iNeuron

Python

Interview Questions

1. How do you handle table errors in PL/SQL programming?
Table errors can occur in PL/SQL programming when a table requested by the application program does not exist or has incorrect column types.

In such cases, an exception must be handled to inform users of an error in processing their request. This is possible because of the Exception managing the design of PL/SQL, which has predefined exceptions for various types of problems like 'NO_DATA_FOUND' and 'TOO_MANY_ROWS' for managing unusual data retrieval cases.

If required, specific user-defined exceptions can also be written inside the application programs to handle those specific table errors and any other type of Exceptions.

2. What types of exceptions can be encountered while working with PL/SQL programs?
Any programming language, including PL/SQL, must provide exception management. Exceptions are runtime errors that can occur during the execution of a program and cause unexpected results or terminate the processing.

In PL/SQL, there are both predefined and user-defined exceptions, which help handle various error conditions gracefully during runtime. The different types of Predefined Exceptions in PL/SQL include NO_DATA_FOUND, TOO_MANY_ROWS, DUPE_VAL _ON _INDEX, etc.

In contrast, User Defined Exceptions can be written as required inside application programs to handle specific scenarios effectively by raising one’s custom exception messages (using the raise statement).

Exception Handling in PL/SQL

3. Describe the compilation process for a PL/SQL program?
The compilation process for a PL/SQL program involves converting SQL and PL/SQL code into an executable format that the Oracle Database can interpret. This is done with the help of the SQL*Plus command-line interface, which first parses and translates all source code statements, checks syntax errors (if any), and resolves schema objects like tables & views referenced in query statements to their respective base address before finally creating a database procedure context from within which it executes those instructions.

4. List various schema objects associated with SQL tables in PL/SQL?
In PL/SQL, various schema objects are associated with SQL tables used to define database structure and the interaction between different queries in an application program. These include:

1. Tables - Tables are collections of connected data items made up of rows and columns that include details about a specific subject or entity;

2. Views - A view is a virtual table containing data from one or multiple tables;

3. Sequences - Sequences generate unique numerical values automatically whenever required;

4. Indexes – An index helps improve query performance by providing quick access to specific records within large datasets;

5. Synonyms - Synonyms act as aliases for schema objects like tables & views to be referenced using more structured naming conventions without repeating their underlying structures across applications programs repeatedly each time they need fetching from databases at runtime.

6. Stored Procedures - Stored procedures contain PL/SQL code blocks that execute predefined tasks on behalf of application programs, making them easier to maintain centrally rather than repeating similar sections in all related client programs independently.

7. Triggers – Triggers react dynamically based on defined events inside databases, like before update, after insert etc., allowing developers to write custom code for processing complex operations.

8. Packages – Packages contain predefined procedures and functions that provide abstracted access to underlying data structures (like tables & views).

It is easier to use the same Table/View structure in multiple applications programs without having to repeat its definitions repeated each time a query needs to run against those objects from within any given application or program.

Schema objects associated with SQL tables in PL/SQL

5. What is the syntax for creating a SQL table in PL/SQL?
The syntax for creating a SQL table in PL/SQL involves using the CREATE TABLE statement, which requires specifying an optional list of

column definitions and other necessary parameters. Generally, this is done by providing details like name, data type to be used (e.g., text, number, etc.), size if applicable, and any default constraints that need setting up before actual row data can be added to those tables.

For example:
CREATE TABLE myTableName(id INTEGER PRIMARY KEY NOT NULL, name VARCHAR2(50) DEFAULT 'John Doe');

In this example, "myTableName" is the new Table being defined internally inside the Oracle database having two columns named id & Names, both conforming to predefined types 'INTEGER' & 'VARCHAR2' respectively.

Syntax for creating a SQL table in PL/SQL

6. What is a Rollback statement in PL/SQL?
A PL/SQL command known as a rollback statement is used to undo database changes that have been made since the last commit or rollback was executed. It undoes all transactions that have been completed in the current session by reverting any data changes and returning to its prior state. This statement is commonly used to undo any unintentional changes or incorrect transactions that users may have committed.

Rollback statement in PL/SQL

7. How do you write a block of statements in PL/SQL?
A block of statements in PL/SQL is a set of SQL or PL/SQL commands grouped to execute as a single unit. A basic structure for writing

blocks of statements consists of declarations, an execution section, an exception-handling section and an optional end statement.

Declarations can include variable definitions, which must be on the first line followed by BEGIN keyword; this is followed by one or more lines with executable statements within it such as SELECT, INSERT etc.;

finally, any exceptions should be handled so the process completes successfully, without errors using EXCEPTION keyword before ending the block with END keyword.

8. What are the record-type statements available in PL/SQL?
Record-type statements in PL/SQL store data from database tables or other queries as a set of related elements. These statements provide the ability to process multiple values simultaneously instead of looping through each row individually.

The four record types in PL/SQL are CURSOR, RECORD, TABLE, and %ROWTYPE.

Cursor is a special type that enables users to iterate the results set multiple times;
Record allows access to individual columns within rows contained by query results;
Table can hold one or more records for further operations such as sorting and selection;
%ROWTYPE provides variable declarations which contain all fields from an associated table automatically populated with their corresponding values upon execution against it.

9. How to create and use database links in different schemas using PL/SQL programs?
Creating and using database links in different schemas using PL/SQL programs is a process that allows users to access or identify remote databases from their local ones. It enables distributed queries, transactions, and heterogeneous system integration by allowing data to be accessed across platforms or between networks for applications such as replication and migration.

To create a link, use the CREATE DATABASE LINK command followed by the name of the link being defined; then specify information regarding the target hostname (if applicable) along with its username and password credentials if required before concluding statement execution.

Create database links in different schemas using PL/SQL programs

10. How can one identify common errors while executing a set of commands using the syntaxes within a program unit defined by declarative sections, executable sections, exception handling sections, etc.?
Common errors while executing a set of commands written using the PL/SQL syntax in a program unit can be identified by understanding its structure and meaning.

To identify common errors,

users must review each line within these sections and verify their correctness with regard to data type definitions associated with existing parameters, including formal ones.;

this is also applicable when dealing with scalar datatypes for column values for tables, plus checking if correct table names have been specified, among other details and specific requirements stated beforehand, including trigger declarations, etc.
These steps will help determine where mistakes might exist, allowing issues to be addressed promptly, correcting them during the development phase itself, and preventing actual code production failures from occurring later on due to rigorous testing and earlier assessments undertaken before submission into the system environment.

11. Explain the concept of formal parameters used for passing data within subprograms written with reference to procedures and functions supported by PLSQL language?
Formal parameters are used for passing data within subprograms written using PL/SQL. They typically consist of two parts: an argument name and a default value (if applicable).

The argument name is used to define the scope of a parameter, and the default value allows users to provide any additional information required when calling this function or procedure from other code blocks.
This also ensures that if no values are specified during execution, then some predetermined fallback option exists, thereby preventing program crashes due to missing data elements not supplied correctly beforehand or throughout entire session processes.

12. What scalar data types are available for defining variables associated with identifiers according to Oracle's specific ANSI standards-compliant PL/SQL definition?

Scalar data types are available for defining variables associated with identifiers according to Oracle's specific ANSI standards-compliant PL/SQL definition. Scalar types, also known as elementary or primitive datatypes, represent a single value and can be either number (NUMBER), string (VARCHAR2) or date type (DATE).

13. How can table column be modified using ALTER command as part of DDL operations?
To modify a table column using ALTER command as part of DDL operations, the server-side hosting instance-related services running under RDBMS software must support a dynamic SQL execution mechanism.
Tables should reside inside schema-based repositories developed based on a relational paradigm consistent with following ACID principles adopted from transaction processing theory.
In order to execute such a statement, the user will have to login into a database and acquire the necessary privileges for performing the operation without any conflicts generated by other transactions in the same or different schemas involved in the current session.

14. How do you create an update statement in PL/SQL?
An update statement in PL/SQL is a command used to modify or change data from an existing table in the database. It has the syntax of:

UPDATE <table_name> SET column1 = value1, [column2 = value2...]
WHERE <condition>

where you can specify which columns need updating and what conditions should be met before making the changes. For example, UPDATE Employee's SET salary = 1000 WHERE id = 100; This statement would update all employees whose id equals 100 to have a salary of 1000.

15. What is the purpose of a database trigger in PL/SQL?
A database trigger in PL/SQL is a stored program that gets automatically executed when certain conditions are satisfied. They are employed to uphold data integrity, enforce business rules, or audit record-keeping modifications. They can be associated with INSERT, UPDATE, or DELETE operations on one or more tables and execute arbitrary SQL code every time they activate.

For example, A 'salary change' trigger could enforce a rule that salary must not go below 1000 for any employee, preventing updates from being committed if this condition is violated.

Database trigger in PL/SQL

16. How can numeric values be manipulated using PL/SQL?
Numeric values can be manipulated using PL/SQL in various ways. This includes performing basic arithmetic operations (such as addition, subtraction, multiplication, and division), rounding numbers to the nearest integer or decimal place, and calculating exponentials or logarithms with complex functions such as SQRT(), LOG() or POW().

17. Can you explain how to use an exception block within a PL/SQL package body?
An exception block within a PL/SQL package body can be used to handle any errors that occur during execution. It is defined by using the keyword EXCEPTION followed by one or more WHEN clause statements which check for specific error conditions and execute particular code if they are met.

For example:

EXCEPTION WHEN OTHERS THEN INSERT INTO log_table (error) VALUES (SQLCODE);

This block would catch all other exceptions not explicitly handled in an earlier statement, log them into a separate table and then continue with normal program flow. You can also define your own custom exceptions and create additional nested blocks when needed.

18. What are actual parameters, and how do they work with current transactions when programming with PL/SQL?
Actual parameters are values passed to a procedure or function during invocation. When programming with PL/SQL, they can be used within the current transaction by referencing them in SQL statements (e.g., SELECT * FROM TABLE WHERE id =:p_id). This enables developers to conditionally execute database operations depending on the values provided during execution.

For example: if some procedure was designed to retrieve data only from recent entries, then p_id being not null would trigger a query that selects all records created after the given date. Having it set as NULL would cause those lines of code to be skipped altogether.

19. Can video clips be included inside a PL/SQL program code?
No, it is impossible to include video clips inside a PL/SQL program code. This language is designed for manipulating structured data within Oracle databases. It, therefore, does not support media formats like videos or audio files that are better served through other technologies (e.g., HTML5).

20. Describe a Database Management System (DBMS) and explain how PL/SQL uses one.

Data that is held in a database is stored, managed, and handled by a piece of software called a database management system (DBMS). Users can execute various activities, such as creating, changing, or removing records within their databases, thanks to the interface it offers between them and the underlying database server.

The DBMS plays a crucial role in PL/SQL by offering effective access techniques for querying data from tables or views and guaranteeing that transactions are carried out correctly with the least amount of overhead. This includes employing locks to stop other sessions from changing shared resources.