A Critical Comparison Between Distributed Database Approach and Data Warehousing Approach

Sohrab Hossain, Farhana Islam, Razuan Karim, Kazy Noor-E-Alam Siddique

Abstract—Distributed database system and data warehouse are used to analyze data for decision support. There are some similarity between distributed database and data warehouse. However, the Concepts of federated database systems are confusing with the concepts of data warehouse. But they are quite different. There is a debate which approach is better for decision support. It actually depends on the types of business and information required by the business.

Index Terms—Data Warehousing, Distributed Database, Database Management System, Data Analysing Approach, Database, Data Analysis, Distributed system

I. INTRODUCTION

This report promotes to illustrate the concepts of database, distributed database, and data warehouse design and implementations. It shows how to implement a system for data analysis purpose from a root level. Firstly a relational database is developed that is not suitable for data analysis, and then a distributed database system is developed that has limited capabilities to analyze data, finally a data warehouse environment is developed which is the best way to analyze the data. However, it shows the clear difference between distributed database and data warehousing approaches.

II. LITERATURE REVIEW

A. *Research Methodologies*

The aim of this chapter is to review the theoretical framework of the database, Distributed Database System (DDBS), and Data warehousing (DW). The main aspects of database system are designing a data model and implement the data model into a database system. The widely used data model is relational data model and ERD is used to design a relational data model. The main aspects of DDBS are to Semantic and syntax mapping of heterogeneous database & their integration. The major areas in data warehousing are designing a data warehouse using star schema and snowflake, and OLAP architecture.

B. *Database System*

A database is a collection of self-describing information relevant to an organization. A database management (DBMS) system is software system that contains a collection of interrelated data and specifies how to access those data. The primary goal of a DBMS is to provide a convenient and efficient way to store and retrieve information from a database. [1] Basically a DBMS is developed for defining structures for storage of information and specifying how to manipulated information. Safety of the information is another major concern for a DBMS. It means how a DBMS protect data from unauthorized access or system crashes. The system need to avoid anomalous results when same data is shared by a number of users. [1]

C. *Distributed database system*

Distributed database system (DDBS) technology consists of two components: database system and computer network technologies. One of the motives for using database systems is to integrate the operational data of an enterprise in order to control access to that data. The computer network technology goes against all centralization efforts. Database system and computer network technology create a more powerful technology than either one alone. The important factor in distributed database system is the integration not the centralization. Integration and centralization does not imply the each other. The distributed database want achieve integration without centralization.

A distributed database contains two or more logically related databases and distributed over a computer network. A distributed database management system is software package that control and manage the distributed database system. [2]
In the above definition, we got two important terms “logically interrelated” and “distributed over a computer network”. A DDBS is not a collection of files that store in computer over a computer network. A DDBS has three characterizes: it should be logically related, it should have file structure, and access should be via a common interface. The physical distribution data is important. If two distributed databases reside in a single computer, we cannot encounter some problems those

may arise when data will be in two different computers. The physical distribution does not mean that the computers should be geographically apart. The computers can be in the same room. But the communication between them should be done over a computer network not in shared memory. [2]

A classification of distributed data management systems
There are four types of distributed database systems: Distributed database systems, Replicated database systems, Multi-database systems, and Federated database systems.

D. Data Warehousing

Bill Inmon, father of data warehouse, provides four characteristics of data warehouse: subject oriented, integrated, time variant, and non-volatile. A data warehouse is build to support management's decisions. [3]

1) Subject-oriented Data

In operational system, data is application processing oriented. For example, order processing application need data for entering order, checking availability, authenticate the customer's payment, and arranging shipment the order to the customer. On the other hand, a data warehouse stores data by subject. Data store by business subjects which may vary organizations to organizations. For example, a manufacturing company's critical business subjects are sales, inventory, and delivery of a shipment. Subway critical business subject is sales at the point of sales terminal (POST). [3]

2) Integrated data

In a data warehouse, data comes from various operational systems. Venda Ltd has two subways. Both stores have different operational systems. One store is using SUBSHOP 08 and another one is using SUBSHOP 05. So the inconsistencies should be removed from both operational systems before data stored in the data warehouse. [3]

3) Time-Variant Data

In an operational system, data contains current values. For example, an account receives system store information about the outstanding balance in the customer balance. On the other hand, data warehouse store historical data for analysis and decision making. For example, if Subway management wants to discover the hidden patterns of a customer spending, they need data from both current purchase and past purchase. The time variant of data will help Subway management to analyze past data by relating current information to forecasts the future. [3]

4) Non-volatile data

Data from operational systems are transformed, integrated, and store in a data warehouse for analysis purpose not to run day to day business operations. For example, when an order comes from a customer, a data warehouse will be used to know the current status of the stock. The operational system will be used to check the status of the product. Usually data in a data warehouse is not update. Data read form a data warehouse for query and analysis. [3]

5) Dimensional Modelling

Dimensional Modeling is a logical design technique deriving its name from business dimensions. This model is suitable for queries and analysis. The characteristics of the dimensional modeling are:

- ✓ It provide the way to access the data
- ✓ It is a query centric data model.
- ✓ It shows the interactions among dimensions and fact tables.It is flexible for drilling down, rolling up along dimension hierarchy

One of the major problems in the OLAP is that usage of information is totally unpredictable. Users cannot define their requirements clearly. Even they do not know how they would like to use the information or process it On the other hand, in OLTP precise functions are specified by end-users. [3]

6) Star Schema

Star Schema also known as dimensional model. It forms 'star-like' structure, which is called a star schema or star join. Every dimensional model (DM) is composed of one (or more) fact tables, and a number of dimension tables. Fact table contains numeric data value and dimension table contain description of the fact table.For example, what is the sale amount in Consumer Product category, for young customers in April 2008? Here sales amount is in the fact table and dimension tables are customer and time. Basically fact tables are narrow, big (many rows), numeric, growing over time. On the other hand, dimension tables are wide, small (few rows), descriptive, static. [10]

7) snowflake schema

In order to keep the data warehouse simple and easy to understand, dimension tables are not in normal form. They contain huge redundant information about hierarchies. Normalizing dimension tables leads to snowflake schema. In reality, snow flaking not recommended in most cases. Because more tables introduce more complex design, more joins and make the queries slower. [10]

8) Star Flake Schema

The star flake schema is a hybrid schema derived from star and snowflake schema. The star flake schema contains a fact table and a set of demoralized and normalized dimension tables. [6]

III. DESIGNING THE DATABASE

The company needs a new system for their daily operations. Designing a new database system needs three steps: analyze the user requirements, design a data model, and implementing the database. The functional requirements were collected from the company by interviewing the store manager and non

functional requirements were also consider during design and implementation phases. A relation data model was design for from the functional requirements and finally the database system was implemented by using the relational data model.

A. Entity Relationship Data Model Dessign

There are two types of requirements: Functional requirements and non functional requirements.

1) Functional requirements

A functional requirement describes what the functions of a software system are. A function defines the inputs, operation, and output of the software system. Functional requirements show how a use case is to be fulfilled. A use case was design to show the functional requirements of Subway.

2) Use case design

Customers are logical actors and employees are the physical actors for the system. Manager and staff are the specification of the employees. Both manager and Staff can input waste for usage adjustment. Normally customers start the "Buying use case" and get benefited. So customers are the logical actors. Physical actors, Staff, serve customers and finish the transaction. Manager can only maintain user accounts.

"Enter Item" and "Make a Payment" use cases contain <<include>> relationship with "Buy Item". The <<include>> relationship allow to reuse the use cases. Now if a customer wants refund, the Buy Item use case can be re use.

3) Entity Relationship Diadram

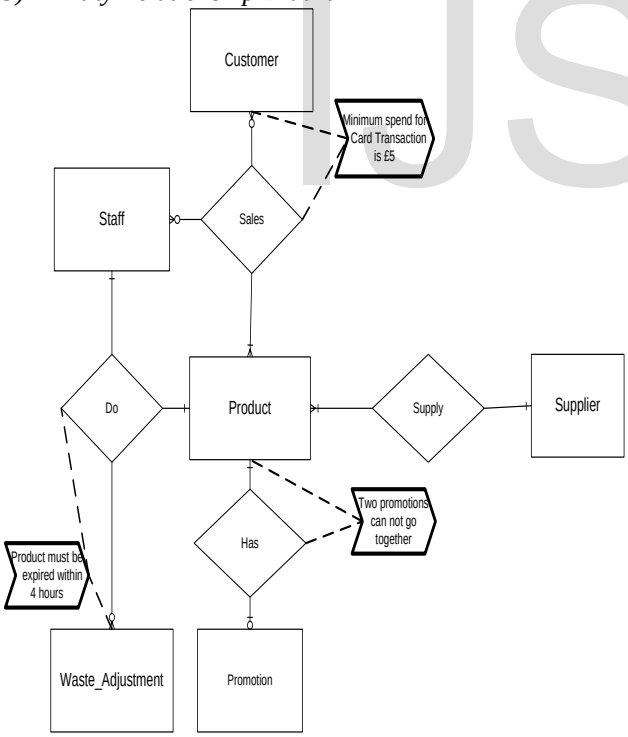

Figure 1. A conceptual ERD a Fast Food Shop

4) ERD's description

A staff may sell zero or many products. A product may be sold by one or many staff s. A customer must by one or many products. A product may be sold to one or many customers. A product may not be sold at all. A staff may do zero or many waste adjustments. A waste adjustment must be done by one staff. A product may have zero or many waste adjustments. A waste adjustment must be done for one product. A product may have zero or many promotions. A promotion must have a product. A supplier must supply one or many products. One product must be supplied by one supplier.

5) Business rules for ERD

A customer must spend at least £5 to make a debit or credit card transaction. In order to make a waste adjustment, the product self life must be four hours or less than four hours.

A customer cannot mix up two or more promotions for one transaction. The minimum amount of cash transaction must be £0.50 and maximum amount of transaction is £999.

B. Distributed Database System Design

A fast food company wants to merge their information system and data. The management wants to extracts information about sales, stock, supplier, promotion, usage adjustment, food cost, and labor percentage from both store databases. The two databases need to be merged to satisfy the management queries. The DDBS building process need two phase: Design and Implementation.

In a multi-database system global users can access database by using global external schema. However, local users can access local database by using local external schema.

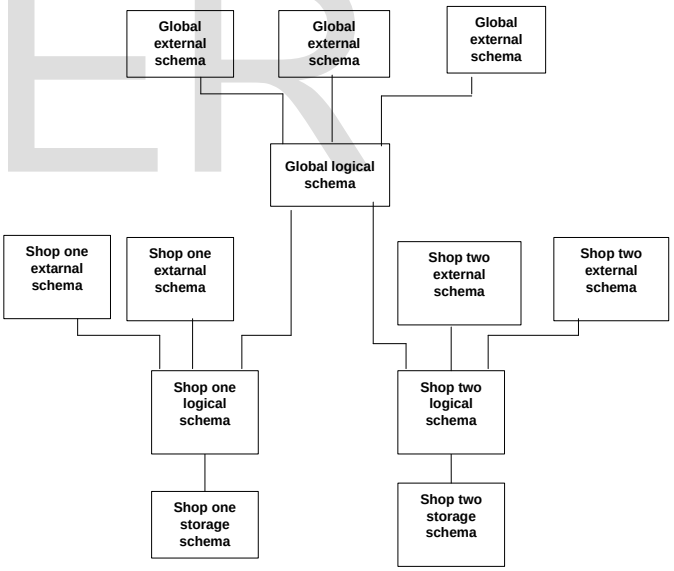

Figure 2. A multi-database system form a fast food shop

Both shop one and two can operate independently of the global system. However, they can apply full range of DBMS functionality to the local data. Local users can access local database through local external schema. Global external schema and a global logical schema allow users to access shared information from any sites. For example, operational users from store one and store two can access the system via local external schema without effecting the global schema. In

addition, Subway management can access the data from both stores through global schema without affecting the local users. [9]

In this stage, the above multi-database system design will be used to merge the two databases and implement a global view for the Subway management. Both databases contain similar number of entities but there are huge naming conflicts among them. However, the numbers of attributes between the entities are also different.

The entities and attributes for the store one are Staff (staff_id, staff_first_name, staff_last_name, street, city, address, region, postal_code), Customer (customer_id, customer_first_name, customer_last_name, street, city, address, region, postal_code), Product (product_id, product_name, plu, price, stoke, amount_on_order, spplier_id, promotion_id), Promotion (product_id, discount), Sales (customer_id, staff_id, product_id, date_commited, selling_price), Supplier(supplier_id, supplier_name, address, phone), and Waste (staff_id, product_id, waste_description).

The entities and attributes for the store two are Worker (worker_id, name, address), Client (client_id,name, address), Food (food_id,food_name, plu, buying price, stoke, amount_on_order, spplier_id, promotion_id), Offer (product_id, discount), Transction (client_id, worker_id, food_id, Transction_ date,price), Supplier(supplier_id, supplier_name, address, phone,city, post_code), Usage_adjsutment (worker_id,food_id, adjustment_details).

C. Data Warehousing Design

The data warehouse contains data from both the operational systems. The operational system use ERD model for the data base design and fully normalized in order to remove the redundancy. But for the data warehouse design, the dimensional modeling is used that reintroduce the redundant data. In the design part, a conceptual star schema, a logical star schema, a snowflake, and a starflake schema are design. In the implementation part, the dimension and fact tables are created using SQL language. Then, hierarchies are put on the top of the tables. After that, the staging area is selected where data comes from operational system. For loading the data warehouse, PL/SQL, SQL*Loader, and Exporting from Access techniques are used. To increase the efficiency of the data warehouse, B*tree, Bitmap, and Bitmapped Join Indexes are created. Finally, in order to frequent the queries, materialized views are implemented.

1) Star Schema

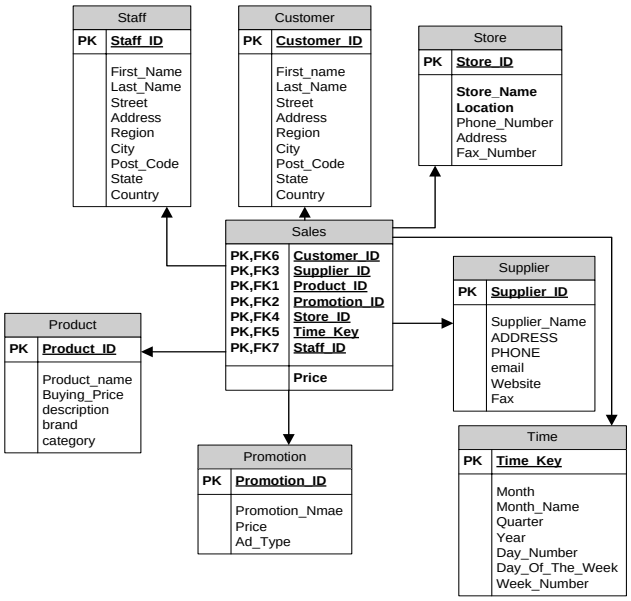

Figure 3. Logical Star Schema for a fast food shop

A conceptual Star Schema shows the dimension and fact table for the data warehouse. Seven dimension tables are Staff, product, promotion, customer, supplier, time and store. One fact table is Sales.

The logical star schema shows the dimension tables and fact table with all possible columns, Primary Keys, and Foreign Keys. Fact table sales contains a composite primary key (Customer_ID, Product_ID, Promotion_ID, Time_Key, Supplier_ID, Store_ID, Staff_ID) comes from all dimension tables as foreign keys. The lowest grain for the operation system is selected because most of the queries are based on single transaction. The lowest grain is the single product per transaction.

2) Snowflake Schema

Snow flaking removes the redundant data from dimension tables. The snowflakes are linked together via foreign keys. For example, product table splits into three tables. The flake contains information about product name, price and description. It does not contain information about Brand but one foreign key from Brand table. The Brand table does not contain information about Category but one foreign key from Category.

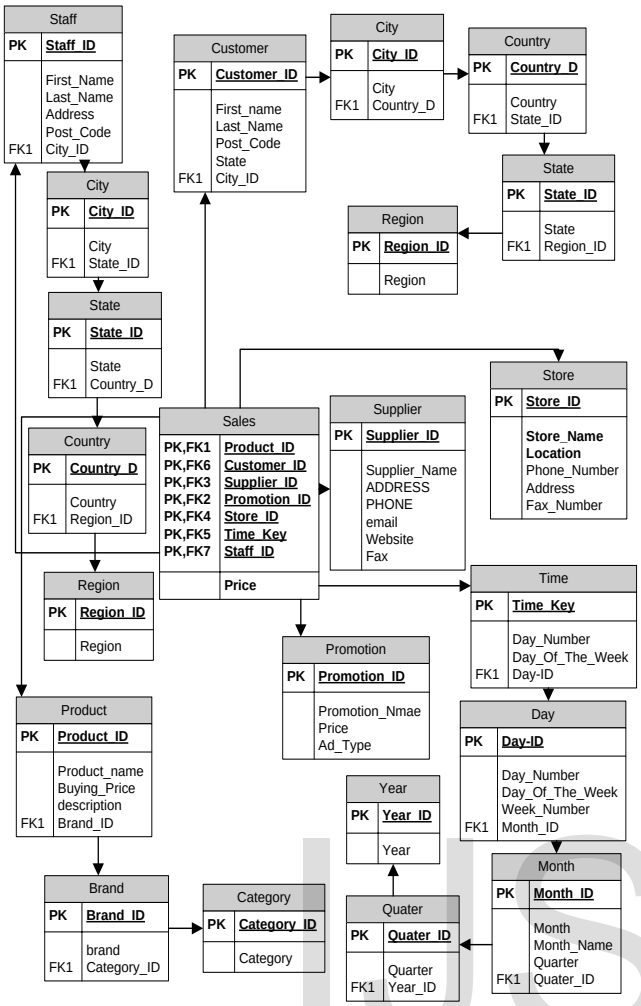

Figure 4. Snowflake Schema for a fast food shop

IV. COMPARISONS BETWEEN DDBS AND DW

Table 1. Comparison between DDBS and DW

DDBS vs. DW	DDBS	DW
Source of data	Data comes from day to day transaction	Data comes from various operational systems
Data content	Data contains current values	Data is historical, derived, summarized
Purpose of data	To run day to day operation	To make decisions
Access Type	read, write, delete, update, add	Data is used for query purpose
Usage	Users can define almost all the requirements at the early stage	Unpredictable
Response time	Very fast (Sub-second)	Depends on the complexity and data involve in the query (Several second to minutes).
Users	Depends on the system (thousands).	The number of users is less than OPTP (hundreds).
Database Design	Entity Relationship Modeling is used to develop the system. Data is highly normalized	Data is demoralized. Dimensional modeling is used for the OLAP design.
Space Requirement	OLTP takes less space than OLAP because is archived regularly (100 MB to GB)	OLAP takes much more space than OLTP because this the place where data is archived from operational system (100 GB to TB)
Queries	Queries are relatively simple and need retrieval of few records	Queries are complex, often needs aggregations, multi dimensional views of data
Access Frequency	Access frequency is very high as it is used for day to day operation	Access frequency is low as it is used for query.
What the data reveals	OLTP provides information about business processes	OLAP provides multidimensional views of business activities

V. CONCLUSION

The concluding chapter is an overview from chapter I to chapter IV. In addition, the findings, future works, and personal evaluation are added in this chapter.

A. *Summary*

Chapter I discuss about the goals of the report. Chapter II discusses about the literature review of database, distributed database and data warehouse and those fit in the proposed system. Chapter III discusses how to design the software system. Chapter IV is about the comparisons between distributed database system and data warehouse.

B. *Findings*

- ERD is suitable for the operational database system because data is update frequently. Entity Relationship model normalized data fully which prevent Insertion Anomaly, Deletion Anomaly, and Modification Anomaly.

- The star schema is the most suitable way to design a data warehouse. It has a fast query performance although it introduces redundant data in the data warehouse.

C. *Future Works*

- A dimension about the weather can make the data warehouse more robust. If a dimension containing the information about the weather (for example, Rainy, Cloudy, Sunny, extra hot, worm, and so on), Food Shop management can predict the impact of weather on sales. For example, if it is a rainy day, tuna sandwiches are sold most with coffee, if it is a sunny day, they got lots of tourist customers.

D. *Personal Evaluation*

The project got several twist during its lifetime. The proposed system was to merge databases from SQL Server 2000 and Oracle database 10g to create a distributed database system and to build the data warehouse by extracting data from both databases. Due to merging problem of databases in Windows XP, both databases are implemented in Oracle 10g.

Acknowledgment

We really are grateful to Ms Elena Teodorescu, Senior Lecturer, University of Greenwich, London, UK for her advises, support, feedback, and guidance.

REFERENCES

[1] Abraham Silberschatz, Henry F. Korth, S. Sudarshan (2006), Database System Concepts, McGrawHill, New York, YN 10020.

[2] M. Tamer Özsu , Patrick Valduriez (1999) , Principle of Distributed Database System, Prentice Hall, Upper Saddle River, New Jersey..

[3] Ponniah, Paulraj (2001), Data warehousing fundamentals, Wiley, Chichester.

[4] Oracle Database 10g vs. Microsoft SQL Server 2000: Technical Overviewhttp://www.oracle.com/technology/products/database/oracle10g/pdf/cwp_general_o10g-vs-ss2k.pdf 12/05/2008 20:12:51

[5] http://www.dba-oracle.com/t_oracle_10g_pc_hardware_windows_requirements.htm Oracle 10g PC Windows and hardware requirements? 12/05/2008 17:21:12

[6] Data Warehousing - Design Methodologies for Data Warehouses (DWHs) http://www.juergen-konicek.de/Informatik/BI/ dwh.html 12/05/2008 16:43:26

[7] Njovu, Dr Chiyaba, Distributed Data Management Technology (COMP1419) MSc Teaching Schedule 2007/08 The Database Environment, Lecture 1 https://cms1.gre.ac.uk/teachmat0708/COMP1419/ course/schedule.asp?banner=COMP1419 [Accessed 12 May 2008]

[8] Njovu, Dr Chiyaba, Distributed Data Management Technology (COMP1419) MSc Teaching Schedule 2007/08 The Relational Data Model Lecture2 https://cms1.gre.ac.uk/teachmat0708/COMP1419/course/schedule.asp?banner=COMP1419 [Accessed 12 May 2008]

[9] Blake, Dr Simon, Distributed Data Architecture & Management (COMP1421)
MSc Teaching Schedule 2006/07An introduction to distributed data management, Lecture 2
https://cms1.gre.ac.uk/teachmat0607/COMP1421/course/schedule.asp?banner=COMP1421 [Accessed 12 May 2008]

[10] Business Intelligence & Data Warehousing (COMP1420)
MSc Teaching Schedule 2007/08 , More data modelling.
Data Warehousing Concepts, Lecture 2 & 3,
https://cms1.gre.ac.uk/teachmat0708/COMP1420/course/schedule.asp?banner=COMP1420 [Accessed 12 May 2008]

[11] INTRODUCTION (OLTP vs. OLAP)
http://www.rainmakerworks.com/pdfdocs/OLTP_vs_OLAP.pdf
13/05/2008 18:38:17

[12] Non-functional requirements
http://en.wikipedia.org/wiki/Non-Functional_Requirements
14/05/2008 18:02:00

[13] Software Testing
http://www.ece.cmu.edu/~koopman/des_s99/sw_testing/
14/05/2008 14:43:43

[14] System testing
http://en.wikipedia.org/wiki/System_testing
5/14/2008 2:57:16 PM