第18卷 第10期
2008年10月

计算机技术与发展
COMPUTER TECHNOLOGY AND DEVELOPMENT

Vol.18 No.10
Oct. 2008

ETL过程的思考

王 亮,葛 玮
(西北大学 软件工程研究所,陕西 西安 710127)

摘 要:ETL是构建数据仓库的一个非常重要的环节,可以这样认为:ETL就是整个数据仓库系统乃至整个决策支持系统的基石。如何设计高效的ETL过程就成为了众多计划或正在实施数据仓库项目的企业考虑的重要问题。从前期的数据理解阶段入手,分别讨论了数据的抽取、清洗转换、装载等不同阶段需要考虑的设计问题及相应的解决方案。提出了以数据理解为根基,以清洗转换为中心的设计思想,并给出了具体的实施步骤。

关键词:ETL;商业智能;数据仓库
中图分类号:TP311 **文献标识码**:A **文章编号**:1673-629X(2008)10-0130-03

Thinking in ETL Process

WANG Liang,GE Wei
(Software Engineering Institute of Northwest University, Xi'an 710127,China)

Abstract:ETL is one of the major processes when building a data warehouse. Can consider that ETL is the basis of the data warehouse even though the whole decision support system. Many enterprises which is planning or beginning to build a data warhorse now is highly concerned about how to design the ETL process effectively. Starts with the data comprehension, and discusses the design issues and its solutions about data extraction, data cleaning and data loading. Propose a method which is based on data comprehension and centered on data cleaning to design the ETL process and describe the steps to follow.

Key words:ETL; business intelligence; data warehouse

0 引 言

在构建商业智能系统的时候,如何正确有效地将分散在各个不同数据源中的信息整合到系统中成为了整个系统成败的关键,直接影响到系统的运行效率和最终结果。ETL正是解决这一问题的有力工具。ETL是指把数据从数据源装入数据仓库的过程,即数据的抽取(Extract)、转换(Transform)和装载(Load)过程。ETL过程的实质就是符合特定规则的数据流动过程,从不同异构数据源流向统一的目标数据。其间,数据的抽取、清洗、转换和装载形成串行或并行的过程,每个过程都必须符合特定的规则。根据国内外众多实践得到的共识,ETL规则设计和实施所需工作量约占整个项目的60%~80%[1]。由于ETL过程的重要性和复杂性,如何设计正确、高效的ETL过程已经成为了商业智能系统构建过程中无法回避的重要问题。

收稿日期:2008-01-23
基金项目:国家"863"计划资助项目(2004AA115090)
作者简介:王 亮(1982-),男,陕西咸阳人,硕士研究生,主要研究方向为计算机应用技术、数据仓库;葛 玮,副教授,硕士生导师,研究方向为软件工程、工作流。

从ETL的定义中可以看出ETL过程主要包括三个部分:数据抽取、数据转换以及数据的加载[2]。另外,为了更好地完成ETL过程的设计还需要对将要操作的数据有一定的了解,在这里将探查数据的过程称为数据理解。因此,在设计ETL的时候需要从以下几个方面进行考虑,那就是数据理解、数据抽取、数据转换和数据加载。

1 数据理解

在设计ETL过程之前,有一项非常重要但经常被人们所忽略的工作,那就是数据理解。这一阶段需要大量的调研和统计工作,需要了解数据的存储方式、数据量的大小、数据的格式、数据的业务含义等信息。同时还需要统计各种数值型数据的最大值、最小值和平均值,统计非数值型数据中各种不同的取值以及各种不同取值的个数。有了以上信息,ETL以后各个步骤的设计才能做到有的放矢,达到正确、高效的目的。

2 数据抽取

数据抽取是将数据从各个不同的数据源抽取到

ODS[3](Operational Data Store,操作型数据存储)中,在抽取的过程中需要挑选不同的抽取方法,尽可能地提高ETL的运行效率。在数据理解阶段,已经搞清楚了数据是从几个业务系统中来,各个业务系统的数据库服务器运行什么DBMS,是否存在手工数据,手工数据量有多大,是否存在非结构化的数据等相关的信息。根据这些信息,就可以开始进行数据抽取部分的设计。

2.1 不同数据源的处理方法

对于与存放DW的数据库系统相同的数据源,一般情况下,DBMS都会提供数据库链接功能,在DW数据库服务器和原业务系统之间建立直接的链接关系就可以直接访问;对于与DW数据库系统不同的数据源,一般情况下也可以通过ODBC的方式建立数据库链接。如果不能建立数据库链接,可以通过工具将数据导出成文本文件或者是二进制文件,然后再将这些文件导入到ODS中;对于文件类型数据源,可以培训业务人员利用数据库工具将这些数据导入到指定的数据库,然后从指定的数据库中抽取,或者还可以借助工具实现。

2.2 抽取过程中的数据清洗

由于数据可能来源于许多不同的系统,因此可能出现数据冗余甚至冲突的情况,这时可以在数据抽取时对重复或冲突的数据进行处理,一般情况下可以不抽取这一类数据。另外,有些明显不符合业务需求的数据,也可以在征得业务单位的同意之后直接舍弃,使其不得进入后续的过程。这样一来既可以在一定程度上提高抽取到的数据的质量也可以明显降低后续ETL步骤的负担,很大程度上提高了ETL的效率。

2.3 增量抽取的问题

对于数据量大的系统,必须考虑增量抽取。利用数据源的增量数据对数据仓库进行维护,可以有效提高ETL效率[3]。一般情况下,业务系统会记录业务发生的时间,可以用来做增量的标志,每次抽取之前首先判断ODS中记录最大的时间,然后根据这个时间去业务系统取大于这个时间所有的记录。若业务系统中没有记录业务发生的时间,则可以通过扫描日志文件的方式来获得增量信息。

3 数据的清洗转换

数据清洗转换实际上是利用有关技术如数理统计、数据挖掘或预定义的数据清洗转换规则将脏数据转化成满足数据质量要求的数据[4]。ETL三个部分中,花费时间最长的就是"T"(Transform,清洗、转换)的部分,一般情况下这部分工作量是整个ETL的2/3。通常,数据仓库分为ODS、DW两部分[5],目前的做法是从业务系统到ODS做清洗,将脏数据和不完整数据过滤掉,再从ODS到DW的过程中转换,进行一些业务规则的计算和聚合。

3.1 数据清洗

数据清洗的任务实际上就是过滤不符合要求的数据,将过滤的结果交给业务主管部门,由业务单位确认应该过滤掉或是修正之后再进行抽取。不符合要求的数据主要是有以下几种:

1)数据格式错误,例如缺失数据、数据值超出范围或是数据格式非法等。对于同样处理大数据量的数据源系统,通常会舍弃一些数据库自身的检查机制,例如字段约束等。他们尽可能将数据检查在入库前保证,但是这一点是很难确保的。这类情况诸如身份证号码、手机号、非日期类型的日期字段等。对于这一类错误,可以根据前期的数据理解阶段的各种统计分析结果结合SQL查找的方式找出来,然后要求客户在业务系统修正之后抽取。

2)不完整的数据:这一类数据主要是一些应该有的信息缺失,如供应商的名称、分公司的名称、客户的区域信息缺失、业务系统中主表与明细表不能匹配等。对于这一类数据过滤出来,按缺失的内容分别写入不同Excel文件向客户提交,要求在规定的时间内补全。补全后才写入数据仓库。

3)数据一致性,数据源系统可能会为了性能的考虑,而在一定程度上舍弃外键约束,这通常会导致数据不一致。例如在帐务表中会出现一个用户表中没有的用户ID,再例如有些代码在代码表中找不到等。对于这一类问题,应当由业务部门进行确认,修正后再进行抽取。

4)业务逻辑的合理性,这一点很难说对与错。通常,数据源系统的设计并不是非常严谨,例如让用户开户日期晚于用户销户日期都是有可能发生的,一个用户表中存在多个用户ID也是有可能发生的。对这种情况,比较合理的解决方案就是与业务单位进行交流,对于是否过滤,是否修正一般要求客户确认,对于过滤掉的数据,将其写入Excel文件或者写入错误数据表,在ETL开发的初期可以每天向业务单位发送过滤数据的邮件,促使他们尽快地修正错误,同时也可以作为将来验证数据的依据。这里需要注意的是不要将有用的数据过滤掉,对于每个过滤规则认真进行验证,并要求用户确认。

3.2 数据转换

在大多数情况下,数据转换是将数据汇总,以使它更有意义。在转换结构中,确保能找出一种最好的方法保证数据从传统的数据存储器到数据仓库的同

步[6]。具体说来,包括以下几个方面。

1)不一致数据转换:这个过程是一个整合的过程,将不同业务系统的相同类型的数据统一,比如同一个供应商在结算系统的编码是XX0001,而在CRM中编码是YY0001,这样在抽取过来之后统一转换成一个编码。

2)参照转换:在转换中通常要用数据源的一个或多个字段作为Key,去一个关联数组中搜索特定值,而且应该只能得到唯一值。这个关联数组使用Hash算法实现是比较合适也是最常见的,在整个ETL开始之前,它就装入内存,对性能提高的帮助非常大。

3)字符串处理:从数据源某个字符串字段中经常可以获取特定信息,例如身份证号。而且,经常会有数值型值以字符串形式体现。对字符串的操作通常有类型转换、字符串截取等。但是由于字符类型字段的随意性也造成了脏数据的隐患,所以在处理这种规则的时候,一定要加上异常处理。

4)日期转换:在数据仓库中日期值一般都会有特定的、不同于日期类型值的表示方法,例如使用8位整型20071001表示日期。而在数据源中,这种字段基本都是日期类型的,所以对于这样的规则,需要一些共通函数来处理将日期转换为8位日期值、6位月份值等。

5)日期运算:基于日期,通常会计算日差、月差、时长等。一般数据库提供的日期运算函数都是基于日期型的,而在数据仓库中采用特定类型来表示日期的话,必须有一套自己的日期运算函数集。

6)既定取值:这种规则和以上各种类型规则的差别就在于它不依赖于数据源字段,对目标字段取一个固定的或是依赖系统的值。

7)聚集运算:业务系统一般存储非常明细的数据,而数据仓库中数据是用来分析的,不需要非常明细的数据。一般情况下,会将业务系统数据按照数据仓库粒度进行聚合。

8)商务规则计算:不同的企业有不同的业务规则、不同的数据指标,这些指标有的时候不是简单的加加减减就能完成,这个时候需要在ETL中将这些数据指标计算好了之后存储在数据仓库中,以供分析使用。

4 数据加载

数据加载是将转换后的数据加载到数据仓库中。数据加载策略包括加载周期和数据追加策略,数据加载周期要综合考虑经营分析需求和系统加载的代价,对不同业务系统的数据采用不同的加载周期,但必须保持同一时间业务数据的完整性和一致性。

数据追加策略,根据数据抽取阶段的不同选择,又可以分为两种不同的方式:

一、在数据抽取阶段选择增量抽取,那么在数据转换完成后可以直接将数据加载到数据仓库中;

二、若在抽取阶段选择全量抽取,则可以根据数据的时间标记或源数据的操作日志或采用全表对比的方式选择相应的数据加载,几种方式各有优缺点,应针对不同的情况进行考虑。

5 结束语

ETL是BI项目的关键部分,也是一个长期的过程,同时这部分的工作直接关系数据仓库中数据的质量,从而影响到决策分析的结果的质量。在ETL过程中的每一步都会发现大量的问题,有些可以直接解决,有些则需要回溯到前一个甚至几个过程。通常情况下,每次对ETL过程的修改都需要重新运行整个ETL过程并对结果进行验证。这样一来,开发整个ETL过程的所需的时间必然很长。因此,只有认真、仔细地设计ETL过程中的每一步,尽量使得ETL过程每一步的运行效率更高、结果更准确同时更容易修改,才能有效保证整个BI项目的最终成功。

参考文献:

[1] 程跟上.基于公共仓库模型的ETL系统的研究和应用[D].南京:南京航空航天大学,2005.

[2] Rahmand E, Hong Haido. Data Cleaning, Problems and Current Approaches[J]. IEEE Bulletin of the Tenniel Committee Data Engineering, 2000, 23(4):3-13.

[3] 章水鑫,徐宏炳,于 立.增量式ETL工具的研究与实现[J].现代计算机,2005(3):6-10.

[4] Hernandez M. A Generation of Band Joins and the Merge/Purge Problem[R]. USA: Department of Computer Science, Columbia University, 1995.

[5] Inmon W H. Building the Data Warehouse[M].北京:机械工业出版社,2007.

[6] 张 宁,贾自艳.数据仓库中ETL技术的研究[J].计算机工程与应用,2002,38(24):213-216.

(上接第129页)

[5] Minakuchi Y, Satou K, Konagaya A. Prediction of Protein-Protein Interaction Sites Using Support Vector Machines[J]. Genome Informatics, 2002(13):322-323.

[6] Fariselli P, Pazos F, Valencia A, et al. Predication of protein-protein interaction sites in heterocomplexes with neural networks[J]. Eur. J. Biochem, 2002, 269:1356-1361.