**基于Saiku的电子商务联机分析处理系统 张霞红 指导教师 邱明 厦门大学**

2029320381452802095400

**本 科 毕 业 论 文 （设 计）**

**（主修专业）**

**基于Saiku的电子商务联机分析处理系统**

**On-line Analytical Processing System of Electronic Commerce**

**Based on Saiku**

姓 名：张霞红

学 号：24320132202521

学　 院：软件学院

专 业：软件工程

年 级：2013级

校内指导教师：邱 明 助理教授

**二〇一七 年 五 月 十四 日**

**厦门大学本科学位论文诚信承诺书**

本人呈交的学位论文是在导师指导下独立完成的研究成果。本人在论文写作中参考其他个人或集体已经发表的研究成果，均在文中以适当方式明确标明，并符合相关法律规范及《厦门大学本科毕业论文（设计）规范》。

该学位论文为（ ）课题（组）的研究成果，获得（ ）课题（组）经费或实验室的资助，在（ ）实验室完成（请在以上括号内填写课题或课题组负责人或实验室名称，未有此项声明内容的，可以不作特别声明）。

本人承诺辅修专业毕业论文（设计）（如有）的内容与主修专业不存在相同与相近情况。

学生声明（签名）：

年 月 日

致 谢

值此论文完成之际，谨向所有关心和支持我的人们致以诚挚的谢意！

首先，我要感谢厦门大学软件学院四年来对我的培养。在大学这四年里，我从各位老师身上学到了许多专业知识，也在学生活动中提升了各方面能力，感谢陪伴我成长的老师和同学们！

其次，我要衷心地感谢论文指导老师林坤辉教授和邱明老师。从论文选题、论述内容和整体目录、结构的确定，直到最后的定稿，二位老师都以极其认真负责的态度给予我悉心指导，提出了许多宝贵的意见和建议，使我获益良多。老师们渊博的专业知识、严谨的治学精神、朴实的学术作风以及精益求精的工作态度，都不断激励着我努力完善自己，对我的悉心关怀和谆谆教诲也鼓舞着我在今后的学习与工作上不断奋发向上。在此，谨向二位老师致以最诚挚的感谢！

然后，还要感谢与我一起完成这个系统开发的团队成员刘程、傅青杰。没有他们的帮助和共同努力，就没有最后的圆满成功，也就不会有本文的形成。在此，向他们表示衷心的感谢！

最后，我要感谢我的家人，没有他们的养育和支持就没有今天的我。在此，谨向他们表示由衷的感谢！

摘　要

随着数据库应用系统的迅速普及和人们对数据的分析需求日趋多样化、复杂化及综合化，数据仓库（DW）技术应运而生。联机分析处理（OLAP）是数据仓库的一个最典型的应用，它能将数据仓库中的数据按照不同的粒度级进行聚合和预计算，形成多维立方体。基于多维立方体，OLAP可进一步为用户提供较直观地观察数据仓库中数据的方法，同时可向用户提供多种编程接口，以便用户在开发应用系统时调用。

本次论文设计的主要任务是针对×××销售管理平台的交易状况进行多维建模和分析，从而以直观简单的图表方式呈现出不同交易的状况。该项目的主要内容包括三部分：第一，根据数据库现有的交易报表，通过分析数据源，建立维度和量度，设计维度的级别，最后设计多维立方体模型；第二，使用Java编程语言对源数据进行抽取、转换和清洗，最终按照预先定义好的数据仓库模型，将数据加载到目标数据库中。第三，基于Saiku平台，选择不同指标和维度进行分析，从而揭开其中的关系。

本篇论文将对上述三部分内容做详细的介绍，并配以相应的图表，以此阐明如何基于Saiku进行电子商务联机分析。首先介绍了论文整体的的组织结构，从研究背景出发，介绍了相关的技术和工具，接着详细描述了整个系统设计的过程以及多维分析的结果。最后总结了论文的研究内容，论文设计过程中的收获及学习心得等。

关键词：联机分析处理；多维立方体模型；Saiku

**Abstract**

Data Warehouse (DW) technology came into being as the rapid popularization of database applications and the increasing demand, complexity and integration of data analysis needs. Online analytical processing (OLAP) is one of the most typical applications of Data Warehouse and the data in the DW can be aggregated and precomputed according to different granularity, formating the multi-dimensional cube. Based on the multi-dimensional cube, OLAP can further support users with more intuitive ways to observe the data in the DW, and meanwhile can provide users with a variety of programming interface, so that users in the development of application system call.

The main task of this thesis is to carry out multidimensional modeling and analysis on the transaction status of the sales management platform, so as to show the status of different transactions in an intuitive and simple chart. The main content of the project includes three parts: First, according to the existing transaction reports of the data base, analysing the data sources, establishing the dimensions and metrics and designing dimension levels and finally designing a multi-dimensional cube model; Second, using the Java programming language to extract, convert and clean the data source, and finally load the data into the target database according to the pre-defined data warehouse model; Third, based on Saiku analysis platform, selecting different indicators and dimensions to analyze, thus revealing the relationship.

The three parts of the paper will be described in detail with the corresponding charts to clarify how to analyze E-Business based on Saiku. Firstly, introducing the organization structure of the paper and the related technology and tools according to the research background; Next, describing the process of the whole system design and the results of multidimensional analysis; Finally, summarizing the research contents of the thesis, the harvesting and learning experience in the paper design process.

**Key words:** Online Analytical Processing; Multidimensional Cube Model; Saiku

目 录

[第一章 绪论 1](#_Toc451354666)

[1.1 研究背景 1](#_Toc451354667)

[1.2 当前存在的问题 1](#_Toc451354667)

[1.3 论文设计的主要内容 1](#_Toc451354667)

[1.4 论文组织结构 2](#_Toc451354670)

[第二章 系统设计相关技术与设计原则 4](#_Toc451354671)

[2.1 技术架构 4](#_Toc451354672)

[2.2 数据仓库 4](#_Toc451354672)

[2.3 Mondrian 4](#_Toc451354672)

[2.4 OLAP联机分析处理 4](#_Toc451354672)

[2.5 ETL技术 4](#_Toc451354672)

[2.6 Saiku开源软件 4](#_Toc451354672)

[2.7 系统设计原则 10](#_Toc451354681)

[2.7.1 面向系统用户 10](#_Toc451354682)

[2.7.2 多维模型设计 10](#_Toc451354682)

[2.7.3 系统化思维 11](#_Toc451354683)

[2.8 本章小结 15](#_Toc451354691)

[第三章 电子商务联机分析处理过程 22](#_Toc451354701)

[3.1 OLAP需求分析 22](#_Toc451354702)

[3.2 数据仓库模型设计 29](#_Toc451354707)

[3.3 数据库表设计 29](#_Toc451354707)

[3.4 Mondrian Schema设计 29](#_Toc451354707)

[3.5 ETL 44](#_Toc451354712)

[3.6 本章小结 5](#_Toc451354718)5

[第四章 Saiku多维分析及测试 22](#_Toc451354701)

[4.1 Saiku连接数据库 22](#_Toc451354702)

[4.2 Saiku维表分析 22](#_Toc451354702)

[4.3 本章小结 5](#_Toc451354718)5

[第五章 总结与展望 57](#_Toc451354723)

[5.1 总结 57](#_Toc451354724)

[5.2 展望 58](#_Toc451354725)

[参考文献 59](#_Toc451354726)

**Contents**

[Chapter 1 Preface 1](#_Toc450829100)

[1.1 Introduction 1](#_Toc450829101)

[1.2 The Current Problems 2](#_Toc450829104)

[1.3 The Main Content of the Paper Design 1](#_Toc450829101)

[1.4 The Structure of This Dissertation 2](#_Toc450829104)

[Chapter 2 Related Technology and Design Principles 4](#_Toc450829105)

[2.1 Technology Architecture 4](#_Toc450829106)

[2.2 Design Data Warehouse 10](#_Toc450829115)

[2.3 Mondrian 12](#_Toc450829121)

[2.4 OLAP Theory 4](#_Toc450829106)

[2.5 ETL Technology 10](#_Toc450829115)

[2.6 Saiku - An Open Source Software 12](#_Toc450829121)

[2.7 System Design Principles 12](#_Toc450829121)

[2.7.1 To System Users 16](#_Toc450829128)

[2.7.2 Multidimensional Model Design 16](#_Toc450829128)

[2.7.3 Systematic Thinking 16](#_Toc450829128)

[2.8 Summary 15](#_Toc450829125)

[Chapter 3 E-commerce Online Analytical Process 16](#_Toc450829126)

[3.1 OLAP Requirements Analysis 16](#_Toc450829127)

[3.2 Data Warehouse Model Design 19](#_Toc450829131)

[3.3 Database Table Design 21](#_Toc450829134)

[3.4 Mondrian Schema Design 16](#_Toc450829127)

[3.5 ETL 19](#_Toc450829131)

[3.6 Summary 21](#_Toc450829134)

[Chapter 4 Saiku Multidimensional Analysis and Testing 22](#_Toc450829135)

[4.1 Saiku Connecting to the Database 22](#_Toc450829136)

[4.2 Saiku Multidimensional Analysis 44](#_Toc450829146)

[4.3 Summary 50](#_Toc450829152)

[Chapter 5 Conclusions and Feature works 57](#_Toc450829157)

[5.1 Conclusions 57](#_Toc450829158)

[5.2 Future works 58](#_Toc450829159)

[References 59](#_Toc450829160)

[Acknowledgements 61](#_Toc450909021)

# 第一章 绪论

## 1.1 研究背景

电子商务指在互联网、企业内部网和增值网上以电子交易方式进行交易活动和相关服务的活动，是传统商业活动各环节的电子化、网络化。十二五时期，我国电子商务行业发展迅猛，产业规模迅速扩大，电子商务信息、交易和技术等服务企业不断涌现。在国外，不少软件厂商采取了发展其前端产品来弥补关系数据库管理系统支持的不足，力图统一分散的公共应用逻辑，在短时间内响应非数据处理专业人员的复杂查询要求。联机分析处理（OLAP）系统是数据仓库系统最主要的应用，专门设计用于支持复杂的分析操作，侧重对决策人员和高层管理人员的决策支持，可以根据分析人员的要求快速、灵活地进行大数据量的复杂查询处理，并且以一种直观而易懂的形式将查询结果提供给决策人员，以便他们准确掌握企业的经营状况，了解对象的需求，制定正确的方案[1]。

数据仓库技术的发展以及OLAP技术的成熟和普及使得信息孤岛问题有望得到解决，它为实现数据的多维分析提供了一套整体的解决方案。目前，OLAP技术有了很大的发展，市场上的各种OLAP技术产品层出不穷。OLAP是以数据仓库或数据库为基础的，其最终数据来源是底层的OLTP数据库系统。Framework manager是将数据仓库或者数据立方体中的元数据经过组织发布到Cognos设计环境中的工具，在Framework manager中新建工程，并导入创建好数据集市的数据描述[2]。

## 1.2 当前存在的问题

目前，阻碍电子商务广泛应用的最大问题就是安全问题。计算机安全技术本 身的发展就存在一个时滞的问题。病毒感染、黑客的侵袭更使人们对计算机的安全性，特别是网络上电子商务的安全性产生怀疑。Internet的诞生并不是因为商业目的，而是为了能方便地共享计算机资源，所以Internet的TCP／IP协议及源代码的开设与共享是合理的，是符合当时需要的，要在Internet上进行安全要求高的电子商务活动也就显得勉为其难了。这种情况对电子商务的推行造成了极为不利的影响，使公众在电子商务前裹足不前。

因此，要在现在的基础上增加一些安全技术措施，如防火墙、加密、数字签名、身份认证等技术，以保证数据的保密性、完整性和不可否认性。另一方面，需要通过标准的形式把这些技术手段确定下来，包括制定必要的网络接人标准，允许在采用不同的信息技术的用户间进行电子商务活动；实现商业标准化环境，尽快改革中国传统商业的手工作业方式，适应数字化经济的社会；组织有关企业、研究单位进行电子商务的标准制定，这些标准应符合我国的实际情况，并积极向目标通用标准靠拢。

## 1.3 论文主要内容

本篇论文主要是有关如何基于Saiku开源软件实现电子商务联机分析处理。论文的主要内容包括以下几方面：首先是绪论部分，这部分的内容主要是阐述论文的研究背景、主要问题，明确论文的主要内容，以及对论文的整体组织结构进行规划，有提纲挈领的作用，对后续论文的展开做了良好的铺垫。第二部分的内容是对论文所用到的相关理论和技术背景做详细的介绍。在整篇论文中，相关的理论概念包括：数据仓库、联机分析处理、Mondrian以及ETL；接着对模型设计原则做了详细的描述，在整个联机分析处理过程中，我们要明确数据仓库的模型设计原则，以系统化的思维方式构建整体的模型框架；最后，着重对Saiku server进行了详细的介绍，在整个论文设计过程中，saiku技术是点睛之笔，是呈现多维表格的开源软件。接下来的部分对论文设计中的电子商务联机分析处理系统过程进行具体分析，整个设计过程，先要进行需求分析，分析本次联机分析处理中需要处理的事务；然后对Schema建模分析，分析模型的具体信息，最后是对数据执行ETL，完成ETL，即完成初期的联机分析的数据准备。最后对saiku开源软件如何分析及测试过程进行详述。

## 1.4 论文组织结构

论文从研究背景与意义出发，引出了本文的主要研究内容：联机分析处理，又介绍了相关理论概念、设计原则与技术背景。接着说明了电子商务联机分析处理的整个过程，从搭建OLAP开发环境，到建立多维立方体模型，ETL，通过Saiku呈现多维图形并分析。最后总结论文内容，学习心得等。

论文分为五章进行论述，组织结构如下：

第一章 绪论。本章介绍了研究电子商务联机分析的背景和研究的意义，当前存在的主要问题，以及论文组织结构。

第二章 相关理论与技术背景。本章介绍了数据仓库、联机分析处理、Mondrian以及ETL的相关知识和背景；在模型设计原则方面，涉及到面向用户设计、数据仓库模型设计和系统化思维的知识；最后重点介绍了Saiku server开源软件的体系结构、原理等。

第三章 电子商务联机分析处理过程。本章介绍了OLAP联机分析处理的需求研究；建立多维立方体模型，包括数据仓库模型设计和Mondrian Schema设计；ETL，具体包括对数据进行抽取、转化和清洗，最终加载到目标数据库。

第四章 基于Saiku平台呈现报表。本章介绍了以Saiku为开源平台的报表呈现过程，包括Saiku连接数据库和多维报表分析结果。

第五章 总结与展望。本章对论文内容进行了总结，并对项系统设计提出意见与展望，以及学习心得等。

# 第二章 系统设计相关技术与设计原则

## 在系统设计过程中，涉及到一些相关理论以及技术背景。在电子商务联机分析处理过程中，需要用到数据仓库、联机分析处理的知识；特别地，在多维表格呈现方面，用到了一个新的开源软件分析工具：Saiku，本章着重对Saiku server的体系结构及原理等进行详细的介绍，对系统设计原则做了概述。

## 2.1 技术架构

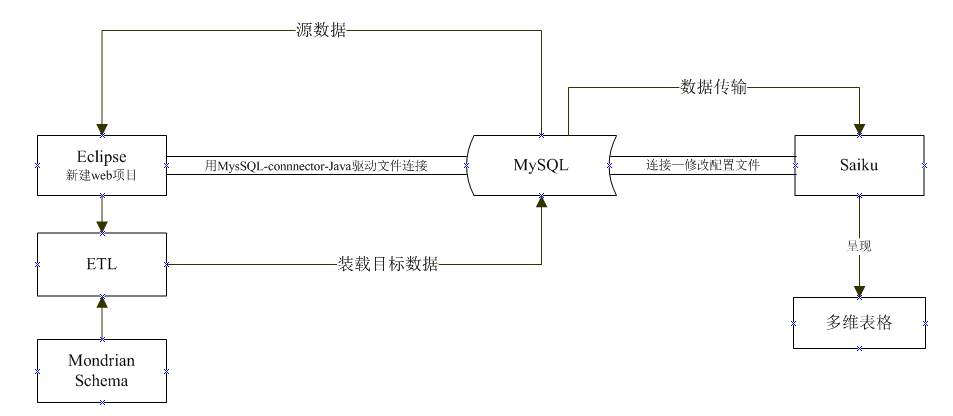
1、本次系统开发采用Java编程语言；

2、开发环境是：Eclipse + Tomcat7.0 + Jdk；

3、数据库系统：Mysql数据库；

4、多维分析：Saiku + Tomcat7.0

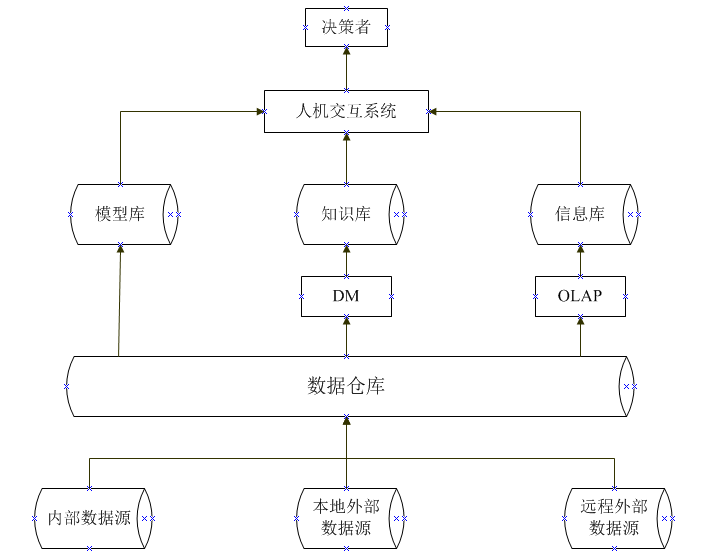
整个系统开发过程中，前期的联机分析处理过程在Eclipse中连接数据库，采用java编程，完成整个ETL；然后在Saiku开源软件进行多维分析。图2-1是系统开发技术架构图：



**图2-1 系统开发技术架构**

## 2.2 数据仓库

数据仓库是决策支持系统和联机分析应用数据源的结构化数据环境。数据仓库研究和解决从数据库中获取信息的问题。数据仓库的特征在于面向主题、集成性、稳定性和时变性[4]。图2-2是数据仓库模型图：



**图2-2 数据仓库模型图**

1、数据仓库是面向主题的；操作型数据库的数据组织面向事务处理任务，而数据仓库中的数据是按照一定的主题域进行组织。主题是指用户使用数据仓库进行决策时所关心的重点方面，一个主题通常与多个操作型信息系统相关。

2、数据仓库是集成的，数据仓库的数据有来自于分散的操作型数据，将所需数据从原来的数据中抽取出来，进行加工与集成，统一与综合之后才能进入数据仓库；数据仓库中的数据是在对原有分散的数据库数据抽取、清理的基础上经过系统加工、汇总和整理得到的，必须消除源数据中的不一致性，以保证数据仓库内的信息是关于整个企业的一致的全局信息。

3、数据仓库是不可更新的，数据仓库主要是为决策分析提供数据，所涉及的操作主要是数据的查询；

4、数据仓库是随时间而变化的，传统的关系数据库系统比较适合处理格式化的数据，能够较好的满足商业商务处理的需求。稳定的数据以只读格式保存，且不随时间改变。

## 2.3 Mondrian

Mondrian是一个OLAP引擎，它用MDX语言实现查询，从关系数据库(RDBMS)中读取数据。然后经过Java API以多维的方式对结果进行展示。Mondrian的使用方式同JDBC驱动类似，可以非常方便的与现有的Web项目集成。

Mondrian OLAP 系统由四个层组成，从最终用户到数据中心, 顺序为:

1、表现层(the presentation layer)

表现层决定了将呈现在用户显示器上的可视化界面，以及如何与系统产生交互。JPivot 是Mondrian的表现层TagLib，jpivot是一个自定义jsp的标签库，它基于XML/XSLT配置来生成相应的html[3]，图2-3为Mondrian多维数据表：



**图2-3 Mondrian多维数据表**

2、维度层(the dimensional layer)

维度层用来解析、验证和执行MDX查询要求。一个MDX查询要通过几个阶段来完成：首先是计算坐标轴，再者计算坐标轴中cell的值。

为了提高效率，维度层把要求查询的单元成批发送到集合层，查询转换器接受操作现有查询的请求，而不是对每个请求都建立一个MDX 声明。

3、集合层(the star layer)

集合层负责维护和创建集合缓存，一个集合是在内存中缓存一组单元值， 这些单元值由一组维的值来确定。

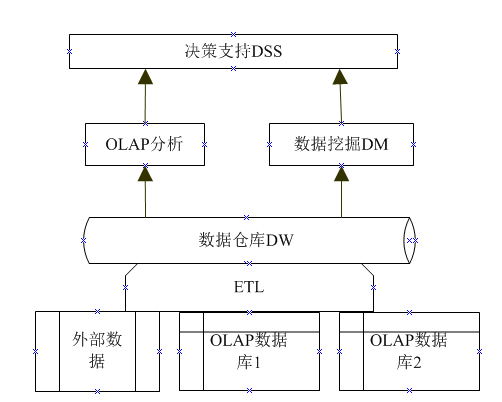
维度层对这些单元发出查询请求，如果所查询的单元值不在缓存中，则集合管理器(aggregation manager)会向存储层发出查询请求。

4、存储层(the storage layer)

存储层是一个关系型数据库(RDBMS)。它负责创建集合的单元数据，和提供维表的成员。

## 2.4 OLAP联机分析处理

联机分析处理（Online Analytical Processing,简称OLAP）是一类软件技术，它使分析人员、经理、管理人员通过对信息的多种可能的观察角度进行快速、一致和交互性的存取以获得对信息的深入理解[5]。OLAP的基本多维分析操作有钻取（有上钻和下钻）、切片和切块、以及旋转等。图2-4是商业智能BI体系结构：



**图2-4 商业智能BI体系结构**

OLAP的显著特征是能提供数据的多维视图概念，数据的多维视图使最终用户能多角度、多侧面、多层次地考察数据库中的数据，从而深入地理解包含在数据中的信息及其内涵。

OLAP的第二个特征是能快速响应用户的分析请求。OLAP的第三个特征是其分析功能。这是指OLAP系统可以提供给用户强大的统计、分析、成本兑换、非过程化建模、多维结构、报表处理功能。OLAP的第四个特征是共享特征。这是指OLAP系统应有很高的安全性。OLAP的第五个特征是它的信息性。这是指OLAP分析所需要的数据及导出的有用信息。

## 2.5 ETL技术

ETL是英文Extract-Transform-Load 的缩写，用来描述将数据从来源端经过抽取（extract）、转换（transform）、加载（load）至目的端的过程。常见于数据仓库开发中将数据由业务系统归集到数据仓库（DW）或者数据集市的过程。在ETL三个部分中，花费时间最长的是“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[6]。

ETL的过程就是数据流动的过程，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。ETL的核心还是在于T这个过程，也就是转换，而抽取和装载一般可以作为转换的输入和输出，或者，它们作为一个单独的部件，其复杂度没有转换部件高。和OLTP系统中不同，那里充满这单条记录的insert、update和select等操作，ETL过程一般都是批量操作，例如它的装载多采用批量装载工具，一般都是DBMS系统自身附带的工具，例如Oracle SQLLoader和DB2的autoloader等[7]。表2-1是不同的ETL工具选择优缺点：

**表2-1 ETL工具选择**

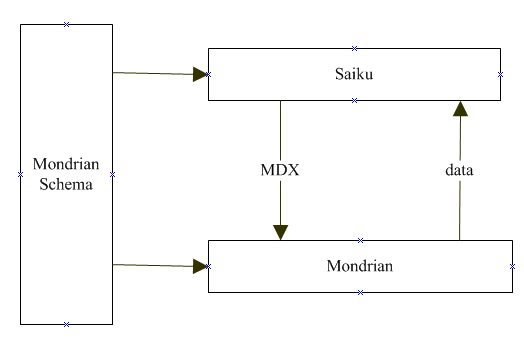
|  |  |  |  |
| --- | --- | --- | --- |
| **工具** | | **优点** | **缺点** |
| 主流工具 | Datastage | 内嵌一种类BASIC语言，可通过批处理程序增加灵活性，可对每个job设定参数并在job内部引用 | 早期版本对流程支持缺乏考虑；图形化界面改动费事 |
| Powercenter | 元数据管理更为开放，存放在关系数据库中，可以很容易被访问 | 没有内嵌类BASIC语言，参数值需人为更新，且不能引用参数名；图形化界面改动费事 |
| Automation | 提供一套ETL框架，利用Teradata数据仓库本身的并行处理能力 | 对数据库依赖性强，选型时需要考虑综合成本（包括数据库等） |
| udis睿智ETL | 适合国内需求，性价比高 | 配置复杂，缺少对元数据的管理 |
| 自主开发 | | 相对于购买主流ETL工具，  成本较低 | 各种语言混杂开发，无架构可言，后期维护难度大。 |

## 2.6 Saiku开源软件

1、Saiku成立于2008年，由Tom Barber和Paul Stoellberger研发。最初叫做Pentaho分析工具，起初是基于OLAP4J库用GWT包装的一个前端分析工具。经过多年的演化、重写之后，于2010年，改名为Saiku，如今已成为一个强大的独立工具，基于Mondrian OLAP引擎，用于查询构建数据集市[9]。

2、Saiku是一个轻量级的OLAP分析引擎，可以方便的扩展、嵌入和配置。Saiku通过REST API连接OLAP系统，利用其友好的界面为用户提供直观的分析数据的方式，它是基于jQuery做的前端界面。它的用户界面完全可以自定义，它的界面是用HTML、CSS和Javascript做的，从而使得它非常容易改变或完全替代。通过使用REST的标准，服务器可以很容易地集成到不同的用户界面和第三方应用程序上，唯一的要求是第三方应用程序可以发送和接受HTTP通信和接收JSON格式的数据。客户端程序不需要了解MDX和相关的查询语言。

3、Saiku的代码是前后端分离的结构，前端是由node使用backbone/express/ace/jquery/protovis等js框架开发。后台restful接口由Java jersy框架开发，身份认证是使用的springframework的security框架，内容管理是使用的开源的Apache Jackrabbit进行存储，使用WebDAV 进行访问管理，全部代码由maven进行管理。protovis 数据可视化图表工具，由于社区不活跃，已由D3.js继承。Apache JackRabbit是一个开放源码的JSR-170 实现，实现了Level 2，但它还有许多扩展的功能。WebDAV （Web Distributed Authoring and Versioning）是一种基于 HTTP 的通信协议，是对 HTTP 的一种扩展，它在 HTTP 协议的方法的基础上添加了一些新的方法，使得用户可以通过客户端编辑和管理存储在远程的文件[16]。图2-5是Saiku框架结构图：



**图2-5 Saiku框架结构**

## 2.7 系统设计原则

### 2.7.1 面向系统用户

内部系统不同于互联网应用，系统有稳定的用户群，不需要挖空心思发展用户，系统主要是为内部生产服务的，如何满足用户需求尤其是最常用业务才是最重要的。信息架构的考虑上，首先我们从不同用户的视角出发，将渠道相关的操作都汇集起来，然后再进行分类，比如可以分为基础信息、地理信息等等，最后根据工作场景串联起来。

把系统看作一个平台。用户需求、用户习惯、信息架构、交互设计、界面设计、系统风格原理上并不复杂，但是用户体验最难的是体验整个过程，这是环环相扣的，所以实施时要注重每一个细节，任何一个细小方面的考虑不足都可能造成很差的用户体验。总得来说，成功的用户体验就是事先知道用户的期望。并且做系统的时候要从每一个细节上做到超出用户预期，给用户带来惊喜。

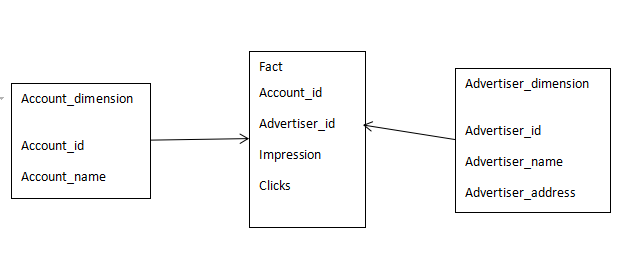
### 2.7.2 多维模型设计

在多维分析的商业智能解决方案中，根据事实表和维度表的关系，又可将常见的模型分为星型模型和雪花型模型。

1、星型架构是一种非正规化的结构，多维数据集的每一个维度都直接与事实表相连接，不存在渐变维度，所以数据有一定的冗余。

星形模型实用的是反规范化数据。在星形模型中，维度直接指的是事实表，业务层级不会通过维度之间的参照完整性来部署。

图2-6是一个简单的星型模型图：



**图2-6 星型模型**

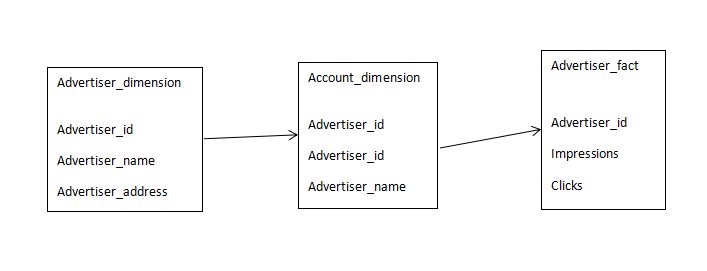
1. 当有一个或多个维表没有直接连接到事实表上，而是通过其他维表连接到事实表上时，其图解就像多个雪花连接在一起，故称雪花模型。

雪花模型是对星型模型的扩展。它对星型模型的维表进一步层次化，原有的各维表可能被扩展为小的事实表，形成一些局部的 " 层次 " 区域，这些被分解的表都连接到主维度表而不是事实表。

它的优点是 : 通过最大限度地减少数据存储量以及联合较小的维表来改善查询性能。雪花型结构去除了数据冗余。

相比星型模型，雪花模型的特点是贴近业务，数据冗余较少，但由于表连接的增加，导致了效率相对星型模型来的要低一些。

图2-7是一个雪花模型图：



**图2-7 雪花模型**

### 2.7.3 系统化思维

多维分析设计的另一个设计原则是系统化思维。维度、事实表、服务程序和环境四大类的接触不是单独服务设计对多维设计的个体影响，而是相互作用、相互依存、具有特定目标和功能的有机整体。

这就要求在设计过程中，不能孤立地考虑设计中的每个要素，必须利用系统化思维确保服务设计的完整与统一，需要对不同的要素进行合理、妥善的规划与组织，打造一个全面、系统的流程，保证整个流程是环环相扣的。这样才使得服务设计中的体验需求是完美的，而不是因为某些触点的缺失、要素的考虑不当、流程的断裂导致服务缺口，破坏整体的设计体系。

## 2.8 本章小结

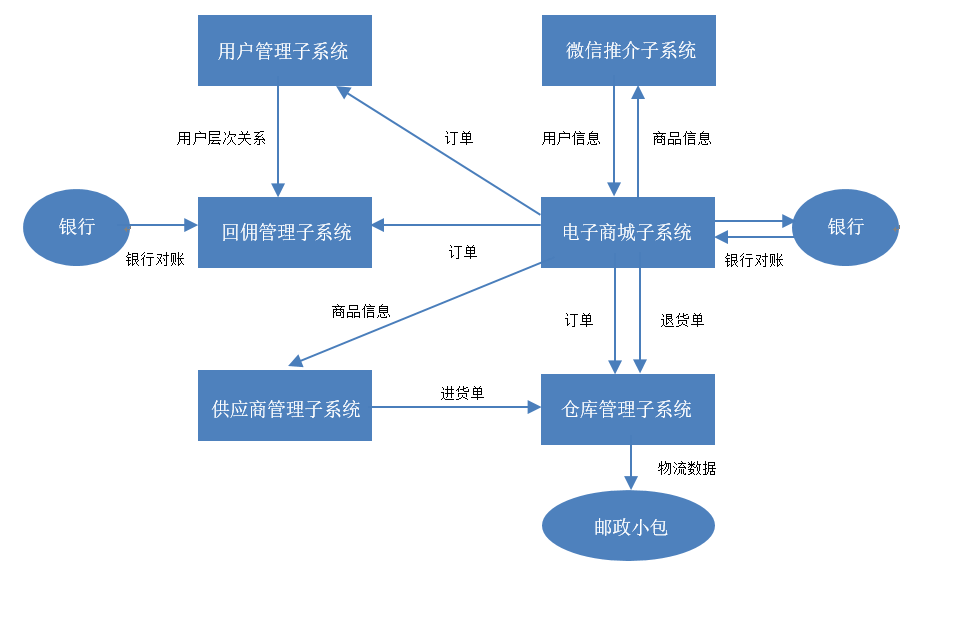
本章先介绍了系统开发的技术架构，以及整个项目中涉及到的相关理论概念，涉及到的理论概念有：数据仓库、Mondrian、OLAP联机分析处理、Mondrian、ETL技术，以及Saiku开源软件；接着介绍了系统设计原则，包括面向系统用户、多维建模原理和论文设计中渗透到的系统化思维。

# 第三章 电子商务联机分析处理过程

## 本章内容主要是电子商务联机分析处理的具体过程，在论文设计的初始，先对联机分析处理的需求进行分析，明确多维建模要解决的问题；然后建立多维立方体模型，建立雪花模型，对事实表和维度表进行了详细介绍；接着对整个设计的核心内容ETL展开介绍，对Saiku多维表格呈现进行分析、测试。

## 3.1 OLAP需求分析

×××销售管理平台包含如下子系统：用户管理子系统、电子商城子系统、微信子系统、仓库管理子系统、供应商管理子系统、回佣管理子系统、后台管理子系统。图3-1是该销售管理平台的业务分析图：



**图3-1 该平台的业务分析图**

该销售管理平台对不同的子系统的活动流程做了详细的分析，但是对后台数据的多维分析没有做足够的研究，本次系统开发旨在对多维数据分析做深入的研究和开发。

具体地，本次OLAP分析的主要工作包括以下四点：

1、对不同种类的商品的销量和价格进行可视化分析。根据商品种类统计出其销量，对其销量和价格之间的分布进行跟踪；

2、对不同级别的机构的销量和价格分布进行统计；

3、对不同年份或者月份的销售量统计分析，将商品的销售情况随时间分布直观展现出来，以便后期做决策时参考；

4、对用户所购买的商品的价格和数量进行统计，呈现出已支付的价格变化情况。

## 3.2 数据仓库模型设计

电子商务企业运营的最终目标是提高销售利润，掌握销售情况是企业决策者及时准确的进行决策的一个重要方面。能够随时根据订单的历史销售情况进行分析，可以更好地进行下一步的经营决策规划。要建立的多维立方体模型是星型模型，包括事实表和维度表，事实表是订单表，维度表包括商品表、机构表、用户表和时间表，其中商品表和品牌表关联。

订单主题中，订单事务数据是用以分析最为理想的数据，基于本项目的开发背景，一般情况下决策者需要通过时间、用户、商品、机构四个模型进行分析，其中商品表和品牌表通过品牌编号相关联。

如下是订单主题雪花模型结构图：

1、订单主题

决策者查看订单情况主要用于分析销售情况，以便于采取相应的措施。因此考虑的内容有相应商品的销量，以及订单总金额，且商品数量具有良好的可加性，决策者需要经常查看，在事实表中存放订单总金额数据可以减少数据仓库工作时的工作量，还能够保证所有用户在使用这一重要数据时的一致性。

2、时间维

时间维是许多数据仓库应用中的常用维度，它的设计方式与其他模型有差别，是多维模型中比较特殊的一个维度。本项目中的时间维是指订单的下单时间，来源于源数据库order\_info表中的add\_time字段。源数据库中该字段是时间戳类型，在做ETL时候，将时间戳类型转换成String型。

对于本系统而言，电子商务销售系统随时都有可能发生，按不同的时间对订单数据正在不同的级别上进行统计汇总，也有着不同的作用。要对各种数据在年度、季度、月度、日等的层次上进行分析，就需要将时间维的粒度设计为时间编号、年(year)、月（month）、日（day）、时（hour）、分（minute）、秒（second）。这里，时间维的粒度级别全部表示的是数据涉及的所有时间，即最长的时间跨度。

3、用户维

用户维度可以包括用户编号、用户名、用户类型，以及机构编号。其中，机构编号的字段是不能省略的，由于不同的用户隶属于不同的机构，根据机构编号，可以很容易的找到所属机构的具体名称，从而可以方便地进行多维分析。

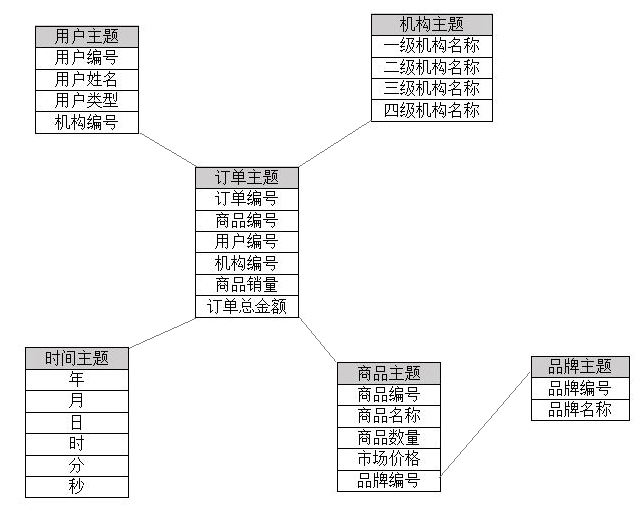
4、商品维

商品维度包括商品编号、商品名称、商品数量、商品的市场价格以及品牌编号，通过品牌编号，商品表和品牌表之间关联。品牌表中存储的字段有：品牌编号、品牌名称，通过品牌编号，商品表和品牌表关联起来了。

5、机构维

机构维度里面的字段有机构编号、一级机构地址名称、二级机构地址名称、三级机构地址名称、四级机构地址名称。机构维度要和用户维度关联起来，某一个机构中会有多个用户，在多维分析时会涉及到，因此，在建模的时候需要设立好字段。

图3-2是本次系统设计中以订单为事实表的雪花模型图：



**图3-2 订单主题雪花模型图**

## 3.3 数据库表设计

要建立的多维立方体模型是星型模型，包括事实表和维度表，事实表是订单表，维度表包括商品表、机构表、用户表和时间表，其中商品表和品牌表关联。

1、表3-1是订单表的字段：

**表3-1 订单表字段**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **类型** | **长度** | **小数点** | **字段含义** |
| Order\_id | mediumint | 8 | 0 | 订单表主键 |
| User\_id | mediumint | 8 | 0 | 用户id |
| Goods\_id | mediumint | 8 | 0 | 商品id |
| Time\_id | mediumint | 8 | 0 | 时间id |
| Institution\_id | mediumint | 8 | 0 | 机构id |
| Goods\_number | smallint | 5 | 0 | 商品数量 |
| Order\_amount | decimal | 10 | 2 | 订单总额 |

订单表的字段包括订单编号、用户编号、商品编号、时间编号、机构编号以及商品数量和订单销售金额，订单表作为整个雪花模型中的事实表，和用户表、商品表、时间表以及机构表关联，订单表中有两个量度字段：商品数量和订单总金额，在Saiku呈现多维分析表格的时候会体现。便于决策者分析数据，做出下一步的决策和安排。

2. 表3-2是商品表字段：

**表3-2 商品表字段**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **类型** | **长度** | **小数点** | **字段含义** |
| Goods\_id | mediumint | 8 | 0 | 商品id |
| Goods\_name | varchar | 120 | 0 | 商品名称 |
| Goods\_number | int | 11 | 0 | 商品数量 |
| Market\_price | decimal | 10 | 2 | 市场价格 |
| Brand\_id | smallint | 5 | 0 | 品牌id |

商品表的字段有商品编号、商品名称、商品数量、商品的市场价格以及品牌编号，存放品牌编号的目的是为了让商品表和品牌表相互关联，便于分析某一商品属于何种品牌，进而了解到该种商品所属品牌的具体信息，在分析决策的时候品牌也会发挥相当大的作用。

3. 表3-3是机构表的字段详情：

**表3-3 机构表字段**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **类型** | **长度** | **小数点** | **字段含义** |
| Institution\_id | mediumint | 8 | 0 | 机构id |
| first | varchar | 150 | 0 | 一级地址 |
| second | varchar | 150 | 0 | 二级地址 |
| third | varchar | 150 | 0 | 三级地址 |
| fourth | varchar | 150 | 0 | 四级地址 |

机构表的字段包括机构编号、一级机构地址、二级机构地址、三级机构地址、四级机构地址，存放四个不同级别的机构名称。机构表要和用户表关联起来，因为在进行多维分析的时候，同一机构下有多个用户，因此将机构编号存放在用户表中，有利于Saiku多维分析表的呈现。

4. 表3-4是用户表的字段：

**表3-4 用户表字段**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **类型** | **长度** | **小数点** | **字段含义** |
| User\_id | mediumint | 8 | 0 | 用户id |
| User\_name | varchar | 60 | 0 | 用户姓名 |
| User\_type | tinyint | 3 | 0 | 用户类型0/1 |
| Institution\_id | int | 11 | 0 | 机构id |

用户表中的字段包括用户编号、用户名、用户类型和机构编号。用户类型的值为0或者1。在本项目的整个设计过程中，不同的用户隶属于不同的机构，根据机构编号，可以很容易的找到所属机构的具体名称，从而可以方便地进行多维分析。

5. 表3-5是时间表的字段：

**表3-5 时间表字段**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **类型** | **长度** | **小数点** | **字段含义** |
| Time\_id | int | 11 | 0 | 时间id |
| year | varchar | 255 | 0 | 年 |
| month | varchar | 255 | 0 | 月 |
| day | varchar | 255 | 0 | 日 |
| Hour | varchar | 255 | 0 | 时 |
| minute | varchar | 255 | 0 | 分 |
| second | varchar | 255 | 0 | 秒 |

时间表中的字段包括时间编号、年、月、日、时、分、秒，在源数据库这噢乖有order\_info表，其中有add\_time字段，是用户下订单的时间，在转化的过程中将时间戳类型转换成时间类型，以String类型存放在目标数据库的time表中。

6. 表3-6是品牌表的字段：

**表3-6 品牌表字段**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **字段名** | **类型** | **长度** | **小数点** | **字段含义** |
| Brand\_id | smallint | 5 | 0 | 品牌id |
| Brand\_name | varchar | 60 | 0 | 品牌名称 |

品牌表的字段包括品牌编号和品牌名称，品牌表的数据可以直接从源数据库中抽取，最后将转换后的数据加载到新建的数据库中。

## 3.4 Mondrian Schema设计

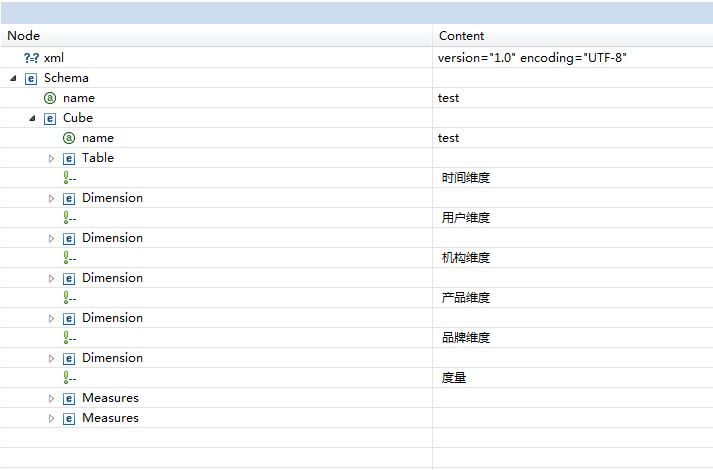
Mondrian是开源项目Pentaho的一部分，是一个用Java写成的OLAP引擎。它实现了MDX语言、XML解析、JOLAP规范。它从SQL和其它数据源读取数据并把数据聚集在内存缓存中，然后经过Java API用多维的方式对结果进行展示，同时可以不写SQL就能分析存储于SQL 数据库的庞大数据集，可以封装JDBC数据源并把数据以多维的方式展现出来。

Mondrian分成了四个大部分Schema manager、Session Manager、Dimension Manager、Aggregate Manager.

在Schema模型中，整个Schema只有一个Cube，有五个Dimension，分别是：time, user, institution, goods, brand，和两个量度：amount和price，通过Schema可以写出MDX查询语句。

一个Cube就是measures和dimensions的集合，measures和dimensions的共同之处就是都是来自于事实表。

图3-3是本次系统设计的Cube框架图：



**图3-3 本次系统开发的Cube框架**

Dimension是一个层次（Hierarchies）的集合，维度一般有其相对应的维度表。他的组成是由层次（Hierarchies），而层次（Hierarchies）又是有级别（ Level ）组成。表3-7是Dimension的属性详情：

**表3-7 Dimension属性**

|  |  |
| --- | --- |
| **属性名** | **含义** |
| name | Dimension 的名称 |
| type | 类型，有两个可选的类型： StandarDimension 和 TimeDimension ，默认为 StandardDimension |
| caption | 标题 , 在表示层显示的 |
| UsagePrefix | 加前缀 , 消除歧义 |
| foreignKey | 外键，对应事实表中的一个列，它通过 <Hierarchy> 元素中的主键属性(primaryKey)连接起来。 |

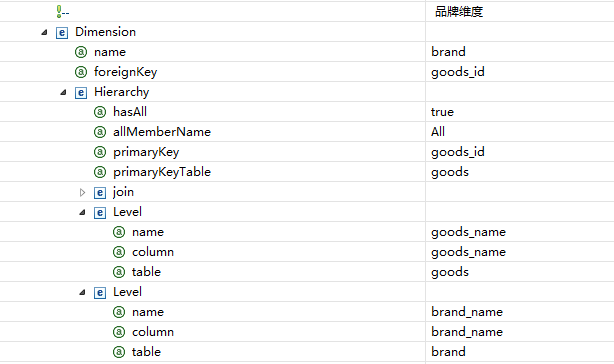
在本系统设计中，多维模型设计中有五个Dimension，分别是：time, user, institution, goods, brand。

如下图3-4是时间维度：



**图3-4 Time维度**

级别（Level）是组成Hierarchy的部分，级别的属性很多，并且是 schema 编写的关键，使用它可以构成一个结构树，Level的先后顺序决定了Level在这棵树上的的位置，最顶层的 Level 位于树的第一级，依次类推。从图4-7可以看出，这里有两个Level，name是当前级别的名称，Column是用上面指定的表中某一列作为该Level的关键字，图3-5是品牌表中的不同level：



**图3-5 品牌表的Level详情**

此外，Level还有其他很多属性，比如levelType，该 Level 的类型，默认为regular（正常的），如果你在其Dimension属性type里选择了TimeDimension那么就可以选择TimeYears、TimeMonth等。

每一个measure包含一个name,column(在事实表中)，aggregator。aggregator通常是”sum”，但是”count”,”min”,”max”,”avg”,”distinct-count”也是支持的。表3-8是本次系统设计中用到的两个度量：

**表3-8 Cube中的两个度量**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **name** | **column** | **aggregator** | **datatype** | **formatString** |
| amount | goods\_number | sum | Numeric | —— |
| price | order\_amount | sum | —— | #,##0,00 |

“datatype”属性指定了cell value如何在Mondrian的cache中展现，它有”String”,”Numberic”,”Integer”,”Boolean”,”Date”,”Time”,”TimeStamp”,默认是”Numeric”，“formatString”:指定了值是如何展现的，”,” 和”.”符号可以作为分隔。Measure可以不仅来自于列，可以使用cell reader,或者是SQL语句，图3-6为Measure框架图：



**图3-6 Measure框架**

## 3.5 ETL

在系统设计过程中，ETL包括三部分：抽取、转换和加载，首先要从源数据库中抽取出所需要转换的数据和字段，然后把需要转换的数据转换成便于后续多维分析的字段类型，最后把处理过的数据加载到目标数据库中。

1、数据源确认

首先确认是否拥有源系统提供需要的数据；然后，从源系统中建立每一个数据元素对应的正确的数据源；进行论证来保证确认的数据源是真正需要的。

这里需要特别说明订单表、时间表和机构表的数据来源，而用户表、品牌表和商品表这三个表的数据来源相对单一，其字段均直接来源于源数据库中对应表的字段。在本次系统开发过程中，源数据库命名为wn，目标数据库命名为zxh.

表3-9是订单表的数据来源：

**表3-9 订单表的数据来源**

|  |  |  |  |
| --- | --- | --- | --- |
| **字段名** | **数据库来源** | **所来源的表** | **对应的字段** |
| order\_id | 源数据库 | order\_info | order\_id |
| time\_id | 源数据库（间接）  目标数据库（直接） | wn.order\_info  zxh.time | time.time\_id |
| institution\_id | 源数据库（间接）  目标数据库（直接） | wn.users  zxh.institution | institution.  institution\_id |
| user\_id | 源数据库 | order\_info | user\_id |
| goods\_id | 源数据库 | order\_goods | goods\_id |
| order\_amount | 源数据库 | order\_info | order\_amount |
| goods\_number | 源数据库 | order\_goods | goods\_number |

在订单表的字段中，time\_id和institution\_id两个字段并不是直接来源于源数据库中的表，这两个字段需要利用到源数据库中的表的相关字段，利用其相关字段转换而来，最后得到time\_id和institution\_id，然后加载到订单表中。 具体地，在时间表中，字段有time\_id, year, month, day, hour, minute, second. 而通过分析数据库的表得知，在源数据的order\_info表中，下单时间add\_time字段是用时间戳类型来存储的，因此，通过把add\_time字段的数据抽取出来，转换成时间类型，从而得到time\_id，将time\_id加载到订单表中。

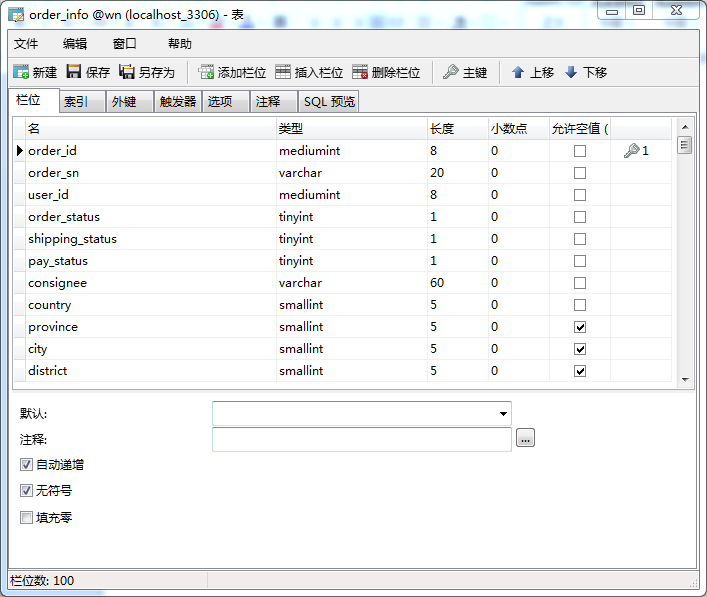
同样地，在源数据库中，机构表是每一级地址都分开存储，这样非常不方便查找某一地区的销售量，因此，也是先从源数据库中抽取转换，最后将新的institution\_id加载到订单表中。

2、批量抽取数据

为了保证数据抽取时数据的准确性、完整性和唯一性，同时降低抽取作业对源数据库造成的压力，抽取作业的加载必须避开源数据的生成时间。批量抽取数据的方法一般用于实时性要求不高的数据，比如T+1或者每月1日进行抽取。

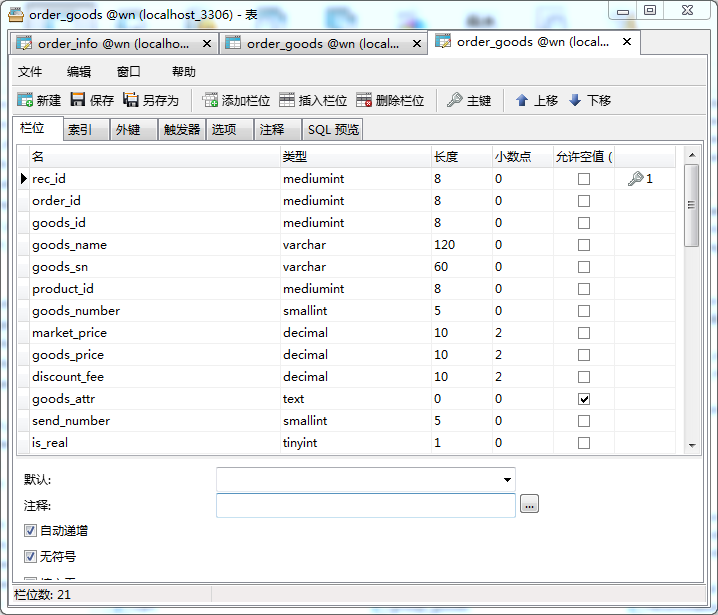
具体地，在本次系统设计中有6个目标数据库表：order, user, time, goods, institution和brand表，如下分别介绍其抽取过程：

对于order表，order\_id,user\_id,order\_amount,FROM\_UNIXTIME(add\_time)来源于order\_info表，goods\_id和goods\_number来自order\_goods表，order\_info表和order\_goods表有共同的字段order\_id可以连接，将两个表连接起来，就可抽取到上面的字段。图3-7是order\_info表的设计：



**图3-7 order\_info表设计**

下面图3-8是order\_goods表的设计：



**图3-8 order\_goods表设计**

对于time表，我们只需要抽取源数据库order\_info表中的add\_time，后面的转换工作在后面详述。而对于institution表，采取的是边抽取边转换。先从源数据库的users表中取出institution\_id，然后在institution\_level表中查找上一级地址，进行再抽取。对于user表、brand表和goods表，只需分别到源数据库中的users表、brand表和goods表中找到对应字段抽取。

3、数据转换

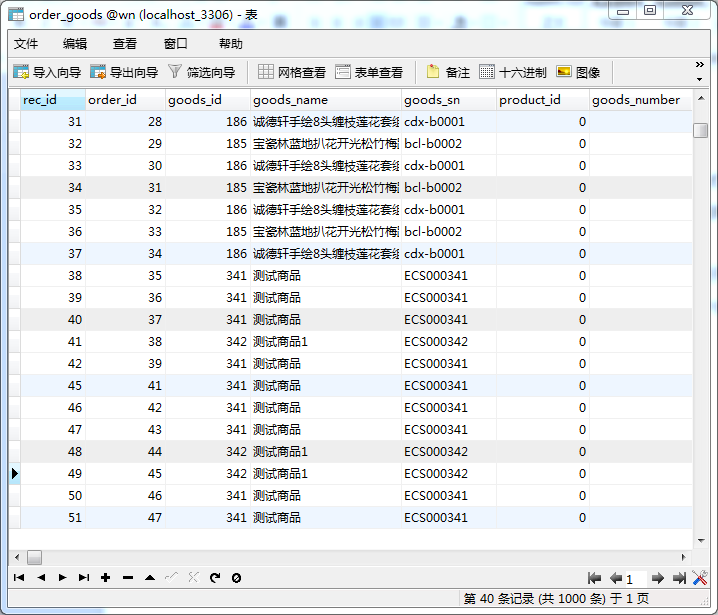
这一步包含了数据的清洗和转换。因为数据源的数据缺乏质量保障，采集数据时总会发生很多人为或者客观因素导致的问题。因此，清洗数据非常有必要。

针对本次系统设计的数据，清洗的重点主要体现在一下两方面：

1. 数据格式错误。

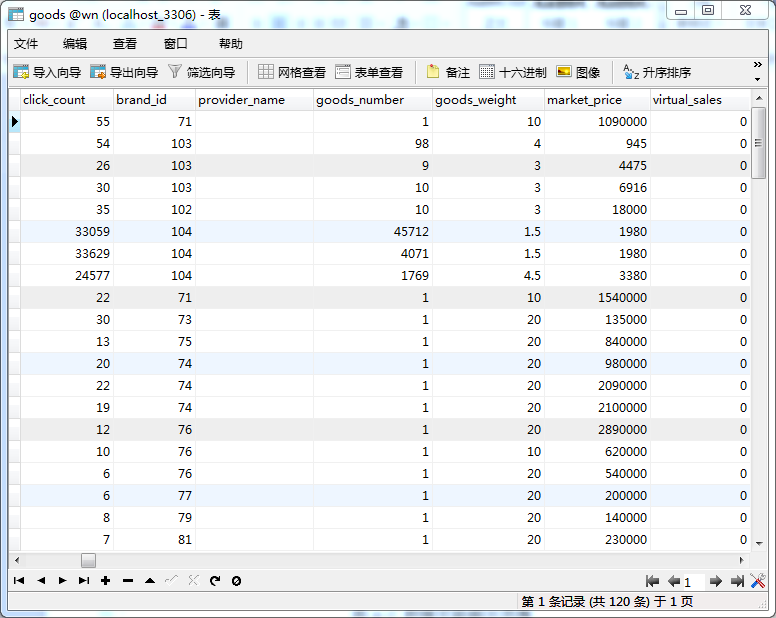
例如缺失数据、数据值超出范围或是数据格式非法等。要知道对于同样处理大数据量的数据源系统，他们通常会舍弃一些数据库自身的检查机制。

比如在order\_goods表中，goods\_name指的是商品名称，它必须是存在于数据库中的商品。但是，由于数据源可能已经被使用并修改过，数据源中会有无用信息，如下图3-9可能会出现无效数据：



**图3-9 无效数据**

还有一些数据可能会超出范围，比如，在goods表中，有一个字段是goods\_number，这个字段的意义是商品数量，所以取值必然是非负数。但是，采集数据的时候由于人为记录，无法保证如此海量的数据完全不存在失误，如下，图3-10可能出现数据超出范围：



**图3-10 数据可能超出范围**

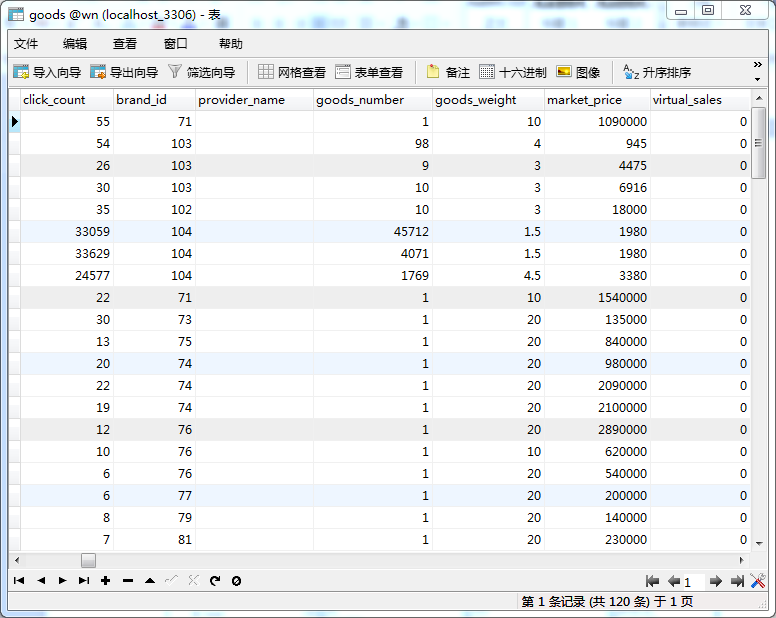
针对这些情况，首先要分析字段，根据字段特征判断是否是无效数据，如果是无效数据，在转换的时候不计入这部分数据。如果是数据缺失，那么，需要认为地将数据补充完整，但是绝不允许删除数据库的数据。

1. 数据一致性。

数据源系统为了性能的考虑，会在一定程度上舍弃外键约束，这通常会导致数据不一致。

在源数据库中，商品表和品牌表之间通过字段brand\_id关联，但是我们在抽取的过程中，可能会抽取到品牌不存在的商品，也就是说，在品牌表中查找某个商品数据的brand\_id对应的品牌名，发现brand表中不存在该brand\_id，那么，这就造成数据不一致性。

下图3-11是源数据库中的部分商品表的数据：

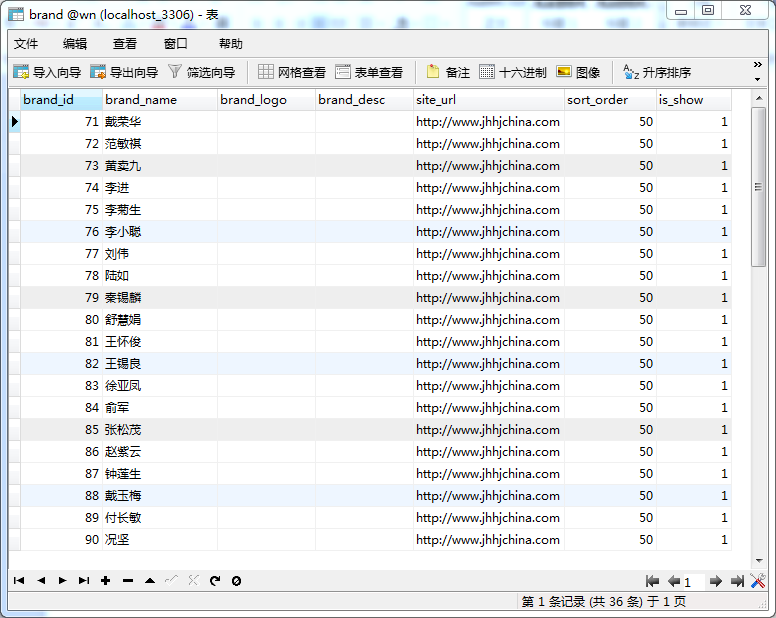


**图3-11 源数据库中的商品表数据**

假如现在要分析在目前库存中的商品品牌分类，则我们需要根据商品表中的brand\_id在品牌表中查找对应的brand\_name。

但是有个问题是：我们可能无法在品牌表中查到对应的品牌名，也就是说，当前查找的商品的品牌编号无法在品牌表（brand）中找到对应的brand\_id，这样的话，该商品的品牌名就无从查找，当我们分析商品销量和品牌之间的关系的时候就会出现数据不完整。

图3-12是品牌表的部分数据：

**图3-12品牌表对应数据**

数据清洗完毕之后，接下来开始数据转换。数据转换包含简单的数据不一致转换，数据粒度转换和耗时的数据关联整合或拆分动作。

就本系统设计而言，数据转换包括三方面：

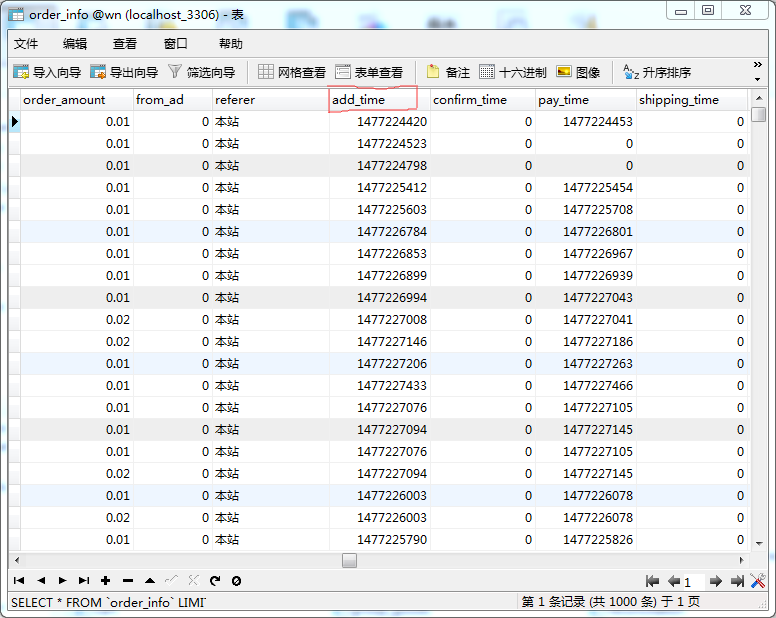
1. 时间戳转换

根据前期的星型模型，作为维度表之一的时间表，包括的字段有年、月、日、时、分、秒，这里的时间是指用户下单的时间，在源数据库中的数据存储在order\_info表的add\_time字段中。

使用FROM\_UNIXTIME(unix\_timestamp,format)函数把时间戳转换成时间类型，即FROM\_UNIXTIME(add\_time)，使用substring函数将时间类型转换成年月日时分秒，如：

String year= add\_time.substring(0,4);

图3-13是源数据库中的order\_info表中的时间戳类型：



**图3-13 源数据库表中的时间戳**

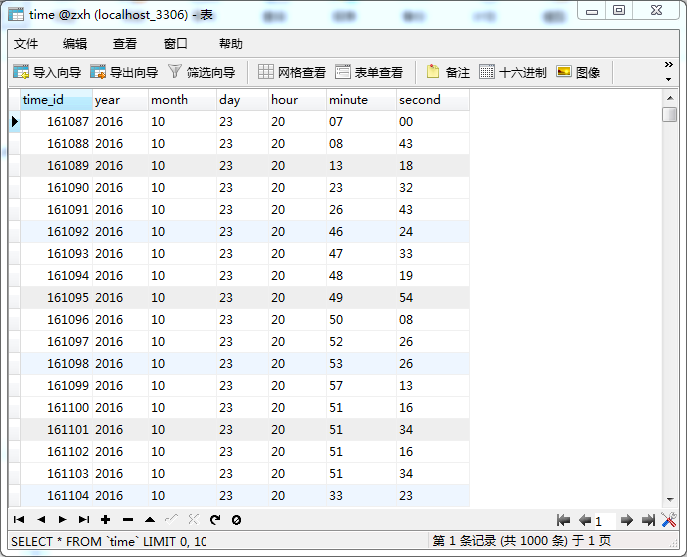
转化为时间后，装载到目的数据库的time表中，在装载之前，先清空原有表再把数据装载进去。

Statement是一个连接数据库的接口，Statement 接口提供了三种执行 SQL 语句的方法：executeQuery、executeUpdate 和execute。

方法 executeQuery 用于产生单个结果集的语句，例如SELECT语句，方法 executeUpdate 用于执行 INSERT、UPDATE 或 DELETE 语句，方法 execute用于执行返回多个结果集、多个更新计数或二者组合的语句，这个功能在实际中较少用到。

先实例化Statement，然后使用executeUpdate函数执行sql语句："delete from time"。

图3-14是转换后的时间表数据：



**图3-14 转化后的时间表数据**

（2）机构表转换

在源数据库中，机构表是分级向上查找的，这样的弊端是无法直接就获知用户的具体机构，而且销售量很大时单独查找，非常消耗时间。因此，在转换机构数据时，要把四级机构放在同一条机构信息表中。

表3-10是在源数据库的institution\_level表中用到的字段：

**表3-10 institution\_level机构**

|  |  |
| --- | --- |
| **字段名** | **含义** |
| id | 和users表中的institution\_id相对应 |
| name | 当前level的机构名 |
| level | Level的值为1、2、3、4，指当前机构是第几级机构 |
| Up\_level | 是指当前机构的上一级机构的id号 |

在转化机构表的时候，首先要从用户表入手，遍历users表，每遍历一个元组，根据当前元组的institution\_id到institution\_level表中查找它的向上机构，直到找到一级机构为止。

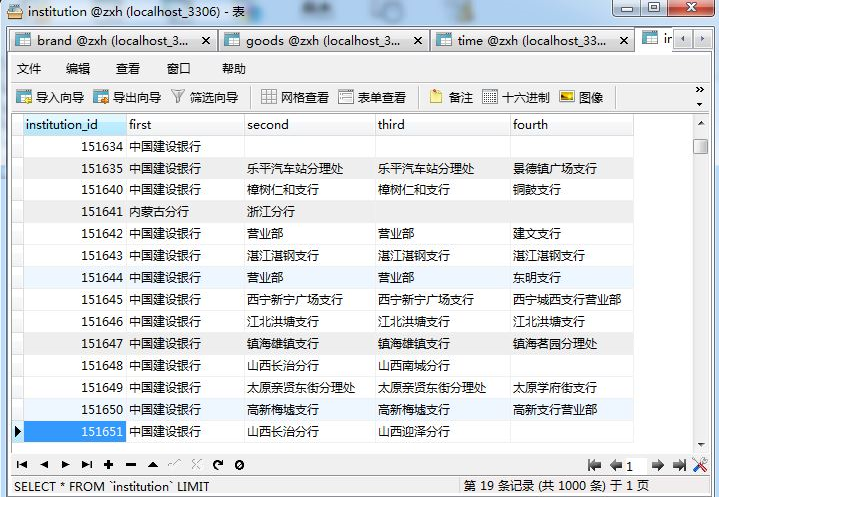
每转换一组地址，将这个元组存入目标数据库的institution表中，并且产生新的自增的institution\_id，注意，这个institution\_id和源数据库表中的不一样。

在将新的地址存入institution表中之前，需要先判断当前地址是否已经在institution表中存在；如果已经存在的话，继续判断下一条，无需将此地址存入数据库；反之如果不存在，执行sql语句：

select institution\_id from institution where first='"+first+"' and second='"+second+"' and third='"+third+"' and fourth='"+fourth+"'";

将返回的institution\_id存入目标数据库中的user表和order表。

图3-15是加载后的目标数据库institution表：



**图3-15 加载数据后的institution表**

（3）订单表转换

订单表作为整个模型的事实表，字段包括维度表主键和度量字段，其中，time\_id和institution\_id随时间表和机构表转化而存入数据库。

同样地，每次给order表中存入一个time\_id时，要先判断当前时间是否已存在于time表中，执行下面的sql语句：

select year,month,day,hour,minute,second from time

如果当前时间已经存在于目的数据库的time表中，则无需查找该地址的id；反之，如果不存在，则查找当前时间对应的time\_id，然后执行下面的数据库语句：

"select time\_id from time where year="+year+" and month="+month+" and day="+day+" and hour="+hour+" and minute="+minute+" and second="+second;

将返回的time\_id存入order表和user表。

Order表的字段分布在两个表中：order\_info和order\_goods，两个表都有共同字段order\_id，转换order\_info表中的order\_id,user\_id,order\_amount字段，转换order\_goods表中的goods\_number字段。

至此，order表的转化工作完成。

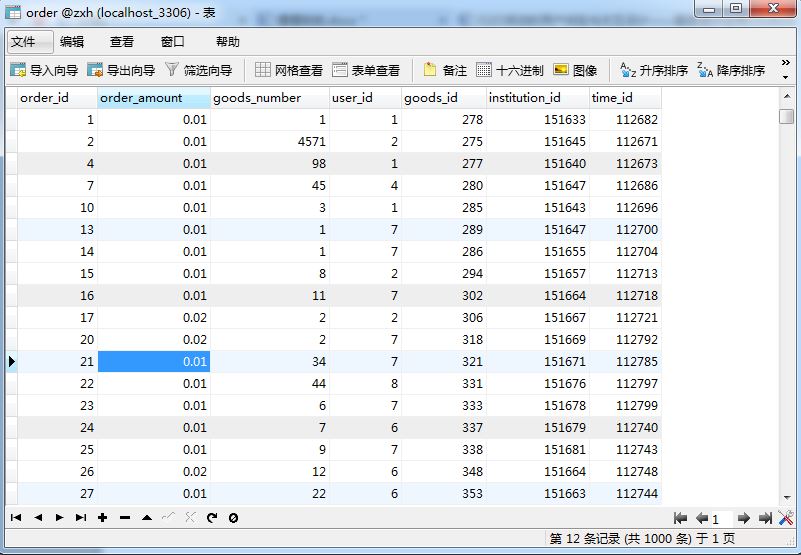
此外，在目的数据库系统中，goods表的外键brand\_id是品牌表brand的主键，关联时候使用外键brand\_id关联。

3、数据加载

转换之后生成的数据可直接插入目标数据库，ETL作业位于目标数据库上，加载作业只需要使用Insert方式导入目标表即可。事实上，转换和加载其实是在同一加工中完成的。

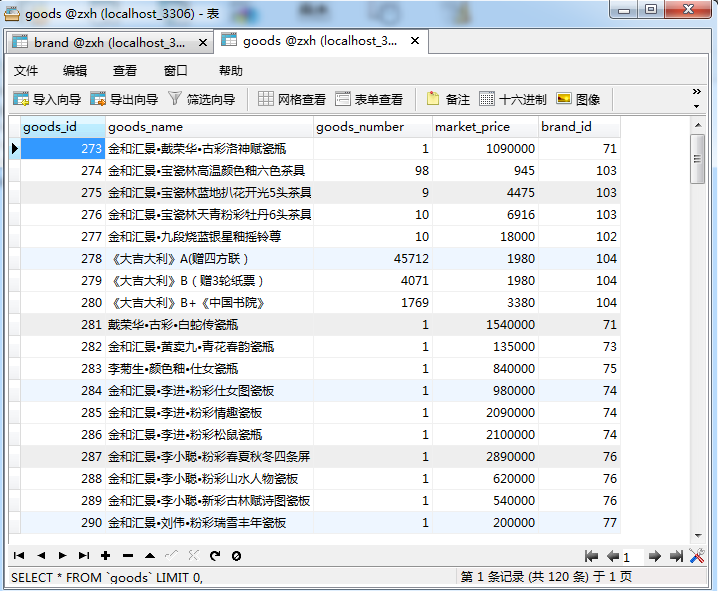
订单表作为事实表，字段包括各维度表中的主键，包括用户编号、时间编号、机构编号商品编号，订单编号以及商品销量和订单总金额。

下图3-16是装载数据后的订单表：



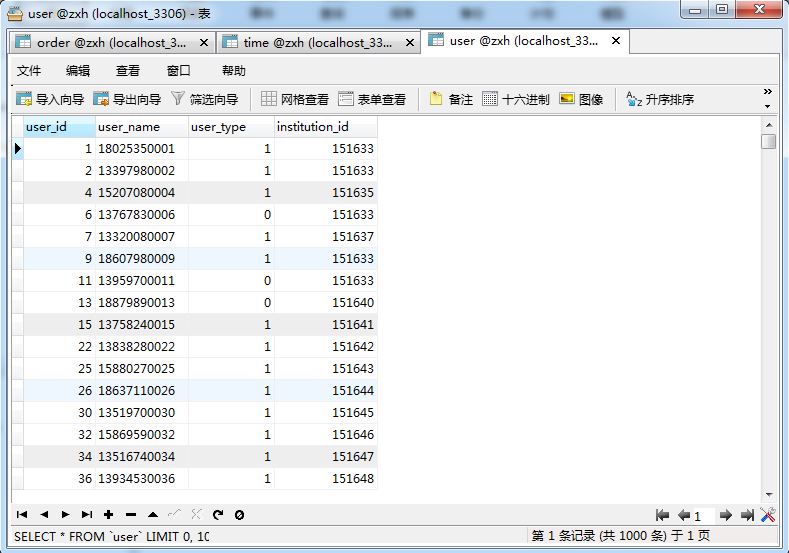
**图 3-16 订单表数据**

商品表中的字段包括商品编号、商品名称、商品数量、市场价格以及品牌编号，品牌编号用来关联品牌表和商品表。图3-17是加载数据后的商品表：



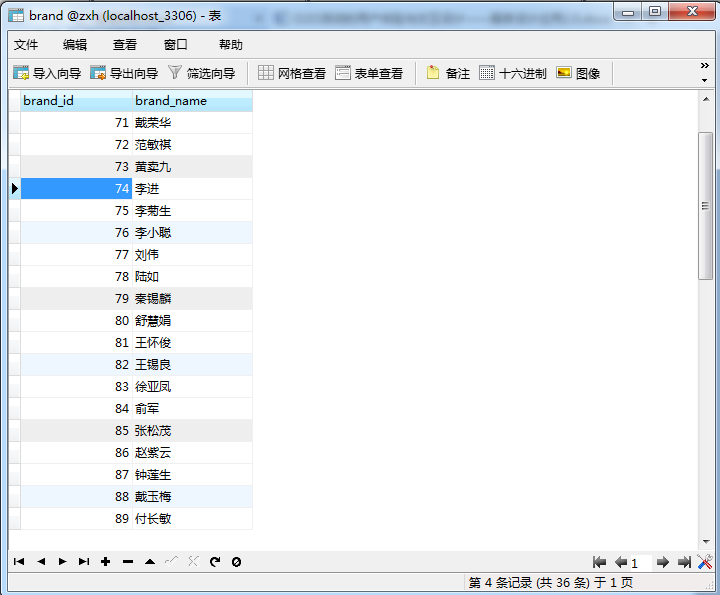
**图3-17 商品表数据**

在用户表中，有机构编号属性，机构编号来自目标数据库中的机构表。因此，必须首先将机构表中的地址抽取并转换。图3-18是用户表的装载数据：



**图3-18 用户表数据**

下图3-19是装载数据后的品牌表：



**图3-19 品牌表数据**

品牌表里面存放两个字段：品牌编号和品牌名，其中，brand\_id的字段类型是smallint型，brand\_name的字段类型是60位长的字符串类型。

## 3.6 本章小结

本章的主要内容是有关项目在整个电子商务联机分析处理的具体过程。首先，搭建OLAP开发环境，安装并配置好Java, tomcat7和jdk，然后安装配置Mysql数据库。第二步是建立多维立方体模型，先进行数据分析，然后进行数据仓库模型设计，对数据进行抽取转换和清洗，最后将相关的数据加载到目标数据库。

# 第四章 Saiku多维分析及测试

完成ETL之后，系统设计所需要的数据已被加载到目标数据库中，接下来，需要把tomcat部署到Saiku server，并将Saiku连接到Mysql数据库，进行多维数据分析。

## 4.1 Saiku连接数据库

在把Saiku连接到MySQL数据库之前，先部署Saiku工程到Tomcat。然后，添加Cube和数据库连接，每个cube对应一个配置文件，同时配置说明文件，这样才能在Saiku中显示。说明配置的属性中：

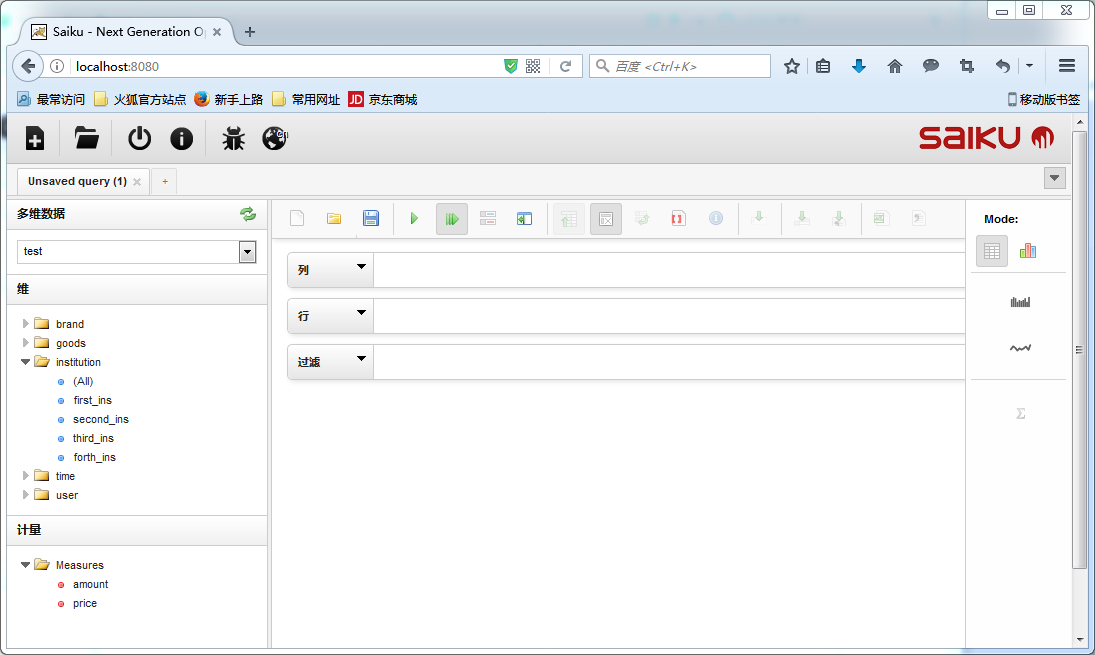
type=OLAP指定一个OLAP引擎；

name : 给当前数据源命名；

driver: 指定 Mondrian driver；

JdbcDrivers为：com.mysql.jdbc.Driver，指明java 类文件作为数据库连接驱动[16]。

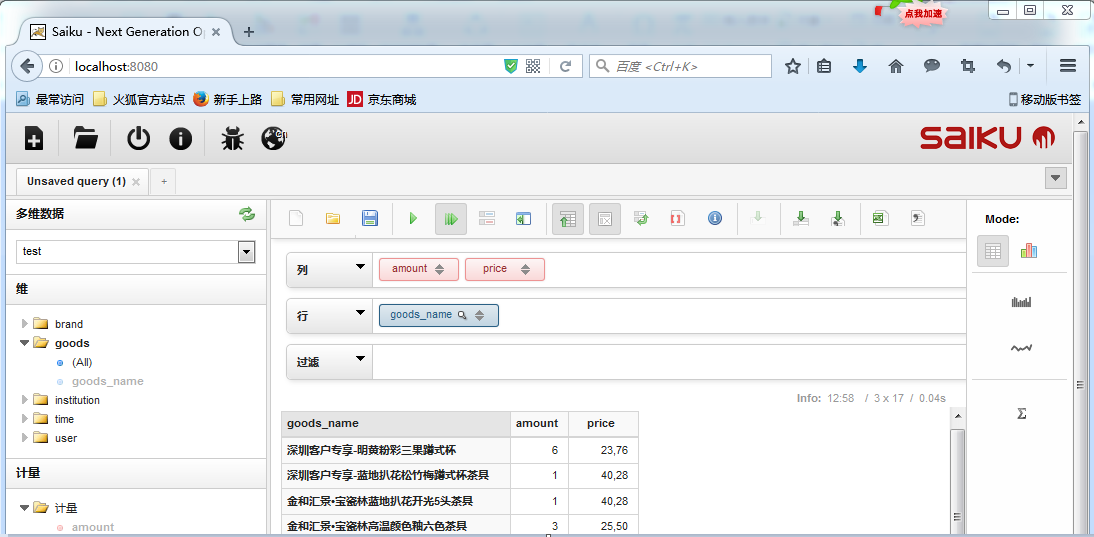
配置完成后，输入http://localhost:8080/访问系统。用户名：admin，密码：admin，登录之后选择多维数据集test，如下图4-1启动Saiku的界面：



**图4-1 启动Saiku界面**

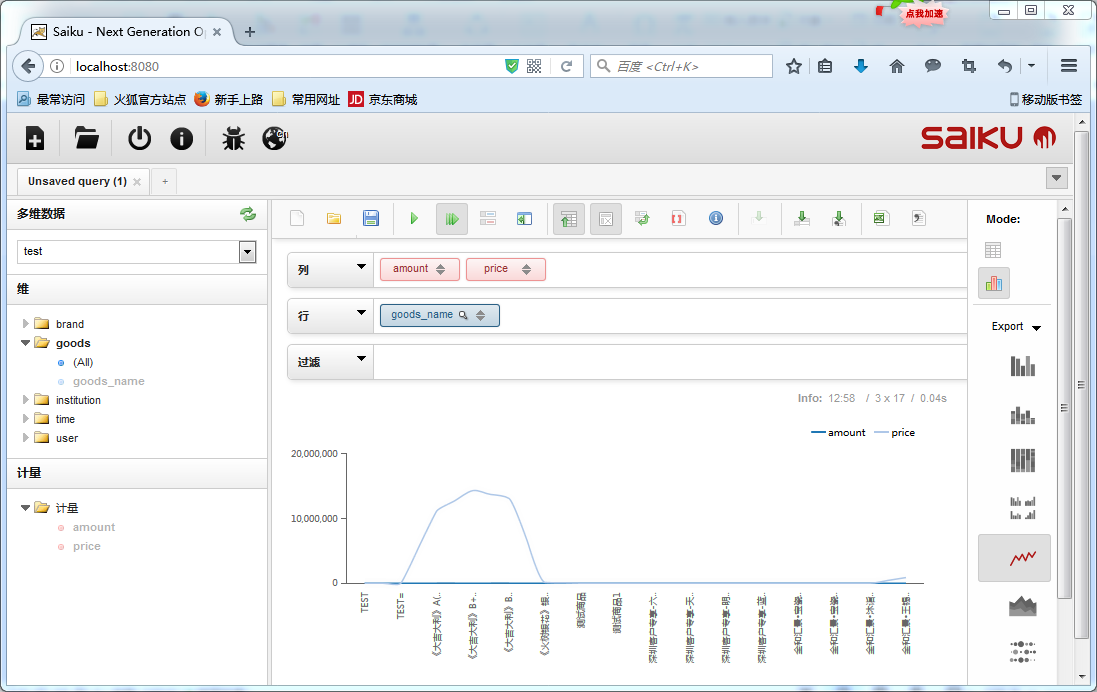
## 4.2 Saiku维表分析

1、列选取amount和price，行选取goods\_name，下图4-2是商品销售的数量个价格统计表：



**图4-2 商品销售及价格统计**

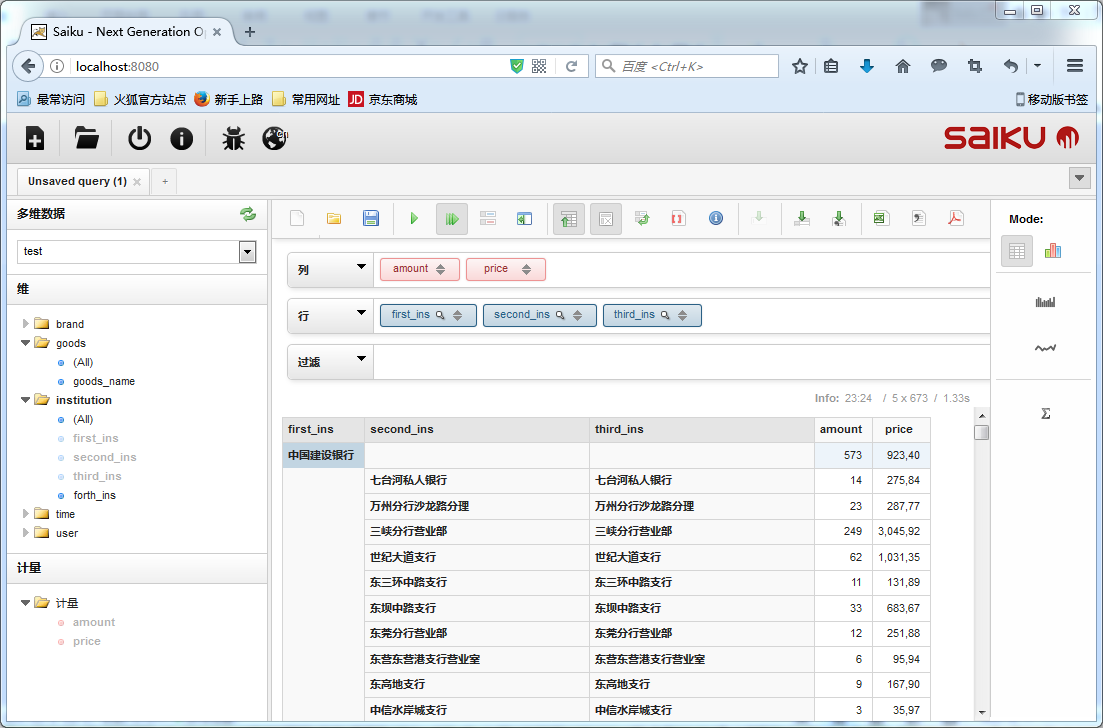
图4-3是不同商品销售变化趋势图：



**图4-3 不同商品销售变化趋势**

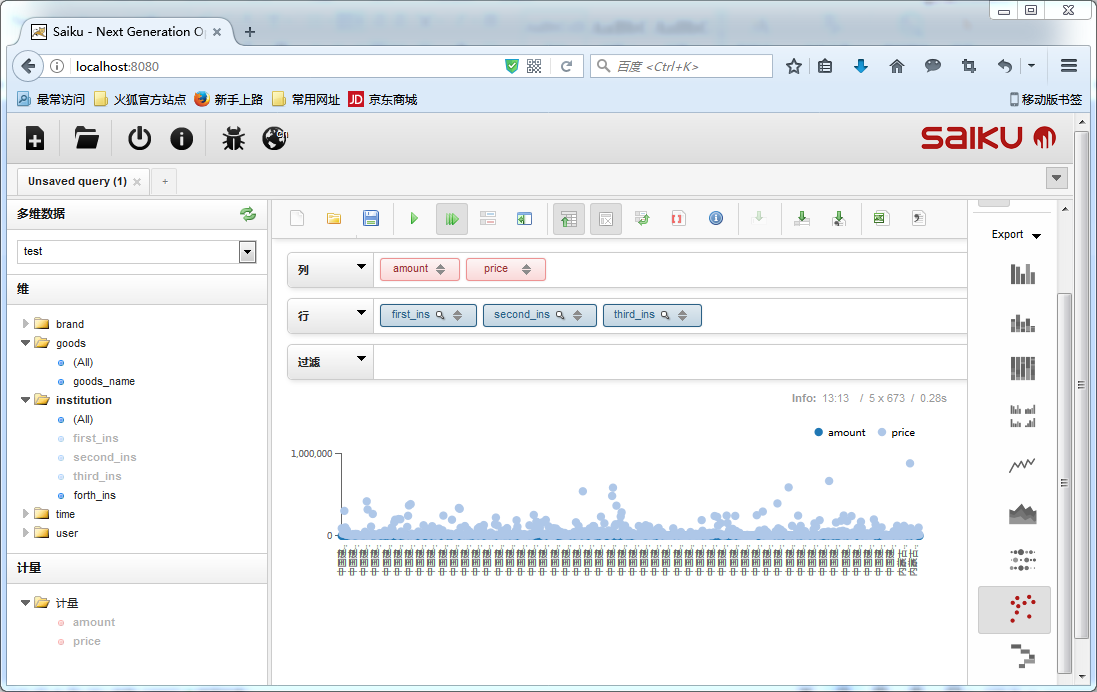
根据图表可以看出，《大吉大利》A(赠四方联）、《大吉大利》B（赠3轮纸票）和《大吉大利》B+《中国书院》这三种销售量是最高的，其他种类的商品销量基本持平，其销量普遍较低。

2、列选取amount和price，行选取一级、二级、三级机构，图4-4是三级机构范围的销量价格统计表：



**图4-4 三级机构范围的销量价格统计表**

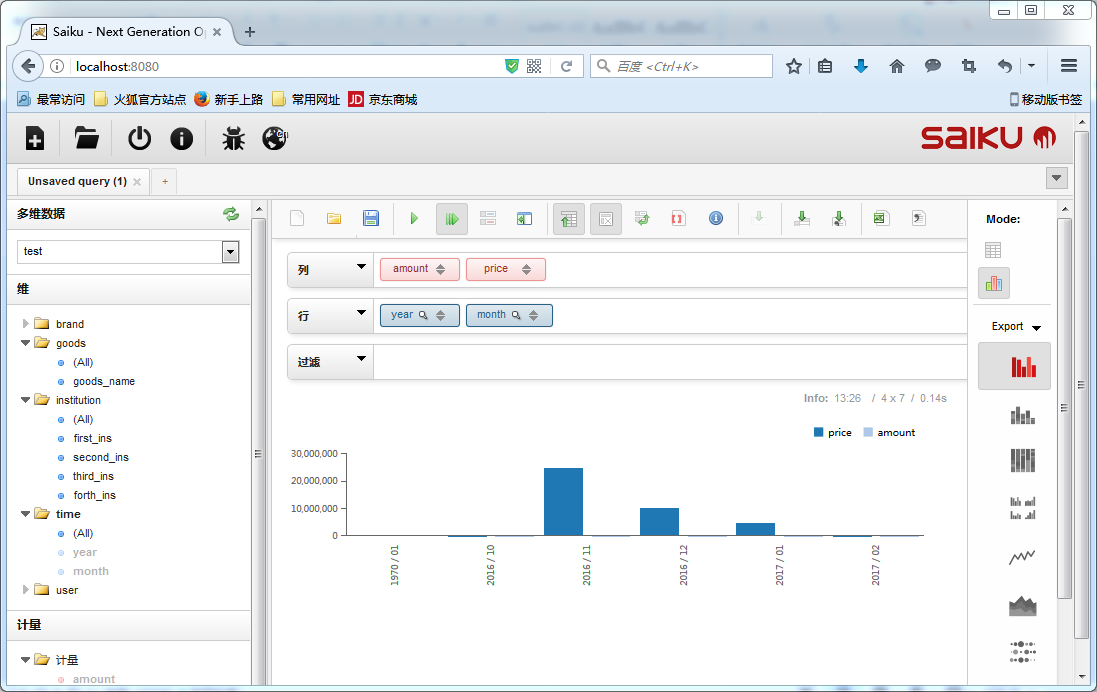
图4-5是三级机构范围的销量价格散点图：



**图4-5 三级机构范围的销量价格散点图**

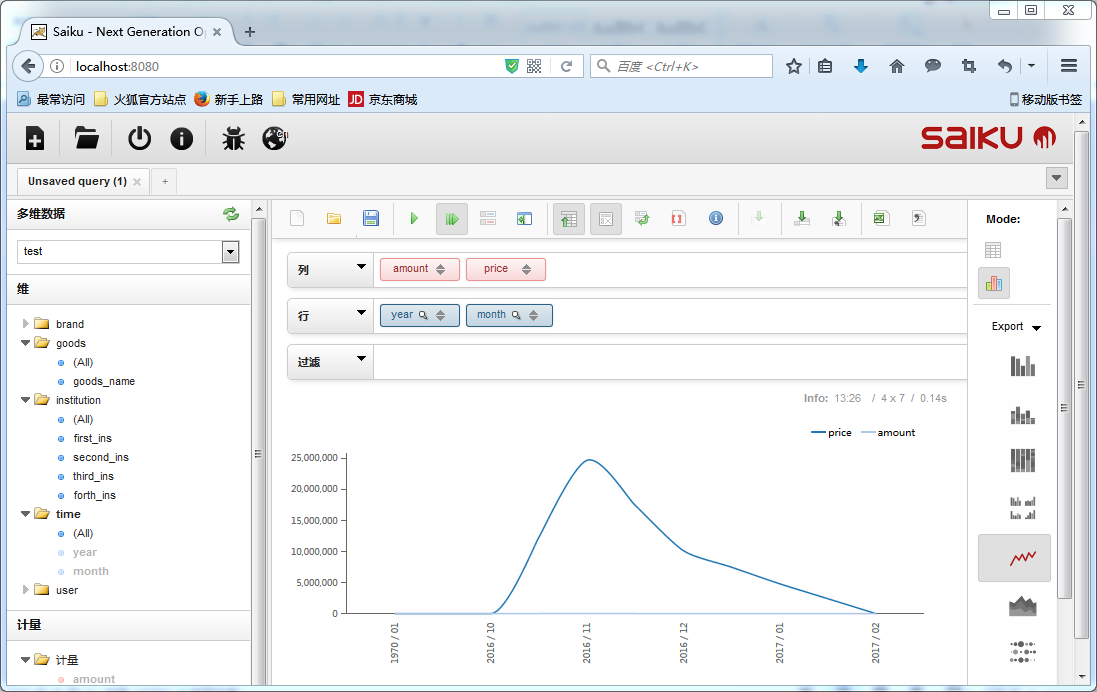
从散点图分布可以看出，整体不同机构的商品销量分布比较均匀，也有零星的机构地址的销量比较高。

3、列选取amount和price，行选取year和month，图4-6是其树状分布图：



**图4-6 具体月份的商品销量分布**

图4-7是具体月份的折线分布图：

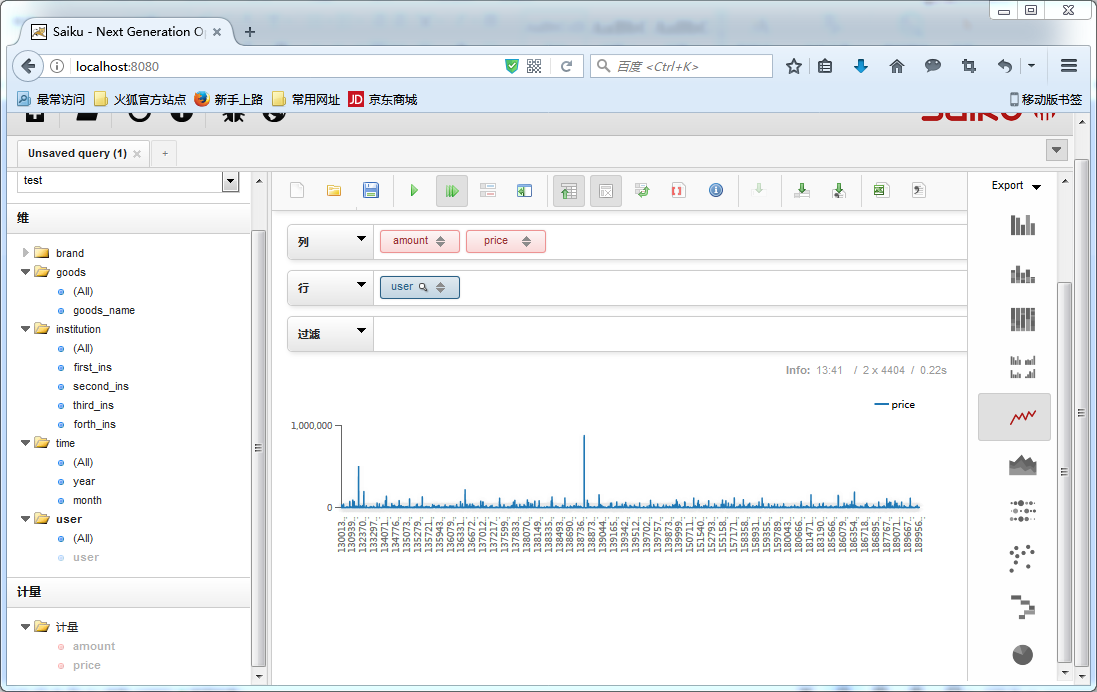
****

**图4-7 具体月份的折线分布图**

根据图4-6的树状图可以看出，2016年11月的销量是最高的，2016年12的销量稳居第二，随着时间的推移，销量逐渐减少。2017年2月的销量达到最低。

观察图4-7的销量分布折线图，不难发现，整个销量的变化随时间近似呈现开口向下的抛物线状，在2016年11月的时间结点上达到最高点，在该时间点左侧，销量随时间增加而增加；在这个时间点之后，销量随时间增加而减少。

4、列选取amount和price，行选取user，图4-8是不同用户的支付价格分布图：

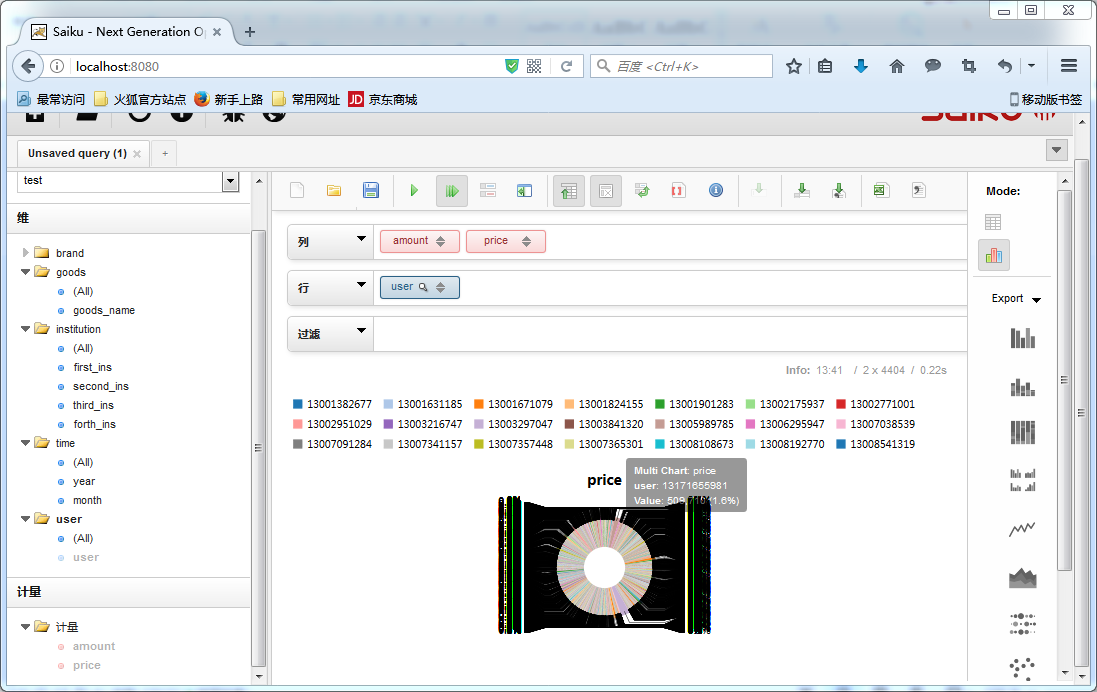


**图4-8 不同用户的支付价格分布图**

观察图4-8，横坐标是用户编号，纵坐标是购买的商品价格，整张图呈现了不同用户为其购买商品所支付的价格分布，可以看出不同用户的价格分布差，从图中得知，只有极少数用户支付的价格超出均衡价格，绝大部分用户的支付价格以较小的波动幅度均衡在同一价格水平。

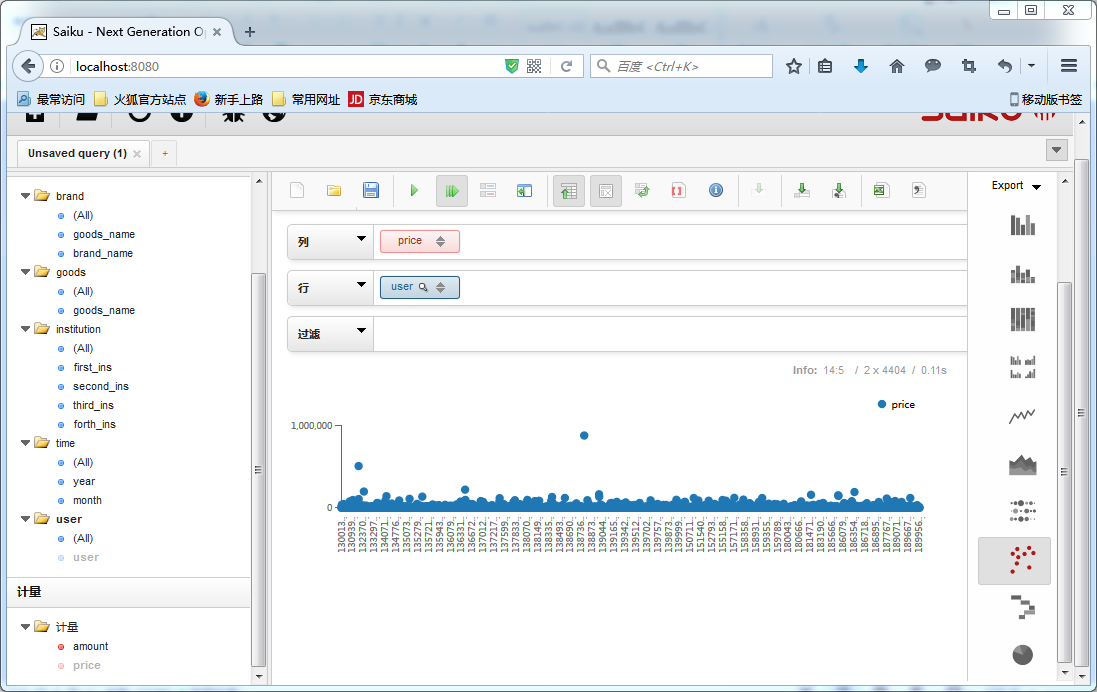
整体来说，大多数用户的愿意支付价格或者对该种类的商品的购买力处于同一水平线。

下图4-9是对应的饼状分布图，将鼠标靠近即可看到每一用户的价格分布：



**图4-9** **不同用户的支付价格饼状图**

如下，图4-10是不同用户支付价格的散点分布图，对用户和价格两个属性做散点图统计，更加清晰看出整体用户支付价格的分布水平。



**图4-10 不同用户支付价格的散点分布图**

## 4.3 本章小结

本章先简单介绍了在Tomcat下部署Saiku，将Saiku连接到Mysql数据库，修改配置文件，然后对Saiku多维表格进行可视化分析，对不同种类的商品销售数量和价格进行分析，选取三级机构范围的商品销售进行统计并分析，对不同年月份的销售量做统计，同时也对用户的购买分布情况进行分析，通过观察图表和数据，分析商品的销售变化情况，从而得出结论，为之后的决策分析打下基础。

# 第五章 总结与展望

## 5.1 总结

随着信息技术的高速发展，人们已经深刻的体会到准确的信息对于决策的重要性,许多企业,都已经在使用管理信息系统处理管理事务和日常业务,并且关注的焦点已经逐步转移到如何快速的对现有的大量历史数据根据不同的需求进行分析和处理,从中获取有价值的信息指导决策。数据仓库和联机分析处理技术是目前决策支持系统的热点,能够有效地为决策者提供各种类型的、有效的数据分析,起到决策支持的作用。

本论文突出了联机分析处理在电子商务中的应用实践，主要研究内容与工作任务包括：

1、阐述了相关理论与技术背景。在查阅相关文献的基础上，归纳了数据仓库、联机分析处理、ETL的概念定义并比较三者关系。

2、对项目进行总体设计。从具体的实际项目出发，对项目背景、项目任务进行了梳理，并介绍了项目的工作流程和时间进度管理。

3、进行OLAP多维分析。首先进行多维建模，建好Mondrian Schema模型之后开始进行数据ETL，将相关的数据加载到目标数据库中，最后在Saiku开源平台上呈现不同的多维表格，进行分析、总结。

4、进行项目方案的实现与测试。总体执行效果进行了测试评估与结果分析。

综上可以看出，通过整个过程的分析与研究，该项目在多维分析呈现方面获得了一定的成效，通过Saiku开源软件很好地展示了多维表格，更直观形象地阐释了不同维度数据的关联，动态多维分析整合数据，，达到了最初的项目目标。

## 5.2 展望

总之，OLAP的功能是非常强大的。当然，由于时间和经验的缺乏，本项目整个联机分析处理过程中还存在着许多的不足，将会对其进行进一步的完善与改进。联机分析处理技术是使分析人员、管理人员或执行人员能够从多角度对信息进行快速、一致、交互地存取，从而获得对数据更深入的了解的一类软件技术。

目前，OLAP技术有了很大的发展，市场上各种OLAP产品层出不穷。而通过OLAP，Saiku允许用户选择指标和维度用来分析，并且可以对数据进行“切块和切片”和钻取到细节中从而揭开其中的关系。用户可以在非常友好的界面下利用OLAP和内存引擎进行向下钻取，过滤、分类、排序和生成图表。

通过本次系统设计，对Saiku有了深入地学习和应用，Saiku是一个轻量级的OLAP分析引擎，可以方便的扩展、嵌入和配置。Saiku通过REST API连接OLAP系统，Saiku友好的界面呈现为分析数据提供了直接简明的方式。

# 参考文献

[1] 刘晓知. 网上电子商务信息联机分析系统的研究与实现. 中国地质大学出版社，2010.

[2] Inmon W H. Building the data Warehouse [M] . John Wiley & Sons , Inc ,1993.

[3] CSDN博客. Mondrian使用教程.

http://blog.csdn.net/xiaolang85/article/Details/45248289.

[4] 于秀梅. 数据仓库的多维分析展现技术应用研究. 北方工业大学出版社，2011.

[5] 王淑蓉, 赵颖. 数据仓库的OLAP多维展现技术的研究与应用. 2012 , 20 (14) :18-20.

[6] 齐金鹏. 数据挖掘模型可视化研究及其应用实例. 吉林大学，2004.

[7] 陈晴光. 电子商务数据挖掘可视化系统模型研究及应用. 2007 , 43 (5) :242-245.

[8] 林元元. JDBC连接MySQL数据库的方法浅析. 《湖南邮电职业技术学院学报》, 2009 , 8 (1) :42-45.

[9] 网易博客．Saiku Server环境搭建．

http://blog.163.com/zhk\_hoba/blog/static/140738113201412455554409/.

[10] 王晓鹏. 数据挖掘技术在电子商务管理中的应用研究.《中国新技术新产品》 , 2013 (18) :25-25.

[11]

[12]

[13]

[14]

[15]

[16] CSDN博客．Saiku的安装教程．

<http://blog.csdn.net/longshenlmj/article/details/17359645>.

[17] K Lefevre，DJ Dewitt，R Ramakrishnan. Mondrian Multidimensional K-Anonymity. International Conference on Data Engineering , 2006 , 6 (3) :25-25.

[18] CSDN博客. Mondrian Schema详解.

<http://blog.csdn.net/huangzijuan1/article/>details/6286380.

[19] 游章勇. 基于Mondrian在商业智能系统的设计与实现. 电子科技大学，2013.

[20] 杨真，李也白，籍志兵，常一帆. 数据分析技术在商品销售中的应用研究[A]. 2007.北京地区高校研究生学术交流会通信与信息技术会议论文集（上册）[C]，2008.

[21] 吴泽雄. 数据仓库浅谈[A]. 海南省通信学会学术年会论文集（2005）[C]，2005.

[22] 陈燕. 数据仓库的设计与实现[D]. 大连理工大学, 2000.

[23] 宋杰. 面向多类型数据源的数据仓库构建及ETL关键技术的研究[D]. 东北大学, 2008

[24] Michael kofler. MySQL权威指南[M]. 北京: 人民邮电出版社, 2006.

[25] 萨师煊, 王珊. 数据库系统设计(第三版)[M]. 北京: 高等教育出版社, 2004.

[26] 唐汉明,翟振兴,关宝军,王洪权,黄潇. 深入浅出MySQL[M]. 人民邮电出版社, 2014.

[27] 张强，邓清. 谈联机分析处理. 《信息技术》 , 2000 (2) :46-47.

[28] 叶得学，韩如冰. 浅谈数据仓库与OLAP技术. 《甘肃科技纵横》 , 2009 , 38 (2) :25-25.

[29] 张连第，王艳辉. 数据仓库技术在数据存储与数据处理中的应用. 2010 (26) :93-93.

[30] 黄晔玮. 基于数据仓库的联机分析处理. 《中国新技术新产品》 , 2009 (4) :11-11.