Sessional Assignment

Data warehousing

Name: Shahab ud din ID: 13791 B.cs

1. **Implementation of Datawarehouse in Oracle 9i.?**

Ans: Oracle Warehouse Builder is comprised of a set of graphical user interfaces to assist you in implementing complex data system designs. Your designs are saved as metadata in a centralized repository.

The centralized repository, known as the Warehouse Builder repository, is hosted on an Oracle Database. The Design Center is the interface that provides a visual representation of the Warehouse Builder repository. Use the Design Center to import source objects such as tables and views, design ETL processes such as mappings, and ultimately design the target warehouse.

A mapping is an object in which you define the flow of data from sources to targets. Based on a mapping design, Warehouse Builder generates the code required to implement the ETL logic in your oracle database. Warehouse Builder can generate PL/SQL, SQL\*Loader, or ABAP code for mappings.

After you complete the design of a mapping, for example, and prompt Warehouse Builder to generate the code, the next step is to deploy the mapping. Deployment is the process of copying the relevant metadata and code you generated in the Design Center to a target schema. The target schema is generically defined as the database which will execute the ETL logic you designed in the Design Center. Specifically, in a traditional data warehousing implementation, the data warehouse is the target schema and the two terms are interchangeable.

The Design Center is the primary user interface. It is also a centralized interface in that you can launch from it all the client based tools, including the Control Center Manager. A secondary user interface is the web-based Repository Browser. In addition to browsing design metadata and auditing execution data, you can view and create reports. You can also perform limited lineage impact analysis in the Repository Browser. However, the Design Center has more advanced, interactive capabilities for lineage and impact analysis.

**General Steps for Creating an Oracle Data Warehouse and implementing it:**

**The following are the stages to build your dataware house builder in your oracle database and implementing it :**

**Stage 1:** preparing the warehouse builder Design center.

**Stage2:** importing the source metadata.

**Stage 3:** designing the oracle data warehouse.

**Stage 4**: deploying the design and implementing the data warehouse.

Before you can use any of the Warehouse Builder client components, first ensure access to a Warehouse Builder repository, which is an Oracle Database schema that stores the metadata for the system you design.

**Stage 1: Preparing the Warehouse Builder Design Center**

To integrate data and design a data warehouse, you primarily utilize the Project Explorer and the Connection Explorer.

In the Project Explorer, Warehouse Builder creates a single default project, MY\_PROJECT. As a project, it contains nodes for each type of design object that you can create or import.

The Connection Explorer is the window you use to establish connections between the Warehouse Builder repository to databases, data files, and applications.

 To prepare the Design Center, complete the following steps:

1. Adjust the client preference settings as desired or accept the default preference settings and proceed to the next step.
2. Connect to the source and target data objects.
3. Organize the design environment.

If you are satisfied with the default project, MY\_PROJECT, continue with the next step.

1. Identify the Oracle Database target.

Although you can use a flat file as a target, the most common and recommended scenario is to use an Oracle schema as the data warehouse target as described in these steps.

To define the Oracle target, begin by creating a *module.* Modules are grouping mechanisms in the Project Explorer that correspond to locations in the Connection Explorer. The Oracle target module is the first of several modules you create in Warehouse Builder.

1. Create a separate Oracle module for the data sources. (Optional)

At your discretion, you can either create another Oracle module to contain Oracle source data or simply proceed to the next step.

1. Identify the execution environment.

Under the Connection Explorer, notice the Control Centers node. A control center is an Oracle Database schema that manages the execution of the ETL jobs you design in the Design Center in subsequent steps.

During installation, Warehouse Builder creates the DEFAULT\_CONTROL\_CENTER schema on the same database as the repository.

1. Prepare development, test, and production environments. (Optional)

**Stage 2: Importing the Source Metadata**

1. Import metadata from the various data sources.

In the Project Explorer, select a node and determine the locations from which you intend to ultimately extract data. Now create a module for each relevant location. After you create a module, right-click the module and select Import to extract metadata from the associated location. Warehouse Builder launches a wizard to guide you through the process of importing data.

1. For the metadata you imported, profile its corresponding data. (Optional)

**Stage 3: Designing the Oracle Data Warehouse**

1. Create and design the data objects for the Oracle target module.

To create data objects, you can either launch the appropriate wizard or use the Data Object Editor. To use a wizard, right-click the node for the desired object and select New. After using a wizard, you may want to modify the object in the editor. In that case, right-click the object and select Open Editor.

1. As you design your data warehouse, be sure to frequently validate the design objects.

You can validate objects as you create them, or validate a group of objects together. In the Project Explorer, select one or more objects or modules, then click the Validate icon.

Examine the messages in the Validation Results window. Correct any errors and try validating again.

To redisplay the most recent validation results at a later time, choose Validation Messages from the View menu.

1. When satisfied with the design of the target objects, generate the code.

Generation produces a DDL or PL/SQL script to be used in subsequent steps to create the data objects in the target schema.

In the Data Editor, you can generate code for a single object by clicking the Generate icon.

In the Project Explorer, select one or more objects or modules, then click the Generate icon. Examine the messages in the Generation Results window. To redisplay the most recent generation results at a later time, choose Generated Scripts from the View menu.

You can save the generated script as a file and optionally deploy it outside Warehouse Builder.

1. Design mappings that define the flow of data from a source to target objects.
2. To manage dependencies between mappings, refer to ["Designing Process Flows"](https://docs.oracle.com/cd/B31080_01/doc/owb.102/b28223/concept_processflows.htm#CHDHGEAA).

**Stage 4: Deploying the Design and Implementing the Data Warehouse**

Recall that deployment is the process of copying the relevant metadata and code you generated in the Design Center to a target schema. This step is necessary to enable the target schema to execute ETL logic such as mappings.

To deploy and execute the data warehouse design, complete the following steps:

1. Deploy objects from either the Design Center or Control Center Manager.

In this step, you define the objects in the target schema. You need do this only once.

The simplest approach is to deploy directly from the Design Center by selecting an object and clicking the Deploy icon. In this case, Warehouse Builder deploys the objects with the default deployment settings.

1. Execute the ETL logic to populate the target warehouse.

In this step, you move data for the first time. Repeat this step each time you want to refresh the target with new data.

You have two options for executing the ETL logic in mappings and process flows. You can create and deploy a schedule as described in "Process for Defining and Using Schedules". Or you can execute jobs manually as described in ["Starting ETL Jobs"](https://docs.oracle.com/cd/B31080_01/doc/owb.102/b28223/concept_deploy.htm#BEIDEDIG).