Computer Science and Information Technology 5(1): 18-26, 2017 http://www.hrpub.org
DOI: 10.13189/csit.2017.050103

Data Warehousing: A Practical Managerial Approach

Max North[1,*], Larry Thomas[1], Ronny Richardson[2], Patrick Akpess[2]

[1]Information Systems Department, Coles College of Business, Kennesaw State University, USA
[2]Management & Entrepreneurship Department, Coles College of Business, Kennesaw State University, USA

Abstract The primary goal of this article is to provide managers and executives the tools and information needed to make informed decisions concerning data warehousing, to understand the processes and technology involved, and to identify individuals' responsibilities. The information is presented in clear, understandable terms and is designed for decision makers with little or no Information Technology (IT) background.

Keywords Data Warehouses, Data Marts, Database, DBMS, Database Administrators, Data Administrators

1. Introduction

This article is a *practical managerial approach.* Therefore, we will not get too deeply into specifics, but rather supply managers with the tools to ask the right questions and the basic knowledge required to understand answers received from technical staff. Managers want to know how to make the right decisions in the information technology (IT) field without having to become experts. We designed this article to accomplish just that. While there have been a good number of published articles on data warehousing, there exist no specific article to address the needs of managers in particular and to assist them in understanding technology, processes, and communication with all the parties involved. Consequently, this article measurably contributes to the literature of this field and fills the missing gaps between theories, practices, and management approach.

To begin with, we will take a look at the differences between databases, data marts, and data warehouses and set the stage for the remainder of the article. Some people define a data warehouse system as a collection of data used to make decisions; however, this definition would equally describe databases and data marts. Others may ask, 'Isn't a data warehouse simply an extremely large database?' The answer is yes, and yet no. So what makes a data warehouse?

1.1. Databases

Let's start with the smallest building block: a database. A database is a collection of related data tables (Radhakrishna, Kumar & Janaki, 2015; Idreos, Papaemmanouil, & Chaudhuri, 2015; Hoffer, Venkataraman, & Topi, 2016; Coronel, & Morris, 2016). The data tables might use something as simple as a spreadsheet or as complicated as Microsoft's SQL Server® or Oracle®. One example of a database might be the tables used by Human Resources (HR), such as a personnel table listing employee id, first name, last name, department, job, home address information, and so on, in a company. All of the tables relate in some way to HR. Most IT professionals will agree that a single database must use the same database management system (DBMS) and be located on the same server or personal computer (PC). Finally, databases generally contain operational data using on-line transaction processing, or OLTP (Bog, Sachs & Zeier, 2011; Lafuente, Downs, Yang & Stone, 2015). This data is constantly changing with each transaction processed. Thus, a car dealer could ask the database how many cars were sold so far today and get an answer.

1.2. Data Marts

Data marts are designed for a particular group of users and are also based on operational data. They can come from a single database but generally will incorporate several databases grouped together for specific relational purposes (Bonifati, Cattaneo, Ceri, Fuggetta & Paraboschi, 2001; Malhotra, 2015; Nguyen, Wagner & Schoepp, 2014). Although data marts contain mostly operational data, they can also contain historical data. If, for example, a major corporation pools all of the operational databases from each of its HR departments, it will have created a data mart. Another example might comprise an entire business unit of a major corporation which has multiple business units. Remember: a data mart is designed for specific group of users.

1.3. Data Warehouses

That leaves the data warehouse (DWH). We prefer the definition used by Michal Boehnlein and Achim Ulbrich vom Ende (1999): "*Data warehouse systems offer efficient*

access to integrated and historical data from different, partly heterogeneous and autonomous information sources in order to help managers in planning and decision making" (Boehnlein, 1999). The key component of this definition is that data warehouses are built off different databases. These databases may be on separate servers, use different DBMS, be located in different parts of the globe, or be fully functional in their own rights (Golfarelli & Rizzi, 2009; Jarke, Lenzerini, Vassiliou, & Vassiliadis, 2013; Jarke, Jeusfeld, Quix & Vassiliadis, 2013).

A data warehouse is not an end product; it is an information environmental system for executives and managers to make decisions, and thus, it is a Decision Support System (DSS). Data warehouses are always changing and always evolving (Dedić & Stanier, 2016; Chhabra, Kumar & Pahwa, 2016). As often decision makers change the questions they ask, so may the data warehouse change the data it contains or the processes it runs.

Figure 1 shows an example of a Data Warehouse for a corporation with two Data marts, one for HR and one for Operations. If the corporation were to have two Groups, East and West, then we could add a mirror image of this figure for the Western Group, and have their DWs feed data into a centralized DW. At which time, the corporation would refer to the Group DWs as Data marts, while within each Group, it is still a Data Warehouse.

Data warehouses (DWH) contain integrated operational and historical data. The DWs may contain the underlying data or be capable of directly querying it. Raw data should not be changed, but may need cleansing. Cleansing does not necessarily mean correcting as much as it means standardizing and may exclude certain raw data which cannot be standardized or does not conform. This may result in the data being corrected at the source—that is, in the original system that initiated the data. There are two other two key elements to discuss here within a successful data warehouse. The first is a good data dictionary describing the data coming from other sources and how they translate into the existing DW. The second key element is an exhaustive set of Master Tables containing reference codes and their definitions. For example, most everyone in the United States knows the two-character code for each of the 50 states. However, if your company has territories based on the states, then a master table containing the state codes and the territory to which each relates is a great idea.

The best way to determine if your business needs a data warehouse is to read this article to see what is involved and what benefits can be realized. We will take the assumption that databases already exist. Some things we discuss here must be accomplished in a specific order, while others can and should be performed concurrently. Keep in mind that data warehouses cannot spring up overnight, and it is best to bring your data warehouse on-line in phases.

If you are not familiar with these terms, they are defined at the end of this article: DBMS, OLAP, ETL, Primary Key (PK), Foreign Key (FK), Drill-down, DTS, Star Schema, Snowflake, large flat tables.

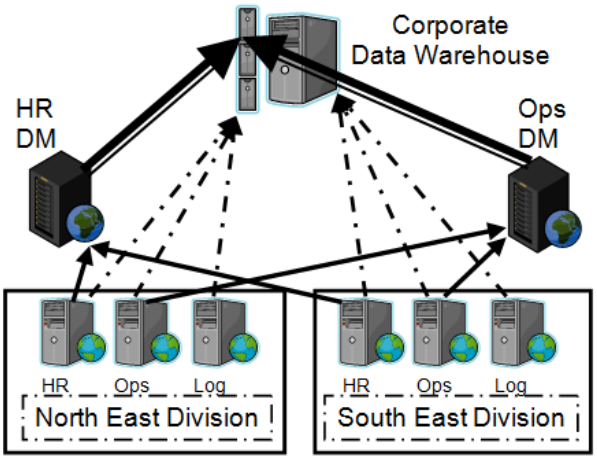

Figure 1. An example of a data warehouse for a corporation with two data marts: one for HR and one for Operations.

2. Determine the Scope

What the non-IT professional does and what the IT professional does.

2.1. Non-IT Professional

Before IT professionals can begin to assist management in determining scope and designing a data warehouse, they need to know what information decision makers need, and what the data warehouse is expected to do. The best recommendation is to create a mission statement that is neither too specific nor weighted towards a specific department or function. This is the responsibility of management and is best performed by a team (hereafter referred to as The Team), with representation from all of the decision making departments. The mission statement should focus on the mission of the end users (the people who will ultimately be using the product) and how you intend to employ the Data Warehouse without describing specific data sources, technologies or products.

Some things to keep in mind when creating the mission statement include the following considerations: the level of detail in the data (i.e. do you need the serial number and list of all components of every item sold or just the count by type?), the amount of summary, or roll-up, of the data, and the currency of the data (do you need it refreshed once daily, twice a day, hourly, etc.). Remember that operational data uses OLTP and is real-time, live data; whereas a data warehouse is not designed to operate as such. You will not necessarily address these in the mission statement, and it is probably best if you don't, but the mission statement should not hinder any of these considerations (Smith, 1997).

Clearly defined user requirements are a necessity before the IT professional can begin assessing the scope. This information is best gathered using both surveys and

interviews. Ensure that users are looking both at immediate needs and future possibilities. Also, look at all standard reports for the possibility of automation as well as standard inquiries.

Once the mission statement is written and accepted, the data warehouse architect or data analyst can begin to assist in the design of surveys and questionnaires, which will assist in determining the above design components. Some good questions you may want to ask decision makers are which reports or types of information are 1) fairly routine, 2) time-consuming and, 3) would save considerable time if they could be fully or at least partially automated?

Ralph Kimball, considered one of the foremost experts in data warehousing states, "*Very few organizations or human beings can develop the perfect, comprehensive plan for a data warehouse up front. Not only are the data assets of an organization too vast and complex to describe completely up front, but the urgent business drivers, and even the staff, will change significantly over the life of the first implementation. Start with a lightweight "data warehouse bus architecture" of conformed dimensions and conformed facts, and build your data warehouse iteratively. You will keep altering and building it forever" (Kimball 2002).*

2.2. IT Professional

The first thing to determine after gathering clearly defined user requirements is what data is needed and where will it come from. Will the business be combining different internal databases? If yes, the IT professional will need to do an assessment to answer the following questions:

1. How many existing tables?
2. How many existing records?
3. How much existing hard drive space is currently being used?
4. What is the projected growth rate?
5. What external data will we gather?
   a. Data from other Corporate/Government entities?
   b. Data from direct downloadable sites?

Next, what external data will be gathered and where will it come from? The data may need to be uploaded manually, and these are questions the IT professional is best positioned to examine and answer. Your company may be paying for market research data and need this data uploaded into your data warehouse. The preferred method is to connect the two databases together using a protocol such as ODBC, Open DataBase Connectivity. This allows the databases to directly communicate and transmit the data—often automatically. This means the database administrator does not have to log in at midnight to transfer the data; the two servers can accomplish this without having to pay someone to run the process every night.

Armed with all of this information, the IT professional can begin determining complexity, which will aid in the decision of hardware specifications.

3. Determine Complexity

Determining complexity is performed primarily by IT professionals, namely database administrators (DBAs) and Data Administrators. Among the many areas they will need to research are the following questions:

1) What DBMS are being used and what connectivity issues may exist?
2) Are the databases compatible?
3) How unique are existing data fields? Are the source tables' small relational tables or have numerous data fields? What Primary and Foreign Keys (PKs & FKs) are used?
4) How will exist data be merged?
5) Will new data use existing legacy systems or new integrated systems?
6) How does the data relate or interrelate with each other?

Many volumes of books exist on data modeling techniques, and we will not go into much detail here except to say that the IT professional will need assistance in correlating data from tables which have the same or similar names. One might think that a field called customers means the same thing to everyone, but sales people may consider anyone they have spoken with to be a customer while marketing may limit that term to those who have placed orders—each may use separate unique identifiers for each 'customer'. Another consideration is standardization of data and the need to keep key fields unique. For example, the company which hired one of the writers recently merged with a major corporation. In order to merge into a single payroll system, they needed to add two characters to the front of the employee ID in order to keep his employee ID pointing only to him and to ensure that all employee IDs had the same number of characters.

The Team will need to assist the IT professionals with these additional questions:

How many concurrent users will there be? This cannot be confused with, "How many total users?" Concurrent means how many may be accessing the data warehouse simultaneously. This impacts how powerful the processors need to be, how much memory is needed, and may indicate the need for distributed data marts.

Who is going to access the data warehouse and how will it be accessed? Internal-Use Only by employees? Are these employees in one location or at least operating off of the same network?

Will the users access via an Intranet, Extranet, or Internet? Employees typically will access from a business facility using an intranet. Business partners generally access using an extranet and customers using the internet. If non-employees are using it, what are the security requirements? Although many of these questions seem like a network issue, it directly impacts how the data warehouse and front-end software will implement security, what limitations on access to data is instituted, and what limitations on processing might be required (Jones 1998).

Enterprise? All of the above may impact on the licensing

requirements for the DBMS. Therefore, a thorough analysis is required and it is best to err on the higher side.

Cascading effects? This involves how strict relationships between data are maintained. For example, if a supplier is deleted from the database, should it delete any and all references to that supplier? Careful analysis is required here also, because if all references are deleted, then how would you track warranties on products purchased from that supplier? There are times when a cascading effect is beneficial. For example, keeping the primary record within the database, but coding it as deleted, could trigger the deletion of records in other tables which are no longer needed and simply clutters up the database. Another alternative would be to move those records to an archive database before deletion.

4. Determine the DBMS

Determining the Database Management System (DBMS) is also an important step in designing the Data warehouse. A data architecture covers the need for a DBMS, which is a system containing a set of programs that controls the organization of data within a database. It includes:

- A modeling language to describe the design of each database hosted in the DBMS according to the data model.
- Data structures optimized to tackle huge amounts of data stored on a data storage device.
- A query language to allow users to access and modify its data according to privilege levels.
- A mechanism to ensure data integrity.

The most popular Database Management Systems include Oracle®, Microsoft Access®, Microsoft SQL Server®, MySQL®, SQLite®, Sybase Adaptive Server Enterprise®, and DB2©. Each one of them has advantages and drawbacks. We will examine the basic features to look for when choosing a DBMS, and then compare and contrast the most popular ones.

4.1. Basic Criteria

For selection purposes, the following criteria are used:

- **Query ability:** Users must be allowed to interactively query the database and update it according to the users' level of privilege. It also controls the security of the database.
- **Backup and replication:** Duplicates of attributes must be constantly made in case the main storage devices fail.
- **Security:** Data security restrains illegitimate users from viewing or updating the database. It allows legitimate users to read, write, and execute procedures based on their individual or group privileges. It is better to limit the number of persons allowed to change attributes or groups of attributes.
- **Computation:** Each computer can rely on the Database Management System to supply computations, such as mathematical functions, sorting, and statistical analysis.
- **Automated Optimization:** If there are frequently occurring usage patterns or requests, some DBMS can adjust themselves to improve the speed of those interactions. In some cases, the DBMS can also provide tools to monitor performance, thus allowing the necessary changes to be made.

4.2. DBMS Comparison

Based on the users' requirements and IT recommendations, the choice of the DBMS is made. For example, a small business with fewer transactions might choose Microsoft Access, while a large corporation would consider a more powerful tool such as Oracle. MySQL is characterized as a fast, robust database with a good feature set, but one which lacks all the extras of SQL Server. On the other hand, Oracle is highly flexible, runs on many platforms and has a full and sophisticated feature set. This section highlights the advantages and drawbacks of the most popular DBMSs.

SQL: SQL Server can work with large databases and high numbers of concurrent users. SQL Server is less expensive than Oracle. Since SQL Server only runs on windows platforms, you are limited in your OS choice. Also, SQL Server is easier to administer compared to Oracle and since provides many data management tools (such as data transformation services, reporting services, and analytical services) for free.

Benefits

1) Fully web-enabled: Web-Enabled Analysis, Web Access to Data, Application Hosting, Security, Full-Text Search.
2) Highly scalable and reliable**:** Scalability, High Availability, Security, Distributed Partitioned Views, Indexed Views.
3) Fastest time to market: Simplified Database Administration, Data Transformation Services, English Query, Data Mining, Analysis (OLAP) Services.

While these advantages are appealing to any administrator, he must also consider some serious drawbacks that come with the use of SQL. As Tim Chapman (2006) demonstrated in his article *Explore the benefits and drawbacks of clustering SQL Server*, SQL drawbacks included but aren't limited to the following:

Drawbacks

1) One server will always be in standby mode and idle.
2) This configuration requires more licenses to be purchased than with an Active/Passive cluster.

3) Full-text indexing cannot be used to perform searches on large tables.
4) SQL-Mail does not support open standards like SMTP.

Oracle: Oracle is another leading commercial DBMS; it's highly flexible, runs on many platforms and has a full and sophisticated feature set. According to the article release ***High Availability Architecture and Best Practices (2004*)**, the following are the main benefits for choosing Oracle as your DBMS:

Benefits

1) Oracle Real Application Clusters (RAC) provide the following benefits:
   Node and instance failover in seconds, integrated and intelligent connection and service failover across various instances, planned node, instance, and service switchover and switchback.
2) Oracle Data Guard provides the following benefits:
   Disaster recovery, data protection and high availability, complete data protection, providing an easier and more efficient means for content providers to publish structured data and distribute it to customers running Oracle on a different platform, simplification of the distribution of data from a data warehouse environment to data marts which are often running on smaller systems with a different platform, enabling the sharing of read-only table spaces across a heterogeneous cluster
3) Recovery Manager (RMAN) provides the following benefits:
   Block media recovery enables the data file to remain online while fixing block corruptions, persistent RMAN configurations simplify backup and recovery operations, retention policy ensure that relevant backups are retained.

Drawbacks

1) Takes longer to learn and is not as simple, meaning fewer qualified professionals are available.
2) Doesn't perform as well as SQL server out of the box.
3) Costs a bit more if you don't include downtime cost.
4) Very expensive.
5) Security procedures issues.

MySQL: Professionals in the database industry adopt MySQL because of its ease compared to other DBMSs. It also has the advantages of being small in size in addition to its speed. For an in-depth look at its benefits, let's refer to the article *Benefits of MySQL* (2005). The article emphasizes five main reasons of why a DB administrator should choose MySQL.

Benefits

1) Security: MySQL includes solid data security layers that protect sensitive data from intruders.
2) Cost: It's obtainable by free download online.
3) Speed: To make it faster, MySQL programmers made the decision to offer fewer features compared to other major database competitors.
4) Scalability: This DBMS can handle any amount of data, as much as 50 million rows or more.
5) Memory: MySQL server has been thoroughly tested to prevent memory leaks.

As to the two previous DBMS, MySQL also has some serious deficiencies when it comes to MySQL Cluster. From the section navigation 16.10 of the article *Known Limitations of MySQL Cluster* we can spot the following drawbacks about using MySQL:

Drawbacks

1) SQL Syntax non-compliant: Temporary tables are not supported and indexes and keys in tables Keys and indexes on MySQL Cluster tables are subject to some limitations.
2) Limitation on Transaction Handling: There is no partial rollback of transactions. A duplicate key or similar error rolls back the entire transaction.
3) Error Handling: Starting, stopping, or restarting a node may give rise to temporary errors causing some transactions to fail.

5. Determine the Interface Software if Needed

Data warehouses are great, but are absolutely useless if end users cannot get the information they require, when they require it, and in the format in which they need it. This is where application architecture is required. The application architecture will create a plan to bring the data from the data warehouse to the user.

Since this article is directed towards data warehousing, we will not spend a lot of time or depth on the numerous types of interface software which currently exist. We will however, describe various types and cite an example or two.

COTS – or Commercial off the shelf packages. These are software packages designed to be used 'as-is' with a high success rate. A simple example of COTS which most everyone is aware of is Microsoft's Word and Excel.

MOTS – or Modified off the shelf packages. These packages tend to use open source code or code which the user can modify to meet their needs.

Most anything else requires hiring programmers to develop your interface system from scratch.

Before the IT professionals can determine which ones to recommend, they need to know all of the possible outputs the end users will need. What printed reports are needed? Are they all on standard 8.5 x 11 or 8.5 x 14 inch article, or will you need to do larger printing forms? Will you need to output to wireless devices like Personal Digital Assistants (PDA) or cell phones?

Cognos® is one example of COTS. They sell a suite of products designed such that your need for IT professionals is kept to a minimum. Many of their products use drag and drop technology to make the ease of end users customizing and designing outputs a priority. Cognos, version 8, has OLAP cubes technology, graphical output, and ad-hoc query capabilities. Cognos is designed as a read-only software suite, which means that the user cannot write to the database using Cognos.

Macromedia's Coldfusion® is an example of a MOTS package. Coldfusion can interface directly with a database in both a read and a write mode. Coldfusion requires programmers who are familiar with HTTP, Javascript, and basic programming techniques. Programmers design and build the interfaces used by the end users.

Microsoft's .Net (pronounced dot-net) and C++ are examples of home-grown products. These are more programmer-intensive solutions for an interface.

In order to determine the best fit, the IT professional needs to review all of the information gathered thus far and meet with the management team to determine the degree of customization required.

6. Determine the Hardware Requirements

Now that you determined the preceding items, what hardware do you need to support your systems? A technical architecture plan will outline the infrastructure needed to support the data warehouse. Again, we will give an overview of environments to consider and some of the pros and cons within each.

To begin with, we suggest using a three-tiered production environment—using three servers. Keep your database server separate from your web server, and keep these separate from the ETL processing server. The ETL server handles Extracting data from its source servers; Transforming that data, and Loading it into the DWH DBMS. Based upon the size and complexity of your environment, attempting to run all three or any combination on less than three servers will impede performance. While the database is running complex queries and operations, it will consume as much processor and memory as it can, which may degrade your web's response time.

The other environments to consider are a development environment and a test/training environment. The larger and more complex the production environment is, the more you need to keep development separate and the more it will assist to have a test and training environment.

When money is a major consideration, the best equipment should be assigned to production, development, and then test and training. You should, however, try to keep to the same configuration in each environment to make troubleshooting easier.

Determining what types of hardware and which operating systems to use can be a personal choice, but we will give you some information to consider. The major operating system platforms are Windows-based, UNIX based, and Linux based. Windows is by far the dominant brand; however, it is also the most attacked by viruses, hackers, etc. UNIX and Linux are similar in nature and are often considered the most secure; however, the number of available software packages is limited.

The COOP environment is discussed later under Support.

7. Support

Now that the first three architectures are defined along with the logical design, it is time to plan the support and protection of all your efforts and funding. There are several issues you need to address which will ensure the success of your data warehouse.

7.1. Disaster Recovery

You must develop a disaster recovery plan: whether you build a data warehouse or not, you need a solid disaster recovery plan. Far too many companies, even major corporations, fail to adequately plan to recover from a disaster.

The first plan to consider is the back-up plan. All data must be backed up on a regular basis, and the best place to back it up to is somewhere other than the primary location. The off-site should be far enough away that it cannot be affected by the same disaster that affects the primary site. The farther away, the more sophisticated the methodology must be, since bandwidth will become a major factor.

The second plan is how to recover the back-up; this will depend on the cause of the original failure. There are numerous methods for recovery, but you should plan on the possibility the entire primary system is off-line. For these situations, consider a COOP (Continuity Of Operations), which is a mirror image of the production environment that can immediately become operational with all DNS (Domain Name System) servers pointing to the COOP without any requirements on the part of the user. A COOP is an essential part of success. It is one of those pieces of insurance you hope to never use and may never use, but if needed could mean the survival of your business.

7.2. Version Control & Configuration Management

Another issue to address is managing the versions of the software and coding written to ensure integrity and to ease rollback capabilities in the event that a newer version fails to perform as planned and tested. One piece of this is version control. Whether you use Microsoft's Visual Source Safe© or an Open Source version, your IT professionals should investigate and recommend some form of version control.

Configuration Management (CM) is the business plan used to ensure adequate controls are in place for managing and maintaining the system. CM covers such areas as defining the base-line, installation of upgrades and patches,

recoverability, and virus protection. A simple search of the Internet will result in a large sampling of CM Plan templates.

7.3. Performance Monitoring

A system of integral as a data warehouse requires constant performance monitoring. Whether the database server stops functioning due to processor overload or the web server maxes out the available memory, built in safeguards and notifications are needed.

A thoroughly written and implemented support plan will insure your investment against preventable failure.

8. Implementation

You should now have the four architectures as described above to go with an identification of scope and complexity. Hopefully, by this time, your IT professionals will also have some prototypes built, and it is now time to begin an implementation phase. Now is the time the data warehouse can be optimized to serve the users. Having completed enough of the data modeling to create broad-spectrum tables, routines, and stored procedures, you can begin populating the data tables with sample data from the primary sources. At the same time, you need to be writing a thorough test plan that we will discuss a little later.

Here, the development of the database schema commences, metadata is defined, and source data is expanded to include requisite data for the area covered in the initial project. Source styles and techniques being previously selected and documented, begin following the plan.

As demonstrated in the article *Process/Project DWH - Data Warehouse Process* (2007) published by the online community for IT project managers, Gantthead, the design stage must be carefully planned before starting the implementation, because it is during this stage that the physical data warehouse model is developed and the source data inventory is updated and expanded to include all of the necessary information needed for the implementation project. The article adds that the following procedures are necessary to achieve the above activities:

- Warehouse Capacity Growth
- Disaster Recovery
- Data Extraction /Transformation/Cleansing
- Data Archiving
- Data Load
- Help Desk
- Security
- Transition to Production
- Data Refresh
- Configuration Management
- Data Access
- User Training
- Backup and Recovery
- Testing

Once the design stage is complete, the project to implement the current Data Warehouse iteration can proceed quickly.

The preferred method of implementing is to begin with one subject area selected from the plan laid out by the team. Although you are beginning with one subject area, do not exclude subsequent areas that might be affected by this one. For example, even though HR may use some method to identify locations, the real estate management division will also need that information as will shipping and receiving. The schema should take all possible uses into consideration.

Your IT professionals should use metrics to capture statistics for all of the processes. Use these metrics to quantify your server efficiencies and effectiveness as you build the data warehouse. Once the programs are developed and tested, all components are in place, and the system, security, and back-up procedures are established, begin training and deployment. Ensure the team and all IT professionals routinely meet to discuss progress using definable metrics.

Finally, as new routines are designed and implemented, you will also begin the maintenance phase. As covered previously in the support architecture, you must have policies and systems in place to protect the data; this cannot be emphasized strongly enough.

As the data is imported, the data needs validation. Some data may not be capable of validation due to bad source data. If the originating source system failed to have adequate data validation procedures in place, then there is no telling what the data will be like. For example, I have seen year-of-manufacture fields with ‘abcd’, ‘xxxx’, ‘####’, and ‘1905’ (instead of 2005). It is in these instances that management may choose to attempt to correct the error or exclude it. However, users must be aware of the old adage: garbage in, garbage out. If erroneous or even bizarre data is seen within the data warehouse, it is not necessarily the fault of the IT professionals building the data warehouse but the direct result of the source data. Using the example listed above, when asked what the average age is of a fleet of vehicles, the DBMS will either ignore the data which cannot be translated into a real year or the procedure will fail. If this is the year 2007 and the date of manufactured is listed in the source system as 2010, it will skew the data. This is the failure of the source system, not the DWH.

Lastly, the users need training to become familiar and comfortable using the new data warehouse. They need to see how meaningful and easy data mining, analysis, and decision support is within their reach. Without the buy-in of the end user, the data warehouse is useless to the firm.

9. Conclusions

This article took a ***practical managerial approach***, and we trust it meets the needs of managers looking for a data warehouse solution. Although the entire data warehouse system is a complex environment, the managerial approach to developing one is not. The manager need not be afraid of

the data or of the complex systems involved. The most important role that managers play in the development of data warehouses is to develop a solid team of users to assist the IT professionals in answering mostly simple, basic questions. This team may or may not understand the programming algorithms which go into creating the solutions, but they must provide the IT professionals with the mathematical formulas and the data elements used to create the answers. For example, if the program should calculate back-log, the IT department needs to know what data elements are used and how back-log is calculated when performed manually. They also need the corporation's 'buy-in' at the highest level, understanding that they may not be capable of pinning down a return on investment (ROI) for the DWH in terms of real dollars. However, careful analysis of cost avoidance through the use of the DWH will give them some indication as to its importance.

Managers also need trusted IT professionals who have some understanding of the company's purpose and operations, and who can grasp what the Team is trying to say. The better the DBA (Database Administrator) understands the data as well as the inter-relations of the data, the more efficient and cleaner the outputs will be.

Managers should understand that the less complex each record is, the faster and easier it is for the IT professionals to return the results needed for each query. In the IT arena, we refer to this as how flat or how wide the record is. If every possible piece of information is placed in the base record, then the database must read that entire record before moving to the next, even when you do not want that piece of information. That is why we recommend the use of reference codes as much as possible. When that data is retrieved through the use of star-schemas or snowflake schemas, the IT professional can return exactly what is needed quickly and efficiently.

Finally, remember that it takes time and that the data warehouse is always evolving, growing and enhancing. The data warehouse is what the corporation makes it into. Its worth may not be measurable in real dollars, but the cost savings can be real and immense.

Notes

The authors of this article take no stand on specific recommendations on software or operating systems. It is our intent to compare several systems to highlight specific advantages and disadvantages for managers to review and ask questions.

Definitions

Star-Schema is a database schema typically styled a simple fact table (Employee ID and Employee Fullname) and then joining to many dimension tables using the primary key of the fact table (Employee ID) where all dimension tables' center around the single fact table. The primary key of the fact table points to the foreign key of the dimension table. Dimension tables may be Employee Personal Information (Employee ID, SSN, address, phone numbers, date of hire), Department Information (Department ID, Department Name, Employee ID), Personnel Evaluations (Employee ID, last evaluation date, last evaluation grade, current pay scale), Finance Payroll (Employee ID, current salary, vacation days accrued, number of deductions).

Snowflake Schema is a database schema similar to the star-schema but with more fact tables and more dimension tables, thus looking more like a snowflake than a star.

Large Flat Tables are the least favored method of storing data. Instead of fact and dimension tables, all relevant data is stored in each record. This creates tables with numerous data elements, and the database must read each record in full before moving to the next record. The preferred method is to have fact tables consisting of a primary key and some foreign keys, which, when queried, return the entire piece of information.

Drill-down is a data warehouse term which refers to the ability to start at a summary piece of information and through mouse or keyboard entries work your way down to the lowest detail available.

DTS (Data Transformation Services) is a Microsoft automated ETL process to move data to or from a database. DTS allows Microsoft SQL Server to interface with heterogeneous sources using ODBC and OLE-DB (another Microsoft product easing the translation between Microsoft database objects).

Primary Key (PK) is a unique key in a table that may relate to a foreign key in another table in order to create a point of reference when joining the tables. Primary keys must be unique, cannot be null (empty), and should be the least possible size.

Foreign Key (FK) is a referential key relating data between two or more data tables. The FK can be considered a code which, when queried from the referencing table to the referred table, returns the longer answer. For example, sending the zip code 30067 to the zip code table will return Marietta, GA, and may also give boundaries and population demographics for that zip code.

OLAP (Online Analytical Processing), or Cubes, is a multi-dimensional method for calculating data (numbers) based on fact and dimension tables and predefined measures. The purpose of an OLAP Cube is give numerical answers quickly without putting a strain on the data warehouse. The OLAP Cube calculates all possible permutations of aggregated data defined within the cube. OLAP Cubes are sometimes thought of in the same manner as pivot tables. Since all possible answers have already been calculated, the

cube can typically return answers in 0.1% of the time required for the database to re-query the information. Another benefit is that OLAP allows aggregation of data across time dimensions without having those dimensions previously built in (Week, Month, Quarter, Fiscal or Calendar Year).

A good example of what a cube can do might be reporting new auto sales. The cube will have calculated sales of all autos by exterior color, interior color, interior fabric, make, body style, size, category, location, sales person, price range, tire brands, and engine size. The database will not need to run a query on how many red, 4-door sedans with tan interior leather and Pirelli brand tires were sold in the south-eastern United States over the past 6 quarters by age and gender of the sales force—that calculation will have already been computed by the cube.

Acknowledgements

This article is a collaborative effort between Larry Thomas and Patrick Akpess, Data Warehousing professionals/managers and their perspective Master of Business Administration advisors, Max North and Ronny Richardson. Both advisors acknowledge their extraordinary technical and practical experiences contributed to this article.

REFERENCES

[1] Boehnlein, M, & Ulbrich-vom Ende, A (1999). Deriving initial data warehouse structures from the conceptual data models of the underlying operational information systems. *Proceedings of the 2nd ACM international workshop on Data warehousing and OLAP DOLAP '99*, ISBN: 1-58113-220-4.

[2] Bog, A., Sachs, K., & Zeier, A. (2011, March). Benchmarking database design for mixed OLTP and OLAP workloads. In *Proceedings of the 2nd ACM/SPEC International Conference on Performance engineering* (pp. 417-418). ACM.

[3] Bonifati, A., Cattaneo, F., Ceri, S., Fuggetta, A., & Paraboschi, S. (2001). Designing data marts for data warehouses. *ACM transactions on software engineering and methodology*, *10*(4), 452-483.

[4] Chapman, T. (2006). *Explore the benefits and drawbacks of clustering SQL Server 2000*. Retrieved on 2/8/2016 from http://www.techrepublic.com/article/explore-the-benefits-and-drawbacks-of-clustering-sql-server-2000/6130580/

[5] Chhabra, R., Kumar, P., & Pahwa, P. (2016). An approach to Design Object Oriented Data Warehouse. *International Journal of Research and Engineering*, *3*(3), 54-56.

[6] Coronel, C., & Morris, S. (2016). *Database systems: design, implementation, & management*. Cengage Learning.

[7] Ganthead.com (2007). *Process/Project DWH - Data Warehouse Process.* Retrieved on 11/24/2015, from http://www.projectmanagement.com/content/processes/9076.cfm

[8] Dedić, N., & Stanier, C. (2016). An Evaluation of the Challenges of Multilingualism in Data Warehouse Development. In *18th International Conference on Enterprise Information Systems-ICEIS* (p. 196).

[9] Golfarelli, M., & Rizzi, S. (2009, November). A comprehensive approach to data warehouse testing. In *Proceedings of the ACM twelfth international workshop on Data warehousing and OLAP* (pp. 17-24). ACM.

[10] Hoffer, J., Venkataraman, R., & Topi, H. (2016). *Modern database management*. Pearson Education Limited.

[11] Idreos, S., Papaemmanouil, O., & Chaudhuri, S. (2015, May). Overview of data exploration techniques. In Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data (pp. 277-281). ACM.

[12] Jarke, M., Jeusfeld, M. A., Quix, C., & Vassiliadis, P. (2013). Architecture and Quality in Data Warehouses. In *Seminal Contributions to Information Systems Engineering* (pp. 161-181). Springer Berlin Heidelberg.

[13] Jarke, M., Lenzerini, M., Vassiliou, Y., & Vassiliadis, P. (2013). *Fundamentals of data warehouses*. Springer Science & Business Media. ISBN: 3-540-42089-4

[14] Jones, K. (1998). An introduction to data warehousing: what are the implications for the network? *International Journal of Network Management,* 8(1), 42-56.

[15] Kimball, R. (2002). The Anti-Architect: How not to design and roll out a data warehouse. *DB2 Magazine*, *502*.

[16] Lafuente, B., Downs, R. T., Yang, H., & Stone, N. (2015). The power of databases: the RRUFF project. *Highlights in Mineralogical Crystallography*, 1-30.

[17] Malhotra, N. (2015). Implementation of Data marts in Data ware house. International Journal of Advance Research, Ideas and Innovations in Technology, 1(2), ISSN: 2454-132X.

[18] MySQL AB (2007). Known Limitations of MySQL Cluster. Retrieved on: 10/26/2015 http://dev.mysql.com/doc/refman/5.0/en/mysql-cluster-limitations.html Section 16.10

[19] Nguyen, T. B., Wagner, F., & Schoepp, W. (2014). Federated data warehousing application framework and platform-as-a-services to model virtual data marts in the clouds. *International Journal of Intelligent Information and Database Systems*, *8*(3), 280-294.

[20] Novell, Inc. (2005). Benefits of MySQL. Retrieved on 1/5/2016 from: https://www.novell.com/documentation/nw65/web_mysql_n/data/aj5bj52.html

[21] Oracle® Database (2004). *High Availability Architecture and Best Practices*. Retrieved on 10/31/2007 fromhttp://download.oracle.com/docs/cd/B14117_01/server.101/b10726/hafeatures.htm

[22] Radhakrishna, V., Kumar, P. V., & Janaki, V. (2015, September). A Survey on Temporal Databases and Data mining. *In Proceedings of the The International Conference on Engineering & MIS 2015* (p. 52). ACM.

[23] Smith, A.M. (1997). *How to Create a Data Administration Mission Statement*. Retrieved on 2/19/2016, from http://tdan.com/how-to-create-a-data-administration-mission-statement/4143.