Winter 12-2019

Graph-Based Methodology for Data Warehouse Schema Design

Roy Lev

Adir Even

Graph-Based Methodology for Data Warehouse Schema Design

Research-in-Progress

Roy Lev
Industrial Engineering and Management
Ben-Gurion University of the Negev
Beer-Sheva, Israel
royl@post.bgu.ac.il

Adir Even
Industrial Engineering and Management
Ben-Gurion University of the Negev
Beer-Sheva, Israel
adireven@bgu.ac.il

Abstract

A Data Warehouse (DW) is the infrastructural data foundation for Business Intelligence (BI) systems. This study proposes a methodology for designing a database (DB) schema for a DW. The core of the proposed methodology is a source-to-target schema conversion solution, based on directed-graph representation of a relational DB schema. It converts the graph representation of a source DB (e.g., one that supports operational information system) to a schema that would better fit DW requirements (a.k.a, "Star Schema"). The methodology does not aim at fully-automated conversion, but rather permits expert-user intervention for handling schema-design decisions that would require in-depth understanding of business context and interpretation. This manuscript presents the methodology foundations - graph-representation of a relational DB schema, and the schema conversion process. It also describes a prototype implementation of the proposed methodology, and discusses direction for future research progress.

Keywords

Business Intelligence (BI), Data Warehouse (DW), Database Schema, Star Schema, Graph Theory.

Introduction and Background

This study, which is still progressing, aims at developing a methodology for designing a database (DB) schema for a Data Warehouse (DW), an infrastructural data foundation for Business Intelligence (BI) systems. The methodology proposes a source-to-target schema conversion solution, based on directed-graph representation of relational schemas. It converts the graph representation of a source DB (e.g., one that supports operational information system) to a schema that would better fit DW requirements (a.k.a, "Star Schema"). The methodology does not aim at fully-automated conversion, but rather permits expert-user intervention for handling schema-design decisions that would require in-depth understanding of business context and interpretation.

BI systems offer infrastructure, tools and techniques for data visualization and analysis, toward data-driven decision support. BI systems have become an essential asset, as organizations growingly rely on data resources to remain competitive in highly uncertain environments. A DW, the data-infrastructure for BI systems, commonly integrates and restructures data from multiple sources, toward supporting business analysis and managerial decision support. The need to restructure data stems from the different nature of data use. Operational use typically mandates access to specific data records, while maintaining "One version to the truth", by avoiding unnecessary value duplications. Conversely, analytical use more often mandates aggregative view of a large number of data records, toward detecting possible correlations and effects.

The study addresses scenarios where both the data source and the DW are based on a relational DB schema with multiple interlinked tables. However, the different nature of data use mandates different approach toward schema design. Operational use is commonly supported by a normalized DB schema (Figure 1.a) - multiple tables, each with multiple attributes that functionally dependent on the table's primary key (PK's),

and some are linked by a foreign key (FK) to other tables. On the other hand, analytical use commonly relies on a "flat" DB structure (Figure 1.c), that stores all attributes in a single relation without necessarily enforcing functional dependencies. Flat structures may underlay various BI applications – reports, digital dashboards, interactive data inquiry (ROLAP – Relational On-Line Analytical Processing), data mining, and possibly others.

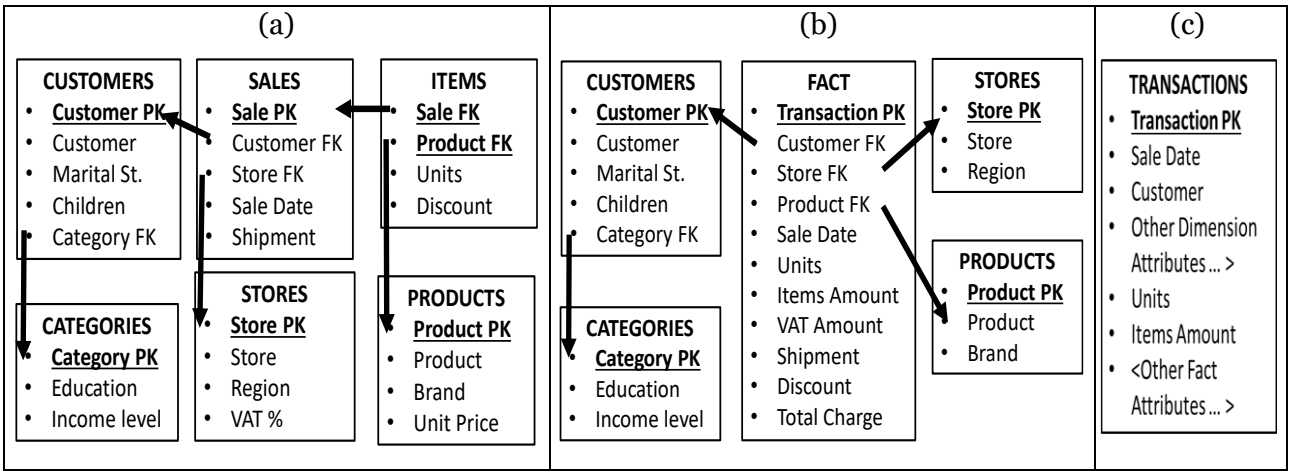

Figure 1. Database Schema Examples - (a) "Normalized", (b) "Star", and (c), "Flat"

(PK – Primary Kay, FK – Foreign Key)

Basing analytical data use on a normalized database schema might suffer from slow retrieval performance, as it may rely on multiple computationally-expensive JOIN operations. The semi-normalized "Star" schema (Figure 1.b) is a common DW solution to these kinds of performance issues. The Star-schema is based on a single fact table, containing numeric attributes (a.k.a., fact variables attributes) that reflect measurements of business activities and performance, and can be aggregated. The fact table is linked by foreign keys to multiple dimension tables, containing mostly-categorical characteristics of relevant subjects that can be associated with business activities and may influence performance (a.k.a., dimension variables or attributes). A well-designed "Star" schema is a convenient baseline for generating flat structures along various dimension/fact variable combinations. The generation would typically be much faster, vs. a normalized DB schema, as it would require less JOIN operations.

The design of a normalized DB is guided by well-grounded methodologies and supported by helpful tools (e.g., the ERD - Entity-Relationship Diagram), the Star-schema concept has not been introduced together with methodological conversion method; hence, commonly guided by a set of good practices and "rules of thumb" that have evolved over the years. A few studies have looked into defining methodologies that would support DB design in DW environments. Lechtenbörger and Vossen (2003) proposed formal definition of multidimensional normal forms that would guide DW design. Moody and Kortnick (2000) offer a few approaches for mapping source schema components into a DW. Phipps and Davis (2002) propose an algorithmic method for automated conversion; however, note that a successful application of such algorithm would require in-depth business understanding. Song at al. (2007) explore a semi-automated method that would guide the conversion not only by analysis of structure, but also by attribute semantics; however, they recognize the need for end-user intervention, in cased where the proposed analysis of semantics might fail to detect the correct business interpretation.

The limitations of previously-proposed approaches have motivated a different direction taken by this study – representation of a relational DB schema as a directed-graph, and a conversion that would permit expert-user intervention. This study suggests that some typical DW schema-design decisions cannot be directed by structure and data-type analysis alone, but rather require in-depth understanding business contexts and meaning; hence, likely to mandate expert-user intervention – e.g., adding calculated attributes and aggregations, tracking attribute-value transitions over time (a.k.a., the "slowly changing dimensions" issue), and attribution of fact-variable values. This paper describes the progress so far – it lays the foundation for the proposed methodology, demonstrates its preliminary prototype application, and discusses future research directions.

Relational Schema Conversion

The proposed methodology addresses conversion of a relational source DB schema to a relational DW schema, where the former is assumed to be 3rd-form normalized while the latter adheres to a semi-normalized Star schema.

Directed-Graph Representation: Extending Radev's representation (2013), a relational DB schema is represented as a directed graph $G = (T, F)$, where $T = \{T_1, T_2, T_3, \dots\}$ is the set of all schema tables and $F = \{F_1, F_2, F_3, \dots\}$ is the set of all schema foreign keys (FKs). Each table T_i is considered as a graph node and defined by its set of attributes $\{P_{i1}, P_{i2}, \dots, A_{i1}, A_{i2}, \dots\}$, where {P} denote a primary key (PK) attribute, and {A} denote attributes that have not been assigned as PKs. Each FK is considered as a directed edge and defined by $F_i = (T_a \{A_{a1}, A_{a2}, \dots\}, T_b)$, where $T_a \{A_{a1}, A_{a2}, \dots\}$ denotes the referring table and attributes and T_b denotes the referenced table. FK directs at the PK of the referenced table; hence, a referring attribute may participate in one FK at the most. The set of referring attributes may include PK attributes, and must match the referenced table PK in terms of number of attributes, their order and their data types. Two tables may be linked by multiple FKs; where is defined by a different set of referring attributes.

Source Node and Sink Node Tables: The set of tables that refer to table T_a with foreign keys are denoted $IN_a = \{T_i\}$, where $\exists\ F = (T_i\{A_{i1}, A_{i2}, \dots\}, T_a,)$. Similarly, the set of tables that table T_a refers to by foreign keys is denoted $OUT_a = \{T_i\}$, where $\exists\ F = (T_a\{\dots\}, T_i,)$. The number of tables in IN_a and OUTa are denoted $N^{IN}{}_a$ and $N^{OUT}{}_a$, respectively. A table T_a is defined as a source node if not being referenced by any FK (i.e., $N^{IN}a = o$). A table Ta is defined as a sink node if has no referencing foreign keys associated with its attributes (i.e., $N^{OUT}a = o$).

Directed Walk and Connected Subgraph: A directed walk exists from table Ta to table T_b if and only if there is a sequence of one or more directed FK edges that links the former to the latter. For each source node table T_a the connected subgraph G_a is combines the table T_a, all other tables $\{T_i\}$ that can be reached from T_a by any directed walk, and all the foreign keys $\{F_i\}$ that form those directed walks. A directed-graph representation of a relational DB schema may therefor contain several connected subgraphs $\{G_i\}$, one for each source node table $\{T_i\}$. The connected subgraphs may overlap, as some tables can be reached by directed walks multiple source node tables.

Figure 2 shows an example of a relational DB schema and its directed-graph representation. Table T_1 is the source node for that schema, as not being referred by other FK's. T_3 is a sink for that schema, as not referring other tables, and T_2 is neither a source nor a sink. This schema has a single connected subgraph G_1, with T_1 as its source node table, T_2 and T_3 that can be reached from T_1 by directed walks, and $\{F_1, F_2, F_3\}$, as the set of FK edges that form those walks.

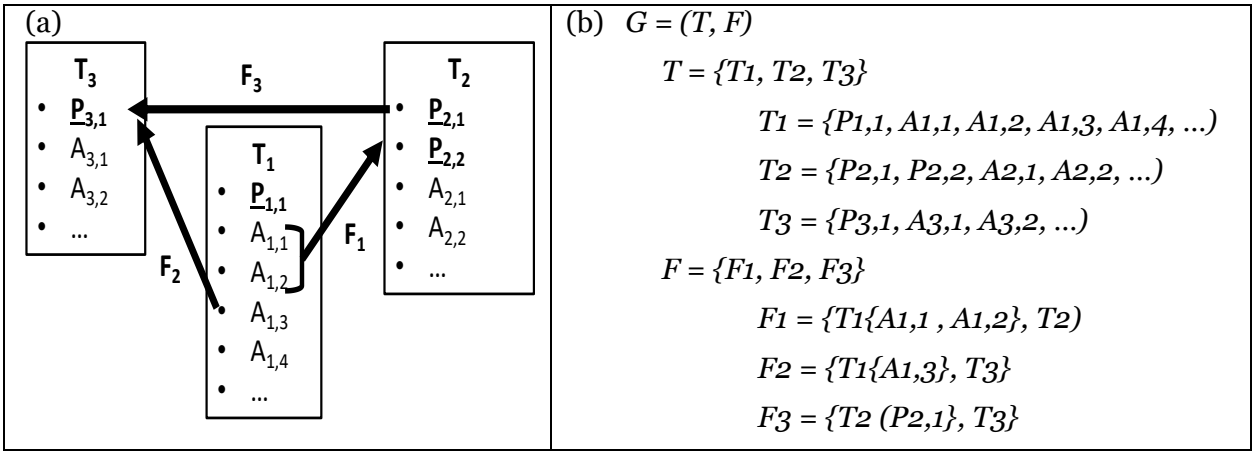

Figure 2. (a) A relational DB schema, and (b) Its directed-graph representation

Star Schema as a Connected Subgraph: A relational DW is commonly formed by multiple Star-schema structures. A typical star schema has a fact table that contains fact variables - numeric measurements that reflect business activities and performance. The fact table is linked by FK's to multiple dimension tables, reflecting business subjects that may influence performance (e.g., "Customers", "Products", "Locations"). Each dimension table contains dimension variables – mostly-categorical dimension characteristics that are potentially relevant for performance analysis. A dimension table can be linked to another dimension table, to reflect dimensional hierarchy (a.k.a., "Snowflake Schema") – e.g., "Customers" linked to the "Categories" dimension in Figure 1.b, to reflect hierarchical product categorization by brands. Dimension tables are often shared by multiple Star-schemas, reflecting subjects that affect multiple business activities and perspectives.

Figure 3 shows a directed-graph representation of a DW, where each star schema forms a connected subgraph, with a fact table as a source node and the associated dimensions, linked to the fact table by directed-walks. Some dimensions (e.g., DIM_1, DIM_2 and DIM_3 in Figure 3) can be possibly linked to more than one fact tables; hence, will be included in multiple subgraphs.

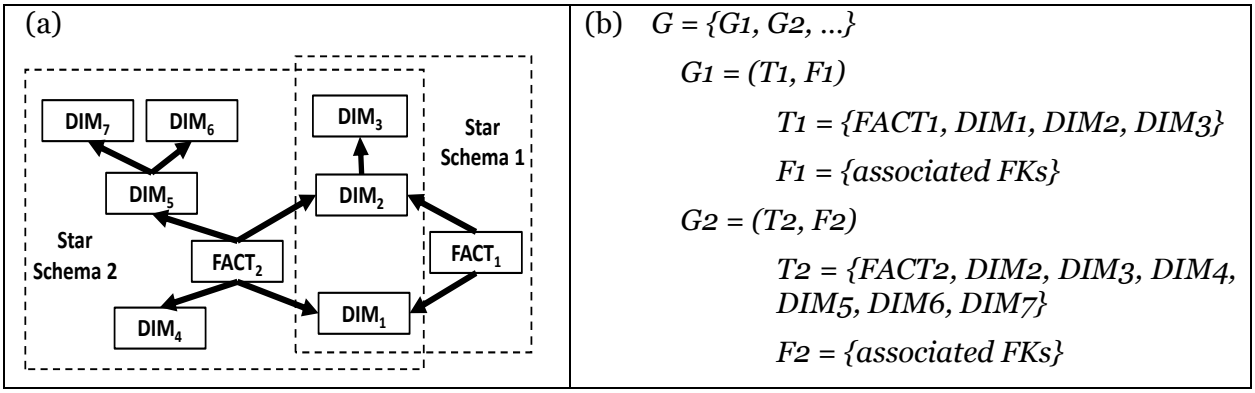

Figure 3. (a) Relational DW Schema, and (b) Its Directed-Graph Representation

Converting Source Connected Subgraphs to a Star Schema: A proposition that underlies the schema conversion methodology is that a connected subgraph at the source DB may become a candidate for conversion into a star schema. The source node table would direct the fact table design, and the other tables would become candidates for the associated dimensions. However, from analytical use perspective some star-schema candidates might be irrelevant; hence, the need for a preliminary assessment of connected subgraphs that can be detected in the source DB schema. The following example shows the conversion of the DB schema is Figure 1.a:

1. Directed graph representation $G^{Source} = (T, F)$, with ITEMS being the source node table:

T = (CUSTOMERS, CATEGORIES, STORES, SALES, PRDOCUTS, ITEMS)

CUSTOMERS = {CustomerPK, Customer, Marital., Children, CategorFK}

CATEGORIES = {CategoryPK, Education, Income Level}

STORES = {StorePK, Store, Region, VAT %}

SALES = {SalePK, CustomerFK, StoreFK, SaleDate, Shipment}

PRODUCTS = {ProductPK, Product, Brand, UnitPrice}

ITEMS = {SaleFK, ProductFK, Units, Discount}

F = (F1, F2, F3, F4, F5) The associated foreign keys

2. Directed-walks analysis, where each ITEMS transaction reflects a product included in a sale.

ITEMS → SALES:	SALES reflect transactions with "one-to-many" relationship with ITEMS; hence, can be de-normalized.
	Derived: TransactionPK, a surrogate PK
	Derived: Shipment, attributed along associated items.
ITEMS → PRODUCTS:	Relevant dimension
	Derived: Items Amount = Units * Unit Price
ITEMS → STORES:	Relevant dimension
	Derived: VAT Amount = Units * Unit Price * VAT %
	Derived: Total Charge =
	Items Amount + VAT + Shipment -Discount
ITEMS → SALES → CUSTOMERS:	Relevant dimension
ITEMS → SALES → CUSTOMERS → CATEGORIERS:	Relevant hierarchy

3. Directed graph representation GTarget = (T, F), of the outcome Star Schema

T = (FACT, CUSTOMERS, CATEGORIES, STORES, PRDOCUTS)

FACT = {TransactionPK, CustomerFK, StoreFK, ProductFK, SaleDate, Units, ItemsAmount, VATAmount, Shipment, Discount, TotalCharge}

CUSTOMERS, CATEGORIES, STORES, PRODUCTS Same as above (step 1).

$F = (F_1, F_2, F_3, F_4, F_5)$ The associated foreign keys

Prototype Implementation of the Schema Conversion Process

A preliminary prototype of a design-support system for expert users (Figure 4) implements the proposed schema conversion methodology. The prototype was developed for research-support purposes, and will be used for further development and evaluation. It was programmed with Visual Studio, using C# for programming. It is connected to relational DB's installed on Microsoft's SQL-Server, and utilizes graph-related libraries, programmed in R language.

The prototype implementation embeds a schema conversion process, which is based on the proposed directed-graph representation and consists of several steps:

1. Source schema extraction: The metadata that describes the source DB structure is extracted from the associated system tables, including table names, attributes and data types, primary and foreign keys. The source schema is then converted to the directed graph representation.
2. Star-schema candidates' detection: For each detected source node table, the associated directed walks are analyzed to form connected subgraphs - candidates for conversion into DW star schemas. Notably, the source DB schema can be potentially converted to multiple star-schema structures, possibly with shared dimensions. However, the user may decide at this point which connected subgraphs are relevant for the target DW and may choose to ignore the others.
3. Schema analysis: Each relevant star-schema candidate is then presented to the user for furthers analysis. After reviewing a preliminary full-scale conversion, the user may apply various schema

manipulations and adjustments, based on business need and the expected forms of data usage; e.g., (a) Removing attributes or even entire tables from the schema, if appear to be irrelevant, (b) Deciding whether certain attributes will act as fact or as dimension variables, (c) adding calculated attributes, based on other existing attributes, (d) Setting dimension hierarchies, by merging linked dimension tables. Other forms of adjustments are currently under development and will be offered to users in later releases of the prototype.

4. Target schema loading: Each directed graph that represents a star schema is translated back to a relational table structure, using metadata format that matches the target system tables.

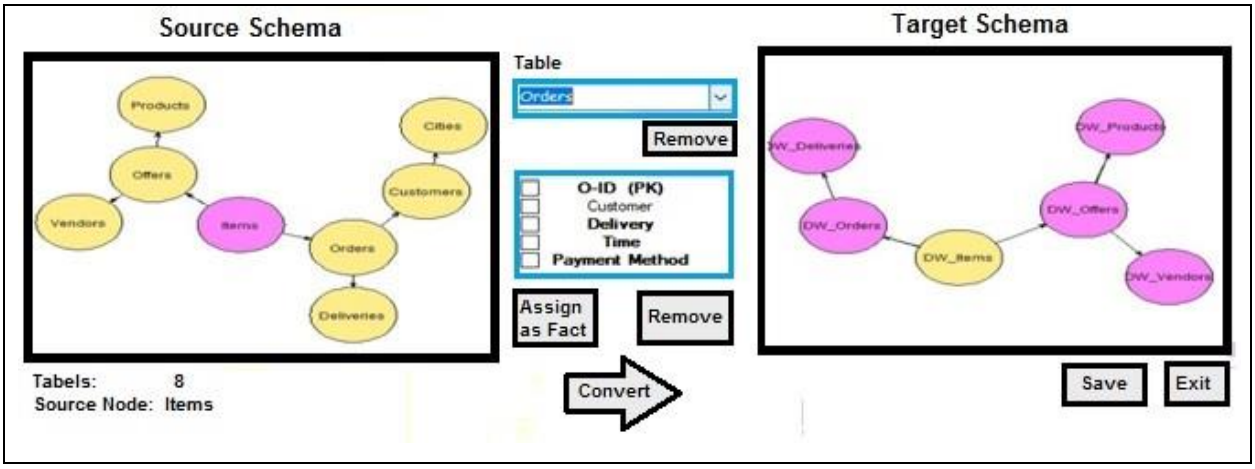

Figure 4. Conversion Support Tool – Prototype Implementation

The prototype served as a proof-of-concept for the feasibility and the potential contribution of the proposed methodology. It has been tested successfully so far with a few schema conversion scenarios. After enhancing the methodology to support a fuller range of DW design challenges, the prototype will be enhanced accordingly and tested more comprehensively with expert-users.

Conclusions

With the growing reliance on data resources for business analysis and decision support – careful design of a DW, as a data foundation for BI system, becomes a critical task. This study aims at developing a comprehensive methodology for DW design, based on directed graph representation of a relational DB schema. This manuscript presented the foundations for the proposed methodology and described the research efforts so far. However, full-scale design of a DW must address many other challenges. The study is currently developing solutions for other common DW design challenges that are based on the proposed directed-graph representation. For example:

- **Surrogate keys:** Replacing source table PK's with surrogate keys – e.g., for simplifying too-complex key structures, or for merging dimensional values from multiple sources.
- **Dimension hierarchies:** Representing hierarchies by multiple interlink tables (a "snowflake"), versus de-normalizing hierarchy representation to a single dimension table.
- **Slowly-changing dimensions:** Tracking transitions in dimension variable values and linking them correctly to fact table records.
- **Date/time dimensions:** Adding dimension tables that reflect hierarchical structure of date (year, month, quarter, etc.) and/or time (shift, hour, minute, etc.) variables.
- **Derived variables and fact granularity:** adding calculations based on other existing variables, possibly by reducing granularity (summation) or extending it (attribution).
- **Schema integration:** Merging multiple star structures along shared dimensions.

A major challenge currently addressed is the design of DW schema under multi-tenancy – the use of the same baseline schema for multiple IS instantiations, where each may assign different meaning and interpretation to a certain attribute. Multi-tenancy has become common with the growing popularity of cloud-based Platform as a Service (PaaS) solutions for information systems (IS) implementation. Multi-tenancy implies that multiple PaaS customers ("tenants") enjoy a highly configurable application, while multiple applications are assigned to the same database instance. From PaaS providers' viewpoint, multi-tenancy permits increased utilization of hardware resources lowers overall costs. The conversion of a normalized DB schema to a Star schema has so far assumed single-tenancy; hence, the motivation to extend this study to address multi-tenancy.

References

Lechtenbörger, J., and Vossen, G. 2003. "Multidimensional Normal Forms for Data Warehouse Design," *Information Systems* (28:5), pp. 415–434.

Moody, D., and Kortink, M. 2000. "From Enterprise Models to Dimensional Models: A Methodology for Data Warehouse and Data Mart Design," in *Proceedings of the 2nd Intl. Workshop on Design and Management of Data Warehouses (DMDW)*, Stockholm, Sweden.

Phipps, C., & Davis, K. 2002. "Automating Data Warehouse Conceptual Schema Design and Evaluation," in *Proceedings of the 4th Intl. Workshop on Design and Management of Data Warehouses (DMDW)*, Toronto, Canada.

Radev, R. 2013. "Representing a Relational Database as a Directed Graph and Some Applicationsm," *Brain-Computer Interfaces* (2013), pp. 19-26.

Song, I. Y., Khare, R., & Dai, B. 2007. "SAMSTAR: A Semi-Automated Lexical Method for Generating Star Schemas from an Entity-Relationship Diagram," in *Poceedings of the ACM 10th Intl. Workshop on Data Warehousing and OLAP*, Lisbon, Portugal.