

Oracle Warehouse Builder 10g

在您转移数据的任何时候
帮助您节省时间和金钱

Oracle 白皮书
2004 年 2 月

ORACLE®

Oracle Warehouse Builder 10 g

在您转移数据的任何时候 帮助您节省时间和金钱

引言

在 IT 中，您面对的常见任务之一是加载数据。大多数机构运行着许多将数据从一个系统转移到另一个系统或从文件转移到数据库中（或相反）的作业。在这些情况下，大多数时候都要对特别编写 PL/SQL 程序以处理所有这些作业。但这是否是执行数据转移的最有效率的方式还是个问题。

为什么机构需要到处转移数据存在许多原因。让我们看一个比较常见的例子，以便了解在这种操作中涉及到的一些挑战。我们的例子是假设您是一个机构中销售 widget 的 IT 经理，您管理的是从 OLTP 订单系统中请求每月销售收入和预测的各种报表。

要回答这个问题，您一般将数据从 OLTP 系统转移到一个单独的系统中，从而允许您优化查询性能并结合高级的预测功能。

您转移数据所需的是一些从各种 OLTP 表中提取数据并将其结合以获得期望的结果的一些代码。您将必须集合您的订单，生成每月的报表，然后存储数据。要得到这些代码，您有一些选择：您可以手动编写用于提取数据的代码，或者可以使用如 Oracle Warehouse Builder 之类的工具来生成这种代码。

本文将详细讨论为什么 Oracle Warehouse Builder 是转移数据的最佳选择。正如您将看到的，底线是使用 Oracle Warehouse Builder 将为您节省时间和金钱。

定义 WAREHOUSE BUILDER 的范围

虽然 Oracle Warehouse Builder 最常被描述为一种 ETL¹ 或数据集成工具，但它所做的事情实际上远不止转移数据。本质上它是为数据仓库提供的一种设计工具，它允许您设计保存数据的（关系和多维）目标模式、使数据转移到目标模式中的 ETL 过程和分析数据的最终用户应用程序基础，从而为数据仓库基础架构提供一站式销售。

Oracle Warehouse Builder 在一个收集您的设计的元数据信息库之上构建。一旦您在该信息库中创建了您的设计，您就可以生成代码并将其部署到数据库中。模式和对象定义使用 SQL 创建。该工具为数据集成例程创建了一个 SQL 和 PL/SQL 的组合，从而允许您在要转移的数据上执行复杂的转换。

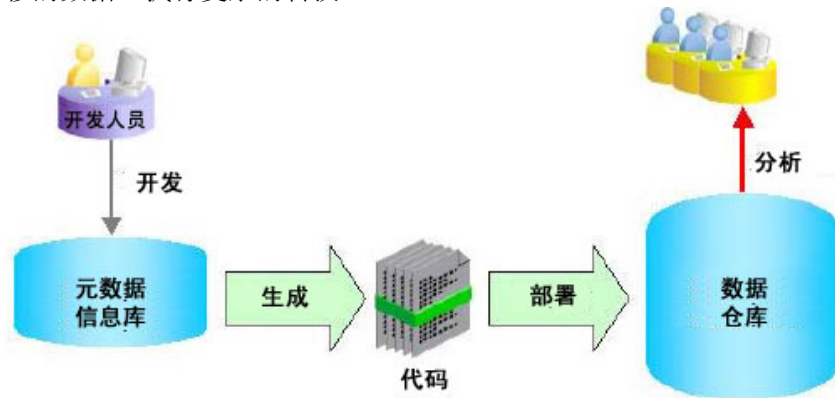


图 1 从元数据信息库中生成代码

很可能您没有在 Oracle Warehouse Builder 中为您原来的源系统和目标系统对您的模式建模，而在如 Oracle Designer 或 Computer Associates ERWin 之类的工具中进行了建模。在这种情况下，您可以在这些工具和 Oracle Warehouse Builder 之间交换元数据，从而允许您将两种工具都用于您偏好的活动。

可以从各种数据源中执行提取。当然，Oracle 数据库可以作为数据源，而其它常见的数据库（如 IBM DB2、Microsoft SQL Server 等）同样也可以作为数据源。纯文本文件属于 Oracle Warehouse Builder 的源系统和目标系统，您可以从纯文本文件中进行提取也可以写纯文本文件。

一旦您设计了您的数据仓库或模式并加载了数据，Oracle Warehouse Builder 允许您设计并将额外的元数据转移到市场上常见的查询工具（如 Oracle Discoverer 或 Business Objects）中。您还可以创建专门的 OLAP 对象，以适合高级分析（如我们在示例中讨论的预测）。

¹ ETL 代表提取、转换和加载。这依次是将数据从一个系统加载到另一个系统（一般是数据仓库）中去的行业术语。

在利用元数据方面，大多数工具止步于此 — 但 Oracle Warehouse Builder 不是这样。Oracle Warehouse Builder 最强的一点是它通过容易地允许添加和修改元数据来继续支持您的设计周期。Oracle Warehouse Builder 允许您预先了解修改对系统会有怎样的影响，然后帮助您在 ETL 和模式设计中实施修改。换句话说，Oracle Warehouse Builder 在您的系统的整个生命周期为您提供辅助，而不只是辅助您初始的设计。

了解 ORACLE WAREHOUSE BUILDER 如何为您节省时间和金钱

到现在为止，您可能想知道一个元数据信息库、一个工具和一些生成的代码如何能够为您节省时间或金钱，因此让我们从两个方面来说明这一点：技术方面 — 这个工具能否解决我的问题，以及是否可能以更少的时间和精力来解决我的问题；业务方面 — 这个工具能否解决我的问题并为我省钱。

利用 Oracle Warehouse Builder 节省金钱

要为您的项目节省金钱，必须满足一些基本的业务规则：

- 使您的资本投资最大化 — 这个工具能否使我们现有的技术的利用率最大化而不是增加更多的技术？
- 高投资回报 — 我能够多快地从我的投资中获取利益？
- 低知识障碍 — 机构是否拥有所需的知识或者能否快速地获取它？

使您的资本投资最大化

由于最近 IT 支出的减少，增加硬件在任何机构中都不一定是件容易的事情。因此，目标是增加在您拥有的框架上的工作负载。我们需要着眼的这种范例有两个主要的方面：

- 更高的硬件利用率
- 更高的新软件特性的使用率

Oracle Warehouse Builder 有一些独特的特性正好能够帮助您为您的机构实现这些目标。

与基于 ETL 工具的引擎相反 — 它们提取数据然后在把数据加载到数据库中之前在它们自己的引擎中执行转换，Oracle Warehouse Builder 提取数据，然后直接在最终存储数据的数据库内部执行任意的转换。

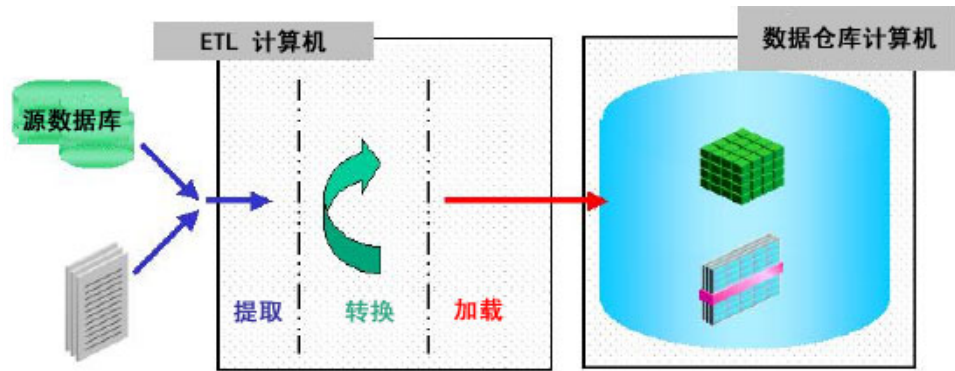


图 2 需要两台同样大小的计算机来完成工作

这意味着使用基于引擎的技术，您要使用两台计算机（参见图 2）来执行相同的工作负载，而 Oracle Warehouse Builder 在一台计算机上执行工作负载。因此当数据仓库计算机在晚上空闲时，ETL 计算机正赶着执行您的作业。当您的用户在查看他们每月的销售情况时，数据仓库计算机在工作而 ETL 计算机什么也不做。这不是有效使用资源的方式。

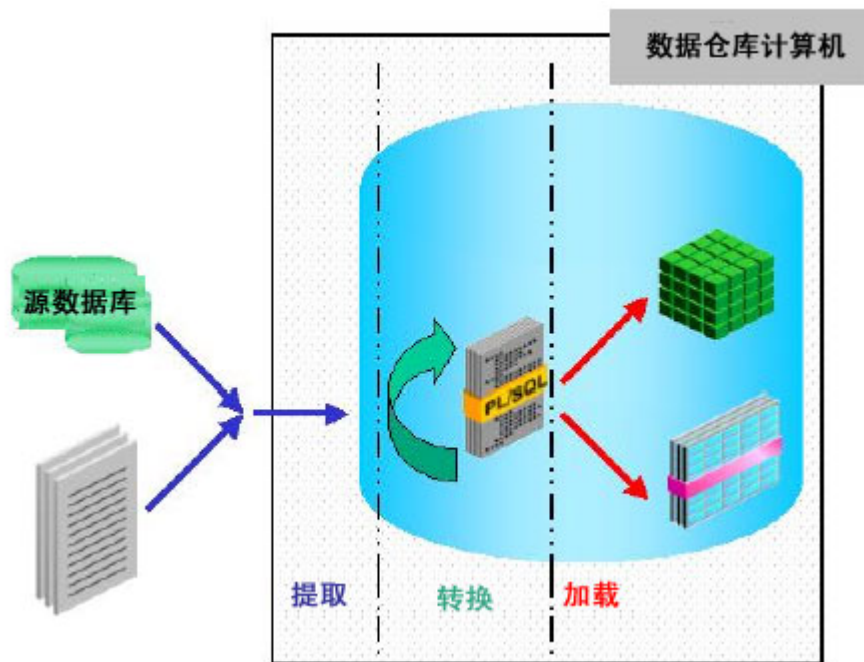


图 3 Warehouse Builder 使用一台计算机完成所有的工作²

正如您在图 3 中所看到的，Oracle Warehouse Builder 执行的所有工作都发生在数据仓库计算机上。因此在晚上运行 ETL 作业、在白天执行查询使得计算机的利用率最大化。由 Oracle Warehouse Builder 的更简单的体系结构提供的一个额外的好处是管理变得更简单了。现在您的 Oracle DBA 能够管理正在执行的所有工作，而不需要一个 DBA 和一个 ETL 管理员。

² 图 2 和 3 仅为示意图。您在两个场景中将需要相同大小的计算机。在基于引擎的场景中，您将需要两台这种计算机。

Oracle Warehouse Builder 为您提供的另一个好处是与 Oracle 数据库中的功能的紧密对齐。由于 Oracle 数据库的不同版本中的增强和新的特性，相同的逻辑数据转移可以利用新的功能通过不同的方式来实施。

例如，将 ‘INSERT’ 语句和一条 ‘UPDATE’ 语句修改为一条组合的 ‘MERGE’ 语句，使您的代码能够更好地执行，而无需增加任何额外的硬件。如果您自己编写代码，那么您将必须投入大量的时间来修改所有的例程并验证它们是正确的。这是代价很大的操作。

Oracle Warehouse Builder 不仅将这些新的语句与新的代码一起使用。该工具还允许您通过“转一下开关”来升级您现有的全部设计。在这个特定的例子中，您将简单地把生成的代码的版本改为 9i，然后立即在所有适用的地方生成新的语句。当然，您也可以通过使用付费获得的功能（而不是为单独的 ETL 引擎支付额外的费用）来更好地利用数据库。

投资回报 (ROI)

这是帮助决定实际投资于一种工具是否值得并估计投资回报将怎么样的一种经典的方法。典型的投资回报 (ROI) 计算用在更实际的资产而不是软件上，因此我们并不打算试图进行科学的 ROI 计算。不过，我们可以着眼于典型的 ROI 计算中的一些要点，并将它们应用到软件上。

这些要点之一是成本，对于 Oracle Warehouse Builder，这是一个主要的竞争优势。它作为 Oracle 的开发人员套件的一部分提供，整个套件的每开发人员价格很低（更不用说许多 Oracle 客户已经拥有了它），据一些行业分析家称，Oracle Warehouse Builder 的成本比同类最佳工具低 50 到 100 倍。因此，即使所有其它的因素和功能都相同，单单因为这个原因，Oracle Warehouse Builder 就将拥有比同类产品更高得多的投资回报。

- 现在，让我们在计算中加入我们之前讨论的一些好处：
- 由于众所周知的单一的技术环境，实现了更低的管理成本
- Oracle 环境中的更低的培训成本（关于更多详细信息，请参见下一段落）
- 更高的现有软件和硬件资源的利用率

如果我们把这些因素加到许可成本中，那么与其它 ETL 工具相比 Oracle Warehouse Builder 的 ROI 就更高了。

3 在 Oracle9i Release 1 中引入

虽然许多客户实际上购买了如 Oracle Warehouse Builder 之类的工具，但仓库的最常见的实施仍然是定制编写的代码。在 Oracle 实施中，提供模式和 ETL 例程的一般是 SQL 和 PL/SQL。因此，让我们试着列出一些活动来看看 Oracle Warehouse Builder 将多大程度地加速开发和修改，以及这将如何为相关的总体 ROI 计算作出贡献。

初始开发活动：

任务	使用 Warehouse Builder 超过定制编码的优势
初始的 ETL 设计/编码	通过逻辑建模而不是编写成百上千的代码行获得的时间增益
构建错误处理	Warehouse Builder 免费自带
测试	所需时间几乎一样，在 Warehouse Builder 中更易于重现
调试	时间和工作显著减少
应用调试修改	由于在各个地方用逻辑修改代替了独立的代码修改，速度更快了
最终测试	所需时间几乎一样，在 Warehouse Builder 中更易于重现
创建最终用户层	通过使用 Warehouse Builder 实现自动化，大大减少了时间
投入生产	使用 Warehouse Builder 更简单更易于管理，减少了错误和重复
构建管理基础架构	自带了工具且使之启动并运行几乎不需要工作量
文档	大多数 ETL 代码没有提供文档说明，而使用 Warehouse Builder 将直接为逻辑提供文档说明，因为您将在根据设计说明了大多数逻辑的图形化图表中工作
总体	减少了工作量并减少了开发时间

对设计或代码进行修改:

任务	使用 Warehouse Builder 超过定制编码的优势
识别修改	Warehouse Builder 为您完成这一工作并报告修改
确定修改的影响	预先显示影响并提供了文档说明, 从而在您开始修改代码之前为您提供了任何修改的一个清楚的显示
设计修改	逻辑设计环境为您节省了重新设计的时间, 因为显示了修改的影响并且可以查明设计修改
实施修改	Warehouse Builder 显著减少了实施, 因为您只需要修改您的设计, 而不用检查您所有的代码行
测试	存在更少的修改和更小的影响, 因此涉及到的测试也更少。Warehouse Builder 在这里为您节省了时间
调试	使用调试器查找错误更快了
应用调试修改	修改被应用到模型中而不是成百上千的代码行中
最终测试	仍然需要, 但因为修改更少了而且影响更小了, 所以所有这些都更快和更高效
更新最终用户层	Warehouse Builder 从相同的设计中完成这一操作, 因此几乎不需要额外的工作
投入生产	Warehouse Builder 生成更新脚本以使影响最小化并减少了所需的时间量
文档	在执行修改时创建文档, 从而减少所需的时间
总体	Warehouse Builder 允许您在更新和修改您的应用程序时显著地减少时间

即使我们假定您的开发人员需要进行额外的培训来掌握 Oracle Warehouse Builder, 很显然在相关的 ROI 计算中

Oracle Warehouse Builder 仍将独占鳌头。实际上每一个项目经理都会问一个问题: Oracle Warehouse Builder 是否能解决项目需求。如果回答是肯定的, 那么反对使用这一工具的意见将非常少, 大多数反对意见无疑来自于业务方面。

保留定制编码解决方案的一个主要的因素是培训。如果需要花几个星

期来掌握一种工具，是否值得花这个时间？当然还有我们要花多少时间和成本。因此让我们在下一段落着眼于此进行讨论。

知识

正如我们都知道的，培训要花费金钱，在当前的经济环境中，培训在每一个机构中都是要经过详细审查的事情。我们已经讨论了您在 Oracle Warehouse Builder 中开发的元数据和您从其中获得的一些好处。这种元数据的一个额外的好处是您不实际创建代码，而让 Oracle Warehouse Builder 来创建它。现在，您是否想知道这对于您的培训费用将意味着什么？回答是您可以将相同的设计应用到许多数据库版本上，而且因为您不编写代码，所以您从第一天开始就不需要了解数据库中的新特性的所有细微差别。

换句话说，当您想了解您的数据库时，并不需要查看每一种新特性类就可以从数据库升级中获益。您的开发人员不需要首先在 PL/SQL 类上花费几天的时间，他们可以简单地调节 Oracle Warehouse Builder 中的设置，并立即使用最新的特性。这不仅直接减少了您的开发人员的培训需求，还提高了您能够从数据库的更新版本中获益的程度。

利用 Oracle Warehouse Builder 节省时间

现在您了解了为什么这种工具能够为您节省时间和金钱的业务原因，接着我们需要确保该工具能够实际满足您的技术需求。它能否真正地把我的数据从 OLTP 系统转移到您的数据仓库中？为了使 Oracle Warehouse Builder 实际满足您的需求，该工具必须满足一些基本的高水平的技术需求：

- 可伸缩性和性能 — 该工具和创建的应用程序都能够伸缩并适用于大型的实施
- 灵活性 — 能够解决许多技术问题
- 适应性 — 能够处理变化

前两个准则常常在许多关于 ETL 工具的文章中看到，但最后一个准则不幸被大多数文章所遗漏 — 即使它是任何技术工具决策的最重要的准则之一。让我们着眼于这些准则，并看看 Oracle Warehouse Builder 如何实现可伸缩性、灵活性和适应性。

可伸缩性和性能

可伸缩性和性能涉及到开发人员工作的多个方面。讨论通常围绕着加载性能，或者您能够多快地加载数据。我们将讨论这一点，但同时我们别忘了该工具自身的可伸缩性可能成为设计仓库的一个主要的因素。

在 Oracle Warehouse Builder 的情况下，工具在各种客户站点上进行了测试，能够为小型数据中心以及大型的全球数据仓库处理元数据。在后面一种情况下，Oracle Warehouse Builder 已被证明拥有一个可伸缩的信息库，从而允许一些客户在单个信息库中存储成千上万的表和复杂的映射对象。一个例子是 Oracle 嵌入式数据仓库 (EDW)，它在 Oracle Warehouse Builder 信息库中拥有整个 Oracle 应用程序模型的所有提取和表。这是真正的企业级元数据。

回到讨论的主题 — 您能够多快地加载数据 — 这是 Oracle 数据库能够多快地加载数据的问题。问这个问题的原因很简单：Oracle Warehouse Builder 创建了可在 Oracle 数据库上运行的高效代码。几个使用 Oracle Warehouse Builder 的客户运行了一个几千 G 字节的数据仓库，并使用 Oracle Warehouse Builder 来刷新数据。目前使用 Oracle Warehouse Builder 的更大的数据仓库之一是 2.5 TB，它为欧洲电信公司构建。

灵活性

只要您在 Oracle 数据库上运行您的仓库，Oracle Warehouse Builder 就为您提供了许多灵活性。灵活性的至关重要的方面之一是工具能否处理多样化的系统需求。我能否将数据加载到一个 OLAP 数据库？我能否从纯文本文件中加载数据？我能否将数据加载到一个第三范式的操作数据仓库中？我能否在 Oracle Warehouse Builder 中创建维度和事实数据？这些只是您将问的和 Oracle Warehouse Builder 能为您解决的问题中的一些例子。

除此之外，遵循开放标准允许您作为一个用户更灵活地把这一工具嵌入到您当前的基础架构中。如果您在 Oracle Designer 中设计您的模式，那么您可以将这些定义导入到 Oracle Warehouse Builder 中并在 ETL 系统上工作，然后将这个 ETL 系统部署到您的目标数据库中，在其中代码可以容易地与您现有的模式设计并存。

如果您打算使用 Oracle Warehouse Builder 作为一个元数据仓库，那么您将很高兴了解到您可以将该信息库扩展，以存储您自己的常见对象上的属性。

适应性

推动任何工具选择的主要原因是工具能够多容易地适应您以前的设计，以及能够多容易地将其升级到当前的新设计中。这（而不是其它因素）正是使 Oracle Warehouse Builder 区别于其它所有工具的原因。对象级的依赖性管理允许您即时地预测和检测变化，而不妨碍日常业务。

与其它 ETL 工具相比使得 Oracle Warehouse Builder 在这个领域中看起来非常不错，而将 Oracle Warehouse Builder 与最常见的实施（手动编码）相比就近乎不公平了。这就像拿汽车和自行车相比较，然后得出结论说汽车将更快。

为了理解这一点，让我们再看看之前的示例。在我们的销售收入数据中心中，来自 ERP 系统的订单项目表在许多 PL/SQL 过程中用于执行收入的提取和计算。现在这个表由于 ERP 系统中的一次定制（其中对某些订单增加了折扣）发生了变化（例如增加了一列）。这个变化影响了从这个时间点起加载的数据，因此您将必须将这个变化合并到系统中。

如果您是手动编写 PL/SQL 的，那么您现在被迫必须查看您所有的代码，搜索受影响的区域。此外，如果您不是代码的原始作者，那么在您能够做必需的修改之前您需要首先理解它。一旦您找到了所有的区域并做了修改，您将必须测试所有的代码。这不只是一项非常耗时的工作；它还容易出现错误，并对您的项目资源影响很大。

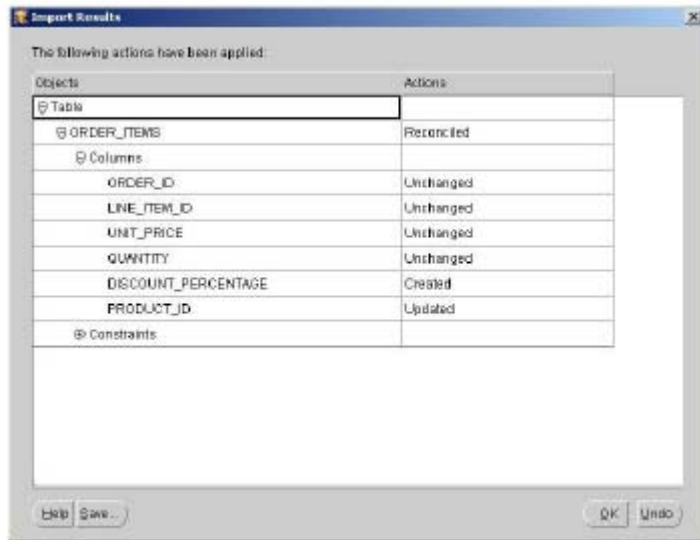


图 4 在 Warehouse Builder 中查看您的元数据的变化

如果在这种情况下您使用了 Oracle Warehouse Builder，那么您在查看差异之后（图 4）就能够导入新的订单表定义。在此之后，您将通过运行图 5 中的影响分析报表来评估整个系统的变化。

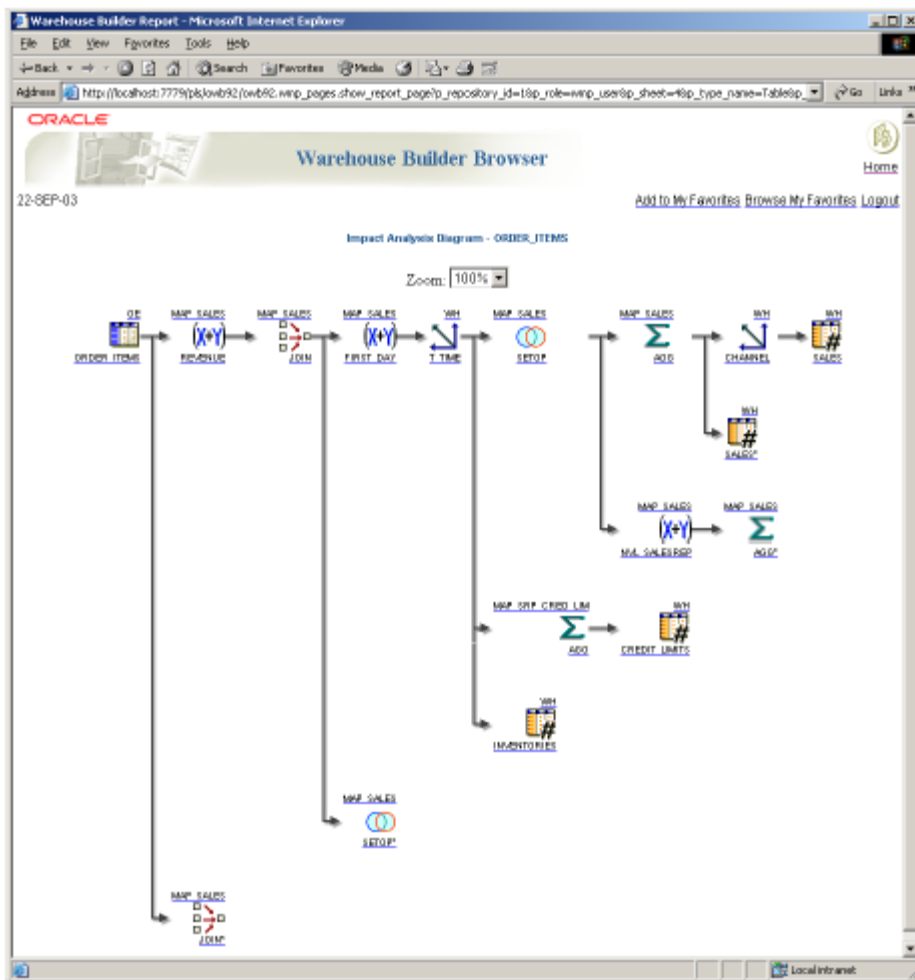


图 5 为一个小型项目中的 order_items 表确定的影响分析

在此之后，您根据影响分析报表的指导来核对修改并更新元数据。

现在您修改了您的 ETL 过程，以合并这些变化，接着您将需要重新测试它们。如果您是手动编写代码的，那么您将不得不同样手动测试并可能要调试这些过程。在 Oracle Warehouse Builder 中，您可以使用调试器来测试您的修改（图 6）。

调试器允许您逐行运行整个 ETL 过程和查看您已建模的每个步骤的结果。如果您在数据中检测到问题，那么您可以立即修补您正在查看的步骤。这不仅更快得多，而且还是一种更有序的提交更新的方式。即使在这个简单的例子中，这种技术提供的好处也是巨大的。

结论

正如许多用户所发现的，Oracle Warehouse Builder 是数据仓库环境中的一个强大的助手。从技术上讲，这一工具拥有许多的优点，这些优点能够使您的开发更快、质量更高，同时仍为您提供可伸缩性和高性能。利用其以数据库为中心的体系结构，它使您能够更好地利用您的数据库和支持仓库的硬件。

使这个工具业务智能化绝对很有意义。如果您将其与市场上其它的工具作比较，其价格定位几乎无疑将为 Oracle Warehouse Builder 提供比同类工具更高的 ROI。如果您将其与定制编码工作作比较，Oracle Warehouse Builder 可以让您以更少的精力完成更多的工作，再次为 Oracle Warehouse Builder 提高 ROI 并使其优于定制编码。因此如果您正在（或者将在下一阶段）为您的数据仓库作计划，那么请务必考虑 Oracle Warehouse Builder — 它可能使您甚至更加成功！

Oracle Warehouse Builder 10g — 在您转移数据的任何时候帮助您节省时间和金钱

2004 年 2 月

作者: Jean-Pierre Dijcks

Oracle Corporation

全球总部

500 Oracle Parkway

Redwood Shores, CA 94065

U. S. A.

全球咨询热线:

电话: +1.650.506.7000

传真: +1.650.506.7200

www.oracle.com

版权所有 © 2004, Oracle. All rights reserved.

本文档只用于提供信息, 其中的内容如有更改, 恕不通知。

不保证本文档中没有错误, 也不提供任何其它保证或

条件 (无论是口头表达还是法律暗示), 包括商用的隐

含保证和条件或者对特殊目的的适用性。我们明确拒绝

与本文档有关的任何责任, 并且本文档不构成任何直接或

间接的契约义务。未经预先书面许可, 不允许以任何形式

或任何方式 (电子或机械的)、出于任何目的复制或传

播本文档。Oracle 是 Oracle Corporation 和/或其会员的注册商标。

其他名称可能是其各自所有者的商标。