A User-Friendly Approach to Design a Multidimensional Model from Relational Databases

Geetika Chaudhary
Khyati Jain
Neha Sharma
Ruchira Gupta
IT
Bhagwan Parshuram Institute of Technology
Delhi, India

Prof. (Dr.) Payal Pahwa
CSE & IT
Bhagwan Parshuram Institute of Technology
Delhi, India

***Abstract*— Data in a data warehouse is organized in a multidimensional model. This multidimensional model helps in faster query processing and efficient OLAP operations for data analysis and decision making. In this paper, we introduce a framework which proposes design methodologies to map the relational database into a multidimensional model. The process starts with first cleaning the relational database and then categorizing the attributes of this cleansed relational database into metrics and dimensional attributes by applying the proposed set of mapping rules. This approach has been analyzed using the case study of bank management system and has been implemented successfully using WAMP 2.0 (PHP 5.3.0 and MySQL 5.1.36).**

***Keywords*— Data Warehouse, Extraction Transformation Loading, multidimensional model, star schema.**

I. INTRODUCTION

A database is a data repository which includes a collection of entity sets in the form of rows and columns, and each of these entity sets contains any number of entities of the same type. A relational database is a shared repository of data [1]. These databases are used for day to day transaction processing and are also known as OLTP (On-Line Transaction Processing) systems. These systems do not fulfill the out-of-the-ordinary tasks of information processing and data analysis, therefore to overcome this drawback and to introduce added advantages from a data repository, data-warehouse concept was presented. Data-warehouse is a subject oriented, integrated, non- volatile and time variant collection of data in support of management's decision [2][3]. Data-warehouse has a broader scope and is fully equipped with the latest technologies supporting the decision making process.

The ETL (Extract, transform and load) process works at the backend for the design of data warehouse. ETL is a process in database usage and especially in data warehousing that involves:

- Extracting data from outside sources
- Transforming it to fit operational needs (which can include quality levels)
- Loading it into the end target (database or data warehouse)[4]

The main source of the data is cleansed, transformed, loaded and made available to managers and other business professionals for data mining, online analytical processing, market research and decision support. The ETL process basically takes place in the data staging area where the data is integrated and cleansed. Data cleaning, refers to the process of removing errors and inconsistencies from the data so as to improve the quality of the data**.** In a data-warehouse data comes from different sources and is therefore available in various different forms and formats. The syntax and semantics are different for each of the multiple data source. So, if the data is stored as it comes from the data sources, it would create inconsistency in the data-warehouse, leading to confusion, and hence degrading the quality of the data.

Multidimensional models like cubes, hyper cubes, star schema, snowflake schema etc are used for representing the data in a data warehouse. In our approach, we use star schema for this purpose. A star schema comprises of two major elements - fact (areas of interest for making strategic and analytical decisions for example: sales) and dimension (a base for fact analysis, for example: time, employee etc) [2]. Each fact contains a set of numerical attributes called metrics E.g. number of products sold, profit etc. and each attribute within a dimension is called dimensional attribute. E.g. price of a product, customer salary etc.

The concept of metadata is also being focused upon in this paper. Every database has a Data Dictionary which stores metadata about the structure of the database, in particular the schema of the database. For example, Oracle Designer stores the design in Oracle Repository, which serves as a single point of metadata for the application. Metadata describes all the pertinent aspects of the data in the database fully and precisely. While building a data warehouse, one requires metadata about the source systems, source to target mappings and data transformation rules. The metadata can then be used to generate forms and reports. [1][2]

Data-warehouse design and development proposed earlier required ad-hoc methodologies. The star schema is one such logical representation which supports the conceptual modeling phase in the data-warehouse design. The earlier proposed work was based on the mapping of UML diagrams to obtain a multidimensional model (star schema or multi-dimensional ER model). We have formulated an approach to simplify the

transformation of relational database schema to multidimensional form. We retrieve the metadata of relational database and use it to design the star schema of the corresponding relational database. The column names fetched from the metadata are segregated to figure out dimensional attributes and metrics. For this, we suggest a set of mapping rules. Before applying the mapping rules, the data in relational database is cleansed. A cleaning algorithm has been devised for cleaning the name and email id fields.

II. LITERATURE REVIEW

[5] has proposed a tool that helps companies to create their own multi-dimensional model from a collection of relational databases, and to make a web-based environment supporting flexible views of the multidimensional model. This requires information from a user and extracting information about the tables, the attributes and the relations between them from the relational database. With this information it makes a new multidimensional database. If the relational database does not have this information, the tool cannot create the model. The model is semi-automatic in nature as the user has to select attributes depending upon what he wants to analyze. At the end there is a system in which it is possible to make queries for the decision support. The above thesis does not consider laying out explicit mapping rules for the classification of attributes into dimensional attributes and metrics in the multidimensional model. Our paper aims to classify attributes in a clear and unambiguous manner.

[6] presents MD, a logical model for OLAP systems, and shows how it can be used in the design of multidimensional databases. Unlike other models for multidimensional databases, MD is independent of any specific implementation. In MD facts and dimensions are abstract entities, described by mathematical functions. It follows that, in querying an MD database, there is no need to specify complex joins between fact and dimension tables, as it happens in a star schema. The methodology they propose for building an MD database starting from a pre-existing E-R scheme consists of four steps: Identification of facts and dimensions, Restructuring of the E-R scheme, Derivation of a dimensional graph and translation into the MD model. The above paper suggests a new logical approach to designing a multidimensional model but does not mention or improve upon the designing of the existing multidimensional models from the relational database.

[7] describes a method for developing data warehouse and data mart designs from an enterprise data model. The steps of the method are: Develop enterprise data model, design central data warehouse, classify entities in the central data warehouse model as transaction, component or classification entities, identify hierarchies which exist in the data model and design data marts. This approach supports the development of independent but consistent data warehouses based on a common enterprise data model. The above paper does not state explicit mapping rules for the creation of a multidimensional model from a relational database which have been addressed in our paper.

In [8], a feasible architecture for integrating UML data sources in order to build a multidimensional model for OLAP has been presented. The integration framework takes into account the benefits of UML (its concepts, relationships and extended features) which is closer to the real world and can model even the complex problems easily and accurately. The integration framework involves two steps. The first one is to convert UML schemas into UML class diagrams and the second is to build a multidimensional model from the UML class diagrams. Further, it clearly defines mapping rules that convert the semantics of UML Schemas into UML Class diagrams in order to extract the multidimensional information from multiple UML data sources. It also elucidates how a multidimensional model like star or snowflake schema can be represented using UML.

In [9] an approach has been introduced in which a set of minimal constraints and extensions to UML are used for representing multidimensional modeling properties for applications. It uses UML class diagram to specify the structure of a multidimensional model to design data warehouses. At the structural level, the process generates the star schema first which will enclose the multidimensional data. Then it produces the equivalent multidimensional information to be used in the theoretical design. Multidimensional data and the underlying multidimensional model also called the metadata, provides the model's key semantics, such as facts, measures, and dimensions.

[10] outlines the major steps for data transformation and data cleaning and emphasized the need to cover schema- and instance-related data transformations in an integrated manner. Further, they have classified data quality problems that are addressed by data cleaning in various data sources differentiating between single- and multi-source data source and between schema- and instance-level problems. Also, they have described the various approaches to data cleaning. They go on to discuss various commercial data cleaning, data analysis and re-engineering tools. Primarily, they have suggested that the discovery of data cleaning rules during the data warehouse design can advocate improvements to the constraints imposed by the existing schemas.

[11] has described an object oriented approach to model the process of data extraction as part of extraction, transformation and loading process. The hierarchies of each data element have been explicitly defined, thus highlighting the data granularity and hence simplifying the data extraction process. The object oriented features of generalization, aggregation, composition and association have been incorporated. These features help in identifying and establishing the relations between various data sources, thereby making the process of data extraction more reliable. Solving queries has been made easier because data sources at every level of granularity can be

identified and targeted directly.

In all the above mentioned approaches, no explicit rules have been stated to identify metrics and dimensional attributes from the attributes of the relational databases. This paper presents an approach to design a star schema from relational database by proposing simple and efficient mapping rules which allow the identification of metrics clearly and unambiguously.

III. THE PROPOSED FRAMEWORK

The ETL framework (Figure 1) so proposed is based on a user friendly approach. Data from all the relational databases is gathered and integrated at one place which we call the integrated relational database. Since the data comes from disparate relational databases, we need to have all the data in a consistent format. We propose the following cleaning algorithm for this purpose. This algorithm has been devised for the name field and the email field.

A. Data Cleaning

Data Cleaning is a process used to determine inaccurate, incomplete, or unreasonable records from a record set, table, or database and then improving the quality through correction of detected errors and omissions. Data cleaning is important for the efficiency of any data dependent organization. In any organization incorrect data can be costly. Incorrect or inconsistent data can lead to false conclusions and misdirected investments. The job of data cleaning is to ensure that the data within a system is correct, so that the management is able to use this data. [12]

In this paper, an approach for data cleaning is proposed. Here, two data fields namely - Name and Email are primarily cleaned. Formats are set for both the fields and they are cleaned on the basis of those formats only.

1) E-mail Address field cleaning:

The paper proposes a pattern matching algorithm for the cleaning of the E-mail id field. This algorithm works on the basis of comparison and string matching. All the e-mail ids are compared with the e-mail id format which is defined below. The pattern matching algorithm searches for irregularities in the e-mail ids and invalidates those e-mail ids which do not comply with the required format. In this way, the inconsistent and invalid database entries can be easily searched in the vast data pool and then can be removed later or fetched again from the corresponding sources.

The format of the e-mail field is:

[_a-z0-9-]+(\.[_a-z0-9-]+)*@[a-z0-9-]+(\.[a-z0-9-]+)*(\.[a-z]{2,3})

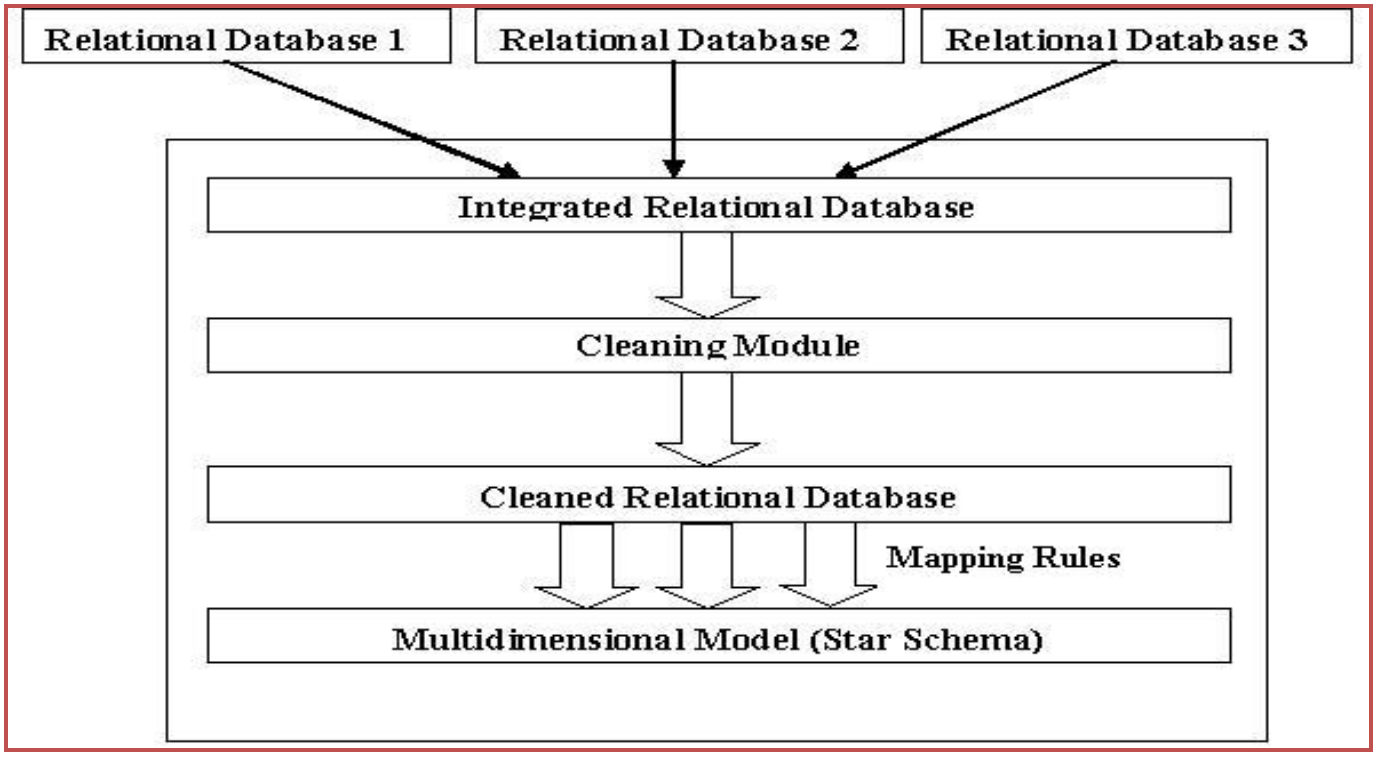

Figure 1. Proposed Framework

It basically implies that the user name in the email address can consist of alphabets, digits or some special characters followed by '@' sign which is compulsory. After the '@' sign, domain name is specified to which email would be sent. Domain name consists of at least one 'dot' and at most two 'dots' and after the 'dot', specified domain needs to have minimum of two characters. This pattern is to be matched with all the e-mail ids for validating them.

2) Name field cleaning:

The name field encompasses – salutation, first name, middle name, and last name. All these fields combine to form the name field. It is critical to clean this field because it may have been entered in different formats in different databases and tupples. Here we suggest an algorithm to clean all the names by grouping them under different categories based on the types of salutations. These categories are further divided into groups based on different possible formats for that particular category. Finally, all the names are cleansed and stored in one particular format. The salutations according to which names are categorized can be – Mr. , Ms. ,Dr., Prof., Mrs. and Miss. and each name under every category may be written in a different format due to which the category is further divided into groups.

E.g. Mr. Sheldon Cooper may be written in the following formats in the database table (groups under Mr.) –

Mr.sheldon cooper

Mr sheldon cooper

Mr. Sheldon cooper

sheldon cooper

There can be many more permutations and combinations of upper case and lower case characters with which the above name can be written. The algorithm would check for all the possible cases and convert the name into –

Mr. Sheldon Cooper (Proposed for the algorithm)

The format proposed for this algorithm is -

"Salutation. First name Middle name Last name"

This means that in the beginning, there should be salutation followed by a 'dot' and blank space. Then first name, middle name and last name should follow. The first letter of the first name, middle name and last name should be in uppercase and all other alphabets should be in lowercase.

The algorithm also considers the case where no salutation has been specified. In this case, the algorithm proposes to append the salutation in front of the name on the basis of the gender specified. The cleaning according to these formats has been implemented and the respective snapshots are shown in Figure 4 and Figure 5.

After cleaning, we get the cleansed relational database as the output. Now this integrated and cleansed relational database is to be transformed into a STAR SCHEMA. For the design of star schema, we make use of a set of mapping rules. These mapping rules have been explained in the next section. The basic objective of these mapping rules is to segregate the attributes of the relational database into dimensional attributes and metrics which form the star schema.

Mapping rules for designing a star schema from a database schema as input:

- A database schema is considered to have tables with textual as well as numerical attributes.
- These tables are considered as temporary dimension tables on which further operations need to be performed to yield required dimension tables.
- Classify all the attributes of these temporary dimension tables as textual and numerical attributes and also identify the primary keys of all the tables.
- Let the textual attributes remain in the temporary dimension tables and remove the numerical attributes.
- Create an intermediate fact table and insert the primary keys of all temporary dimension tables in it. These keys would act as foreign keys in this fact table.
- Set the primary key for the fact table which can either be a concatenation of all the primary keys of dimension tables or an auto-generated key.

[Now all the textual attributes along with the primary keys are present in the corresponding intermediate dimension tables and all the numerical values (whether they are metrics or not) and a set of foreign keys are present in the intermediate fact table.]

- For all the numerical attributes in the intermediate fact table perform the following steps (Figure 2):
  - If an attribute is a derived attribute, it is not to be put in final_fact table. E.g. Age becomes a derived attribute in presence of DOB (Date of Birth).
  - If there exists a client, stakeholder or employee table in the database then attributes directly related to them are not to be put in the final_fact table. E.g. income_level of customer is directly related to customer and hence is not a fact.
  - If there exists a degenerate dimension, it is to be put in the final_fact table. E.g. order_no in case of a Sales department having order_id as primary key is a degenerate dimension.
  - If there exists an attribute which can be analyzed along the time dimension, it is to be put in the final_fact table.
- The attributes finally existing in final_fact table after the above set of eliminations and selections performed, are the required metrics to be used in query processing and the non metrics (integer values remaining in intermediate fact table) are filtered out in a way that they move to their respective dimension tables and exist as dimensional attributes.
- This step results in the conversion of the intermediate fact table into the final_fact table and temporary dimension tables to final_dimension tables.

The next step involves creation of required fact table with all the metrics present in final_fact table as its columns, the foreign keys referring to the dimension tables and a primary key. This follows the creation of required dimension tables having a primary key and the columns as the dimensional attributes present in final_dimension tables . The logical structure of a star schema implies a

one-to-many mapping from the dimension tables to the fact table with no relationship between the dimension tables. Our framework efficiently lays out the process of identifying the relations between various the tables and modifying them according to the logical structure of the star schema. To implement this structure, we specifically take care to create the fact table's foreign keys corresponding to the dimension tables' primary keys and remove the primary key – foreign key relationship from the final_dimension tables while transforming them into the required dimension tables (Figure 3). Thus a relational database schema is converted into a multi dimensional model (star schema).

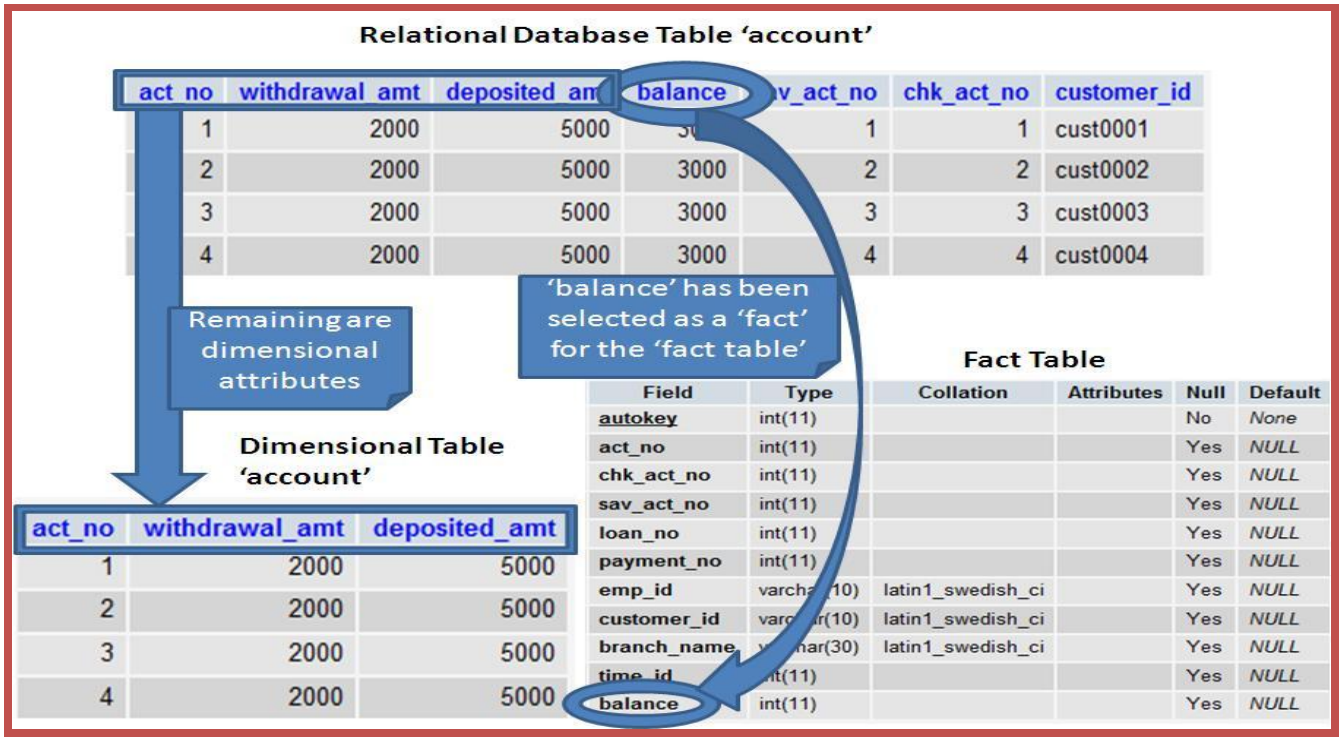

Figure 2. Selecting a fact on basis of proposed mapping rules

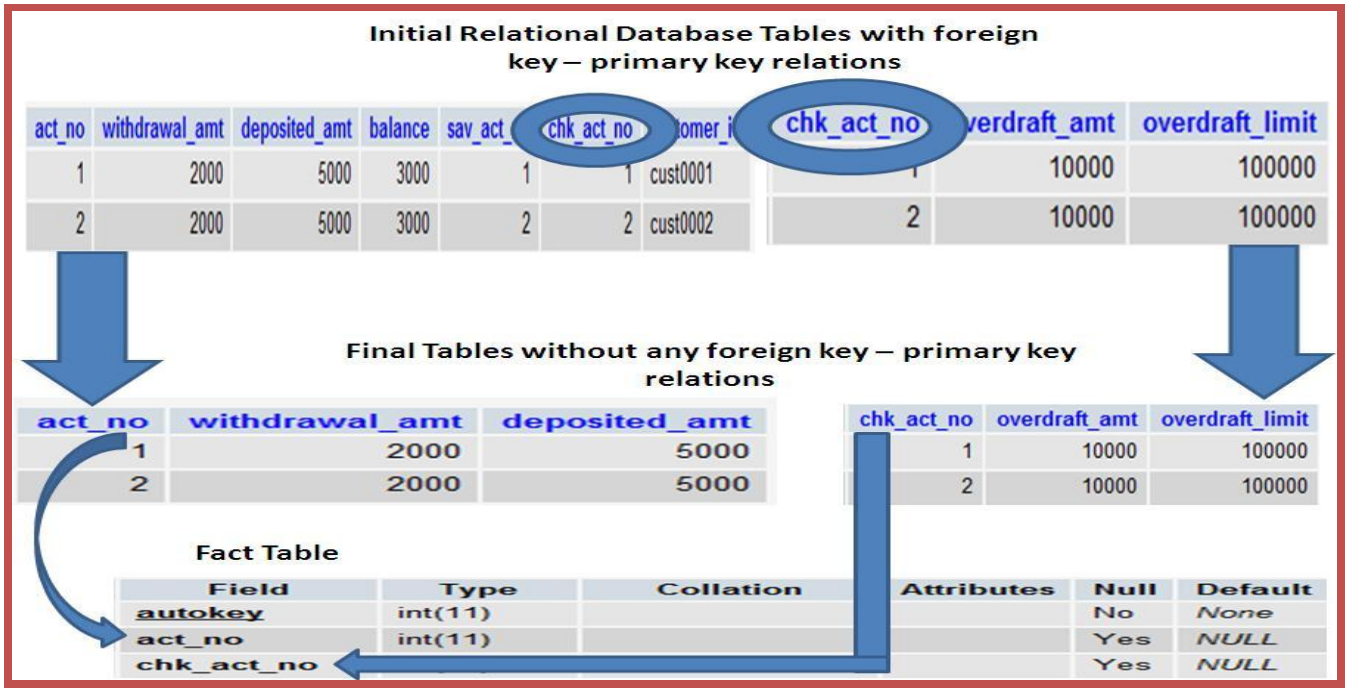

Figure 3. Removing foreign key – primary key relations between dimension tables.

Some advantages of the above framework are:

- The mapping rules are generic and have been implemented and found successful for different examples/case studies.
- One can find facts very easily and effectively with the help of the mapping rules.
- Since, cleaning is done before the mapping procedure, so the data in the data warehouse will be consistent, complete, accurate format.

These mapping rules strictly follow the basics of designing dimension tables and fact table. Here, metrics are checked and selected according to the different properties of the fact table. The above mentioned mapping rules have been successfully implemented and the system so designed would find its application in small and medium sized organizations, which do not have the resources to employ a separate staff for data warehouse designing.

IV. CASE STUDY: BANK MANAGEMENT SYSTEM

For explaining our concept we have used the case study of Bank Management System in this paper. UML class diagrams have been used to depict the database schema of the Bank Management System (Figure 6).

Using the cleaning methodology described above, first the customer_name and the email_id fields of the customer table in the database of the Bank Management System have been cleansed and displayed in a uniform format. Then, using by applying the mapping rules, the relational database schema is transformed into a star schema for the Bank.

A. Implementation

The system has been implemented using the programming language PHP and MySQL as backend. The frontend has been designed in HTML, XML and JavaScript. It is a semi-automatic system which requires user interaction at different stages.

1) Data Cleaning:

Here, a database named 'bank_mgmt_sys' is considered having a table named 'customer' with fields named 'customer_name', 'email_id' and 'gender' among others.

a) Name Cleaning: The system first fetches a particular name from the table and then compares it with the above mentioned formats. When it gets a match it performs the corresponding set of operations. It stores the full name in a text file and then fetches the first character of first name, middle name and last name and stores them in different variables and converts them into uppercase. Similarly, it converts the rest of the characters of the name into lowercase. If the name in the table initially contained a salutation, then that salutation is cleansed to the correct format and is appended in front of the name. Else, a salutation is decided on the basis of gender and appended. Finally, the corrected name replaces the earlier corresponding entry in the table.

The above described implementation has been applied for all categories of name cleaning.

b) Email ID Cleaning: A pattern matching algorithm has been implemented for email id cleaning. The entry in the table is verified against the email format (as described above) which is stored in a variable. The email id which is found to be inconsistent with the format is considered invalid and that particular entry in the table is replaced by "invalid 'email_id' address" note. The valid email ids are kept unchanged inside the table.

2) Mapping from relational database schema to star schema:

The input to the system is a relational database schema of a Bank Management System.

A generic approach has been devised for implementing the above stated mapping rules. The system requires only the database name as input in order to retrieve the table-names and their column-names, primary keys and data types. This information is retrieved from the 'INFORMATION SCHEMA' database of MySQL.Information_schema stores the metadata about all the relational databases created in Mysql. All the column-names fulfilling the mapping rules as mentioned earlier are selected as facts for the fact table and those which do not fulfill the mapping rules criteria are considered to be the part of dimension tables. These dimension tables are created with respect to the earlier relational database tables. Now the relationship between various dimension tables is removed and all dimensional tables are linked with the fact table with the help of primary key- foreign key relation. A new time dimension table is created and is linked with the fact table.

The framework facilitates the design of a multidimensional model (star schema) (Figure 7) from the relational database schema. A person need not have sound knowledge of database querying and operations to efficiently use this system, although he should be well-versed with the kind of attributes he needs to select for the fact table.

However, the system has certain limitations as it has not taken weak entity set and relation into account.

customer_id	street	city	customer_name	dob	age	email_id	gender	act_no
cust0001	lutyens	delhi	Mrs. Sonia gandhi	1970-08-21 00:00:00	41	sg2011@gmail.com	female	1
cust0002	lutyens	delhi	Mrs.Indira Gandhi	1973-09-23 00:00:00	38	ig2011@gmail.com	female	2
cust0003	vasant vihar	delhi	dr. Menaka Kumari	1964-08-24 00:00:00	45	mg2011@gmail.com	female	3
cust0004	rohini	delhi	mr.Rakesh kumar Sharma	1987-11-01 00:00:00	23	rg2011gmail.com	male	4
cust0005	vasant vihar	delhi	Mr. Varun Kumar Jain	1950-08-10 00:00:00	61	vg2011gmail.com	male	5
cust0006	greater kailash	delhi	Miss. vandana Kumari Luthra	1980-08-19 00:00:00	31	ng2011@gmail.com	female	6
cust0007	greater kailash	delhi	Prof.Raj Mittal	1960-08-02 00:00:00	51	jg2011@gmail.com	male	7
cust0008	chandni chowk	delhi	Mr. Rahman chaudhary	1961-08-20 00:00:00	50	hg2011@gmail.com	male	8
cust0009	lutyens	delhi	mr. rahul Kumar aggarwal	1976-08-15 00:00:00	35	ug2011@gmailcom	male	9
cust0010	lutyens	delhi	ms. Priyanka Gupta	1965-10-31 00:00:00	46	pg2011@gmail.com	female	10

Figure 4. Table with erroneous entries

customer_id	street	city	customer_name	dob	age	email_id	gender	act_no
cust0001	lutyens	delhi	Mrs. Sonia Gandhi	1970-08-21 00:00:00	41	sg2011@gmail.com	female	1
cust0002	lutyens	delhi	Mrs. Indira Gandhi	1973-09-23 00:00:00	38	ig2011@gmail.com	female	2
cust0003	vasant vihar	delhi	Dr. Menaka Kumari	1964-08-24 00:00:00	45	mg2011@gmail.com	female	3
cust0004	rohini	delhi	Mr. Rakesh Kumar Sharma	1987-11-01 00:00:00	23	invalid email_id address	male	4
cust0005	vasant vihar	delhi	Mr. Varun Kumar Jain	1950-08-10 00:00:00	61	invalid email_id address	male	5
cust0006	greater kailash	delhi	Miss. Vandana Kumari Luthra	1980-08-19 00:00:00	31	ng2011@gmail.com	female	6
cust0007	greater kailash	delhi	Prof. Raj Mittal	1960-08-02 00:00:00	51	jg2011@gmail.com	male	7
cust0008	chandni chowk	delhi	Mr. Rahman Chaudhary	1961-08-20 00:00:00	50	hg2011@gmail.com	male	8
cust0009	lutyens	delhi	Mr. Rahul Kumar Aggarwal	1976-08-15 00:00:00	35	invalid email_id address	male	9
cust0010	lutyens	delhi	Ms. Priyanka Gupta	1965-10-31 00:00:00	46	pg2011@gmail.com	female	10

Figure 5. Table with correct entries

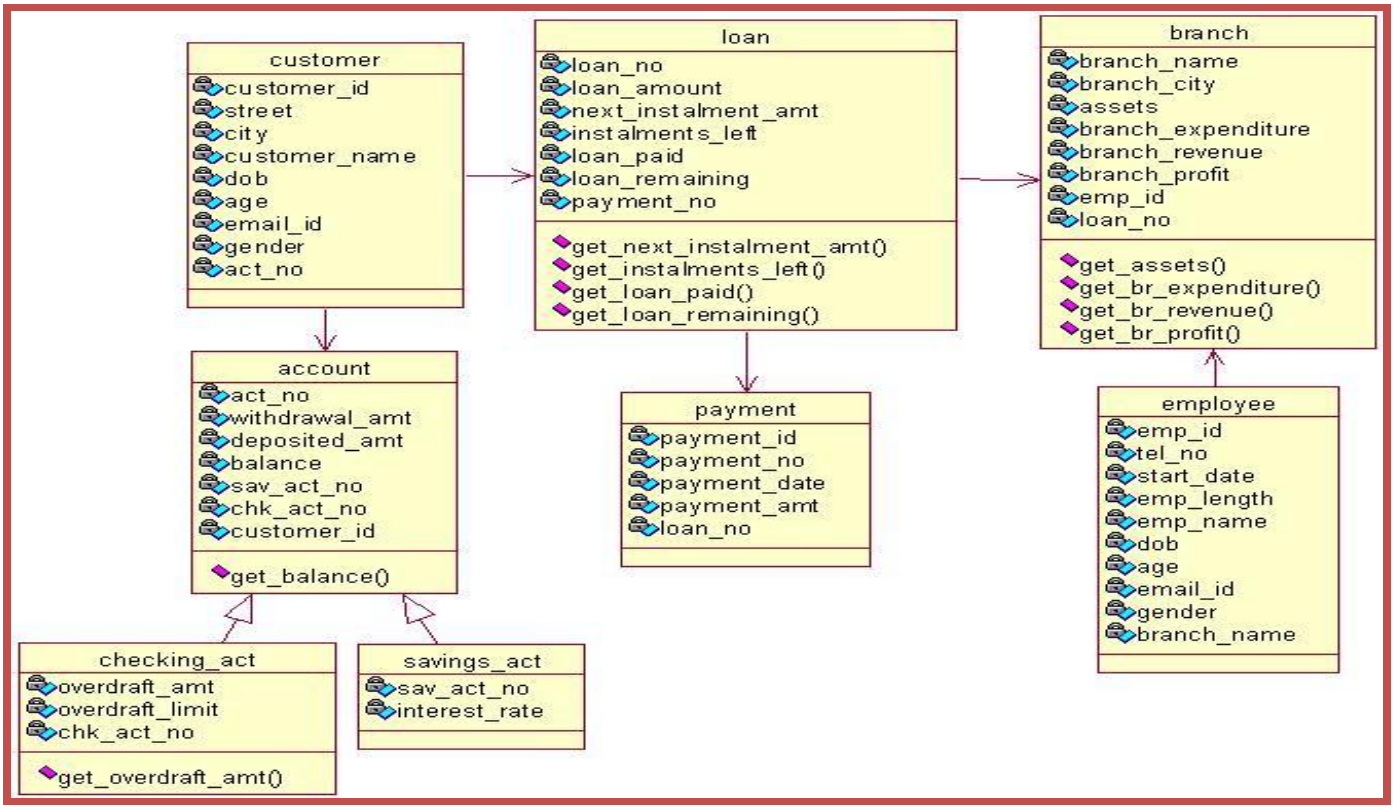

Figure 6. Bank Management System Class Diagram

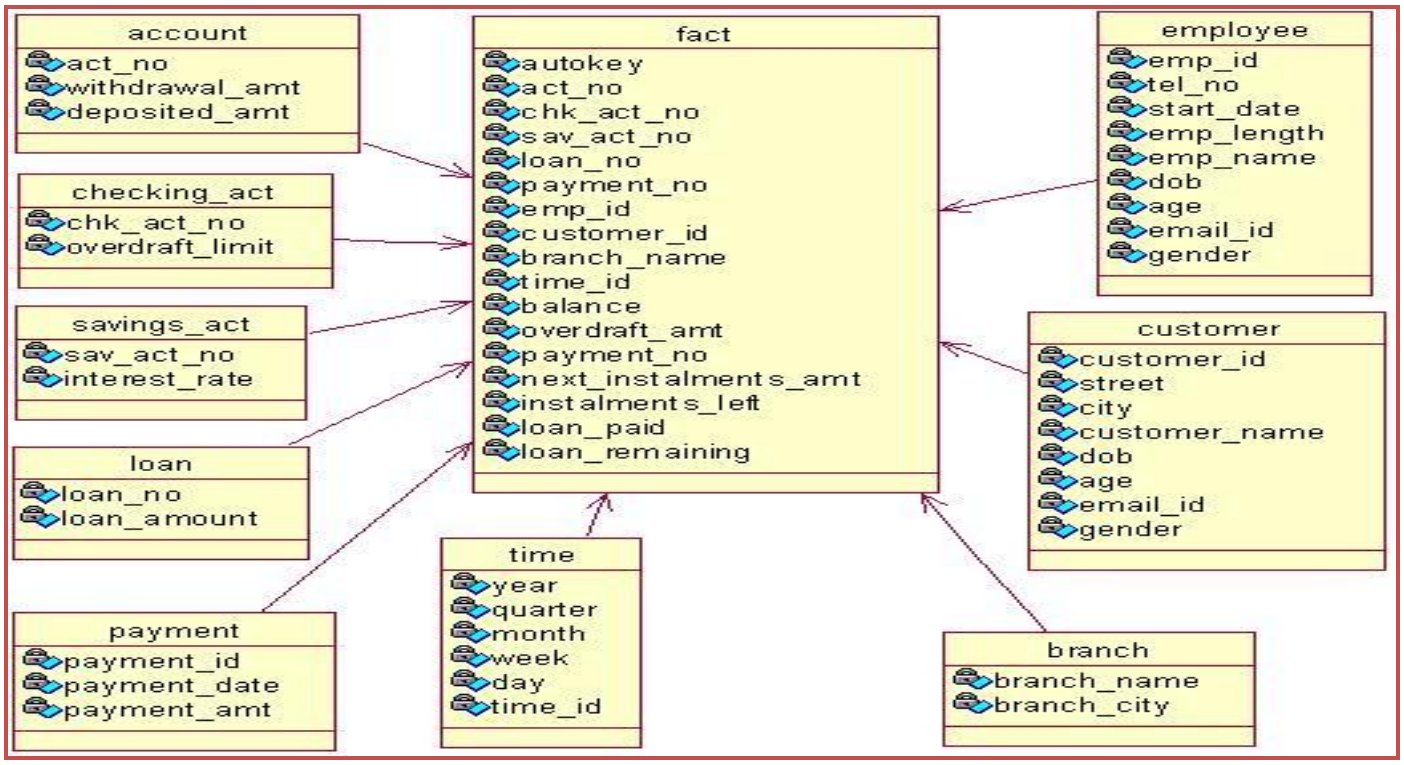

Figure 7. Star Schema

V. CONCLUSION

The proposed methodology for designing a multidimensional model from the relational database schema by categorizing the relational database attributes into dimensional attributes and metrics is unique and efficient in nature. A cleaning algorithm has been proposed to perform the cleaning of names and email id fields using suggested standard format. The design of a data warehouse from relational databases is made possible in a user-friendly and semi-automated manner. The mapping of relational database to multidimensional model has been done by practically implementable mapping rules. These mapping rules are successful at conceptual as well as practical level mapping and are easy to understand. The approach followed is generic and simplified in nature. This approach has been examined using the case study of bank management system and has been realized successfully using WAMP 2.0 (PHP 5.3.0 and MySQL 5.1.36).

ACKNOWLEDGMENT

We would like to thank our Chairman Sir Shri A.P. Kaushik, our General Secretary Sir Shri Vinod Vats and our Secretary Sir Shri B.N. Sharma, Bhagwan Parshuram Institute of Technology, G.G.S.I.P.U., Delhi, India for their valuable support and encouragement.

REFERENCES

[1] Elmasri, R., Navathe, S.B.: Fundamentals of Database Systems. Addison Weasely Pub Co. ISBN 0201542633 (2000).

[2] Paulraj Ponniah, Data Warehousing Fundamentals, Wiley India Pvt. Ltd., Reprint 2008.

[3] Inmon, W.H., Building the Data Warehouse (2nd Edition), New York: Wiley, 1996.

[4] Wikipedia, http://en.wikipedia.org.

[5] María, A. and Castillo, G., 'Semi-Automatic Support for the Design of Multi-Dimensional Databases', Kongens Lyngby, 2007, www.imm.dtu.dk, ISSN 0909-3192, Chapter 3.

[6] Cabibbo, L. and Torlone, R., 'A Logical Approach to Multidimensional Databases'

[7] Moody, D.L., Kortink, M.A.R. "From Enterprise Models to Dimensional Models : A Methodology for Data Warehouse and Data Mart Design", 2nd International Workshop on Design and Management of Data Warehouses (DMDW 2000), Stockholm,Sweden in June 2000.

[8] Pahwa, P., Taneja, S. and Jain, S., Design of a Multidimensional Model Using Object Oriented Features in UML. UTIT, International Conference on Upcoming Trends in IT, (Mrach 26, 2011) at PCTE Ludhiana, Punjab, India.

[9] Trujillo, J., Palomar, M., Gomez, J. and Song, I.L., Designing DataWarehouses with OO Conceptual Models. Computer, 0018-9162/01/$17.00 © 2001 IEEE.

[10] Hang-Hai, D., & Erhard, R.: Data Cleaning: Problems & Current Approaches. IEEE bulletin of the technical committee on Data Engineering, 24, 4 (2000).

[11] Pahwa, P., Chaudhary, G., Jain, K., Sharma, N. and Gupta, R., 'Hierarchical Approach to Data Extraction using UML 2.0', Proc. of the International Conference on Advanced Computing and Communication Technologies (ACCT 2011), Copyright © 2011 RG Education Society, ISBN: 978-981-08-7932-7.

[12] http://www.ifonlyihadit.com/data-cleaning-made-simple.php.htm