

IBM DB2 Cube Views



业务建模方案

版本 8

注意

注意：在使用本资料及其支持的产品之前，请阅读第 29 页的『声明』中的信息。

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

您可以在线方式或通过当地 IBM 代表订购 IBM 出版物。

- 要以在线方式订购出版物，可访问 IBM Publications Center，网址为：www.ibm.com/shop/publications/order
- 要查找您当地的 IBM 代表，可访问 IBM Directory of Worldwide Contacts，其网址为：www.ibm.com/planetwide

要从美国或加拿大的 DB2 市场营销部门订购 DB2® 出版物，请致电 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	第 4 章 计算因特网订单数	17
本书的适用对象	v	方案的详细信息	17
第 1 章 计算一段时间以来仓库中的库存流动和 价值	1	创建量度的步骤	18
方案的详细信息	1	第 5 章 对销售数字分等级	21
创建量度的步骤	3	方案的详细信息	21
第 2 章 使广告成本与销售相关	7	创建量度的步骤	22
方案的详细信息	7	第 6 章 使用事实表中存储的时间数据来创建 Time 维	25
创建量度的步骤	7	方案的详细信息	25
第 3 章 计算商店的利润和利润率	13	创建属性和维的步骤	26
方案的详细信息	13	声明	29
创建量度的步骤	13	商标	30

关于本书

本书提供《DB2 Cube Views 安装与用户指南》的补充信息，即有关如何使用 DB2 Cube Views 元数据对常用实际业务方案建模的信息。

本书的适用对象

如果您是使用 OLAP 元数据和 DB2 Universal Database™(DB2) 的数据库管理员 (DBA)，请阅读本书。您应熟悉：

- OLAP 概念（例如，星型模式）
- DB2 Cube Views 元数据对象（例如，立方体模型、事实对象、维、连接、量度和属性）

第 1 章 计算一段时间以来仓库中的库存流动和价值

零售企业 XYZ Retail 在将库存发送至特定商店销售之前将库存保存在仓库中。XYZ Retail 保留有关一段时间以来仓库中的库存状态的数据并且想要分析此数据。公司尤其想要检查它的库存的两个方面：

- 商品流进和流出仓库的情况
- 仓库中的商品在给定时间的价值

第一个方面，商品的流动涉及查看一段时间以来的数据。第二个方面，商品的价值生成仓库在特定时间点的快照。

方案的详细信息

XYZ Retail 具有包含下列与仓库有关的列的事实表：QUANTITY_IN、QUANTITY_OUT、CURRENT_QUANTITY、PRODUCT_VALUE、PRODUCT_ID 和 TIME_ID。此数据是以周为单位输入到表中的。数据库还具有“产品”表和“时间”表。例如，一组样本事实表数据显示在表 1 中。

表 1.

PRODUCT_ID	TIME_ID	QUANTITY_IN	QUANTITY_OUT	CURRENT_QUANTITY	PRODUCT_VALUE
1234	1	5	0	5	5
1234	2	20	10	15	5
1234	3	10	20	5	5

三个样本数据条目中的每一个的 PRODUCT_ID 值都是一样的，原因是一种产品类型可以进出仓库多次。

XYZ Retail 的 DBA 必须创建三个不同的量度：

流进 对商品流进仓库的情况建模。

流出 对商品流出仓库的情况建模。

当前价值

对给定时间的商品价值建模。

要创建前两个量度（“流进”和“流出”），DBA 创建分别映射至 QUANTITY_IN 和 QUANTITY_OUT 列的量度并对所有维的数据求和。这又称为全加量度，原因是仅对所有维使用 SUM 函数来聚集数据。例如，表 2 显示 PRODUCT_ID 为 1234 的产品的 QUANTITY_IN 和 QUANTITY_OUT 列三个月来的一组样本数据。“流进”和“流出”量度对这些按月计的值求和以计算该季度以来进出仓库的总数量。

表 2. 计算 PRODUCT_ID 1234 的“流进”和“流出”全加量度的样本数据

	一月	二月	三月	第一季度
QUANTITY_IN	5	20	10	35
QUANTITY_OUT	0	10	20	30

全加量度是要创建的最简单而且最常用的量度，它们通常用作较复杂量度的构建块。对于基于数字源数据的量度，缺省情况下，OLAP 中心会创建全加量度。

为创建第三个量度“当前价值”，DBA 会创建通过将 PRODUCT_VALUE 乘以 CURRENT_QUANTITY 来计算价值的计算的量度。例如，如果 PRODUCT_ID=1234 的产品价值为 5，于是样本数据的“当前价值”量度显示在表 3 中。

表 3. 计算 PRODUCT_ID 为 1234 的产品的“当前价值”量度的样本数据

	一月	二月	三月
CURRENT_QUANTITY	5	10	20
当前价值	25	50	100

然后必须对这些维聚集此数据。但是，因为此量度计算特定时间点的价值，所以对 Time 维求和没有意义。相反，聚集将对“产品”维的数据求和并求得一段时间以来的数据的平均值。这又称为半加量度，原因是只有聚集的部分涉及 SUM 函数。

计算快照数据（表示特定时刻的数据，例如，月库存数据）的量度通常是半加量度，原因是将月份加入季度没有意义。如果产品整个季度都保留在仓库中，则该产品包括在该季度三个月的每月仓库库存的 CURRENT_QUANTITY 快照数据中。如果 CURRENT_QUANTITY 数据是对一段时间以来的数据求和得到的，则会对在仓库中保留了三个月的产品计数三次。如第 3 页的表 4 中所示，第一季度的值 25 对于仓库的活动没有意义。该表显示仓库中从未有 25 个产品，因此计算 25 个产品的价值没有意义。

表 4. 对 *PRODUCT_ID* 为 1234 的产品的 *Time* 维使用 *SUM* 函数来计算 *CURRENT_QUANTITY* 列的样本数据

	一月	二月	三月	第一季度
<i>SUM</i> (<i>CURRENT_QUANTITY</i>)	5	15	5	25

可以对 *Time* 维执行其它聚集函数（例如，*AVG*、*MIN* 和 *MAX*），而不是对所有维使用 *SUM* 函数。例如，在一月、二月和三月具有相同的一组样本数据的情况下，可以对 *Time* 维使用第二个聚集函数（如表 5 中所示）以便为该季度创建有意义的值。“当前价值”量度可表示该季度以来存储在仓库中的商品平均总价值或者该季度期间任何时间点的最高或最低价值。

表 5. 将 *AVG*、*MAX* 和 *MIN* 函数用于 *PRODUCT_ID* 为 1234 的 *Time* 维来计算 *CURRENT_QUANTITY* 列的样本数据

	一月	二月	三月	第一季度
<i>AVG</i> (<i>CURRENT_QUANTITY</i>)	5	15	5	8.3
<i>MAX</i> (<i>CURRENT_QUANTITY</i>)	5	15	5	15
<i>MIN</i> (<i>CURRENT_QUANTITY</i>)	5	15	5	5

创建量度的步骤

下列步骤说明如何使用“*OLAP* 中心事实属性”窗口来在现有事实对象中创建“流进”、“流出”和“当前价值”量度：

1. 要打开“事实属性”窗口，用鼠标右键单击 *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“流进”量度：
 - a. 在“量度”页上，单击**创建计算的量度**来创建“流进”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. 要在表达式中创建流，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** 列以将其添加至表达式。

- 单击 **OK** 以关闭 “SQL 表达式构建器” 窗口。不必更改 “聚集” 页上的缺省聚集函数 **SUM**。**SUM** 函数是 “流进” 量度的缺省函数，原因是数据源是数字的并且该量度涉及列（而不仅仅是现有量度）。
3. 创建 “流出” 量度:
 - a. 在 “量度” 页上，单击**创建计算的量度**以创建 “流出” 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 “流出” 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 **OK** 以关闭 “SQL 表达式构建器” 窗口。不必更改 “聚集” 页上的缺省聚集函数 **SUM**。**SUM** 函数是 “流出” 量度的缺省函数，原因是数据源是数字的并且该量度涉及列（而不仅仅是现有量度）。
 4. 创建 “当前价值” 量度:
 - a. 在 “量度” 页上，单击**创建计算的量度**以创建 “当前价值” 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 “当前价值” 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 5 页的图 1 显示可在 “SQL 表达式构建器” 窗口中创建的 “当前价值” 表达式。

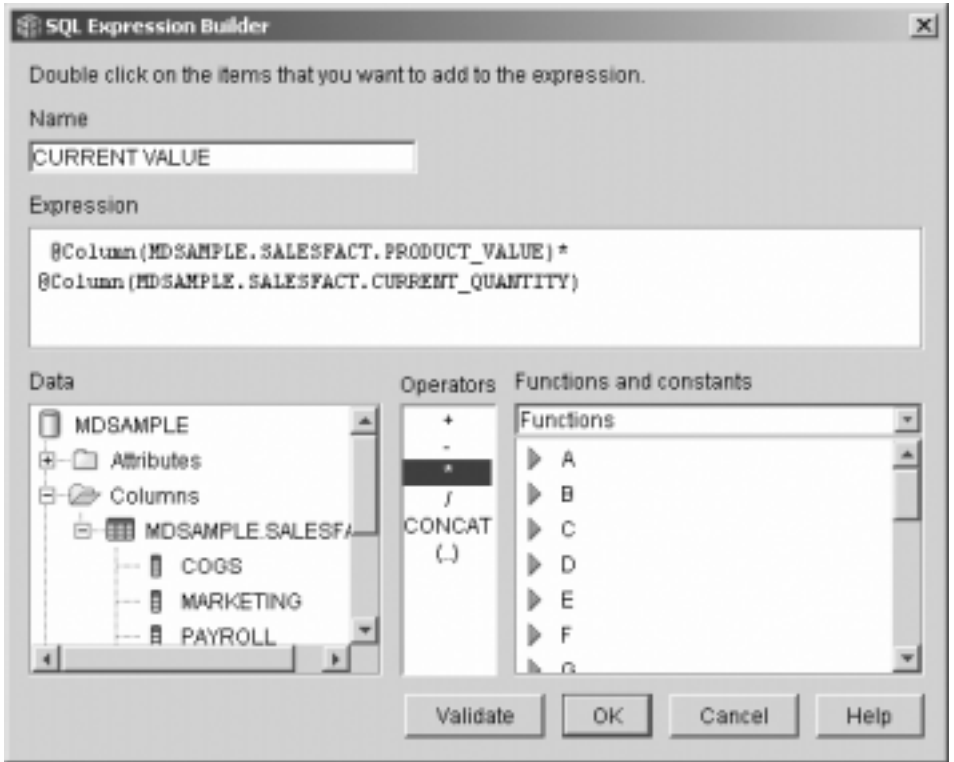



图 1. 在“SQL 表达式构建器”窗口中完成“当前价值”表达式

- d. 单击 **OK** 以关闭“SQL 表达式构建器”窗口。
- e. 在“聚集”页上，单击“当前价值”量度的聚集并从列表中单击**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. 必要的话，可通过选择 **Time** 并单击  按钮来向下移动 **Time** 维，以使它成为脚本中列示的最后一个维。然后，选择了 **Time** 维后，双击**列函数列表**中的 **AVG** 函数。聚集脚本（如第 6 页的图 2 中所示）对除 **Time** 维之外的所有维的数据求和（已对 **Time** 维的数据求平均值）。

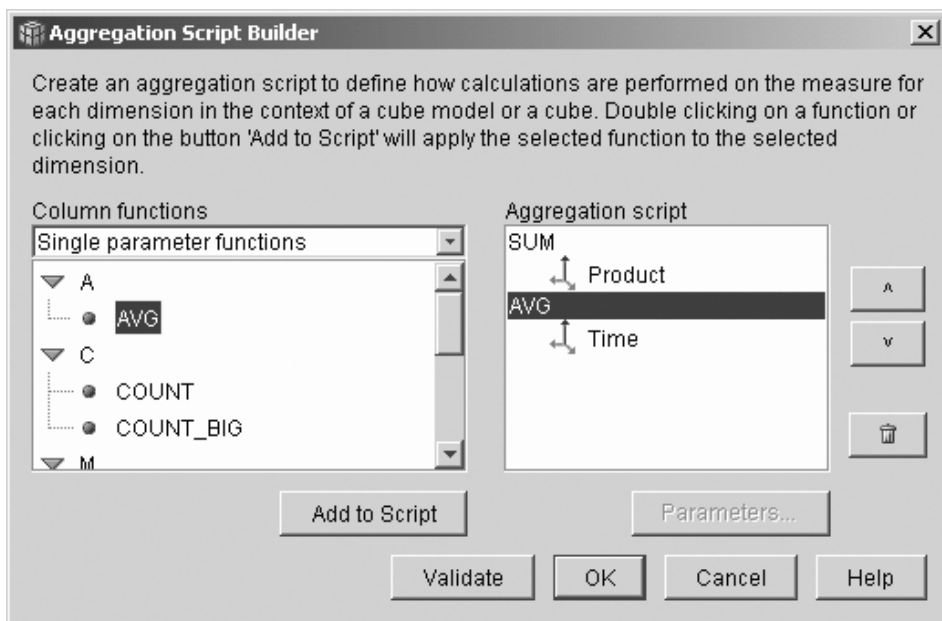


图 2. “当前价值” 量度的聚集脚本

- 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**OK**以保存聚集脚本并关闭窗口。
- 5. 单击**OK**以保存对事实对象的更改并关闭“事实属性”窗口。

现在有三个计算的量度用于仓库中的库存。可使用这些量度来分析产品流进和流出仓库的模式。

第 2 章 使广告成本与销售相关

汽车经销商正考虑增加其广告费用。为作出基于经验的决定，经销商首先想要分析广告费用与销售之间的历史关系。经销商感兴趣的是确定变动广告等级是否影响销售，特别是增加广告是否与销售增加紧密关联。

方案的详细信息

经销商的数据库具有包含“销售”和“广告成本”列的事实表。该数据库还具有其它几个维表。DBA 可创建使用 DB2 CORRELATION 函数来执行成本与销售之间的相关计算的量度。CORRELATION 函数是需要两个输入参数的多参数函数。在这种情况下，DBA 会将“销售”和“广告成本”列用作两个输入参数。

DBA 必须先 在聚集脚本中应用多参数聚集函数。可对所有维应用多参数函数或首先对除 Time 维之外的所有维应用多参数函数，并且可对 Time 维应用第二个函数（例如，MAX 函数）。DBA 会定义该量度的 SQL 表达式以便它直接映射至“广告成本”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参数。DBA 将第二个参数定义为直接映射至“销售”列的 SQL 表达式。CORRELATION 函数被定义为唯一聚集函数，以便量度可计算所有维的广告成本与销售成果之间的统计信息的相关。

创建量度的步骤

下列步骤说明如何使用“OLAP 中心事实属性”窗口来在现有事实对象中创建“广告 - 销售相关性”量度：

1. 通过用鼠标右键单击 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. 定义还将用作聚集脚本中的多参数 CORRELATION 函数的第一个参数的量度表达式。要定义表达式，展开**数据**列表中的**量度**文件夹并双击 **AD COSTS** 量度以将其添加至**表达式**列表。第 8 页的图 3 显示在“SQL 表达式构建器”窗口中创建的表达式。

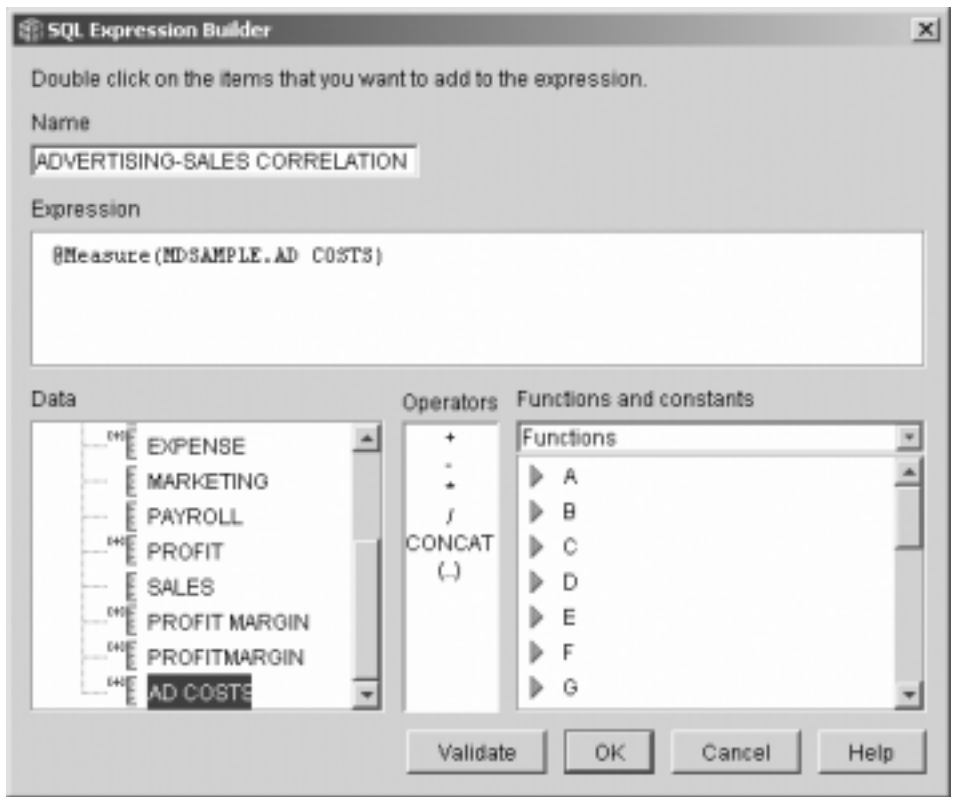


图 3. 在“SQL 表达式构建器”窗口中完成广告 - 销售相关性表达式

5. 在“聚集”页上，单击 **ADVERTISING-SALES CORRELATION** 量度的聚集函数并选择**聚集脚本**，如第 9 页的图 4 中所示。“聚集脚本构建器”窗口打开。

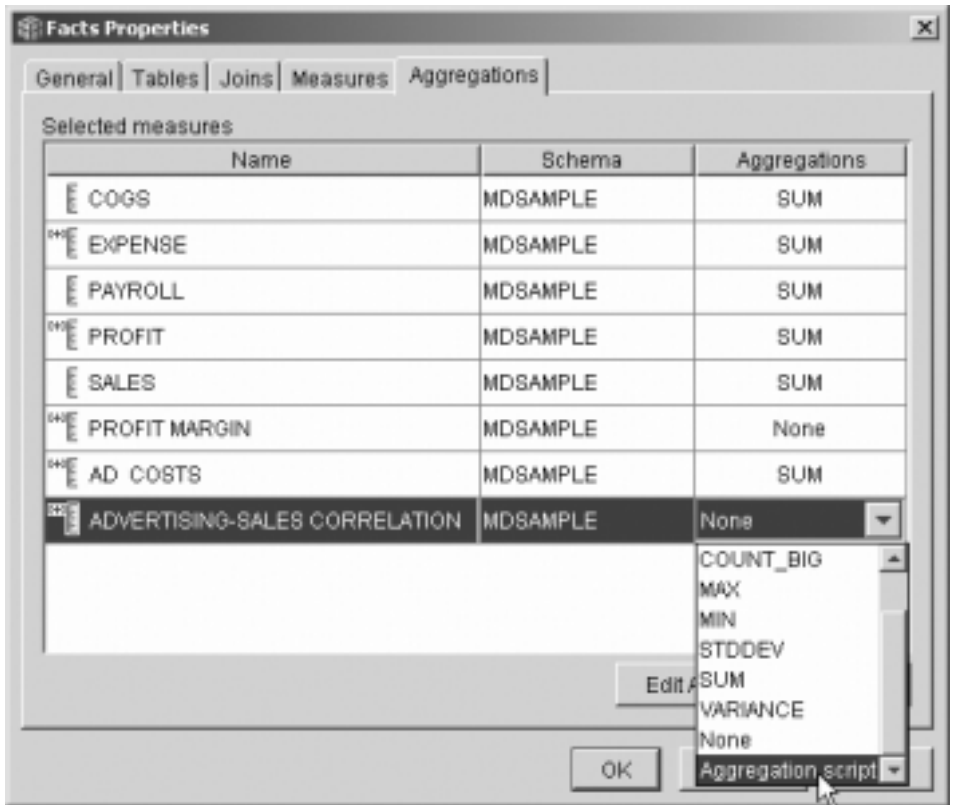


图4. “事实属性”窗口的“聚集”页

- 在列函数字段中，选择多参数函数。在多参数函数的列表中，选择 **CORRELATION** 函数并单击添加至脚本。第 10 页的图 5 显示 **CORRELATION** 函数在脚本中的维的列表的顶部。

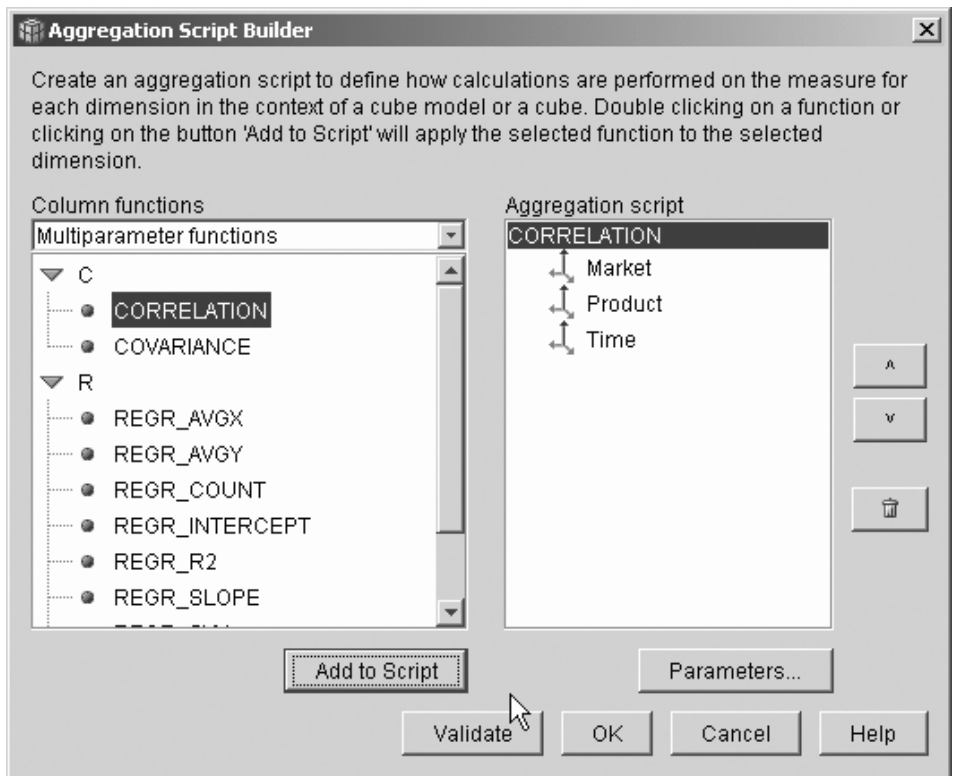


图 5. “广告 - 销售相关性” 量度的聚集脚本

- 单击**参数**按钮以指定 **CORRELATION** 函数的第二个参数。“函数参数”窗口打开。选择**使用现有量度**并选择 **SALES**。第 11 页的图 6 显示将“销售”量度指定为第二个参数的“函数参数”窗口。单击 **OK** 以保存选择并关闭“函数参数”窗口。

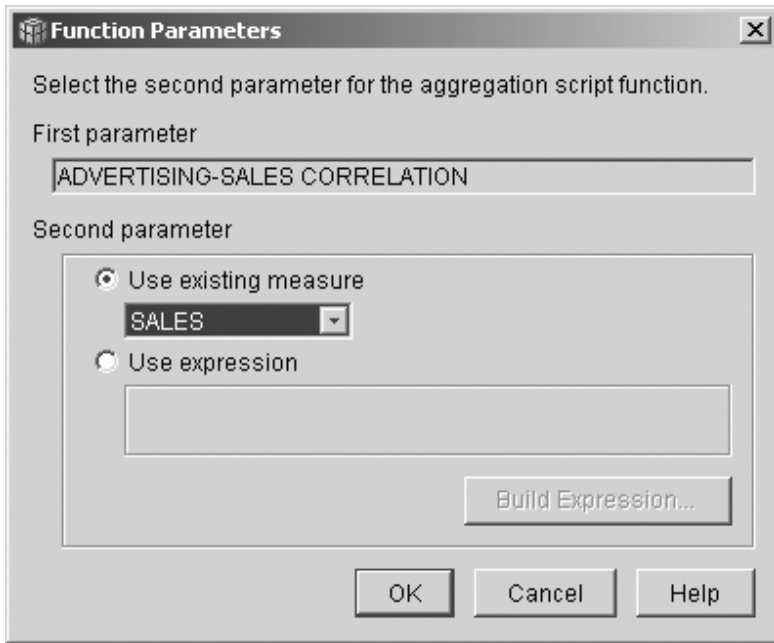


图 6. 在“函数参数”窗口中指定为第二个参数的“销售”量度

8. 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。单击**OK**以保存聚集脚本并关闭窗口。
9. 单击**OK**以保存对事实对象的更改并关闭“事实属性”窗口。

现在就有使数据库中的两种类型数据相关的量度了。可使用此量度根据销售成果的历史趋势来决定将来的广告费用。

第 3 章 计算商店的利润和利润率

玩具店的总理想要能够分析各种因素（例如，年份和产品类型）对利润和利润率的影响。玩具店的 DBA 必须先创建利润和利润率量度，才能完成较高级的分析。接着，DBA 可以创建附加量度，这些量度使不同因素与利润和利润率量度相关并对它们作出比较。

方案的详细信息

玩具店的数据库有一个事实表，该表具有“销售”、“商品销售成本”（COGS）和“开支”列以及几个维表中的每一个的相应外键列。DBA 已创建“销售”、COGS 和“开支”量度，它们分别映射至“销售”、COGS 和“开支”列。可以通过这些现有量度完整地创建“利润”和“利润率”量度。

为创建“利润”量度，DBA 创建了这样一个量度，它在 SQL 表达式中计算 $SALES-(COGS+EXPENSE)$ 并将对所有维进行计算得到的数据相加。可以通过引用现有量度和 / 或列来创建“利润”量度。

在创建“利润”量度之后，DBA 可创建“利润率”量度。“利润率”量度是两个现有量度的比率，表示为百分比 $(Profit / Sales)*100$ ，它不需要自身的聚集函数。不需要聚集函数是因为该量度只引用其数据已聚集的其它量度。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 不需要定义附加聚集。大多数聚集函数（例如，SUM）用比率表示是无意义的。例如，如果玩具店连续四个季度的利润率是 40%、32%、28% 和 37%，则这段时间的比率相加将得到年利润率为 137%，这没有任何意义。

创建量度的步骤

下列步骤说明如何使用“OLAP 中心事实属性”窗口来在现有事实对象中创建“利润”和“利润率”量度：

1. 要打开“事实属性”窗口，用鼠标右键单击 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“利润”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“利润”表达式，展开**数据**列表中的**量度**文件夹并完成下列步骤：

- 双击数据列表中的 **SALES** 量度以将其添加至表达式。
- 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，突出显示表达式的 `@Measure(MDSAMPLE.COGS)+@Measure(MDSAMPLE.EXPENSE)` 并从运算符列表双击 **(..)** 运算符以将表达式的所选部分包括在圆括号中。

图 7 显示可在“SQL 表达式构建器”窗口中创建的利润表达式。

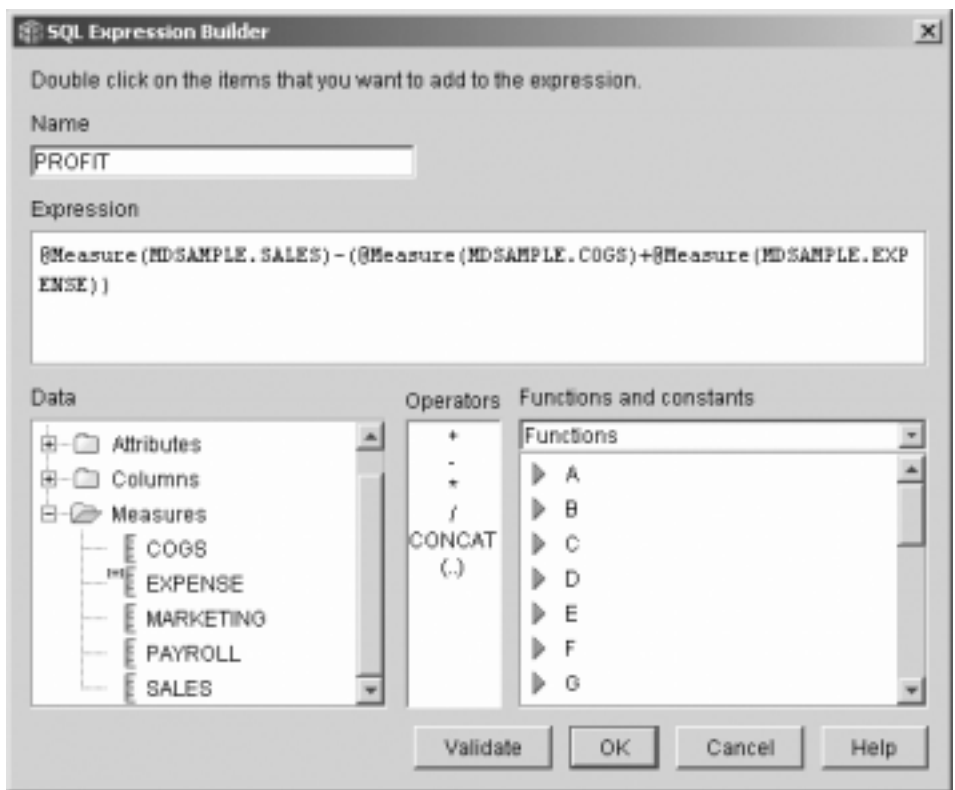


图 7. 在“SQL 表达式构建器”窗口中完成利润表达式

- d. 单击 **OK** 以创建“利润”量度并关闭“SQL 表达式构建器”窗口。
 - e. 在“聚集”页上，单击“利润”量度的聚集并选择 **SUM** 函数。“利润”量度完成。
3. 创建“利润率”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“利润率”表达式，展开**数据**列表中的**量度**文件夹并完成下列步骤：
 - 双击**数据**列表中的 **PROFIT** 量度以将其添加至表达式。
 - 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包括在圆括号中。
 - 将光标放在表达式的结尾并双击**运算符**列表中的 * 运算符。
 - 在**表达式**字段中的表达式的结尾输入 100。

第 16 页的图 8 显示可在“SQL 表达式构建器”窗口中创建的利润率表达式。

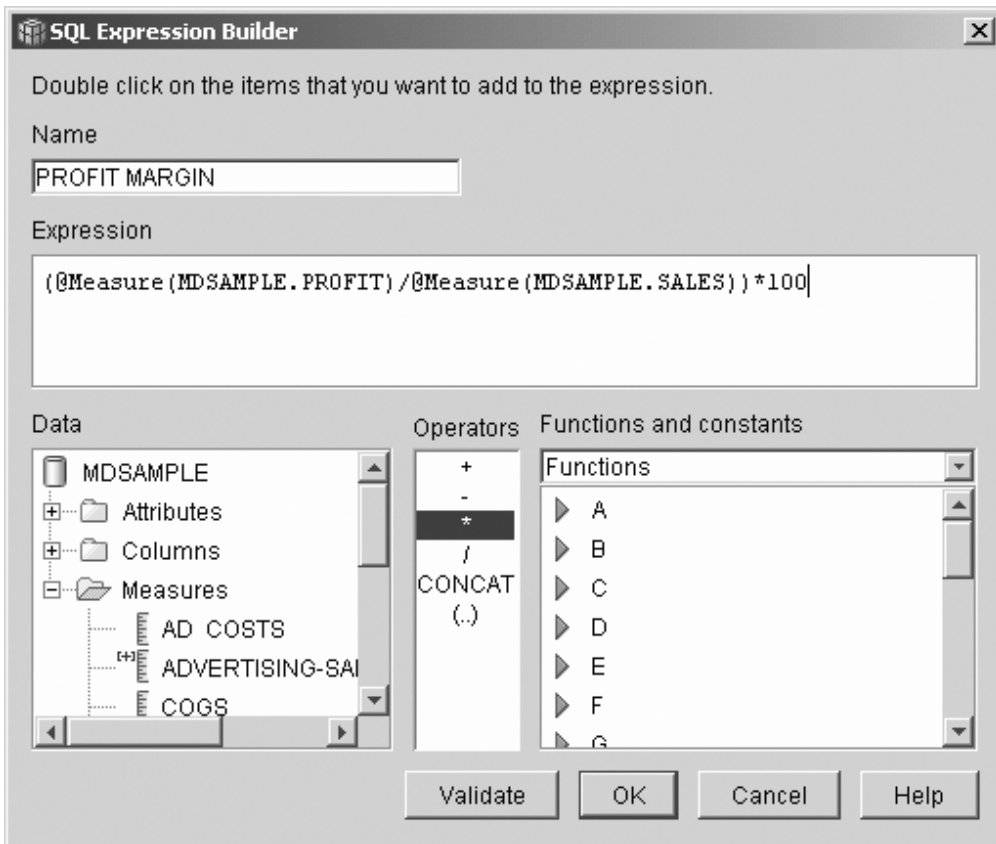


图 8. 在“SQL 表达式构建器”窗口中完成利润率表达式

d. 单击 **OK** 以创建“利润率”量度并关闭“SQL 表达式构建器”窗口。

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

4. 单击 **OK** 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 创建这两个量度之后，可根据这两个重要量度完成附加分析。

第 4 章 计算因特网订单数

零售公司几年前通过增添因特网销售扩展了业务。现在，公司想要分析因特网销售的效果。公司首先需要计算的对象之一是通过因特网完成的订单数。

方案的详细信息

公司的数据库具有因特网订单的事实表，该表具有 `ORDER_ID`、`PRODUCT_ID`、`QUANTITY` 和 `TIME_ID` 列。`PRODUCT_ID` 列包括相应订单销售的每个产品，而 `QUANTITY` 列存储订单购买的产品数量。具有多个产品的订单的行条目的数目与订单销售的唯一产品的数目相同。例如，表 6 显示三个订单，其中订单 1 包括三个产品 A、一个产品 O 和一个产品 G。

表 6. 部分事实表内容

<code>ORDER_ID</code>	<code>PRODUCT_ID</code>	<code>QUANTITY</code>
1	A	3
1	O	1
1	G	1
2	L	1
2	Q	2
3	P	5

DBA 可创建“订单计数”量度，它计算 `ORDER_ID` 列中每个唯一条目的数目。

“订单计数”量度是在 SQL 表达式中使用 `DISTINCT` 关键字和所有维的聚集的 `COUNT` 函数定义的。量度的 SQL 表达式将创建不同订单的列表，在聚集期间会对它们计数。因为该量度不涉及任何求和，所以它称为非可加量度。

当具有想要对其计数的字符数据或其它数据时，非可加量度也很有用。例如，您可能使用非可加量度来对将产品装运至的地区的邮政编码数计数。

在此示例中，DBA 决定定义直接映射至 `ORDER_ID` 列的“订单标识”量度。但是，可以选择以相同方式使用 `ORDER_ID` 列。根据是在 SQL 表达式中使用列还是使用量度，缺省聚集是不同的，但是在任一情况下，您都需要将缺省聚集更改为 `COUNT` 函数，如第 18 页的『创建量度的步骤』中所述。

创建量度的步骤

下列步骤说明如何使用“OLAP 中心事实属性”窗口来在现有事实对象中创建“订单计数”量度:

1. 要打开“事实属性”窗口，用鼠标右键单击 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. 要创建订单计数表达式，展开**数据**列表中的**量度**文件夹并完成下列步骤:
 - 在**函数和常量**字段中，选择**其它**。在其它函数和常量的列表中，双击 **DISTINCT** 关键字。
 - 双击**数据**列表中的 **ORDER ID** 量度。

第 19 页的图 9 显示可在“SQL 表达式构建器”窗口中创建的订单计数表达式。

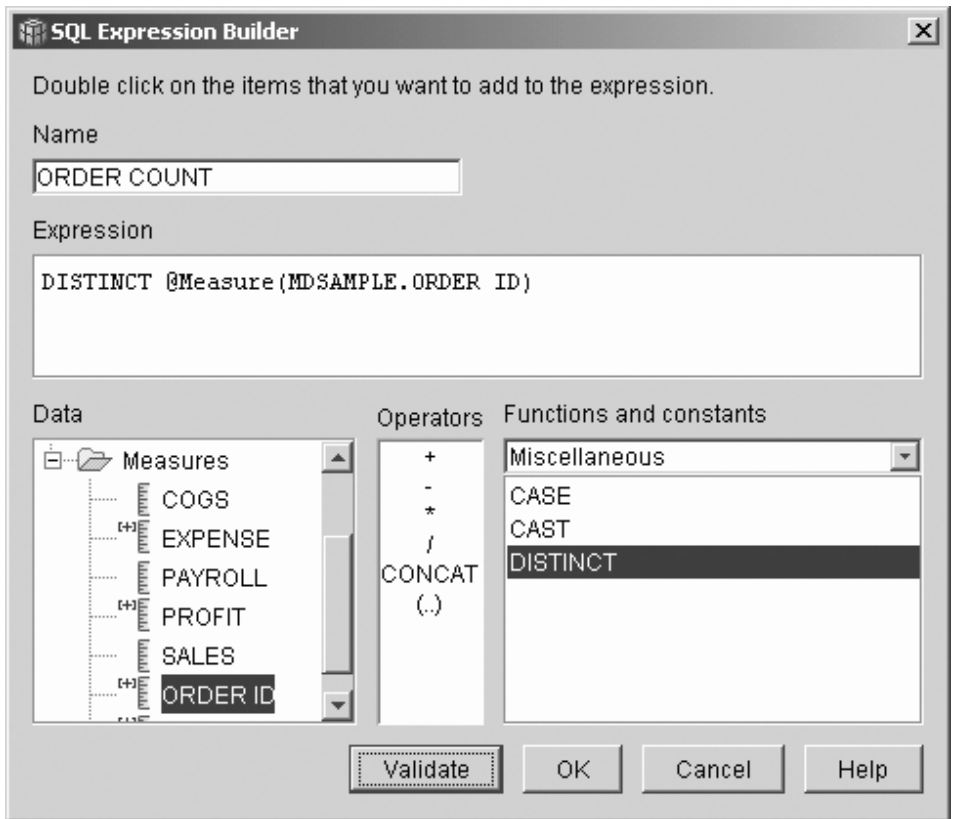


图 9. 在“SQL 表达式构建器”窗口中完成订单计数表达式

5. 单击 **OK** 以关闭“SQL 表达式构建器”窗口。
6. 在“聚集”页上，单击“订单计数”量度的聚集并选择 **COUNT** 函数。
7. 单击 **OK** 以保存对事实对象的更改并关闭“事实属性”窗口。

现在就有对不同“订单标识”行值的数目计数的量度了。可将此量度与其它量度配合使用来进一步分析数据。

第 5 章 对销售数字分等级

过去几年以来，某办公用品连锁店扩展迅速。业务经理考虑关闭其中一些绩效最低的商店以节省成本和增加利润。商店的销售历史是决定关闭商店的一个重要因素。分析人员需要能够对销售数字分等级并在各个维之间下寻以完成他们的分析。

方案的详细信息

办公用品商店的数据库具有包含“销售”列以及其它列的事实表。该数据库还具有几个维表。DBA 可创建使用 RANK 函数的“销售等级”量度，该函数是由 DB2 通用数据库 (DB2 UDB) 提供的 OLAP 函数。

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

RANK

对行排序并对每行指定等级。等级定义为就排序而言唯一的之前行的数目加 1。如果两行或多行的相对顺序因为具有重复行值而不能确定，则会指定相同等级编号。如果有重复的行值，则分等级结果可能造成编号中有间隔。第 22 页的表 7 显示哪些分等级结果来自一组样本行值的 RANK 函数的示例。

RANK 函数的典型语法为:

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

其中 *sort-key-expression* 是要分等级的数据集，而 *expression-order* 是关键字 (**ASC** 或 **DESC**)，它按升序或降序次序对 *sort-key-expression* 的值排序。DB2 Cube Views 要求 *sort-key-expression* 充当现有量度而不是列或属性。而且，DB2 Cube Views 不支持由 DB2 UDB 随此函数提供的 PARTITION BY 子句。有关 RANK 函数的更多信息，可在“DB2 UDB 信息中心”中找到。

DENSERANK

对行排序并对每行指定等级。行的等级定义为确实在该行之前的行数加 1。因此，分等级结果将是连续的，在等级编号中没有间隔。第 22 页的表 7 显示哪些分等级结果来自一组样本行值的 DENSERANK 函数的示例。

DENSERANK 函数的典型语法为:

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

其中 *sort-key-expression* 是要分等级的数据集，而 *expression-order* 是关键字 (**ASC** 或 **DESC**)，它按升序或降序次序对 *sort-key-expression* 的值排序。DB2 Cube Views 要求 *sort-key-expression* 充当现有量度而不是列或属性。而且，DB2 Cube Views 不支持由 DB2 UDB 随此函数提供的 **PARTITION BY** 子句。有关 **DENSERANK** 函数的更多信息，可以在“DB2 UDB 信息中心”中找到。

ROWNUMBER

根据排序计算该行的顺序行号，第一行从 1 开始。如果未指定 **ORDER BY** 子句，则以任何顺序将行号指定给各行。

ROWNUMBER 函数的典型语法为：

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

其中 *sort-key-expression* 是要分等级的数据集，而 *expression-order* 是关键字 (**ASC** 或 **DESC**)，它按升序或降序次序对 *sort-key-expression* 的值排序。DB2 Cube Views 需要现有量度（而不是列或属性）用作此函数的数据源。而且，DB2 Cube Views 不支持由 DB2 UDB 随此函数提供的 **PARTITION BY** 子句。有关 **ROWNUMBER** 函数的更多信息，可以在“DB2 UDB 信息中心”中找到。

这些 OLAP 函数未列示在“SQL 表达式构建器”函数和常量列表中。

表 7. 使用 **RANK** 和 **DENSERANK** 函数对一组样本值分等级的结果

行值	排序	来自 RANK 函数的分等级结果	来自 DENSERANK 函数的分等级结果
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

下列步骤说明如何使用“OLAP 中心事实属性”窗口来在现有事实对象中创建“销售等级”量度：

1. 要打开“事实属性”窗口，用鼠标右键单击 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“销售等级”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“销售等级”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**表达式**字段中的函数语法中的单词 **measure** 并双击 **SALES** 量度以将 SALES 量度添加至表达式。

最后的表达式显示在图 10 中。

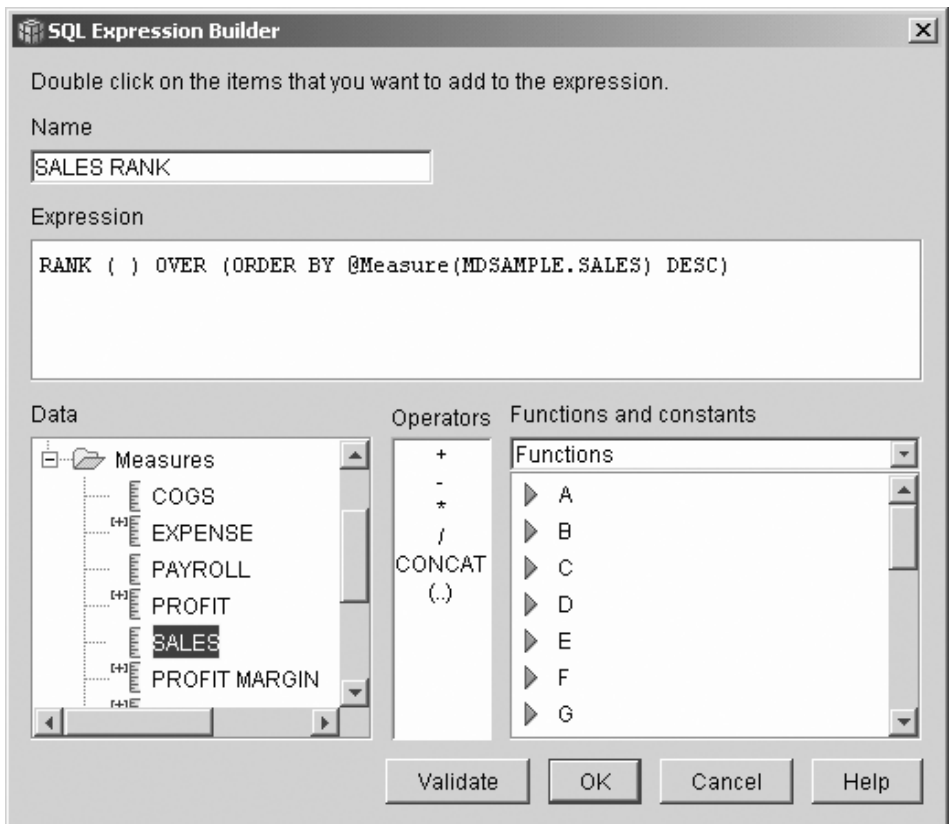


图 10. 在“SQL 表达式构建器”窗口中完成“销售等级”表达式

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

不必更改“聚集”页上的缺省聚集“无”。“无”选项是“销售等级”量度的缺省聚集，原因是数据源是数字并且只引用现有量度。

在量度的表达式中使用 RANK 函数来以降序次序对“销售”列排序，分析人员可下寻其它维以确定去年以来销售成绩最差的商店（对于特定产品线或根据存储在数据库中的任何其它维数据）。

第 6 章 使用事实表中存储的时间数据来创建 Time 维

零售企业 XYZ Retail 正在 DB2 Cube Views 中为他们的销售交易数据建模以便更有效地分析他们的数据。但是，由于数据的交易特性，所以可用的唯一时间信息是与每次交易相关联的日期。在 Time 维中建模的时间信息是将上下文添加至许多常用计算（例如，按季度分析销售趋势和计算每周的平均库存值）所需的。

许多 DBA 避免将时间数据存储为交易的日期或时间戳记，这是因为，如果某一天没有任何交易，则数据上会出现空白区域，这在精确聚集和显示数据时会造成问题。通常，对时间表中的时间数据建模是比较好的选择。但是，XYZ Retail 的 DBA 确信每天将至少有一笔交易并决定保留数据的当前结构。

方案的详细信息

XYZ Retail 有一个事实表，包括有关每笔交易的可量度数据，这些数据包括“销售”、“成本”、“销售数量”和“日期”。另外，数据库包含“区域”维表和“产品”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来创建维对象。

根据事实表中的单列日期数据创建 Time 维有两个唯一需求：

- 因为有效立方体模型中的所有维对象必须连接至事实对象，而且 Time 维对象和事实对象是基于同一事实表的，所以 Time 维对象必须通过使用自连接来将事实表连接至它本身以连接至事实对象。
- DBA 必须构建将日期数据聚集成有意义的级别（例如，星期、月份、季度和年份）的计算属性。

自连接是将表连接至其本身的连接类型，在这种情况下，该表是事实表。自连接应该连接这样的一列或多列，它们在一起可唯一地标识事实表中的任何行。主键是最好的选择。但是，如果未定义主键，则适合的候选主键是用来将事实表与维表连接的一组列。要优化立方体模型，必须定义主键。连接基数必须是 1:1 且连接类型必须是内部连接。

第 26 页的图 11 显示事实对象、基于事实表的维和事实至维的连接如何映射至同一事实表。

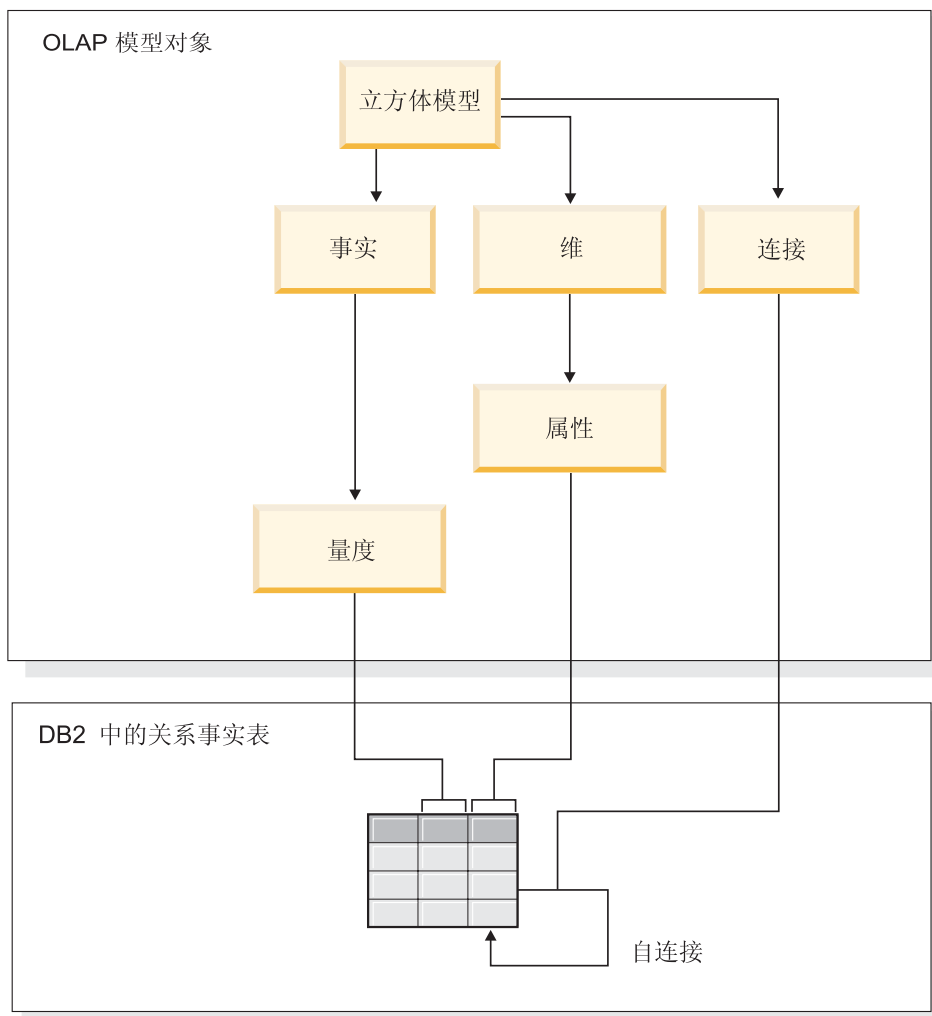


图 11. 自连接如何将表连接至其本身

创建属性和维的步骤

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

1. 要打开“维”向导，用鼠标右键单击 OLAP 中心对象树中的立方体模型并单击**创建维**。“维”向导打开。

2. 在“名称”页上，在**名称**字段中输入 Time。可选择更改业务名并输入注释。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.
5. 在“维属性”页上，选择**时间戳记列**。
6. 可选：创建将时间戳记数据聚集成较大块（例如，月份、季度和年份）的附加的计算属性。要创建计算属性，单击**创建计算属性**按钮来打开“SQL 表达式构建器”并为每个属性定义表达式以将源时间戳记列计算为月份、季度和年份。在定义每个计算属性之后，单击**验证**按钮以确保表达式有效，然后单击 **OK** 以关闭“SQL 表达式构建器”并返回至“维”向导。在选择并创建想要的所有属性之后，单击**下一步**。
7. 在“维类型”页上，选择 **Time**。单击**下一步**。
8. 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。输入名称并单击**下一步**。同时为左边和右边的属性选择唯一定义事实表中的任何行的列或一组列（例如，主键）。一次选择一对并单击**添加**以将属性对添加至连接。选择内部连接类型和 1:1 基数。在添加必需的属性对之后，单击**完成**。“连接”向导关闭。
9. 在“事实维连接”页上，单击**完成**。

在立方体模型中定义了 Time 维之后，XYZ Retail 可将新级别的含义添加至其数据分析。他们现在可以执行与时间有关的分析（包括库存）。

声明

本信息是为在美国提供的产品和服务编写的。

IBM 可能在其它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证。您可以用书面方式将许可证查询寄往：

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

有关双字节（DBCS）信息的许可证查询，请与您所在国家或地区的 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：国际商业机器公司以“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；这些更改将编入本出版物的新版本中。IBM 可以随时对本出版物中描述的产品和 / 或程序进行改进和 / 或更改，而不另行通知。

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它程序（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。

本信息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 国际程序许可证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、其出版说明或其它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、兼容性或任何其它关于非 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。

此信息包含了日常商业操作中使用的数据和报表的示例。为了尽可能完整地对它们进行说明，示例中包含了个人、公司、品牌和产品的名称。所有这些名称纯属虚构，如与实际商业企业使用的名称及地址雷同，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，在其它操作环境中获取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，因此不保证与在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、其出版说明或其它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、兼容性或任何其它关于非 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。

如果您正在查看本信息的软拷贝，则可能不会出现照片和彩色插图。

商标

以下各项是国际商业机器公司在美国和 / 或其它国家或地区的商标：

DB2

DB2 Universal Database
IBM
Office Connect
Redbooks

以下各项是其它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 / 或其它国家或地区的商标。

UNIX 是在美国和 / 或其它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. 在美国和其它国家或地区的商标或注册商标。

其它公司、产品或服务名称是其它公司的商标或服务标记。