# 

Enterprise Data Warehouse Proposal for ABC University

# Shahin Mohammadkhani

# Keller Graduate School of Management

# Business Intelligence Systems

# April 15, 2012

# Dr. Miriam Masullo

Table of Contents

Introduction 3

Enterprise Database Design 3

ETL Tool Summary And Features 4

Data Mining and Reporting Tool Features 5

Conclusion 6

References 7

Enterprise Data Warehouse Proposal for ABC University

# **Introduction**

In such large undertaking, getting of to a good start is key to success. Nothing delays the success of a well-developed data warehouse environment more than undue deliberation at the outset. In many occasions, or development of systems one may think that to have a successful end result, they must gather all requirements before design and development starts. However, in the data warehouse environment, all requirements cannot be discerned until the system is build and the data in the warehouse in available for analysis. Thus, only some of the requirements can be known before the data warehouse is built. For this reason, we will build the data warehouse iteratively, in small, fast bursts of development where we can cross check our work and tweak as needed.

We know that the ABC University needs a data warehouse to be able to see a single view of all university information for students, faculty and staff. We also know that information is scattered among many databases between different departments, which makes data synchronization difficult. The information contained in these databases include both student’s and faculty where. Both students and faculties have distinct aspects and well as common aspects that connects them to each other.

The data warehouse that is going to be built will be a web application where aggregate tables will combine student and faculty information in them and will be used to retrieve final reporting information from. The reporting database will include have an ER diagram which will show the relationship between each table. It will be a normalized database except the aggregate tables.

With all the information discussed above, any data warehouse will need to have a solid database design as well as and ETL and data reporting tool that a user can use to generate report for themselves. These features will be discussed in this proposal.

# **Enterprise Database Design**

Enterprise Database design represents a coherent set of corporate policy across a subject area. After much research, the dimensional data model developed by Kimball is the best way to approach the database design.

As the publication attached in the treads summarizes “Kimball approach is much easier to implement, as we have to deal with small modules to begin with. Also Kimball approach provides more balance in terms of centralized and localized flexibility, as data warehouse is the collection of data marts which can be on different servers across the enterprise.” (Kumar, 2007)

As mentioned above, it is easy to understand the dimensional data model. With this type of model you can conceive the model, as a database cube was we could access a “slice” or section at any given time. It allows us the “visualize” the data.

Because of the how the dimensional model is presented, there is not need for a data mart. The dimensional data model’s cube can be seen as an OLAP where are queries are pre determined. Each cube in the dimension can be used as the Data Mart.

Each record needs to be identified easily. Thus I would like to give each cube a unique “packet id” where it describes the dimensions. The unique Id therefore will already have a description of what is contained in the cube and allows for easier queries. The packets can then be joined for multiple reporting. The id will be a combination of all three/four dimensions.

In regards to the physical database, I would like to use Oracle 10g. The first choice would have been MySQL because of cost and ease of use. However, due to the nature of business, we need a largely scalable database. Even though Oracle may have a high learning curve for some and a bit more difficult for some, it is highly scalable and has some neat features. As far as the application perspective goes oracle can be compiled in xml, which is the bridge between any programming languages and is easy to understand. Moreover, Oracle is extremely popular in medium to large enterprise business applications and medium to large data warehouses. Oracle also excels in medium to large data warehouses. Furthermore, Oracle uses table space for system metadata, user data and indexes and Reo and archive log files are used for point in time recovery. Oracle’s tables also have tons of features. Even though partitioning is costly, it comes with lots of options. Same in terms of replication where there is high complexity but allows for lots of data filtering and manipulation. Oracle’s Recovery Manager is also pretty good. As far as stored procedures, oracle offers good features with lots of built in packages that add great functionality.

As with anything program or suite that I use, if the same company makes the products I need, I will purchase it from the same company. Doing so allows for easier integration and support. Thus, I will use the Oracle SQL Developer Data Modeler mainly because I’m already using Oracle as my DB and the tools is easy to use and oracle has come up with some neat features such as version history, one click synch between the model and the data dictionary and their diagramming is pretty cool as well. Oracle has also improved their reporting where it allows exports to multiple formats and support for the business information model.

# **ETL Tool Summary And Features**

In the Database Design, we chose Oracle as the DB solution. Thus, I believe that if one company offers multiple tools for work, it is best to stick to them, as integration is a lot easier as is the support. Thus, I would like to use Oracle’s 11g Oracle Warehouse Builder as the tool for the use of the ETL processes. This ETL tools has been going through many generations and revisions thus giving the Oracle team many opportunities for feedback from consumers and improvements on the product.

Oracle Warehouse Builder (OWB) like many other ETL tools is frameworks for creation of extraction, transformation, and loading scripts for populating data stores, data marts or a date warehouse. In addition, OWB also includes a component to design the target warehouse or operational data store. Furthermore, the design component can be used to create data warehouses design as star schema or warehouses as third normal form.

Oracle Warehouse Builder facilitates ETL by generating code based on a metadata repository, which means that the design is translated internally by OWB, which in result in code generation. The generated code is used to create a data warehouse.

It is fairly easy to create a warehouse schema with OWB as the tool gives the user a wizard-driven design environment to create the dimensional warehouse schema. Wizards guide the user through the creation of dimensions with multiple levels and hierarchies, the creation of keys on these levels, and the linking of these dimensions to the fact tables. OWB provides wizards for all database objects supported in OWB. Furthermore, creating the ETL process is handled by defining and deploying mappings. OWB gives the user a highly graphical and easy to use interface, allowing users to model the ETL process by simply dragging and dropping objects on a mapping canvas. Once the objects are connected, OWB will generate a pl/sql package from this logical drawing, which will provide the actual data movement (Oracle, 2008)

One great aspect of OWB is that it seamlessly propagates changes in the meta data to the actual warehouse. This means that the user can make incremental design changes and load it to the data warehouse. Furthermore, the ETL generated code can be administered by the user and the generated code can be altered if necessary.

As with any Oracle product, attaining this tool has different levels of licensing the basic, and enterprise ETL depending on the users needs. This can cost the consumer anywhere from five to fifteen thousand dollars depending on number of licensees cans plugins needed.

# **Data Mining and Reporting Tool Features**

While using the same vendor such as Oracle is the best choice to have for all your tools, using multiple vendors that allow easy integration with each other is also a great idea. This gives the client the freedom to use the best tools from different vendors. With that said, we will use Cognos as my reporting system as it is very intuitive software with a huge development history. It also incorporates well with Oracle. I also believe it can be used by users with different technical experiences.

Cognos is IBM's business intelligence (BI) and performance management software suite. The software is designed to enable business users without technical knowledge to extract corporate data, analyze it and assemble reports. Moreover, Cognos is composed of nearly three-dozen software products. Since, Cognos is built on open standards, the software products can be used with relational and multidimensional data sources from multiple vendors, including Oracle. (SearchCIO, 2001) In addition, we all know that smartphones are becoming more like small personal computers these days and more often business is performed on them. Therefore, Cognos 10 comes with mobile capability, allowing users to access a complete version of Cognos from mobile devices such as tablets and smartphones.

Cognos Business Intelligence helps everyone in the organization explore data freely and analyze key facts, quickly collaborate and become aligned with the organization. Cognos reporting provides business users with the capability to modify queries and personalize their reports without relying on IT, receive and interact with reports on their mobile devices with Cognos Mobile and when offline with IBM Cognos Active Report. IBM analysis capabilities enable your workforce to analyze information wherever and whenever they need with fully interactive, personalized workspaces for the web, mobile and the desktop - and even offline. With all the reporting and analysis, users are able to monitor and analyze metrics through the organization with scorecards. These scorecards can be meaningful and measurable where actions can be taken appropriately. Furthermore, with its real time reporting capabilities, Cognos is able to fire alerts to notify you about exceptional business events and task lists indicate actions that are being taken for each exceptional event. (IBM)

The best aspect of using Cognos is that it integrates well with Oracle. Because both Oracle and IBM have done extensive work on their tools, we are able to combine data from our Oracle system and take advantage our Oracle Fusion Middleware and non-Oracle infrastructure. This provides flexibility as the organization changes.

Overall, Cognos is a great tool for user of all levels of experience to use. It provides intuitive and robust features to create scorecards and dashboards. Its flexibility also allows the tools to incorporate well with other systems such as Oracle.

# **Conclusion**

In conclusion, to fulfill the ABC University data warehouse requirements we are going to use Kimball’s dimensional data model as the data warehouse design since Kimball approach provides more balance in terms of centralized and localized flexibility. Following Kimball’s theory we will use Oracle 11g as our data warehouse. We will also utilize the Oracle 11g Warehouse Builder as the tool fit use of the ETL Process. Finally, we will utilize IBM's Cognos tool as our data mining and reporting as it allows easy integration with the Oracle system and also offers a easy and intuitive interface for the user to generate reports.

# **References**

IBM. (n.d.). *Cognos Business Intelligence*. Retrieved 04 05, 2012, from IBM: http://www-01.ibm.com/software/analytics/cognos/business-intelligence/features.html

IBM. (n.d.). *Cognos solutions for Oracle*. Retrieved 04 05, 2012, from ibm.com: http://www-01.ibm.com/software/analytics/cognos/solutions/oracle/

Kumar, N. (2007). *Inmon versus Kimball.* Naveem Kumar.

Oracle. (2008, 05). *Best Practices for Data Quality Process with OWB*. Retrieved 03 30, 2012, from Oracle: http://www.oracle.com/technetwork/developer-tools/warehouse/bestpracticesfordq-withowb-1-132734.pdf

oracle. (n.d.). *SQL Developer Data Modeler*. Retrieved 03 25, 2012, from Oracle: http://www.oracle.com/technetwork/developer-tools/datamodeler/overview/dm31-newfeatures-1499905.html

SAP. (2006, 05 16). *SAP*. Retrieved 03 25, 2012, from SAP: http://help.sap.com/bp\_bw370/documentation/Multi\_Dimensional\_Modeling.pdf

SearchCIO. (2001). *Definition Cognos*. Retrieved 04 05, 2012, from searchcio.com: http://searchcio.techtarget.com/definition/Cognos

Trujillo, G. (2008, 05 28). *MySQL versus Oracle Features/Functionality*. Retrieved 03 25, 2012, from Oracle: https://blogs.oracle.com/GeorgeTrujillo/entry/mysql\_versus\_oracle\_features\_functionality