SQL Interview questions

https://www.linkedin.com/in/suraj-netke

Q1. What is PL/SQL?

The Procedural Language/Structured Query Language is a programming language used to create, manage and manipulate Oracle database objects. It provides powerful procedural control of the data stored in an Oracle Database, including triggers, procedures, functions, and packages. In addition, it offers many features not available with regular SQL, such as explicit cursor handling for iterative processing, error trapping capability, and support for object-oriented programming extensions.

Q2. What are the different cursor attributes in PL/SQL?

In PL/SQL, a cursor is an interface for retrieving and processing data from the database. Cursor attributes are used to manage cursors in Oracle databases. Cursor attributes give information about the execution states of SQL statements or the conditions that affect the execution of these statements within PL/SQL blocks. Some common cursor attributes in PL/SQL include %FOUND, %NOTFOUND, %ROWCOUNT, and %ISOPEN.

The first two (found and not found) indicate whether a SELECT statement successfully retrieved records as part of an explicit or implicit cursor declaration;
rowcount indicates how many rows have been selected so far;
If ISOPEN returns TRUE if a corresponding query has not yet been executed
until its end or else FALSE if it already has ended with no more rows being available for selection from now on.

Q3. What are the advantages of the PL/SQL in points?

Advantages of PL/SQL:
It is faster than other conventional programming languages.
Simplified Application Development as it supports procedural, functional, and object-oriented Programming approaches.
Secure Access Management: It allows better control of data changes with the help of privileges, constraints & triggers.
Enhanced Productivity due to advanced features like Object Oriented capabilities debugging support etc.

Q4. How to create and use user-defined exceptions in PL/SQL?

User-defined exceptions in PL/SQL allow developers to create custom exception handling for their applications. They can be defined using the EXCEPTION statement and raised with the RAISE statement. Once created, these user-defined exceptions can be caught within a code block using an exception handler containing WHEN clauses (for expected errors) and other statements for unexpected errors or conditions. The code then handles the error gracefully or allows it to propagate further.

Q5. Explain the features of PL/SQL?

Features of PL/SQL:

It supports both SQL(Structured Query Language) & DML (Data Manipulation commands).
Easy integration with web technology like HTML & XML via embedded call interface or external procedure feature.
Error Handling mechanism helps in the debugging process, i.e., identify code errors quickly.
Allows execution of complex business logic by means of user-defined functions without giving access to the table structure itself.
Supports package concept that enables developers separate functionality into a logical group.
Supports Procedural Programming Paradigm – conditional statements, loops, iteration, etc.
PL/SQL Features

Q6. 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.

Q7. 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

Q8. 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

Q9. 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.

Q10. 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).

Q11. 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().

Q12.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 ES 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)
VALU (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.

Q13. 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.

Q14. In PL/SQL, how do you add a new table row using DML operations?

New table rows can be added using DML (Data Manipulation Language) methods in PL/SQL. The INSERT statement, which describes the columns and values of a specific row that you want to add to your database, is used to do this.

For instance: (column1, column2,...) INSERT INTO MY TABLE VALUES ('value1', 'value2',...); This will create a new table row in "my_table" containing the appropriate values for each provided column. You can insert several rows simultaneously using stored procedures or dynamic SQL.

Q15. How can you use Pseudo columns to manipulate data within PL/SQL?

Pseudo columns are virtual columns that present particular values in a SELECT statement but have no actual existence within the underlying table. In PL/SQL, pseudo-columns can manipulate data by providing additional information for operations such as sorting or replacing existing column names when selecting records from tables.

For example, you could use the ROWID pseudo-column to retrieve the unique row identifier of each record without having it explicitly defined in your SELECT

query like so: SELECT ... FROM my_table ORDER BY ROWID;

This would result in all queried rows being sorted according to their row identifiers instead of any other previously specified order criteria.

Q16. Explain what happens when the commit statement changes are made using DML operations within the context of a sequence of statements implemented using PL/SQL programming language?

The commit statement is used in PL/SQL to save changes made using DML operations such as INSERT, UPDATE, and DELETE. When this commit statement is included at the end of a sequence of statements, it tells the database server that all data manipulation language (DML) commands should be permanently recorded within its tables on disk.

This means that even if an error occurs during execution or the program stops before completion for any other reason, whatever modifications have been successfully committed will remain effective and won't require manual intervention from users afterwords.

The commit statement also serves to release any resources that were previously held by a cursor or other database objects while the sequence of statements was being executed

Q16. What is a SQL Cursor, and what purpose does it serve within the context of PL/SQL programming language?

Within the context of PL/SQL programming language, a SQL Cursor is an internal structure used to handle and process multiple rows of data. It acts as a pointer within your program which points at each row returned from queries in sequence so that you can take appropriate action depending on the result set specified by such statement (e.g., updating records).

As a result, developers may use SQL cursors to retrieve data regularly without constantly relying on database server resources, which greatly reduces overhead when working with huge amounts of data.

Q17. How can you use string literals within my queries while programming with PL/SQL?

String literals are used when writing queries in PL/SQL to represent text-based information such as names, titles and descriptions, etc., stored inside database tables or views previously by other applications using character strings instead of numerical integers and floats, etc.

You can write string literals directly into your queries simply by enclosing them within single quote marks ('): SELECT * FROM <tablename> WHERE name = 'John Doe';

Q18. How can we write a multi-line comment within a PL/SQL program?

Multi-line comments in PL/SQL are used to add remarks and notes over multiple lines of code. They begin with a double hyphen (--) followed by an asterisk (*), then the comment text, and finally, they end with an asterisk (*) followed by two hyphens (--).

Anything between these characters will be ignored when compiling your program or script, allowing developers to clarify their work for other readers without changing its functions.

For example, the following multi-line comment in PL/SQL explains what several lines of code do together: --* This block checks if there is new data added to the database table since the last execution *--

Q19. Can PL/SQL commands store or display graphic images?

Yes, PL/SQL commands can store and display graphic images. Graphics files such as JPEGs or PNGs can be stored in the database within BLOB (binary large objects) fields that efficiently store digital properties. Likewise, these binary data types can then be used to output graphical elements from the results of a query via display packages like Oracle's own HTP package.

This package allows developers to store pictures directly into their application databases and then render them on webpages without any extra work by utilizing functions like htp.image or even generating dynamic bar charts with graphs with code similar to htp.p ('');

Q20. How can network traffic be monitored with the help of PL/SQL commands?

Network traffic can be monitored with the help of PL/SQL commands using database packet sniffing. Packet sniffing is a method of intercepting and analyzing network packets sent over various networks, including local area and wide-area networks.

With this technique, developers can use PL/SQL packages such as DBMS_NETWORK_PACKET to look at all incoming and outgoing network requests from their program or script in order to detect malicious activity or analyze performance issues like slow response times.

Was this helpful?
follow for more

SURAJ NETKE

https://www.linkedin.com/in/suraj-netke