一种可靠的数据仓库中 ETL 策略与架构设计

尤玉林　张宪民
（上海交通大学图像处理与模式识别研究所，上海 200030）
E-mail：yl_you@sjtu.edu.cn

摘　要　作为数据仓库系统的关键部件，ETL 完成数据抽取、清洗、转换和装载的工作，它是构建数据仓库的重要环节，同时也是构建数据仓库过程中出现问题最多的环节，所以针对这点，该文给出了一个可靠的同时易于扩展的 ETL 策略和架构。文章首先简单地介绍了数据仓库技术和 ETL 技术，包括 ETL 的相关概念、ETL 在数据仓库中的功能和重要地位；然后重点介绍了这种 ETL 的具体策略和架构设计。

关键词　数据仓库　ETL　数据抽取　数据转换　数据清洗　数据装载

文章编号 1002-8331-（2005）10-0172-03　**文献标识码** A　**中图分类号** TP311.13

A Reliable Strategy and Design of Architecture of ETL in Data Warehouse

You Yulin　Zhang Xianmin
（Institute of Image Processing & Pattern Recognition，Shanghai Jiaotong University，Shanghai 200030）

Abstract：As the key component in the data warehouse system，ETL supports the processing about data extracting，cleaning，transforming and loading.It is one of the most important steps in building the data warehouse，at the same time，there are a lot of bugs about ETL in building the data warehouse.To avoid those potential bugs，this paper puts forward a reliable and easily distensible strategy and architecture of ETL.This paper briefly introduces the technology of data warehouse and ETL，including the concepts related with data warehouse and ETL，ETL's functions and the important location in data warehouse system，and then it emphasizes the details about this strategy and design of architecture of ETL.

Keywords：data warehouse，ETL，data extract，data transform，data clean，data loading

1　引言

作为数据仓库系统中最基本而且极为重要的一部分——ETL，它是数据仓库的核心技术之一，它将为数据仓库提供高质而准确的数据。

目前，国外关于数据仓库的定义很多，业界公认的数据仓库概念的形成是以被称为“数据仓库之父”的 W.H.Inmon 出版《Building the Data Warehouse》一书为标志。该书对数据仓库作了这样的定义：数据仓库就是面向主题的、集成的、非易失的、随时间变化的数据集合。

但就数据仓库的实质来讲，它可以被视为一个存储了依据业务需求经过转换和清洗后数据的数据库。数据类别及子目取决于业务人员及决策者对信息的要求。存储的原则是易存、易取、易用而且有效（时间和空间）。

一个数据仓库系统综合了多个部分（ETL，原始数据库，报表生成）、与多个系统接口（用于数据交换），同时生成的结果供各个部门的业务人员和决策者使用。设计数据仓库的一般步骤如图 1 所示。

图 1　数据仓库设计一般步骤

数据的准确性和一致性是一个成功的数据仓库必须具有的特点。因为数据仓库本身依赖于各个业务系统（数据源），同时灵活性也是一个成功数据仓库的关键。所以，如何有效地从源数据中把需要的数据加载到数据仓库中是至关重要的一步。

数据仓库的数据源一般是存储在异构数据库中的业务系统数据。根据业务需求，从这些数据库中抽取相关数据，并进行转换和清洗，然后同步或者异步的方式装载到数据仓库中。这是一个工作量巨大的作业，根据已有经验，这也是日常运作中问题最多也最为繁琐的部分，ETL 就是完成这部分工作的。而且数据仓库中数据的质量是数据仓库项目成功与否的最主要判断标准，所以 ETL 部分的设计成为整个数据仓库系统设计中最重要的部分之一。

2　ETL 简介

ETL 是指数据抽取（Extract）、数据转换（Transform）以及数据加载（Loading），是构建数据仓库中极其重要的一环，其在数据仓库系统中的位置如图 2 所示。

2.1　ETL 的功能

ETL 首先要做的是按业务需求从源数据（业务系统/外部数据等）中抽取（Extract）数据仓库所需要的数据；然后对抽取

作者简介：尤玉林（1979-），男，硕士研究生，主要研究方向：数据仓库和数据挖掘。张宪民，男，硕士生导师，主要研究方向：模式识别，人工智能，视频图像检索。

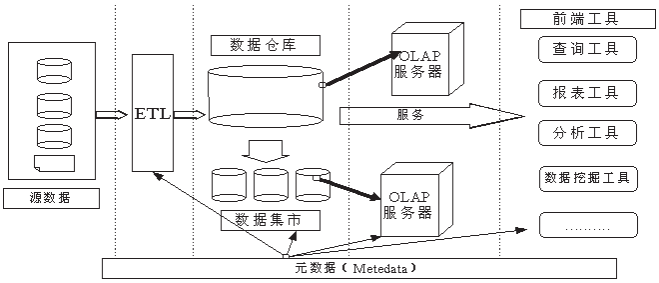

图 2　数据仓库系统框架

出来的数据进行变换（Convert）、转换（Transform）、清洗（Cleaning），去除不必要信息，转化为数据仓库要求的统一格式，再进行必要的处理；最后，将数据按着物理数据模型定义的数据结构类型装载（Loading）到数据仓库中。这个阶段必须要考虑到异常情况，比如空值处理、字段类型或长度不符合要求等。

2.2　ETL 的必要性

ETL 是必要的，这主要是由所抽取的源数据和数据仓库功能决定的。

首先，源数据是分布式异构的。可以是企业内部业务数据，来自不同的部门，也有可能是外部数据，所以为了保证数据仓库的数据是全面的、可信的，必须要把这些不同来源数据抽取出来，整合在一起。

其次，由于数据来源的多样性，所以数据表示形式必然不一致，其内容甚至可能相互矛盾。这种情况下，必须要对从各个不同来源的数据进行变换，统一表示形式，同时对相互矛盾的数据辨识真伪，保证加载到数据仓库数据的统一性和准确性。

再次，由于数据来源的多样性，不可避免地会造成数据的空间和时间冗余度较大，因此须对数据进行清洗，保证数据的唯一性。

还有，在 ETL 过程中，对数据进行适当处理，比如变换、关联、拆分或合并等，加上时间戳及其他特性，形成符合物理数据模型要求的多维数据，才能将数据顺利加载到数据仓库中。

从用户角度来考虑，经过 ETL 将不同来源数据集中到数据仓库中，使企业有了一个更好的信息平台，打破先前部门界限，方便企业内部信息流通，使各部门得到信息将更方便，这必将提高各部门决策水准。

2.3　ETL 在数据仓库系统中的重要性

从数据仓库的系统架构可以看出，ETL 是数据仓库中的非常重要的一环。它是承前启后的必要的一步。相对于关系数据库，数据仓库技术没有严格的数学理论基础，它更面向实际工程应用。所以从工程应用的角度来考虑，按着物理数据模型的要求加载数据并对数据进行一些系列处理，处理过程与经验直接相关，同时这部分的工作直接关系数据仓库中数据的质量，从而影响到在线分析处理（OLAP）和前端工具处理的结果的质量。

数据仓库是一个独立的数据环境，需要通过抽取过程将数据从联机事务处理环境、外部数据源和脱机的数据存储介质导入到数据仓库中；在技术上，ETL 主要涉及到关联、转换、增量、调度和监控等几个方面；数据仓库系统中数据不要求与联机事务处理系统中数据实时同步，所以 ETL 可以定时进行。但多个 ETL 的操作时间、顺序和成败对数据仓库中信息的有效性至关重要。

总之，ETL 是数据仓库系统中非常重要的组成部分，它从数据源中抽取、转换和加载数据到数据仓库或数据集市中，以备前端工具分析使用。

3　ETL 的设计与实现

3.1　ETL 的逻辑架构

一个 ETL 系统需要能够在限定的时间内完成对日常数据周期性的自动加载，支持对初始数据及历史数据的加载，并满足未来扩充的要求。数据仓库系统中数十个或者更多目标数据表及其相应数量的源数据意味着 ETL 程序的复杂性，庞大的数据量则需要充分考虑系统运行的效率，为方便开发复杂的程序，就要求灵活而简单明了的程序结构；而程序的效率的优化的要求又往往需要针对不同数据做个性化设计。因此，ETL 的设计必须在开发的可管理性和程序性能之间取得平衡，有些实现复杂、个性化突出的做法就要让位于要求一致的 ETL 程序结构。太注重对不同数据的个性化设计，给 ETL 测试和维护造成很多隐患，很有可能在运行过程中带来不稳定性。所以，这样的平衡应是 ETL 设计中很重要的参考因素。

在基于此设计思路的 ETL 策略下，每个数据表的 ETL 流程都按照 ETL 的特性统一分为 3 个标准步骤，即数据抽取/变换（Extract/Convert）、数据转换（Transform）和数据加载（Loading），所有数据的 ETL 都被纳入到这个标准框架中。因此，所有需开发的 ETL 程序的流程也就被对应地分为 3 个主要的步骤，每个步骤需要记录完整的处理中间状态及完善的日志信息。对于一个开发团队来说，遵循统一的架构开发可以保证每个开发人员开发的程序的结构一致性，便于 ETL 的管理，同时对于测试和维护人员来说，根据不同步骤的中间状态记录及日志信息也很容易定位及修正程序的错误。

图 3 是该 ETL 系统逻辑架构示意图。从宏观设计上，历史数据、初始数据加载和日常数据加载的 ETL 都将按照此架构设计。该架构将 ETL 作为一个整体来设计。

对于数据仓库的加载，ETL 分为数据抽取（Extract）、数据变换（Convert）、数据转换（Transform）以及数据加载（Load）4 个阶段。每个阶段之间以文本文件作为接口，即数据抽取（Extract）阶段读取数据源产生 EXF（Extract Format）文件，CSS（Converting/Sort/Split）阶段读取 EXF 文件产生 CIF（Common Interface Format）文件，数据转换（Transform）阶段读取 CIF 文件产生 PLF（Pre-Load Format）文件，数据加载（Loading）阶段读取 PLF 文件加载到数据仓库中。

此架构设计的优点是：将数据抽取、转换和加载分隔开，以 CIF 作为数据仓库表和数据源之间的桥梁，从而使每个功能相对独立，减少各功能相互间的耦合度；同时，每个模块的功能被细分后，逻辑更加简单，更容易控制开发错误，提高开发效率；另外，也便于系统运行过程中的错误追综和异常恢复。所以说，它是可靠的而且易于功能扩展的。

3.2　数据抽取（Extract）

数据抽取是从数据源获取所需数据的过程。数据抽取过程会过滤掉数据仓库中不需要的源数据字段或数据记录。

在数据抽取之前，首先要考虑源数据环境和 ETL 开发环境的接口问题。对于不同平台、不同形式、不同业务和不同数据量的源数据应采取不同的数据抽取接口。典型的源数据接口有数据库接口（ODBC、OLEDB、专用数据库接口等）和文件接口。

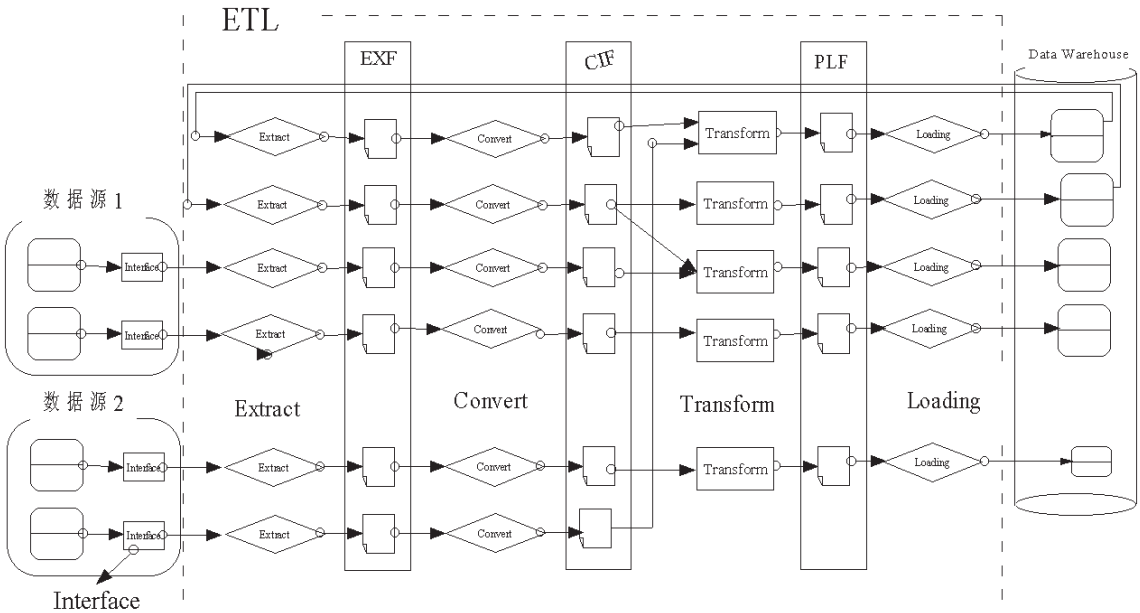

图3 ETL逻辑架构

根据ETL实际,考虑抽取的效率和可靠性,选择合适的源数据接口。

数据抽取可以采用PULL和PUSH两种方式。PUSH就是指由源系统按照双方定义的数据格式,主动将符合要求的数据抽取出来,形成接口数据表或数据视图供ETL系统使用。采用PUSH的方式会对源系统或其他开发团队产生依赖,对源系统的性能和网络有较高要求。PULL则是由ETL程序直接访问数据源来获取数据的方式。这种方式ETL工作就比较独立,但是要自己进行数据抽取工作。所以应根据实际项目要求选择合适的方式。

3.3 数据变换(Convert)和分割(Split)

数据变换的任务是逐条记录的检查数据,将每个字段转换为遵循数据仓库标准的数据格式,即对数据类型和数据格式进行转换,并对空字段赋予适当的缺省值,形成规整的数据结构,对于不符合要求的数据,写入拒绝(Reject)文件中。

数据变换主要的工作有:格式变换,如所有日期格式统一为yyyy-mm-dd;赋缺省值,在数据仓库中定义取值不为空的字段在源数据对应的字段可能存在没有取值的记录,这时根据业务需要,可能有两种处理办法,一是将该记录写入到拒绝文件中,由业务部门根据拒绝文件检查并修补源数据,另一种是在数据变换阶段直接赋一个缺省值;类型变换,如将源系统的Number类型转为char类型等;长度变换,如将源系统中定义的char(10)转为char(20)等;代码转换,如源系统的某些字段经过代码升级以后,将老的代码转为新的代码等;数值转换,如数值单位由万元转为元等。

此外,同一个数据源表的数据可能被多个数据仓库表使用,这就需要将一个数据源按不同的条件通过数据抽取和变换过程分成多个CIF文件以对应于不同的目标表的转换加载。

3.4 数据转换(Transform)

数据转换(Transform)是按照目标表的数据结构,对一个或多个源数据的字段进行翻译、匹配、聚合等操作得到目标数据的字段。

数据转换主要包括格式和字段合并(Merge)与拆分(Split)、数据翻译(Lookup)、数据排序(Sort)、数据匹配(Matching)、数据聚合(Aggregate)以及其他复杂计算等。

原则上,数据转换只处理规律而重复性大的数据聚合,如汇总、取平均值、找最大最小值等,而不用于复杂计算,以减少开发成本和系统负载。对于不规律而且复杂的计算,可由源系统端将数据计算好。

3.5 数据加载(Loading)

加载(Load)主要完成将前面生成的PLF文件的数据加载到数据仓库的表中。

根据模型的设计和源数据的情况,有4种数据ETL模式,所以就有4种不同的加载方式。

(1)刷新(Refresh,Type 1):数据仓库数据表中只包括最新的数据,每次加载均删除原有数据,然后完全加载最新的源数据。如大多数参数表的加载都采用这种模式。

这种模式下,数据抽取程序抽取源数据中的所有记录,在加载前,将目标数据表清空,然后加载所有记录。

(2)镜像增量(Snapshot Append,Type 2):源数据中的记录定期更新,但记录中包括记录时间字段,源数据中保存了数据历史的记录,ETL可以通过记录时间将增量数据从源数据抽取出来以附加的方式加载到数据仓库中,数据的历史记录也会被保留在数据仓库中。

(3)事件增量(Event Append,Type 3):每一个记录是一个新的事件,相互之间没有必然的联系,新记录不是对原有记录数值的变更,记录包括时间字段,可以通过时间字段将新增数据抽取出来加载到数据库中。

(4)镜像比较(Snapshot Compare,Type 4):源数据每天都被更新。而数据仓库数据具有生效日期字段以保存数据的历史信息。因此,只能将新的镜像数据与上次加载的数据的镜像进行比较,找出变更部分,更新历史数据被更新记录的生效终止日期,并添加变更后的数据。大多数源数据中需保存历史信息的维表。

所以对应于上面4种ETL模式而需要用到的加载方式有

(下转229页)

进程的主要内容如下：

```
Main ……
  r *R :15 e q
    i $c (R)'=stx q
    s record="" f r *R q :$c (R)=etx s record=record_$c (R)
     ;以累计方式将从设备读取出的字符记入字符串变量 record 中，
     ;直到遇到结束符 etx 为止
    s type=$e (record,1)d trace^MIF000 (mi,record,"H<--M")
     ;设置变量 type 为 record 的第一个字符
    i type="P" d ACK,POL q
    i type="M" d ACK,KTMIF q
    i type="R" d ACK,RES q
    i type="C" d ACK,CAL q
     ;根据不同的 type，调用不同的进程段以做出不同的响应
    ……
    q
  ……
  POL ……
    s xx2=$p (record,del,4) ;将 record 的第 4 个字符赋给变量xx2
    ……
    i xx2=0 d N51 q    ;若设备 '忙 '则等待
    …… ;组成以 'D '打头的医嘱字符串 rec，
    d SEND (rec) ;向设备发送 rec
    q
KTMIF …… ;清空医嘱字符串
RES    …… ;数据采集并调用入库进程将结果入库
    s rec="M"_del_"A"_del_del
    d SEND (rec) ;生成成功接收结果的响应信号并向设备发送
    q
CAL  s rec="M"_del_"A"_del_del
    d SEND (rec) ;生成成功接收设备信号的响应信号并向设备发送
    q
N51  s rec="W" d SEND (rec) ;设备 '忙 '时向设备发送等待信号
    Q
SEND (rec)    …… ;向设备发送信号
……
```

对于其它设备，根据厂商自定义的信号或字符含义，与上述实例的开发过程类似，可以开发出相应的通讯进程。

4 结束语

最初为提高工作效率、缩短报告单发出时间、方便查询检验结果、缓解数字化设备测定的高速度与手工报告结果的低效率之间的矛盾而产生的实验室信息系统，发展到今天已经是一个比较成熟的系统。如何更好地利用检验设备的资源，节约投资成本，最大程度地避免工作差错，更多地减少手工操作步骤，简化工作流程，成为人们关注的问题。该系统采用的网络结构、流程设计在一定程度上解决了上述问题，并且在与 CIS 的联系上更为紧凑。系统在北京安贞医院运行的一年多时间里，收到了很好的效果，目前已联机的检验设备包括具备双向通讯和条形码识别功能的 5 台设备以及具备单向通讯功能的十几台设备。 （收稿日期：2004 年 7 月）

参考文献

1.李桂祥，王放.ORACLE 数据库下的联机检验系统设计[J].计算机工程与应用，2003，39（16）：218~220

2.王忠民，黄如春.临床检验信息系统设计[J].医疗设备信息，2003，18（4）：28~30

3.陈敏，兰小鹏，陈金雄.条形码检验信息管理系统[J].福建医药杂志，2003，25（6）：148~150

（上接 174 页）

3 种：

（1）插入（Insert）：只需要将 PLF 文件所有数据完全插入到目标表中。主要是对事件增量类型进行操作。

（2）增加：需要对目标表同时做更新（Update）及插入（Insert）操作，根据主键（primary key），对于已有的记录进行更新操作，对于不存在的记录做插入的操作，对于数据量大的表，由于此操作的效率非常低，可以采用先将 PLF 文件分割为要删除文件及需插入文件，然后先将要删除文件中的记录根据主键对应从数据仓库中删除，然后再从插入文件中将所有记录全部插入到目标表中。这种方式主要是对镜像增量和镜像比较的类型进行操作。

（3）刷新（Refresh）：即将目标表的数据完全更新。主要针对上面第一种类型进行操作。

总之，采用哪种加载方式主要根据效率和业务现实等多种因素。

3.6 ETL 进程调度

进程调度的功能比较单纯，就是在规定的时刻启动程序，并记录系统运行情况和运行结果。

不同数据表的更新周期不同，因此，进程调度软件需要能够支持日周月等多种不同的启动周期，并通过设定启动时间来确定每个任务在何时启动运行。

只有日常数据加载才需要考虑进程调度的问题。而对于初始数据及历史数据的加载，由于是一次性的工作，将采取手工启动加载的方式，所以无需制定对初始数据及历史数据加载制度化的进程调度。

4 结束语

该文具体地介绍了一种数据仓库系统中 ETL 策略和架构设计，对数据抽取、数据转换和数据加载的方法作了一定阐述。在上海通用汽车报表中心项目中应用证明，这里所给出的 ETL 方法具有良好的效果，比较有效的控制了 ETL 工作中可能会出现的重大失误，而且其功能也是容易扩展，按这种方法基本可以保证 ETL 工程开发的进度，加载到数据仓库中的数据质量较高，从而保证了构建整个数据仓库的性能。对其他数据仓库的 ETL 有借鉴意义。 （收稿日期：2004 年 7 月）

参考文献

1. 美）H Inmon 著.王志海，林友芳等译.数据仓库[M].第二版，北京：机械工业出版社，2003-03

2.王珊等编著.数据仓库技术与联机分析处理[M].北京：科学出版社，1999

3.Michael F Jennings.Strategies for Custom Data Warehouse ETL Processing，2000

4.陈德军，盛翊智，陈绵云.基于数据仓库的 OLAP 在 DSS 中的应用研究[J].计算机工程与应用，2003，39（1）：30~31，61

5.王文彬，伍庆华，吴国平.基于移动行业的 ETL 方法及策略探讨[J].计算机工程，2003，29（1）