Starflake Schema Implementation Using Depth-First Algorithm in Automating Data Normalization

Ria A. Sagum*, Rosemarie R. Afan, John Vincent R. Biscocho, Jed Simon D. Manansala, Alleen Princess Dianne T. Moncada
University of Santo Tomas, Espana Blvd, Manila, 1008, Philippines

* Corresponding author. Email: rasagum@ust-ics.mygbiz.com
Manuscript submitted February 10, 2015; accepted July 25, 2015.
doi: 10.17706/jcp.10.6.374-380

Abstract: The two most popular schemas used to implement a multi-dimensional model in a relational database are star schema and snowflake schema. A combination of a star schema and snowflake schema, whose aim is to utilize the advantages of both schema, is called a Starflake schema. This study discusses the application of Starflake schema to automate data normalization. The researchers created a system that accepts a file input of a sample inventory data from a business inventory system and its corresponding Entity Relationship Diagram structure, and established a rule-based methodology using the depth-first algorithm. The system successfully implemented the Starflake Schema and achieved data normalization. The final output of system was successfully implemented and it comprehensively showed the time of execution of each query, the Entity-relationships, the attributes of each entity and overall space utilization.

Key words: Data mining, data normalization, depth-first algorithm, multidimensional model, relational database, starflake schema.

1. Introduction

Data analysts usually go through the data by sending queries in iterations. Given this situation, the preferred modelling approach is based on multiple dimensions. The two most popular of the schemas used to implement a multi-dimensional model are *star schema* and *snowflake schema* [1]. Both schemas handle measures by mapping them into large-sized fact tables, recording every detail in a fact table. Clustered around these fact tables are dimension tables, which are of different categories. [2]

Star Schema consists of fact tables, referencing to a number of dimension tables. It is diagrammed by surrounding each fact with associated dimensions. It is more effective in handling simpler queries compared to other models [3]. Star Schemas are unnormalized which results in simpler queries, as the join logic is simpler compared to a highly normalized transactional schema. It also has improved performance for aggregation operations, which can easily be filtered and grouped by dimensions [4].

The snowflake schema is represented by centralized fact tables which are connected to multiple dimensions. The schema normalizes dimensions to eliminate redundancies, grouping multiple tables instead of one large table [5]. It saves storage space, but increases the dimension tables and require more foreign key joins. This results in more complex queries and reduced query performance, as well as added maintenance efforts. Data load in the snowflake schema must be highly controlled and managed to avoid update and insert anomalies [6].

A combination of a star schema and snowflake schema is called a *Starflake Schema*. It is a snowflake schema wherein some of the dimension tables have been unnormalized. It allows more efficient access to the data by various tools, and the ability to adapt to changes in user requirements [2]. It is also extensible, as it supports adding new dimensions. It is normalized to remove the overlap between dimensions, but results to an increased complexity of the load processes and inconsistency in the query results [6].

The researchers intend to create a system implementing the Starflake schema, which will make use of the depth-first algorithm to create an approach that will reduce or eliminate the problems in the schema regarding the processing speed, data integrity and space utilization in data normalization. After gathering information and modelling and implementing the schema, tests were performed during the implementation stage, where results were obtained and comparisons were drawn after each test case. The evaluators were able to conclude that the procedures using the depth-first approach were able to carry out the expected results and has successfully eliminated the data redundancies and the transitive dependencies from the inventory file.

The paper is organized as follows. In Section 2, we discuss the problem statements considered in making the system, and research design that we followed in modelling the schema and developing the system. Section 3 presents the test results on the simulation of data, while Section 4 presents its analysis and interpretation. Finally we conclude our paper in Section 5.

2. Research Design

The study considers several problem statements as to how it would benefit future computer-based applications, and researches. First of which concerns the performance of the application of the Starflake schema in terms of integrating query operations together with space utilization. The second highlights whether all transitive and data redundancies were properly eliminated during the data transformation as carried out by the system. Lastly, the effectiveness of entering unnormalized data into the system was considered, given that data integrity and accuracy must remain uncompromised for procedure.

In line with the problem statements, the researchers have laid out the procedures for the study that will answer the previous questions imposed before the creation of the system. The discussed system starts with the input of the *Entity-Relationship Model*, and the *Business Inventory Data from the Business Repository*. The database administrator will be asked to input the entities of the inventory through a GUI and identify its corresponding Entity Relationship Diagram Notations, and classify the Transaction entities. The inputted ERD will be processed to become the basis for the schema design. Other entities will be identified as component or classification entities. This differentiation will be based on the ERD, which shows the relationships that exists between entities. Hierarchies are also to be identified, grouping all related entities in the process, in reference to the research by Moody and Kortink, entitled “From Enterprise models to Dimensional models (2006)”.

The data will then be extracted for further processing through the ETL process. Data will be transformed to inherit the format of the data warehouse and remove redundancies. After which, the data will be prepared to be loaded into the master table and transferred back to the database.

During the schema implementation, existing data will be normalized based on the derived dependencies on the ERD. Upon completion of normalization, the fact and dimension tables will be created, and the normalized data will be loaded in the tables. This process will be repeated continually until 3NF is reached in each dimension. The output of the system would be the normalized table structure in Starflake Schema, and the space occupied and total execution time.

3. Test Results

In order to demonstrate the effectiveness of the system implementing Starflake Schema, the researchers tested the system on 2 test subject laptops, one with an AMD A8-4558M APU 1.6 GHz Quad-Core processor and another with an Intel(R) Centrino 2 (1.30 GHz). The program platforms used were MySQL 5.6.12 for the database and WAMP Server 2.4 for the server. The testing procedures assessed the speed and space utilization in transferring data and speed of normalization, with and without consideration to CPU utilization and Physical Memory Allocation. Records of 10, 50, 150, 450, 1000, 5000, 20000, and 100000 in .sql file were used as the test cases in the system.

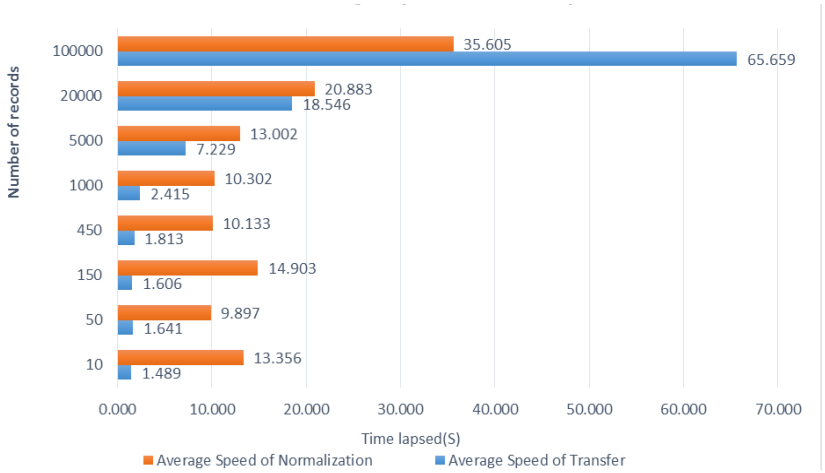


Fig. 1. Average speed of query (AMD).

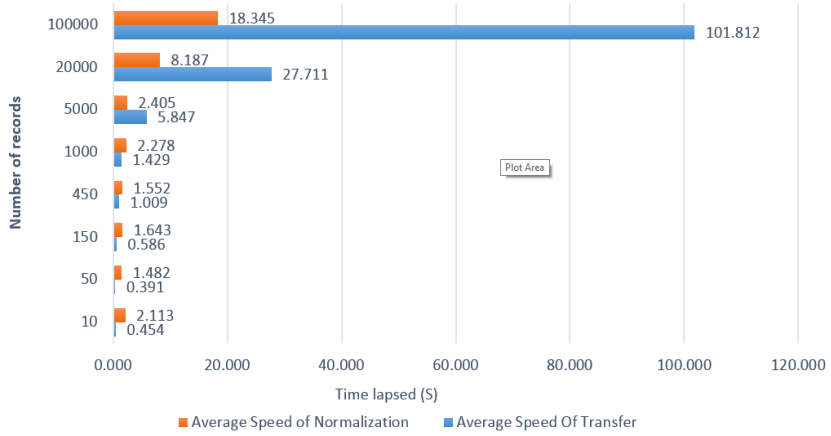


Fig. 2. Average speed of query (Intel).

Fig. 1 and Fig. 2 both show the average speed of normalization and average speed of Data Transfer in relation to the number of records imported in the system. It could be seen that the average speed of transfer increases as the number of records imported increased. On the other hand, the average speed of normalization exhibited a direct proportionality with the number of records on an import of 450 records onwards. For the files containing 10, 50, and 150 records, the average speed of normalization has no specific trend.

Fig. 3 and Fig. 4 show the space utilization of the system after import of files (initial database size), transfer of data, and normalization in relation to the number of records. For the 10, 50, and 150 records, there are no significant changes on the space occupied after the normalization process. However, on 450 records onwards, the space occupied increases as the number of records inputted also increases. It could also be seen that the amount of space occupied after the import of files and after the normalization process

have very minimal difference. The space occupied after the transfer of files, starting from the 450 records of input onwards, has an increasing value because of existing redundancies on the tables.

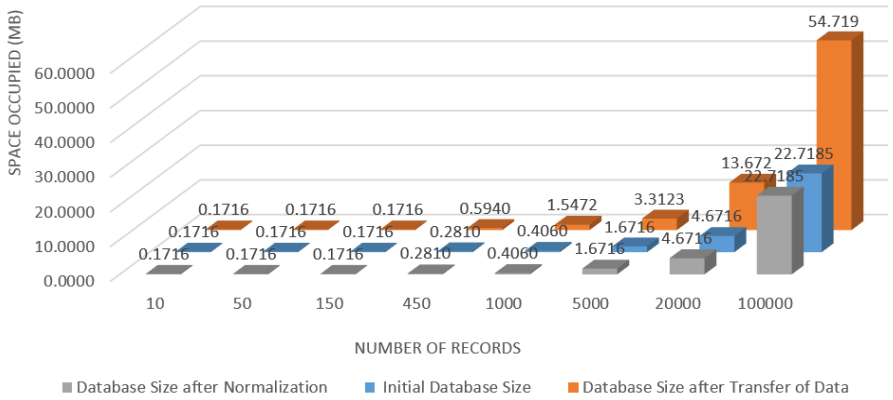


Fig. 3. Space utilization (AMD).

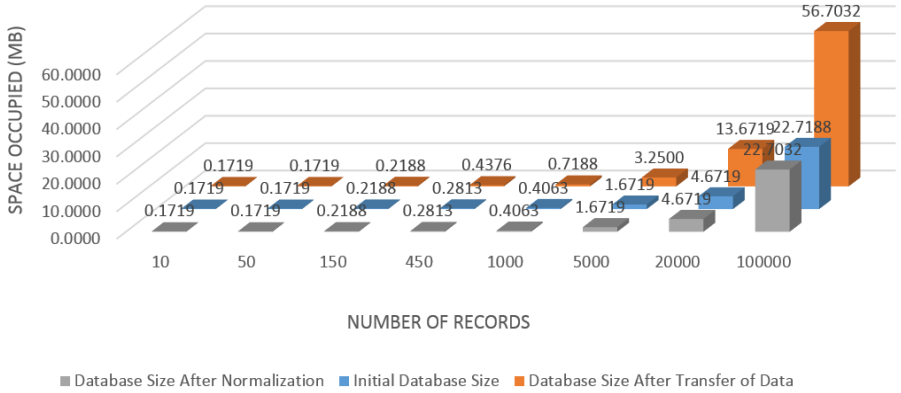
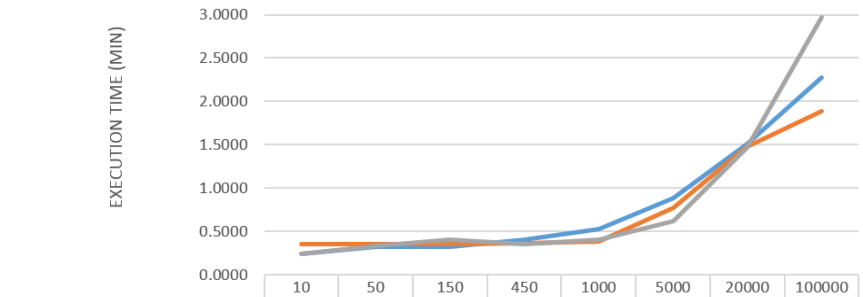


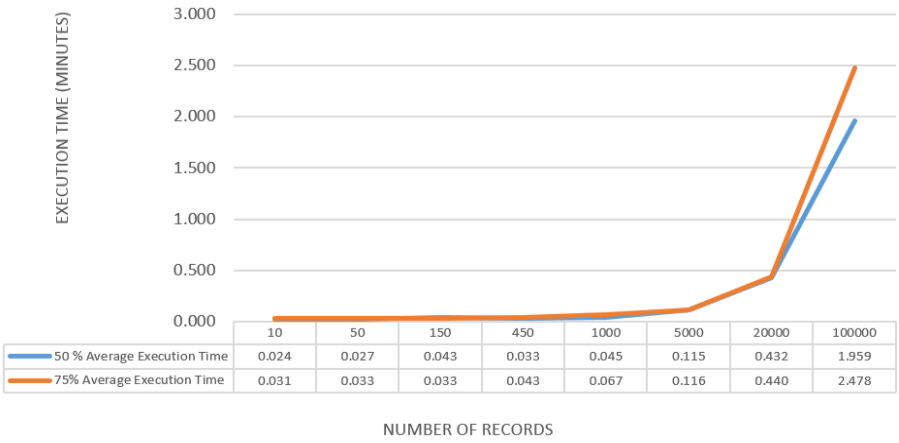
Fig. 4. Space utilization (Intel).

Fig. 5 and Fig. 6 show the average execution time depending on the percentage of physical memory usage in relation to the number of records. In both simulation, the average execution time of the 75% physical memory usage exhibited the highest increase in the time of execution. The average execution time on AMD at 50, and 20,000 records has the closest value in 25%, 50% and 75% physical memory usage. On the other hand, the average execution time on Intel has the closest value on the average execution time of 50% and 75% physical memory usage for all number of records imported in the system, except on 100,000 records.



	10	50	150	450	1000	5000	20000	100000
25% Memory Usage Average Execution Time	0.2468	0.3219	0.3300	0.4097	0.5302	0.8918	1.5105	2.2760
50 % Memory Usage Average Execution Time	0.3595	0.3533	0.3597	0.3653	0.3907	0.7694	1.4798	1.8870
75% Memory Usage Average Execution Time	0.2458	0.3370	0.4051	0.3532	0.4115	0.6171	1.4666	2.9710

Fig. 5. Average execution time (AMD).



	10	50	150	450	1000	5000	20000	100000
50 % Average Execution Time	0.024	0.027	0.043	0.033	0.045	0.115	0.432	1.959
75% Average Execution Time	0.031	0.033	0.033	0.043	0.067	0.116	0.440	2.478

Fig. 6. Average execution time (Intel).

4. Analysis and Interpretation of Results

Based on the given test cases and on the hardware used, it shows that the system could perform faster considering that the system runs alone than using it with several applications open. In addition, it also shows that factors involving the CPU utilization and the physical memory being used plays a key determiner on examining the system's performance upon delivery of desired results. It shows that the higher the percentage of memory utilization, the slower it is able to process the operations requested by the user, though it is still able to deliver the expected results regardless of the execution time.

In reference to the algorithm used, the system uses a depth first algorithm approach wherein it breaks down the system into subdivisions and traverses downward until it is able to fetch the needed attributes for the operations that will take place in the system.

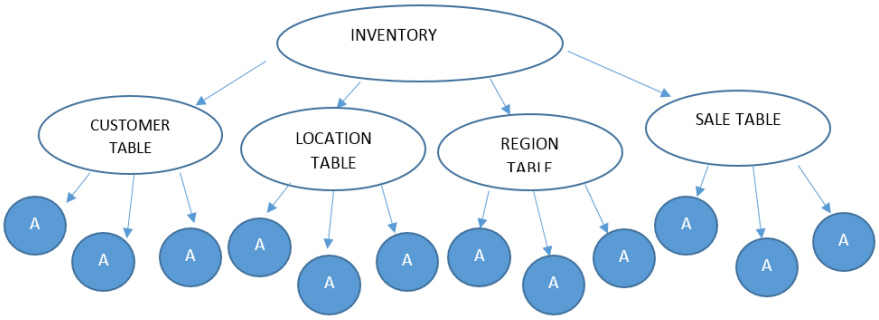


Fig. 7. Depth first algorithm.

For Fig. 7, the system fetches the children of the inventory table based on the entities entered into the system and will be stored into a table, making it possible for the system to transfer data based on their corresponding table and attribute name. With this approach, time complexity upon execution increases as memory utilization and CPU utilization increases in relation to time.

In reference to the testing of data integrity and bulk data gathering, three experts in the field of Database Administration were invited to test the system for evaluation and possible recommendations. After the sequence checks done by the evaluators a 100% rating was given confirming that the system has carried out the tasks expected based on the study's statement of the problem.

5. Conclusion

Through test cases and series of experimentations, the study shows that the depth-first approach is a

possible way to create a Starflake Schema structure using inventory data as input for this study. This is supported by documents made during the study, proving that the schema can be built by the system given that the constraints are followed correctly. A hierarchy was formed based on the classifications, allowing entities to be grouped based on their role before procedures are given off by the end-users through the system. Multiple inner joins were applied in executing queries to how whether the system table data are the same within a particular table or not. The accuracy of data produced, the speed of query, and the scalability of the system for larger data were also proven by the results obtained during the simulation testing and bulk data monitoring. The program also proved its capability to manage data by undergoing several stress cases with consideration to the CPU utilization and the physical memory. The researchers were able to conclude that as the number of records increase, a small interpolation exists among the tables imported to the system, proving that the system is capable to work under stress scenarios and still able to perform the automated data normalization process.

Acknowledgments

The researchers would like to express their sincerest gratitude to the guidance of several individuals who have assisted in the preparation and completion of this study. To Ms. Charmaine Ponay and Mr. Cecil Jose Delfinado, for your insightful comments, which helped us identify areas of improvement in our study. To Mr. Ellison Bartolome, for sharing his expertise in this subject matter. To our friends and family, who inspires us to strive for excellence. Lastly, we thank God our Father, for the many blessings he has bestowed upon us.

References

[1] Moody, D. A. (2003). From ER models to dimensional models: Bridging the gap between OLTP and OLAP design. *Journal of Business Intelligence*, *8*.

[2] Arfaoui, A. (2000). Data warehouse: Conceptual and logical schema-survey. *International Journal of Enterprise Computing and Business Systems*, *22*.

[3] Moody, M. K. D. (2006). From enterprise models to dimensional models: A methodology for data warehouse and data mart design.

[4] Teklitz, F. (2000). *The Simplification of Data Warehouse Design*. Sybase.

[5] Moody, D. L. M. A. (2008). From ER models to dimensional models part II: Advanced design issues. *Journal of Business Intelligence*, 1-12.

[6] Başaran, B. P. (2005). *A Comparison of Data Warehouse Design Models.* Retrieved February 2014, from http://www.uk.sagepub.com/upm-data/38123_Chapter2.pdf

Ria A. Sagum was born on August 31, 1969. She got the bachelor degree of computer data processing management from the Polytechnic University of the Philippines and done the professional education at the Eulogio Amang Rodriguez Institute of Science and Technology. She received her master's degree in computer science from the De La Salle University in 2012 and pursuing her doctorate in information technology at De La Salle University. She is currently teaching at the College of Computer and Information Sciences, Polytechnic University of the Philippines as an assistant professor, and a lecturer at the Institute of Information and Computing Sciences, University of Santo Tomas in Manila. Ms. Sagum has been a presenter at different conferences, including the International MultiConference of Engineers Computer Scientists 2015 and 6th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management 2013. She is an active member of different

international/national professional organizations.

Rosemarie R. Afan was born on May 27, 1993. She obtained a bachelor degree of science in computer science from the University of Santo Tomas, Manila, Philippines in 2015. She currently has a PHILNITS (Philippine National Information Technology Standards) Level 1 (Information Technology Passport) license after satisfying all of the requirements set by ITPEC (Information Technology Professionals Examination Council). She also obtained the IBM-DB2 Academic Associate certification after successfully satisfying the certification assessment by IBM in DB2 database management. She has undergone intensive trainings involving programming, systems engineering, database management, and networking. She is currently the university's Thomasian Youth Ambassadress and an active member in different organizations in the university.

John Vincent R. Biscocho got a bachelor degree of science in computer science from the University of Santo Tomas Manila, Philippines. He obtained his IT passport license after satisfying all of the requirements set by the Information Technology Professionals Examination Council (ITPEC). He is a qualified IBM-DB2 academic associate after successfully satisfying the certification screenings given by IBM in the field of DB2 database management. He has also received the Outstanding Member Awards coming from 5 student organizations back in 2011 and was also awarded with a Service Distinction Award in the same year for his contributions and services to his past alma mater.

Jed Simon D. Manansala was born on October 28, 1994 in Manila, Philippines. He obtained a bachelor degree of science in computer science from the University of Santo Tomas España Sampaloc, Manila in 2015. He obtained his PHILNITS (Philippine National Information Technology Standards) Level 1 (Information Technology Passport) license after satisfying all of the requirements set by the Information Technology Professionals Examination Council (ITPEC) on year 2013. He has undergone several trainings involving CISCO, compiler design, and DB2 database management (SQL). In college, he was chosen as one of the students on the Dean's List on two separate semesters, and his team's research was chosen by the College Board as part of the top 10 researches under the Computer Science Research Category for the AY 2014-2015.

Alleen Princess Dianne T. Moncada was born on March 20, 1995. She obtained her bachelor degree of science in computer science from the University of Santo Tomas in 2015. She got her PHILNITS (Philippine National Information Technology Standards) Level 1 (Information Technology Passport) license at November 28, 2013 after satisfying all of the requirements set by the Information Technology Professionals Examination Council (ITPEC). She had several exposures in business accounting in college including SAP (Systems Applications and Products) and has background in CISCO CCNA Routing and Switching to Cyber Security. She was able to design a compiler for her own programming language and has experience in developing web applications. Ms. Moncada's team's research was also chosen by the College Board as part of the top 10 researches in school year from 2014 to 2015.