**Birla Institute of Technology & Science, Pilani**

**Work-Integrated Learning Programmes Division**

**Second Semester 2013-2014**

**Mid-Semester Test**

**(EC-3 Regular)**

Course No. : SS ZG515

Course Title : DATA WAREHOUSING

Nature of Exam : Open Book

No. of Pages = 1

# No. of Questions = 6

Weightage : 50 %

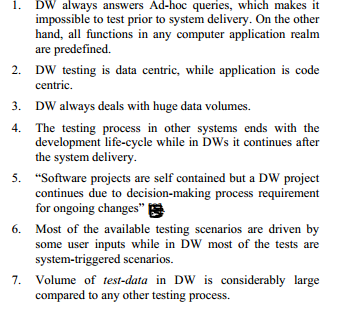
Duration : 2 Hours

Note:

1. Please follow all the *Instructions to Candidates* given on the cover page of the answer book.
2. All parts of a question should be answered consecutively. Each answer should start from a fresh page.
3. Assumptions made if any, should be stated clearly at the beginning of your answer.
4. Explain different challenges you will encounter while Testing Data Warehouse Applications? [7]

Answer:

As discussed in DW lectures, DW Testing is completely different from other systems such as Computer Applications or Transactional database systems. Consequently, the testing techniques used for these other systems are inadequate to be used in DW testing. Here are some of the differences. **[1 mark for each of the points noted below ]**

. 

1. Explain Partitioning and its significance in DW environment? (give an example). What are the different partitioning techniques supported by DBMS software. [7]

Answer: **[3 marks for Partitioning explanation and its significance. 2 marks each for Partitioning types.]**

Partitioning is a physical database design technique that many data modelers and DBAs are quite familiar with. Although partitioning can be used to accomplish a number of various objectives, the main goal is to reduce the amount of data read for particular SQL operations so that overall response time is reduced.

There are two major forms of partitioning:

1. **Horizontal Partitioning** – this form of partitioning segments table rows so that distinct groups of physical row-based datasets are formed that can be addressed individually (one partition) or collectively (one-to-all partitions). All columns defined to a table are found in each set of partitions so no actual table attributes are missing. An example of horizontal partitioning might be a table that contains ten years worth of historical invoice data being partitioned into ten distinct partitions, where each partition contains a single year's worth of data.
2. **Vertical Partitioning** – this partitioning scheme is traditionally used to reduce the width of a target table by splitting a table vertically so that only certain columns are included in a particular dataset, with each partition including all rows. An example of vertical partitioning might be a table that contains a number of very wide text or BLOB columns that aren't addressed often being broken into two tables that has the most referenced columns in one table and the seldom-referenced text or BLOB data in another.

Before database vendors began building partitioning (mainly horizontal) into their engines, DBAs and data modelers had to physically design separate table structures to hold the desired partitions, which either held redundant data (separate tables with data that were based off a live parent table) or were linked together to form one logical parent object (usually via a view). This practice has since been made obsolete for the most part for horizontal partitioning, although it is sometimes still done for vertical partitioning.

1. How do you classify the importance of Metadata in a DW environment? List and explain three main categories of Metadata. [7]

Answer: **[2 marks for Meta data importance, 1 mark for listing the three types of meta dat and 2 marks for each of metadata category.]**

An essential component of a data warehouse/business intelligence system is the metadata and tools to manage and retrieve metadata. One of our authors describe Metadata as DNA of DW.

Metadata defines the elements of the data warehouse and how they work together. Kimball et al. refers to three main categories of metadata: **Technical metadata, business metadata and process metadata.**

Technical Metadata:

Technical metadata defines the objects and processes in a DW/BI system, as seen from a technical point of view. The technical metadata includes the system metadata which defines the data structures such as: Tables, fields, data types, indexes and partitions in the relational engine, and databases, dimensions, measures, and data mining models. Technical metadata defines the data model and the way it is displayed for the users, with the reports, schedules, distribution lists and user security rights.

Business Metadata:

Business metadata is content from the data warehouse described in more user-friendly terms. The business metadata tells you what data you have, where it comes from, what it means and what its relationship is to other data in the data warehouse. Business metadata may also serves as documentation for the DW/BI system. Users who browse the data warehouse are primarily viewing the business metadata.

Process Metadata:

Process metadata is used to describe the results of various operations in the data warehouse. Within the ETL process all key data from tasks are logged on execution. This includes start time, end time, CPU seconds used, disk reads, disk writes and rows processed. When troubleshooting the ETL or query process, this sort of data becomes valuable. Process metadata is the fact measurement when building and using a DW/BI system. Some organizations make a living out of collecting and selling this sort of data to companies – in that case the process metadata becomes the business metadata for the fact and dimension tables. Process metadata is in interest of business people who can use the data to identify the users of their products, which products they are using and what level of service they are receiving.

1. Explain the strengths and weaknesses of ETL and ELT approaches. [8]

Answer: **[2 marks for each for ETL/ELT strengths and weaknesses ]**

**ETL: Strengths**

Development Time :Designing from the output backwards ensures that only data relevant to the solution is extracted and processed, potentially reducing development, extract, and processing overhead; and therefore time.

Targeted data :Due to the targeted nature of the load process, the warehouse contains only data relevant to the presentation.

Administration Overhead:Reduced warehouse content simplifies the security regime implemented and hence the administration overhead.

Tools Availability :The prolific number of tools available that implement ETL provides flexibility of approach and the opportunity to identify a most appropriate tool. The proliferation of tools has lead to a competitive functionality war, which often results in loss of maintainability.

**ETL Weaknesses**

Flexibility :Targeting only relevant data for output means that any future requirements, that may need data that was not included in the original design, will need to be added to the ETL routines. Due to nature of tight dependency between the routines developed, this often leads to a need for fundamental re-design and development. As a result this increases the time and costs involved.

Hardware :Most third party tools utilize their own engine to implement the ETL process. Regardless of the size of the solution this can necessitate the investment in additional hardware to implement the tool’s ETL engine.

Skills Investment :The use of third party tools to implement ETL processes compels the learning of new scripting languages and processes.

Learning Curve :Implementing a third party tool that uses foreign processes and languages results in the learning curve that is implicit in all technologies new to an organization and can often lead to following blind alleys in their use due to lack of experience.

**ELT Strengths**

Project Management :Being able to split the warehouse process into specific and isolated tasks, enables a project to be designed on a smaller task basis, therefore the project can be broken down into manageable chunks.

Flexible & Future Proof :In general, in an ELT implementation all data from the sources are loaded into the warehouse as part of the extract and load process. This, combined with the isolation of the transformation process, means that future requirements can easily be incorporated into the warehouse structure.

Risk minimization :Removing the close interdependencies between each stage of the warehouse build process enables the development process to be isolated, and the individual process design can thus also be isolated. This provides an excellent platform for change, maintenance and management.

Utilize Existing Hardware :In implementing ELT as a warehouse build process, the inherent tools provided with the database engine can be used. Alternatively, the vast majority of the third party ELT tools available employ the use of the database engine’s capability and hence the ELT process is run on the same hardware as the database engine underpinning the data warehouse, using the existing hardware deployed.

Utilize Existing Skill sets :By using the functionality provided by the database engine, the existing investments in database skills are re-used to develop the warehouse. No new skills need be learned and the full weight of the experience in developing the engine’s technology is utilized, further reducing the cost and risk in the development process.

**ELT Weaknesses :**

Against the Norm :ELT is an emergent approach to data warehouse design and development. Whilst it has proven itself many times over through its abundant use in implementations throughout the world, it does require a change in mentality and design approach against traditional methods. Though some principles remain true to all approaches, ELT is different in many ways. To get the best from an ELT approach requires an open mind.

Tools Availability :Being an emergent technology approach, ELT suffers from a limited availability of tools.

1. Why should we stage the data in an ETL architecture? Do you see any situations where there is no need of Staging data in ETL? if so, explain with an example. [7]

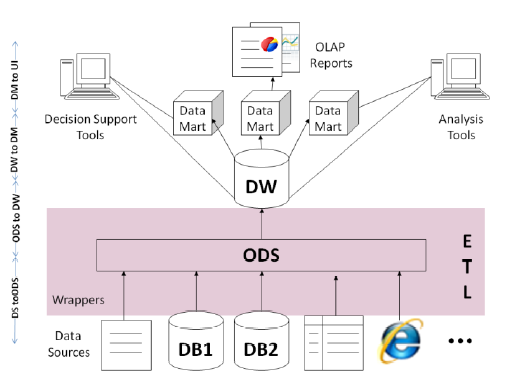
Answer:

In today's data warehousing environment, it's quite possible for ETL tools to establish a direct connection to the source database, extract and stream the data through the ETL tool to apply any required transformation in memory, and finally write it, only once, into the target data warehouse table. From a performance viewpoint, this is a great capability as writes, especially logged writes into the RDBMS, are very expensive; it's a good design goal to minimize them. However, despite the performance advantages, **this may not be the best approach**.[3 marks for this answer, if student comes up with an example where he proves Staging is not required then he has a strong supporting material otherwise he gets zero out of 3 marks.] There are several reasons an organization might decide to physically stage the data (i.e., write it to disk) during the ETL process:[4 marks for below points]

* The most appropriate CDC method requires a compare of the current copy of the source table to the prior copy of the same table.
* The organization has elected to stage the data immediately after extract for archival purposes -- possibly to meet compliance and audit requirements.
* A recovery/restart point is desired in the event the ETL job fails in midstream -- potentially due to a break in the connection between the source and ETL environment.
* Long running ETL processes may open a connection to the source system that create problems with database locks and that stresses the transaction system.

Q.6. Explain each of the components listed in the below DW system architecture. [14]

Answer:



**Answer: Student should explain below areas with a clear understanding:**

**ETL [1 marks]**

**Data Sources [1 mark]**

**DS to ODS load [1 mark]**

**ODS to DW load [1 mark]**

**DW to DM load [1 mark]**

**DM to UI load [1 mark]**

**OLAP reports/BI [2 marks]**

**Data Mart [2 marks]**

**DW [2 marks]**

**ODS [2 marks]**