GENERATING DATA WAREHOUSE SCHEMA

Reyan Abdalaziz Ahmedl and Tarig Mohamed Ahmed

1Department of Computer Science, University of Khartoum, Khartoum, Sudan
2Faculty of Mathematical Sciences, University of Khartoum, Khartoum, Sudan

ABSTRACT

A data warehouse is an integrated set of data, derived basically from operational data to use in decision making strategy and business intelligence using (OLAP) techniques. Most of the creation of data multidimensional data warehouses is done manually, but it is a very complex and takes a long time and usually has a risk of fail. In addition, a set of complex mappings must be performed. Despite this, there is no noticeable efforts has been done in order to find a practical solution structured to resolve the issue. To overcome, the proposed method presents a new strategy to automate the multidimensional design of Data Warehouses; this algorithm uses an enterprise schema of an operational database as a starting point to design data warehouse schema. As a result, the user has to choose the candidate schema which meets the system requirements.

KEYWORDS

Data warehouse, Automation, Design, Schema, Generating, OLTP.

1. DATA WAREHOUSE

Data warehousing is now playing a significant role in strategic decision making. It is gaining importance day by day in enterprises. It provides a multidimensional view of huge amounts of historical data from operational sources, thus supplying useful information for decision makers to improve their business intelligence which has become an integral part of decision making strategy [14].

It is a collection of integrated, subject-oriented databases designed to support the DSS function, where each unit of data is non-volatile and relevant to some moment in time.

There are two forms for data management. The operational databases are where the data is put in. Users of this type almost deal with one record at a time and they usually perform the same tasks. The data warehouse is where we get the data out. Users of this type almost deal with set of row at a time; their questions require that thousands of rows be fetched into an answer set [13], [19].

Data warehouse is (a) Subject-Oriented means that the main objective of data warehouse is to facilitate decision process of a company, and within any company data naturally concentrates around subject areas, so information gathering in data warehouse is aiming for a specific subject rather than for the functions of a company. (b) Integrated Being integrated means that the data is collected within the data warehouse, that can come from different tables, databases or even
servers, but can be combined into one unit that is relevant and logical for convenience of making strategic decision.  
(c) Non-volatile being the snapshot of operational data on a given specific time, the data in the data warehouses should be stable. The data in the data warehouse usually be added, but it should rarely be deleted.  
(d) Time-variant Time-variant means that all the data within the data warehouse can be found with a given period of time [14], [18].

The main objectives of data warehouse are:

Data availability: Data is structured to be ready and available for analytical processing activities such as OLAP, data mining, querying, reporting and any other decision supporting applications.  
Easily accessible: Data warehouse content must be understandable and labelled meaningfully. It must also return query results with minimal wait times.  
Consistently: Warehouse must be credible and its data must be collected from a variety of sources in a relation, cleansed and quality checked.  
Adaptive to change: The data warehouse must be able to handle changes. The existing data and applications should not be affected with changes or asking new questions or adding new data to the warehouse.  
Security and protection: The data warehouse must control access to the confidential information.  
Improve decision making: The data warehouse must have a trusted data to support decision making.

2. DATA WAREHOUSE’S COMPONENTS

2.1. Operational Source Systems

The operational systems manage the business transactions. They use queries with one-record-at-a-time. The main objectives are: processing performance and availability.

2.2. Operational Data Store

They are frequently updated. When OLTP systems give insufficient operational reports, ODS is used to generate them. These reports are specified by a limited set of queries that can be used in a reporting application [16].

2.3. Data Staging Area

It is storage area and also it is a set of processes of extract-transformation-load (ETL). The data staging area is everything between the operational source systems and the data presentation area. It transforms the operational data into a deliverable for user query [18].

2.4. Data Presentation

The data presentation area is where data is organized, stored, and become available for users querying, reporting, and any analytical applications, it is just like a series of integrated data marts.
Data in the presentation area of the data warehouse must be atomic and adhere to the data warehouse bus architecture. It must be also dimensional, sense that it is applicable to both relational and multidimensional databases [17], [18].

2.5. Data Access Tools

A data access tool may be simple such as an ad hoc query tool which is powerful, understandable and effectively used by a small percentage of the data warehouse business users. It may also be complex like a data mining or modelling application which uploads their results back into operational source systems or the staging/presentation areas of the data warehouse [11], [20].

3. Building Data warehouse

The major steps of building data warehouse are described as (a) Data collection which is the first step before starting to build data warehouse, the data source will be identified. There is a need to figure out what are the data that are required to be put into the data warehouse. The owners of source data are responsible for maintaining quality, and this may require substantial effort. (b) Transformation & cleansing, this can be the most time consuming part where needed to grab the data from data source and store it into the staging database. In this process, data are usually restructured to optimize subsequent use for querying, reporting and analysis. Task of this stage is very hard and time-consuming, and usually can be done with the help of ETL tools. (c) Aggregation & analysis Selected data are taken from the central warehouse using query tools and processed to produce useful results. Often, the most frequently accessed data are first summarized and stored in data marts to improve response times. Additional performance measures are typically derived at the same time. Analytic applications may also be developed to help users get useful information. (d) Presentation is displaying results for end users, usually in the form of reports. Several different report types are normally needed to suit different types of user. The results might appear as text, tables or charts and could be viewed on-line, printed, published on a web server or distributed by email [15], [19], [24].

4. Data warehouse Dimensional modelling

There are three main concepts of the dimensional modelling [20]:

1- The fact is a set of related data, contains analytical context data and measures. It used to represents business items or business transactions.
2- The measure is a numeric attribute in the fact table which illustrates the behaviour of the business of the dimensions.
3- The dimension is a set of contextual data which describe one business dimension. It determines the referenced background for the facts; they are the identifiers over which we want to analyze.

In the data warehouse environment, the relational model must be transformed to other architectures [15], [13]. The determination of the suitable model must be based on the requirements, access tools and the team preferences. There are many models designed for data warehousing, but the most common are:
4.1. Star Schema

This model architecture is the simplest data warehouse schema; it is most commonly used nowadays. It is a central fact table is surrounded by dimensional tables; each one contains information about the entries for a particular attribute in the fact table [21]. Usually the fact tables in a star schema are in third normal form, whereas dimensional tables are de-normalized [23], [19]. The main characteristics:

1- Simple structure, easy to understand schema.
2- Great query effectives, small number of tables to join.
3- The direct mapping between the business entities, are being analysed by end users and the design.
4- Provide highly optimized performance for star queries.
5- Supported by a large number of business intelligence tools.

4.2. Snowflake schema

The snowflake schema architecture is a more complex, because the dimensional tables are normalized. It is an enhancement of star schema. It normalizes dimensions to eliminate redundancy [14], [15]. The dimension data has been grouped into multiple tables instead of one large table to saves space, but this lead to more complex queries and reduced query performance. It is useful when there are low cardinality attributes in the dimensions.

4.3. Fact Constellation Schema

The fact constellation architecture contains multiple fact tables that share many dimension tables. It is possible to construct fact constellation schema by splitting the original star schema into more star schemes each of them describes facts on another level of dimension hierarchies [14], [18]. The dimensions in this schema are large. They must be split into independent dimensions based on the levels of hierarchy. It is used mainly for the aggregate fact tables and for better understands.

4.4. Galaxy Schema

Galaxy schema contains many fact tables with some common dimensions (conformed dimensions) [14], [17].

This schema is a combination of many data marts. This type is not commonly used.

5. Data warehouse Designing Automation

Data Warehouse design defines the elements and structures which ensures the ability to access information. It is important to take into account the information requirements, the source databases to builds relational or multidimensional structures [13].

This paper aims to present a strategy to generate the Data Warehouse schema from OLTP. Also to generate mappings between them which are very useful to program the data warehouse loading
and refreshment processes. This strategy aims to help the data warehouse designer by automating some tasks which reduces manual designing time. This should be performed using an application of transformation rules [19], [26]. At this stage of the proposed strategy establishment, the work will focus on the single source OLTP and multisource horizontal fragmented database.

6. Related Works

Recently, the Data warehousing became an important technique in enterprises because it improves their business intelligence.

There are many efforts have been done in the automation of creating data warehouse which maximize accuracy, minimize dependability on users and speed up transactions. A survey of some works will be reviewed in this section.

One approach to conceptual warehouse design is described in [1]. It shows how to systematically derive a conceptual warehouse schema that is even in generalized multidimensional normal form. The generalized multidimensional normal form can be considered as a quality factor which avoids aggregation anomalies which leads to correct query results. It also sorts out dimensions, dimensional hierarchies, and measures, and determines database attributes mapping.

The method first determines whether an attribute is a dimension level or a property then captures the result using graphical formalism, finally show how to obtain generalized multidimensional normal form (GMNF). The processes of conceptual modelling are three steps, Context definition of measures, Dimensional hierarchy design and Definition of summarized constraints.

Another algorithm is proposed for automatic conceptual schema development using an operational database to derive data warehouse schema design [2]. It used ME/R (Multidimensional Entity-Relationship) model which is simple for users to read, and aggregation. Also another evaluation algorithm is created to support queries of user needs. Both of two algorithms are using the TPC-H Benchmark schema and queries.

Input to the algorithm is an ER schema represented in table data structures and the outputs a set of ME/R schemas in a tabular form.

The first step of the algorithm is to find entities with numeric fields and create a fact node for each entity identified. Then create numeric attributes of each fact node, date/time dimensions and create a dimension containing the remaining entity attributes. Final step is to examine examines the relationships of the entities to add additional levels in a hierarchy.

Also there is a mechanism to generate structured dimensional model and data marts from Entity Relationship model addressed in [3] with design options. Use of this mechanism supports the development of independent and consistent data warehouses based on a common enterprise data model.

The steps of the method are:

1- Develop Enterprise Data Model.
2- Design Central Data Warehouse.
3- Classify Entities as transaction, component or classification entities.
4- Identify Hierarchies which exist in the data model.
5- Produce dimensional model.
6- Design Data Marts.
7- Evaluation and refinement.

A hybrid approach integrates information needed by decision makers with existing data source based on the Model Driven Architecture (MDA) [4]. This framework reduces the Complexity and time which are important features. All pieces are prepared in the framework using integration of Model Driven Development (MDD). In the same time a set of transformations are done to get the final version of the MD model automatically.

Also there is a method to obtain information requirements model (Computation Independent Model, CIM) from decision makers. This CIM can be used to develop a conceptual multidimensional model of the data warehouse (Platform Independent Model, PIM). This initial PIM can be consisted with the data sources to generate a hybrid PIM. As a result, a lot of logical models can be generated from this hybrid PIM such as Platform Specific Models (PSMs), and bearing in mind different deployment platforms.

The hybrid technique presented in [5] automates entity relationship model into data warehouse logical model to generate semi-star schema using artificial neural networks. It used feed forward back-propagation neural networks to automate the step of differentiating dynamic from static behaviour dimensions. The proposed technique is then compared the existing technique, with that do not involve any intelligence. A comparison is based on simplicity and minimality metrics. Results show that this technique is comparable to the existing ones, and in some cases, it gives better results.

In Semi-Star (SS) modelling schema, the OLTP is considered as a source of analytical processing. SS schema reduces the cost of ETL process. In the same time, it saves the cost of creating and another analytical source. ERD to SS Schema Conversion Steps:

Normalizing ERD: The input ERDs are converted into binary ERDs.
Differentiating Entities: The entities need to be classified as transactional, classical or component entities.
Adding Fact Table: Fact table is connected to all the dimensions except Shared Dimension Tables (SDTs).
Joining Dimensions to Fact Table: The dimensions added to the fact table have all their relations attached to them.
Separating SBDs from DBDs: Dimensions are categorized as either SBDs or DBDs. SBDs.

A new methodology for designing multidimensional scheme, derived from the source of XML Schemas and XML documents in addition to testing tool to verify this methodology [6]. In the conceptual design, a dependency graph is created then dimensions and measures are defined. In the logical design, a star schema is created using the conceptual design automatically.

The XML Schema is the start point of the Conceptual design. At the beginning, a schema graph (SG) which describes the structure of XML data is created. XML elements and attributes defined in XML Schema are represented as vertices of the SG. A semi-automatically dependency graph is
restructured, and then dimensions and measures are determined. In the process of logical design, the resulting conceptual schema is simply transformed into a star schema.

In [7] is same as [6], but this one asks interactively questions during the tree building phase to detect -to-one relationships, and to avoid trivial queries.

A semi-automatic approach for building the conceptual schema and a data mart starting from the XML sources is addressed in [7]. On the basis of the supposition that the XML document has a DTD, sub-elements of DTDs are chosen to represent DTD graph, which is used later to deduce the facts. The methodology steps are as follow:

1- Simplifying the DTD.
2- Creating a DTD graph.
3- Choosing facts.
4- For each fact:
   Building the attribute tree from the DTD graph.
   Rearranging the attribute tree.
   Defining dimensions and measures

A semi-automated method can detect patterns of dimensionality [8]. It aimed to find the business multidimensional concepts from an ontology domain. It represents different and potentially heterogeneous data sources of business domain, with nothing in common but they are all described by ontology.

In each step this method automatically looks for a certain multidimensional concept. The first task is identifying facts; each fact will lead to a multidimensional schema. The second task is to determine sets of concepts to be used as base for each fact identified in the previous task. These set of bases are labelled as dimensions. The third task is to produce dimension hierarchies. Every dimension's hierarchy has been assorted with Levels related to it.

7. Proposed Solution

A typical operational database consists of sets of tables linked by a complex web of relationships. Even simple queries may require multi-table joins, which are error prone.

Basically the dimensional modelling has two main objectives the primary is to produce database schemas that are simple in structures and easy for querying. A secondary is to maximize the efficiency of queries [13], [18].

These objectives can be achieved by using less tables and relationships, which reduces the complexity of the database and minimizes the number of joins in queries.

In this work an environment has been proposed, where the DW schema is built automatically from the source schema through an application with predefined transformation rules.
The main components are:

7.1. **Schema Descriptor**

It is an intermediate database which has been designed using SqlServer; it serves as the data dictionary which is the metadata that contains full description about the data sources, target data warehouse databases in addition to the processes required to cleanse, transform, aggregate, and maintain the data.

It contains many tables as follow:

**SchemasTable**: this table is used to store name and description for any schema will be included in the warehouse schema.

**EntitiesTable**: this table contains name and descriptions of source database table and show to which schema it belongs to.

**FieldsTable**: this table contains all fields’ data, data type, default value and to which table each field belongs. It also determines if it is primary key or not.

**RelationsTable**: this table describes the relationships between the tables and the constraints of primary keys between them.

**CandidateSchemasTable**: every candidate schema should be stored in this table.

**WHEntitiesTable**: this table contains name and descriptions of data-warehouse table and show to which schema it belongs to.

**WHMappedFieldsTable**: this table contains all fields’ data, data type, default value and to which table each field belongs. It also determines if it is primary key or not.

**WHRelationsTable**: this table describes the relationships between the tables and the constraints of primary keys between them.

7.2. **Schema Translator**

It is a subsystem which connects to the schema and read it, then stores this translation in to the “Schema Descriptor”. Then it can be browsed or selected to view its table’s definitions and their contents, its constraints tickets and then represent them in a hierarchy to show its dependencies and levels.
7.3. Manage_Schema

It is a subsystem which analyses the stored schema then Normalize/Denormalize it according to the data warehouse model determined.

Merge-Split-Change-Eliminate-Unify-Join

7.4. Warehouse_Designer

It is a subsystem which analyses and applies some processes in the managed schema. Dimensions-Time Dimensions-Fact Table –Measures-Joining Dimensions to Fact

7.5. Draw_Model

It is a subsystem which describes the candidate model in a form of database diagram and represents it using some of the functionalities of MS Visio in addition to drawing libraries.

7.6. Mapping_Table

It is a subsystem which defines the rules of mapping which has a direct affect of the transformation process and the load process.

Data Transformation is the operation which concerns with the mapping between the given data format and the expected target format. It also concerns with the schema matching and integration [14].

Data loading is the operation which apply filtering and aggregation. It ensures that every value would be stored in its right location according to mapping rules [17].

Mapping records include:

1- Identification of original source
2- Attribute conversions
3- Physical characteristic conversions
4- Encoding/reference table conversions
5- Naming changes
6- Key changes
7- Values of default attributes
8- Logic to choose from multiple sources
9- Algorithmic changes

8. The Proposed Algorithm

8.1. Measures

The first step is defining measures, the numerical attributes within transaction entities that can be aggregated by dimensions attributes [14].
Choose all fields which can represent the measures, they must meet the following conditions:
Numeric fields, of type (int, bigint, smallint, tinyint, decimal, numeric, money, smallmoney, float, and real).
Non-key fields, neither primary key nor foreign key.

**8.2. Fact Tables**

The next step is to define the transaction entities which are record details about particular events that occur in the business [18]. They are the most important entities in a data warehouse, and form the basis for constructing fact tables in star schemas [16].

Always, they are events that decision makers want to understand and analyse. The key characteristics of a transaction entity are:

- It describes an event that happens at a point in time.
- It contains measurements or quantities that may be summarized.
- The next step is to define all candidate fact tables, So that each of them contains one or more of the previously determined measures.

For each candidate fact table, it would be considered as a centre of a candidate schema, therefore apply next steps to obtain a complete structure of this schema.

Determine the fact table fields:

- Select primary key of this table which is numeric and not foreign key.
- Select the measures that belong to this fact table.
- Select the fields of type date, time which belong to this fact table.

**8.3. Dimension Tables:**

Select all remaining fields of this table to create the first dimension of this schema.
Select all tables that in relations with this table to use them in the other dimension tables’ representation.

For each dimension table:

- Define hierarchy of dependencies for this table which is any sequence of entities joined together by one-to-many relationships, all aligned in the same direction.
- Collapse this hierarchy by integrating them to a de-normalized dimension table [12].
- Each dimension table has only one primary key, this primary key must be added as a foreign key in the fact table to link it.
- Create the last dimension which is date time dimension the most important for granularity, and then link it with the fact table in date time fields [15], [17].

**8.4. Storage:**

All these candidate schemas should be stored in the intermediate database to be used later.
9. Test and Discussion

TPCH schema is an industry benchmark example. It is an OLTP schema in a normalized format used as input in this application. It is well suited to a transaction processing environment. It contains no redundancy cause of the accurate web of relations that it contains, but even simple query requires multi-table joins and complex sub queries.

Most decision makers would find that is very complex schema to query, so there is a need for technical specialists to write queries for them [9]. This database is translated and stored in the intermediate database.

By selecting the database then select warehouse Schema choice in the menu strip. This results a range of sub nodes of this database node, each one is a candidate schema generated by the application can be renamed. By selecting any of the generated schemas, a complete data dictionary will be displayed.
The process of generating the result schemas is described as follows. Select the first fact table.

Fact Table: LineItemFact
Measures: Qty- Tax- Discount -ExtendedPrice
Primary Key: LineNumber-PartKey-SuppKey-OrderKey

DateTime Dimension
CommitDate- ReceiptDate – ShipDate

LineItemDimen: Remained Fields
LineNumber-ShipinStruct-LineStatus-Shiphome-Comment

DateTime Dimension:
DateValue-YearValue-MonthValue-Week-Day

Dimension Hierarchies
PartSuppDimen:
PartSupp – Part
PartSupp –Supplier –Nation-Region
OrderDimen:
Order-Customer-Nation-Region

After collapsing the dimension hierarchies by Denormalization:

<table>
<thead>
<tr>
<th>PartSuppDimen</th>
<th>OrderDimen</th>
</tr>
</thead>
<tbody>
<tr>
<td>SuppKey</td>
<td>OrderKey</td>
</tr>
<tr>
<td>PartKey</td>
<td>OrderDate</td>
</tr>
<tr>
<td>SupplierName</td>
<td>CustomerName</td>
</tr>
<tr>
<td>PartComment</td>
<td>Address</td>
</tr>
<tr>
<td>Phone</td>
<td>NaComment</td>
</tr>
<tr>
<td>Container</td>
<td>OrderStatus</td>
</tr>
<tr>
<td>RetailPrice</td>
<td>RegionName</td>
</tr>
<tr>
<td>SuppComment</td>
<td>NationName</td>
</tr>
<tr>
<td>PartName</td>
<td>Comment</td>
</tr>
<tr>
<td>AvailQty</td>
<td>RegComment</td>
</tr>
<tr>
<td>MFGR</td>
<td>OrderPriority</td>
</tr>
<tr>
<td>Size</td>
<td>MKTSegment</td>
</tr>
<tr>
<td>NaComment</td>
<td>ShipPriority</td>
</tr>
<tr>
<td>Address</td>
<td>AcctBal</td>
</tr>
<tr>
<td>RegComment</td>
<td>Clerk</td>
</tr>
<tr>
<td>PartType</td>
<td>Phone</td>
</tr>
<tr>
<td>Brand</td>
<td>TotalPrice</td>
</tr>
<tr>
<td>AcctBal</td>
<td>Comment</td>
</tr>
<tr>
<td>SupplyCost</td>
<td></td>
</tr>
<tr>
<td>NationName</td>
<td></td>
</tr>
<tr>
<td>Comment</td>
<td></td>
</tr>
</tbody>
</table>

Figure 4. The main dimensions after collapsing
By reassembling the multidimensional diagram using the results dimensions, final data warehouse schema will be obtained.

Many actions can be done on any candidate schema; first is to view a complete design using DiagrammerClass which generate the relationship diagram.

Second, generating and save script file which can be used to create this database.
3. CONCLUSIONS

This work presents an application to generate data warehouse candidate schemas from an OLTP schema using a new approach to automate the multidimensional design. It applies an algorithm to derive the fact tables, base on some criteria and factors. The user can view complete design of any generated schema or save script which can be used to create the schema. The user also can view the mapping table between the generated schema and the source schema. TPC-H Benchmark schema is an OLTP schema which used as an example. This application can be used to aid the designer of the data warehouses beside the enterprise data models. He will be able to choose the most suitable generated schema that meets the business requirements.

Also the application generates a mapping table, which can be used as a reference in ETL operations.
**FUTURE WORK**

In future, this proposed technique can be extended to be more powerful. Some functionalities can be added, therefore, be applicable in a fully automated system without any involvement of the user.

The mapping table represents a good start to design a protocol of data cleansing and data transformation between source and data warehouse schema.

The strategy can be expanded to include the heterogeneous database.

Additional algorithm can be designed to allow the user to refine the selected schema and to process these refinements.

**ACKNOWLEDGEMENTS**

I would like to express my greatest thanks to Dr. Tarig Ahmed who kindly supervised this study with greatest care. He has been a real source of useful support and careful guidance. I would also like to thank university of Khartoum, especially the college of Post Graduate Studies and the Faculty of Mathematical Sciences department of Computer Sciences. Thank to all those who encouraged, advised and supported me, extend my deepest appreciation.

**REFERENCES**


[19] Joseph Guerra, Vice President & Chief Architect David Andrews, President, “Why You Need a Data Warehouse”, 700 West Johnson Avenue Cheshire, CT 06410 800.775.4261


