**STANDARD ODI 12C FAQS FOR ALL LEVELS OF INTERVIEW PREPARATION**

1. WHAT IS THE DIFFERENCE BETWEEN **ETL** AND **ELT**?

**ETL:**

Traditional ETL tools will first *E*xtract the data from various sources and *T*ransform the data in a proprietary, middle-tier ETL engine and then *L*oads the transformed data into the target data warehouse.ETL engine performs data transformations/data quality checks on a ETL Engine in a row-by-row basis.

This leads to data movement over the network twice – once between the sources and the ETL server, and again between the ETL server and the target data warehouse. This will create performance issues when processing huge volumes of Data.

**ELT:**

In ELT approach, Extract the data from the source tables, Load the tables into the target database server, and then Transform the data on the target RDBMS using native SQL operators. This Architecture eliminates the need for a standalone (separate) ETL server and proprietary engine as it leverages the inherent power of RDBMS engines. It can utilize source and target database servers to perform complex transformations. Elimination of ETL engine reduces the hardware and software and maintenance costs.

1. EXPLAIN ODI 12C ARCHITECTURE?

A repository is a main component of ODI. The Repository consists of a Master Repository and typically several Work Repositories.

All objects that the ODI modules configure, develop, or use like (information about source, targets, metadata of all applications, projects, scenarios, and the execution logs ) are stored in one of these repositories

ODI Repository is composed of one *Master Repository* and several *Work Repositories.*

**Master Repository***:*  Usually master repository that stores the following information:

* Security information including users, profiles and rights for the ODI platform
* Topology information including technologies, server definitions, schemas, contexts, languages and so forth.
* Versioned and archived objects.

**Work Repository**: Project objects are stored in a **Work Repository**. We can create one work repository for one major project lifecycle to reflect a particular version or module. Example: Development, Test and Production environments.

Work Repositories are two types

i) Development

ii) Execution

Development repository stores developed objects.

* Models, including schema definition, datastores structures and metadata, fields and columns definitions, data quality constraints, cross references, data lineage and so forth.
* Projects, including business rules, packages, procedures, folders, Knowledge Modules, variables and so forth.
* Scenario execution, including scenarios, scheduling information and logs.

Execution Repository contains only the execution information. This type of repository can be created only for production purpose.

**ODI studio:**

ODI Studio provides four Navigators for managing the different activities and steps of an ODI project. They are **Designer**, **Operator**, **Topology** and **Security**

**Agents:**

Standalone Agent: It can be installed on the source or target systems and requires a Java Virtual Machine.

J2EE Agent: It is deployed on Oracle WebLogic Server and can benefit from the application server layer features such as clustering for High Availability requirements

Co-located Agent: It is deployed on Oracle Weblogic Server but will not use the features of application server

1. WHAT IS A FLOW BASED ARCHITECTURE? EXPLAIN?

ODI12c is built on a ELT Architecture and provides declarative flow based design approach to defining data transformation.

1. WHAT IS THE DIFFERENCE BETWEEN ODI 11.X AND 12C?
2. WHAT ARE THE NEW FEATURES OF ODI 12C?

* Declarative Flow-Based User Interface
* Reusable Mappings
* Multiple Target Support
* Step-by-Step Debugger
* Runtime Performance Enhancements
* Oracle Warehouse Builder Integration
* Unique Repository IDs
* Oracle GoldenGate source and target systems are now configured as data servers in Topology.

1. WHAT IS A REPOSITORY, EXPLAIN MASTER AND WORK REPOSITORY?

Please refer 2nd Question

1. HOW MANY TYPES OF WORK REPOSITORIES ARE THERE AND EXPLAIN?

Please refer 2nd Question

1. CAN WE HAVE MULTIPLE MASTER AND WORK REPOSITORIES?

No, only one master repository is always associated with multiple work repositories

1. WHAT ARE THE COMPONENTS OF ODI STUDIO?

ODI Studio provides four Navigators for managing the different activities and steps of an ODI project. They are **Designer**, **Operator**, **Topology** and **Security**

1. WHAT IS AN AGENT AND EXPLAIN?

**Agent** coordinates the execution of the ODI developed components .Typically, scenarios. It retrieves the code stored in the ODI repository, connects to the various source and target systems and orchestrates the overall data integration process.

**Standalone Agent** runs in a separate Java Virtual Machine (JVM) process. It connects to the work repository and to the source and target data servers via JDBC. Standalone agents can be installed on any server with a Java Virtual Machine installed. This type of agent is more appropriate when you need to use a resource that is local to one of your data servers (for example, the file system or a loader utility installed with the database instance), and you do not want to install a Java EE application server on this machine.

**Java EE Agent** is deployed as a web application in a Java EE application server (for example Oracle WebLogic Server or IBM WebSphere). The Java EE agent can benefit from all the features of the application server (for example, JDBC data sources or clustering for Oracle WebLogic Server). This type of agent is more appropriate when there is a need for centralizing the deployment and management of all applications in an enterprise application server, or when you have requirements for high availability.

**Co-located Agent** is deployed on Oracle Weblogic Server but will not use the features of application server

1. WHAT IS ODI CONSOLE?

ODI console is another web-based application which enables users to access information through a Web interface and an extension for Oracle Fusion Middleware Control Console

1. WHAT IS TOPOLOGY AND HOW DO YOU CONFIGURE?

Topology is one of the main component of ODI Studio which contains below components to design and configure Infrastructure(like source target database connections..

Physical Architecture

Contexts

Logical Architecture

Agents

Languages

Repositories

Physical Architecture: The physical architecture defines the different elements of the information system, as well as their characteristics taken into account by Oracle Data Integrator.

Technology: A *technology* handles formatted data. Therefore, each technology is associated with one

or more data types that allow Oracle Data Integrator to generate data handling scripts.

Data servers: The physical components that store and expose structured data are defined as *data servers*. A data server is always linked to a single technology. A data server stores information according to a specific technical logic which is declared into *physical schemas* attached to this data server.

Contexts bring together components of the physical architecture (the real Architecture) of the information system with components of the Oracle Data Integrator logical architecture (the Architecture on which the user works).

Logical architecture allows a user to identify as a single Logical Schema a group of similar physical schemas - that is containing datastores that are structurally identical - but located in different physical locations. Logical Schemas, like their physical counterpart, are attached to a technology.

Context allows to resolve logical schemas into physical schemas. In a given context, one logical schema resolves in a single physical schema.

Only Logical architectures are used in designer to refer to Actual Physical infrastructure like database name, database server name, schema name, user/pwd etc.

Always Logical schema will be mapped to a Physical schema using a Context. One Logical schema can be mapped to multiple physical schemas through different Context. by default a Context by name "GLOBAL" exists in ODI. We can create additional context in case if we need to map one logical schema to many physical schemas.

Note: multiple context can help to run a developed map to run on different environments like development, test and production by choosing different context.

1. WHAT DO YOU CONFIGURE IN SECURITY NAVIGATOR?
2. WHAT ACTIVITIES YOU DO IN DESIGNER NAVIGATOR?

This is the core module for developers where developers can develop mappings to Extraction Transform and Load data in a declarative approach using graphical components and knowledge modules (KM's).

1. WHAT ACTIVITIES YOU DO IN OPERATOR NAVIGATOR?

**Operator Navigator** contains below detailed sub components

**Session List** The Session List accordion displays all sessions organized per date, physical agent, status, keywords, and so forth.

**Hierarchical Sessions** The Hierarchical Sessions accordion displays the execution sessions organized in a hierarchy with their child sessions.

**Load Plan Executions** The Load Plan Executions displays the Load Plan Runs of the Load Plan instances

**Scheduling** The Scheduling accordion displays the list of physical agents and schedules.

**Load Plans and Scenarios** The Scenarios accordion displays the list of scenarios available

**Solutions** The Solutions accordion contains the Solutions that have been created when working with version management.

1. WHAT IS ROLE OF CONTEXT? EXPLAIN GLOBAL CONTEXT?

Context allows to resolve logical schemas into physical schemas. In a given context, one logical schema resolves in a single physical schema.

Always Logical schema will be mapped to a Physical schema using a Context. One Logical schema can be mapped to multiple physical schema through different Context. By default a Context by name "GLOBAL" exists in ODI.

1. WHAT IS DIFFERENCE BEWEEN DATA MODEL AND DATA STORE?

A Model is the description of a set of datastores. It corresponds to a group of tabular data structures stored in a data server.

A datastore represents a data structure. It can be a table, a flat file, a message queue or any other data structure accessible by Oracle Data Integrator.

1. WHAT ARE PROJECTS COMPONENTS?

**Folder**

Folders are components that help organizing the work into a project. Folders contain

packages, mappings, procedures, and subfolders.

**Packages**

A package is a workflow, made up of a sequence of steps organized into an execution

diagram. Packages assemble and reference other components from a project such as

mappings, procedure or variable

**Mappings**

A mapping is a reusable dataflow. It is a set of declarative rules that describes the

loading of one or several target datastores from one or more source datastores

**Procedure**

A Procedure is a reusable component that groups a sequence of operations that do not fit in the mapping concept.

Examples of procedures:

Wait and unzip a file

Send a batch of files via FTP

Receive emails

Purge a database

**Variable**

A variable's value is stored in Oracle Data Integrator. This value may change during the execution.

**Sequence**

A sequence is a variable automatically incremented when used. Between two uses the value is persistent.

**User Functions**

User functions allow you to define customized functions or "function aliases," for which you will define technology-dependent implementations. They are usable in mappings and procedures

**Scenario**

When a package, mapping, procedure, or variable component has been fully

developed, it is compiled in a scenario. A scenario is the execution unit for production.

Scenarios can be scheduled for automated execution.

**Global Components**

Global components are similar to the project objects. The main different is their scope.

They have a global scope and can be used in any project. Global objects include

Variables, Knowledge Modules, Sequences, Markers, Reusable Mappings, and User

Functions.

**Knowledge Modules:** Known as KM's which are ready made code templates which are responsible for generating SQL's to extract transform and load.

1. WHAT IS A PACKAGE?

A package is a workflow, made up of a sequence of steps organized into an execution

diagram. Packages assemble and reference other components from a project such as

mappings, procedure or variable

1. WHAT IS A PROCEDURE IN ODI?

A Procedure is a reusable component that groups a sequence of operations that do not fit in the mapping concept.

1. WHAT IS A VARIABLE AND HOW MANY TYPES ARE THERE?

A variable's value is stored in Oracle Data Integrator. This value may change during the execution. There are four types of a variable declare, refresh, set and evaluate the value of a variable.

■ **Declaring a variable**: When a variable is used in a package (or in certain elements of the topology that are used in the package), it is strongly recommended that you insert a Declare Variable step in the package. This step explicitly declares the variable in the package. Other variables that you explicitly use in your packages for setting, refreshing or evaluating their values do not need to be declared.\

■ **Refreshing a variable from its SQL SELECT statement**: A Refresh Variable step allows you to re-execute the command or query that computes the variable value.

■ **Assigning the value of a variable**: A Set Variable step of type Assign sets the

current value of a variable. In Oracle Data Integrator you can assign a value to a variable in the following ways:

**– Retrieving the variable value from a SQL SELECT statement**: When creating your variable, define a SQL statement to retrieve its value. For example, you can create a variable NB\_OF\_OPEN\_ORDERS and set its SQL statement to: select COUNT(\*) from <%=odiRef.getObjectName("L","ORDERS","D")%> where STATUS = 'OPEN'.

Then in your package, you will simply drag and drop your variable and select the "Refresh Variable" option in the Properties panel.

At runtime, the ODI agent will execute the SQL statement and assign the first returned value of the

result set to the variable.

**– Explicitly setting the value in a package**: You can also manually assign a value to your variable for the scope of your package. Simply drag and drop your variable into your package and select the "Set Variable" and "Assign" options in the Properties panel as well as the value you want to set.

**– Incrementing the value**: Incrementing only applies to variables defined with a numeric data type. Drag and drop your numeric variable into the package and select the "Set Variable" and "Increment" options in the Properties panel as well as the desired increment. Note that the increment value can be positive or negative.

**– Assigning the value at runtime**: When you start a scenario generated from a package containing variables, you can set the values of its variables. You can do that in the StartScenario command by specifying the VARIABLE=VALUE list.

**Incrementing a numeric value**: A Set Variable step of type Increment increases or decreases a numeric value by the specified amount

■ **Evaluating the value for conditional branching**: An Evaluate Variable step acts like an IF-ELSE step. It tests the current value of a variable and branches in a package depending on the result of the comparison. For example, you can choose to execute mappings A and B of your package only if variable EXEC\_A\_AND\_B is set to "YES", otherwise you would execute mappings B and C. To do this, you would simply drag and drop the variable in your package diagram, and select the "Evaluate Variable" type in the properties panel. Evaluating variables in a package allows great flexibility in designing reusable, complex workflows.

1. WHAT IS A SEQUENCE AND EXPLAIN IT TYPES?

A **Sequence** is a variable that increments itself automatically each time it is used. Between two uses, the value can be stored in the repository or managed within an external RDBMS table. Sequences can be strings, lists, tuples or dictionaries.

Oracle Data Integrator supports three types of sequences:

**Standard sequences**, whose current values are stored in the Repository.

**Specific sequences**, whose current values are stored in an RDBMS table cell.

Oracle Data Integrator reads the value, locks the row (for concurrent updates) and updates the row after the last increment.

**Native sequences, It** maps a RDBMS-managed sequence.

1. WHAT IS THE USE OF STANDARD AND SPECIFIC SEQUENCE?

To make sure that a sequence is updated for each row inserted into a table, each row must be processed by the Agent.

Oracle Data Integrator standard and specific sequences were developed to compensate for their absence on some RDBMS. If native sequences exist, they should be used. This may prove to be faster because it reduces the dialog between the agent and the database.

1. WHAT IS A SCENARIO?

When a package, mapping, procedure, or variable component has been fully

developed, it is compiled in a scenario. A scenario is the execution unit for production.

Scenarios can be scheduled for automated execution.

1. WHAT IS KNOWLEDGE MODULE? EXPLAIN?

A Knowledge Module is a code template for a given integration task. This code is independent of the Declarative Rules that need to be processed. At design-time, a developer creates the Declarative Rules describing integration processes. These Declarative Rules are merged with the Knowledge Module to generate code ready for runtime

1. EXPLAIN LKM, JKM, IKM AND CKM?

LKM (Loading Knowledge Modules) are used to extract data from source systems (files, middleware, database, etc.).

JKM (Journalizing Knowledge Modules) are used to create a journal of data modifications (insert, update and delete) of the source databases to keep track of the changes. These KMs are used in data models and used for Changed Data Capture

IKM (Integration Knowledge Modules) are used to integrate (load) data to the target tables. These KMs are used in mappings

CKM (Check Knowledge Modules) are used to check that constraints on the sources and targets are not violated. These KMs are used in data models' static check and mappings' flow checks

1. WHAT ARE PROJECTOR AND SELECTOR COMPONENTS?

Mapping components can be divided into two categories which describe how they are used in a mapping

a) Projector components

b) Selector components.

Projector Components: Projectors are components that influence the attributes present in the data that flows through a mapping. Projector components define their own attributes: attributes from preceding components are mapped through expressions to the projector's attributes. A projector hides attributes originating from preceding components; all succeeding components can only use the attributes from the projector

SOURCE/TARGET

SETS

AGGREGATES

DISTINCT

REUSABLE MAPPINGS

DATA SETS

Selector Components: Selector components reuse attributes from preceding components. For example, a Filter component following a datastore component reuses all attributes from the datastore component. As a consequence, selector components don't display their own attributes in the diagram and as part of the properties; they are displayed as a round shape. (The Expression component is an exception to this rule.) When mapping attributes from a selector component to another component in the mapping, you can select and then drag an attribute from the source, across a chain of connected selector components, to a target datastore or next projector component. ODI will automatically create the necessary queries to bring that attribute across the intermediary selector components.

Join and Lookup selectors combine attributes from the preceding components. Below are the selector components available in palette

Filters

Joins

Lookups

Sorts

Splits

Expressions

1. WHAT IS A REUSABLE MAPPING? EXPLAIN WITH EXAMPLE?

Reusable mappings are modular, encapsulated flows of components which wecan save and re-use. You can place a reusable mapping inside another mapping oranother reusable mapping (that is, reusable mappings may be nested).

1. CAN WE PLACE SOURCE AND TARGET IN REUSABLE MAPPINGS?

Yes, but there will be of no use as it will be shown as blank

1. WHAT IS THE USE OF SUB SELECT IN REUSABLE MAPPINGS?

The analytical functions are supported by reusable mappings

1. CAN WE USE ANALYTICAL FUNCTIONS IN NORMAL MAPPING? IF NO, THEN HOW TO IMPLEMENT?

No, you can implement analytical functions using reusable mappings

1. WHAT IS LOOKUP AND EXPLAIN ITS TYPES?

A Lookup is a selector component that returns data from a lookup flow being given a value from a driving flow. The attributes of both flows are combined, similarly to a join component. A lookup can be implemented in generated code either through a Left Outer Join or a nested Select statement.

1. EXPLAIN “SQL EXPRESSION IN THE SELECT CLAUSE” IN LOOKUPS?
2. WHAT IS A SLIT COMPONENT? WHAT IS THE USE OF REMAINDER IN SPLIT COMPONENT?

A Split is a selector component that divides a flow into two or more flows based on specified conditions. Split conditions are not necessarily mutually exclusive: a source row is evaluated against all split conditions and may be valid for multiple output flows. If a flow is divided unconditionally into multiple flows, no split component is necessary: you can connect multiple downstream components to a single outgoing connector port of any preceding component, and the data output by that preceding component will be routed to all downstream components. A split component is used to conditionally route rows to multiple proceeding flows and targets.

Split component can combine multiple filter conditions to a single component.

It extracts data from single input and loads data to multiple tables (or) output links based on the filter conditions provided.

We can have n number of output links along with one remainder output.

1. CAN WE HAVE MULTIPLE INPUT AND OUTPUT CONNECTOR POINTS IN SPLIT COMPONENT?

No, It extracts data from single input and loads data to multiple tables (or) output links based on the filter conditions provided.

1. WHAT IS A JOIN? AND EXPLAIN HOW MANY TYPES OF JOINS ARE THERE?

A Join is a selector component that creates a join between multiple flows. The attributes from upstream components are combined as attributes of the Join component

Join Type by checking the various boxes (Cross, Natural, Left Outer, Right Outer, Full Outer (by checking both left and right boxes), or (by leaving all boxes empty) Inner Join

1. WHAT IS THE DIFFERENCE BETWEEN UNION AND UNION ALL?

UNION Combines two inputs and removes duplicates

UNION ALL Combines two inputs; duplicates are not removed

1. HOW DO YOU CONFIGURE DELIMITED FILE?
2. WHAT ARE THE TYPES OF INTEGRATION STRATEGIES? EXPLAIN?

Control Append: Only INSERT

Incremental Update: INSERT, UPDATE OR INSERT ELSE UPDATE

Slowly Changing Dimension: Integrate data into a table using Type 2 slowly

changing dimensions (SCD).

1. WHAT IS SLOWLY CHANGING DIMENSIONS AND EXPLAIN ITS TYPES?
2. WHAT IS DIFFERENCE BETWEEN STATIC AND FLOW CONTROL?

**Static Control c**hecksto verify the integrity of the data contained in a data model. This operation is performed to assess the quality of the data in a model when constraints do not physically exist in the data server but are defined in Data Integrator only.

**Flow Control c**hecksto verify the integrity of a data flow before it is integrated into a given datastore. The data flow is checked against the constraints defined in Oracle Data Integrator for the datastore that is the target of the data flow.

1. WHAT IS DIFFERENCE BETWEEN INCREMENTAL UPDATE AND INCREMENTAL EXTRACTION?
2. WHAT IS CDC? HOW DOES IT WORK IN ODI?

Change data capture is a process of capturing changed data from the database. Usually this is required for extracting daily incremental data from source to load into datawarehouse.

CDC can be implemented in many ways

1. Using the date columns: In source systems (oltp) every table will have data columns to represent when a record has been changes and when a record has been inserted. Common column names are like created\_dt,updated\_dt,last\_updated\_dt ..etc

When we have these columns in oltp tables then we can write sql such a way that every day this SQL will extract records which are changed are created on that particular day. We can use variables for this implementation.

1. Example : we have a source table in oltp like

Cust\_id,name,address,created\_dt,last\_updated\_dt.

1. Then create a variable in ODI say example “ var\_Current\_date”
2. Then write a SQL or use a filter condition like

SELECT \* FROM CUSTOMERS WHERE LAST\_UPDATED\_DT>=#var\_Current\_date

1. Every day change the var\_Current\_date value to have currentdate-1 so that it will read the records which are changes yesterday. We can use packages to change these variable dates automatically before running the mappings.
2. Using manual ETL: In case if there are no data columns available in source tables then it is not possible to identify the changed records. So we can develop a logic using ODI mappings by creating a temporary lookup table and compare every data source data with lookup data. This a very tedious /complex tasks .
3. Using CDC Tools: In case if we want to automatically identify and capture only changes records from source, we can use CDC Tools like Oracle Golden Gate, Informatica Power Exchange, IBM CDC, Attunity, Sybase Replication Server …etc.
4. These Software’s have to be installed on source database to capture the changed data from sources continuously.
5. These Tools usually capture the data from source database logs so that there will not be any impact on source systems as customers will be using source systems continuously for their day to day activities like bank withdrawal , pre-paid mobile recharges, billing etc.

ODI is integrated with OGG (Oracle Golden Gate) to easy configure and extract

Apart from all above ODI has a separate concept to configure and capture incremental data from source database tables using Database Triggers.

This is called Trigger based CDC. This is not recommended to use as this will create performance issues on source systems as they are critical for business.

Using CDC in ODI (Trigger based).

To Supports CDC ODI has separate knowledge modules called Journalized knowledge modules.

Journal:

Subscriber:

It support two types of CDC

1. Simple
2. Consistent set
3. WHAT IS JKM AND EXPLAIN THE USE OF JOURNALIZING MODES?

JKM (Journalizing Knowledge Modules) are used to create a journal of data modifications (insert, update and delete) of the source databases to keep track of the changes. These KMs are used in data models and used for Changed Data Capture

Simple and Consistent set

1. WHAT IS VERSIONING? EXPLAIN?

In Versioning, the objects are archived and assigned a version number. By right clicking an object and selecting version, you can CREATE VERSION, COMPARE WITH VERSION AND RESTORE the selected object.

1. WHAT IS SMART EXPORT / IMPORT? EXPLAIN?

In normal Import ODI will export only selected object. In case a mapping requires table, logical and physical schema then these objects will not be exported.

Smart Export can export complete dependencies which are required to run mapping like source, target tables, models, context, physical and logical schemas and knowledge modules.

In addition to this we can also export Master and Work repositories, Topology…etc

Every time, when you import object in normal import we have to choose

DUPLICATE creates a copy in case same object is already present

INSERT only inserts new objects

UPDATE only updates existing objects

INSERT\_UPDATE (MERGE) new objects get inserted and existing ones get updated

1. WHAT IS A SCENARIO? HOW TO RE GENERATE A SCENARIO?

Scenarios are executable objects of mappings, procedures, packages. Everything which we develop in ODI designer will get stored in Work repository. This code design will store as it is in Repository. We can run this code from within designer, load plans . This code can be changed by user whenever they want some times it may lead to accidental unwanted changes. To help protecting changes and run as is code we can make a executable of mappings, procedure and packages.

It is similar to c or java code executables which are created when we compile program. Any changes made on programs will not reflect on these executables unless we recompile them.

We can generate scenarios for packages, procedures, mappings, or variables.

**Scenario variables** are variables used in the scenario that should be set when starting the scenario to parameterize its behavior.

Once scenarios are generated, they will get stored inside the work repository. The scenario can be exported then imported to another repository (remote or not) and used in different contexts.

Usually we can export the scenarios to Production execution repository and run the code.

Scenarios can be launched from a command line, from the Oracle Data Integrator Studio and can be scheduled using the built-in scheduler of the run-time agent or an external scheduler

When we create scenario it will generate following values:

**– Name**: The same name as the latest scenario generated for the component

**– Version**: The version number is automatically incremented (if the latest version is an integer) or set to the current date (if the latest version is not an integer)

We can create group of scenarios by selecting multiple objects and **Generate All Scenarios**

**We can perform**

■ **Replace**: Overwrites for each object the last scenario version with a new one with the same ID, name and version. Sessions, scenario reports and schedules are deleted. If no scenario exists for an object, a scenario with version number 001 is created.

■ **Re-generate**: Overwrites for each object the last scenario version with a new one with the same id, name and version. It preserves the schedule, sessions and scenario reports. If no scenario exists for an object, no scenario is created using this mode.

■ **Creation**: Creates for each object a new scenario with the same name as the last scenario version and with an automatically incremented version number. If no scenario exists for an object, a scenario named after the object with version number 001 is created

1. WHAT IS THE USE OF ODI OUT FILE IN A PACKAGE? It writes some content to a file and can also append the data to an existing content
2. CAN WE PLACE PACKAGE WITHIN A PACKAGE?

No, we can place package with in a package only by converting the package into scenario.

1. WHAT IS ODI FILE COPY, ODI FILE DELETE, ODI FILE APPEND, ODI FILE MOVE, ODI FILE WAIT?

ODI FILE COPY copies source file content to target file

1. NAME SOME EVENT DETECTION TOOLS IN A PACKAGE TOOLBOX?
2. EXPLAIN ODI FILE WAIT, ODI READ MAIL, ODI SLEEP, ODI WAIT FOR CHILD SESSION, ODI WAIT FOR DATA, ODI WAIT FOR TABLE, ODI WAIT FOR LOG DATA, ODI WAIT FOR LOAD PLANS?
3. NAME SOME INTERNET TOOLS IN A PACKAGE TOOLBOX?
4. EXPLAIN ODI FTP, ODI FTP GET, ODI FTP PUT, ODI READ MAIL, ODI SCP GET, ODI SCP PUT, ODI SEND MAIL, ODI SFTP, ODI SFTP GET, ODI SFTP PUT?
5. NAME SOME PACKAGE TOOLS WHICH ARE USED IN YOUR PROJECT? EXPLAIN WITH AN EXAMPLE?
6. CAN WE RUN SCENARIOS IN A PARALEL STEPS? Yes, by configuring in asynchronous mode
7. GIVE AN EXAMPLE USING THE EVALUATE VARIABLE?

An Evaluate Variable step acts like an IF-ELSE step. It tests the current value of a variable and branches in a package depending on the result of the comparison. For example, you can choose to execute mappings A and B of your package only if variable EXEC\_A\_AND\_B is set to "YES", otherwise you would execute mappings B and C. To do this, you would simply drag and drop the variable in your package diagram, and select the "Evaluate Variable" type in the properties panel. Evaluating variables in a package allows great flexibility in designing reusable, complex workflows.

1. HOW TO REFRESH A VARIABLE? EXPLAIN?

Refreshing a variable from its SQL SELECT statement: A Refresh Variable step allows you to re-execute the command or query that computes the variable value.

1. CAN WE INCREMENT A VARIABLE WITH CHARACTER DATA TYPE?

No, Incrementing only applies to variables defined with a numeric data type. Drag and drop your numeric variable into the package and select the "Set Variable" and "Increment" options in the Properties panel as well as the desired increment. Note that the increment value can be positive or negative.

1. HOW MANY TYPES OF ISOLATION LEVELS ARE THERE?

Committed, Uncommitted (Dirty Read), Repeated, Serializable

1. WHAT IS A LOAD PLAN? EXPLAIN?

A Load Plan is an executable object in Oracle Data Integrator that can contain a hierarchy of steps that can be executed conditionally, in parallel or in series. The leaf nodes of this hierarchy are Scenarios. Packages, mappings, variables, and procedures can be added to Load Plans for executions in the form of scenarios

A Load Plan is made up of a sequence of several types of steps. Each step can contain several child steps. By default, a Load Plan contains an empty root serial step. This root step is mandatory and the step type cannot be changed.

Serial Step: Defines a serial execution of its child steps. Child steps are ordered and a child step is executed only when the previous one is terminated. The root step is a Serial step

Parallel Step: Defines a parallel execution of its child steps. Child steps are started immediately in their order of Priority

Run Scenario Step Launches the execution of a scenario

Case Step When Step Else Steps: The combination of these steps allows conditional branching based on the value of a variable

Exception Step: Defines a group of steps that is executed when an exception is encountered in the associated step from the Step Hierarchy. The same exception step can be attached to several steps in the Steps Hierarchy

1. WHAT IS LOAD BALANCING?

The **Load Balancing** feature enables large volumes of data to be processed by automatically balancing the workload between several Agents.

1. WHAT IS SOLUTIONS?

Solution can group the objects which need to be exported, rather than selecting individual objects every time to export and import.

This will simplify the operations (code export/import) from one repository to another.

Once solutions are imported, we can restore them into new repository.

1. WHAT IS A DEBUGGER?

It can help to debug the mappings during execution.

First we will get the blue print and it will show the execution path with all the steps.

We can also add break points in between the steps to debug in detail. Also, we can watch and execute the SQL generated at a debug point step.

You can also get all the details from the simulation.

1. WHAT IS A DEPLOYMENT SPECIFICATION?

This feature can help to run different mapping with different KM and configuration.

It can also help to run one mapping more than once at a time.

Example: We can create one deployment flow for a mapping to run First time Load (Full Load) with different KM and Truncate option

For the same mapping, we can create another deployment flow and configure to run for daily runs without truncate and different KM.

1. WHAT IS INTERNAL IDENTIFIERS?

To ensure object uniqueness across several work repositories, ODI 11*g* used a

Mechanism to generate unique IDs for objects (such as technologies, data servers,

Models, Projects, Mappings, KMs, etc.). Every object in Oracle Data Integrator 11*g* is

identified by an internal ID.

1. EXPLAIN LOAD PLAN STRUCTURE?

A Load Plan is made up of a sequence of several types of steps. Each step can contain

several child steps. Depending on the step type, the steps can be executed

conditionally, in parallel or sequentially. By default, a Load Plan contains an empty

root serial step. This root step is mandatory and the step type cannot be changed

1. EXPLAIN LOAD PLAN LIFECYCLE?

When running or scheduling a Load Plan you provide the variable values, the contexts

and logical agents used for this Load Plan execution.

Executing a Load Plan creates a *Load Plan instance* and a first *Load Plan run*. This Load Plan instance is separated from the original Load Plan, and the Load Plan Run

corresponds to the first attempt to execute this instance. If a run is restarted a new *Load*

*Plan run* is created under this Load Plan instance. As a consequence, each execution

attempt of the Load Plan Instance is preserved as a different Load.