**Enterprise Database Design**

Week 4

shahin mohammadkhani

**Abstract**

This document will answer the questions asked in assignment area in regards of Enterprise Database Design

Because I like the dimensional data model I would have to go with Kimball’s theory. 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, I really like the dimensional data model, as it is easy to understand. 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.

# Works Cited

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

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

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