**摘要:** 本文讨论了某数据集市项目的数据集成方法与过程。该系统在2008年12月启动，在2009年5月正式上线使用。该系统是以oracle系统为主要的数据库，同时集成DB2系统中的数据。每天的话费清单系在DB2数据库中存储，通过ETL调度程序把DB2中的数据进行汇总并把结果写入到ORACLE数据仓库中。本文首先讨论了建立数据集市项目异构数据库的两个数据库系统的背景以及用户对该项目的需求。接着讨论了使用Perl技术来集成两个数据库中的业务逻辑的过程，并说明了该技术在集成过程中出现的问题，如：数据分层，ETL调度程序改造，以及参数化SQL处理等问题。最后讨论了该集成方法的优点和缺点，并对改进该项目提出了优化Perl技术的设想。在本次的项目开发过程中，我主要担任了系统分析与设计的工作。

某数据集市项目是在2008年12月启动的，我所在的部门主旨了该项目的开发过程，我有幸在该项目中担任了系统分析与设计的工作。该项目的实施是基于以下的背景的。在2005年12月，我们部门已经组织了开发\*\*经营分析系统，该系统已经建立了数据仓库系统，把BOSS，财务数据，DSMP，MIS等项目的业务操作数据作为数据源，通过清洗，转换，装载等方式把所有的清单一级的数据全部集中到该系统中了，该经营分析系统是基于DB2的数据库开发而成的，把数据仓库分为了STAGE层，ODS层，EDS层，REF层。STAGE层是数据清单的原始数据，ODS层是经过日汇总的数据，EDS层是经过维度统一化后的数据，REF层是维度数据层。每一层数据之间的业务逻辑是通过ETL调度程序实现，该程序主要调用的是数据库中的存储过程或者Shell脚本完成每层中的数据汇总的过程。  
对于数据集市项目，我们比较有优势的就是该项目是基于上面的系统进行扩展而成，使用经营分析系统的经验和技术，我们可以快速开发完成。但是该项目由于企业的需求以及供应厂商等问题，最后企业选用了ORACLE数据库作为该项目的建立主要数据库，在该项目中必须把所有的EDS层的汇总数据记录在以ORACLE的主的数据库中以供企业的每个地市使用。 在上面的背景与需求下，对于采用两种数据库系统集成的问题是我们完成该项目的主要障碍。  
为了完成该项目的开发，我们对原有的系统和新开发系统的做了一些技术和方法的调整。这些方面主要有如下方面：  
一、重新规划数据层次。通过我们对用户需求的分析，对数据分布的理解，觉得了在集成异构数据库层的时候，我们采用尽量保护数据的原则。我们按照之前的数据分层方法，在ORACLE的数据仓库中，重新增加了两个层次的数据，分别是DM层和REF层，其中DM层的数据是EDS层数据的较大的粒度汇总过程，主要的来源数据是DB2的EDS层。通过调度业务逻辑程序，实现如下的数据汇总过程，STAGE层记录数据库清单的原始数据，没有经过任何的转换和变化，而ODS层主要是针对时间和地市维度进行的第一级汇总的数据，该数据基本上是以天为单位的历史数据。EDS层是对其他维护进行统一化后的汇总数据，该数据主要是对ODS数据进行转换和汇总的过程结果。DM层则是在EDS层的数据上，通过企业需要的业务逻辑，如每月统计数据，平均值，指标考核值，预测值等业务逻辑，把EDS层的数据汇总到DM层的数据中，并且通过特定的数据库权限和视图的方法，把各个地市需要查询的数据汇总到该层的数据库表中。通过这样的数据划分后，我们就等到的很好的数据分层结构，为下面进行异构数据库业务逻辑集成奠定了重要的基础。  
二、ETL调度程序的异构数据库处理。在规划了数据分层后，我们遇到的另外一个难题是ETL调度程序如何在这两个数据库中调度处理的过程，按照我们原先的设想是通过Shell程序来调度两个数据库中不同的存储过程，完成汇总数据的过程。但是这样的方法显然会出现异构数据库中很多的调度问题，而且会出现在调度程序中出现Shell再调用Shell等问题，使得业务逻辑混乱而无法以后的维护结果。为了缓解这样的问题，我们大胆的采用了Perl技术作为业务逻辑层的脚本处理平台。通过该平台，我们只要处理不同数据库的SQL就可以了，对于连接数据库以及调度的脚本的开发，调试，测试等提供了重要的保障，另外Perl程序另外一个特点就是可以面向对象，这样只要我们开发一些公共的模块，通过对象的形式，就很容易把业务逻辑的SQL嵌入到Perl程序中，从而解决异构数据库集成的问题。   
三、异构SQL的函数处理。在搞好可以统一调度的问题后，数据仓库基本上能够通过调度程序跑出对应的数据了，但是我们开发人员在开发的过程中非常的头疼，因为他们比较熟悉的DB2函数，在ORACLE中要寻找其他的替换方法，并且某些业务逻辑还要书写两套的SQL语句来处理业务逻辑。这样的妨碍了我们快速开发的目的。为了解决该问题，我们设计了一套通过的Perl函数库，该函数库式以SQL-92标注为基础，通过DB2中我们经常使用的函数集合的汇总过程，我们通过参数的输入设计方法，把按照我们标准编写的SQL语句转换成数据库中真正执行的SQL语句，通过正则表达式以及SQL字符串特定分解的过程，转换成特定数据库的SQL，并通过Perl函数执行该业务逻辑。这样，我们完成了异构数据库集成的中的SQL异构的处理。   
通过以上的三个种处理过程，我们最终解决了数据集市项目中的异构数据库问题，并且在2009年5月份完成了该项目的验收，把真正的数据通过经营分析系统的ETL调度程序下发到各个地市中，获得了用户的一致好评。同时由于该系统集成的时候充分考虑了开发人员的使用SQL习惯，所以开发的效率比较高，比使用存储过程的开发过程减少了三分之一的开发时间。  
但是该系统还是存在很多的问题，首先是异构SQL的函数还不是很多，并且对于一些特殊的函数，由于需要考虑到两个数据库转换等问题，曾经出现果效率的瓶颈问题，后来经过多方的考虑，把某些功能定义为只能有一个数据库使用的方法来解决。另外，在Perl执行的业务逻辑中，我们还是比较陌生，虽然开发处理了很多的通用模块，但是没有定义比较好的接口，如果以后需要改进的时候，可能会出现大面积的修改工作量。我们建议在Perl开发平台上，应该多做一些规划，如果能够把Perl模块做成数据库连接构件，SQL执行构件，函数转换构件等等构件化的形成后，并且定义良好的接口形式，这样会使数据集市项目在集成异构数据库上更加的好，同时可以集中解决一些性能的问题，以优化系统的运行效率。