An Approach for Automatically Generating Star Schema from Natural Language

Rosni Lumbantoruan, Elisa Margareth Sibarani, Monica Verawaty Sitorus, Ayunisa Mindari, Suhendrowan Putra Sinaga
Del Institute of Technology
Jl. Sisingamangaraja, Sitoluama, Laguboti, Kabupaten Toba Samosir, Sumatera Utara
*Corresponding author, e-mail: rosni@del.ac.id,elisa@del.ac.id

Abstract

The star schema is a form of data warehouse modelling, which acts primary storage for dimensional data that enables efficient retrieval of business information for decision making. Star schemas can be generated from business needs that we refer to as a user business key or from a relational schema of the operational system. There are many tools available to automatically generate star schema from relational schema, such as BIRST and SAMSTAR; however, there is no application that can automatically generate it from a user business key that is represented in the form of human language. In this paper, we offered an approach for automatically generating star schema from user business key(s). It begins by processing the user business key using a syntactical parsing process to identify noun words. Those identified words will be used to generate dimension table candidates and a fact table. The evaluation result indicates that the tool can generate star schema based on the inputted user business key(s) with some limitations in that the star schema will not be formed if the dimensional tables do not have a direct relationship.

Keywords*: star schema, user business key, parsing process, fact table, dimensional table*

1. Introduction

The data warehouse is dimensional data storage containing accumulated transactional data from a wide range of sources within an organisation and is used to guide management decisions [1]. Its purpose is to support decision-makers to take decisions quickly, since the operational database has a vast amount of data and it is possibly changing over time. The initial step to data warehouse modelling is explicitly stating the user business key in the form of natural language so as to identify which data needs to be accumulated to support the decision-makers. However, it is a time-consuming process since many steps need to be completed, starting from understanding the user business key which is expressed in natural language, continuing with identifying the dimensional table based on the user business key and so forth. From a practical point of view, data warehouse modelling processes need to be improved in terms of the efficiency of decision-making time. There are several kinds of data warehouse models, and this research will focus on dimensional modelling, specifically on star schemas. The star schema is a relational schema that has one or multiple fact tables which are referenced to one or more dimension tables [2].

Related research regarding automatic generation tools of star schema such as SAMSTAR [3], BIRST [4] is available and was carried out in [5] with the main objective being to transform an entity relationship model into star schema. The transformation process accepts the transactional database and the expected output by looking into the cardinality of each table is candidate fact tables. The main challenge is if the database contains many tables, as it would be a time-consuming process. Also, it would be difficult for the user to find and select the fact table if many candidate fact tables are generated by the application.

Therefore, the proposed solution is processing the user business key which is written in human language by implementing several steps defined in natural language processing (NLP). NLP is a set of techniques used to examine the grammar structure and the meaning of the sentence which was given by the user for the purpose of attaining information from the sentence [6]. The main difference with the research in [5] is in the input, which was user business keys, and therefore the whole process of modelling the star schema is also completely different. The aim of our research was to build an application that is able to process user business key using

Received January 30, 2014; Revised May 14, 2013; Accepted May 27, 2014

natural language processing, continue with designing and creating a data warehouse using star schema modelling in accordance with the user business key.

The scope of this research is to translate a user business key in the form of natural language written in English to provide the information desired by the user. To do so, a series of natural language processing tasks must be done to obtain words which are categorised as nouns and to be able to identify candidates for dimension tables. Dimension tables are used as the basis for the establishment of a related fact table, hence both fact and dimension tables form a star schema. The application was built using C# language and the library which was used for implementing natural language processing is Proxem Advanced Natural Language Processing Object-oriented Environment (Antelope) [7]. The evaluation process of the application uses AdventureWorks and Northwind databases.

This paper is organised as follows: Chapter 1 describes clearly the problems occurring and the motivation for doing the research. Therefore, the general objective and proposed solution should also be stated. We then continue with chapter 2, where we explain the steps of our research in order to get the solution to the problem stated in chapter 1. We continue with chapter 3, in which a description of the literature study is given. In chapter 4, the analysis result will be explained to present our solution to solve the problem and define the result of the evaluation process. Finally, in chapter 5 we point to the conclusion of the research based on an evaluation of the result and also raise recommendations for future research work related to this field.

2. Research Method

Our research is mainly analytical. It tries to find a solution to automatically generate star schema from requirements expressed in human language. Our research was conducted by applying approaches usingthe following steps:

1. Conduct literature study on natural language processing and data warehouse concept, specifically on star schema modelling. At this stage, the learning and understanding of those concepts was by reading and understanding papers and textbooks available on those fields.
2. Explore and analyse several natural language processing tools. Some tools are provided in the form of libraries such as Natural Language ToolKit (NLTK) and Proxem Advanced Natural Language Processing Object-oriented Environment (Antelope). The result was used as practical consideration in the selection of the tools that will be used in this research.
3. Analyse the process for star schema design to implement several findings in the defining solution to answer the research problem.
4. Develop an application which consists of basic software engineering steps such as application analysis, design, and code programming to build the final application.
5. Evaluate the application by using existing databases in SQL Server which are AdventureWorks and Northwind. We used those two databases as the sources to buildour data warehouse. The evaluation was done by analysing the star schema result based on the user business key which was entered into the application. Based on those results, we evaluated the result manually and decided whether the resulting star schema was good or not.

Analyse the evaluation result in order to summarise the research question and conclude whether the problem was answered or not.

3. Analysis

In this section, we explain our analysis results and approach findings to automatically generating the star schema.

3.1. User Business Key

Practically, the decision-makers present their information needs in a user business key that is usually in the form of natural language which has characteristics of unlimited structure and sometimes is not based on the true grammar of that particular language. Therefore, in order to enable a computer to understand natural language, research is done in the field known as natural language processing [8]. The general architecture of natural language processing consists of (1) a chunker, which is the software for parsing one sentence into a set of words and is usually stored in an array or a list, (2) a parser, which gives the tag based on the type of words and also rules of grammar based on the chunking results and finally forms a sentence tree, and (3) a part-of-speech tagger (POS Tagger), which will tag every word that has been parsed in a sentence based on a complete tag set in order to obtain the relationship between words. The tags that are given are more complex and detailed than the tag by the parser, so that POS Tagger can distinguish variations in verb forms, variations of singular and plural nouns, and variations in conjunctions [8].

Several studies were conducted on building natural language processing tools, and one of the results was Antelope (Advanced Natural Language Object-Oriented Processing Environment). It was a library written in C# for doing natural language processing which consists of three processes, which are tagging, chunking, and parsing, and also a list of words needed to process sentences in English.

We define a format to simplify the process for the user who provides the user business key as an input to the tool. Several guidances are available to help users define their user business key:

1. The sentence must be an interrogative sentence.
   Basically, a data warehouse is built to provide the information needed by the user, such as “What are the average sales?” and “What is the most ordered product?”.
2. The sentence must contain more than one noun and should have the same word with the name of an existing table in the database source. There are two reasons why we need a user business key that must contain two nouns:
   a. Star schema modelling could be done by finding any relationship between candidates of dimension tables which are found from any nouns in the user business key. Therefore if the user business key only has one noun, then it will need much effort to find all relationships which one candidate dimension table has.
   b. If the user business key has more than one noun, then the process can continue to form the star schema. However, the number of nouns can affect the performance of the application. It must search through all possible relationships among the dimension table candidates. For example if the user provides the user business key: "Who are the vendors whose products are the most widely purchased by any customers?". There are three nouns which were found: "product", "customer", and "vendor". Therefore the next process needs to find a combination of possible relationships between those candidates: relationship between "product" and "customer", between "customer” and "vendor", and between "vendor" and "product". The result will definitely be that more time will be required to search the relationship between those nouns. Therefore, in this research we restrict the number of nouns to be processed to only two.

There are two defined structures used in general for interrogative sentence in English: yes/no question and word question categories. The structure for user business key must be in the form of a question. The structures of interrogative sentences which use a question word are as follows:

1. *Question Word* + *Verb* + *Object*
   Example: *What is your name*? The object can be a noun.
2. *Question Word* + *Auxiliary* + *Subject* + *Verb* + *Object*
   Example: *When does the teacher meet the student*? Both subject and object could be noun words.

Therefore the user business key format which is written in interrogative form must follow the structures below:

1. [w*hat* / *who* / *where* / *when*] + *Verb* + [*Measure*] + of + [*Table's Name*] + *Verb* + by + [*Table 's Name*] + [*Adverb of time* / *Null*]
   Example: *What is the highest price of the product purchased by a customer*?
2. [*what* / *who* / *where* / *when*] + *Verb* + [*Table's Name*] + and + [*what* / *who* / *where* / *when*] + *Verb* + [*Table's Name*]
   Example: *Who are our customers and what products are they buying*?

3.1.1. User Business Key Processing

The natural language processing library that we used was Antelope, which performs three processes: tagging, chunking, and parsing. This research only used the tagging process in obtaining a list of nouns because the tool requires those noun words to build a data warehouse. In detail, the user business key process can be described as follows:

1. Splitting to deconstruct a sentence to get a list of words in the sentence. An example input is: *What is the most ordered product by customer in June 2010?* The process will use String.Split, which will store the information as a list of words in an array of strings. The result of this process are: { "What", "is", "the", "most", "ordered", "product", "by", "customer", "in", "June", "2010"}.
2. Noun identification is the process of identifying and giving a tag to each word. Identification is performed using BrillTaggerLexicon.txt. The list of noun words which were found should be stored in a list of strings. If no noun word has been identified, then the tool will ask the user to enter a new user business key. The output of this process is: "product", and "customer".
3. Measure identification is the process of searching a word that would be a measure in the fact table. To obtaining a measure word a comparisonwith the list of statistical word which is listed in Table 1. If the process could not find any measure word in the user business key, we automatically assigned "quantity" to be used as the measure.
4. Noun form identification is the process of searching a singular or plural form of the noun. If the noun which was identified is in the singular, then the process will find the plural form of the noun and vice versa. Identification was performed using Proxem.Lexicon.dat file. The result of this process is added to the list of nouns.

Table 1. List of Statistical Word

No	Function	Definition	Keyword
1	AVG	average	*Average, Mean*
2	SUM	total	*Total, Summary*
3	MIN	minimum amount	*Minimum, minimal, highest*
4	MAX	maximum amount	*Maximum, maximal, lowest*
5	COUNT	number of occurence	*Count, Number of*

5. Finding dimension tables based on the noun word. The input is a list of nouns which was produced by the previous process. Based on our example, the list of words are "product", "customer", "products", "customers". The first step is to elicita list of table’ names from the database source. This process uses sysdatabase to retrieve all table names. The query which was used to retrieve all table names had the following syntax: "Select * from Information_Schema.Tables where Table_Catalog = 'AdventureWorks' and Table_Type = 'BASE TABLE' and Table_Name NOT IN ('sysdiagrams','dtproperties')". The second step is to compare the name of the table with our list of nouns. The output of this process based on the above processed user business key are: "product " and "customer ".

3.2. Star Schema Modelling

Data warehouse modelling aims to facilitate the presentation, design, availability, accessibility, reliability, and understandability of information so that it can present the facts needed by decision-makers of the top-level management in a company. Data warehouse modelling can be performed using two techniques, which are entity relationship and dimensional modelling. Entity relationship modelling is a technique used to model the data at the conceptual level of database design [9].

Star schema modelling is the most commonly used technique, which consists of a large central table (fact table) to store the largest collection of useful data and supporting tables (dimension tables) [10].

3.2.1. Creating Dimension Tables

The main objective of this process is to find the table and column of that table which has sa imilar name to the candidate dimension table from the processed user business key and also to denormalise the dimension table which was found. The detailed process of creating dimension table is as follows:

1. Validate the number of dimension table candidates; if it is more than one, then proceed to the next process, but if it is smaller or equal to one, then the end the process to ask the user to enter another user business key.
2. Trace the relationship that dimension table candidates have had in the source database. This process will check through the entire transaction table in the source database to examine the relationships among the dimension table candidates. If no candidates are interconnected with each other, then no new fact table will be formed and the process will stop. The process of finding relationships between the dimension table candidates has three steps:
   a. Check the transaction table that has a primary key of the dimension table candidates. This process can be done using sysdatabases by checking for a table that has a foreign key as well as a primary key.
   b. If the transaction table has no primary key column of the dimension table candidates then the process will check the primary key of the dimension table of the transaction table that was checked. If the dimension table from the transaction table has relationship to the dimension table candidate then we could say that they are related. This process can be done using sysdatabases by examine the source table of the foreign key in the dimension table of the transaction table.
   c. If the transaction table that is checked only has a relationship to one of the dimension table candidates then other dimension table candidates should be checked with the dimension of the transaction table. This process can be done using sysdatabases by examining the source table of the foreign key in the dimension table of the transaction table.
3. The denormalisation of dimension table candidates has three steps:
   a. Finding all foreign keys of each table to retrieve all attributes required by the table to support the denormalisation process of those tables. The search process can be completed by executingQuery 1.

```
SELECT ac1.[Name] AS 'ColumnName' FROM
sys.foreign_key_columns fk INNER JOIN sys.all_columns ac1 ON
fk.parent_object_id = ac1.[Object_id AND fk.parent_column_id
= ac1.column_id Where OBJECT_NAME (fk.parent_object_id) =
'Product '.
```

Query 1. Finding Foreign Key

   b. Finding the source table of the foreign key and retrieving all attributes that the source table has. This process is carried out by executingQuery 2.

```
SELECT OBJECT_NAME (f.rkeyid) AS 'PKTable' FROM sysforeignkeys
f INNER JOIN syscolumns c1 ON f.fkeyid = c1.[Id] AND f.fkey =
c1.colid INNER JOIN syscolumns c2 ON f.rkeyid = c2.[Id] AND
f.rkey = c2.colid Where OBJECT_NAME(f.fkeyid) = 'Product' + "'
and c1.[name] = '<foreign key>';
```

Query 2. Finding the Source Table

   c. Adding all the attributes generated in the previous processes into dimension tables.

The output result of this process includes all dimension tables together with all attributes of those tables, as can be seen in Figure 1.

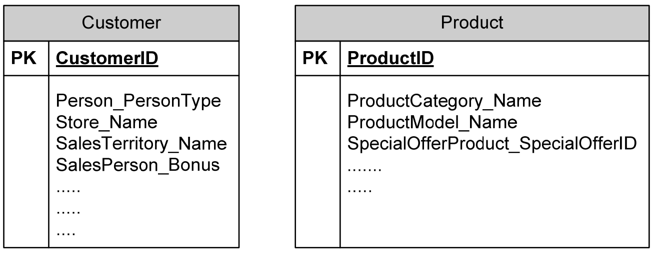

Figure 1. New Dimension Tables

3.2.2. Creating Fact Tables

A facttable was created from each primary key found in every dimension table and the statistical word contained in the user business key will become the measure in the fact table. The process continues by asking the user for feedback about what the right measures is, which will be used for numerical attributes. The detailed process of creating fact tables is as follows:

1. Choose name for fact table by combining all dimension tables' names. For example, if we have dimension tables "product" and "customer", then our fact table's name is "CustomerProduct".
2. The search for a primary key in the dimension tables is carried out because a fact table contains all keys in the dimension tables. To find the primary key, Query 3 can be executed.

```
SELECT col_name (ic.OBJECT_ID, ic.column_id) AS ColumnName FROM
sys.indexes as i INNER JOIN sys.index_columns AS ic ON
i.OBJECT_ID = ic.OBJECT_ID AND i.index_id = ic.index_id WHERE
i.is_primary_key = 1 and OBJECT_NAME (ic.OBJECT_ID) =
<product>;'<foreign key>';
```

Query 3. Finding Primary Keys

3. Adding a primary key from a dimensional table into the fact table. All primary keys which are found in the previous step are stored in the list of strings that holds the column names and column types.
4. Adding measures to the fact tables which were found in the user business key beside "quantity". The measure would become a value in the fact table so that it has a conclusion or summary. The input of this process is the list of measures which was found in the previous step, which is the user business key process. It continues by obtaining the measures unit by following these two steps:
   a. Looking for a column in the dimension table that has a numeric value by checking the data type of each column in the dimension table such as integer, double, long, etc. and storing it in the list of columns.
   b. Asking for feedback from the users by displaying a list of columns to the user and asking them to select which column can be calculated based on the measures. The measure which is created is semi-additive or additive.
5. Defining fact table columns by following these steps:
   a. Finding additional measure candidates by obtaining all measures contained in the transaction table that have been obtained in the process of searching for relationships between dimension table candidates.
   b. Getting feedback from users by displaying all measures which are contained in the transaction table and which connect dimension tables and asking them to select an additional measure used in the fact table.

c. Creating fact table and giving it a name by joining all dimension table names, for example "ProductCustomer". Columns of the fact table include the primary key of each dimension table, identified measures and user-selected measures.

3.2.3. Creating Time Dimension Tables

The time dimension table is created by checkingthe data type of the transaction table column which connects the dimension tables. If the data type is datetime then the time dimension table can be created. Only if the process finds any columns with the datetime data type, can we establish a time dimension table. After we succeed in creating the schema for dimension tables, fact tables, and time dimension tables, then the process can continue anda data warehouse schema can be created and implemented into the database server.

3.3. Building Data Warehouse Schemas

Input in this process is a list of fact tables, dimensional tables, and a list of time dimension tables (if established). The user is asked to provide feedback for data warehouse name. In detail, the process can be described as follows:

1. Validating the data warehouse name to prevent data warehouse duplication. If the data warehouse name already exists, then the next step is to check for redundant tables. If it exists, the application will alter the table by deleting redundant tables. If the entered data warehouse name does not exist, the process will create a new data warehouse schema.
2. Validating the fact table to prevent fact table duplication. If it exists then the next process is checking all columns of the fact table; if any column differs from the columns of the fact table, then the process will alter the existing table. If the process does not find the same fact table then the application will add the new fact table.
3. Validating the dimension tables to prevent dimension table duplication. If it exists then the next process is checking all columns on the dimension table, if any column differs from the columns of the new dimension table then the process will alter the existing table. If the process does not find the same dimension table, then the application will add the new dimension table.
4. Defining a query to build the new data warehouse by defining all table names, column names, data type of each column, and relationship of each table in string type and storing it as a variable. The variable is added using keywordswith specific SQL syntax to form query syntax to create database tables on the server.
5. Adding log changes to the log database to record any changes to the data warehouse. The log database design can be seen in Figure 2.
6. A data warehouse schema could be created by tracking all columns and relationships between dimension and fact tables. A data warehouse schema can be seen in Figure 3.

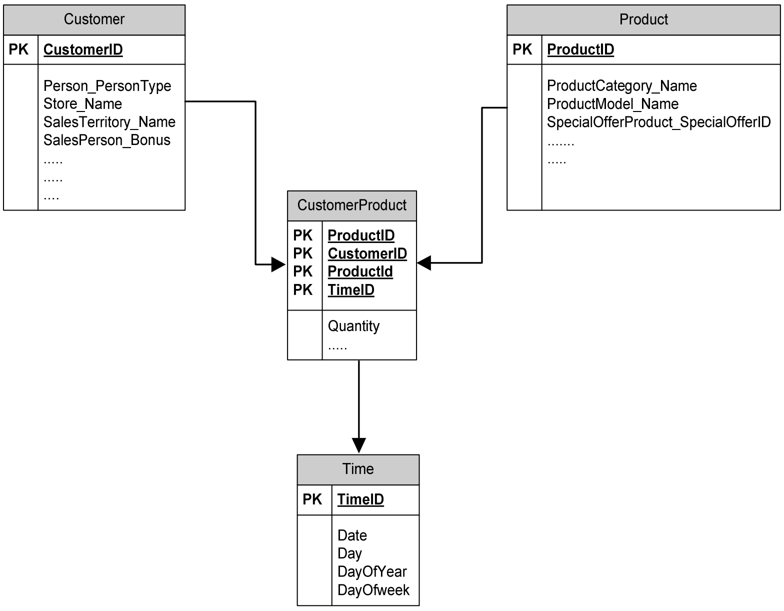

Figure 2Data Warehouse Schema

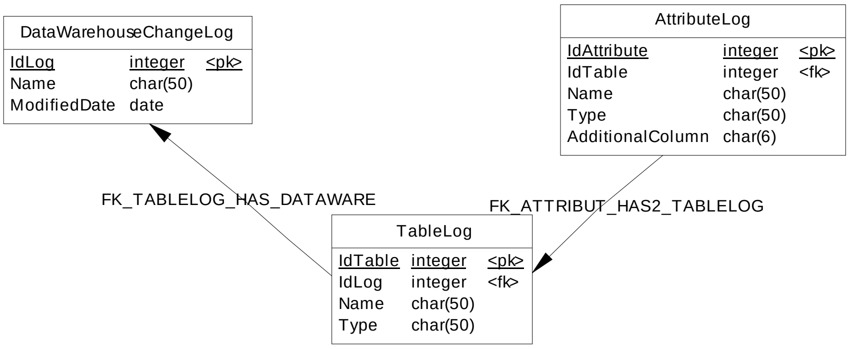

Figure 3.Log Database

4. Result and Discussion

A user business key is a sentence that identifies the key information that decision-makers need to support their business. The key information contains basic dataof an organisation, which is used to conduct detailed analyses and derive business value. Since a user business key is often presented using natural language, we delivered an approach to retrieve the information by processing the natural language to elicit noun words through syntactical parsing. Identified noun words will then be mapped with the tables in the operational database to get the dimensional table candidates. The process continued to identify the fact tables with the summarisation of the measurement data associated with dimensional tables. Therefore, we have to firstly find the relationship between dimensional table candidates.

Based on the approach, we developed a tool which is able to automatically generate a data warehouse by accepting user input, which is then used to create a star schema. There are two main functions of the tool: (1) processing user business key to obtain noun words for dimension table and the measures that will be used (Measures are needed for fact table and key of the dimension table); (2) build the data warehouse based on the dimension table and measure resulting from the previous process. The fact table is formed using the primary key of each dimension table and measure. The result of this process is the data warehouse schema.

In order to evaluate the analysis result, we executed the tool by providing the user business keys as an input. Due to the wide range of user business key sentences, we categorisedthem by doing the following:

A. User business key has a statistical word and two nouns corresponding to the name of the table in a relational database.
   1. What is the highest price of the product purchased by a customer?
   2. What is the average purchase of the products supplied by vendors?

B. User business key does not contain statistical word and two nouns corresponding to the table's name in the relational database.
   1. Who is our customer and what products are they buying?
   2. What product is the most widely purchased from any vendor?

C. User business key has more than two nouns corresponding to the table name in the relational database.
   1. Who are the vendors whose products are the most widely purchased by any customers?
   2. What is the address of the customer who bought the product in 2003?

D. User business key has a single noun.
   1. What is the amount of product sales in 2006?
   2. What are the average sales of products in June 2006?

E. User business key requires semantic understanding.
   1. What products are profitable?
   2. Who is the most frequent customer who makes a purchase?

F. User business key which are not structured as an interrogative sentence.
   1. Give me the average purchase product from vendors in 2004.
   2. I want to know our customers and what products are they buying.

G. User business key which has noun in plural form.
   Who are our customers and what product are they buying?

From the evaluation, in general, the output result of each user business key being processed by the tool could be considered successful; however, we found three points that need to be considered for the proposed approach:

1. The number of nounsin the user business key will affect the algorithm complexity; the more nouns there are, the more complex the algorithm. In order to create a fact table, we have to identify a transactional table that connects all the dimensional tables. For example, with two dimensional tables, the algorithm will run the method to find the relationship once only. As the number of dimension table candidates (represented as n) increased, the number relationships will form a polynomial series with Equation 1.

$$\frac{n^2-n}{2} \tag{1}$$

2. The approach was not able to define a star schema if there is no direct relationship between the dimensional table candidates.

The approach was not able to define a star schema for user business keys that require semantic understanding. For example, the test scenario F1 has the word "profitable", which means all products that have the highest sales. Therefore, the transaction table which should be identified was SalesOrderDetail. It should display the star schema that has the Product table and Customer table as dimension tables and CustomerProduct tables that have the same column as the SalesOrderDetail table. However, due to limitations of our natural language processing approach in identifying semantic meaning, those kinds of user business keys cannot be processed properly.

5. **Conclusion**

The created tool is able to create a star schemaby processing a user business key which is written in English and contains exactly two nouns. The user can use the tool repeatedly and ensure that it will display the newest star schema based on the needs of the user.

Therefore, the expected future development is to process user business keys with more than two nouns in Indonesian. The process of creating a star schema requires considerable time, but by improving search algorithms for dimension and fact tables, the tool is expected to create star schema in a shorter time.

This research acts as a pioneer for future large tool development which is able to integrate all processes starting from user business key processing which is able to identify semantic meaning by implementing all processes in natural language processing. It can be done by providing the lexicon in a specific language, and thusenabling it to identify whether the user mistyped the business key and notify the user. The output of the user business key should be shown on a dashboard to help the user understand the information generated by the tool.

References

[1] Rainardi V. *Building a Data Warehouse: With Examples in SQL Server*. First Edition. CA USA. Apress Berkely. 2011.

[2] Stephen F*. Dashboard Design for Real-Time Situation Awareness*. Perceptual Edge Consultancy. 2007.

[3] Song IY, Khare R, Dai B. *SAMSTAR: A Semi-Automated Lexical Method for Generating Star Schemas from an Entity-Relationship Diagram*. In: 10th ACM Int'l Workshop on Data Warehousing and OLAP (DOLAP 2007). ACM, New York. 2007: 9-16.

[4] http://www.birst.com/product/technology/data-warehouse-automation. Automatic star schema generation, BIRST.

[5] Andreas P, Daniel N, Rina S. Transformator Entity Relationship Model into Star Schema. *Diploma Thesis*. Toba Samosir. Politeknik Informatika Del. 2012.

[6] Ahmad S. *Tutorial on Natural Language Processing*. University of Northern Iowa. Artificial Intelligence Fall. 2007.

[7] https://www.proxem.com. Proxem Antelope for Microsoft .net. Version 0.8.7. March 2009.

[8] Goyal S, Bhat S, Gulati S, Anantaram C. Ontology-driven Approach to Obtain Semantically Valid Chunks for Natural Language Enabled Business Applications. *Research in Computing Science. Special Issue: Natural Language Processing and Its Applications*. 2010; 46(1): 105-116.

[9] Ralph K, Margy R, Warren T, Joy M, Bob B. *The Data Warehouse Lifecycle Toolkit*. Second Edition. Wiley. 2008.

[10] Jiawei H, Micheline K. *Data Mining: Concepts and Techniques*. Second Edition. Morgan Kaufmann. 2005.