www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 7 Issue 12 December 2018, Page No. 24422-24429
Index Copernicus Value (2015): 58.10, 76.25 (2016) DOI: 10.18535/ijecs/v7i12.02

Design and Implementation of an Enterprise Data Warehouse

Arshad Gulzar

Abstract

The reporting and sharing of information has been synonymous with databases as long as there have been systems to host them. Now more than ever, users expect the sharing of information in an immediate, efficient, and secure manner. However, due to the sheer number of databases within the enterprise, getting the data in an effective fashion requires a coordinated effort between the existing systems. There is a very real need today to have a single location for the storage and sharing of data that users can easily utilize to make improved business decisions, rather than trying to traverse the multiple databases that exist today and can do so by using an enterprise data warehouse.

The Paper involves a description of data warehousing techniques, design, expectations, and challenges regarding data cleansing and transforming existing data, as well as other challenges associated with extracting from transactional databases. The Paper also includes a technical piece discussing database requirements and technologies used to create and refresh the data warehouse. The Paper discusses how data from databases and other data warehouses could integrate. In addition, there is discussion of specific data marts within the warehouse to satisfy a specific need. Finally, there are explanations for how users will consume the data in the enterprise data warehouse, such as through reporting and other business intelligence.

This discussion also includes the topics of system architecture of how data from databases and other data warehouses from different departments could integrate. An Enterprise Data Warehouse prototype developed will show how a pair of different databases undergoes the Extract, Transform and Load (ETL) process and loaded into an actual set of star schemas then makes the reporting easier. Separately, an important piece of this paper takes an actual example of data and compares the performance between them by running the same queries against separate databases, one transactional and one data warehouse. As the queries expand in difficulty, larger grows the gap between the actual recorded times of running that same query in the different environments.

Introduction

Data is harder to analyze when it is fragmented and is stored at multiple places. An enterprise data firm multiple sources, giving the users to access to the right information so that they can take necessary action. The vision for this paper is to study components of a theoretical Enterprise Data Warehouse within the context of a higher education environment. The reason for such an implementation would give users at all levels of the university an integrated, secure and consistent data source from which they could report on and set their business needs more efficiently than possible without one. In this process we take the following steps.

1) DWH Staging(Data Warehouse Staging)

This is the temporary location where data from source system is copied. All required source data is made available before data is validated/transformed/transferred into the Data warehouse. Thus having an staging area considerably reduces the risk of data loss at any point of time. Staging is designed to keep original copy of data for further use or for troubleshooting purposes.

2) Data Types and Mapping

It is used as a first step for a wide variety of data integration tasks including data transformation/data mediation. We perform an analysis of the data in order to get an accurate account of the different types of data being stored in the data warehouse and the nature of the different sources being used. It includes identifications of attributes, size, data type, allowed values and mapping between the source, staging and Data warehouse model.

3) Extract Transform and Load (ETL)

The Extraction Transform and Load processes the inconsistent data, cleans "bad" data, and then loads data from source to staging and then from staging to target database. All three phases are scheduled and executed in parallel.

Problem

The implementation of an Enterprise Data Warehouse, in this case in a higher education environment, looks to solve the problem of integrating multiple systems into one common data source. With the diverse roles that a college has both on the academic and non academic sides of the business, data about a person, whether they are an employee, alumnus, student, donor, or possibly all of the above, are more likely to reside in different transactional databases across the campus. Much of the information exists in a silo in and of itself, and the information does not translate across the spectrum to help the business of higher education grow. An example of this may be that the admissions department has information about the occupation of an incoming student's parent, information that may be important to the fundraising department since the parent's income from this occupation could possibly lead to a boon in donations; however, this useful information is not shared between these two departments. Certainly, having a data warehouse that shares this kind of information with the masses could cause internal strife or possible breaches of security. Therefore, devising a plan that restricts data, as appropriate, makes reasonable sense. One universal problem of not having an Enterprise Data Warehouse is how users consume the data in the form of actual reports. So often, those who need the information or require knowledge from the data they utilize must wait for a report based on someone else's schedule. Furthermore, once they get the report, they may have to manipulate data within an application such as Microsoft Excel to fit their needs, and this can often lead to error or miscommunication. As both public and private organizations recover from the corporate fraud of the last decade, it is now more important than ever to report results correctly, which calls into question the current procedures. Rather than executives or managers having their morning coffee waiting for the delivery of reports, it should be possible for them to be able to go get their own information when they need it.

Explanation

A major component of Business Intelligence is the use of a data warehouses for making better decisions based on the information it provides. The data warehouse has a history of storing data in a way that is optimal for reporting and other analysis, such as OLAP, or Online Analytical Processing. The data warehouse uses a combination of efforts of hardware and software not only to acquire the data but also to present data for users to access it in their desired format. Once the data is in a data warehouse, there is the availability to develop them into data marts, which specialize in the arrangement of data for specific business purposes. Finally, there are hosts of reporting tools available that make the data warehouse a user-friendly experience for those who want to pull the data themselves rather than waiting for distribution from others. The creation and evolution of the data warehouse make it an invaluable tool that makes Business Intelligence possible.

Motivation

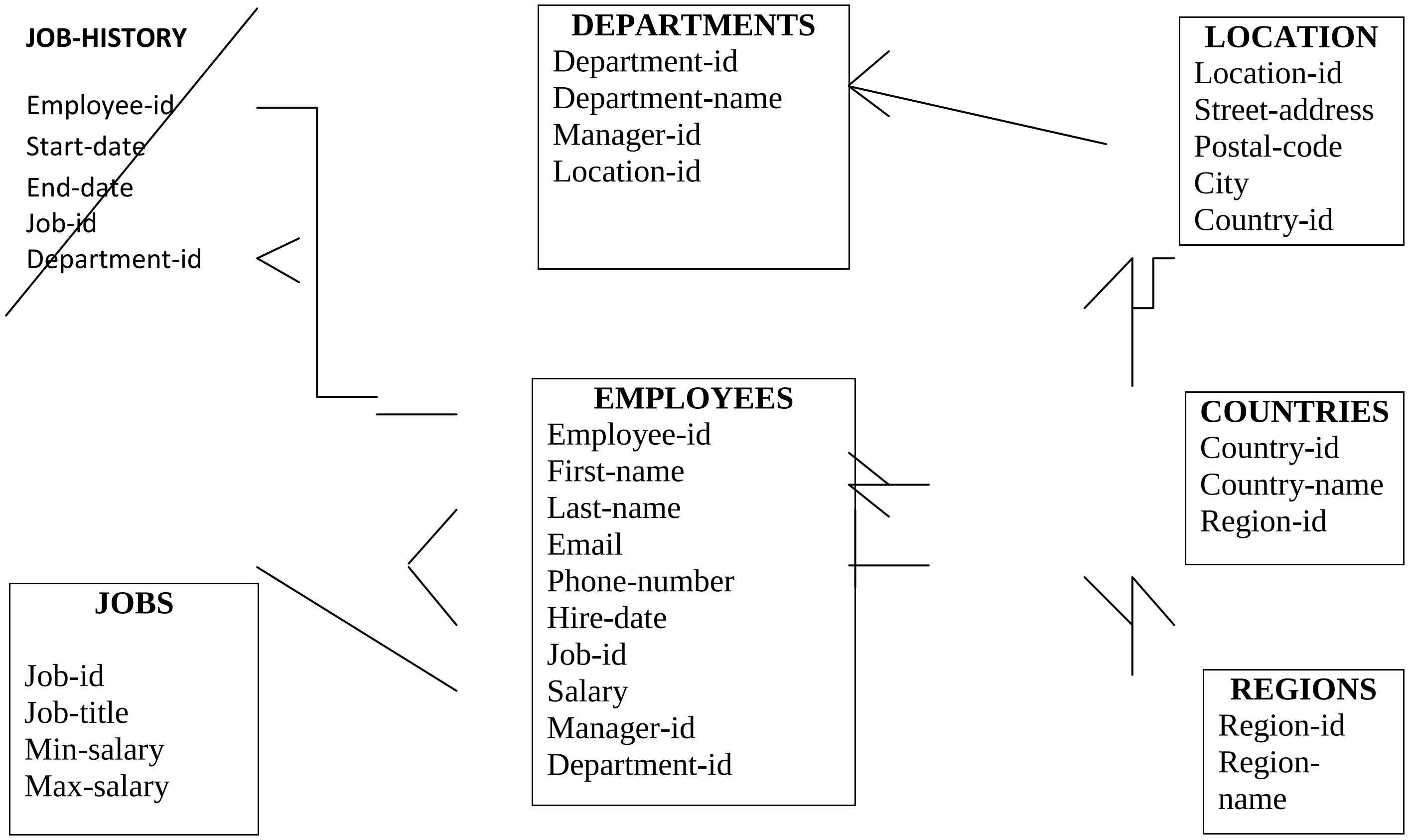
There are many contributing factors involved when considering the implementation of an Enterprise Data Warehouse. Although executing such a project could require a significant time, resource and/or monetary investments on the part of a company, there are many motivating factors to move forward with the implementation of such a project. The most significant motivation to implement a data warehouse is to have a better platform on which to report data. Combining the data from all the other databases in the environment, the data warehouse becomes the single source for users to obtain data. All reporting would be based on a single database, rather than on individual repositories of data. Using a data warehouse, a report writer does not need to learn multiple databases or attempt to try to join data between more than one database, a task which may prove difficult. In addition to having one single database to report from, there are other advantages to not reporting from transactional databases. Namely, reporting from the transactional database can slow down the general database performance by taxing resources already allotted to the system. Running such a report against certain parts of a database could potentially cause slower experience for the user, because it could render a database or the application using it unresponsive. Finally, data can be in the process of an update or delete transaction in the database, which could render the report incorrect while it is running. Another advantage to completing reports against the data warehouse is the fact that there can be a universal reporting tool and report distribution system deployed in order to allow a single platform for users. The advantages of such a tool would allow for ensuring timely distribution, consolidation, management and distribution of data as well as the ability to maintain an audit trail of the data. Once the users have the data from the data warehouse, they can work with the data in order to make better decisions for their business. Data presented in a data warehouse is available for massaged by users in which users can work with data in Excel, Power Pivot, pivot tables based off OLAP, cubes and Key Performance Indicators (KPIs). By using warehouse data, just about anyone can create complex models, build charts and calculations, manage a variety of reporting functions, analyze and make decisions. Besides making reporting easier for a user and having data better available for consumption, from the Information Technology (IT) department’s point of view, the data warehouse will make it easier to control security. Rather than controlling security on each database for scores of users, with the data warehouse, there will be just one database to grant/deny users and rights. Furthermore, IT will have control and manageability over enterprise data that users can then access to fulfil their business needs. Further helping control the IT department’s manageability of databases, the data warehouse affords the opportunity for smaller databases to have their data represented in an Enterprise Data Warehouse. Most transactional databases have no associated data warehouse platform and there would never otherwise be a reason to have a data warehouse as part of normal business operations for that database. Even if there was a need for the warehouse for that particular database, there might be too much time, resources or consulting fees to develop just one data warehouse, which would be helpful to only a small group of users. By having an Enterprise Data Warehouse, even a small database could be included as part of a large-scale solution. Perhaps, one of the most underrated reasons for a data warehouse is the fact that it is able to maintain historical data. A basic transactional database does not keep historic data as well as a data warehouse does because a user normally quickly updates or deletes this data on a regular basis during the course of a day. In a basic transactional database, what a value was only hours before could disappear forever, and there would be no record of its existence. A data warehouse can accumulate old data and potentially store the data forever. As data change and rules adapt to directives, the storing of historical data may lend a chance to snapshot data within the data warehouse and, as a result, store earlier behaviours as they relate to data for current or future users to report upon. The final reason for developing an Enterprise Data Warehouse is that a reasonable data model exists deployed for users so that consistency is the key to consuming the data across the business. Regardless of the source of data, the data model gives users a common way to get the data they need to satisfy their own purposes. This data model exists for use across the business to enhance the need for getting data in reports to make better decisions. Now more than ever, businesses rely on accurate, secure and up-to-date information in their reports to help their business operations and comply with increasingly stringent regulations. Having a common source of data for users and taking away the burden reporting places on a transactional database can improve the efficiency and data sharing across the business. The solution of the data warehouse replaces Excel and other reporting platforms with a modern-day, centralized reporting and analysis solution. The ability to consolidate all significant and related data for a single version of the truth is fundamental to reporting accuracy. Such an example for having more accurate reports has been since the implementation of the Sarbanes-Oxley legislation, making compliance mandatory.

Implementation

This section includes a sample implementation of an Enterprise Data Warehouse on a significantly smaller scale that has only two star schemas. The solution shown here is an example of using the Business Intelligence tools with SQL Server and a SQL Server database, the same software that is appropriate for a full-scale implementation.

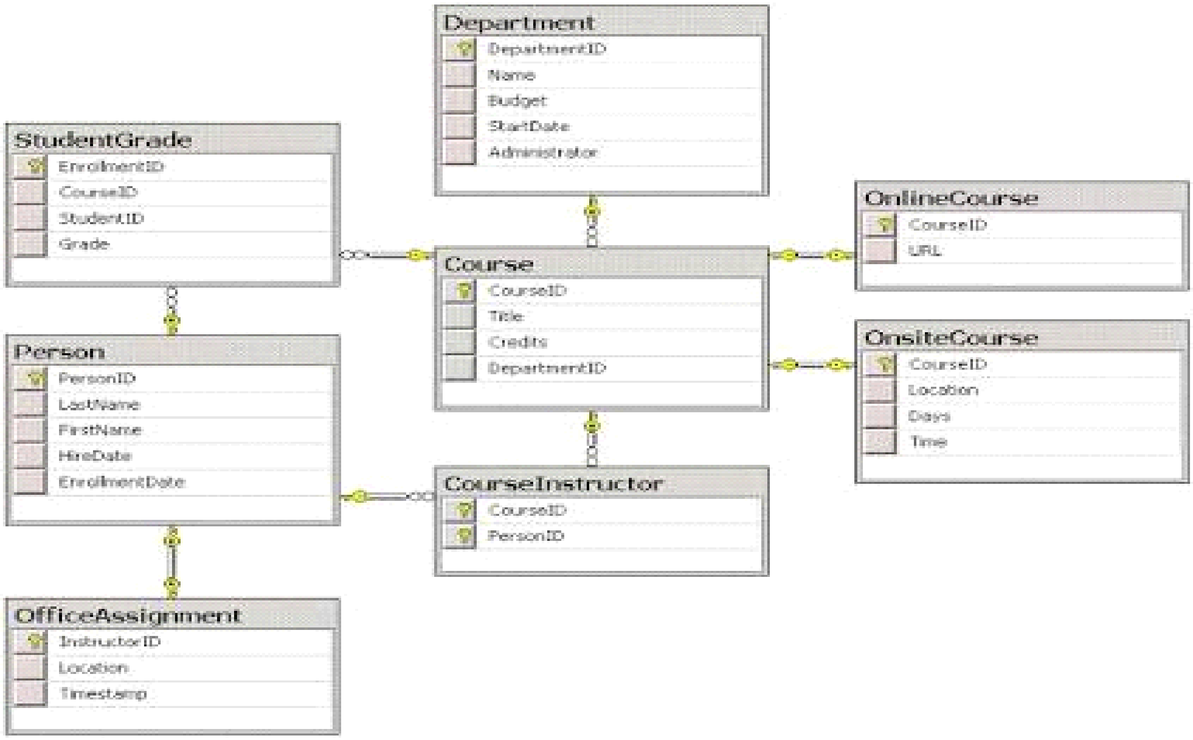
Enterprise Data Warehouse Prototype

The culmination of the paper comes together with a prototype of an Enterprise Data Warehouse in a higher education environment. The purpose if this initial design of the Data Warehouse is to show not only how a series of tables can become a denormalized star schema, but also to show how an enterprise data warehouse can integrate data between two star schemas from two different systems all together. Although this is on a smaller scale, the prototype will depict two transactional databases commonly found on a college campus, which are vital to the community in which they serve. The next step is to take this data out of the database by extracting it, then transforming it to put it a logical grouping, then finally loading it into an organized star schema. To show the user what is available, there are a few simple reports that will show data that a user at the college may find useful. The report illustrates the unique opportunity that a user can have in one report. The data that began as two very different databases and are in two different star schemas still are able to relate within the data warehouse. As the prototype illustrates, the ability to link data could be helpful in situations where one would need to have a comprehensive report that would otherwise come from report writers in separate departments. It bears noting that this is the simplest of examples for the creation of an Enterprise Data Warehouse, but it shows the concept of the bigger picture of implementing a data warehouse. The final data warehouse is a SQL Server 2008 R2 database, which uses SQL Server Integration Services as the ETL tool and SQL Server Reporting Services as the reporting tool. The first transactional database is the “HR” database, which comes with an Oracle XE 10.2 (Express Edition) installation. As depicted in Figure 6, the HR database portrays a database that would keep track of employees and job assignments as well as salary and personal information. Within this HR database, “each employee has an identification number, e-mail address, job identification code, salary, and manager”. This database used in a college environment could not only to track regular employees but also the faculty who are also in the student system, which will be mentioned next.



(Hr Schema from ORACLE Express Edition)

The other transactional database that is part of this prototype is a database named “School,” which is an example database that portrays a student system, a vital part of any college. This database keeps track not only of students but also instructors, classes, and locations, and it also has the potential for expansion if necessary. The database is in a Microsoft SQL Server 2008R2 relational database management system, hosted on a Windows operating system, as would typically be the case with this type of database software. The database is available from the Microsoft Developer Network, which shows the relations of a theoretical student system as well as installation instructions. In Figure below, the schema is as designed as follows:



The design of the data warehouse will involve a star schema for each transactional database, with the student system represented as the enrollment data, whereas the HR database represents the employee. The dimension tables are direct from the database tables, adding a dimension ID that is an identity-valued field and incrementing one by one. Some of the dimension tables will have data from other tables, such as a lookup table. The Fact table will have another unique Fact ID with all the related Dimension IDs along with the vital measures that will easily report data. For the sake of this example process and because of the size, all of the destination tables need to start empty before being loaded with data. The idea is to normally increment rather than loading the entire data warehouse with each run. However, due to the size and purpose, this data warehouse will not involve every detail of a data warehouse, but it will still convey the basis design and result.

Performance Evaluation Overview

The Enterprise Data Warehouse is able to show how reporting can be improved, but it cannot offer a chance to compare the same report and query against an existing system. The purpose of this exercise is to take a real world example of the possible improvement in query time, comparing a transactional database and a denormalized star schema in a data warehouse. In order to make the data warehouse a viable option and to make it a worthwhile endeavor, it is important to show that there will be a big improvement in query time. This effort involves solving a problem that existed where the solution was to create a data warehouse star schema. In a large database, it is especially time consuming to run reports against due to the complexity of the view and multiple joins to other tables. Due to the complexity of the data, I took apart the view piece by piece and designed a working ETL and arranged the tables into a star schema. The goal was to have a fact table that matches the count of records as the view does so that the exact same data that is present in the results of the database view would also be in the fact table. The fact table links to a collection of dimension tables, which contain the bulk of the attributes that the view returns. The expectation in this testing set is that the data returned from the data warehouse will be less time consuming than from running the same query against the transactional database. The reason that there is a probability for a difference in the query times is due to the essence of the organization of the data warehouse. The source system, or transactional database, is a typically large "normalized" database designed, "to organize data into stable structures, and thereby minimize update anomalies" (Sanders et al. 1). In the star schema used for this exercise, the data appears in a denormalized state, which is typical in the design and implementation of star schemas, organized to optimize querying. In this example, one of the star schema dimension tables built by the ETL uses a denormailization technique, which collapses tables, through joining by one to one relationships, adding only the necessary columns into a dimension table rather than the need for completing additional joins at runtime. For example, one of the main dimension tables used for the warehouse testing set collapses five tables from the transactional system to one, using only the necessary columns from each for the query. Having the data from one or more source tables organized into a logical dimension table will improve query results in the warehouse by having the columns readily available, having had the ETL complete the table join processing up front. By using a the view in the source system that was developed to query data for a report, it attempts to denormalize the data for the purpose of the report, but "since most DBMS products process view definitions at run time, a view does not solve performance issues". The desire to move away from the complexity of this or a similar view that used as part of this testing set could provide the ultimate motivation for using a data warehouse to complete reporting for users. It bears noting that the ETL for this star schema runs for approximately two minutes as it executes the SQL that the view does, but it runs in separate sections for the dimension tables then runs in for the fact table. Two natural downsides to having this data warehouse star schema is the time it takes to load the data plus the fact that the data will only be up-to-date as of the last refresh of the ETL. Since there is an initial time investment to load the data warehouse, it reasons that since the same time is necessary to run the query as to run the refresh of the data warehouse, there is no real reason to do so in the first place. However, on a larger scale, if this data is to be repeatedly queried and reported upon all day, having a better performing system overall is a logical choice.

Conclusion

This paper introduced the idea of the implementation of an Enterprise Data Warehouse within the context of a higher education environment, in order to better integrate systems for simpler and improved reporting. In order to implement a data warehouse, it is necessary to have a strong business commitment spearheaded by an executive sponsor who believes in the project and is responsible for keeping the project team motivated. The IT department becomes involved to execute the project to install the data warehouse into the enterprise and integrate it with the existing systems. The Enterprise Data Warehouse implementation is most effective using the up-to-date techniques of data modeling and studied practices as perfected through the years, which is the basis for data warehousing worldwide. By organizing data into a collection of star schemas in the form of dimension and fact tables for efficient organization by the need of the business, user reporting improves. When it comes to reporting, there are many improved tools available instead of the simple ad-hoc reporting or query tools, as enterprise reporting and dash boarding becomes a possibility. The paper concludes by studying some actual studies, involving construction of an Enterprise Data Warehouse example as well as studying an actual case of improving queries. By designing and implementing small scale, downsized Enterprise Data Warehouse, it was simple to see how star schemas assemble from an organized and efficient ETL tool from a transactional database out of multiple sources. With the practical exercise, we looked into running the same query against the same data but in very different systems. When comparing the results of running the queries, it is clear in every instance that the data return dramatically quicker from the organized star schema in the data warehouse than from the transactional database. Such practical examples as well as the study of data warehouse architecture and reporting capabilities, show the advantages of implementing an Enterprise Data Warehouse for an improved experience for users who are querying and reporting on data for better-informed decisions.

References

[1] Adamson, Christopher, Star Schema: The Complete Reference. New York: McGraw Hill, 2010.

[2] Bauwens , Christian. Et al. "Oracle Database Sample Schemas

[3] The Data Warehouse Lifecycle Toolkit. Second Edition. New York: Wiley Publishing, Inc.,2008.

[4] Rogers, Denise. “Data Warehousing Architecture - Designing the Data Staging Area.”

[5] Whitehorn, Mark. Et al. “Best Practices for Data Warehousing with SQL Server 2008 R2.”

[6] Kimball, Ralph. Et al. The Data Warehouse Toolkit. Second Edition. New York: Wiley Computer Publishing.