I.J. Engineering and Manufacturing **2011, 2, 21-27**
Published Online April 2011 in MECS (http://www.mecs-press.net)
DOI: 10.5815/ijem.2011.02.04

Available online at http://www.mecs-press.net/ijem

Database Design of a General Data Analysis System of Commodity Sales Information

Wang Guan[a], Wang Xiaolu[a]

[a] *Computer Institute Beijing University of Technology Beijing, China*

Abstract

A data analysis system for the general commodity sales information is researched on and further designed. The key problems of designing this system are that, it should enable the system to adjust to user-data with different structures, enable users to define or change data structures as well as retrieval methods. Through the research on the general structure and retrieval method of the commodity sales data, the system realizes users' customization of the database, thus is applicable to sales data with different structures. This paper focuses on the structure of the database, the creation method and process of the database by the user, the structure of the data dictionary and data exchange between database and software, with case examples in the final illustration.

Index Terms: database design; data analysis; user customization

1. Introduction

General data analysis system has the advantages of low cost and wide range of applications. However, most general software is industry software, it divides users by fields of industries. User customization stays in the level of application. Little software could enable users' customization of databases.

Companies may have different types of business and management processes, but they all have the same demands to analyze sales information. Based on this, the paper researches and designs a database structure of a general data analysis system. It discusses data exchange between database and software. The system realizes users' customization of databases, which will save more space and make the system slimmer and more efficient.

The data analysis system is delimited to analysis of commodity sales information. Focusing on discussion of the building of the database structure and its data exchange with software, this paper has proposed a core solution. It has proposed to build up different modules of commodities, customers, orders, salesmen or other

* Corresponding author.
E-mail address: wangguan@bjut.edu.cn

information that users may want to include. Users could load some of the modules as attributes or entities according to their demands. The system should make database tables balanced with efficiency and their paradigms.

2. Key Points to be Solved

Though the system is delimited to a commodity sales analysis system, and functions of the system are focused, users' commodities, ways of management and analysis information differ a lot. To make the system general, flexible, accurate, strong and efficient, problems below should be solved: (1) how to make the system adequate to different structures of sales information, (2) how to enable users to define and change structures of sales information, and (3) how to enable users to define and change their retrieval methods of sales information.

Design is the creative process of transforming problems into solutions [1]. This paper has proposed a solution for database system for general commodity sales analysis. It has designed a set of database tables and related database dictionary. It has illuminated the way of the software exchanging data with the database on the key point of the software operation. These problems above should be solved perfectly by this solution and thus a general data analysis system should be realized finally.

3. Database and Software Design

3.1. Structure Design of the Database

The most important attributes of database are table attributes, which include fields, types of fields, length of fields and if the fields are NULL [2]. Structures of data here mean different attributes of the sales information. The attributes should be different for different users' commodities and sale methods. These attributes are related and sometimes bounded to each other. To make the system applicable to different structures of data, structures of database tables must be customizable. Sales information could come from other sales managing system or could be input manually. To analyze the data, the system should be able to load the data freely.

To make it adequate to different structures of sales information, common features of the data should be found, according to that, a database structure should be designed, relations of each table should be defined. The system enables users to create and update different database tables in the structure. In this way, the system is accurate, flexible and safe.

The table to store sales data is named "sales". Each item records one kind of commodity in one sale. To analyze sales data is to retrieve data from this table. Attributes of this table decide what data should be analyzed, so it's the user's call what attributes and what types of them should be defined. As values of some attributes should be limited and users may want to manage some attributes separately, tables for entities with the attributes should be created. Some of the attributes would become foreign keys of the tables, and thus relate the tables with the sales table.

Sales data analysis is what the system to realize, it doesn't manage business process or other information not related. Retrieval methods defined by users may be complex as well. To avoid retrieving data from different tables, key attributes should be included to the sales table. In other words, if an entity set doesn't have one attribute included in the sales table, the entity should not be created at all.

Based on the analysis above, the structure or the database should be a single-layer divergent chart centered by the sales table, as Fig.1

The sales table is a weak entity set, its main key is formed by many foreign keys of other tables. It's not convenient to define an item by a formed key, so a main key generated by database system which is auto-incensement is added to the table. This makes the sales table a strong entity set.

Tables in the structure chart could all be created by users. It meets different demands, makes the system flexible and adequate to different structures of sales data. At the same time, user operation of the database is delimited to the framework, which ensures the database to work correctly and stable.

3.2. Database Creating Process and Data Dictionary

There are several problems caused by user-define databases. First, how to guide users to create those tables. Second, how to label the tables and attributes that users create, how to relate names of tables and attributes used in software with the ones used in database. Last, how to ensure referential integrity constraints of the tables.

To solve the problems and enable users to define and create structures of sales information, the database must have a data dictionary. Tables of the data dictionary will be introduced below by the order of database creating process.

1) Create sales table

The sales table is created by the system, it has one attribute which is the main key generated by the system from the very beginning, its type is bigint (auto-increasement). At the beginning of using the software, the user should add attributes to the sales table according to his demands, and assign types to the attributes he adds. Names of the attributes should be unique.

There is one table to store all types of attributes, as below:

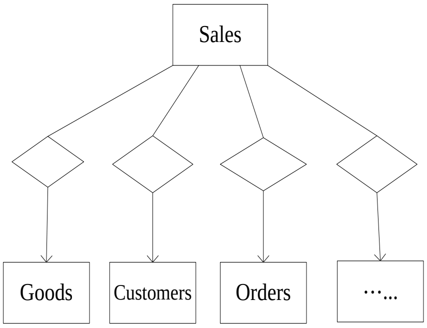

Fig. 1. Structure of the database

Table 1. Table to store attributes' types

code	**type**
1	INT
2	VARCHAR
……	…….

On the software interface, these types are listed in a dropdown list. When the user adds an attribute, he could choose a type from the list, and define the length or enumerate values if needed. All the attributes added will be recorded in a table, as below:

The table is to record attributes the user adds, their identifiers and the relations with entity table."attName" is the keyword, it should be unique. It's the name used in the software interface. "identifier" is generated by the system, it's the name used in the database. "belongTo" is used to record which entity table includes this attribute. “Keyword” is to record if this attribute is a keyword in another table, default “0”. “inSale” is to record if the attribute is also an attribute of the sales table, it’s used for the referential integrity constraints.

Table. 2. Table to record attributes added

<u>attName</u>	**identifier**	**belongTo**	**keyWord(0/1)**	**inSales(0/1)**
orderCode	a1	NULL	0	1
customerName	a2	NULL	0	1

Table. 3. Entities table

<u>tableName</u>	**tableIdentifier**
order	T1

Table.4. Attributes table

attName	**identifier**	**belongTo**	**keyWord(0/1)**	**inSales(0/1)**
orderCode	a1	T1	1	1
customerName	a2	NULL	0	1
orderDate	a3	T1	0	0

Table.5. Table to store commonly used retrievals

<u>Code</u>	**Name**	**Sql**
000001	Bigsale	Select * From sale Where a15 >= 10000;

2) Create entity table

After creation of the sales table, the user could select one or more attributes and add an entity table which includes the attributes selected. The user should name the entity table and make the table’s name unique. In order to make the table be in compliance with BCNF, the user should assign a keyword to the table, otherwise the system will add one as a keyword. Structure of the sales table will be updated, the newly added key of the entity table will be set a foreign key of the sales table. The user could then add other attributes of the entity table that are not included in the sales table. There’s a table recording entity tables the user adds, it has two fields, “tableName” and “tableIdentifier”. “tableName” is the keyword, it’s the name of the table used in the software, it’s unique. “tableIdentifier” is generated by the system, it’s the name of the table used in database.

For example, the user select “orderCode” and add an entity table named “order”, he assigns “orderCode” as a keyword and add another field “orderDate”. The information is recorded as below:

Fields of “belongTo” and “keyWord” in sales table will be updated. One item of “orderDate” is added as well.

All entity tables are connected with the sales table, they don’t link to each other in other connection, so the entity tables don’t have foreign keys. The sales table must have all the main keys of entity tables. When a value of attribute changes in an entity table, the system will check if the attribute is in the sales table, if it does, the items in the sales table should be updated as well.

The user could create other entity tables like “customers” and “commodities” as well in the same way. By the process above, the user could easily build a database system according to his needs. Entity tables and attribute tables record the structure of the database, they synchronize data information of all the tables.

3.3. Retrieval Schema

There are some conditions to make users’ retrievals relatively universal: the programmer does not need to write or edit sequenced query language (SQL) codes for retrievals, the retrievals could adjust to the change of database structure [3]. So the system should enable users to define and change retrieval methods. According to different needs, several retrieve schemas should be defined. Under the schemas, users could define retrieval methods. A common used retrieval could be saved.

Retrieval schemas could be defined by types of screening, aggregate functions and grouping. The three schemas could be used singly or together. The user should first choose a schema and then define his retrieval by select or input parameters. He can also write SQL in a text box.

4. Formulas preset

Users of the system could be technical staff or not. Configuration of the system requires some database knowledge and it has sort of difficulty to set the retrieval methods. To simplify user’s operation, Different formulas of the system could be preset, such as for convenience stores or brand stores, etc. Users could choose a most similar formula and adjust the system in details.

There should be several complete preset formulas. They include sales tables, all entity tables, attributes and the data dictionary generated automatically as well as retrieval methods. In short, try to make users adjust the system rather than create the system as far as possible.

5. System realization

A sales analysis system for convenient stores is realized by this system here.

First, the user defines structure of the sales table:

Table. 6. Sales table

code	**saleDate**	**commodityCode**	**commodityName**	**commodityPrice**	**saleNum**	**saleDiscount**	**saleAmount**
bigint auto-increasement	date	varchar	varchar	float	float	float	float

Table. 7. Commodities table

commodityCode	**commodityName**	**commodityPrice**	**commodityType**	**commodityPackage**
varchar	varchar	Float	varchar	varchar

Table. 8. Attributes table

attName	**identifier**	**belongTo**	**keyWord(0/1)**	**inSales(0/1)**
saleDate	a1	NULL	0	0
commodityCode	a2	T1	1	1
commodityName	a3	T1	0	0
commodityPrice	a4	T1	0	0
saleNum	a5	NULL	0	0
saleDiscount	a6	NULL	0	0
saleAmount	a7	NULL	0	0
commodityPackage	a8	T1	0	0
commodityType	a9	T1	0	0

Table.9. Entities table

tableName	**tableIdentifier**
Commodities	T1

Table 10. commonly used retrievals table

code	**name**	**sql**
000001	sales amount of the day	Select SUM(a7) From sale Where a1= SystemDate();

After that, the user chooses “commodityCode”, “commodityName” and “commodityPrice” and creates an entity table named “commodities” with them.

Meanwhile, the attributes and tables created by the user are recorded.

The user defines two retrievals: Sum up “saleNum” of the same commodities and sort records by “saleNum” in descending order. Calculate sales amount of the day. He defines the latter one as a commonly used retrieval and it’s saved in the table.

It’s a comparatively easy system for convenient stores, as there is only one entity table needed. A convenient store usually doesn’t manage orders or customers information. However, this system illuminates that the structures could make it applicable to sales data with different structures, users could define and change structures of data and retrieval methods. The convenient stores data analysis system could be preset as a schema as well.

6. Summary

Taking the “general data analysis system of commodity sales information” as an example, this paper introduces a solution of database design for general systems. It focuses on designing of database structures and data dictionaries, data exchanging between database and software. The data dictionary ensures accuracy and flexibility, by which the functions could extend as well. After feeds back from users time and time again, the system will become better and better.

Comparing with regular databases, general databases are more abstract, more flexible and more complex. To accurate the system's design, to make applications feasible, a framework should be defined. After further research in the framework, a solution with certain schemas or other feasible methods will be found.

References

[1] Shari Lawrence Pfleeger, "Software engineering theory and practice", 2001, pp195.

[2] Chun-ji Hao, "Converting a relation database into object automabily", Microcomputer information[J], 2007,4-3，pp192-194.(in Chinese)

[3] Ren-peng Sun, "Design and implementation of universal dynamic query system based on system table" [J],Computer Engineering and Design, 2007,28-23，pp5819-582.(in Chinese)