型	数据类型	有效值	描述
status	状态消息	DB2 Cube Views 消息结构	请参阅第 134 页的『消息结构』小节。	指示“导入”操作的状态的消息。
newList	引用列表	XML 元素	请参阅第 131 页的『操作参数』中有关方式的描述。	引用新对象的名称-模式对的列表。
collisionList	引用列表	XML 元素	请参阅第 131 页的『操作参数』中有关方式的描述。	引用与其它对象冲突的对象的名称-模式对的列表。

## API 如何处理修改操作的功能相关性

DB2 Cube Views 元数据 API 根据特定规则管理层对象的功能相关性。第 127 页的表 53 显示 API 对每个层修改请求采取的操作。

表 53. API 针对功能相关性的操作

请求	操作
创建层	API 将创建相应的 DB2 功能相关性（如果可能的话）。如果不能创建功能相关性，API 将创建不带功能相关性的层并返回警告消息。
删除层	API 将删除该层及相关联的 DB2 功能相关性。如果 API 在尝试删除该层或功能相关性时遇到错误，API 将返回错误消息而不删除层对象。
改变层（不带功能相关性）	如果在改变层时不创建功能相关性，将不采取任何操作。  如果在改变层时创建功能相关性，API 将创建相应的 DB2 功能相关性（如果可能的话）。如果不能创建功能相关性，API 将改变不带功能相关性的层并返回警告消息。
改变层（带有功能相关性）	如果该层具有功能相关性且您想要在改变该层时删除功能相关性，API 将改变该层并删除相关联的 DB2 功能相关性。如果 API 在尝试改变该层时遇到错误，API 将返回错误消息而不改变层对象。  如果想要改变层并保留功能相关性，API 将删除功能相关性并重新创建它。如果不能删除功能相关性，API 将返回错误消息但继续改变该层并以另一名称重新创建新的功能相关性。
导入层（Create new: ignore collisions 方式）	API 将创建相应的 DB2 功能相关性（如果可能的话）。如果不能创建功能相关性，API 将创建不带功能相关性的层并返回警告消息。
导入层（Create new: replace collisions 方式）	与“改变”相同。

## 管理操作：验证和转换

DB2 Cube Views 包括两个管理操作：验证和转换。仅当从 DB2 Cube Views V8.1 迁移至 DB2 Cube Views V8.2 时才使用“转换”操作。转换操作将版本 8.1 元数据 XML 映射至版本 8.2 元数据 XML。

### 验证

“验证”操作检查一个或多个元数据对象的有效性。有效性是根据对象是否符合 DB2 Cube Views 对象规则定义的。对于此操作验证的对象，指定 *objectType* 自变量和 *restriction* 参数。使用 *mode* 参数来指定验证的扩展数据块。

“验证”操作检查下列问题：

- 元数据对象信息的完整性
- 元数据对象之间的引用完整性
- 是否存在被引用的关系表、视图、别名和昵称列。
- 元数据对象（例如，属性和量度）中存储的 SQL 表达式是否正确。

当“验证”操作发现无效元数据对象时，它会停止。当此操作发现违规时，它会返回描述验证违规的消息。其它操作（创建、改变和导入）还会隐式地验证元数据对象。转换操作不验证元数据对象。

表 54. 验证请求参数

名称	类型	数据类型	有效值	描述
objectType	自变量	XML 字符串	请参阅第 131 页的『操作参数』。	正在验证的 DB2 Cube Views 元数据对象的类型。可对此参数指定其中一个有效值。
restriction (可选)	自变量 [可选]	XML 字符串	请参阅第 131 页的『操作参数』。	限制元数据验证的作用域。这与 SQL 中的谓词相似。
mode	自变量	XML 字符串	base cubeModel completeness optimization	定义要执行的验证操作的扩展数据块。

表 55. 验证响应参数

名称	类型	数据类型	有效值	描述
status	状态消息	DB2 Cube Views 消息结构	请参阅第 134 页的『消息结构』小节。	指示“验证”操作的状态的消息。
info	消息列表	消息结构列表	请参阅第 134 页的『消息结构』小节。	描述“验证”操作生成的警告和错误的消息列表。

## 转换

“转换”操作通过验证 XML 模式检查入局元数据 XML 在句法上是否正确，但不验证列或其它引用。可以使用“转换”操作来转换任意 DB2 Cube Views 元数据。“转换”操作需要完整的 XML 文档。对不在 XML 文档中的对象的引用可能导致错误。

“转换”操作会映射一些对象（如下表所示）。除非另行说明，否则每个版本 8.1 对象都将映射至版本 8.2 中的同一对象。

表 56. 版本 8.1 对象如何映射至版本 8.2 对象

版本 8.1 对象	版本 8.2 对象
属性	<ul style="list-style-type: none"> <li>属性被扩展为具有新的属性可空性。</li> <li>现有属性的可空性为未知。</li> </ul>
连接	无任何更改。
属性关系	<ul style="list-style-type: none"> <li>不再被层次结构和立方体层次结构引用。</li> <li>所有属性关系都变为孤立的。（没有其它对象引用这些属性关系。）</li> </ul> <p>当从 OLAP 中心导出立方体模型或立方体时，导出的 XML 文件中不会再显示属性关系。从 OLAP 中心中，必须导出所有元数据以导出属性关系。</p>



表 56. 版本 8.1 对象如何映射至版本 8.2 对象 (续)

版本 8.1 对象	版本 8.2 对象
层次结构 + 属性 + 属性关系	<ul style="list-style-type: none"> <li>• 将为被任何版本 8.1 层次结构引用的每个属性恰好创建一个层对象。</li> <li>• 该层对象将与版本 8.1 源属性使用相同的名称（模式和名称等等）。</li> <li>• 该层的缺省属性将是源属性。</li> <li>• 该层的键包含基于其中一个版本 8.1 层次结构的源属性以及该源属性之上的祖先属性。</li> <li>• 如果在多个版本 8.1 层次结构中引用了该源属性，则会选择具有最少祖先的层次结构（如果祖先个数相同，则会任意选择一个）并将祖先属性作为版本 8.2 层中的键属性自顶向下添加至源。</li> <li>• 左边属性等同于源属性的层次结构引用的所有版本 8.1 属性关系都将用来创建该层中的相关属性。对于每个匹配（在左边）属性关系，会将右边属性添加至该层作为相关属性。创建相关属性列表时，将除去重复内容。</li> </ul>
立方体层次结构 + 属性 + 属性关系	<ul style="list-style-type: none"> <li>• 将为每个版本 8.1 立方体层次结构引用的每个属性创建唯一的立方体层对象。</li> <li>• 立方体层是根据属性名称和立方体名称来命名的。如果立方体名为 Sales 而属性名为 Region，则该立方体层将名为 Region (Sales)。</li> <li>• 每个立方体层都会引用它相应的层对象。例如，Region (Sales) 指向 Region 层。</li> <li>• 立方体层的相关属性是基于给定立方体层次结构中的属性关系对象的。对于左边属性等同于源属性（在父层中）的每个属性关系，右边属性将添加至立方体层作为相关属性。</li> </ul>
层次结构	每个版本 8.1 层次结构都映射至具有一组已排序的层的版本 8.2 层次结构（无属性且无属性关系）。
立方体层次结构	每个版本 8.1 立方体层次结构都映射至具有一组已排序的立方体层的版本 8.2 立方体层次结构（无属性且无属性关系）。
维	添加了关联层。
立方体维	无任何更改。
量度	与属性一样，量度被扩展为具有新的属性可空性。现有量度的可空性为未知。
事实	无任何更改。
立方体事实	无任何更改。
立方体模型	无任何更改。
立方体	转换期间无任何更改，但是在版本 8.2 中，立方体可以包含使用信息。

## 元数据操作中的样本输入和输出参数

以下样本显示如何在三种类型的元数据操作中构造参数。在这些示例中 XML 结构部分被排除，它们用省略号 (.) 表示。

### 检索操作

以下样本显示如何构造称为描述的检索操作。有关描述操作的更多信息，请参阅第 117 页的『检索操作：描述』。在以下示例中，元数据参数在输入上为空，但在输出上已填充。

### 请求和元数据

```
<olap:request xmlns:olap="http://www.ibm.com/olap" ... >
  <describe objectType="cube" recurse="no">
    <restriction>
      <predicate property="schema" operator="=" value="myschema"/>
    </restriction>
```

```

    </describe>
</olap:request>

<olap:metadata xmlns:olap="http://www.ibm.com/olap" ... />

```

### 响应和元数据

```

<olap:response xmlns:olap="http://www.ibm.com/olap" ... >
  <describe>
    <status id="0" text="Operation completed
      successfully."type="informational"/>
  </describe>
</olap:response>

<olap:metadata xmlns:olap="http://www.ibm.com/olap" ... >
  <cube name="cube1" schema="myschema" ... > ... </cube>
  ...
  <cube name="cubeN" schema="myschema" ... > ... </cube>"
</olap:metadata>

```

### 修改操作

以下样本显示如何构造称为创建的修改操作。有关“创建”操作和其它修改操作的更多信息，请参阅第 123 页的『修改操作：改变、创建、删除、导入和重命名』。元数据参数在输入上已填充但在输出上为空。

### 请求和元数据

```

<olap:request xmlns:olap="http://www.ibm.com/olap" ... >
  <create/>
</olap:request>

<olap:metadata xmlns:olap="http://www.ibm.com/olap" ... >
  <attribute name="LocationID" ... > ... </attribute>
  <attribute name="Country" ... > ... </attribute>
  <attribute name="State" ... > ... </attribute>
  <attribute name="City" ... > ... </attribute>
  <dimension name="Location" ... type="regular">
    <attributeRef name="LocationID" ... </attributeRef>
    <attributeRef name="Country" ... </attributeRef>
    <attributeRef name="State" ... </attributeRef>
    <attributeRef name="City" ... </attributeRef>
    ...
  </dimension>
</olap:metadata>

```

### 响应和元数据

```

<olap:response xmlns:olap="http://www.ibm.com/olap" ... >
  <create>
    <status id="0" text="Operation completed
      successfully."type="informational"/>
  </create>
</olap:response>

<olap:metadata xmlns:olap="http://www.ibm.com/olap" ... >

```

### 管理操作

以下示例显示如何构造称为验证的管理操作。有关验证操作的更多信息，请参阅第 127 页的『管理操作：验证和转换』。

## 请求和元数据

```
<olap:request xmlns:olap="http://www.ibm.com/olap" ... >
  <validate objectType="cube" mode="base">
    <restriction>
      <predicate property="schema" operator="=" value ="myschema"/>
    </restriction>
  </validate>
</olap:request>

<olap:metadata xmlns:olap="http://www.ibm.com/olap" ... />
```

## 响应和元数据

```
<olap:response xmlns:olap="http://www.ibm.com/olap" ... >
  <validate>
    <status id="1" text="...Additional information
      returned."type="informational"/>
    <info><message id="6299" text="At least one
      database view was found during validation."
      type="warning"/></info>
  </validate>
</olap:response>

<olap:metadata xmlns:olap="http://www.ibm.com/olap" ... >
```

可与 `db2mdapiclient` 实用程序配合使用的其它操作 XML 样本文件在 `SQLLIB\samples\olap\xml\input` 目录中。

## 操作参数

为每个元数据操作提供了各种参数。这些参数定制操作的行为以形成其特定应用。

DB2 Cube Views 为元数据操作提供了五个参数:

- objectType
- recurse
- restriction
- mode (用于“导入”操作)
- mode (用于验证操作)

### objectType 参数

此参数指定请求的操作中涉及的元数据对象的类型。以下对象类型直接对应于 DB2 Cube Views 元数据对象模型。

- all
- cubeModel
- cube
- cubeDimension
- cubeFacts
- cubeHierarchy
- cubeLevel
- dimension
- facts
- hierarchy

- attributeRelationship
- level
- join
- measure
- attribute

### **recurse** 参数

此参数控制操作是否以递归方式执行。在非递归方式中，操作仅对与 `objectType` 参数和 **限制** 参数规范直接匹配的元数据对象执行其操作。操作从非递归方式元数据对象集开始，此外还对所有其它元数据对象执行其操作，这是因为它会遍历对象关联路径（从非递归方式对象集开始）。**recurse** 参数包括 `yes` 选项和 `no` 选项。

例如，非递归操作可能返回维列表；而递归操作可能不仅返回维列表，还返回那些维引用的所有其它对象（不同类型）和那些维对象引用的对象。

### **restriction** 参数

此参数指定元数据操作的作用域受到限制。此参数类似于 SQL 查询中的谓词。限制在 XML 中是通过使用 `<restriction>` 和 `<predicate>` 标记（由 DB2 Cube Views XML 模式定义）表达的。

限制可以基于所有元数据对象公用的对象属性以及元数据对象之间的关系。

基于特性的谓词包含下列属性：

#### *property*

与谓词标记相关联，且必须指定 `name` 属性或 `schema` 属性。

#### *operator*

与谓词标记相关联且必须指定等号 (=)。

*value* 与谓词标记相关联，且充当要与 *property* 属性指定的特性进行比较的值的字符串说明。

有关**限制**参数如何与操作步骤的总体顺序相关的描述，请参阅第 135 页的『操作步骤的顺序』。

此示例将操作的作用域限制为 ABC 模式中的对象：

```
<restriction>
  <predicate property="schema" operator="=" value="ABC">
</restriction>
```

### **mode**（用于导入）参数

此参数设置“导入”操作的方式。下表描述可用方式。

当作为输入传递至“导入”操作的对象已经存在于元数据目录中时，将发生冲突。

表 57. 导入方式

方式	描述	返回的引用列表
Create new: ignore collisions	<ul style="list-style-type: none"> <li>创建不冲突的输入对象。</li> <li>不创建冲突的输入对象。</li> <li>不改变预先存在的对象。</li> <li>冲突不生成错误。</li> </ul>	<p><b>newList</b></p> <p>包含成功创建的对象名称 - 模式对。</p> <p><b>collisionList</b></p> <p>包含冲突中被忽略且因此未创建的对象名称 - 模式对。</p>
Create new: replace collisions	<ul style="list-style-type: none"> <li>创建不冲突的输入对象。</li> <li>冲突的输入对象替换预先存在的对象。</li> <li>输入对象替换预先存在的对象。</li> <li>冲突不生成错误。</li> </ul>	<p><b>newList</b></p> <p>包含成功创建的对象名称 - 模式对。</p> <p><b>collisionList</b></p> <p>包含冲突中被替换的对象名称 - 模式对。</p>
Create new: abort on collision	<ul style="list-style-type: none"> <li>只有在整个操作不存在冲突时才创建输入对象。</li> <li>在出现冲突的情况下，不会将对象创建为操作的一部分。</li> <li>不改变预先存在的对象。</li> <li>冲突生成错误。</li> </ul>	<p><b>newList</b></p> <p>包含成功创建的对象或未创建的非冲突对象名称 - 模式对。</p> <p><b>collisionList</b></p> <p>包含冲突中未创建的对象名称 - 模式对。</p>
Report new: report collisions	<ul style="list-style-type: none"> <li>不创建对象。</li> <li>报告输入对象的冲突状态。</li> <li>不改变预先存在的对象。</li> <li>冲突不生成错误。</li> </ul>	<p><b>newList</b></p> <p>包含冲突中未涉及未创建的对象名称 - 模式对。</p> <p><b>collisionList</b></p> <p>包含冲突中未创建的对象名称 - 模式对。</p>

newList 和 collisionList 引用列表的标记必须遵循预定义排序。以下列表显示 IMPORT 操作的引用类型的排序:

1. <attributeRef>
2. <joinRef>
3. <attributeRelationshipRef>
4. <levelRef>
5. <cubeLevelRef>
6. <hierarchyRef>
7. <cubeHierarchyRef>
8. <dimensionRef>
9. <cubeDimensionRef>
10. <measureRef>

11. <factsRef>
12. <cubeFactsRef>
13. <cubeModelRef>
14. <cubeRef>

### mode (用于验证) 参数

此参数设置验证操作的方式。下表描述验证操作的可用方式。下表中的规则引用 DB2 Cube Views 对象规则类别。

表 58. 验证方式

方式	描述
base	检查是否符合基本规则
cubeModel completeness	<ul style="list-style-type: none"> <li>• 检查是否符合立方体模型完整性规则</li> <li>• 检查是否符合基本规则</li> </ul>
optimization	<ul style="list-style-type: none"> <li>• 检查是否符合优化规则</li> <li>• 检查是否符合立方体模型完整性规则</li> <li>• 检查是否符合基本规则</li> </ul>

## 操作的操作数

当操作要求随请求附带元数据对象或元数据对象的引用时，这些对象或引用称为操作的“操作数”。

通过使用**请求**或**元数据**参数传递至元数据操作的操作数包括:

**object** 此操作数包含所依据的元数据对象。在第 141 页的『元数据对象格式』中描述了用于表示元数据对象的格式。

### currentRef

此操作数是用于“重命名”操作的，它包含元数据对象的模式和名称。

### newRef

与 *currentRef* 操作数类似，此操作数用于“重命名”操作，它包含元数据对象的模式和名称。

## 消息结构

DB2 Cube Views API 返回具有特定结构的参考消息、警告消息或错误消息。

下表描述 DB2 Cube Views 消息的组件:

表 59. 消息组件

组件	描述
标识	消息的唯一整数标识。
类型	消息可以为以下三种类型中的一个: <ul style="list-style-type: none"> <li>• 参考</li> <li>• 警告</li> <li>• 错误</li> </ul>

表 59. 消息组件 (续)

组件	描述
文本	包含消息正文的字符串。
标记	<p>代入消息的文本字符串的值。消息可以包括任意数目的标记。以下 XML 元素可以作为标记出现在消息中:</p> <ul style="list-style-type: none"> <li>• attributeRef</li> <li>• joinRef</li> <li>• attributeRelationshipRef</li> <li>• levelRef</li> <li>• cubeLevelRef</li> <li>• hierarchyRef</li> <li>• cubeHierarchyRef</li> <li>• dimensionRef</li> <li>• cubeDimensionRef</li> <li>• measureRef</li> <li>• factsRef</li> <li>• cubeFactsRef</li> <li>• cubeModelRef</li> <li>• cubeRef</li> <li>• column</li> <li>• text</li> </ul>

以下是不包含标记的消息的示例:

```
<status id="0" text="Operation completed successfully." type="informational"/>
```

以下是包含标记的消息的示例:

```
<status id="6331" text="The left attribute for
the &quot;MDOBJ_ID_ATTRIBUTERELATIONSHIP.MDSAMPLE.State_PopGroup&quot;
attribute relationship is not a part of the
&quot;MDOBJ_HIERARCHY.MDSAMPLE.RegionState&quot; hierarchy."
type="error">
<tokens>
<attributeRelationshipRef name="State_PopGroup" schema="MDSAMPLE"/>
<text value="MDOBJ_HIERARCHY.MDSAMPLE.RegionState"/>
</tokens>
</status><
```

## 操作步骤的顺序

只有三个操作自变量确定操作的作用域。

这三个自变量按它们应用于操作的顺序列示如下:

1. objectType
2. restriction
3. recurse

以下示例显示与预计的限制明显不匹配但可能会在操作的递归阶段中返回的某些对象的返回方式。

示例: 以递归方式描述属于模式 myschema 的立方体:

操作自变量:

```
objectType = "cube"
restriction = <restriction>
    <predicate property="schema" operator="=" value="myschema"/>
</restriction>
recurse = "yes"
```

描述操作通过将其作用域限制为立方体对象开始。在这些立方体对象中, 只会选择属于 myschema 模式的立方体对象。对于每个选择的立方体对象, 将选择它们引用的对象, 并且这些对象具有不同的类型并可能具有不同的模式。然后, 所有选择的对象将作为响应的一部分返回至立方体请求。

---

## 日志记录和跟踪

### DB2 Cube Views API 的运行时跟踪

API 支持三种跟踪优先级。使用配置文件, 管理员可以设置要记录的跟踪的级别。运行时跟踪在缺省情况下是关闭的。缺省跟踪文件名为 db2mdtrace.log。

通常无需运行跟踪。如果在 API 内出现错误并且 IBM 软件支持人员要求您提供跟踪文件, 则可能需要跟踪。

下表描述了不同跟踪级别。

表 60. 跟踪级别

级别	描述	示例
无	<ul style="list-style-type: none"><li>禁用跟踪</li></ul>	不适用
高	<ul style="list-style-type: none"><li>仅跟踪外部和内部 API 条目以及出口点</li><li>跟踪组件之间的流程</li><li>可以包括函数自变量</li></ul>	<ul style="list-style-type: none"><li>开始和结束分析</li><li>开始和结束创建、描述和删除等等</li></ul>
中	<ul style="list-style-type: none"><li>跟踪外部和内部 API 中复杂函数之间的控制流程</li><li>跟踪组件之间的流程</li><li>包括高级别跟踪点</li></ul>	<ul style="list-style-type: none"><li>显示“创建”操作进行的复杂函数调用</li></ul>
低	<ul style="list-style-type: none"><li>跟踪内部 API 中的简单或原子函数</li><li>包括高等级和中等级别跟踪点</li><li>对大多数跟踪点使用此级别</li></ul>	<ul style="list-style-type: none"><li>显示对元数据对象 get 或 set 的方法的调用</li></ul>

在启用跟踪时, 如果将级别设置为除无之外的值, 则可能会同时在错误日志和跟踪日志中记录 API 中发生的错误, 这要视这些日志的级别和严重性设置而定。

### DB2 Cube Views API 的日志文件

API 日志文件在 DB2 实例层生成。错误日志文件的名称为 db2mderror.log, 跟踪日志文件的名称为 db2mdtrace.log。

对于运行 DB2 Cube Views API 的给定 DB2 实例, 将在 DB2 诊断数据目录 (也称为 DB2DIAG) 中生成 API 的日志文件。此 DB2 诊断目录通常在下列目录中:



## 在 Windows 上

DB2 实例路径目录, 例如 c:\sqllib\

## 在 AIX 上

DB2 instance path/db2dump, 例如, ~my inst /sqllib/db2dump

您可以通过使用 DB2DIAG db2 dbm cfg 设置来更改缺省 DB2DIAG 路径。

db2idrop 实用程序可清除与 DB2 实例关联的日志文件。如果不使用 DB2DIAG 的缺省值, 则 db2idrop 实用程序无法清除 DB2 Cube Views API 的日志文件。您必须手工清除 db2idrop 实用程序无法清除的日志文件。DB2 Cube Views API 装入配置文件时发生的错误记录在 db2mdapi.log 文件中。与其它 API 日志一样, db2mdapi.log 文件在 DB2DIAG 路径中。

## 错误日志记录

API 识别三种严重性的错误。缺省严重性设置为中等, 缺省错误日志文件名为 mdererror.log。如果在读取配置文件时出错, 此错误将记录在名为 db2mdapi.log 的文件中。

下表描述了错误严重性级别。

表 61. 错误严重性级别

严重性	描述	示例
无	<ul style="list-style-type: none"><li>忽略所有错误和警告</li></ul>	不适用
高 (大多数错误具有此严重性)	<ul style="list-style-type: none"><li>仅记录关键的不可恢复错误</li><li>将调用堆栈转储至日志</li></ul>	<ul style="list-style-type: none"><li>内部编码错误</li></ul>
中	<ul style="list-style-type: none"><li>记录用户可修复的错误</li><li>同时记录高严重性错误</li><li>将调用堆栈转储至日志</li></ul>	<ul style="list-style-type: none"><li>最终用户错误, 如试图创建重复的对象</li><li>元数据验证错误</li><li>内存不足。可以增加内存或减少使用情况。</li></ul>
低	<ul style="list-style-type: none"><li>记录警告情况</li><li>同时记录高严重性和中等严重性错误</li><li>低严重性错误不会转储调用堆栈</li></ul>	<ul style="list-style-type: none"><li>内部错误的警告</li><li>参考消息</li></ul>

在将 API 配置为高或中等级别错误日志记录时, 如果出现高或中等级别错误, API 将从错误在 API 中出现的位置开始生成调用堆栈。此调用堆栈与中等级别跟踪类似, 但将数据发送至错误日志而不是跟踪日志。

## 日志记录和跟踪方案

下列日志记录和跟踪方案演示如何捕获错误。

**方案 1** (跟踪级别: 中等; 错误严重性: 高): 发生高严重性错误时, 错误同时出现在错误和跟踪日志中。

```
<log>
  <trace level="medium" logFile="db2mdtrace.log" bufferSize="0" />
  <error level="high" logFile="db2mderror.log" bufferSize="0" />
</log>
```

**方案 2**（跟踪级别：中等；错误严重性：低）：发生低严重性错误时，错误仅出现在错误日志中，原因是跟踪日志仅允许中等级别或高级别的条目。

```
<log>
  <trace level="medium" logFile="db2mdtrace.log" bufferSize="0" />
  <error level="low" logFile="db2mderror.log" bufferSize="0" />
</log>
```

与缺少环境变量或访问日志文件失败相关的错误通过存储过程调用的 `SQLSTATE` 返回至数据库客户机应用程序。如果在处理配置文件时出错，此错误将记录在 `db2mdapi.log` 文件中。如果在打开任何用户指定的日志文件时出错，则不会捕获错误。

## 代码页支持

DB2 Cube Views 使用两种代码页：DB2 客户机代码页（应用程序代码页）和 DB2 数据库代码页。请参阅 *DB2 Administration Guide: Planning* 中的“Supported territory codes and code pages”以获取有关如何确定 DB2 客户机代码的信息。DB2 Cube Views API 存储过程在 DB2 数据库代码页中运行。创建数据库时设置 DB2 数据库代码页。DB2 客户机代码页可能与 DB2 数据库代码页不同。CLI 为存储过程将存储过程字符大对象（CLOB）参数从客户机代码页转换为数据库代码页。

下列图表说明客户机如何通过调用级接口（CLI）与服务器通信。CLI 将客户机代码页转换为数据库代码页。

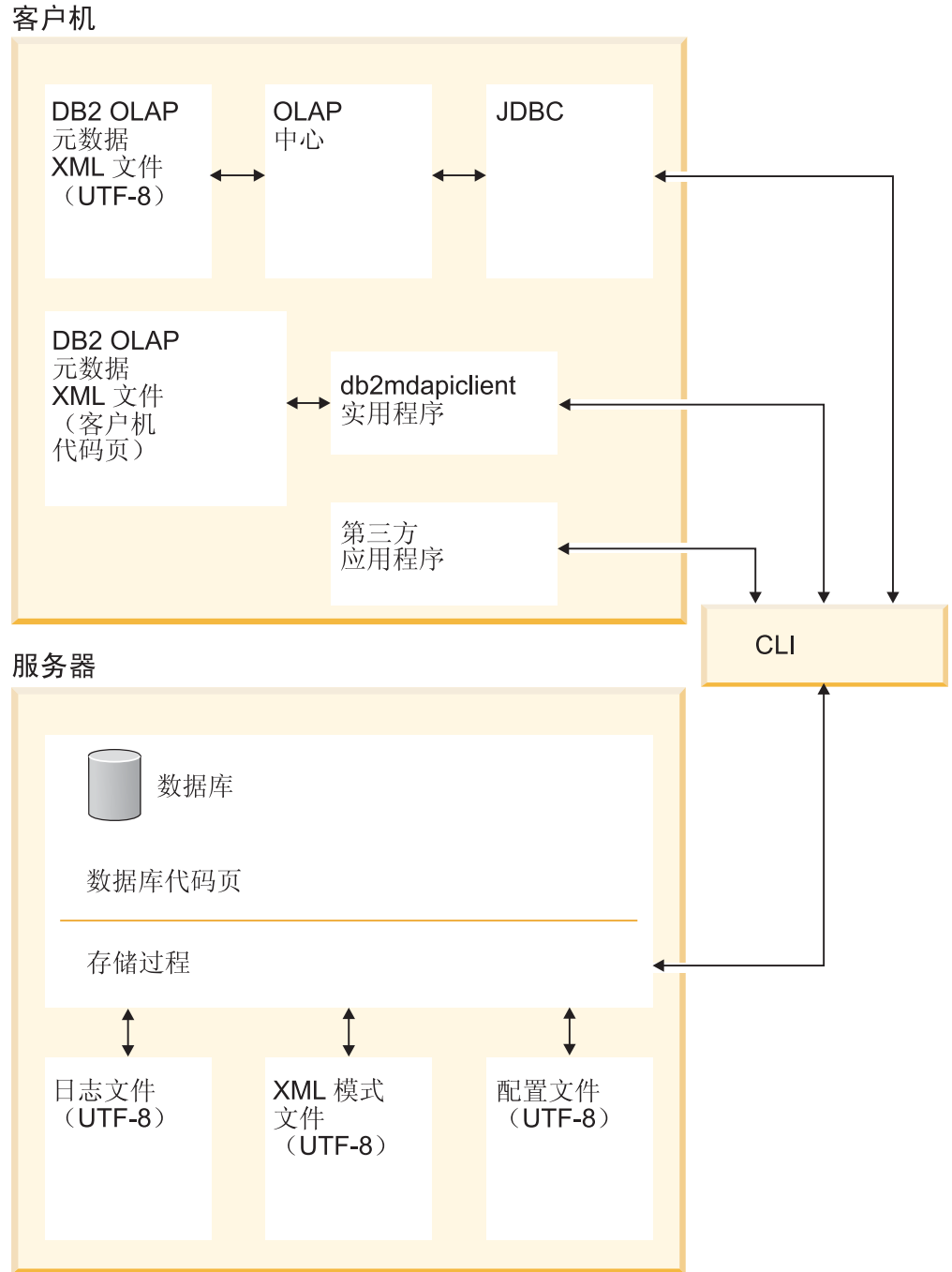


图 41. 数据如何从使用代码页或 UTF-8 格式的不同客户机经过 DB2 CLI 流向数据库服务器

CLI 管理应用程序代码页与数据库代码页之间的转换。将从 DB2 Cube Views 客户机发送到 API 的数据视为输入。将从 API 发送到 DB2 Cube Views 客户机的数据视为输出。输入和输出数据以 DB2 客户机代码页编码。

DB2 Cube Views 的组件具有以下代码页规范。OLAP 中心:

- 仅接受并生成以 UTF-8 编码的 DB2 Cube Views XML 文件
- 如果输入 DB2 Cube Views XML 文件指定 UTF-8 之外的编码, 则会返回错误
- 将文件缺少编码规范解释为文件以 UTF-8 编码

- 通过导出功能创建具有 UTF-8 显式编码规范的 DB2 Cube Views XML 文件

db2mdapiclient:

- 将输入 DB2 Cube Views XML 文件解释为以 UTF-8 编码（除非指定 -1 选项），因此将忽略这些文件中列出的显式编码规范。
- 生成以 UTF-8（除非指定 -1 选项）编码的 DB2 Cube Views XML 文件，在这些文件中不包括显式编码规范。

存储过程 API:

- 将 CLOB 参数解释为以 DB2 客户机代码页编码
- 忽略输入 DB2 Cube Views XML 文件中的显式编码规范
- 生成不包含任何显式编码规范的 DB2 Cube Views XML 文件
- 使用 DB2 数据库代码页处理输入和输出 XML 文件
- 生成使用 UTF-8 编码的 API 日志文件，包括所有嵌入式 DB2 消息。API 日志文件并非 XML 格式。
- 未创建明确声明 UTF-8 编码的日志文件
- 以 UTF-8 编码 API 使用的 XML 模式文件
- 以 UTF-8 编码 XML API 配置文件

对于第三方应用程序，直接调用 DB2 Cube Views API 的其它应用程序必须将以 DB2 客户机代码页编码的 XML 文件作为参数传递和接受。

---

## DB2 Cube Views 元数据表和 XML 模式文件

如果版本号不正确，API 中的某些函数将不受支持。

### 元数据表的版本

仅当 API 连接至具有一组当前的 DB2 Cube Views 元数据表 DB2 数据库时，API 才能起作用。DB2 Cube Views 版本 8.2 的当前版本是 8.2.0.1.0。DB2 Cube Views 版本 8.1 的版本号是 8.1.2.0。DB2 Cube Views 元数据目录表版本号存储在 SYSINFOVERSION 表中。

有关元数据表版本的更多信息，请参阅第 143 页的『db2mdapiclient 实用程序概述』。

DB2 Cube Views XML 模式文件（或 XSD 文件）用于 DB2 Cube Views API。XML 模式文件由 API DB2INFO.MD\_MESSAGE 存储过程使用。

### XML 模式文件的版本

传递至或传递自元数据 API 的所有 XML 文档都必须有版本号。此版本号使得存储过程能识别客户机使用的 XML 模式。XSD 模式文件指定哪些版本号适用于特定的操作。

DB2 Cube Views 版本 8.2 的当前版本是 8.2.0.1.0。

下列模式文件包含了有关存储过程 md\_message() 的信息。

**db2md\_parameter.xsd** 文件

此文件包含有关请求和响应操作的信息。此文件适用于版本 8.1.2.1.0 和版本 8.2.0.1.0。

#### db2md\_metadata.xsd 文件

此文件包括 db2md\_types.xsd 文件并且仅指定元数据元素。此文件不包含版本信息。

#### db2md\_types.xsd 文件

此文件包含有关所有元数据对象的信息。此文件适用于版本 8.1.2.1.0 和版本 8.2.0.1.0。

API 可支持模式文件的先前版本，即版本 8.1.2.1.0，但是仅支持描述和转换操作。如果请求的版本 8.1.2.1.0 操作不是描述或转换，API 将返回错误。

---

## DB2 Cube Views 配置文件

API 可以在 DB2 实例层进行配置。您可以更改名为 db2md\_config.xml 的配置文件的参数。

每次安装 DB2 Cube Views 后，db2\_installation\_path/cfg 目录中都会有一个缺省配置文件。例如，在 Windows 上，缺省配置文件可能位于 c:\sqllib\cfg 目录中，而在 AIX 上，缺省配置文件可能位于 /usr/opt/db2\_08\_01/cfg 目录中。

运行 DB2 Cube Views 的每个 DB2 实例在 db2\_instance\_path 目录中都会有一个 db2md\_config.xml 文件的物理副本。例如，在 Windows 上，物理副本可能在 c:\sqllib\my\_inst 目录中，而在 AIX 上，物理副本可能在 ~my\_inst/sqllib 目录中。

db2icrt 实用程序将缺省配置文件复制至 db2\_instance\_path 目录并创建新实例。对于在安装 DB2 Cube Views 之前创建的 DB2 实例，如果安装程序未成功复制配置文件，您可以手工将该文件复制到实例目录中。如果 API 在实例目录中无法找到配置文件，则 API 会尝试将缺省配置文件复制到实例目录中。

配置文件 db2md\_config.xml 可用于设置错误记录和运行时跟踪。通过修改配置文件的内容，管理员可以指定跟踪的级别、要记录的错误的严重性、记录时使用的缓冲区大小（以字节为单位）。

db2md\_config.xml 配置文件的内容结构由 db2md\_config.xsd XML 模式文件定义。以下示例显示配置文件的内容。

```
<olap:config xmlns:olap="http://www.ibm.com/olap
  xmlns:xsi= http://www.w3.org/2001/XMLSchema-instance
  xmlns:xsd= http://www.w3.org/2001/XMLSchema
  xsi:schemaLocation= http://www.ibm.com/olap db2md_config.xsd">
  <log>
    <trace level="none" logFile="db2mdtrace.log" bufferSize="0"/>
    <error level="medium" logFile="db2mderror.log" bufferSize="0"/>
  </log>
</olap:config>
```

---

## 元数据对象格式

DB2 Cube Views XML 模式定义直接映射至 DB2 Cube Views 元数据对象模型中的对象的基本 XML 元素。然后，复杂元数据结构被表示为这些基本元素的序列。复杂元数据结构内对象之间的关联是通过基本元素之间的名称引用捕获的。

名称引用的一个示例是立方体元素可用于包含对维元素的引用的方式。以下示例显示为下面的 XML 模式定义的立方体对象提供的数据的类型。在此示例中只显示文本描述；在应用程序中将使用信息的 XML 表示法。

```
cube
->cube model reference
->cube dimension references
->cube facts reference
->view
```

在存在立方体对象的情况下，对其它类型对象的引用都包含在表示立方体的基本元素内。通过非递归检索操作，将显示立方体对象（并且仅为立方体对象）的序列。通过递归检索操作，不仅会显示有关立方体对象的信息，而且会显示有关标识的立方体引用的任何其它对象（不同类型）的信息。

对象的排序由 DB2 Cube Views XML 模式定义。在单一操作的作用域内，相同类型（例如，立方体对象）的对象被分组在一起。在这些组内，元素的顺序受相同类型对象之间的引用影响。被引用对象必须出现在引用对象之前。这些组之间的顺序如下所示：

1. attribute
2. join
3. attributeRelationship
4. level
5. cubeLevel
6. hierarchy
7. cubeHierarchy
8. dimension
9. cubeDimension
10. measure
11. facts
12. cubeFacts
13. cubeModel
14. cube

对象类型组的顺序与对象之间的关联无关。属性和连接在与不同对象类型关联时扮演不同角色的事实不影响它们在“关联”格式内的顺序。

有关显示元数据信息的类型和顺序的示例 XML 文件，请参阅随 CVSAMPLE 包括在一起的 XML 元数据文件。

---

## 第 8 章 样本文件

---

### db2mdapiclient 实用程序概述

db2mdapiclient 实用程序以样本源代码形式提供，用于对用于 DB2 Cube Views 多维服务的应用程序进行编码。可使用该实用程序导入、导出和优化元数据对象。

注：随 DB2 Cube Views V8.2 交付的实用程序已增强，所以不能与 `sqllib\samples\olap\client\db2mdapiclient.cpp` 文件中的样本源代码完全匹配。

您可以使用 db2mdapiclient 实用程序来执行 DB2 Cube Views 存储过程 MD\_MESSAGE()（在下表中描述）支持的任何操作：

表 62. 操作. db2mdapiclient 实用程序可执行的多维服务操作

任务	操作
将元数据对象导出至文件	DESCRIBE
从文件导入元数据对象	CREATE 或 IMPORT
更改现有元数据对象	ALTER 或 RENAME
删除现有元数据对象	DROP
验证现有元数据对象是否有效	VALIDATE
优化立方体模型	ADVISE
将 XML 元数据从版本 8.1 迁移至版本 8.2	TRANSLATE

db2mdapiclient 实用程序使用文件来容纳传递至和从 MD\_MESSAGE() 存储过程中接收的 XML。

对于导入，db2mdapiclient 实用程序通常使用 DB2 Cube Views 桥产生的或从 OLAP 中心导出的 XML 文件。对于导出，db2mdapiclient 实用程序产生 DB2 Cube Views 桥实用程序可以使用的 XML 文件以将元数据添加至数据库或 OLAP 工具。用于输入和输出 XML 文件的字符编码很重要。有关字符编码的更多信息，请参阅第 138 页的『代码页支持』。

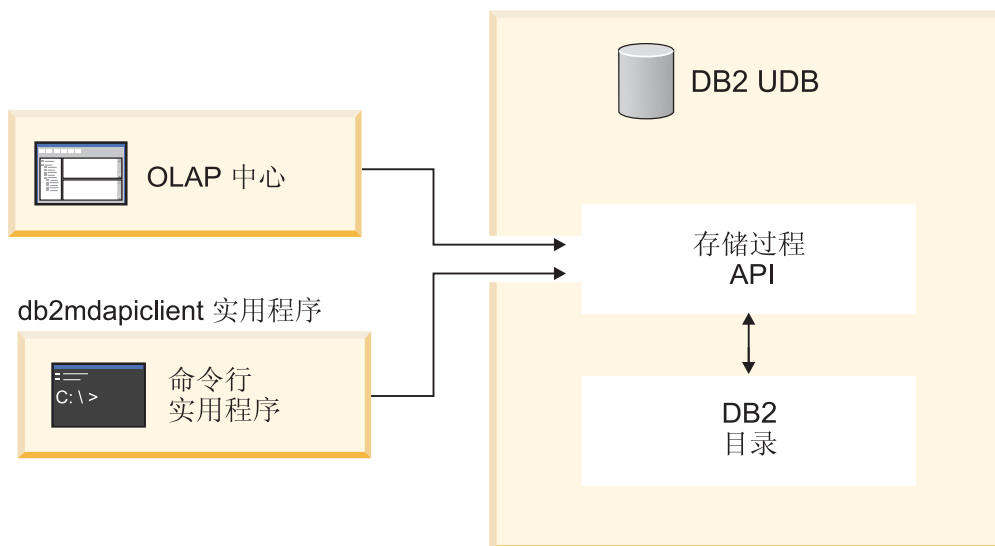


图 42. 元数据传送. db2mdapiclient 实用程序和 OLAP 中心通过多维服务传送元数据

## db2mdapiclient 命令: 处理元数据对象

可从命令行使用 db2mdapiclient 实用程序。

要查看 db2mdapiclient 命令参数的列表，可以在命令行中输入 db2mdapiclient。  
**db2mdapiclient** 命令的语法为:

```
db2mdapiclient -d dbname [-u user] [-p password] -i request.xml -o response.xml
  [-m inputmetadata.xml] [-n outputmetadata.xml] [-a parameter_buffer_size]
  [-b metadata_buffer_size] [-v] [-l] [-h]
```

- d *dbname* 指定数据库的名称。
- u *user* 指定用于连接至指定数据库的用户标识。
- p *password* 指定用于连接至指定数据库的密码。
- i *request.xml* 指定包含要执行的操作的必需输入文件。
- o *response.xml* 指定包含来自 MD\_MESSAGE() 存储过程的响应 XML 的必需输出文件。MD\_MESSAGE() 存储过程中的第三个自变量返回此响应 XML。
- m *inputmetadata.xml* 是包含 DB2 Cube Views 元数据对象 XML 的输入文件。此选项是“创建”或“导入”操作所必需的。
- n *outputmetadata.xml* 是一个可选输出文件，包含来自 MD\_MESSAGE() 存储过程的第二个自变量的响应元数据对象 XML（如果适用的话）。
- a *parameter\_buffer\_size* 指定参数的缓冲区大小。缺省值为 1048576 字节。
- b *metadata\_buffer\_size* 指定元数据对象信息的缓冲区大小。缺省值为 1048576 字节。
- v 指定想要在处理命令时显示的额外信息。
- l 禁用 UTF-8 支持并指定本地和输出文件在本地代码页中。
- h 显示使用信息。



下图显示 MD\_MESSAGE() 存储过程如何与两个输入文件和两个输出文件关联:

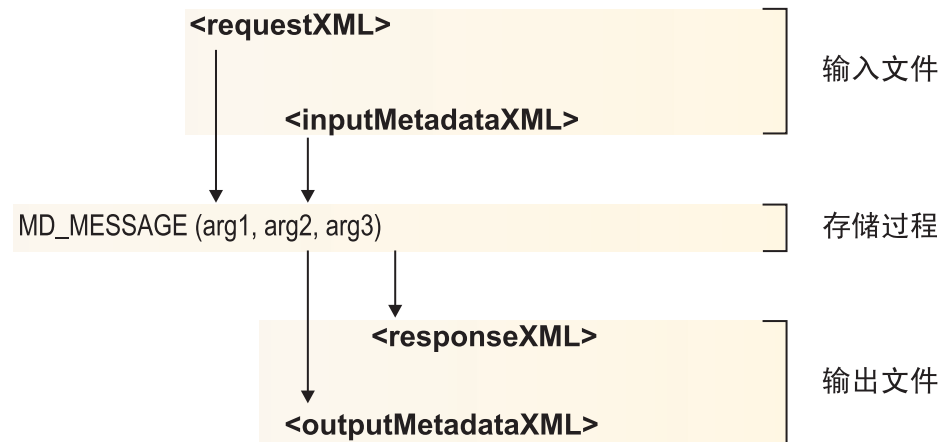


图 43. 存储过程如何处理 db2mdapiclient 实用程序中的两个输入和输出文件

例如, 要导入 Windows CVSAMPLE 数据库的 DB2 Cube Views 元数据对象, 切换至 `..\SQLLIB\samples\olap\xml\input` 目录并输入以下命令:

```
db2mdapiclient -d CVSAMPLE -u db2admin -p mypasswd -i create.xml  
-o myresponse.xml -m ..\..\CVSAMPLE\CVSampleMetadata.xml.xml
```

有关提供的样本文件的描述, 请参阅第 146 页的『API 样本文件』。有关多维服务操作的更多信息, 请参阅第 113 页的『DB2 Cube Views API 概述』。

## 样本数据库文件

下列与 CVSAMPLE 数据库有关的所有文件位于 `\SQLLIB\samples\olap\cvsample\` 目录。

### CVSampleMetadata.xml

包含 CVSAMPLE 元数据的 XML 文件。将此文件与 OLAP 中心和 db2mdapiclient 实用程序配合使用以导入 CVSAMPLE 元数据。

### CVSampleTables.sql

用于填充 CVSAMPLE 表的 SQL 脚本。

### FAMILY.txt、LINE.txt、LOCATION.txt、PRODUCT.txt、SALESFACT.txt、STORE.txt 和 TIME.txt

包含 CVSAMPLE 表数据的一组文本文件。

### CVSampleExplain.sql

可以用来确定 DB2 是否将查询重新路由至摘要表的 SQL 脚本。

`\SQLLIB\samples\olap\xml\input` 目录还包含与 CVSAMPLE 数据库有关的文件。

### Create.xml

包含 CREATE 操作的 XML 文件。使用此文件通过 db2mdapiclient 实用程序装入样本。

---

## API 样本文件

CVSAMPLE 数据库的样本 API 文件是随 DB2 Cube Views 提供的。可将样本文件与 db2mdapiclient 实用程序配合使用来执行样本方案。db2mdapiclient 实用程序是多维服务存储过程接口的瘦包装器。此实用程序以样本源代码形式提供，用于显示如何针对 API 对应用程序进行编码。该源代码位于 \SQLLIB\samples\olap\client\db2mdapiclient.cpp 中。

将作为参数为每个方案列示的样本文件的内容传递至 MD\_MESSAGE() 存储过程。有时存储过程的元数据参数在输入上被忽略，或者在输出上未返回，这在以下方案中被指定为 <empty>。如果不需要输入元数据，则不要在 db2mdapiclient 实用程序命令中指定 -m 选项。如果不需要输出元数据，则不要在 db2mdapiclient 实用程序命令中指定 -n 选项。

有关通过 db2mdapiclient 实用程序使用 MD\_MESSAGE() 存储过程的更多信息，请参阅第 143 页的『db2mdapiclient 实用程序概述』。有关通过过程本身使用 MD\_MESSAGE() 存储过程的更多信息，请参阅第 115 页的『DB2 Cube Views 存储过程』。

所有样本文件位于 \SQLLIB\samples\olap\xml\ 目录中。您可以使用样本文件来执行以下样本方案：

### DROP

使用这些文件来删除元数据目录中的所有元数据对象。该样本假定元数据目录不为空。如果元数据目录为空，则会收到一条找不到操作对象的警告消息。

#### 包含输入参数的文件

- 请求: input\Drop.xml
- 元数据: <empty>

不要在 db2mdapiclient 实用程序中指定 -m 选项。

#### 包含输出参数的文件

- 响应: output\Drop\_response.xml
- 元数据: <empty>

不要在 db2mdapiclient 实用程序中指定 -n 选项。

### CREATE

使用这些文件在元数据目录中创建元数据对象。该样本假定元数据目录为空。

#### 包含输入参数的文件

- 请求: input\Create.xml
- 元数据: input\CVSampleTestMetadata.xml

#### 包含输出参数的文件

- 响应: output\Create\_response.xml
- 元数据: <empty>

不要在 db2mdapiclient 实用程序中指定 -n 选项。

### DESCRIBE

使用这些文件来描述元数据目录中的所有元数据对象。

#### 包含输入参数的文件

- 请求: input\Describe.xml
- 元数据: <empty>

不要在 db2mdapiclient 实用程序中指定 -m 选项。

#### 包含输出参数的文件

- 响应: output\Describe\_response.xml
- 元数据: <metadata XML file>

### DESCRIBE (受限)

使用这些文件以递归方式描述 CVSample.Daily Sale 立方体。此样本假定您先前导入了 CVSAMPLE 的元数据 CVSampleMetadata.xml。

#### 包含输入参数的文件

- 请求: input\CVSampleDescribe\_restricted.xml
- 元数据: <empty>

不要在 db2mdapiclient 实用程序中指定 -m 选项。

#### 包含输出参数的文件

- 响应: output\DescribeRestricted\_response.xml
- 元数据: <metadata XML file>

### ALTER

使用这些文件来改变元数据目录中的元数据对象。此样本假定您先前导入了 CVSAMPLE 的元数据 CVSampleMetadata.xml。

#### 包含输入参数的文件

- 请求: input\Alter.xml
- 元数据: input\CVSampleAlter.xml

#### 包含输出参数的文件

- 响应: output\Alter\_response.xml
- 元数据: <empty>

不要在 db2mdapiclient 实用程序中指定 -n 选项。

### RENAME

使用这些文件来重命名 CVSAMPLE.Sales Model 立方体模型。将立方体模型重命名为 CVSAMPLE.Sales Model (2004)。此样本假定您先前导入了 CVSAMPLE 的元数据 CVSampleMetadata.xml。

#### 包含输入参数的文件

- 请求: input\CVSampleRename.xml
- 元数据: <empty>

不要在 db2mdapiclient 实用程序中指定 -m 选项。

#### 包含输出参数的文件

- 响应: output\Rename\_response.xml

- 元数据: <empty>

不要在 db2mdapiclient 实用程序中指定 -n 选项。

## VALIDATE

使用这些文件通过优化验证方式来验证元数据目录中的所有元数据对象。

### 包含输入参数的文件

- 请求: input\Validate.xml
- 元数据: <empty>

不要在 db2mdapiclient 实用程序中指定 -m 选项。

### 包含输出参数的文件

- 响应: output\Validate\_response.xml
- 元数据: <empty>

不要在 db2mdapiclient 实用程序中指定 -n 选项。

## VALIDATE (受限)

使用这些文件通过优化验证方式来验证 CVSAMPLE.Daily Sale 立方体。此样本假定您先前导入了 CVSAMPLE 的元数据 CVSampleMetadata.xml。

### 包含输入参数的文件

- 请求: input\CVSampleValidate\_restricted.xml
- 元数据: <empty>

不要在 db2mdapiclient 实用程序中指定 -m 选项。

### 包含输出参数的文件

- 响应: output\ValidateRestricted\_response.xml
- 元数据: <empty>

不要在 db2mdapiclient 实用程序中指定 -n 选项。

## IMPORT (以 *create new - ignore collisions* 方式)

使用这些文件通过 *create new - ignore collisions* 导入方式将元数据对象导入元数据目录中。该样本假定元数据目录为空。

### 包含输入参数的文件

- 请求: input\Import\_model.xml
- 元数据: CVSampleTestMetadata.xml

### 包含输出参数的文件

- 响应: output\Import\_model\_response.xml
- 元数据: <empty>

不要在 db2mdapiclient 实用程序中指定 -n 选项。

## IMPORT (以 *create new - replace collisions* 方式)

使用这些文件通过 *create new - replace collisions* 导入方式将元数据对象导入元数据目录中。该样本假定您已经完成 **IMPORT** (以 *create new - ignore collisions* 方式) 方案。

#### 包含输入参数的文件

- 请求: input\Import\_mode2.xml
- 元数据: CVSampleTestMetadata.xml

#### 包含输出参数的文件

- 响应: output\Import\_mode2\_response.xml
- 元数据: <empty>

不要在 db2mdapiclient 实用程序中指定 -n 选项。

### **IMPORT** (以 *create new - abort on collision* 方式)

使用这些文件通过 *create new - abort on collision* 导入方式将元数据对象导入元数据目录中。该样本假定您已经完成了 **IMPORT** (以 *create new - replace collisions* 方式) 方案。

#### 包含输入参数的文件

- 请求: input\Import\_mode3.xml
- 元数据: CVSampleTestMetadata.xml

#### 包含输出参数的文件

- 响应: output\Import\_mode3\_response.xml
- 元数据: <empty>

不要在 db2mdapiclient 实用程序中指定 -n 选项。

### **IMPORT** (以 *create new - report collisions* 方式)

使用这些文件通过 *create new - report collisions* 导入方式将元数据对象导入元数据目录中。该样本假定您已经完成了 **IMPORT** (以 *create new - abort on collision* 方式) 方案。

#### 包含输入参数的文件

- 请求: input\Import\_mode4.xml
- 元数据: CVSampleTestMetadata.xml

#### 包含输出参数的文件

- 响应: output\Import\_mode4\_response.xml
- 元数据: <empty>

不要在 db2mdapiclient 实用程序中指定 -n 选项。

### **TRANSLATE**

使用这些文件来将 Cube Views V8.1 元数据转换为 Cube Views V8.2 元数据。

#### 包含输入参数的文件

- 请求: input\Translate.xml
- 元数据: CVSampleTranslateMetadata.xml

#### 包含输出参数的文件

- 响应: output\Translate\_response.xml
- 元数据: <metadata XML file>

## ADVISE

使用这些文件以对立方体模型 CVSAMPLE.Sales Model 运行优化顾问程序。此样本假定您先前导入了 CVSAMPLE 的元数据 CVSampleMetadata.xml, 并且还未运行先前的重命名方案。

### 包含输入参数的文件

- 请求: input\CVSampleAdvise.xml
- 元数据: <empty>

不要在 db2mdapiclient 实用程序中指定 -m 选项。

### 包含输出参数的文件

- 响应: output\Advise\_response.xml
- 元数据: <empty>

不要在 db2mdapiclient 实用程序中指定 -n 选项。

---

## 附录. 消息

以下消息来自服务器、API 和 DB2 Cube Views 的 OLAP 中心。

**Socket error:** 多次打开和关闭数据库连接可导致套接字错误。在极少数情况下，当您在 Windows 2000 Advanced Server 上的分区环境中一起运行 DB2 Cube Views 与 DB2 通用数据库企业服务器版版本 8.1.2 时，会出现套接字错误。如果您在单个 Windows 会话中快速地重复执行下列步骤的次数超过 10,000 次，则可能会出现该错误：

1. 打开与 DB2 数据库的连接。
2. 调用 DB2 Cube Views 存储过程以执行元数据操作。
3. 关闭数据库连接。

变通方法是重新启动 Windows 工作站以重新激活套接字。

---

## SQLSTATE、API 和其它服务器消息

### API SQL 状态

**01HQ1:** 请参阅输出 XML 和服务日志。

#### 解释

对存储过程的调用已完成，但在执行某个请求的元数据操作过程中检测到错误。

#### 用户响应

检查存储过程输出参数的内容以获取信息。您也可以检查服务器日志中的条目以获取更多信息。

**38Q00:** 有关更多信息，请参阅服务器日志。

#### 解释

对存储过程的调用失败。未执行已请求的一个或多个元数据操作。没有信息通过输出参数从存储过程中返回。

#### 用户响应

检查服务器日志中的条目以获取更多信息。

**38Q01:** 安装路径未知。

#### 解释

对存储过程的调用失败，原因是存储过程进程无法确定 DB2 的安装目录。未执行已请求的一个或多个元数据操作。没有信息通过输出参数从存储过程中返回。

## 用户响应

如果您使用 Windows 操作系统，则确保缺省情况或用户操作为 DB2PATH 环境变量设置了正确的值。重新启动数据库管理器，然后重新发出对存储过程的调用。请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

### **38Q02: 不能打开服务器日志文件。**

#### 解释

对存储过程的调用失败，原因是无法打开存储过程使用的至少一个日志文件以供存储过程进程写入。未执行已请求的一个或多个元数据操作。没有信息通过输出参数从存储过程中返回。

## 用户响应

确保可以创建或打开在存储过程配置文件中指定的日志文件（例如 olap\_config.xml），以供在适当的文件系统上读和写。如果日志文件尚未存在，则存储过程将尝试创建这些文件。在 AIX 上，确保受保护的数据库用户标识能够读和写日志文件。

### **38Q03: 缺少元数据输入参数。**

#### 解释

对存储过程的调用失败，原因是已请求的元数据操作要求将元数据作为输入传递至存储过程，但是未通过输入元数据参数提供元数据。没有信息通过输出参数从存储过程中返回。

## 用户响应

使用输入元数据存储过程参数为下次调用存储过程时请求的元数据操作提供必需的元数据。

### **38Q04: [error\_type] 错误: 响应输出缓冲区太小。**

#### 解释

对存储过程的调用失败，原因是操作响应的输出参数缓冲区太小，无法容纳正在返回的 CLOB 结构。没有信息通过输出参数从存储过程中返回。

## 用户响应

通过对输出响应参数使用较大大小，重新编制存储过程的目录。

## 常规

### 成功代码

**0: 操作成功完成。未遇到错误。:**

#### 解释

请求的元数据操作已成功完成。操作执行过程中未出错。



## 用户响应

此信息仅供参考。无需操作。

**1: 操作完成。返回了附加信息。:**

### 解释

请求的元数据操作已完成。操作已返回可能描述警告或出错情况的其它信息。

## 用户响应

检查 INFO 元素以获取返回的其它信息。

**2: 操作完成。未对元数据作任何更改。:**

### 解释

请求的元数据操作已完成。操作未对数据库目录中的元数据进行更改。

## 用户响应

如果要更改数据库目录中的元数据，请通过使用另一方式重新发出元数据操作请求。

## 常见外部错误 / 警告代码

**100: 未能为 *operation* 分配内存。确保内存可用。:**

### 解释

在执行所请求的元数据操作的过程中，存储过程未能分配必需的内存段。

## 用户响应

增加可用于受防护存储过程进程的内存。

**101: 在处理 *object name* 对象时出现内部错误。:**

### 解释

在执行所请求元数据操作的过程中出现意外内部错误。

## 用户响应

请与 IBM 客户支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**102: 大小为 *buffer\_size* 的输出缓冲区太小。将缓冲区大小更改为至少 *size*。:**

### 解释

可用于存储过程的输出参数缓冲区太小，无法容纳由存储过程生成的 CLOB 结构。

## 用户响应

如果可能，则使用较大大小的 OUT 和 INOUT 参数重新为存储过程编目。否则，必须限制查询以便存储过程返回较少的信息。

**103: 此产品的有效许可证不存在。:**

## 解释

无法执行元数据操作，因为用于该产品安装的有效产品许可证不存在。

## 用户响应

在系统上安装有效的产品许可证，或者与 IBM 软件支持人员或 IBM 软件销售人员联系以购买新的产品许可证。

**104: 发生了内部错误。已返回下列标记: *token0, token1, token2, token3*。:**

## 解释

在执行所请求元数据操作的过程中出现意外内部错误。

## 用户响应

请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**599: 未执行该操作。:**

## 解释

执行此操作之前出错。因此未执行本操作。

## 用户响应

检查以前的元数据操作的结果，这些操作在相同的存储过程调用期间执行。您也可以检查服务器日志中的条目以获取更多信息。更正了引起先前操作失败的问题之后，再次调用存储过程并请求相同的元数据操作。

## 常见内部错误 / 警告代码

**600: 输入 *parameter\_name* 参数无效，并且系统显示此消息: *message*。检查参数并重试。:**

## 解释

作为输入传递至存储过程内部方法的其中一个参数无效。

## 用户响应

请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**601: 输入 *parameter\_name* 参数为 *NULL*。:**

## 解释

作为输入传递至存储过程内部方法的参数具有无效值 *NULL*。

## 用户响应

请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**602: 值为 *value* 的 *parameter\_name* 参数不在 *range\_value1* 至 *range\_value2* 的有效范围内。:**

**解释**

作为输入传递至存储过程内部方法的其中一个参数具有超出有效范围的值。

**用户响应**

请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**603: Unicode 字符串 *string* 是不正确的或无效的。可能存在内存问题。:**

**解释**

存储过程中的 Unicode 字符串不正确或无效。这可能表明系统上或存储过程中存在内存问题。也可能由于存储过程装入了错误版本的 ICU 库引起的。

**用户响应**

确保有足够内存以容纳由存储过程处理的数据量。确保装入要与当前版本的存储过程配合使用的 ICU 库的版本。可能需要检查环境中的运行时库搜索路径设置以确定正确的设置。

请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**604: 未能将字符串 *string* 的内容从 Unicode 转换为活动代码页 *code\_page*。:**

**解释**

将 Unicode 字符串对象转换为使用另一编码的字符串失败。这可能表明系统上或存储过程中存在内存问题。也可能表明系统上存在代码页转换问题，或者存储过程装入了错误版本的 ICU 库。

**用户响应**

确保在数据库服务器系统上安装了必需的 ICU 代码页转换文件。确保有足够内存以容纳由存储过程处理的数据量。确保装入要与当前版本的存储过程配合使用的 ICU 库的版本。可能需要检查环境中的运行时库搜索路径设置以确定正确的设置。

请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**605: 需要释放在 *memory\_buffer* 中分配的内存。:**

**解释**

存储过程的内部方法已返回需要由另一内部方法释放的内存缓冲区。

**用户响应**

存储过程的内部方法应释放已返回的内存缓冲区。请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**606: 为 UChar 将 XMLCh 转换为 UChar 失败。:**

解释

XMLCh 字符和 UChar 字符之间的转换失败。

用户响应

请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**607: 输入缓冲区大小 size 太小。将缓冲区大小更改为至少 new\_size。:**

解释

存储过程的内部内存缓冲区太小，无法容纳必需消息的文本。

用户响应

请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**608: stored\_procedure\_name 的类型在当前上下文中无效。:**

解释

存储过程处理期间遇到意外的类型。

用户响应

与 IBM 软件支持人员联系，通过提供尝试进行的元数据操作的状态标识和文本以获取进一步的帮助。如果可能，请提供数据库服务器中的存储过程日志文件。

**609: data\_type 数据类型在当前上下文中无效。:**

解释

存储过程处理期间遇到意外的数据类型。

用户响应

与 IBM 软件支持人员联系，通过提供尝试进行的元数据操作的状态标识和文本以获取进一步的帮助。如果可能，请提供数据库服务器中的存储过程日志文件。

## MDOBJECT.LIB 错误

**1000: 未能克隆对象 object\_name。:**

解释

克隆存储过程的内部类对象时出错。

用户响应

请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**1001: hierarchy\_name 层次结构的部署不能设置为递归，原因是该层次结构具有两个以上的层。:**

## 解释

层次结构的层过多，无法进行递归部署。

## 用户响应

在将部署更改为递归之前，修改层次结构以使其不超过两个层。

**1002: 不支持被调用的函数 *function\_name*。:**

## 解释

没有为存储过程的某个类实现存储过程的内部虚方法。

## 用户响应

请与 IBM 客户支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**1003: 容器意外地为空。:**

## 解释

存储过程的内部容器结构意外地为空。

## 用户响应

请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**1004: 无法在容器中找到 *object\_name* 对象。:**

## 解释

意外地缺少在存储过程的某个内部容器结构中搜索的对象。

## 用户响应

请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**1005: *element\_name* 元素的副本在容器中已存在。:**

## 解释

已存在不应存在于某个存储过程内部容器结构中的对象。

## 用户响应

请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**1006: 在列表操作期间出现异常。:**

## 解释

在存储过程的其中一种内部列表结构中执行操作时出现意外的异常。

## 用户响应

请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**1007: 在容器中出现内部错误, 错误代码为 *error and number*, 消息为 *message*。:**

### 解释

在存储过程的其中一种内部容器结构中执行操作时出错。

## 用户响应

请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**1008: 复制操作未完全复制所有属性。值为 *value* 的 *property\_name* 属性的复制操作失败。:**

### 解释

在存储过程的其中一个内部对象上执行复制操作时出错。未能复制内部对象的其中一个属性。

## 用户响应

请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**1009: *type1* 的对象类型无效。期望值为 *Type2*。:**

### 解释

存储过程处理期间遇到意外的对象类型。

## 用户响应

请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**1010: *parameter\_name* 参数不具有完整的标识。:**

### 解释

作为输入传递至存储过程内部方法的其中一个参数是不完整的元数据对象标识。

## 用户响应

请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**1011: *object\_name* 对象不具有完整的标识。:**

### 解释

元数据对象标识在存储过程中意外地不完整。

## 用户响应

请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**1012: *parameter\_name* 参数与对象相同。:**

### 解释

作为输入传递至存储结构内部方法的其中一个参数是一个对象，该对象意外地与拥有该方法的对象相同。

## 用户响应

请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**1013: 遇到意外的 *NULL* 指针。:**

### 解释

存储过程处理期间遇到意外的 *NULL* 指针。

## 用户响应

请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**1014: 容器游标到达容器末端。:**

### 解释

存储过程内部其中一个容器结构上的游标已意外地到达容器末端。

## 用户响应

请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**1015: *object\_name* 对象无效。原因标识 =*ID*, 消息 = *message*。:**

### 解释

存储过程的内部元数据对象无效。

## 用户响应

请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**1016: *object\_name* 对象具有类型为 *list\_type* 的列表，而且意外地为空。:**

### 解释

存储过程的内部容器结构意外地为空。

## 用户响应

请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

## MDOPERATION.LIB 错误

**2001: 生成的查询 *query* 不包含必需列 *objectType*。:**

### 解释

存储过程生成的 SQL 查询缺少必需列。

## 用户响应

请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

## MDPARSER.LIB 错误

**3001: 解析器在 *operation* 期间遇到 XML 异常, 消息为 *message*。:**

### 解释

在分析传递至存储过程的 XML 时, 存储过程中出现意外的异常。

## 用户响应

确保传递至存储过程的 XML 格式良好, 且对于为本产品发布的 XML 模式是有效的。请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能, 同时提供数据库服务器中的存储过程日志文件。

**3002: 在 *operation* 中遇到意外的解析器异常。:**

### 解释

在分析传递至存储过程的 XML 时, 存储过程中出现意外的异常。

## 用户响应

确保传递至存储过程的 XML 格式良好, 且对于为本产品发布的 XML 模式是有效的。请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能, 同时提供数据库服务器中的存储过程日志文件。

**3003: 解析器在 *operation* 期间遇到 *SAXParseException*, 消息为 *message*。:**

### 解释

在分析传递至存储过程的 XML 时, 存储过程中出现意外的异常。

## 用户响应

确保传递至存储过程的 XML 格式良好, 且对于为本产品发布的 XML 模式是有效的。请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能, 同时提供数据库服务器中的存储过程日志文件。

**3004: 系统未能获取 *operation* 的解析器错误消息。:**



## 解释

在分析传递至存储过程的 XML 时，存储过程中出现意外错误。无法检索 XML 解析器中的错误消息。

## 用户响应

确保传递至存储过程的 XML 格式良好，且对于为本产品发布的 XML 模式是有效的。请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**3100: 系统未能对 *parameter type* 的 XML 进行语法分析 (行: *line*, 字符: *character*, 消息: *message*)。:**

## 解释

存储过程无法分析输入 XML。输入 XML 可能格式不良好，或者对于为本产品发布的 XML 模式无效。

## 用户响应

确保传递至存储过程的 XML 格式良好，且对于为本产品发布的 XML 模式是有效的。

**3101: 遇到未知元数据对象。 *parser\_message*。:**

## 解释

传递至存储过程的 XML 中存在未知类型的元数据对象。存储过程无法处理此输入 XML。

## 用户响应

确保传递至存储过程的 XML 格式良好，且对于为本产品发布的 XML 模式是有效的。

**3102: 遇到未知 XML 属性。 *attribute\_name*, *attribute\_value*。:**

## 解释

传递至存储过程的 XML 中存在未知类型的 XML 属性。存储过程无法处理此输入 XML。

## 用户响应

确保传递至存储过程的 XML 格式良好，且对于为本产品发布的 XML 模式是有效的。

**3103: 对于名称为 *name* 且值为 *value* 的属性，处理程序遇到无效的枚举值。:**

## 解释

传递至存储过程的 XML 中存在无效的枚举值。存储过程无法处理此输入 XML。

## 用户响应

确保传递至存储过程的 XML 格式良好，且对于为本产品发布的 XML 模式是有效的。

**3500: 必须为属性或元素名称 *name* 提供数据。:**

## 解释

存储过程未能为指定的 XML 属性或 XML 中的元素（将由存储结构返回）设置值。

## 用户响应

请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**3501: 属性 *attribute\_name1* 和 *attribute\_name2* 需要数据。:**

## 解释

存储过程未能为指定的 XML 属性或 XML 中的元素（将由存储结构返回）设置值。

## 用户响应

请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**3502: 对于名称为 *name* 且值为 *value* 的属性，格式化程序遇到无效的枚举值。:**

## 解释

在格式化将由存储过程返回的 XML 时，存储过程中出现无效的枚举值。

## 用户响应

请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

## MDDATABASE.LIB 错误

**4000: 数据库连接失败。数据库名称 *database\_name*, 用户名 *user\_name*。:**

## 解释

存储过程未能建立自己与数据库的连接。

## 用户响应

确保存储过程使用的用户标识具有用于连接至数据库的适当特权。

**4001: 数据库连接由于连接已存在而未发出。:**

## 解释

存储过程意外地遇到了重复的与数据库的内部连接。请求的元数据操作未能完成。

## 用户响应

请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**4002: 数据库操作失败。:**

## 解释

执行由存储过程向数据库发出的 SQL 语句时出错。

## 用户响应

请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**4003: 执行 CLI 调用 *call\_name* 失败。:**

### 解释

执行指定的 CLI 调用时出错。

## 用户响应

检查客户机和服务器上的数据库管理器日志文件。请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**4004: 返回的数据被截断。:**

### 解释

在失败的数据库操作期间返回的诊断信息被截断。

## 用户响应

检查客户机和服务器上的数据库管理器日志文件。请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**4005: 从数据库接收到警告。SQLSTATE = code, 消息 = message。:**

### 解释

存储过程发出的 CLI 调用返回警告信息。

## 用户响应

检查客户机和服务器上的数据库管理器日志文件。

**4008: 遇到未知的 DB2 数据类型。:**

### 解释

在执行数据库请求时，存储过程遇到未知数据类型。

## 用户响应

检查客户机和服务器上的数据库管理器日志文件。请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**4009: 未生成有效的保存点名称。:**

### 解释

存储过程无法生成有效的数据库事务保存点名称。存储过程使用其数据库应用程序标识来构成保存点名称。

## 用户响应

重新调用存储过程。重新调用可能会为存储过程生成新的数据库应用程序标识，因此可能允许生成有效的保存点名称。请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**4010: 试图设置 DB2 保存点失败。:**

### 解释

存储过程无法设置数据库事务保存点。当前事务中已存在与此存储过程实例所使用的保存点同名的保存点。

## 用户响应

如果可能，释放当前事务的保存点，然后重新调用存储过程。您也可以通过使用新的事务来重新调用存储过程。

请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**4011: 在此执行点之前未设置保存点。:**

### 解释

在存储过程的某一点中意外地缺少事务保存点。存储过程可能未设置缺少的保存点，或者在存储过程外部完成的数据库操作可能已释放保存点。

## 用户响应

重新调用存储过程。请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**4012: 存在无效的保存点字符串存储。:**

### 解释

未将数据库事务保存点名称正确地存储在存储过程内部的数据结构中，可能的原因是没有足够的内存可用于存储过程进程。

## 用户响应

重新调用存储过程。如果问题仍然存在，则增加可用于受防护存储过程进程的内存。请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**4013: 未能清除保存点。:**

### 解释

存储过程无法清除数据库事务保存点。存储过程可能未设置缺少的保存点，或者在存储过程外部完成的数据库操作已释放保存点。

## 用户响应

重新调用存储过程。请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**4014: 尝试确定 DB2 AUTOCOMMIT 设置失败。:**

### 解释

存储过程确定 DB2 AUTOCOMMIT 功能是启用还是禁用的尝试失败。

## 用户响应

检查客户机和服务器上的数据库管理器日志文件。请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**4015: 尝试设置 DB2 AUTOCOMMIT OFF 失败。:**

### 解释

存储过程禁用 DB2 AUTOCOMMIT 功能的尝试失败。

## 用户响应

检查客户机和服务器上的数据库管理器日志文件。请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**4016: CLI 调用 SQLFetch() 未返回数据。:**

### 解释

CLI 函数 SQLFetch() 未将数据返回至存储过程。这可能是可接受的，但存储过程不能允许这种错误在未更改的情况下在整个存储过程中传播。

## 用户响应

检查客户机和服务器上的数据库管理器日志文件。请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**4017: 未正确构造 Object\_name 对象。:**

### 解释

未正确初始化存储过程的内部数据库对象。

## 用户响应

请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**4018: 数据库断开连接失败。:**

### 解释

存储过程未能断开与数据库的连接。

## 用户响应

检查客户机和服务器上的数据库管理器日志文件。请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**4019: DB2 SQL 错误 - SQLCODE sqlcode, SQLSTATE sqlstate, SQLMESG sqlmesg. :**

### 解释

执行由存储过程向数据库发出的 SQL 语句时出错。

## 用户响应

检查客户机和服务器上的数据库管理器日志文件。请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**4020: DB2 SQL 错误 - 没有可用的详细信息. :**

### 解释

对于执行由存储过程向数据库发出的 SQL 语句时出现的错误，诊断信息不可用。

## 用户响应

检查客户机和服务器上的数据库管理器日志文件。请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**4021: DB2 SQL 错误 - 没有可用的详细信息. :**

### 解释

试图收集执行 SQL 语句（由存储过程向数据库发出）时出现的另一错误的诊断信息时出错。

## 用户响应

检查客户机和服务器上的数据库管理器日志文件。请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**4022: 分配 DB2 handle\_name 句柄失败. :**

### 解释

试图在存储过程中分配 DB2 句柄时出错。

## 用户响应

检查客户机和服务器上的数据库管理器日志文件。请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**4023: 释放 DB2 handle\_name 句柄失败. :**

## 解释

试图在存储过程中释放 DB2 句柄时出错。

## 用户响应

检查客户机和服务器上的数据库管理器日志文件。请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**4028: 未停止事务。:**

## 解释

试图结束存储过程的事务时出错。

## 用户响应

检查客户机和服务器上的数据库管理器日志文件。请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**4029: 在主对象表中发现共享同一名称和模式的重复行。:**

## 解释

在某个元数据目录表中意外地找到共享同一名称和模式的重复行。此共享表示存储过程中存在内部错误。

## 用户响应

请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**4030: 未初始化 DBINFO 结构。确保存储过程是通过使用 DBINFO 选项在数据库中创建的。:**

## 解释

存储过程未从数据库客户机中接收 DBINFO 结构。

## 用户响应

确保已通过使用 DBINFO 选项将存储过程编目在适当的数据库中。

**4031: 将模式设置为 DB2INFO 失败。:**

## 解释

存储过程未能将 DB2INFO 设置为当前模式。

## 用户响应

检查客户机和服务器上的数据库管理器日志文件。重新调用存储过程。

**4032: 由于主对象表中的对象与正在插入的对象之间存在冲突，因此操作失败。:**

## 解释

存储过程中的 SQL INSERT 语句失败，因为它会导致在某个元数据目录表中产生重复的元数据对象条目。

## 用户响应

请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**4033: 列函数的操作数包括列函数。:**

## 解释

在存储过程发出的其中一个 SQL 语句中检测到嵌套在另一个列函数中的列函数。SQL 语句中不能嵌套列函数。

## 用户响应

修改输入属性或量度对象的 SQL 表达式模板，以便存储过程生成的 SQL 语句中不再出现嵌套的列函数。

**4034: 未确定 DB2 ISOLATION LEVEL 设置。:**

## 解释

存储过程确定数据库事务隔离级别的尝试失败。无法确定隔离级别。

## 用户响应

检查客户机和服务器上的数据库管理器日志文件。请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**4035: 将 DB2 ISOLATION LEVEL 设置为 READ STABILITY 失败。:**

## 解释

存储过程设置数据库事务隔离级别的尝试失败。存储过程要求隔离级别为 READ STABILITY。

## 用户响应

检查客户机和服务器上的数据库管理器日志文件。请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**4036: 当前安装的 DB2 通用数据库的版本未能确定。:**

## 解释

存储过程确定数据库管理器版本级别的尝试失败。

## 用户响应

检查客户机和服务器上的数据库管理器日志文件。请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。



**4037: 当前安装的 DB2 通用数据库版本 *version\_number* 修订包 *number* 与此版本的 DB2 Cube Views 不兼容。升级 DB2 通用数据库的版本。:**

**解释**

当前安装的 DB2 通用数据库版本与此版本的 DB2 Cube Views 不兼容。升级 DB2 通用数据库的版本以使其位于 DB2 Cube Views 版本级别或之上。

**用户响应**

确保在同一服务器上安装的 DB2 通用数据库的版本与 DB2 Cube Views 的版本是兼容的。参阅安装和设置文档以获取更多信息。

请与 IBM 软件支持人员联系以获得进一步帮助。提供尝试进行的元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**4038: SQL 语句太长或太复杂，系统无法对其进行处理。:**

**解释**

系统无法处理的存储过程发出语句，原因是它超出系统在长度或复杂性方面的限制，或者因为它涉及太多的约束或触发器。

**用户响应**

请与 IBM 软件支持人员联系以获得进一步帮助。提供尝试进行的元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**4039: 必需的 SYSINFOVERSION 元数据表在 DB2 目录表中不存在。迁移元数据。:**

**解释**

必需的 SYSINFOVERSION 元数据表在 DB2 目录表中不存在。迁移元数据。

**用户响应**

通过迁移元数据来更新 DB2 目录表。有关迁移的信息，请参阅《DB2 Cube Views 指南和参考》一书。与 IBM 软件支持人员联系并提供尝试进行的元数据操作的状态标识和文本以获取其它帮助。如果可能，请提供数据库服务器中的存储过程日志文件。

**4040: 元数据表版本与 DB2 Cube Views API 的当前版本不匹配。元数据表的版本为 *version\_number1*，而当前 API 版本为 *version\_number2*。需要将元数据表迁移至当前 API 版本。:**

**解释**

SYSINFOVERSION 表中的 VERSION 值与您尝试用来执行操作的 DB2 Cube Views API 的版本不匹配。您需要迁移元数据表并确保 SYSINFOVERSION 表中具有正确的 VERSION 值。

**用户响应**

迁移 DB2 目录中的元数据表。有关迁移的信息，请参阅《DB2 Cube Views 指南和参考》。与 IBM 软件支持人员联系并提供尝试进行的元数据操作的状态标识和文本以获取进一步的帮助。如果可能，请提供数据库服务器中的存储过程日志文件。

**4041: 未指定元数据表的版本。需要将元数据表迁移至当前 API 版本。:**

**解释**

在 SYSINFOVERSION 表中不存在 VERSION 值。需要运行 db2mdmigrate.sql 脚本以迁移 DB2 目录中的元数据表。

**用户响应**

迁移现有元数据表。有关迁移的信息，请参阅《DB2 Cube Views 指南和参考》。与 IBM 软件支持人员联系并提供尝试进行的元数据操作的状态标识和文本以获取进一步的帮助。如果可能，请提供数据库服务器中的存储过程日志文件。

**4042: 指定了多个元数据表版本或指定了重复的版本。检查元数据表以确保指定了唯一的正确的版本。:**

**解释**

在 SYSINFOVERSION 目录表中指定了多个元数据表版本或指定了重复的版本。仅应在 SYSINFOVERSION 表中指定一个正确的版本。

**用户响应**

请与 IBM 软件支持人员联系以获得进一步帮助。如果可能，请提供数据库服务器中的存储过程日志文件。

## **MDUTILITY.LIB 错误**

**5000: 实用程序未能对字符串 *string* 进行语法分析。:**

**解释**

在对内部字符串值进行语法分析时，存储过程的内部方法遇到错误。

**用户响应**

请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**5001: 实用程序编号格式未成功初始化。确保有足够内存可用。:**

**解释**

未能在存储过程中正确初始化 ICU 编号格式化程序。这可能是由于存储过程进程没有足够的可用内存资源引起的。也可能由于存储过程装入了错误版本的 ICU 库引起的。

**用户响应**

增加可用于受防护存储过程进程的内存，并重新调用存储过程。确保装入要与当前版本的存储过程配合使用的 ICU 库的版本。可能需要检查环境中的运行时库搜索路径设置以确定正确的设置。

请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**5002: 实用程序资源束未成功初始化。错误代码 = *code*。确保该资源束存在且位于路径 *path\_name* 中。:**

## 解释

未能在存储过程中正确初始化 ICU 资源束。不正确的初始化可能会导致以下问题：内存资源不足以供存储过程进程使用；存储过程装入错误版本的 ICU 库；或者为存储过程装入错误的资源束。

## 用户响应

增加可用于受防护存储过程进程的内存，并重新调用存储过程。确保装入要与当前版本的存储过程配合使用的 ICU 库的版本。可能需要检查环境中的运行时库搜索路径设置以确定正确的设置。确保在数据库服务器系统上安装了正确版本的存储过程的资源束。

请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**5003: 未找到环境变量 `variable_name` 中的数据路径。检查是否正确地设置了该环境变量。:**

## 解释

未设置存储过程使用的 DB2 环境变量。

## 用户响应

确保在系统上正确地安装了 DB2。请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**5004: 目标流已关闭。:**

## 解释

存储过程内部使用的数据流意外关闭。数据库系统上可能没有可用的足够文件句柄。

## 用户响应

确保操作系统中有足够的可用文件句柄。请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**5005: 目标正在通过使用缺省编码写入字符。:**

## 解释

为存储过程提供的缺省编码正由存储过程使用，以将数据写入数据库服务器文件系统上的文件。

## 用户响应

读取文件（由存储过程写入）的应用程序必须能够解释使用存储过程缺省编码进行编码的数据。

**5006: 未写入输入日志字符串 `string`。字符串的级别为 `string_level`，而日志的级别为 `log_level`。:**

## 解释

当前记录级别不允许将指定的消息写入为存储过程设置的某个日志文件中。

## 用户响应

如果需要将指定的消息写入存储过程的某个日志文件中，请修改记录级别。

**5007: 未找到错误代码 *code* 的消息正文。:**

## 解释

在存储过程的资源束文件中未找到所示错误代码的文本。可能在使用错误版本的资源束文件。

## 用户响应

确保在数据库服务器系统上安装了正确版本的存储过程的资源束文件。请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**5008: 访问全局静态 *MsgBase* 对象的 *operation* 时失败。:**

## 解释

试图访问内部消息对象时存储过程中出错。

## 用户响应

请与 IBM 软件支持人员联系以获得进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

## 外部 API 错误

### 6000-6199

**6000: *OLAPMSG()* 失败，错误代码为代码。:**

## 解释

存储过程在执行过程中失败。

## 用户响应

根据返回码，请解决问题并重新调用存储过程，或与 IBM 软件支持人员联系以获得进一步帮助。如果与 IBM 软件支持人员联系，请提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**6001: *object\_name* 对象的实例化 SQL 模板无效，值为 *value*。原因标识 = *ID*，消息 *message*。:**

## 解释

实例化 SQL 模板是 SQL 语句代码段，该代码段可通过组合复合属性或复合量度中涉及的所有属性和量度的 SQL 表达式模板形成。指定对象的实例化 SQL 模板存在问题。

## 用户响应

请参阅指定的原因标识和消息。在完成原因消息建议的任何更改之后，重新调用存储过程。

**6002: object1 对象引用 object2 对象, 但 object2 对象在数据库中不存在。:**

### 解释

仅当其它对象在数据库中存在时，数据库对象才可以引用它们。

## 用户响应

在数据库中创建要引用的对象，然后重新发出元数据操作请求。或者除去对缺少对象的引用，然后重新发出元数据操作请求。

**6003: 未能打开指定路径中的 log\_name 日志。确保指定路径存在且文件具有写访问权。:**

### 解释

无法打开存储过程使用的至少一个日志文件。

## 用户响应

确存储过程配置文件中指定的路径已存在。确保在数据库服务器上运行存储过程的用户标识具有创建、读取和写入所需日志文件的权限。

**6005: 输入元数据参数对于此操作意外地为空。缺少的元数据参数对于此操作是必需的。:**

### 解释

请求的元数据操作要求提供元数据作为输入。用于交换元数据的存储过程参数意外地为空。

## 用户响应

使用要求的元数据重新发出元数据操作请求。

**6006: 找不到与搜索条件搜索条件相匹配的对象。:**

### 解释

元数据操作找不到与指定搜索条件相匹配的元数据对象。未更改元数据目录中的内容。

## 用户响应

如果要更改元数据目录的内容，请通过新的搜索条件重新发出元数据操作。

**6007: 目录中的对象和正被导入的对象之间发生冲突。未对元数据作任何更改。:**

### 解释

正被导入的对象和元数据目录中已存在的对象之间检测到冲突。归因于指定的导入方式，因此未更改元数据目录中的对象。

## 用户响应

如果要更改元数据目录的内容，请通过使用其它导入方式重新发出元数据操作。

**6008: 具有标识标识的元数据输入中存在重复对象。:**

### 解释

在此元数据操作的元数据输入中检测到重复的元数据对象。不允许将重复的对象作为元数据操作的输入。

## 用户响应

从输入元数据中除去重复的元数据对象，并且重新发出元数据操作。

**6009: 与输入 *object\_name* 对象共享同一标识的对象已在元数据目录中存在。:**

### 解释

无法执行元数据操作，因为具有相同标识的元数据对象在元数据目录中已存在。

## 用户响应

在重新发生失败的元数据操作之前，从与所创建对象共享同一标识的元数据目录中删除对象。或者，您可以改变现有对象以与要创建的新对象的属性相匹配。否则，您必须将导致此错误的新对象排除在要执行的元数据操作之外。

**6010: 输入 *object\_name* 对象已存在对 *object\_name* 对象的引用。:**

### 解释

指定对象之间的引用在元数据中目录中已定义。不允许重复引用。

## 用户响应

从元数据操作请求中除去某个重复的引用。

**6011: 无法使用重命名操作来更改 *object\_name* 对象的模式。:**

### 解释

无法使用重命名操作来更改元数据对象的模式。

## 用户响应

确保为正在对其进行重命名的对象指定的模式保持不变，或使用改变操作。

**6013: 服务器上的 API 不支持客户机所使用的 XML 模式的版本 *version1*。服务器上的 API 支持 XML 模式的版本 *version2*。:**

### 解释

服务器上存储过程的版本不支持客户机所使用的并且嵌入输入参数字符串中的 XML 模式的版本。

## 用户响应

确保客户机应用程序和存储过程使用随本产品发布的同一版本的 XML 模式。

**6014: 无法用公式表示 *object\_name* 对象的 SQL 模板。原因标识 ID, 消息 *message*。:**

**解释**

存储过程通过组合所有属性和量度（包括在复合属性或复合量度中）的 SQL 表达式模板来用公式表示属性和量度的 SQL 模板。在用公式表示 SQL 模板的过程中，存储过程中出现问题。

**用户响应**

请参阅指定的原因标识和消息。在完成原因消息建议的任何更改之后，重新调用存储过程。

**6015: 数据库用户标识没有在活动数据库中创建数据库模式的权限。:**

**解释**

在数据库服务器上拥有存储过程进程的用户标识没有在活动数据库中创建数据库模式的权限。数据库模式是为每个唯一的元数据对象模式创建的。

**用户响应**

检查客户机和服务器上的数据库管理器日志文件。将在活动数据库中创建模式的权限授予拥有存储过程进程的用户标识。重新调用存储过程。

**6016: 数据库用户标识没有在活动数据库中执行所需操作的权限。从数据库服务器中返回以下错误消息: *message*。:**

**解释**

在数据库服务器上拥有存储过程进程的用户标识没有在活动数据库中执行所需操作的权限。

**用户响应**

检查客户机和服务器上的数据库管理器日志文件。将所需的权限授予拥有存储过程进程的用户标识。重新调用存储过程。

**6017: *object\_name* 对象在元数据目录中不存在。:**

**解释**

请求的操作要求元数据目录中存在指定的对象。

**用户响应**

重新发出元数据操作请求之前，在元数据目录中创建指定的对象。

**6018: 所需的表在数据库中不存在。从数据库服务器中返回以下错误消息: *message*。:**

**解释**

请求的操作所需的表在数据库中不存在。

## 用户响应

如果缺少的表是用户表，则创建表并重新发出元数据操作请求。如果缺少的表是元数据目录表或数据库系统表，则与 IBM 软件支持人员联系以获取进一步帮助。提供所尝试元数据操作的状态标识和文本。如果可能，同时提供数据库服务器中的存储过程日志文件。

**6019: API 操作使用版本 *version\_number* 的 XML 而元数据使用版本 *version\_number* 的 XML。API 操作和元数据 XML 版本必须相匹配。:**

### 解释

API 操作 XML 的版本必须与元数据 XML 的版本相匹配。

## 用户响应

确保 API 操作 XML 和元数据 XML 使用同一版本号。

**6020: *operation\_name* API 操作不支持版本 *version\_number* 的元数据 XML。:**

### 解释

对于 TRANSLATE 操作，API 操作 XML 必须为版本 8.2.0.1.0，而元数据 XML 必须为 8.1.2.1.0。

## 用户响应

有关元数据规则、元数据验证和查询优化的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6021: 仅可对 DESCRIBE 操作使用 *version\_number* XML。:**

### 解释

仅可对 DESCRIBE 操作使用指定的 XML 版本。

## 用户响应

有关元数据规则、元数据验证和查询优化的更多信息，请参阅《DB2 Cube Views 指南和参考》。

## 验证错误

**6200: *object\_name* 对象不完整。确保设置了所需的属性。:**

### 解释

指定的输入对象不具有所需的全部属性设置。

## 用户响应

为给定对象设置必需的属性并重新发出元数据操作请求。

**6201: *hierarchy\_name* 层次结构无效，因为它是递归类型但却没有恰好引用两个属性。:**



## 解释

递归层次结构必须恰好引用两个属性。标识的层次结构违反此规则。

## 用户响应

修改标识的层次结构以恰好引用两个属性。有关元数据规则的更多信息，请参阅《安装与用户指南》。

**6202: *object\_name* 对象必须至少具有一个 SQL 模板。:**

## 解释

依据产品文档中提供的元数据对象定义，标识的量度必须至少为其定义一个 SQL 模板。

## 用户响应

修改标识的量度以使其至少具有一个定义的 SQL 模板。有关元数据规则的更多信息，请参阅《安装与用户指南》。

**6206: *attribute\_name* 属性应该只具有一个 SQL 模板。:**

## 解释

依据产品文档中提供的元数据对象定义，标识的量度必须仅具有一个定义的 SQL 模板。

## 用户响应

修改标识的量度以使其仅具有一个定义的 SQL 模板。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6207: *attribute\_name* 属性是连接的一部分，但不具有列引用。:**

## 解释

标识的属性对象必须为其引用数据库列，以供元数据连接对象正确地引用。

## 用户响应

修改标识的属性对象以使其引用数据库列，或者在另一个属性对象引用数据库列的情况下，修改相关的连接对象以使其引用另一个属性对象。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6208: *attribute\_name* 属性是连接的一部分，它必须指向与 *table\_name* 相同的表。:**

## 解释

第一个标识的属性对象必须引用与另一个标识的对象相同的数据库表。

## 用户响应

修改第一个标识的属性以使其引用与标识的另一个对象相同的数据库表，或者在另一个属性对象引用与标识的另一个对象相同的数据库表的情况下，修改相关的连接对象以便其引用另一个属性对象。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6209: *object\_name* 对象的模式超出最大长度。:**

## 解释

标识的对象的模式超出最大长度。

## 用户响应

缩短标识的对象的模式。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6210: object\_name 对象的名称超出最大长度。:**

## 解释

标识的对象的名称超出最大长度。

## 用户响应

缩短标识的对象的名称。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6211: object\_name 对象的表名超出最大长度。:**

## 解释

标识的对象的表名超出最大长度。

## 用户响应

缩短表的名称。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6212: object\_name 对象的业务名超出最大长度。:**

## 解释

标识的对象的业务名超出最大长度。

## 用户响应

缩短业务名。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6213: object\_name 对象的注释超出最大长度。:**

## 解释

标识的对象的注释超出最大长度。

## 用户响应

缩短标识的对象的注释。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6214: object\_name 对象的模式不能以 SYS 开头。:**

## 解释

元数据对象的模式不能以字符串 SYS 开头。

## 用户响应

为元数据对象使用不以 *SYS* 开头的模式。有关元数据规则的更多信息，请参阅《*DB2 Cube Views 指南和参考*》。

**6215: *object\_name* 对象的模式不能为 *SESSION*。:**

### 解释

元数据对象的模式不能是字符串 *SESSION*。

## 用户响应

为元数据对象使用不是字符串 *SESSION* 的模式。有关元数据规则的更多信息，请参阅《*DB2 Cube Views 指南和参考*》。

**6216: *object\_name* 对象的名称和模式不完整。原因标识 = *ID*, 消息= *message*。:**

### 解释

标识的对象的名称和 / 或模式缺少或无效。

## 用户响应

为标识的对象的名称和模式提供有效的字符串。有关元数据规则的更多信息，请参阅《*DB2 Cube Views 指南和参考*》。

**6217: *hierarchy\_name* 立方体层次结构无效，原因是它引用了 *hierarchy\_name* 层次结构未引用的层。:**

### 解释

该立方体层次结构引用其父层次结构未引用的层。

## 用户响应

改变立方体层次结构以使它只引用同时被父层次结构引用的层。或者，可改变层次结构以使它引用立方体层次结构引用的相同层。有关元数据规则的更多信息，请参阅《*DB2 Cube Views 指南和参考*》。

## 警告代码

**6250: *API* 不能为 *level\_name* 层对象创建功能相关性，原因是 *level\_name* 层键属性未映射至一个表列。:**

### 解释

*API* 不能为层对象创建功能相关性，原因是层键属性未映射至单个表列。

## 用户响应

有关功能相关性的更多信息，请参阅《*DB2 Cube Views 指南和参考*》。

**6251: *API* 将不为 *level\_name* 层对象创建功能相关性，原因是层键属性对应于维表的现有唯一约束。:**

## 解释

API 将不为该层对象创建功能相关性，原因是层键属性对应于维表的现有唯一约束。功能相关性是冗余的。

## 用户响应

有关功能相关性的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6252: API 不能为 *level\_name* 层对象创建功能相关性，原因是 *level\_name* 层键属性可空。:**

## 解释

如果任何层键属性可空，API 就不能为层对象创建功能相关性。

## 用户响应

有关功能相关性的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6253: API 不能为 *level\_name* 层对象创建功能相关性，原因是这些属性跨越多个表。:**

## 解释

如果层键属性、缺省属性和相关属性跨越多个表，API 就不能为层对象创建功能相关性。

## 用户响应

有关功能相关性的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6254: *level\_name* 层对象的功能相关性排除了 *attribute\_name* 缺省属性或相关属性，原因是该属性已经作为层键属性包括。:**

## 解释

层对象的功能相关性排除了已经作为层键属性包括的相关属性。

## 用户响应

有关功能相关性的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6255: *level\_name* 层对象的功能相关性排除了 *attribute\_name* 缺省属性或相关属性，原因是该缺省属性或相关属性未映射至一个表列。:**

## 解释

层对象的功能相关性排除了缺省属性或相关属性，原因是属性未映射至一个表列。”

## 用户响应

有关功能相关性的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6256: API 不能为 *level\_name* 层对象创建功能相关性，原因是所有缺省和相关属性已被排除。:**

## 解释

至少需要一个缺省属性或相关属性以创建对应于层对象的功能相关性。

## 用户响应

有关功能相关性的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6257: API 不能为 *level\_name* 层对象创建功能相关性，原因是尝试创建功能相关性时发生了错误。:**

## 解释

当 API 尝试执行创建功能相关性的 SQL 语句时发生了错误。

## 用户响应

有关功能相关性的更多信息，请参阅《DB2 Cube Views 指南和参考》。您也可以检查服务器日志中的条目以获取更多信息。

**6258: API 不能改变或删除 *level\_name* 层对象，原因是尝试删除相关联的 *dependency\_name* 功能相关性时发生了错误。:**

## 解释

DB2 Cube Views 不能改变或删除层对象，原因是 API 尝试发出删除功能相关性的 SQL 语句时发生了错误。确保您具有创建或删除功能相关性的权限。

## 用户响应

有关功能相关性的更多信息，请参阅《DB2 Cube Views 指南和参考》。检查服务器日志中的条目以获取更多信息。

**6299: 在验证期间至少找到一个数据库视图。未对涉及视图列的连接执行与约束相关的验证检查。已执行所有其它验证检查。:**

## 解释

未对被发现涉及视图列的那些连接执行与约束相关的验证检查。对所有其它请求的连接执行与约束相关的验证检查，并且对所有请求的对象执行所有剩余的验证检查。

## 用户响应

有关元数据规则、元数据验证和查询优化的更多信息，请参阅《DB2 Cube Views 指南和参考》。

## 与规则有关的验证错误

**6300: *model\_name* 立方体模型未引用一个或多个事实。:**

## 解释

标识的立方体模型对象违反了元数据对象规则。立方体模型必须引用一个或多个事实。

## 用户响应

改变标识的立方体模型以使其引用一个或多个事实。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6301: *model\_name* 立方体模型未引用零个或多个维。:**

### 解释

标识的立方体模型对象违反了元数据对象规则。立方体模型必须引用零个或多个维。

## 用户响应

更改标识的立方体模型以使其引用零个或多个维。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6302: *model\_name* 立方体模型的其中一个维-连接对缺少维和/或连接。:**

### 解释

标识的立方体模型对象违反了元数据对象规则。立方体模型的维-连接对必须同时引用维和连接。

## 用户响应

更改标识的立方体模型以使其所有的维-连接对同时引用维和连接。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6303: *model\_name* 立方体模型引用的 *join\_name* 连接无效。其中一侧的所有属性必须由 *facts\_name* 事实引用，另一侧的所有属性必须由立方体模型的其中一个维引用。:**

### 解释

标识的立方体模型对象违反了元数据对象规则。每个立方体模型的连接必须引用某一侧的立方体模型事实的属性，并引用另一侧的其中一个立方体模型维的属性。

## 用户响应

更改标识的立方体模型的无效连接，以便连接某一侧的所有属性来自立方体模型的事实，而连接另一侧的所有属性来自其中一个立方体模型维。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6304: *measure\_name* 量度中的其中一个聚集直接引用 *dimension\_name* 维，而 *model\_name* 立方体模型不直接引用此维。:**

### 解释

标识的立方体模型对象违反了元数据对象规则。立方体模型使用的量度中的聚集只能引用由同一立方体模型使用的那些维。

## 用户响应

更改所标识量度的聚集以使其仅引用由所标识立方体模型使用的那些维。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6305: *measure\_name* 量度中的空维列表聚集与 *model\_name* 立方体模型中至少一个先前不匹配的维不匹配。:**

## 解释

标识的立方体模型对象违反了元数据对象规则。立方体模型所使用量度中的空维列表聚集必须至少与每个立方体模型中的一个不匹配维相匹配。

## 用户响应

更改用于所标识量度的聚集，以使它的空维列表与标识的立方体模型中至少一个先前不匹配的维相匹配。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6306: *measure\_name* 量度必须仅包含空维列表聚集，因为 *model\_name* 立方体模型未引用任何维对象。:**

## 解释

标识的立方体模型对象违反了元数据对象规则。当立方体模型未引用任何维时，立方体模型的量度必须仅包含空维列表聚集。

## 用户响应

更改标识的量度以使其仅包含空维列表聚集。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6307: *cube\_name* 立方体未引用一个立方体事实对象。:**

## 解释

标识的立方体对象违反了元数据对象规则。立方体必须引用一个立方体事实对象。

## 用户响应

更改标识的立方体以使其引用一个立方体事实对象。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6308: *cube\_name* 立方体未至少引用一个立方体维对象。:**

## 解释

标识的立方体对象违反了元数据对象规则。立方体必须至少引用一个立方体维对象。

## 用户响应

更改标识的立方体以使其至少引用一个立方体维对象。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6309: *cube\_name* 立方体引用的 *cube\_facts\_name* 立方体事实不是从由 *model\_name* 立方体模型引用的事实对象中派生的。:**

## 解释

标识的立方体对象违反了元数据对象规则。标识立方体使用的立方体事实必须从标识立方体模型使用的事实中派生。

## 用户响应

更改一个或多个标识的对象，以便不再违反指定的规则。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6310: cube\_name** 立方体引用的 **cube\_dimension\_name** 立方体维不是从 **model\_name** 立方体模型引用的其中一个维对象中派生的。:

解释

标识的立方体对象违反了元数据对象规则。标识立方体使用的立方体维必须从标识立方体模型使用的其中一个维中派生。

用户响应

更改一个或多个标识的对象，以便不再违反指定的规则。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6311: facts\_name** 事实对象未引用任何量度。:

解释

标识的事实对象违反了元数据对象规则。事实对象必须至少引用一个量度。

用户响应

更改标识的事实以使其至少引用一个量度。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6312: 无法使用事实对象连接来连接事实对象 facts\_name** 引用的某些属性和量度。:

解释

标识的事实对象违反了元数据对象规则。事实对象的属性和量度必须全部是可以使用事实的连接对象连接的。

用户响应

通过从事实对象中引用更多连接对象，使标识的事实对象引用的所有属性和量度成为可连接。或者，从事实对象或量度中除去通过事实当前连接不可连接的那些属性。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6313: facts\_name** 事实对象在两个表之间具有多个连接。:

解释

标识的事实对象违反了元数据对象规则。事实对象不能在相同的两个表之间具有多个连接。

用户响应

改变标识的事实对象，以使它在任何两个给定的表之间仅具有一个连接。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6314: facts\_name** 事实对象包含连接循环。:

解释

标识的事实对象违反了元数据对象规则。标识的事实对象的连接形成路径循环。这是不允许的。



## 用户响应

从标识的事实对象中除去导致循环的某个连接，或者更改导致循环的某个连接以使循环不再存在。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6315: join\_name** 连接没有仅引用 **facts\_name** 事实对象中的那些属性。:

### 解释

标识的事实对象违反了元数据对象规则。事实对象的连接必须仅引用事实对象的属性。

## 用户响应

更改标识的连接以使其仅引用标识事实对象的属性，或者将缺少的属性（由标识的连接对象引用）添加至事实对象。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6316: cube\_facts\_name** 立方体事实对象未引用事实对象或引用了多个事实对象。:

### 解释

标识的立方体事实对象违反了元数据对象规则。立方体事实对象必须引用一个事实对象。

## 用户响应

更改标识的立方体事实对象以使其引用一个事实对象。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6317: cube\_facts\_name** 立方体事实对象未引用任何量度。:

### 解释

标识的立方体事实对象违反了元数据对象规则。立方体事实对象必须至少引用一个量度。

## 用户响应

更改标识的立方体事实对象以使其至少引用一个量度。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6318: cube\_facts\_name** 立方体事实对象引用的 **measure\_name** 量度不是 **facts\_name** 事实对象的一部分。:

### 解释

标识的立方体事实对象违反了元数据对象规则。立方体事实对象必须引用由事实对象所引用的量度，而立方体事实对象是从该事实对象中派生的。

## 用户响应

将标识的量度添加至标识的事实对象，或者从标识的立方体事实对象中除去标识的量度。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6319: dimension\_name** 维未引用任何属性。维必须至少引用一个属性。:

## 解释

标识的维对象违反了元数据对象规则。维对象必须至少引用一个属性。

## 用户响应

更改标识的维对象以使其至少引用一个属性。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6320: 无法使用维连接来连接 *dimension\_name* 维引用的某些属性。:**

## 解释

标识的维对象违反了元数据对象规则。维对象的属性必须全部是可以通过使用维的连接对象连接的。

## 用户响应

通过从维对象中引用更多连接对象，使标识的维对象所引用的全部属性成为可连接的。或者从维对象中除去那些通过该维的当前连接不可连接的属性。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6321: *dimension\_name* 维包含连接循环。:**

## 解释

标识的维对象违反了元数据对象规则。标识的维对象的连接形成路径循环。这是不允许的。

## 用户响应

从标识的维对象中除去导致循环的某个连接，或者更改导致循环的某个连接以使循环不再存在。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6322: *dimension\_name* 维在两个表之间具有多个连接。:**

## 解释

标识的维对象违反了元数据对象规则。维对象不能在相同的两个表之间具有多个连接。

## 用户响应

改变标识的维对象，以使它在任何两个给定的表之间仅具有一个连接。有关元数据规则的更多信息，请参阅产品文档。

**6323: *hierarchy\_name* 层次结构引用了未被层次结构父维 *hierarchy\_name* 引用的层。:**

## 解释

维的层次结构仅可引用该维对象的层。此层次结构引用父维未引用的层。

## 用户响应

改变层次结构以使它仅引用其父维对象的层，或者将层次结构引用的层添加至维对象。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6325: 维的连接必须仅引用该维的属性。join\_name 连接引用不在 dimension\_name 维中的属性。:**

**解释**

维和连接违反了以下元数据对象规则: 维的连接只能引用该维的属性。

**用户响应**

改变连接以使其仅引用父维的属性, 或者将连接引用的属性添加至维。有关元数据规则的更多信息, 请参阅《DB2 Cube Views 指南和参考》。

**6326: cube\_dimension\_name 立方体维未引用维。:**

**解释**

标识的立方体维对象违反了元数据对象规则。立方体维对象必须引用维。

**用户响应**

更改标识的立方体维对象以使其引用维。有关元数据规则的更多信息, 请参阅《DB2 Cube Views 指南和参考》。

**6327: cube\_dimension\_name 立方体维未引用立方体层次结构。:**

**解释**

标识的立方体维对象违反了元数据对象规则。立方体维对象必须引用立方体层次结构。

**用户响应**

更改标识的立方体维对象以使其引用立方体层次结构。有关元数据规则的更多信息, 请参阅《DB2 Cube Views 指南和参考》。

**6328: cube\_dimension\_name 立方体维引用的 cube\_hierarchy\_name 立方体层次结构不是从 dimension\_name 维引用的任何层次结构中派生的。:**

**解释**

标识的立方体维对象违反了元数据对象规则。标识的立方体维使用的立方体层次结构必须从标识的维使用的层次结构中派生。

**用户响应**

更改一个或多个标识的对象, 以便不再违反指定的规则。有关元数据规则的更多信息, 请参阅《DB2 Cube Views 指南和参考》。

**6329: hierarchy\_name 层次结构未引用任何层。:**

**解释**

层次结构对象必须至少引用一个层。

**用户响应**

改变标识的层次结构对象以使其至少引用一个层。有关元数据规则的更多信息, 请参阅《DB2 Cube Views 指南和参考》。

**6330: 使用递归部署的 *hierarchy\_name* 层次结构没有恰好引用两个层。:**

解释

标识的层次结构对象违反了元数据对象规则。使用递归部署的层次结构对象必须引用两个层。

用户响应

改变标识的层次结构对象以使其引用两个层。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6332: *hierarchy\_name* 层次结构的类型与其部署不兼容。:**

解释

标识的层次结构对象违反了元数据对象规则。层次结构类型与部署的兼容性将在产品文档中描述。

用户响应

更改标识的层次结构以使其类型与其部署兼容。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6334: *hierarchy\_name* 立方体层次结构必须恰好引用一个层次结构。:**

解释

立方体层次结构违反了以下元数据对象规则：立方体层次结构必须恰好引用一个层次结构。

用户响应

改变标识的立方体层次结构以使其引用一个层次结构。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6335: *hierarchy\_name* 立方体层次结构必须至少引用一个立方体层。:**

解释

标识的立方体层次结构对象违反了元数据对象规则。立方体层次结构对象必须至少引用一个立方体层。

用户响应

改变标识的立方体层次结构对象以使其至少引用一个立方体层。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6336: *hierarchy\_name1* 立方体层次结构引用了立方体层，但相应的 *hierarchy\_name2* 层次结构未引用相应的 *level\_name* 层。必须将 *level\_name* 层添加至 *hierarchy\_name2* 层次结构，或者从 *hierarchy\_name1* 立方体层次结构中除去相应的立方体层。:**

解释

立方体层次结构违反了以下元数据对象规则：立方体层次结构必须引用从相应层次结构引用的层派生的立方体层。

## 用户响应

将该层添加至层次结构，或者从立方体层次结构中除去该层。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6337: hierarchy\_name** 立方体层次结构中的立方体层的顺序与 **hierarchy\_name** 层次结构中的相应层的顺序不匹配。:

### 解释

标识的立方体层次结构对象违反了元数据对象规则。立方体层次结构中立方体层的相关顺序必须与从中派生该立方体层次结构的层次结构中相同层的相关顺序相同。

## 用户响应

改变某个标识的对象，以便两个标识对象的属性的相关顺序一致。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6340: measure\_name** 量度的其中一个 **SQL** 表达式模板使用的参数不是属性、量度或列。:

### 解释

标识的量度对象违反了元数据对象规则。量度对象的 **SQL** 表达式模板使用的参数必须是属性、量度或列。

## 用户响应

更改标识的量度，以使其 **SQL** 表达式模板使用属性、量度或列作为参数。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6341: measure\_name** 量度的 **SQL** 表达式模板中用作参数的属性或量度之间存在相关性循环。:

### 解释

标识的量度对象违反了元数据对象规则。量度的 **SQL** 表达式模板中用作参数的属性和量度不能形成相关性循环。

## 用户响应

更改标识的量度，以使其 **SQL** 表达式模板不包含涉及参数的相关性循环。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6342: measure\_name** 量度具有为它的其中一个 **SQL** 表达式模板定义的空字符串。:

### 解释

标识的量度对象违反了元数据对象规则。量度的 **SQL** 表达式模板不能为空字符串。

## 用户响应

改变标识的量度以使其 **SQL** 表达式模板不再是空字符串。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6343: measure\_name** 量度的 **SQL** 表达式模板包含聚集函数。:

## 解释

标识的量度对象违反了元数据对象规则。量度的 SQL 表达式模板不能包含聚集函数。

## 用户响应

更改标识的量度以使其 SQL 表达式模板不再包含聚集函数。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6344: *measure\_name* 量度缺少聚集或未正确地引用除量度之外的对象。:**

## 解释

标识的量度对象违反了元数据对象规则。如果量度至少引用了一个其它量度并且只引用了量度，则量度不要求聚集。

## 用户响应

通过添加聚集或确保标识的量度至少引用一个其它量度且仅引用量度来更改标识的量度。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6345: *measure\_name* 量度中的 SQL 表达式模板数与用于第一个聚集函数的参数数量不匹配。:**

## 解释

标识的量度对象违反了元数据对象规则。量度中的 SQL 模板数必须与该量度第一个聚集函数的参数数量相匹配（如果存在聚集）。

## 用户响应

更改标识的量度，以使其第一个聚集函数的参数数量与量度中的 SQL 表达式模板数相匹配。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6346: 具有多个 SQL 表达式模板的 *measure\_name* 量度在聚集脚本中未至少定义一个步骤。:**

## 解释

标识的量度对象违反了元数据对象规则。具有多个 SQL 表达式模板的量度必须在其聚集脚本中至少定义一个步骤。

## 用户响应

更改标识的量度以使其聚集脚本至少具有一个步骤。或者除去量度的其中一个 SQL 表达式模板（只要剩余的 SQL 表达式模板仅引用其它量度）。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6347: *measure\_name1* 量度已定义了聚集脚本。但是，它不应定义任何聚集脚本，原因是被引用的量度 *measure\_name2* 为 SQL 表达式定义了多个模板。:**

## 解释

标识的量度对象违反了元数据对象规则。如果量度 A 引用量度 B（它定义多个 SQL 模板），则量度 A 不能具有聚集脚本。此规则适用于量度引用树中的所有级别。

## 用户响应

从导致问题的量度中除去聚集脚本，或从被引用的量度中除去某个 SQL 表达式模板。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6348: *measure\_name* 量度包含未用作第一个聚集的多参数聚集函数。:**

### 解释

标识的量度对象违反了元数据对象规则。多参数聚集函数只能用作量度的第一个聚集。

## 用户响应

将多参数聚集函数作为标识的量度所使用的第一个聚集，或者从标识的量度中除去多参数聚集函数。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6349: *measure\_name* 量度没有恰好具有一个空维列表聚集。:**

### 解释

标识的量度对象违反了元数据对象规则。当量度定义一个或多个聚集时，某个聚集必须具有空的维列表。

## 用户响应

更改标识的量度以使其拥有一个空的维列表，或者更改标识的量度以使其不定义聚集。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6350: *dimension\_name* 维在 *measure\_name* 量度中被多次引用。:**

### 解释

标识的量度对象违反了元数据对象规则。在量度中，不论是在聚集中还是跨多个聚集，引用维的次数都不能超过一次。

## 用户响应

改变标识的量度以使其仅引用标识的维一次。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6351: *object\_name* 对象的 SQL 表达式模板缺少编号为 *number* 的标记指示符。**

**标记指示符必须从编号 1 开始连续编号。:**

### 解释

标识的量度对象违反了元数据对象规则。在量度的 SQL 表达式模板中，标记指示符必须从 1 开始连续编号。

## 用户响应

更改标识的量度，以使其 SQL 表达式模板的标记指示符从 1 开始连续编号。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6352: *measure\_name* 量度包含未使用提供的引用 *reference* 的 SQL 表达式模板。:**

## 解释

标识的量度对象违反了元数据对象规则。量度的 SQL 表达式模板必须使用提供的每个列、属性和量度引用。每个引用可使用多次。

## 用户响应

更改所标识量度的 SQL 表达式模板，以使其使用提供的每个列、属性和量度引用。或者除去未被标识的量度 SQL 表达式模板使用的列、属性和量度引用。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6353: attribute\_name** 属性的其中一个 SQL 表达式模板使用的参数不是属性或列。:

## 解释

标识的属性对象违反了元数据对象规则。属性对象的 SQL 表达式模板使用的参数必须是属性或列。

## 用户响应

更改标识的属性，以使其 SQL 表达式模板使用属性或列作为参数。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6354: attribute\_name** 属性的 SQL 表达式模板中用作参数的属性之间存在相关性循环。:

## 解释

标识的属性对象违反了元数据对象规则。为属性的 SQL 表达式模板，那些用作参数的属性不能形成相关性循环。

## 用户响应

改变标识的属性，以使其 SQL 表达式模板不包含涉及参数的相关性循环。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6355: attribute\_name** 属性在它的其中一个 SQL 表达式模板中定义了空字符串。:

## 解释

标识的属性对象违反了元数据对象规则。属性的 SQL 表达式模板不能是空字符串。

## 用户响应

更改标识的属性以使其 SQL 表达式模板不再是空字符串。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6356: attribute\_name** 属性的 SQL 表达式模板包含聚集函数。:

## 解释

标识的属性对象违反了元数据对象规则。属性的 SQL 表达式模板不能包含聚集函数。



## 用户响应

更改标识的属性以使其 SQL 表达式模板不再包含聚集函数。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6358: attribute\_name** 属性包含未使用提供的引用 *reference* 的 SQL 表达式模板。:

## 解释

标识的属性对象违反了元数据对象规则。属性的 SQL 表达式模板必须使用提供的每个列引用和属性引用。每个引用可使用多次。

## 用户响应

更改所标识属性的 SQL 表达式模板，以使其使用提供的每个列引用和属性引用。或者除去那些未由所标识属性 SQL 表达式模板使用的列引用和属性引用。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6359: relationship\_name** 属性关系未引用两个单值属性。:

## 解释

标识的属性关系对象违反了元数据对象规则。属性关系对象必须引用两个单值属性。

## 用户响应

更改标识的属性关系对象以使其引用两个单值属性。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6360: 未正确定义 relationship\_name** 属性关系。基数属性设置为 *N:N*，但功能相关性属性设置为“是”。:

## 解释

标识的属性关系对象违反了元数据对象规则。当属性关系的功能相关性属性设置为“是”时，属性关系的基数属性不能设置为 *N:N*。

## 用户响应

更改标识的属性关系，以使其基数未设置为 *N:N* 或使其功能相关性属性设置为“否”。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6361: join\_name** 连接未至少引用一个三元组。三元组包含左边属性、右边属性和运算符。:

## 解释

标识的连接对象违反了元数据对象规则。连接对象必须至少引用一个三元组（包含左边属性、右边属性和运算符）。

## 用户响应

更改标识的连接对象以使其至少引用一个三元组。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6362: join\_name** 连接中左边属性未全部解析为单个表的一个或多个列。:

## 解释

标识的连接对象违反了元数据对象规则。必须将连接左边属性全部解析为单个数据库表的一个或多个列。

## 用户响应

更改标识的连接对象，以将其左边属性全部解析为单个表的一个或多个列。或者更改标识的连接对象的左边属性，以使它们完全符合声明的元数据规则。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6363: join\_name** 连接中右边属性未全部解析为单个表的一个或多个列。:

## 解释

标识的连接对象违反了元数据对象规则。必须将连接右边属性全部解析为单个数据库表的一个或多个列。

## 用户响应

更改标识的连接对象，以将其右边属性全部解析为单个表中的一个或多个列。或者更改标识的连接对象右边属性，以使它们完全符合声明的元数据规则。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6364: join\_name** 连接的至少一个三元组未定义有效的运算。左边的属性和右边的属性的数据类型可能不互相兼容，或它们可能与运算符不兼容。:

## 解释

标识的连接对象违反了元数据对象规则。连接对象的每个三元组必须定义一个有效的运算。考虑到指定的运算，右边和左边属性的数据类型必须互相兼容。

## 用户响应

更改标识的连接对象以使其每个三元组均定义一个有效运算。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6365: model\_name** 立方体模型并非引用一个且仅一个事实对象。:

## 解释

标识的立方体模型对象违反了元数据对象规则。完整的立方体模型必须引用一个事实对象。

## 用户响应

更改标识的立方体模型对象以使其引用一个事实对象。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6366: model\_name** 立方体模型未引用一个或多个维。:

## 解释

标识的立方体模型对象违反了元数据对象规则。完整的立方体模型必须至少引用一个维对象。

## 用户响应

更改标识的立方体模型对象以使其至少引用一个维对象。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6367: join\_name 事实-到-维连接的基数未设置为 1:1 或 N:1。:**

### 解释

立方体模型不会从优化顾问程序建议中受益，因为事实-到-维连接的基数不是 1:1 或 N:1。将不执行优化。

## 用户响应

为了使立方体模型从优化顾问程序建议中受益，每个从事实-到-维对象的连接的基数必须设置为 1:1 或 N:1。事实属性上连接的基数必须为 1 或 N，而维属性的基数必须为 1。有关优化规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6368: join\_name 事实-到-维连接未将 facts\_name 事实对象的表连接至 dimension\_name 维的主表。:**

### 解释

标识的连接对象违反了优化规则。考虑到由维连接形成的连接网络，您必须至少拥有一个表（主表），从该表中发出的所有连接均具有基数 N:1 或 1:1。在立方体模型中，从事实到维对象的连接必须包括维的这个主表。

## 用户响应

在立方体模型对象中，确保所有事实-到-维连接均是从事实对象至每个维的主表。有关优化规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6369: dimension\_name 维不具有主表，如该维的连接形成的连接网络所示。:**

### 解释

标识的维对象违反了优化规则。考虑到由维连接形成的连接网络，您必须至少拥有一个表，从该表中发出的所有连接均具有基数 N:1 或 1:1。如果维没有这样一个主表，则不会执行优化。

## 用户响应

检查在维中使用的连接对象的基数。要执行优化，维必须具有一个主表（如优化规则中所述）。有关优化规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6370: join\_name 连接涉及未定义引用约束的列。:**

### 解释

标识的连接对象违反了优化规则。必须有对参与连接的列定义的约束。如果连接为自连接（即等式两侧使用了相同的列集），则必须定义与列集相匹配的主键。在所有其它情况下，当连接一侧的列集不同于另一侧的列集时，主键必须与连接一侧的列相匹配，外键必须与其它列集相匹配并引用主键。由于缺少约束，因此将不执行优化。

## 用户响应

在参与连接的列上创建约束。如果出于性能原因而不需要标准约束，请创建启用查询优化的信息性约束。有关优化规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6371: 未使用 *join\_name* 自连接中涉及的列来定义主键。:**

### 解释

标识的连接对象违反了优化规则。您必须在参与连接的列上定义约束。如果连接为自连接（即等式两侧使用了相同的列集），则必须定义与列集相匹配的主键。由于缺少约束，因此将不执行优化。

## 用户响应

如果表已定义了主键，请将自连接的属性设置为表示表的主键列的属性。否则，在参与自连接的列上创建主键。有关优化规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6372: 未使用 *join\_name* 连接一侧中的列来定义主键。:**

### 解释

标识的连接对象违反了优化规则。您必须定义已在参与连接的列上定义的约束。当连接一侧的列集不同于另一侧的列集时，主键必须与连接一侧的列相匹配，外键必须与其它列集相匹配并引用主键。由于缺少约束，因此将不执行优化。

## 用户响应

在连接某一侧的列上创建主键。有关优化规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6373: 未使用 *join\_name* 连接一侧中的列来定义外键。:**

### 解释

标识的连接对象违反了优化规则。您必须定义已在参与连接的列上定义的约束。当连接一侧的列集不同于另一侧的列集时，主键必须与连接一侧的列相匹配，外键必须与其它列集相匹配并引用主键。由于缺少约束，因此将不执行优化。

## 用户响应

在连接的主键列和连接另一侧的列之间创建外键约束。如果出于性能原因而不需要标准约束，请创建启用查询优化的信息性约束。有关优化规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6374: 使用 *join\_name* 连接一侧的列的外键未引用使用连接另一侧的列的主键。:**

### 解释

标识的连接对象违反了优化规则。您必须在参与连接的列上定义约束。当连接一侧的列集不同于另一侧的列集时，主键必须与连接一侧的列相匹配，外键必须与其它列集相匹配并引用主键。由于缺少约束，因此将不执行优化。

## 用户响应

在连接的主键列和连接另一侧的列之间创建外键约束。如果出于性能原因而不需要标准约束，请创建启用查询优化的信息性约束。有关优化规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6375: join\_name 连接的基数未设置为 1:1、N:1 或 1:N。:**

### 解释

标识的连接对象违反了优化规则。如果连接基数为 M:N，则无法执行优化。

## 用户响应

根据连接所基于的约束，将连接基数设置为 1:1、1:N 或 N:1。有关优化规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6376: join\_name 自连接的基数未设置为 1:1。:**

### 解释

标识的连接对象违反了优化规则。如果自连接的连接基数未设置为 1:1，则无法执行优化。

## 用户响应

将自连接的基数设置为 1:1。有关优化规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6377: 对于在其上定义了主键的一侧，join\_name 连接的基数未设置为 1。:**

### 解释

标识的连接对象违反了优化规则。连接基数在定义主键的一侧必须为 1，在定义外键的一侧必须为 N。如果外键一侧也定义了主键，则必须使用 1 作为基数。如果不是这种情况，则无法执行优化。

## 用户响应

在定义主键的一侧，连接基数应设置为 1。有关优化规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6378: 对于在其上定义了外键的一侧，join\_name 连接的基数未设置为 N。:**

### 解释

标识的连接对象违反了优化规则。连接基数在定义主键的一侧必须为 1，在定义外键的一侧必须为 N。如果外键一侧也定义了主键，则必须使用 1 作为基数。如果不是这种情况，则无法执行优化。

## 用户响应

在定义外键的一侧，连接基数应设置为 N。有关优化规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6379: 对于在其上定义了主键和外键的一侧，join\_name 连接的基数未设置为 1。:**

## 解释

标识的连接对象违反了优化规则。连接基数在定义主键的一侧必须为 1，在同时定义主键和外键的一侧也必须为 1。如果不是这种情况，则无法执行优化。

## 用户响应

连接基数应设置为 1:1。有关优化规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6380: join\_name 连接引用的 attribute\_name 属性未解析为非可空 SQL 表达式。:**

## 解释

标识的连接对象违反了优化规则。连接中使用的所有属性必须解析为非可空 SQL 表达式。如果连接引用解析为可空 SQL 表达式的属性，则无法执行优化。

## 用户响应

从连接中除去对可空属性的引用。有关优化规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6381: join\_name 连接不具有 INNER JOIN 类型。:**

## 解释

标识的连接对象违反了优化规则。连接类型必须设置为 INNER JOIN。无法执行优化。

## 用户响应

将连接更改为只引用解析为单个列的属性。有关优化规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6382: join\_name 连接的 attribute\_name 属性引用未解析为单个列表表达式，属性引用加入约束时需要此表达式。:**

## 解释

标识的连接对象违反了优化规则。必须将 DB2 约束应用于连接引用的属性。约束只能应用于列，因此连接引用的属性必须解析为表中的单个列。如果不是这种情况，则无法执行优化。

## 用户响应

将连接更改为只引用解析为单个列的属性。有关优化规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6385: model\_name 立方体模型必须至少引用具有层次结构的一个维。:**

## 解释

标识的立方体模型违反了以下优化规则：立方体模型需要至少引用一个具有层次结构的维。

## 用户响应

改变标识的立方体模型的维以使该维至少引用一个层次结构。有关优化规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6386:** 对于 **cube\_name** 立方体中的每个立方体维，每个优化片必须恰好定义有一个优化级别。:

### 解释

立方体不满足以下规则：对于立方体中的每个立方体维，优化片必须恰好有一个优化级别。

## 用户响应

改变优化片以使其对立方体对象中的每个立方体维引用一个优化级别。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6387:** 该优化级别必须引用 **cube\_name** 立方体中的一个立方体维。:

### 解释

立方体不满足以下元数据对象规则：优化级别必须恰好引用属于该立方体的一个立方体维。

## 用户响应

改变优化级别以使其引用立方体对象中的一个立方体维。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6388:** 该优化级别必须引用 **cube\_name** 立方体中的一个立方体层次结构。:

### 解释

立方体不满足以下元数据对象规则：优化级别必须恰好引用属于该立方体的一个立方体层次结构。

## 用户响应

改变优化级别以使其引用立方体对象中的一个立方体维和一个立方体层次结构。

**6389:** 必须将该优化级别设置为 **allLevel**、**anyLevel** 或 **cube\_name** 立方体中的立方体层引用。:

### 解释

立方体不满足以下元数据对象规则：优化级别必须具有 **allLevel**、**anyLevel** 或立方体层引用。

## 用户响应

改变标识的优化级别以使其引用立方体对象中的 **allLevel**、**anyLevel** 或立方体层引用。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6390:** 在优化级别中，被引用立方体维和立方体层次结构必须从 **cube\_name** 立方体中的对象派生。该立方体层次结构必须属于该立方体维。:

## 解释

该立方体不满足以下元数据对象规则：对于该优化级别，被引用立方体维和立方体层次结构必须从立方体中的对象派生。该立方体层次结构必须属于该立方体维。

## 用户响应

改变优化级别以使被引用立方体维和立方体层次结构从立方体中的对象派生。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6391:** 在优化级别中，如果立方体层引用不是 *anyLevel* 或 *allLevel*，则 *level\_name* 立方体层必须属于 *cube\_name* 立方体中的 *hierarchy\_name* 立方体层次结构。:

## 解释

该立方体不满足以下元数据对象规则：在优化级别中，如果该立方体层引用不是 *anyLevel* 或 *allLevel*，则立方体层必须属于立方体层次结构。

## 用户响应

改变优化级别以使被引用立方体层属于立方体层次结构。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6392:** 在 *cube\_name* 立方体中不能同时具有 *MOLAP* 抽取优化片和混合抽取优化片。:

## 解释

该立方体不满足以下元数据对象规则：立方体不能同时具有 *MOLAP* 抽取优化片和混合抽取优化片。

## 用户响应

改变其中一个优化片以使一个立方体中不会同时具有 *MOLAP* 抽取优化片和混合抽取优化片。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6393:** 在 *cube\_name* 立方体中不能有多个 *MOLAP* 抽取优化片。:

## 解释

该立方体不满足以下元数据对象规则：立方体最多只能有一个 *MOLAP* 抽取类型的优化片。

## 用户响应

改变立方体以具有零个或一个 *MOLAP* 抽取类型优化片。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6394:** 在 *cube\_name* 立方体中不能有多个混合抽取优化片。:

## 解释

该立方体不满足以下元数据对象规则：立方体最多只能有一个混合抽取类型的优化片。



## 用户响应

改变立方体以具有零个或一个混合抽取类型优化片。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6395:** 仅当 *cube\_name* 立方体中存在混合抽取优化片时，深入钻取优化片才能存在。:

### 解释

该立方体不满足以下元数据对象规则：仅当立方体中存在混合抽取优化片时，深入钻取优化片才能存在。

## 用户响应

如果立方体中没有混合抽取优化片，则将优化片类型从深入钻取改变为另一类型。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6396:** 被 *dimension\_name* 维中的 *level\_name* 层引用的 *attribute\_name* 属性必须包括在维属性列表中。:

### 解释

该维不满足以下元数据对象规则：被维中的层引用的所有属性必须包括在该维的属性列表中。

## 用户响应

改变该层以使其不引用标识的属性，或者将标识的属性添加至该维。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6397:** *level\_name* 层必须至少引用一个层键属性。:

### 解释

该层不满足以下元数据对象规则：一个层必须至少引用一个层键属性。

## 用户响应

改变该层以使其至少引用一个层键属性。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6398:** *level\_name* 层在层键属性集合中不能多次包含 *attribute\_name* 属性。:

### 解释

该层不满足以下元数据对象规则：层键属性集合不能包含重复的属性。

## 用户响应

改变该层以使其不包括重复的层键属性。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6399:** *level\_name* 层必须恰好引用一个缺省属性。:

### 解释

该层不满足以下元数据对象规则：一个层必须恰好引用一个缺省属性。

## 用户响应

改变该层以使其恰好包括一个缺省属性。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

### 6400-6499

**6401:** 不能将一个属性同时用作 *level\_name* 层中的缺省属性和相关属性。:  
解释

该层不满足以下元数据对象规则：一个属性不能同时用作缺省属性和相关属性。

## 用户响应

改变该层以使其缺省属性未被用作相关属性。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6402:** 不能将一个属性同时用作 *level\_name* 层中的缺省属性和相关属性。:  
解释

该层不满足以下元数据对象规则：相关属性集合不能包含重复的属性。

## 用户响应

改变该层以使其不包括重复的相关属性。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6403:** *level\_name* 层在相关属性集合中不能多次包含 *attribute\_name* 属性。:  
解释

该层不满足以下元数据对象规则：一个立方体层必须恰好引用一个层。

## 用户响应

改变该立方体层以使其恰好引用一个层。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6404:** *level\_name* 立方体层必须恰好引用一个层。:  
解释

该立方体层不满足以下元数据对象规则：立方体层中的所有相关属性还必须是相应层的相关属性。

## 用户响应

改变该立方体层以使其引用同时被父层引用的属性。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6405:** *level\_name* 立方体层在相关属性集合中不能多次包含 *attribute\_name* 属性。:

## 解释

该立方体层不满足以下元数据对象规则：相关属性集合不能包含重复的属性。

## 用户响应

改变该立方体层以使其不包含重复的相关属性。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

## 引用约束相关错误

**6500:** 因为 *attribute\_name* 属性或量度的 SQL 模板仍然包含对其它属性、量度或列的引用，所以无法执行此操作。在执行此操作之前必须删除这些引用。:

### 解释

无法执行请求的操作，原因是它将违反元数据目录中元数据对象之间的引用约束。标识的对象的 SQL 表达式模板包含对其它属性、量度或列的引用，必须在执行此操作之前将它们标识的对象中除去。

## 用户响应

在删除标识的对象之前，请更改标识的对象以使其 SQL 表达式模板不再引用属性、量度或列。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6501:** 因为另一个属性或量度引用 *attribute\_name* 属性或量度，所以无法执行此操作。:

### 解释

无法执行请求的操作，原因是它将违反元数据目录中元数据对象之间的引用约束。标识的属性或量度当前由另一属性或量度引用，因此无法删除标识的属性或量度。

## 用户响应

在删除标识的属性和量度之前，请更改引用对象以使它们不再引用标识的属性或量度。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6502:** 因为量度中定义的聚集引用 *dimension\_name* 维，所以无法执行此操作。:

### 解释

无法执行请求的操作，原因是它将违反元数据目录中元数据对象之间的引用约束。标识的维当前由量度的聚集引用，因此无法删除标识的维。

## 用户响应

在删除标识的维之前，请更改引用对象以使它们不再引用标识的维。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6503:** 无法对 *object\_name* 对象执行该操作。立方体层次结构必须引用已由层次结构（用于派生立方体层次结构）引用的属性。:

### 解释

无法执行请求的操作，原因是它将违反元数据目录中元数据对象之间的引用约束。如果出现以下其中一种情况，则会出现问题:

- 在要除去的属性仍由相关立方体层次结构使用的情况下，试图从层次结构中除去属性。

- 在要被添加的属性已不再由相关层次结构使用的情况下，试图将属性添加至立方体层次结构。

### 用户响应

执行下列其中一个操作：

- 在从相关层次结构中除去相同属性之前从立方体层次结构中除去属性。
- 在将相同属性添加至相关立方体层次结构之前将属性添加至层次结构。

有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6504: 无法对 *object\_name* 对象执行该操作。立方体层次结构必须引用由用于派生立方体层次结构的层次结构已引用的属性关系。:**

#### 解释

无法执行请求的操作，原因是它将违反元数据目录中元数据对象之间的引用约束。如果出现以下其中一种情况，则会出现问题：

- 在要被除去的属性关系仍由相关立方体层次结构使用的情况下，试图从层次结构中除去属性关系。
- 在要被添加的属性关系已不再由相关层次结构使用的情况下，试图将属性关系添加至立方体层次结构。

### 用户响应

执行下列其中一个操作：

- 在从相关层次结构中除去相同属性关系之前从立方体层次结构中除去属性关系。
- 在将相同属性关系添加至相关立方体层次结构之前将属性关系添加至层次结构。

有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6505: 因为 *hierarchy\_name* 层次结构被立方体层次结构引用，所以无法执行该操作。:**

#### 解释

无法执行请求的操作，原因是它将违反元数据目录中元数据对象之间的引用约束。标识的层次结构当前由立方体层次结构引用，因此无法删除标识的层次结构。

### 用户响应

在删除标识的层次结构之前，请更改引用对象以使它们不再引用标识的层次结构。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6506: 无法对 *object\_name* 对象执行该操作。立方体事实必须引用已由用于派生立方体事实的事实引用的量度。:**

#### 解释

无法执行请求的操作，原因是它将违反元数据目录中元数据对象之间的引用约束。如果出现以下其中一种情况，则会出现问题：

- 在要被除去的量度仍由相关立方体事实使用的情况下，试图从事实中除去量度。
- 在要被添加的量度已不再由相关事实使用的情况下，试图将量度添加至立方体事实。

## 用户响应

执行下列其中一个操作:

- 在从相关事实中除去相同量度之前从立方体事实中除去量度。
- 在将相同量度添加至相关立方体量度之前将量度添加至事实。

有关元数据规则的更多信息, 请参阅《DB2 Cube Views 指南和参考》。

**6507: 因为 facts\_name 事实对象被立方体事实对象引用, 所以无法执行该操作。:**

## 解释

无法执行请求的操作, 原因是它将违反元数据目录中元数据对象之间的引用约束。标识的事实当前由立方体事实引用, 因此无法删除事实。

## 用户响应

在删除标识的事实之前, 请更改引用对象以使它们不再引用标识的事实。有关元数据规则的更多信息, 请参阅《DB2 Cube Views 指南和参考》。

**6508: 因为 hierarchy\_name 层次结构被维引用, 所以无法执行该操作。:**

## 解释

无法执行请求的操作, 原因是它将违反元数据目录中元数据对象之间的引用约束。标识的层次结构当前由维引用, 因此无法删除标识的层次结构。

## 用户响应

在删除标识的层次结构之前, 请更改引用对象以使它们不再引用标识的层次结构。有关元数据规则的更多信息, 请参阅《DB2 Cube Views 指南和参考》。

**6509: 因为 join\_name 连接被事实对象引用, 所以无法执行该操作。:**

## 解释

无法执行请求的操作, 原因是它将违反元数据目录中元数据对象之间的引用约束。标识的连接当前由事实引用, 因此无法删除标识的连接。

## 用户响应

在删除标识的连接之前, 请更改引用对象以使它们不再引用标识的连接。有关元数据规则的更多信息, 请参阅《DB2 Cube Views 指南和参考》。

**6510: 因为 cube\_dimension\_name 立方体维被立方体引用, 所以无法执行该操作。:**

## 解释

无法执行请求的操作, 原因是它将违反元数据目录中元数据对象之间的引用约束。标识的立方体维当前由立方体引用, 因此无法删除标识的立方体维。

## 用户响应

在删除标识的立方体维之前, 请更改引用对象以使它们不再引用标识的立方体维。有关元数据规则的更多信息, 请参阅《DB2 Cube Views 指南和参考》。

**6511: 无法对 `object_name` 对象执行该操作。立方体的立方体维必须从派生立方体的立方体模型引用的维派生。:**

#### 解释

无法执行请求的操作，原因是它将违反元数据目录中元数据对象之间的引用约束。如果出现以下其中一种情况，则会出现问题:

- 在要被除去的维仍由相关立方体的立方体维所使用，试图从立方体模型中除去维。
- 在要被添加的立方体维已不再由相关立方体模型使用，试图将立方体维添加至立方体。

#### 用户响应

执行下列其中一个操作:

- 在从相关立方体模型中除去相关维之前从立方体中除去立方体维。
- 在将相关立方体维添加至相关立方体之前将维添加至立方体模型。

有关元数据规则的更多信息，请参阅《*DB2 Cube Views 指南和参考*》。

**6512: 因为 `dimension_name` 维被立方体维引用，所以无法执行该操作。:**

#### 解释

无法执行请求的操作，原因是它将违反元数据目录中元数据对象之间的引用约束。标识的维当前由立方体维引用，因此无法删除标识的维。

#### 用户响应

在删除标识的维之前，请更改引用对象以使它们不再引用标识的维。有关元数据规则的更多信息，请参阅《*DB2 Cube Views 指南和参考*》。

**6513: 无法对 `object_name` 对象执行该操作。立方体维的立方体层次结构必须从用于派生立方体维的相同维所引用的层次结构派生。:**

#### 解释

无法执行请求的操作，原因是它将违反元数据目录中元数据对象之间的引用约束。如果出现以下其中一种情况，则会出现问题:

- 在要被除去的层次结构仍由相关立方体维的立方体层次结构使用的情况下，试图从维中除去层次结构。
- 在要被添加的立方体层次结构已不再由相关维使用的情况下，试图将立方体层次结构添加至立方体维。

#### 用户响应

执行下列其中一个操作:

- 在从相关维中除去相关层次结构之前从立方体维中除去立方体层次结构。
- 在将相关立方体层次结构添加至相关立方体维之前将层次结构添加至维。

有关元数据规则的更多信息，请参阅《*DB2 Cube Views 指南和参考*》。

**6514: 因为 `cube_hierarchy_name` 立方体层次结构被立方体维引用，所以无法执行该操作。:**

## 解释

无法执行请求的操作，原因是它将违反元数据目录中元数据对象之间的引用约束。标识的立方体层次结构当前由立方体维引用，因此无法删除标识的立方体层次结构。

## 用户响应

在删除标识的立方体层次结构之前，请更改引用对象以使它们不再引用标识的立方体层次结构。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6515: 无法对 *object\_name* 对象执行该操作。立方体维的立方体层次结构必须从用于派生立方体维的相同维所引用的层次结构派生。:**

## 解释

无法执行请求的操作，原因是它将违反元数据目录中元数据对象之间的引用约束。如果出现以下其中一种情况，则会出现问题:

- 在要被除去的层次结构仍由相关立方体维的立方体层次结构使用的情况下，试图从维中除去层次结构。
- 在要被添加的立方体层次结构已不再由相关维使用的情况下，试图将立方体层次结构添加至立方体维。

## 用户响应

执行下列其中一个操作:

- 在从相关维中除去相关层次结构之前从立方体维中除去立方体层次结构。
- 在将相关立方体层次结构添加至相关立方体维之前将层次结构添加至维。

有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6516: 因为 *join\_name* 连接被维引用，所以无法执行该操作。:**

## 解释

无法执行请求的操作，原因是它将违反元数据目录中元数据对象之间的引用约束。标识的连接当前由维引用，因此无法删除标识的连接。

## 用户响应

在删除标识的连接之前，请更改引用对象以使它们不再引用标识的连接。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6517: 因为 *attribute\_name* 属性被维引用，所以无法执行该操作。:**

## 解释

无法执行请求的操作，原因是它将违反元数据目录中元数据对象之间的引用约束。标识的属性当前由维引用，因此无法删除标识的属性。

## 用户响应

在删除标识的属性之前，请更改引用对象以使它们不再引用标识的属性。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6518: 因为 *attribute\_name* 属性被层次结构引用，所以无法执行该操作。:**

## 解释

无法执行请求的操作，原因是它将违反元数据目录中元数据对象之间的引用约束。标识的属性当前由层次结构引用，因此无法删除标识的属性。

## 用户响应

在删除标识的属性之前，请更改引用对象以使它们不再引用标识的属性。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6519: 因为 *relationship\_name* 属性关系被层次结构引用，所以无法执行该操作。:**

## 解释

无法执行请求的操作，原因是它将违反元数据目录中元数据对象之间的引用约束。标识的属性关系当前由层次结构引用，因此无法删除标识的属性关系。

## 用户响应

在删除标识的属性关系之前，请更改引用对象以使它们不再引用标识的属性关系。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6520: 因为 *dimension\_name* 维被立方体模型引用，所以无法执行该操作。:**

## 解释

无法执行请求的操作，原因是它将违反元数据目录中元数据对象之间的引用约束。标识的维当前由立方体模型引用，因此无法删除标识的维关系。

## 用户响应

在删除标识的维之前，请更改引用对象以使它们不再引用标识的维。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6521: 因为 *join\_name* 连接被立方体模型引用，所以无法执行该操作。:**

## 解释

无法执行请求的操作，原因是它将违反元数据目录中元数据对象之间的引用约束。标识的连接当前由立方体模型引用，因此无法删除标识的连接。

## 用户响应

在删除标识的连接之前，请更改引用对象以使它们不再引用标识的连接。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6522: 因为 *object\_name* 被事实对象引用，所以无法执行该操作。:**

## 解释

无法执行请求的操作，原因是它将违反元数据目录中元数据对象之间的引用约束。标识的对象当前由事实引用，因此无法删除标识的对象。

## 用户响应

在删除标识的对象之前，请更改引用对象以使它们不再引用标识的对象。有关元数据规则的更多信息，请参阅《DB2 Cube Views 指南和参考》。



**6523: 因为 *attribute\_name* 左边属性被属性关系引用, 所以无法执行该操作。:**  
解释

无法执行请求的操作, 原因是它将违反元数据目录中元数据对象之间的引用约束。标识的属性当前由属性关系引用, 因此无法删除标识的属性。

#### 用户响应

在删除标识的属性之前, 请更改引用对象以使它们不再引用标识的属性。有关元数据规则的更多信息, 请参阅《DB2 Cube Views 指南和参考》。

**6524: 因为 *attribute\_name* 右边属性被属性关系引用, 所以无法执行该操作。:**  
解释

无法执行请求的操作, 原因是它将违反元数据目录中元数据对象之间的引用约束。标识的属性当前由属性关系引用, 因此无法删除标识的属性。

#### 用户响应

在删除标识的属性之前, 请更改引用对象以使它们不再引用标识的属性。有关元数据规则的更多信息, 请参阅《DB2 Cube Views 指南和参考》。

**6525: 因为 *attribute\_name* 右边属性被连接引用, 所以无法执行该操作。:**  
解释

无法执行请求的操作, 原因是它将违反元数据目录中元数据对象之间的引用约束。标识的属性当前由连接引用, 因此无法删除标识的属性。

#### 用户响应

在删除标识的属性之前, 请更改引用对象以使它们不再引用标识的属性。有关元数据规则的更多信息, 请参阅《DB2 Cube Views 指南和参考》。

**6526: 因为 *attribute\_name* 左边属性被连接引用, 所以无法执行该操作。:**  
解释

无法执行请求的操作, 原因是它将违反元数据目录中元数据对象之间的引用约束。标识的属性当前由连接引用, 因此无法删除标识的属性。

#### 用户响应

在删除标识的属性之前, 请更改引用对象以使它们不再引用标识的属性。有关元数据规则的更多信息, 请参阅《DB2 Cube Views 指南和参考》。

**6527: 因为 *model\_name* 立方体模型被立方体引用, 所以无法执行该操作。:**  
解释

无法执行请求的操作, 原因是它将违反元数据目录中元数据对象之间的引用约束。标识的立方体模型当前由立方体引用, 因此无法删除标识的立方体模型。

#### 用户响应

在删除标识的立方体模型之前, 请更改引用对象以使它们不再引用标识的立方体模型。有关元数据规则的更多信息, 请参阅《DB2 Cube Views 指南和参考》。

**6528:** 因为 *cube\_facts\_name* 立方体事实对象被立方体引用, 所以无法执行该操作。:

**解释**

无法执行请求的操作, 原因是它将违反元数据目录中元数据对象之间的引用约束。标识的立方体事实当前由立方体引用, 因此无法删除标识的立方体事实。

**用户响应**

在删除标识的立方体事实之前, 请更改引用对象以使它们不再引用标识的立方体事实。有关元数据规则的更多信息, 请参阅《DB2 Cube Views 指南和参考》。

**6529:** 因为 *facts\_name* 事实对象被立方体模型引用, 所以无法执行该操作。:

**解释**

无法执行请求的操作, 原因是它将违反元数据目录中元数据对象之间的引用约束。标识的事实当前由立方体模型引用, 因此无法删除标识的事实。

**用户响应**

在删除标识的事实之前, 请更改引用对象以使它们不再引用标识的事实。有关元数据规则以及元数据对象之间的引用约束的更多信息, 请参阅《DB2 Cube Views 指南和参考》。

**6530:** 因为 *level\_name* 层被维引用, 所以无法执行该操作。:

**解释**

无法执行请求的操作, 原因是该操作违反元数据目录中元数据对象之间的引用约束。该层当前由维引用, 因此此时无法删除该层。

**用户响应**

在删除该层之前, 改变引用对象以使它们不再引用该层。有关元数据规则以及元数据对象之间的引用约束的更多信息, 请参阅《DB2 Cube Views 指南和参考》。

**6531:** 因为 *level\_name* 层被层次结构引用, 所以无法执行该操作。:

**解释**

无法执行请求的操作, 原因是它违反元数据目录中元数据对象之间的引用约束。该层当前由层次结构引用, 因此此时无法删除该层。

**用户响应**

在删除该层之前, 改变引用对象以使它们不再引用该层。有关元数据规则以及元数据对象之间的引用约束的更多信息, 请参阅《DB2 Cube Views 指南和参考》。

**6532:** 因为 *level\_name* 立方体层被立方体层次结构引用, 所以无法执行该操作。:

**解释**

无法执行请求的操作, 原因是它违反元数据目录中元数据对象之间的引用约束。该立方体层当前由立方体层次结构引用, 因此此时无法删除该立方体层。

## 用户响应

在删除该层之前，改变引用对象以使它们不再引用该立方体层。有关元数据规则以及元数据对象之间的引用约束的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6533:** 无法对 *cube\_name* 立方体执行该操作。立方体维和立方体层次结构必须从该立方体派生。该立方体层次结构必须属于该立方体维。:

### 解释

无法执行请求的操作，原因是它违反元数据目录中元数据对象之间的引用约束。该立方体的立方体维当前由优化片的立方体维和立方体层次结构引用，因此此时无法删除该立方体的优化片。

## 用户响应

在删除该立方体的优化片之前，改变引用对象以使它们不再引用该立方体的优化片。有关元数据规则以及元数据对象之间的引用约束的更多信息，请参阅《DB2 Cube Views 指南和参考》。

**6534:** 无法对 *level\_name* 层执行该操作。因为相应立方体层引用相关属性，所以不能从该层除去相关属性。:

### 解释

无法执行请求的操作，原因是它违反元数据目录中元数据对象之间的引用约束。不能从该层删除相关属性，原因是相关属性已被相应的立方体层引用。

## 用户响应

要删除相关属性，修改引用立方体层对象以使该立方体层不引用该层的您想要删除的相关属性。有关元数据规则以及元数据对象之间的引用约束的更多信息，请参阅《DB2 Cube Views 指南和参考》。

# 优化

## 优化错误代码

**7001:** 没有为 *model\_name* 立方体模型定义的立方体。:

### 解释

没有为立方体模型定义的立方体。

## 用户响应

如果要对抽取查询（将数据从立方体模型读入 MOLAP 立方体）进行优化，您必须定义表示 MOLAP 立方体的立方体。不能在未定义一个或多个立方体的情况下对抽取查询进行优化。

**7002:** *model\_name* 立方体模型不存在。:

### 解释

未定义具有指定名称的立方体模型。

## 用户响应

验证是否指定了正确的立方体模型和模式名。名称和模式区分大小写。使用 OLAP 中心来查看现有立方体模型的列表。

**7003: 未找到表空间 *table\_space\_name*。:**

### 解释

未定义具有此名称的表空间。

## 用户响应

验证是否指定了正确的表空间名称。

**7004: 优化顾问程序无法确定建议。:**

### 解释

您需要为可用于优化此立方体模型的磁盘空间大小指定一定的限制。顾问程序无法产生使用少于或等于指定磁盘空间大小的建议。

## 用户响应

指定更大的磁盘空间限制并再次运行优化顾问程序向导。

**7005: 表空间 *table\_space\_name* 不能用于存储摘要表。:**

### 解释

表空间不具有存储表数据所需的正确数据存储类型。表空间必须是 REGULAR 类型的表空间。不能使用 LONG、USER TEMPORARY 和 SYSTEM TEMPORARY 表空间来存储摘要表。

## 用户响应

指定 REGULAR 表空间来存储摘要表。

**7006: 表空间 *table\_space\_name* 不能用于存储索引。:**

### 解释

指定的表空间不具有存储索引数据所需的正确数据存储类型。表空间必须是 REGULAR 或 LONG 类型的表空间。无法使用 USER TEMPORARY 和 SYSTEM TEMPORARY 表空间来存储索引。

## 用户响应

指定 REGULAR 或 LONG 表空间来存储索引。

**7007: 优化验证 *model\_name* 立方体模型失败。:**

### 解释

立方体模型和关联的元数据对象违反优化所需的一条或多条元数据对象规则。将不会执行优化。

## 用户响应

除非立方体模型和关联的元数据对象符合优化的元数据对象规则，否则无法执行优化。有关优化规则的更多信息，请参阅《安装与用户指南》。

**7008: 立方体模型没有任何维拥有可优化的层次结构。:**

### 解释

由于优化顾问程序找不到带有可优化层次结构的维，因此将不执行优化。

## 用户响应

确保立方体模型至少有一个维具有非递归层次结构。

**7009: 不能指定多个 MOLAP 抽取或混合抽取或者同时对一个立方体指定 MOLAP 抽取和混合抽取类型的优化片。对 cube\_name 立方体指定了多个抽取类型的优化片。:**

### 解释

未优化指定的立方体，原因是对该立方体指定了多个抽取类型的优化片。只能对每个立方体指定一个 MOLAP 抽取或混合抽取类型的优化片。

## 用户响应

确保每个立方体只有一个 MOLAP 抽取或混合抽取类型的优化片。

**7010: 必须至少在一个立方体维中的混合抽取优化片下面的立方体层定义深入钻取优化片。:**

### 解释

必须至少在一个立方体维中的混合抽取优化片下面的立方体层定义深入钻取优化片。

## 用户响应

确保至少在一个立方体维中的混合抽取优化片下面的立方体层定义了深入钻取优化片。

**7011: 优化顾问程序已停止，无法确定分配时间内的任何建议。:**

### 解释

在确定建议之前，优化顾问程序已由停止建议操作停止。

## 用户响应

允许优化顾问程序运行较长时间以使它可以确定建议。

**7012: 无效标识值。:**

### 解释

指定的标识值无效。

**7013: 尝试对基于视图、别名、昵称或不支持采样的其它数据库对象的事实对象进行 DB2 表采样。:**

## 解释

DB2 Cube Views 试图从基于视图、别名或昵称的事实对象中的数据进行采样。视图、别名和昵称不支持 DB2 表采样。

## 用户响应

DB2 Cube Views 可对仅根据表定义（而不是根据视图、别名或昵称定义）的事实对象执行表采样。对优化顾问程序关闭采样选项。

**7014: 优化顾问程序不能确定任何建议。:**

## 解释

优化顾问程序不能使用您指定的参数确定对立方体模型的任何建议。

## 用户响应

检查信息性和警告消息以获取有关优化顾问程序不能建议任何摘要表的原因的更多信息。

## 优化警告

**7200: 因为立方体模型包含一个或多个非分布式量度，所以建议的摘要表将使用延迟刷新。:**

## 解释

为摘要表选择了立即刷新选项。但是，如果在立方体模型中定义了非分布式量度，则无法立即刷新摘要表。分布式量度使用可从任何中间值聚集的简单聚集函数，如 SUM 和 COUNT。非分布式量度使用较为复杂的聚集函数，如 STDDEV，而且它们必须从基本表聚集。

## 用户响应

如果不必要保持摘要表和基本表同步，则无需进行操作。如果摘要表必须保持与基本表同步，则需要更改元数据以便仅定义分布式量度。

**7201: 建议的摘要表 `table_name` 将使用延迟刷新，原因是在该建议的摘要表的全查询中找到一个或多个作为列的可空属性。:**

## 解释

为摘要表选择了立即刷新选项。但是，建议的摘要表包含一个或多个用作摘要表全查询中可空列的属性。使用摘要表全查询中的可空列可能会导致立即刷新的性能变慢。已将摘要表设置为延迟刷新以避免出现此性能问题。

## 用户响应

要更改属性的可空性，您必须更改属性的 SQL 表达式和 / 或属性所使用 DB2 表列的可空性。通常不建议进行这些更改，因为它们可能难于实现。

**7202: `table_name` 表没有统计信息。:**

## 解释

优化顾问程序无法为指定的表找到有效的表统计值。

## 用户响应

使用 RUNSTATS 命令为指定的表创建统计信息。然后，再次运行优化顾问程序向导。

**7203:** 对于 *cube\_name* 立方体中的混合抽取优化片 *slice\_name*，必须对立方体中的每个立方体维指定立方体层。:

### 解释

优化顾问程序不能对指定的混合抽取类型的优化片进行优化，原因是该优化片未经严格定义。混合抽取类型的优化片必须对每个立方体维定义立方体层。不能选择“未指定”。

## 用户响应

确保该优化片对立方体中的每个立方体维指定了立方体层。

**7203:** 对于 *cube\_name* 立方体中的混合抽取优化片 *slice\_name*，必须对立方体中的每个立方体维指定立方体层。:

### 解释

优化顾问程序不能对指定的混合抽取类型的优化片进行优化，原因是该优化片未经严格定义。混合抽取类型的优化片必须对每个立方体维定义立方体层。不能选择“未指定”。

## 用户响应

确保该优化片对立方体中的每个立方体维指定了立方体层。

**7205:** 建议的摘要表必须使用延迟刷新更新方法，原因是立方体模型的一个或多个底层基本表为昵称。:

### 解释

您对摘要表选择了立即更新选项，但如果任何基础表为昵称，则摘要表不能使用立即刷新更新方法。

## 用户响应

无需操作。

**7206:** 优化顾问程序未能使用数据采样来确定建议的摘要表。:

### 解释

您已指定优化顾问程序应使用数据采样来确定优化摘要表建议。由于定义事实对象的方式，优化顾问程序不能执行数据采样。如果事实对象被定义为昵称，则优化顾问程序尝试执行数据采样但可能无法进行，除非立方体模型的维具有适当的基数并且事实-维连接在映射至单列的单个属性上才会如此。对表示维键的事实表列定义的索引也会影响优化顾问程序对定义为昵称的事实对象执行数据采样的能力。如果事实对象被定义为视图，则数据采样不受支持。如果事实对象被定义为别名，则不能进行数据采样（根据别名映射至的内容）。创建建议而不执行数据采样的后果是优化顾问程序不能建议最优摘要表。如果事实对象为表，则优化顾问程序总是可以执行数据采样。

## 用户响应

可以按原样使用建议，或者可以通过对事实对象指定表（而不是视图或别名）并再次运行优化顾问程序来尝试改进建议。

## 优化参考消息

**7400:** 因为立方体模型包含一个或多个非分布式量度，所以使用 **ROLLUP** 运算符定义摘要表。:

### 解释

量度是分布式或非分布式的。分布式量度使用可从任何中间值聚集的简单聚集函数，如 **SUM** 和 **COUNT**。非分布式量度使用较为复杂的聚集函数，如 **STDDEV**，而且它们必须从基本表聚集。为了避免从基本表聚集非分布式量度的开销，已使用预先聚集非分布式量度的 **ROLLUP** 操作程序来定义摘要表。

## 用户响应

无需操作。

**7401:** 建议使用 **table\_name** 摘要表。估计该表具有 **rows** 行、**n MB** 表大小和 **n MB** 索引大小。:

### 解释

此消息是建议的摘要表的描述，包括估计的行数、估计的磁盘空间和估计用于索引的磁盘空间。

## 用户响应

无需操作。

**7402:** 共有 **n** 个摘要表不符合指定的磁盘空间限制。它们的累积估计大小为 **n MB**。

### 解释

此消息提供有关不符合磁盘空间限制的建议摘要表的信息。

## 用户响应

要查看建议中的这些摘要表，请使用较大的指定磁盘空间限制再次运行优化顾问程序。

**7403:** 建议包括对 **cube\_name** 立方体的优化。:

### 解释

已为指定的立方体建议了摘要表。将对该立方体的某些查询进行优化。

## 用户响应

无需操作。

**7404:** 建议未包括对 **cube\_name** 立方体的优化。:



## 解释

没有专门为此立方体建议摘要表。特定于该立方体的查询可能不会显示出性能改进。

## 用户响应

如果由于磁盘空间局限性的原因而未在建议中包括摘要表，请使用较大的指定磁盘空间限制再次运行优化顾问程序。建议可能包括一个或多个摘要表以便优化该立方体的查询。

**7405: 当优化顾问程序确定建议时，超出指定的时间限制。:**

## 解释

优化顾问程序已生成一条建议。如果允许更多时间，优化顾问程序可能会生成更好的建议，因为它可以进行额外的分析。更长时间地运行优化顾问程序并不保证有更好的结果。

## 用户响应

您可以使用指定的更长时间再次运行优化顾问程序，也可以创建建议的摘要表并查看性能是否是可接受的。

**7406: dimension\_name 维不包含任何可由优化顾问程序进行优化的层次结构。:**

## 解释

优化顾问程序无法优化递归层次结构。指定的维不包含任何可优化的层次结构，因此优化顾问程序将忽略此维。未优化从该维中引用属性的查询。

## 用户响应

无需操作。使用此维中属性的查询将不会有任何性能改进。

**7407: 建议的摘要表对立方体模型中 n% 的片进行优化。针对优化过的片运行的查询应具有更佳的性能。:**

## 解释

SQL 查询访问立方体模型中的特定片。分析性能改进的一种方法是考虑将改进哪部分可查询的片。如果立方体模型使用分布式量度，则访问片（逻辑位于摘要表片之上）的查询性能将有所提高。

例如，在具有 Time 维（使用层次结构 [All-Year-Quarter-Month-Day]）和 Region 维（使用层次结构 [All-Country-Region-State-City-Store]）的立方体模型中存在 30 个可能的片。您可以通过将维层次结构中的层数相乘来计算可能片的数量。如果建议的摘要表优化 Month-City 片，则位于该片或该片之上的所有片都将被优化。本示例中，将优化 30 个可能片中的 20 个，或 67%（20/30）的片。优化范围从不会达到 100%，因为那将要求在摘要表中复制基本表。通常，优化最底层的片得益最少，因为它们与基本表差别很大。

## 用户响应

无需操作。如果百分比较低，您可以使用较大的指定磁盘空间限制再次运行优化顾问程序向导。

**7408:** 正在从数据库中读取立方体模型元数据。:

解释

优化顾问程序正在读取描述立方体模型的元数据。元数据包含对优化建议有重要影响的信息。

用户响应

无需操作。

**7409:** 正在选择要包括在摘要表中的聚集。:

解释

优化顾问程序正在测试可能的摘要表配置以确定指定标准的最佳配置。

用户响应

无需操作。

**7410:** 正在从立方体模型中对数据进行采样。:

解释

优化顾问程序正在从事实表和维表中读取数据子集，以便估计摘要表的大小。在优化顾问程序考虑可能的摘要表时，采样可能多次发生。

用户响应

无需操作。

**7411:** 正在为建议的摘要表定义索引。:

解释

优化顾问程序已确定建议的摘要表，并正在为摘要表选择要构建的索引。

用户响应

无需操作。

**7413:** *cube\_name* 立方体没有指定任何优化片，所以优化顾问程序将针对下寻类型的查询优化该立方体。:

解释

如果未对立方体定义任何优化片，则优化顾问程序将针对下寻类型的查询优化立方体。

**7414:** 未对要优化的立方体模型定义任何立方体，所以优化顾问程序将在优化该立方体模型时假设它具有用于报表类型查询的立方体。:

解释

如果未对立方体模型定义任何立方体，优化顾问程序将在优化该立方体模型时假设它具有用于下寻类型查询的立方体。如果未对正在优化的立方体模型定义任何立方体，则优化顾问程序在作出建议时假设您有这样一个立方体，该立方体对每个立方体维在所有层次定义了下寻优化片。

**7415:** 优化顾问程序未对 *cube\_name* 立方体中的优化片 *slice\_name* 建议摘要表。:

**解释**

未对指定优化片建议摘要表。

**用户响应**

无需操作。

**7416:** 已停止操作。该建议可能不是最优的。:

**解释**

优化顾问程序已由停止建议操作停止。该建议可能不是最优的。

**用户响应**

要获取最优建议，再次运行优化顾问程序而不限制时间。

**7417:** 优化顾问程序已停止创建建议，原因是它达到已设置的时间限制。如果允许进行较长时间，优化顾问程序可建议更好的摘要表。:

**解释**

为优化顾问程序提供的时间可能太短了。建议：以较长的时间限制运行优化顾问程序。

**用户响应**

以较长的时间限制运行优化顾问程序。优化顾问程序可能建议更佳的摘要表。

**7418:** 优化顾问程序不能对 *cube\_name* 立方体建议摘要表，原因是同事事实表中的 *number* 行相比摘要表的行数太多了。:

**解释**

优化顾问程序不能对立方体建议摘要表。同事事实表中的行数相比，摘要表中的行数太多了。

**用户响应**

如果指定了优化片，则一个或多个优化片可能指定了过多的层或者指定的层过于接近事实表的粒度。可对某些立方体维指定任何层或指定更接近层次结构顶部的层。还可允许优化顾问程序运行较长时间以改进建议。

---

## OLAP 中心消息

### 10000-10600

**10000:** OLAP 中心无法检索任何数据库名称。

**解释**

从 DB2 中检索数据库名称列表时出错。

## 用户响应

检查是否正确地安装了 OLAP 中心。如果问题仍然存在，请与 IBM 软件支持人员联系。

### **10001: 输入用户名。**

#### 解释

用户名字段为空。

## 用户响应

在用户名字段中输入用户名。

### **10002: 输入密码。**

#### 解释

密码字段为空。

## 用户响应

在密码字段中输入密码。

### **10004: 无法对在 SQL 表达式中输入的属性进行语法分析。**

#### 解释

输入的 SQL 表达式引用既不在给定上下文中有效也不在数据库中出现的属性。

## 用户响应

确保 SQL 表达式仅引用那些出现在 SQL 表达式构建器数据部分中的属性。

### **10005: 已成功地将元数据对象导出至 *file\_name* 文件。**

#### 解释

导出成功。

## 用户响应

无需操作。

### **10006: 输入要将元数据对象导出至其中的文件名。**

#### 解释

export 实用程序将导出的元数据对象写入用户输入的文件名。

## 用户响应

在文件名字段中输入文件名。

**10007: 选择要导出的立方体或立方体模型。**

**解释**

导出窗口可以导出立方体或立方体模型。

**用户响应**

选择要导出的对象。

**10008: 为要创建的对象输入唯一的名称。**

**解释**

在 SQL 表达式构建器中，要创建的属性或量度的名称字段为空。

**用户响应**

在名称字段中输入对象名。对象名在属性和量度的名称空间中必须唯一。

**10009: 输入该对象的 SQL 表达式。**

**解释**

对象的 SQL 表达式字段为空。

**用户响应**

输入对象的 SQL 表达式。

**10010: 未通过表名来限定 *column\_name* 列。**

**解释**

未通过表名来限定 SQL 表达式中输入的列名。

**用户响应**

确保使用将“.”用作分隔符的表名来限定 SQL 表达式中的列名。

**10011: 未通过模式名来限定 *column\_name* 列。**

**解释**

SQL 表达式中的列引用必须同时通过用“.”分隔的表名和模式名来限定。

**用户响应**

确保同时通过用“.”分隔的表名和模式名来限定 SQL 表达式中的列名。

**10012: 聚集脚本中的第一个元素不能是维。**

**解释**

创建的聚集脚本使用维作为第一个元素。

**用户响应**

使用聚集函数作为聚集脚本中的第一个元素。

**10013: 选择现有的量度或输入 SQL 表达式以作为聚集中 *function\_name* 多参数函数的第二个参数。**

解释

在聚集脚本中使用多参数函数时，第一个参数被定义为聚集所关联的量度。对于第二个参数，请选择现有的量度或输入 SQL 表达式。

用户响应

输入量度或 SQL 表达式作为第二个参数。

**10014: *function\_name* 聚集函数没有匹配的维。**

解释

必须将聚集脚本中的每个聚集函数至少应用于一个维。

用户响应

确保聚集脚本中的每个聚集函数至少应用于一个维。

**10015: *object\_name* 对象缺少右括号。**

解释

在 OLAP 中心中，表达式中指定的属性、量度或列应分别由 @Attribute(), @Measure() 或 @Column() 括起来。

用户响应

输入对象的右括号。

**10016: 对 SQL 表达式中 *object\_name* 对象的引用将创建无效引用循环。**

解释

对象在其 SQL 表达式中引用本身。

用户响应

确保 SQL 表达式中的对象不创建引用循环。

**10017: 未发现错误。SQL 表达式有效。**

解释

SQL 表达式有效。

用户响应

无需操作。

**10018: 未发现错误。聚集脚本有效。**

解释

聚集脚本中的聚集集合有效。

用户响应

无需操作。

**10020: 输入名称。**

解释

对象的名称字段为空。

用户响应

在名称字段中输入对象名。

**10021: 输入模式名。**

解释

模式字段为空。

用户响应

在模式字段中输入模式名。

**10022: 输入业务名。**

解释

业务名字段为空。

用户响应

业务名可以显示在商业智能应用程序中用于向最终用户标识对象。在业务名字段中输入业务名。

**10023: 至少选择一个层以包括在立方体层次结构中。**

解释

没有在立方体层次结构中指定层。

用户响应

至少选择一个层以包括在立方体层次结构中。

**10024: 至少为立方体事实选择一个量度。**

解释

没有为立方体事实指定量度。

#### 用户响应

至少为立方体事实选择一个量度。

**10025: 至少选择一个层以包括在立方体层次结构中。**

#### 解释

没有在立方体层次结构中指定属性。

#### 用户响应

至少选择一个属性以包括在立方体层次结构中。

**10026: 在立方体中至少选择一个维。**

#### 解释

没有在立方体中指定维。

#### 用户响应

至少选择一个维，然后单击 [...] 按钮以指定立方体维的详细信息。

**10027: 已清除当前存在于立方体中的一个或多个维。单击“是”以删除立方体维。单击“否”返回到窗口，然后单击“取消”在不保存更改的情况下关闭窗口。**

#### 解释

已清除选择的一个或多个维。将从立方体中删除相应的立方体维。

#### 用户响应

单击窗口中的**是**以从立方体中除去立方体维。单击**否**保留立方体维，然后单击**取消**在不保存的情况下关闭窗口。

**10028: 如果选择了功能相关性复选框，则无法使用多:多基数定义属性关系。**

#### 解释

无法同时使用选择的功能相关性和多:多基数定义属性关系。

#### 用户响应

为属性关系选择另一基数，或者清除功能相关性复选框。

**10029: 选择属性关系的左边的属性和右边的属性。**

#### 解释

如果未同时选择左边的属性和右边的属性，则无法定义属性关系。

#### 用户响应

同时选择左边的属性和右边的属性。



**10030: 无法为属性关系中左边的属性和右边的属性同时选择相同属性。**

**解释**

如果左边的属性和右边的属性相同，则无法定义属性关系。

**用户响应**

选择不同的左边的属性和右边的属性。

**10031: 具有指定名称和模式的对象在数据库中已存在。输入另一名称。**

**解释**

正在创建或修改类型的对象在数据库中已存在，并指定了相同的名称和模式。

**用户响应**

为对象输入唯一的名称。

**10032: 至少指定一个属性对。**

**解释**

连接必须至少具有一个属性对。

**用户响应**

至少指定一个属性对。

**10033: 无法创建重复的属性对。**

**解释**

与新选项匹配的属性对已存在。

**用户响应**

选择不同的左边的属性和右边的属性。

**10034: 至少选择一个表。**

**解释**

尚未选择表。

**用户响应**

至少选择一个表以便继续操作。

**10035: 选择或创建新的连接以连接所有选择的表。**

**解释**

未选择连接。

#### 用户响应

选择或创建将连接所有选择的表的新连接。

#### **10036: 至少选择一个属性。**

##### 解释

未选择属性。

#### 用户响应

至少选择一个属性。

#### **10037: 选择连接以将维与事实对象连接起来。**

##### 解释

未选择连接。

#### 用户响应

选择一个将维与事实对象连接起来的连接。

#### **10038: 在给定的两个表之间仅指定一个连接。 *join\_name1* 连接和 *join\_name2* 连接同时连接相同的两个表。**

##### 解释

为相同的表对选择了多个连接。

#### 用户响应

为每个表对仅选择一个连接。

#### **10039: 必须连接所有选择的表。为 *table\_name* 表选择一个连接。**

##### 解释

必须连接所有选择的表。

#### 用户响应

为指定的表选择连接。

#### **10040: 选择的表数与选择的连接数不对应。验证是否不存在连接循环以及是否连接了所有的表。**

##### 解释

必须连接所有选择的表。

#### 用户响应

确保不存在连接循环并连接了所有的表。

**10042: 至少选择一个量度。**

**解释**

未指定量度。

**用户响应**

至少选择一个量度。

**10043: 选择表列。**

**解释**

未指定表列。

**用户响应**

选择列。

**10044: 选择 SQL 表达式。**

**解释**

未指定 SQL 表达式。

**用户响应**

单击构建表达式按钮以构建表达式。

**10045: 未指定聚集脚本。**

**解释**

未指定聚集脚本。

**用户响应**

单击构建脚本按钮以构建聚集脚本。

**10046: 在打开表达式构建器之前选择量度。**

**解释**

未选择量度。

**用户响应**

从表中选择一个量度。

**10047: 在打开聚集脚本构建器之前选择量度。**

**解释**

未选择量度。

**用户响应**

选择量度。

**10048:** 无法为 *measure\_name* 量度启动聚集脚本构建器，原因是 *model\_name* 立方体模型没有至少具有一个维。

解释

如果立方体模型没有至少具有一个维，则无法指定聚集脚本。

用户响应

在指定聚集脚本之前将维添加至立方体模型。

**10049:** 要编辑表达式，请指定属性。

解释

未选择属性。

用户响应

选择属性。

**10050:** 将依据数据库刷新元数据。出错时正在进行的任何更改将丢失。

解释

调用 DB2 存储过程时出错。

用户响应

单击**确定**来刷新 OLAP 中心显示的元数据。出错时正在进行的任何更改将丢失。将使用允许用户继续工作的数据库中的相应对象来刷新 OLAP 中心中显示的对象。

**10051:** 无法验证 *model\_name* 立方体模型以进行优化。DB2 返回以下消息：*message*。

解释

OLAP 中心无法为选择的立方体模型启动优化顾问程序向导，原因是选择的立方体模型未传递由存储过程 API 执行的验证。

用户响应

检查存储过程 API 文档以了解立方体模型验证规则。遵循从 DB2 返回的消息中的指示信息。

**10052:** 某些装入的属性或量度映射至不再存在于数据库中的 *column\_names* 列。通过复原列所属的表和 / 或删除无效属性或量度来解决问题。

解释

此消息在启动 OLAP 中心时或在单击**查看 -> 刷新**之后出现。出现此消息的原因是装入的属性或量度所映射至的表已被删除或重命名。

## 用户响应

按以下其中一种方法更正问题:

- 复原被删除或重命名的表。
- 将属性 / 量度映射至数据库中已存在的表。
- 删除映射至不存在的列的属性 / 量度。

**10053: *model\_name* 立方体模型优化验证返回一个警告。DB2 返回以下消息: *message***

### 解释

OLAP 中心尝试在启动优化顾问程序之前验证立方体模型, DB2 返回了一个警告。警告可能表示存在无法优化的立方体模型。例如, 立方体模型可能包含引用表 (在表之间未定义约束) 的视图。

## 用户响应

检查 DB2 返回的消息, 并决定是否要继续运行优化顾问程序向导。

**10060: 立方体模型不完整。立方体模型必须包含一个事实对象、至少一个维以及每个维至少一个层次结构, 然后才能创建立方体。**

### 解释

立方体模型的状态对于要创建的立方体无效。

## 用户响应

修改立方体模型以使其具有事实对象和至少一个维。确保每个维至少具有一个层次结构。

**10061: 在删除立方体模型时, 将除去它的维并删除它的事实。除去的维仍可从 **All Dimensions** 文件夹中获得。是否确定要删除 *model\_name* 立方体模型?**

### 解释

删除确认消息。

## 用户响应

确保选择的对象是要删除的对象, 然后单击是。如果不想删除选择的对象, 请单击否。

**10062: 在删除维时, 将同时删除它的层次结构及相应的立方体维。是否确定要删除 *dimension\_name* 维?**

### 解释

删除确认消息。

#### 用户响应

确保选择的对象是要删除的对象，然后单击是。如果不想删除选择的对象，请单击否。

**10063:** 在删除立方体时，将同时删除它的立方体维、立方体层次结构和立方体事实。是否确定要删除 *cube\_name* 立方体？

#### 解释

删除确认消息。

#### 用户响应

确保选择的对象是要删除的对象，然后单击是。如果不想删除选择的对象，请单击否。

**10064:** 在删除立方体维时，将同时删除它的立方体层次结构。是否确定要删除 *cube\_dimension\_name* 立方体维？

#### 解释

删除确认消息。

#### 用户响应

确保选择的对象是要删除的对象，然后单击是。如果不想删除选择的对象，请单击否。

**10065:** 是否确定要删除 *object\_name*？

#### 解释

删除确认消息。

#### 用户响应

确保选择的对象是要删除的对象，然后单击是。如果不想删除选择的对象，请单击否。

**10066:** 在除去维时，将从其立方体中除去所有相应的立方体维。是否确定要从 *object\_name* 中除去 *dimension\_name* 维？

#### 解释

除去维确认消息。

#### 用户响应

确保选择的对象是要除去的对象，然后单击是。如果不想除去选择的对象，请单击否。

**10067:** 具有名称 *file\_name* 的文件已存在。是否要覆盖其内容？

#### 解释

覆盖文件确认消息。

## 用户响应

确保要覆盖所输入文件名的内容。

**10068: 无法确定具有 *object\_name* 名称和 *schema\_name* 模式的对象的数据类型。数据库返回以下信息: *message*。**

### 解释

对于指定的对象, OLAP 中心无法确定源数据类型或聚集的数据类型。

## 用户响应

确保指定对象的 SQL 表达式正确。如果无法解决问题, 请与 IBM 软件支持人员联系。

**10069: 无法确定具有 *measure\_name* 名称和 *schema\_name* 模式的量度的源数据类型。**

### 解释

对于指定的量度, OLAP 中心无法确定源数据类型, 原因是指定的量度具有无效的源表达式。将“无”聚集设置应用于量度时, 量度可能具有无效的源表达式, 因为量度是通过被引用的量度的聚集而不是作为独立的表达式验证的。

## 用户响应

可以执行下列其中一个操作:

- 改变指定量度的源表达式, 以便通过“无”聚集设置正确地进行验证。
- 不要在表达式中使用指定的量度。

**10070: 在删除事实对象时, 将同时删除它的量度。是否确定要删除 *facts\_name* 事实?**

### 解释

删除确认消息。

## 用户响应

确保选择的对象是要删除的对象, 然后单击是。如果不想删除选择的对象, 请单击否。

**10071: 将从数据库中删除所有选择的对象。是否要删除这些对象?**

### 解释

选择了多个对象并选择了删除选项。

## 用户响应

确保选择的对象是要删除的对象, 然后单击是。如果不想删除选择的对象, 请单击否。

**10072: 无法删除某些选择的对象。这些对象将保留在数据库中。**

解释

OLAP 中心不能删除所有选择的对象。这可能是由于某些选择的对象被数据库中的其它对象引用，删除选择的对象会使正在引用的对象无效。

用户响应

无需操作。

**10073: 不能删除任何选择的对象。**

解释

OLAP 中心不能删除任何选择的对象。这可能是由于选择的对象被数据库中的其它对象引用，删除选择的对象会使引用对象无效。

用户响应

无需操作。

**10074: 是否确定要从 *directory\_name* 中除去 *object\_name*?**

解释

除去确认消息。

用户响应

确保选择的对象是要除去的对象，然后单击是。如果不想除去选择的对象，请单击否。

**10075: 将从 *directory\_name* 中除去所有选择的对象。是否要除去这些对象?**

解释

选择了多个对象并选择了除去选项。

用户响应

确保选择的对象是要除去的对象，然后单击是。如果不想除去选择的对象，请单击否。

**10076: 不能除去某些选择的对象。这些对象将保留在数据库中。**

解释

OLAP 中心不能除去所有选择的对象。这可能是由于某些选择的对象被数据库中的其它对象引用，除去选择的对象会使正在引用的对象无效。也可能是因为父对象需要至少一个子对象，例如，在立方体层次结构中

用户响应

无需操作。



**10077: 不能除去任何选择的对象。**

解释

OLAP 中心不能除去任何选择的对象。这可能是由于选择的对象被数据库中的其它对象引用，除去选择的对象会使引用对象无效。

用户响应

无需操作。

**10078: 操作成功。数据库 *database\_name* 将返回以下信息消息。**

解释

数据库操作成功，但返回了一些信息消息。

用户响应

检查这些信息消息并决定是否需要进行进一步的操作。

**10080: 在第二次遍历 XML 的过程中未找到 *type* 类型的对象。**

解释

在正在读取的 XML 中无法找到引用的对象。

用户响应

确保正在导入的 XML 文件格式正确。如果在启动 OLAP 中心时出现此错误，请与 IBM 软件支持人员联系。

**10081: 系统未能对文件 *file\_name* 中的 XML 进行语法分析。在第 *line\_number* 行第 *character\_number* 个字符处发生了错误。解析器返回以下信息: *information*。**

解释

尝试导入 XML 文件时发生了语法分析错误

用户响应

确保要导入的 XML 文件格式正确且为有效的 Cube Views XML 元数据文件。检查该行和字符以找出该错误。

**10082: 在文件 *file\_name* 中遇到了意外解析器异常。返回了以下信息: *information*。**

解释

尝试导入 XML 文件时发生了意外的语法分析错误。

用户响应

确保要导入的 XML 文件格式正确且为有效的 Cube Views XML 元数据文件。检查该行和字符以找出该错误。

**10084:** 在模式 *schema\_name* 中具有名称 *object\_name* 的对象已存在。无法创建对象。为新对象输入唯一的名称和 / 或模式。

解释

OLAP 中心尝试创建新对象，但具有相同名称和模式的此类型对象已存在。

用户响应

为正在创建的对象输入另一名称和 / 或模式。

**10085:** 在模式 *schema\_name* 中名称为 *object\_name* 的对象已存在。无法重命名对象。为正在重命名的对象输入唯一的名称和 / 或模式。

解释

OLAP 中心尝试重命名对象，但具有相同名称和模式的此类型对象已存在。

用户响应

为正在重命名的对象输入另一名称和 / 或模式。

**10086:** 无法建立数据库连接。DB2 返回: *message*。

解释

OLAP 中心无法连接至数据库。消息中包括由 DB2 提供的一些错误信息。

用户响应

请阅读 DB2 返回的文本并更正问题。

**10087:** 无法删除 *object\_name1* 元数据对象，原因是类型为 *type* 的 *object\_name2* 对象引用了该对象。

解释

无法删除选择的元数据对象，原因是至少有另一个元数据对象在使用该对象。

用户响应

从对象所属的任何其它元数据对象中删除对象，然后再次尝试删除对象。

**10088:** : 通过 JDBC 驱动程序管理器注册 DB2 驱动程序时出错。无法建立数据库连接。返回了以下信息: *message*。

解释

在连接至 DB2 数据库之前，OLAP 中心必须注册将用于驱动程序管理器的 JDBC 驱动程序。在 JDBC 驱动程序注册过程中出错。

用户响应

检查 DB2 安装以确保安装了 db2java.zip 和 db2jcc.jar 文件。确保正确安装了 Java 和任何 JDBC 组件。请阅读消息中返回的信息以帮助解决问题。

**10089:** 访问数据库时出错。数据库返回以下信息: \n SQL 状态: *message*\n SQL 错误代码: *code*\n SQL 消息: *SQL\_message*。

解释

OLAP 中心应用程序使用 API 存储过程调用 DB2。执行命令抛出 OLAP 中心无法处理的 SQL 异常。

用户响应

使用消息中提供的附加错误信息来解决问题。如果无法解决问题, 请与 IBM 软件支持人员联系。

**10090:** 正在执行导致错误返回码的 DB2 存储过程。在返回的 XML 文档中未找到错误信息。请与 IBM 软件支持人员联系。

解释

OLAP 中心应用程序使用 API 存储过程调用 DB2。执行命令返回 *false*, 但是在存储过程返回的 XML 文档中没有错误信息。

用户响应

操作可能会成功完成, 但您应向 IBM 软件支持人员报告此问题。

**10091:** 处理数据库 API 调用时出错。返回以下信息: \n SQL 状态: *message*\n SQL 错误代码: *code*\n 操作: *operation*\n 状态标识: *ID*\n 状态文本: *text*。

解释

OLAP 中心存储过程 API 调用在执行某些 OLAP 中心更改时出错。

用户响应

请参阅消息中包含的信息。如果无法解决问题, 请与 IBM 软件支持人员联系。

**10092:** 对数据库 API 调用返回的 XML 进行语法分析时出错。返回了以下信息: *message*。

解释

OLAP 中心存储过程 API 调用了返回不完整或格式错误的 XML。OLAP 中心无法读取返回的 XML。

用户响应

使用消息中包含的信息来解决问题。如果无法解决问题, 请与 IBM 软件支持人员联系。

**10093:** *file\_name* 文件不存在。

解释

指定的文件不存在。

用户响应

指定存在的文件。

**10094:** 读取 *file\_name* 文件时出现 I/O 错误。返回了以下系统信息: *message*。

解释

从文件中读取时出现 I/O 错误。

用户响应

检查系统信息以尝试解决问题, 或指定另一文件。

**10095:** 写入 *file\_name* 文件时出现 I/O 错误。返回了以下系统信息: *message*。

解释

写入文件时出现 I/O 错误。

用户响应

检查系统信息以尝试解决问题, 或指定另一文件。

**10096:** 用于检索数据库模式的查询失败。数据库返回以下信息: *message*。

解释

用于检索数据库模式的查询失败。

用户响应

检查数据库信息以解决问题。

**10097:** 用于检索模式的表的查询失败。数据库返回以下信息: *message*。

解释

用于检索模式的表的查询失败。

用户响应

检查数据库信息以解决问题。

**10098:** 用于检索表的列的查询失败。数据库返回以下信息: *message*。

解释

用于检索表的列的查询失败。

#### 用户响应

检查数据库信息以解决问题。

**10099: 落实 DB2 连接失败。数据库返回以下信息: *message*。**

#### 解释

落实 DB2 连接失败。

#### 用户响应

检查数据库信息以解决问题。

**10100: 回滚 DB2 连接失败。数据库返回以下信息: *message*。**

#### 解释

回滚 DB2 连接失败。

#### 用户响应

检查数据库信息以解决问题。

**10101: 无法删除 *Object\_name*, 这是因为它是 *cube\_name* 立方体中的最后一个立方体维。立方体必须至少有一个立方体维有效。**

#### 解释

OLAP 中心尝试删除立方体中的最后一个立方体维。

#### 用户响应

立方体必须至少有一个立方体维有效。不要尝试从立方体中删除最后一个立方体维。

**10102: 类型为 *type1* 的对象 *object\_name1* 引用无法找到的对象 *object\_name2* 或类型 *type2*。**

#### 解释

正在读取的 XML 文件中的对象引用无法找到的对象。如果在导入过程中出错, 则正在导入的文件中可能不存在正被引用的对象。

#### 用户响应

如果正在执行导入, 确保文件包含成功导入所需的所有对象。如果在启动 OLAP 中心时出现此错误, 请与 IBM 软件支持人员联系。

**10103: 不能删除该量度, 原因是事实对象必须至少包含一个量度。**

#### 解释

不能删除该量度, 原因是事实对象必须至少包含一个量度。

#### 用户响应

无需操作。

**10104:** 不能除去该量度，原因是立方体事实必须至少包含一个量度。

解释

不能除去该量度，原因是立方体事实必须至少包含一个量度。

用户响应

无需操作。

**10105:** 不能除去该立方体层，原因是立方体层次结构必须至少包含一个立方体层。

解释

不能除去该立方体层，原因是立方体层次结构对象必须至少包含一个立方体层。

用户响应

无需操作。

**10106:** 不能除去该层，原因是它已被与 *hierarchy\_name* 层次结构相关联的 *hierarchy\_name* 立方体层次结构中的立方体层引用。

解释

不能除去该层，原因是它已被立方体层次结构（与要从中除去该属性的层次结构相关联）中的立方体层引用。

用户响应

无需操作。

**10107:** 不能除去该属性，原因是它已被与 *level\_name* 层相关联的 *level\_name* 立方体层引用。

解释

不能除去该属性，原因是它已被与要从中除去该属性的层相关联的立方体层引用。

用户响应

无需操作。

**10108:** 不能除去该属性，原因是它已被与 *dimension\_name* 维相关联的 *hierarchy\_name* 层次结构引用。

解释

不能除去该属性，原因是它已被与要从中除去该属性的维相关联的层次结构引用。

用户响应

无需操作。

**10109:** 不能除去该属性，原因是它已被与 *dimension\_name* 维相关联的 *join\_name* 连接引用。

解释

不能除去该属性，原因是它已被与要从中除去该属性的维相关联的连接引用。

用户响应

无需操作。

**10110:** 不能移动该层，原因是它已被与 *hierarchy\_name* 层次结构相关联的 *hierarchy\_name* 立方体层次结构中的立方体层引用。

解释

不能上下移动该层，原因是它已被立方体层次结构中的立方体层引用，该立方体层次结构与要编辑的层次结构相关联的。

用户响应

无需操作。

**10111:** 不能除去该属性，这是因为它是 *level\_name* 层的唯一层键属性。

解释

不能除去该属性，这是因为它是要从中除去该属性的层中的唯一层键属性而且该层必须至少具有一个层键属性。

用户响应

无需操作。

**10112:** 不能除去该属性，这是因为它是 *level\_name* 层的缺省属性。

解释

不能除去该属性，这是因为它是要从中除去该属性的层的缺省属性。

用户响应

无需操作。

**10113:** 不能除去该属性，原因是它已被与 *dimension\_name* 维相关联的 *level\_name* 层引用。

解释

不能除去该属性，原因是它已被与要从中除去该属性的维相关联的层引用。

用户响应

无需操作。

**10114:** 不能除去该层，这是因为它是 *hierarchy\_name* 层次结构中的唯一层。

解释

不能除去该层，这是因为它是层次结构中的唯一层且该层次结构必须至少包含一层。

用户响应

无需操作。

**10200:** 要导入的文件没有 UTF-8 编码。选择具有 UTF-8 编码的文件。

解释

OLAP 中心只能以 UTF-8 编码的形式导入文件。

用户响应

导入具有受支持的编码的文件。

**10201:** 为用于刷新摘要表的 SQL 脚本输入文件名。

解释

当选择“延迟更新”选项时，优化顾问程序向导创建一个 SQL 脚本来刷新摘要表。此脚本应保存在文件中并运行它才能刷新摘要表。

用户响应

输入要将 SQL 脚本保存到其中的文件名。

**10202:** 为用于创建摘要表的 SQL 脚本输入文件名。

解释

优化顾问程序向导生成 SQL 脚本来创建摘要表。此脚本应保存在文件中并运行它才能创建摘要表。

用户响应

输入要将 SQL 脚本保存到其中的文件名。

**10203:** 选择的量度不能将“无”作为它的聚集设置。只有以独占方式引用表达式中其它量度的已计算量度才可以指定“无”聚集设置。

解释

只能为使用表达式（该表达式以独占方式引用其它量度）的量度选择“无”聚集设置。

用户响应

选择另一聚集。



**10204: 没有维存在。创建一个新维以添加至立方体模型。**

**解释**

没有维存在。创建一个新维以添加至立方体模型。

**用户响应**

创建一个新维，而不是添加一个维。

**10205: 没有要添加的维，原因是所有现有的维均已包括在立方体模型中。**

**解释**

所有现有的维均已添加至立方体模型。

**用户响应**

无需操作。

**10206: 您已更改选择的选项。要查看优化顾问程序向导中的新建议，必须再次运行优化顾问程序向导过程。如果不再次运行优化顾问程序向导过程，则会看到为较早的选项创建的的建议。是否要再次运行优化顾问程序向导过程？**

**解释**

运行优化顾问程序向导过程之后已更改所选的选项。要查看更新过的摘要表建议，再次运行优化顾问程序向导过程。如果不再次运行优化顾问程序向导过程，则会看到为较早的选项创建的的建议。

**用户响应**

单击是运行优化顾问程序向导过程。如果不想再次运行优化顾问程序向导过程，则单击否。

**10207: 未检测到维表。**

**解释**

未检测到维表。

**用户响应**

确保正确设置引用完整性约束。

**10208: OLAP 中心无法直接显示的 *Object\_names* 对象存在于数据库中。这些对象可能导致 OLAP 中心将来出现问题。单击是以删除对象，或单击否将对象保留在数据库中。**

解释

OLAP 中心检测到数据库中存在无法直接显示的多个对象（如层次结构或事实）。这些对象可能预先存在，也可能是在导入元数据之后创建的。这些对象可能导致 OLAP 中心中将来出现名称冲突和引用问题。除非您有保留这些对象的正当理由，否则建议您选择删除它们。

用户响应

单击是以删除对象，或单击否将对象保留在数据库中。

**10209: 导入操作期间出现意外错误。检查输入 XML 文件是否有错。**

解释

存储过程 API 在导入期间返回警告，并且在输出 XML 中没有内容。

用户响应

确保输入 XML 元数据符合 OLAP 元数据模式中定义的格式，并且 XML 文件定义其中引用的所有元数据对象。

**10210: 导入操作失败。存储过程 API 返回以下消息: *message*。**

解释

在导入过程期间，存储过程 API 返回一则错误消息。

用户响应

使用消息中提供的信息解决问题。如果无法解决问题，请与 IBM 软件支持人员联系。

**10211: 非数字量度 *measure\_name* 无法使用 *function\_name* 聚集函数，原因是函数期待数字自变量。**

解释

具有非数字数据类型的量度不能具有数字聚集函数。您只能选择 **MIN**、**MAX** 或 **COUNT** 作为非数字数据的聚集函数。

用户响应

选择另一聚集函数。

**10212: 无法从输入 XML 文件中读取对象。检查输入 XML 文件是否有错。**

解释

OLAP 中心未能从输入 XML 文件中读取对象。

#### 用户响应

确保输入 XML 元数据符合 OLAP 元数据模式中定义的格式，并且 XML 文件定义其中引用的所有元数据对象。

**10213: *file\_name* 输入 XML 文件在指定的目录不存在。**

#### 解释

输入 XML 文件在指定的目录中不存在。

#### 用户响应

确保输入 XML 文件存在于指定的目录中。

**10214: 导入文件中包含的 *object\_name* 对象引用数据库中不存在的 *column\_name* 列。确保在导入文件之前元数据对象引用的表和列在导入文件中存在。**

#### 解释

导入 XML 文件包含引用数据库中不存在的表和列的对象。

#### 用户响应

确保导入文件之前数据库中存在导入 XML 文件中对象所引用的表。

**10215: OLAP 中心无法运行优化顾问程序向导建议的 SQL 脚本。数据库返回以下信息: *message*。**

#### 解释

OLAP 中心无法执行优化顾问程序向导建议的 SQL 脚本。可能没有足够的特权来执行 SQL 脚本。

#### 用户响应

确保具有正确的权限来运行优化顾问程序的建议。OLAP 中心联机帮助的“权限和特权”主题中描述了所需的权限。有关优化立方体模型的信息，请参阅《DB2 Cube Views 安装与用户指南》。

**10216: 优化顾问程序的建议已成功保存在指定的文件中。**

#### 解释

已将建议的创建摘要表 SQL 脚本和刷新摘要表 SQL 脚本（如果适用）保存在指定的文件中。

#### 用户响应

无需操作。

**10217: 已成功创建摘要表及其索引。**

#### 解释

已成功在数据库中创建优化顾问程序建议的摘要表和索引。

## 用户响应

无需操作。

**10218:** 您已选择一个视图。优化顾问程序无法验证视图引用的表是否存在引用约束。

### 解释

当使用视图（这些视图引用未包含约束的表）为立方体模型创建摘要表时，优化可能无效。优化顾问程序无法检测视图所引用的表上是否存在约束。

## 用户响应

如果视图引用的表没有约束，而您要运行优化顾问程序，则可以：1. 不在立方体模型中使用视图。2. 在运行优化顾问程序之前为表创建约束。

**10219:** 取消优化顾问程序向导将废弃建议的 **SQL** 脚本而不保存它们。单击“是”以关闭优化顾问程序向导而不保存建议的 **SQL** 脚本。单击“否”以返回至优化顾问程序向导，使您可以保存建议的 **SQL** 脚本。

### 解释

在优化顾问程序向导上单击取消按钮将废弃建议的 **SQL** 脚本而不保存它们。

## 用户响应

单击是 以关闭优化顾问程序向导而不保存建议的 **SQL** 脚本，或者单击否 以返回至优化顾问程序向导，使您可以继续使用优化顾问程序向导并保存这些 **SQL** 脚本。

**10220:** 指定文件的格式是用于旧版本的 **DB2 Cube Views** 的。导入向导可将指定文件描述的元数据对象转换为当前版本的 **DB2 Cube Views**。单击“是”以转换指定文件描述的元数据对象并继续导入。单击“否”以停止导入，使您可以指定另一文件或关闭导入向导。

### 解释

指定的元数据源文件包含的 **XML** 的格式是用于较早版本的 **DB2 Cube Views** 的。导入向导可读取该文件并将其内容转换为当前版本的 **DB2 Cube Views** 理解的格式。

## 用户响应

单击是 以将指定源文件描述的元数据对象转换为当前 **DB2 Cube Views** 格式。“导入选项”页显示指定文件中转换为新版本的 **DB2 Cube Views** 的对象。单击否 以阻止导入向导转换指定文件描述的元数据对象。可指定另一元数据源文件或关闭导入向导。

**10221:** 建议的摘要表应该使用 **disk\_space\_size MB** 的磁盘空间。

### 解释

优化顾问程序建议的摘要表应该使用指定的磁盘空间。

## 用户响应

在运行建议的脚本之前，确保至少具有指定的磁盘空间。

**10222:** 是否要停止优化顾问程序？单击“是”以阻止优化顾问程序创建建议并返回它到目前为止获取的建议。单击“否”以允许优化顾问程序继续创建建议。

解释

在优化顾问程序创建优化建议时单击了停止。

用户响应

单击是 以阻止优化顾问程序创建建议并返回它到目前为止获取的建议。单击否 以允许优化顾问程序继续创建建议。如果单击是，顾问程序向导将显示到目前为止获取的建议。

**10300:** 未能对在 SQL 表达式中输入的 *measure\_name* 量度进行语法分析。

解释

指定的 SQL 表达式引用在给定上下文中无效或者数据库中不存在的量度。

用户响应

确保 SQL 表达式仅引用 SQL 表达式构建器的“数据”列表中出现的量度。

**10301:** 未能对在 SQL 表达式中输入的 *column\_name* 列进行语法分析。

解释

指定的 SQL 表达式引用在给定上下文中无效或者数据库中不存在的列。

用户响应

确保 SQL 表达式仅引用 SQL 表达式构建器的“数据”列表中出现的列。

**10302:** 未通过模式名来限定 *attribute\_name* 属性。

解释

SQL 表达式中属性的引用必须通过用“.”分隔的模式名来限定。

用户响应

确保 SQL 表达式中对属性的所有引用均通过用“.”分隔的模式名来限定。

**10303:** 未通过模式名来限定 *measure\_name* 量度。

解释

SQL 表达式中量度的引用必须通过用“.”分隔的模式名来限定。

用户响应

确保 SQL 表达式中对量度的所有引用均通过用“.”分隔的模式名来限定。

**10304: *object\_name*** 对象标记内缺少对象名称。

解释

指定的 SQL 表达式具有空列标记 @Column、空属性标记 @Attribute 或空量度标记 @Measure。

用户响应

确保对象类型标记 @Column, @Measure 和 @Attribute 均有一个括起来的对象名。

**10305: 指定的表达式无效。数据库返回以下信息: *message*。**

解释

SQL 表达式中存在语法错误。当 SQL 表达式引用那些不带括起来的标记的列、属性或量度时, 也会显示此错误。对列、属性或量度的引用必须分别括在 @Column(), @Attribute() 或 @Measure() 标记中。

用户响应

更正语法错误。确保每个列、属性和量度均括在正确的标记中。

**10306: 输入的表达式的数据类型为非数字。输入数字表达式作为第二个参数。**

解释

第二个参数的数据类型必须是数字。

用户响应

确保输入的表达式的数据类型产生数字数据类型。

**10307: 量度 *measure\_name*** 的表达式产生非数字数据类型。选择其表达式产生数字数据类型的量度。

解释

第二个参数的数据类型必须是数字。

用户响应

确保选择的量度的表达式的数据类型为数字。

**10308: OLAP 中心无法与指定数据库通信。这可能是因为没有为 **DB2 Cube Views** 正确配置数据库。配置数据库可能需要一些时间。单击“是”以配置指定数据库。如果现在不想配置指定数据库, 则单击“否”。**

解释

OLAP 中心可以使用提供的用户名和密码连接至数据库, 但它无法与存储过程 API 通信。

这可能是因为:

- DB2 Cube Views 存储过程 API 未向指定数据库注册。
- 对于指定数据库，DB2 Cube Views 目录表不存在。

#### 用户响应

单击是以便为 DB2 Cube Views 配置数据库，否则单击否。

**10309: OLAP 中心无法连接至指定数据库，原因是该数据库是为较早版本的 DB2 Cube Views 配置的。该数据库必须迁移至当前版本的 DB2 Cube Views。单击“是”以便让 OLAP 中心迁移指定数据库。如果不想 OLAP 中心迁移指定数据库，则单击“否”。**

#### 解释

OLAP 中心可以使用提供的用户名和密码连接至数据库，但它无法从目录检索元数据。这可能是由于 DB2 Cube Views 目录表是为较早版本的 DB2 Cube Views 配置的。

#### 用户响应

单击是将以将 DB2 Cube Views 目录迁移至适当版本，否则单击否。

**10310: 已成功配置 *database\_name* 数据库。**

#### 解释

OLAP 中心已成功创建 DB2 Cube Views 目录表并向指定数据库注册了存储过程 API。

#### 用户响应

无需操作。

**10311: OLAP 中心无法为 DB2 Cube Views 配置数据库。数据库返回以下信息: *information*。**

#### 解释

OLAP 中心无法为 DB2 Cube Views 配置指定数据库。

这可能是由于:

- OLAP 中心无法注册 DB2 Cube Views 存储过程 API。
- OLAP Center 无法创建一个或多个 DB2 Cube Views 目录表。

#### 用户响应

确保您具有 OLAP 中心联机帮助“权限和特权”主题中描述的正确设置和安装权限。有关配置数据库的信息，请参阅《DB2 Cube Views 安装与用户指南》。

**10312: 聚集验证失败。一个或多个指定的聚集函数与源 SQL 表达式不兼容。**

#### 解释

一个或多个指定的聚集函数与源 SQL 表达式不兼容。这可能因为指定的聚集函数期待数据类型与源 SQL 表达式数据类型不同的参数。

## 用户响应

确保聚集函数对于指定量度的源数据类型有效。

**10313: 量度的源表达式只有在包含“无”聚集设置时依照语法才是正确的。量度必须使用“无”聚集设置。**

### 解释

在下列情况下，量度期待“无”聚集设置：

- 当聚集函数未应用于其引用的量度时，SQL 表达式依照语法是不正确的；但是，当这些聚集函数应用于其引用的量度时，SQL 表达式依照语法是正确的。例如，char + int 依照语法不正确，但 COUNT(char) + SUM(int) 依照语法则正确。
- SQL 表达式使用 OLAP 函数，如 RANK()、DENSE\_RANK() 和 ROW\_NUMBER()。

## 用户响应

确保量度已应用了“无”聚集设置。

**10401: 表达式不能包括列函数、标量全查询或子查询。**

### 解释

SQL 表达式不能包括列函数、标量全查询或子查询。

## 用户响应

更正列函数的用法以删除无效的表达式。

**10501: 模式名不能以 *prefix* 开头。**

### 解释

模式名不能以“SYS”和“SESSION”开头。

## 用户响应

输入另一模式名。

**10502: 连接属性对于立方体模型性能优化无效。解决此问题，然后再次运行优化顾问程序向导。数据库返回以下信息：*message*。**

### 解释

连接属性对于立方体模型性能优化无效。

## 用户响应

通过应用优化验证规则为连接指定正确的设置。

**10503: 无法修改层次结构，原因是它具有关联的立方体层次结构。**

### 解释

如果层次结构存在立方体层次结构，则不能修改层次结构。



## 用户响应

在对层次结构进行更改之前，确保没有立方体层次结构引用要修改的层次结构。您也可以通​​过必需的修改创建另一层次结构。

**10504: 此量度必须使用“无”聚集设置，原因是它引用使用多参数聚集函数的量度。**

### 解释

只有使用“无”聚集设置的量度才能引用那些使用多参数函数的量度。不能将聚集函数设置从“无”更改为另一函数。

## 用户响应

可以执行下列其中一个操作:

- 不改变量度的聚集设置。
- 改变指定量度，以使其不使用多参数函数。

**10505: 此量度无法使用多参数函数，原因是使用聚集设置（“无”除外）的 *measure\_name* 量度引用此量度。**

### 解释

只有使用“无”聚集设置的量度才能引用那些使用多参数函数的量度。不能将正在编辑的聚集脚本更改为包括多参数函数，原因是正在编辑的量度由不使用“无”聚集设置的量度引用。

## 用户响应

可以执行下列其中一个操作:

- 不改变量度的聚集脚本。
- 改变指定的量度，以使其不引用正在编辑的量度。

**10506: 包含指定 SQL 表达式的现有聚集设置无效。OLAP 中心将聚集设置复位为 *setting*。**

### 解释

包含新 SQL 表达式的现有聚集设置无效，已复位为缺省聚集设置。这可能是因为:

- 源 SQL 表达式的数据类型已更改。
- 当前聚集设置期待为“无”。在下列情况下，它必须为“无”:
  - 当聚集函数未应用于其引用的量度时，SQL 表达式依照语法不正确，但是当这些聚集函数应用于其引用的量度时，SQL 表达式依照语法正确。例如，char + int 依照语法不正确，但 COUNT(char) + SUM(int) 依照语法则正确。
  - SQL 表达式使用 OLAP 函数，如 RANK()、DENSE\_RANK() 和 ROW\_NUMBER()。

## 用户响应

无需操作。

**10507: 立方体模型中的一个或多个维没有层次结构。它们将不可用于立方体的内含物。**

**解释**

对于要创建的立方体维，它必须基于至少具有一个层次结构的维。您正在试图创建或修改包含一个或多个维（没有层次结构）的立方体。在用于定义立方体维的选择列表中将省略这些维。

**用户响应**

在不引用省略的维的情况下创建或修改立方体，或者确保立方体模型中的每个维均具有层次结构。

**10508: 仅当对层次结构选择了两个层时，递归部署选项才有效。**

**解释**

递归部署必须恰好存在两个层。

**用户响应**

对递归部署恰好选择两个层或选择另一部署选项。

**10509: 未对优化片指定所有必需的属性。对于每个优化片，对每个立方体维指定查询类型和选项。**

**解释**

未对优化片指定所有必需的属性。

**用户响应**

对于每个优化片，对每个立方体维指定查询类型和选项。

**10510: 至少为该层选择一个层键属性。**

**解释**

没有为该层指定层键属性。

**用户响应**

至少为该层选择一个层键属性。

**10511: 不能对同一立方体同时指定 MOLAP 抽取类型的片和混合抽取类型的片。必须从立方体中除去其中一个片。**

**解释**

一个立方体不能同时具有 MOLAP 抽取类型和混合抽取类型的优化片。只能对每个立方体指定一种类型的抽取优化片。

**用户响应**

从立方体中除去 MOLAP 抽取片或混合抽取片。

**10512: 对于每个立方体，最多允许一个 MOLAP 抽取类型的优化片。**

**解释**

对于每个立方体，最多允许一个 MOLAP 抽取类型的优化片。

**用户响应**

确保只有一个 MOLAP 抽取类型的优化片。

**10513: 对于每个立方体，最多允许一个混合抽取优化片。**

**解释**

对于每个立方体，最多允许一个混合抽取优化片。

**用户响应**

确保只有一个混合抽取类型的优化片。

**10514: 仅当已对立方体定义了混合抽取类型的优化片时，才能创建深入钻取类型的优化片。**

**解释**

混合抽取优化片是指定深入钻取类型的优化片所必需的。

**用户响应**

创建混合抽取类型的优化片。然后就可以创建深入钻取类型的优化片。

**10515: DB2 Cube Views 指定 ANY 作为每个现有优化片的新立方体维的优化级别。**

**解释**

如果在为立方体定义优化片之后将立方体维添加至该立方体，DB2 Cube Views 将通过指定 Any 作为新的立方体维的优化级别来扩展现有优化片。

**用户响应**

可通过单击“立方体属性”窗口的“查询类型”页上的“指定”来修改这些片的优化级别。

**10516: 必须至少在“优化片”窗口中指定一个优化片，原因是您对立立方体的查询类型指定了高级设置。**

**解释**

必须至少在“优化片”窗口中指定一个优化片，原因是您对立立方体的查询类型选择了高级设置。

**用户响应**

在“优化片”窗口中单击**指定**并至少创建一个优化片。

**10517:** 对立方体和立方体维的当前更改将在离开“维”页时保存下来。单击“是”以保存数据库中的更改。单击“否”以留在“维”页上进行其它更改或取消。

**解释**

在“立方体属性”窗口中修改立方体维并离开“维”页后，对立方体维进行的所有更改将保存在数据库中，不能通过单击取消来撤销更改。

**用户响应**

单击是保存当前更改。单击否以取消该事务并留在“维”页上。

**10518:** 不能修改立方体层次结构，原因是您已经对 **cube\_name** 立方体定义了高级优化片。必须删除所有高级优化片，然后才能修改立方体层次结构。

**解释**

如果已经对立方体定义了高级优化片，则不能修改立方体层次结构。

**用户响应**

必须删除所有高级优化片，然后才能修改立方体层次结构。

**10519:** 不能修改立方体维的集合，原因是您已经对 **cube\_name** 立方体定义了高级优化片。必须删除所有高级优化片，然后才能对立方体添加或删除立方体维。

**解释**

如果已经对立方体定义了高级优化片，则不能修改立方体维集合。

**用户响应**

必须删除所有高级优化片，然后才能对立方体添加或删除立方体维。

**10520:** 对属性的更改还将修改立方体的现有优化片。可复查优化片以查看和修改更改，或者可在属性窗口上单击“取消”以关闭窗口并取消更改。

**解释**

如果修改立方体维集合或立方体层次结构，则该立方体的现有高级优化片将以下列其中一种方式进行修改：

- 如果已添加新立方体维，则所有现有优化片将扩展为包括处于任何层的新立方体维。
- 如果已除去立方体维，还将从所有现有优化片中除去相应层。
- 如果立方体层次结构已改变以致在优化片中定义的层不存在，则缺少的层次结构层在所有适当的优化片中将被替换为任何层。
- 如果现有立方体层次结构被替换为新的立方体层次结构，则优化片中对相应立方体维的层将设置为“任何”。

## 用户响应

打开“优化片”窗口并验证 OLAP 中心的修改。进行所有必要的更改。

---

## DB2 和 DB2 Cube Views 中的状态消息

当调用 DB2 Cube Views 存储过程时，无论是否执行了该存储过程，DB2 都会将 SQLCODE 和 SQLSTATE 返回至调用应用程序。如果 DB2 Cube Views 存储过程可以执行，存储过程将返回状态消息，作为发送到调用应用程序的 XML 数据的一部分。

下表显示元数据操作返回的状态消息与 DB2 为调用存储过程返回的 SQLSTATE 之间的关系。

表 63. 元数据操作标识与 SQLSTATE 代码对照

SQL CODE	SQL STATE	元数据操作状态消息标识	元数据操作状态消息类型	返回的元数据操作状态消息
0	0	0 2	信息性	无
0	0	1	信息性	有
0	0	599 6006 6299 7200 7201 7202	警告	无
462	01HQ1	0 - 7999 (不包括其它行中列示的标识)	错误	无
443	38Q00	不适用	不适用	不适用
443	38Q01	不适用	不适用	不适用
443	38Q02	不适用	不适用	不适用
443	38Q03	不适用	不适用	不适用



---

## 声明

此信息是为在美国提供的产品和服务编写的。

IBM 可能在其他国家或地区不提供本文中讨论的产品、服务或功能特性。有关您当前所在区域的产品和服务的信息，请向您当地的 IBM 代表咨询。任何对 IBM 产品、程序或服务的引用并非意在明示或暗示只能使用 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，都可以代替 IBM 产品、程序或服务。但是，评估和验证任何非 IBM 产品、程序或服务，则由用户自行负责。

IBM 公司可能已拥有或正在申请与本文档内容有关的各项专利。提供本文档并未授予用户使用这些专利的任何许可。您可以用书面方式将许可查询寄往：

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

有关双字节（DBCS）信息的许可查询，请与您所在国家或地区的 IBM 知识产权部门联系，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

**本条款不适用英国或任何这样的条款与当地法律不一致的国家或地区：** International Business Machines Corporation 以“按现状”的基础提供本出版物，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）对非侵权性、适销性和适用于某特定用途的默示保证。某些国家或地区在某些交易中不允许免除明示或默示的保证。因此，本条款可能不适用于您。

本信息中可能包含技术方面不够准确的地方或印刷错误。此处的信息将定期更改；这些更改将编入本资料的新版本中。IBM 可以随时对本资料中描述的产品和 / 或程序进行改进和 / 或更改，而不另行通知。

IBM 可以按它认为适当的任何方式使用或分发您所提供的任何信息，而无须对您承担任何责任。

本程序的被许可方如果要了解有关程序的信息以达到如下目的：（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，以及（ii）允许对已经交换的信息进行相互使用，请与下列地址联系：

IBM Corporation  
J46A/G4  
555 Bailey Avenue  
San Jose, CA 95141-1003  
U.S.A.

只要遵守适当的条件和条款，包括某些情形下的一定数量的付费，都可获得这方面的信息。

本资料中描述的许可程序及其所有可用的许可资料均由 IBM 依据 IBM 客户协议、IBM 国际程序许可协议或任何同等协议中的条款提供。

涉及非 IBM 产品的信息可从这些产品的供应商、其出版说明或其他可公开获得的资料中获取。IBM 没有对这些产品进行测试，也无法确认其性能的精确性、兼容性或任何其他关于非 IBM 产品的声明。有关非 IBM 产品性能的问题应当向这些产品的供应商提出。

本资料中包含用于日常业务运作的数据和报表的示例。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有雷同，纯属巧合。

此处包含的任何性能数据都是在受控环境中测得的。因此，在其他操作环境中获得的数据可能会有明显的不同。有些测量可能是在开发级的系统上进行的，因此不保证与一般可用系统上进行的测量结果相同。此外，有些测量是通过推算而估计的，实际结果可能会有差异。本文档的用户应当验证其特定环境的适用数据。

涉及非 IBM 产品的信息可从这些产品的供应商、其出版说明或其他可公开获得的资料中获取。IBM 没有对这些产品进行测试，也无法确认其性能的精确性、兼容性或任何其他关于非 IBM 产品的声明。有关非 IBM 产品性能的问题应当向这些产品的供应商提出。

如果您查看的是该资料的软拷贝，则照片和彩色图形可能不会出现。

---

## 商标

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:

AIX  
DB2  
DB2 Connect  
DB2 Universal Database  
IBM  
Office Connect  
Redbooks

以下各项是其他公司的商标或注册商标:

Microsoft、Windows、Windows NT、Windows 2000、Windows XP 和 Microsoft Excel 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。

UNIX 是在美国和 / 或其他国家或地区的注册商标，且经 X/Open Company Limited 排他许可。

Linux 是 Linus Torvalds 的注册商标。Red Hat 和所有基于 Red Hat 的商标和徽标是 Red Hat, Inc. 在美国和其他国家或地区的商标或注册商标。



其他公司、产品或服务名称是其他公司的商标或服务标记。



---

## 词汇表

本词汇表定义此书中使用的术语。

### [B]

**标准部署 (standard deployment)**：使用层次结构的层定义，其中层次结构中的每个属性均定义一个层。例如，Time 维的平衡层次结构通常会依据每个定义的层（包括 Year、Quarter 和 Month）进行组织。标准部署可用于全部四种层次结构类型。

### [C]

**层次结构 (hierarchy)**：在立方体模型特定维中定义一个或多个属性之间关系的 DB2 对象。DB2 Cube Views 支持四种类型的层次结构：平衡、未平衡、未对齐和网络。层次结构可以按标准或递归方式部署。

**层 (level)**：一种 DB2 对象，由具有相关性且在层次结构的排序中作为一个逻辑步骤一起工作的一个或多个属性组成。

### [D]

**递归部署 (recursive deployment)**：使用层次结构属性之间固有的父子关系。使用递归部署的未平衡层次结构表现为父子属性对。

### [G]

**功能相关性 (functional dependency)**：一种 DB2 对象，指示层对象的缺省属性和相关属性在功能上是由层的键属性确定的。功能相关性允许您指定一个或多个列在功能上与一个或多个其它列相关（假定所有列都存在于同一个表中）。

### [H]

**混合立方体 (hybrid cube)**：包含多维数据并引用关系数据，使您可以在基本表中查询较低级别的数据。

### [J]

**计算的量度 (calculated measure)**：包含使用 OLAP 中心中的“表达式构建器”或通过 SQL 创建的内置计算。计算的量度具有执行计算的 SQL 表达式，不会映射到单个列或属性。

**聚集函数 (aggregation function)**：其中一种 DB2 SQL 聚集函数，如 SUM、AVG、MIN 和 MAX。聚集函数用于控制如何对量度执行积累。

**具体查询表 (materialized query table)**：一种表，其定义基于查询结果，数据采用取自具体查询表定义所基于的一个或多个表的预计算结果的格式。

### [L]

**立方体层次结构 (cube hierarchy)**：一种 DB2 对象，它是立方体维的一部分，从与立方体维对应的维中的层次结构派生。立方体层次结构引用层次结构（从中派生立方体层次结构）的属性的子集，其中属性的顺序必须与它们在层次结构中的顺序相同。

**立方体层 (cube level)**：一种 DB2 对象，它是层的子集并在立方体中使用。立方体层引用从中派生它的层（父层）并继承对父层定义的层键属性和缺省属性。

**立方体模型 (cube model)**：描述与量度集合相关的所有数据的 DB2 对象。通常情况下，立方体模型与数据库中的星型模式或雪花模式相关。立方体模型引用单个事实对象和一个或多个维。可以优化立方体模型，以提高向星型模式立方体模型或雪花模式立方体模型发出的 SQL 查询的性能。

**立方体事实 (cube facts)**：一种 DB2 对象，它是立方体的一部分，从与立方体对应的立方体模型中的维派生。立方体事实引用事实对象（从中派生立方体事实）的量度的子集。

**立方体维 (cube dimension)**：一种 DB2 对象，它是立方体的一部分，从与立方体对应的立方体模型中的维派生。立方体维引用维（从中派生立方体维）的属性的子集。它还引用单个立方体层次结构。

**立方体 (cube)**：从立方体模型中派生的 DB2 对象。立方体事实和立方体维是立方体模型中引用的立方体事实和立方体维的子集。由于立方体维仅允许每个立方体维具有一个立方体层次结构，因此立方体适用于不使用多个层次结构的工具 and 应用程序。

**连接 (join)**：连接两个关系表。连接引用属性，属性又引用将要连接的表中的列。最简单的连接形式引用两个属性：一个映射至第一个表中的列，另一个映射至第二个表中的列。连接也可以包含运算符，以指示如何比较列。连接对象还可以用于使组合连接模型化，其中第一个表中的两个或多个列连接至第二个表中相同数目的列。组合连接

使用属性对映射相应的列。每对属性都具有一个运算符，它指示如何比较这些列对。连接还可以具有类型和基数。连接可用于在维中将维表连接在一起或在立方体模型中将立方体模型的维连接至其事实对象，或者在事实对象中连接多个事实表。

**量度 (measure)**：定义量度实体的一种 DB2 对象，在事实对象中使用。量度在维的上下文内变得有意义。常见的量度对象示例为 Revenue、Cost 和 Profit。

## [M]

**模式 (schema)**：在 SQL-92 标准中，单个用户拥有的、形成单个名称空间的数据库对象的集合。名称空间是不能具有重复名称的对象集。例如，两个表只有在独立的模式中才能具有相同名称，同一模式中的两个表不能具有相同的名称。

## [P]

**片 (slice)**：多维数据库或立方体的一个区域。

**平衡层次结构 (balanced hierarchy)**：具有一致深度的包含有意义的层和分支的层次结构。每个属性的逻辑父属性均在其正上方的层中。请参阅网络层次结构 (*network hierarchy*)、未对齐层次结构 (*ragged hierarchy*) 和未平衡层次结构 (*unbalanced hierarchy*)。

## [S]

**事实表 (fact table)**：数据库模式中的中央表，包含将事实与维表相关的数字量度和键。事实表包含描述业务中特定事件（如银行交易或产品销售）的数据。

**事实对象 (facts object)**：将对特定应用程序有意义的相关量度进行分组的 DB2 对象。事实对象存储事实-到-维连接中使用的属性的有关信息，以及用于映射跨多个数据库表的附加量度的属性和连接的有关信息。因此，除一组量度之外，事实对象还存储一组属性和一组连接。事实对象在立方体模型中用作星型模式的中心。

**属性关系 (attribute relationship)**：概括地描述属性对象的关系。关系是由左边的属性和右边的属性、类型、基数以及是否确定功能相关性来描述的。类型描述右边属性的角色是什么（就左边属性而论）。存在两种可能的类型：“描述性的”和“关联的”。“描述性的”类型指定右边属性是左边属性的描述符。

**属性 (attribute)**：一种 DB2 对象，它映射至表中单个列或表达式（一组列和 / 或其它属性的组合）。属性可以充当多种角色。例如，它可以是对立方体中数据的引用，也可以是对连接或其它属性关系使用的列的引用。

## [W]

**网络层次结构 (network hierarchy)**：一种层次结构，其中未指定各层的顺序，但各层具有语义含义。由于属性层没有固有的父子关系，因此层的顺序并不重要。请参阅平衡层次结构 (*balanced hierarchy*)、未对齐层次结构 (*ragged hierarchy*) 和未平衡层次结构 (*unbalanced hierarchy*)。

**维表 (dimension table)**：数据仓库中的一种表，其条目描述事实表中的数据。维表包含从中创建维的数据。

**维 (dimension)**：引用相关属性（描述一组量度的某些特征）的集合的 DB2 对象。维可以引用一个或多个维表中的属性。但是，如果使用多个维表中的属性，表之间必须具有连接，并且这些连接必须由维引用。维还引用一个或多个层次结构，并且可以引用属性之间的关系。

**未对齐层次结构 (ragged hierarchy)**：一种层次结构，其中每个层都具有一致的含义，但分支却由于分支层中的至少一个成员属性空缺而具有不一致的深度。请参阅平衡层次结构 (*balanced hierarchy*)、网络层次结构 (*network hierarchy*) 和未平衡层次结构 (*unbalanced hierarchy*)。

**未平衡层次结构 (unbalanced hierarchy)**：一种层次结构，其中各层具有一致的父子关系，但特定层中的所有成员的语义含义均不一致。同时，层次结构分支具有不一致的深度。请参阅平衡层次结构 (*balanced hierarchy*)、网络层次结构 (*network hierarchy*) 和未对齐层次结构 (*ragged hierarchy*)。

## [X]

**星型连接 (star join)**：事实表（通常为较大的事实表）与至少两个维表之间的连接。事实表在维键上与每个维表连接。

**星型模式 (star schema)**：一种关系数据库结构，在模式中央的单个事实表中维护数据，其它维数据存储在维表中。每个维表直接与事实表相关，通常由键列连接至事实表。星型模式用于数据仓库中。

**雪花模式 (snowflake schema)**：星型模式的扩展，由多个表定义一个或多个维。在雪花模式中，只有主维表连接至事实表。其它维表连接至主维表。

## [Y]

**元数据 (metadata)**：有关数据属性的信息，如列中数据的类型（数字、文本等等）或列的长度。它也可以是有关数据结构的信息，或者是指定对象设计（如立方体或维）的信息。

**约束 (constraint)**：数据库管理器强制实行的规则。有四种类型的约束：唯一、引用、表检查和信息性。

## [Z]

**摘要表 (summary table)**：包含立方体模型所使用基本表的聚集数据。DB2 Cube Views 使用 DB2 摘要表提高向立方体模型发出的查询的性能。摘要表是一种特殊类型的具体查询表 (MQT)，该表专门包括摘要数据。因为 DB2 Cube Views 始终建议使用包含摘要数据的 MQT，因此 DB2 Cube Views 文档中使用术语摘要表来描述建议的 MQT。请参阅具体查询表 (*materialized query table*)。

**支架表 (outrigger table)**：雪花模式中不是维中主维表的任意维表。

**主维表 (primary dimension table)**：在雪花模式中，连接至事实表的维表。

## M

**MQT**：请参阅具体查询表 (*materialized query table*)。



# 索引

## [ A ]

- 安装
  - AIX 5
  - Linux 5
  - Solaris Operating System 5
  - Windows 5
- 安装需求 4

## [ B ]

- 报表查询 84
- 标准部署 22

## [ C ]

- 操作参数 131
  - 导入方式 132
  - 验证方式 134
  - objectType 131
  - recurse 132
  - restriction 132
- 操作的操作数 134
- 操作数 134
- 层
  - 层键属性 23
  - 创建 45
  - 非理想建模 25
  - 关系映射 10, 13
  - 基本规则 34
  - 理想建模 24
  - 描述 23
  - 缺省属性 24
  - 属性 26
  - 添加至立方体层次结构 47
  - 为优化建模 83
  - 相关属性 24
- 层次结构
  - 部署 22
  - 创建 44
  - 创建层 45
  - 关系映射 10, 15
  - 基本规则 34
  - 类型 20
  - 描述 20
  - 属性 23
  - 为优化建模 83
- 层次结构向导 44
- 层键属性
  - 描述 23
- 层向导 45

- 查询
  - 报表 84, 86
  - 捕获 102
  - 重新路由 74, 76, 78
  - 混合抽取 88
  - 类型 83
  - 深入钻取 89
  - 下寻 84, 85
  - 制定基准 102
  - DB2EXPLAIN 102
  - MOLAP 抽取 84, 87
- 重命名操作 124
- 重新路由查询 74, 78
- 创建操作 123
- 存储过程
  - md\_message 115
- 错误处理 114
- 错误日志记录 137

## [ D ]

- 代码页 138
- 导出
  - 元数据对象 41
- 导出窗口 41
- 导入
  - 元数据对象 41
- 导入操作 126
- 导入方式 132
- 导入向导 41
- 递归部署 23

## [ F ]

- 非理想建模 25
- 服务器
  - 联合 107

## [ G ]

- 改变操作 124
- 概述
  - 元数据对象 9
- 跟踪 136
  - 日志记录 136
- 功能相关性
  - 摘要表 78
  - 指定 45
- 关联
  - 广告 56

- 关联 (续)
  - 销售 56
- 关系表 9
- 管理操作 130
- 规则
  - 基本 93
  - 立方体模型完整性 93
  - 优化 93

## [ J ]

- 基本规则 93
  - 层 34
  - 层次结构 34
  - 立方体 35
  - 立方体层 36
  - 立方体层次结构 36
  - 立方体模型 33
  - 立方体事实对象 36
  - 立方体维 36
  - 连接 35
  - 量度 34
  - 事实对象 33
  - 属性 35
  - 属性关系 35
  - 维 34
- 基数 29
- 计数 65
- 计算
  - 边际利润 61
  - 利润 61
  - 流 51
  - 值 51
- 计算的量度 44
- 简单量度 43
- 检索操作 129
- 检索结果 118
- 建模
  - 非理想 25
  - 理想 24
- 将销售数字分等级
  - DENSERANK 66
  - RANK 66
  - ROWNUMBER 66
- 具体查询表 74

## [ K ]

- 快速启动向导 42

## [ L ]

- 理想建模 24
- 立方体 47
  - 创建 47
  - 关系映射 16
  - 基本规则 35
  - 描述 30
  - 属性 31
  - 为优化建模 83
- 立方体层
  - 基本规则 36
  - 描述 32
  - 属性 32
- 立方体层次结构
  - 关系映射 16
  - 基本规则 36
  - 立方体层次结构
    - 指定 47
  - 描述 32
  - 属性 32
  - 添加至立方体维 47
  - 指定 47
- 立方体模型
  - 层次结构, 创建 44
  - 除去维 48
  - 创建 42, 43
  - 关系映射 9
  - 基本规则 33
  - 描述 18
  - 使用快速启动向导创建 42
  - 事实对象, 创建 43
  - 属性 19
  - 完整性规则 33, 93
  - 维, 创建 44
  - 维, 添加 45
  - 优化 98
  - 优化规则 33
- 立方体模型向导 43
- 立方体事实对象
  - 关系映射 16
  - 基本规则 36
  - 描述 31
- 立方体维
  - 关系映射 16
  - 基本规则 36
  - 描述 31
  - 属性 32
  - 添加立方体层次结构 47
- 立方体向导 47
- 立即刷新摘要表 104
- 联合
  - 服务器 107
  - 数据库 108
- 联合系统 108
  - 对 DB2 Cube Views 启用 109

- 联合系统 (续)
  - 概述 107
  - 客户机 107
  - 联合服务器 107
  - 联合数据库 107
  - 昵称 110
  - 数据源 107
  - 远程数据源 109, 110
  - 约束 110
- 连接
  - 创建 46
  - 关系映射 11, 13
  - 基本规则 35
  - 基数 94
  - 类型 94
  - 描述 30
  - 事实-到-维 93
  - 属性 30
  - 优化规则 35
- 连接向导 46
- 量度
  - 关系映射 9
  - 基本规则 34
  - 计算的 44
  - 简单 43
  - 描述 26
  - 属性 27
  - 为优化建模 83
  - 在摘要表中 77

## [ M ]

- 描述操作 118
- 模式
  - 星型 9
  - 雪花 9, 95

## [ N ]

- 内存管理 114
- 昵称 110

## [ P ]

- 配置文件 137, 141
- 片 75
  - 优化 84
- 平衡层次结构 20

## [ Q ]

- 桥接 3
- 权限和特权 38
- 缺省属性
  - 描述 24

## [ R ]

- 日志记录
  - 跟踪 136
- 日志记录错误 137

## [ S ]

- 删除操作 125
- 删除元数据对象 49
- 使用 DB2 Cube Views 108
- 事实对象
  - 创建 43
  - 关系映射 9
  - 基本规则 33
  - 描述 19
  - 属性 19
- 事实向导 43
- 事实-到-维连接 46, 93
  - 创建 45
- 事实-到-维连接, 创建 44
- 事务 114
- 数据库
  - 远程 108
- 数据库连接窗口 40
- 数据源
  - 远程 108, 109
- 属性
  - 层 26
  - 层次结构 23
  - 层键 23
  - 公共 17
  - 关系映射 10, 13
  - 基本规则 35
  - 立方体 31
  - 立方体层 32
  - 立方体层次结构 32
  - 立方体模型 19
  - 立方体事实 31
    - 属性 31
  - 立方体维 32
  - 连接 30
  - 量度 27
  - 描述 28
  - 缺省值 24
  - 事实对象 19
  - 属性 28
  - 属性关系 29
  - 维 20
  - 相关 24
  - 在摘要表中 77
- 属性关系
  - 基本规则 35
  - 类型
    - 关联的 29
    - 描述性的 29



属性关系 (续)  
描述 29  
属性 29  
在摘要表中 77

## [ T ]

添加维向导 45

## [ W ]

完整性规则  
立方体模型 33  
网络层次结构 22  
维  
层次结构, 创建 44  
创建 44  
从立方体模型中除去 48  
关系映射 10, 13, 15  
基本规则 34  
描述 19  
属性 20  
添加 45  
优化规则 34  
维向导 44  
未对齐层次结构 21  
未平衡层次结构 20

## [ X ]

系统配置 114  
系统需求 4  
下寻查询 84  
先决条件 5  
相关属性  
描述 24  
添加至立方体层次结构 47  
信息性约束 74, 93, 98  
星型模式 9  
修改操作 130  
需求  
软件 4  
硬件 4  
雪花模式 9, 12, 95

## [ Y ]

延迟刷新摘要表 104  
过期 103  
验证操作 127  
验证方式 134  
样本  
数据库文件 145  
应用程序文件 146

样本 (续)  
API 参数  
管理操作 130  
检索操作 129  
修改操作 130  
优化 78  
查询类型 97  
磁盘空间局限性 97  
规则 93  
过程 80  
立方体模型 98  
片 97  
时间局限性 98  
数据采样 98  
通过摘要表 74  
性能基准 81  
制定基准 102  
SQL 脚本 99  
优化顾问程序向导 74, 82, 83, 97, 98, 104  
优化规则  
立方体模型 33  
连接 35  
维 34  
优化片 83, 84, 91  
报表 86  
层 84  
混合抽取 88  
深入钻取 89  
下寻 85  
指定 47  
All 84  
Any 84  
MOLAP 抽取 87  
优化片, 指定 47  
元数据操作  
重命名 124  
创建 123  
导入 126  
改变 124  
描述 118  
删除 125  
验证 127  
转换 128  
元数据对象 3  
层 23  
层次结构 20  
层次结构, 创建 44  
层, 创建 45  
常规属性 17  
导出 41  
导入 41  
概述 9  
格式 142  
基本规则 33  
交换 40

元数据对象 (续)  
立方体 30  
立方体层 32  
立方体层次结构 32  
立方体模型 18  
立方体模型, 创建 43  
立方体模型, 使用快速启动向导创建 42  
立方体事实对象 31  
立方体维 31  
立方体, 创建 47  
连接 30  
连接, 创建 46  
量度 26  
命名约定 18  
事实对象 19  
事实对象, 创建 43  
属性 28  
属性关系 29  
维 19  
为优化建模 83  
维, 创建 44  
维, 添加 45  
元数据桥接 3  
远程  
数据源 108  
远程数据源  
定义 109  
约束 74, 98, 110  
外键 93  
信息性 93  
摘要表 78  
运行时跟踪 136

## [ Z ]

摘要表 74, 80, 98  
创建 82  
功能相关性 78  
立即刷新 104  
删除 83, 104  
维护 82, 104  
延迟刷新 103, 104  
约束 78  
SQL 脚本 99  
支架表 95  
转换操作 128

## A

API (应用程序编程接口)  
参数 117  
存储过程 115  
概述 113  
跟踪 136

API (应用程序编程接口) (续)

数据交换 113

API 中的消息结构 134

## C

currentRef 操作数 134

CVSAMPLE 数据库 145, 146

## D

DB2 数据库

断开连接 40

连接至 40

删除元数据对象 49

DB2 优化器 74, 78

DB2 EXPLAIN 设施 76

DB2 SQL 快照监视器 102

db2batch 基准工具 102

## M

md\_message

存储过程 115

MOLAP 抽取查询 84

## N

newRef 操作数 134

## O

object 操作数 134

objectType 参数 131

## R

recurse 参数 132

restriction 参数 132

## S

SQL 脚本 98, 99

## T

Time 维 69

## X

XML 语法分析 114

---

## 与 IBM 联系

如果您有技术问题，在与 DB2 Cube Views 客户支持人员联系之前，请复查并执行产品文档建议的操作。本指南建议您收集以帮助 DB2 Cube Views 客户支持人员为您提供更好服务的信息。

要获得信息或订购任何 DB2 Cube Views 产品，请与当地分支机构的 IBM 代表联系，或者与任何授权的 IBM 软件经销商联系。

如果您居住在美国境内，则可以拨打以下其中一个号码：

- 1-800-237-5511 以获得客户支持
- 1-888-426-4343 以了解可用的服务选项

---

## 产品信息

如果您居住在美国境内，则可以拨打以下其中一个号码：

- 1-800-IBM-CALL (1-800-426-2255) 或 1-800-3IBM-OS2 (1-800-342-6672) 以订购产品或获得一般信息。
- 1-800-879-2755 以订购出版物。

**<http://www.ibm.com/software/data/db2/db2md/>**

提供指向有关 DB2 Cube Views 的信息的链接。

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

此 DB2 通用数据库 Web 页面提供有关新闻、产品描述、培训计划以及更多方面的最新信息。

**<http://www.elink.ibm.link.ibm.com/>**

单击 Publications 以打开 International Publications 订购 Web 站点，该站点提供有关如何订购书籍的信息。

**<http://www.ibm.com/education/certify/>**

IBM Web 站点中的专业认证计划提供了多种 IBM 产品的认证考试信息。

**注：**在某些国家或地区，IBM 授权经销商应与他们的代理商支持机构（而不是 IBM 支持中心）联系。

---

## 对文档的意见

您的反馈将帮助 IBM 提供高质量的信息。请发送您对本书或其它 DB2 Cube Views 文档的任何意见。您可以使用以下任何方法提供意见：

- 使用在线读者意见表发送您的意见，网址为：[www.ibm.com/software/data/rcf](http://www.ibm.com/software/data/rcf)。
- 通过电子邮件将您的意见发送至 [ctscrcf@cn.ibm.com](mailto:ctscrcf@cn.ibm.com)。确保包括产品的名称、产品的版本号以及书籍的名称和部件号（如果适用）。如果对特定文本有意见，请包括文本的位置（例如，标题、表号或页码）。







程序号: 5724-E15

中国印刷

S152-0532-01

