

基于数据仓库的通用 ETL 工具的设计与实现

陈 弦, 陈松乔

(中南大学 信息科学与工程学院, 湖南 长沙 410083)

摘要: 讨论了一种基于异构数据源通用 ETL 工具的设计方法和实现过程, 它能实现异种数据源的数据转换, 并将数据装载到中心数据库中, 具有一定易用性、灵活性和扩展性。该工具根据索引库来获得目标数据库与源数据库的映射关系。

关键词: ETL; 异构数据; 索引库

中图法分类号: TP311

文献标识码: A

文章编号: 1001-3695(2004)08-0214-03

Design and Implementation of General ETL Tool Based on Data Warehouse

CHEN Xian, CHEN Song-qiao

(College of Information Science & Engineering, Central South University, Changsha Hunan 410083, China)

Abstract: In this article, design methods and implementation techniques are discussed to realize the ETL among heterogeneous data sources based on all kinds of database. The general ETL tool with better flexibility, extensibility and capability of error handling has efficiently realized the transforming of Heterogeneous data and loading data into center database. The tool gain the mapping relationship between target database and source database according to index database.

Key words: ETL; Heterogeneous Data; Index Database

1 引言

数据仓库中的数据来自于多种业务数据源, 这些数据源可能是在不同的硬件平台上, 使用不同的操作系统, 因而数据以不同的格式存在于不同的数据库中。如何向数据仓库中加载这些数量大、种类多的数据, 已成为建立数据仓库所面临的一个关键问题。由于不同的事务处理系统必将用到不同的数据库系统, 包括不同的关系型数据库, 非关系型数据库, 甚至文件系统。如有的采用 Oracle 数据库管理系统, 有的采用 Microsoft SQL Server 数据库管理系统等, 先要建立全局的企业级 Intranet, 要求这些不同部门间做到数据共享, 实现全局数据一致性, 并提供全局 Web 查询和决策分析。通常, 企业的数据库源分布在各个子系统和节点中, 利用 ETL 将各地方业务系统上的数据进行抽取、清洗和转换处理, 然后加载到目的数据库。因为现有业务数据源多, 保证数据的一致性, 真正理解数据的业务含义, 跨越多平台、多系统整合数据, 最大可能提高数据的质量, 迎合业务需求不断变化的特性, 是 ETL 技术处理的关键。

ETL 是指从源文件或源数据库中获取数据, 并经过清洗、转换、集成后, 将其加载到数据仓库的过程。其中清洗是指去除那些在给定范围之外或不符合数据仓库要求的数据的操作。转换是将操作数据转换成另一种格式以更加适用于数据仓库设计。在大多数情况下, 转换是将数据汇总, 以使它更有意义。集成是将业务数据从一个或几个来源中取出, 并逐字逐段地将数据映射到数据仓库的新数据结构上。ETL 在整个数据仓库系统中的位置处于源文件或源数据库与数据仓库层之间。

ETL 同时提供数据质量的管理, 并且贯穿到整个商务智能解决方案的全过程, 完成整个系统的数据处理与调度。

为了从异构数据源中转换和集成数据, 各数据库厂商和其他软件开发商提供了很多专用工具来提取和转换数据。例如, Oracle 的 SQL* LOADER 可以转换一定格式的文本文件, 新推出的 Oracle Migration Workbench 可以将大量非 Oracle 对象及数据移植到 Oracle 9i 平台; Microsoft SQL Server 的数据转移服务(DTS), 允许用户在多种数据源之间输入和输出数据或在使用 SQL Server 的多个计算机之间转移数据库和数据库对象。这些工具主要通过 Open Gateway(数据库网关)来完成异构数据库之间的互访, 在一定范围内解决了数据的提取和转换。但是, 这些工具都有一定的局限性: 它们是属于特定的数据库系统的, 较依赖于具体的数据库厂商提供的产品, 通用性不强; 不能自动完成数据的抽取, 用户还需利用这些工具编写适当的转换程序; 数据往往是批量加载。

针对这些问题, 我们设计了一个通用性较强的 ETL 工具。该工具采用向导驱动方式和 GUI 图形用户界面, 可以进行各种数据库系统(如 SQL Server, Sybase 等)与 Oracle 数据库之间的数据存储、转换和调用。它能够搜索数据源找到目标表与源数据表的映射关系。它允许把数据库中的数据(一个或多个表中的部分或全部行)转入至目标数据库的一个表中(这个表可以存在或不存在)。

2 设计思想与总体结构

2.1 通用 ETL 工具的设计思想

在交互式界面上用户定义需要的目标基础表的结构, 由用

户填入表的字段、类型、主键等信息在中心数据库中建立目标表, 并且以树型结构显示这些目标表名和结构; 系统根据建立的索引库得知与某个目标表有关的各个源数据表, 并在页面中显示各表的结构。用户使用可视化界面来选定各个表、各个字段、表达式和函数。选择好后将定义好的映射关系存入元数据库, 在根据映射关系对各种异构的数据源进行提取和转换成统一格式的目标表, 所有这些操作最后都要形成一个可执行文件, 可定时地运行, 当用户需要修改时只需修改元数据库中的映射关系。通用 ETL 工具的系统流程图如图 1 所示。

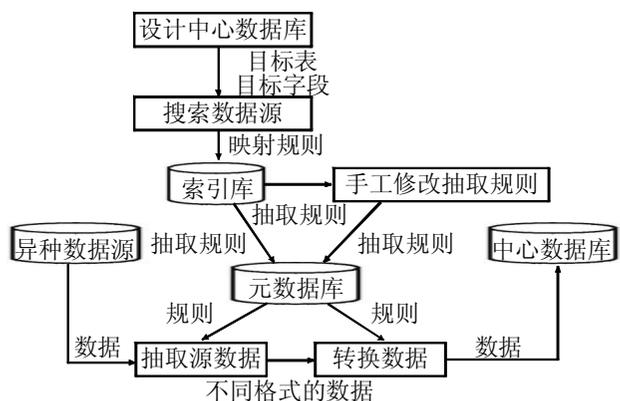


图 1 通用 ETL 工具的系统流程图

2.2 通用 ETL 工具的系统结构

由于 ETL 过程的复杂多样以及用户需求的多变, 设计一个通用的 ETL 工具要充分考虑各种可能的情况, 在此基础上还要使工具具有一定的通用性并且易于使用。系统工具分三个子模块: 数据呈现、数据抽取转换、中心数据库的建立(图 2)。

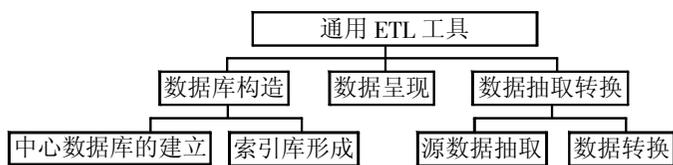


图 2 通用 ETL 工具的系统流程图

2.3 通用 ETL 工具各模块的功能

(1) 元数据库的设计

按照传统的定义, 元数据(Metadata)是关于数据的数据。从各个数据源中抽取的数据要按照一定的模式存入数据仓库中, 这些数据源与数据仓库中数据的对应关系及转换规则都要存储在元数据库中。在元数据的控制下进行统一的调度、管理和监控。

在数据源的识别中需要将非数据库文件的文件名和分隔符存入元数据库中, 以便文件的识别。

用户定义的源数据库与目标表的映射关系以及数据列之间的变换操作, 这些信息需要存到元数据库中。确定的映射关系以树型结构存入元数据库映射关系表(Mapping)中。

表 1 Mapping 表

字段名	类型	说明
targetid	char(6)	外键, 表示一个表的一个字段
souseids	varchar2(100)	每个 id 表示一个源数据库的一个源表的一个字段, 这里是多个 id 的集合
function	varchar2(100)	变换规则

映射表以上面的方法存储可以用尽可能少的查询代价来查询各映射关系。因为目标表的一个字段在表中只会有一条记录, 所有与该字段有关系的源字段都在一个 Souseids 字段中表示。变换字段是指由源数据表通过连接、汇总、计算等操作形成目标表的这些操作。我们将这些变换字段解析成可执行的脚本, 在抽取变换的过程中直接运行脚本。

(2) 数据源的抽取转换

对于关系数据库系统的数据源(如 SQL Server, Oracle, DB2, Sybase, Informix, Access 等), 我们通过 PL/SQL 从数据源中将所需的数据一行一行地进行抽取; 对于非数据库文件, 根据元数据库中的规则将其转换为 Oracle 数据库中的数据格式, 再进行抽取。对于数据源使用树型结构来存储数据源的信息, 如数据库名、表名、字段名等, 这样可以减少冗余, 提高查询效率。

在服务器端将抽取过来的数据利用不同的转换组件转换成统一的数据库格式, 存入本地的缓冲区。对于每个不同数据类型的数据源, 我们会调用不同的数据转换程序, 将其转换为统一的 Oracle 的数据类型。因此我们将这些转换程序都做成分件, 需要转换时调用相应的转换组件。

设计数据转换中不同数据库数据类型对应关系的数据存储结构, 将不同数据库系统数据类型的对应关系和相应的数据转换处理程序分离开, 使数据转换程序相对独立, 而把类型转换关系存储在专门的表结构中。找出了不同数据库系统和不同版本的各个不同类型之间缺省的类型对应关系及可能存在的对应关系, 将这些数据预先存入相应的表中。对于每个关系数据库系统的各种数据类型与 Oracle 数据类型的对应关系, 我们都将创建一张表来表示。

(3) 数据列的变换和连接

变换主要是针对数据仓库建立的模型, 通过一系列的变换来实现将数据从业务模型到分析模型, 通过内建的库函数、自定义脚本或其他的扩展方式, 实现了各种复杂的变换。数据变换是真正将源数据变换为目标数据的关键环节, 它包括数据汇总计算、数据拼接等。为了做到一个通用的工具, 将现在最为常用的操作集成为一个个的构件, 用户只需要拖拉构件来完成参与过程。主要有以下的变换类型: 聚集变换、表达式变换、过滤变换、分级变换、排序变换、串操作变换、数学函数。

数据加载主要是将经过转换和清洗的数据加载到目的数据库, 可以通过数据文件直接装载或以直连数据库的方式进行数据装载, 可以充分体现高效性。

(4) 索引库的建立

索引库是建立索引库表示目标数据表中字段名与哪些源数据有关。在索引库中我们建立了数据源和目标表的数据字典, 以记录数据源的每个数据库名、表名、字段名和目标表名及目标字段名。系统在已经识别了的数据源上自动搜索, 建立索引库。用户可以根据需要对已建立的索引库进行补充和修正。以目标表的字段为单位, 根据识别的数据源, 开始扫描源数据, 若源数据库中有表的字段名与目标表中的字段名相同或相似, 则找到该数据库该表该字段的字段对应的源 ID 号, 并将目的 ID 号和对应的源 ID 号填入映射关系索引表。当源数据都搜索完毕之后, 映射关系索引表就可以确定。

根据建立的索引表, 本系统可能把目标表的同一字段映射到多个数据源, 并把映射结果以图形化的方式显示出来。用户根据提供的结果, 可以进行选择和修改, 最后确定映射关系(目标表的字段对应的具体源数据)。图 3 是该模块的流程图。

当目标表发生了变化, 如新建了目标表或增加了目标表的字段。则需要修改目标表的数据字典, 并对新的字段扫描数据

源找到映射关系,更新映射关系索引表。

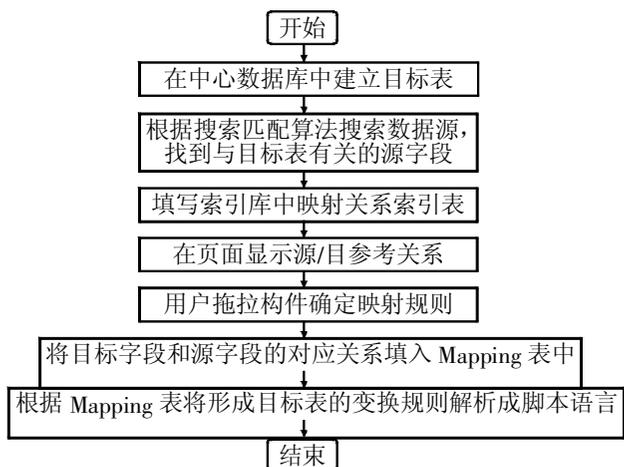


图 3 映射关系建立流程图

3 结束语

通用 ETL 工具的构件式的系统结构,使得程序和组件都能够灵活地进行扩展,以适应数据仓库需求和技术的变化及发展。此通用 ETL 工具采用向导驱动界面方式,向用户提供一个图形用户接口,具有很强的易用性。用户可通过各种友好的图形界面连接数据库,进行各种转换过程和数据映射设置,监视数据转换执行进程,查看转换日志,修改出错记录等。转换过程中可以定制数据类型的映射规则,设置数据转换的各种列映射规则,包括列名对应关系、类型转换关系、数据宽度和精度的设置以及数据表主键设置。可追加数据到目的数据库中与源数据库中不同表名、不同列名但列类型匹配的数据表中。

(上接第 213 页)

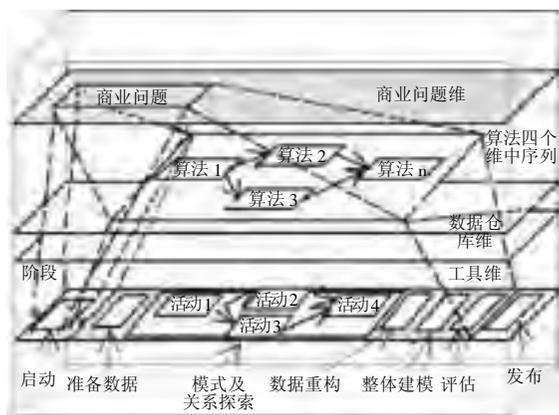


图 2 DM 过程多维结构中商业问题、算法集合和阶段活动的关系

事实上,开发一个在多维框架之下的过程,还需要大量的工作。当在一个多维框架的过程下来做数据挖掘工作时,涉及的主要技术及过程方面的困难问题都得到了解决。在这样的多维空间中,工作分解(WBS)、阶段控制、工件或交付物的定义、工具的选择和问题的可追踪性等将容易处理得多。同时会降低对实施团队的要求,因为通过过程我们尽可能多地把可以预见的问题预先处理了。

6 结论

数据挖掘是算法密集、对经验和商业领域知识非常依赖的工作。成功的数据挖掘项目是需要对各种因素作正确的判断,一个统一的不依赖于算法的过程框架对一个经验和相关知识不够充足的团队来说依然会无所适从。商业目标、算法和过程任务是相互紧密相关的,而流行的数据挖掘过程 CRISP-DM 和

转换前可以预览所设置的条件和数据,这些都使该工具具有很强的灵活性。所有这些操作最后都要形成一个可执行文件包,定时运行。用户使用本工具定制他们所需要的基础表的结构,系统能搜索存在于网络上的各种数据源中的相关的数据,并将其提取、转换到本地数据库中的目标表中,以方便将来的决策支持系统。

为了使通用 ETL 工具更加实用、灵活,需要进一步考虑索引库的更新策略。当目标数据库和源数据库发生变化时,索引库能及时地自动地更新,同时在索引库的建立上我们可以优化搜索算法,使得系统能更精确地找到与目标表有关的数据源。

参考文献:

[1] heth AP, et al. Federated Database Systems for Managing Distributed, Heterogeneous and Autonomous Database [J]. ACM Computing Surveys, 1990, 22(3): 183-203.

[2] 姚领众. 一种基于中间库的数据库间数据转换技术 [J]. 计算机系统应用, 1996, 11: 27-29.

[3] 梁鹰, 罗伟其. 异构数据库的数据转换在大型信息系统中的实现 [J]. 计算机工程与应用, 2000, 9: 103-105.

[4] 王存思, 黄庆荣, 傅清祥. 异构数据库间的数据复制技术及其应用 [J]. 福州大学学报(自然科学版), 2001, 29(3): 21-23.

[5] 连立贵, 金凤, 蔡家楣. 数据库中的数据提取 [J]. 计算机工程, 2001, 27(9): 61-62.

[6] 王元珍, 李海波. 基于 OLE DB 的数据抽取、转换和装入工具的设计与实现 [J]. 小型微型计算机系统, 2002, 23(1): 453-455.

作者简介:

陈弦, 硕士研究生, 主要研究方向为数据仓库; 陈松乔, 教授, 博士生导师, 主要研究方向为软件工程、分布式对象技术。

方法论 SEMMA 并没有把这些因素的相互作用考虑进去。本文认为数据挖掘过程应该置于商业目标、算法和过程任务等多维框架之下。开发一个在商业目标、算法和过程任务等多维框架之下的过程, 还需要大量的工作。本文讨论的问题可用于指导数据挖掘项目开展, 亦可作为数据挖掘过程设计的 CASE 工具设计的初步框架。

参考文献:

[1] avid Hand, Heikki Mannila, Padhraic Smyth. 数据挖掘原理 [M]. 张银奎, 等. 北京: 机械工业出版社, 2003.

[2] avid J Hand. Statistics and Data Mining: Intersecting Disciplines [J]. ACM SIGKDD, 1999, (1): 17-19.

[3] hi-Hua Zhou. Three Perspectives of Data Mining [J]. Artificial Intelligence, 2003, 143 : 139-146.

[4] obert L Grossman, Mark F Homick, Gregor Meyer. Data Mining Standards Initiatives [J]. Communications of the ACM, 2002, 45(8): 59-61.

[5] Dnuggets. What Main Methodology Are You Using for Data Mining? [EB/OL]. http://www.kdnuggets.com/polls/index.html, 2002-07.

[6] hapman P, Clinton J, Khabaza T, et al. The CRISP-DM Process Model 1.0 [EB/OL]. http://www.crisp-dm.org/CRISPWP-0800.pdf, 2000.

[7] 赵明德. 关于数据采矿中取样概念的一些探讨 [J]. 统计学教学, 2001, 24: 1-10.

[8] ohn Brocklebank, et al. Data Mining and the Case for Sampling [R]. by SAS Institute Inc., 1998.

[9] om M Mitchell. 机器学习 [M]. 曾华军, 张银奎, 等. 北京: 机械工业出版社, 2003.

[10] elp Document. SAS Enterprise Miner for Windows NT [Z]. Release 4.1 SAS Institute Inc, 2000.

作者简介:

潘无名, 博士后, 主要研究方向为人工智能、软件工程、数据仓库和商务智能; 潘云鹤, 教授, 院士, 主要研究方向为人工智能、认知科学。