iNeuron

Python

Interview Questions

1. What is a deadlock?
A deadlock situation is created when two processes are competing for the access to a resource but unable to obtain because the other process is preventing it. So, the process cannot proceed until one of the processes to be terminated.

For example: Process A locks the Table A and process B locks the Table B. Now the process A requests the Table B and waiting for it and Process B is also waiting for Table A.

2. What is lock escalation?
Lock escalation is an optimization technique used to convert many fine-grained locks such as row or page locks into table locks for handling the large updates. SQL Server is using row-level locking by default, so it is easier to convert a large number of row locks into a single table lock for optimum results.

3. Does View contain Data?
No, because views are virtual structures.

4. What is CTE?
Common Table Expression (CTE) is an expression that contains temporary results defined in a SQL statement.

5. What are aggregate functions used for?
Aggregate functions are used to perform a calculation on one or more values, and returns a single value of more meaningful information.

Some aggregate functions are COUNT(), SUM(), MAX(), MIN(), AVG(), and ROUND().

6. What is a join?
A join is a way to combine rows from two or more tables, based on a related column between them.

7. What is the difference between an INNER JOIN and LEFT JOIN?
An INNER JOIN is used to include combined rows from two tables that match the ON condition. The final result does not include rows with no match for the ON condition.

A LEFT JOIN is used to keep all rows from the first table, regardless of whether there is a matching row in the second table for the ON condition.

8. What is the purpose of windows functions?
Windows functions are used when you want to maintain the values of your original table while displaying grouped or summative information alongside. It is similar to aggregate functions, but does not reduce the number of rows in the result by combining or grouping them into a few result.

9. What are indexes and why are they needed?
Indexes are a powerful tool used in the background of a database to speed up querying, by acting as a lookup table for data.

They are needed to efficiently store data for quicker retrieval, which can be paramount to the success of large tech companies which need to process on the scale of petabytes of data each day.

10. 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.

11. 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

12. 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.

13. Explain the structure of the PL/SQL?
PL/SQL is a procedural language extension of SQL. It has a block structure that allows you to write programs with multiple executable statements within it. A basic PL/SQL code block consists of three sections:

DECLARE: This section includes variables, constants, and cursors declarations for use in the program

BEGIN-END: This optional section includes any additional executable statements required by the application logic.

EXCEPTION: Include exception-handling routines which help manage errors during execution time.

PL/SQL supports the following statements:

- If-Then-Else

- While Loop
- For Loop
- GOTO
- Cursor Handling Statements
- DECLARE: Declare variables, constants, and cursors
- Fetch records from a result set driven by a cursor
- Exception handling
- Anonymous PL/SQL Blocks
- Triggers & Procedures
- Package variables & functions

Structure of PL/SQL

14. Explain the PL/SQL cursor and its types.
A PL/SQL cursor is a memory pointer that directs the user to a region in memory where SQL statements and data related to statement processing are stored. To retrieve and analyze many rows, this section uses a unique feature called a cursor.
The cursor selects numerous rows from the database, and these picked rows are independently handled within a program.

There are two types of Cursors:

Implicit Cursor: When invoking any of the implicit cursor commands SELECT INTO, INSERT, DELETE, or UPDATE, Oracle automatically constructs a cursor.
These cursors' execution cycles are managed internally by Oracle, which uses the cursor properties ROWCOUNT, ISOPEN, FOUND, and NOTFOUND to return information about the cursor's information and state.
Explicit Cursor: A SELECT statement that was expressly declared in the declaration block makes up this cursor. The cycle of these cursors' execution, from OPEN through FETCH and closure, must be managed by the programmer. Oracle also assigns a cursor to the SQL statement and defines the execution cycle during SQL statement execution.
Cursors in PL/SQL

15. 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.

16. 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.

17. How can we manipulate data stored within database tables using SQL blocks in PL/SQL?
SQL blocks are sections of code in PL/SQL that allow developers to create, delete, update, and manipulate data stored within database tables. Using SQL blocks in PL/SQL offers a powerful way for developers to interact with their databases programmatically by executing standard Structured Query Language (SQL) commands.

This includes common operations like creating new records, modifying existing ones, or deleting certain records from the table entirely, and more complex queries such as searching for particular values or joining multiple tables into one meaningful result set.

Developers can also execute stored procedures using SQL statement execution within an Oracle block and take advantage of various features like transaction control statements (COMMIT & ROLLBACK), parameter substitution variables, and special functions provided by the RDBMS.

18. What is the appropriate way to declare and define temporary tables in PL/SQL?
In PL/SQL, temporary tables can be declared and defined by first using the CREATE GLOBAL TEMPORARY statement, which creates a new object in the system catalog views that persists until explicitly dropped.

After this, information is inserted into this table from permanent or temporary (or global) sources like another query result set, SELECT statements, or INSERT values clauses. Developers may also use the ON COMMIT PRESERVE ROWS clause to ensure any data loaded into these temp-tables is not cleared after each commit operation happens on its parent session defining it.

Finally, once all needed operations have been completed with these objects, they must drop them properly with DROP TABLE syntax for cleanliness and resource efficiency.

19. What precautions should be taken to deal with runtime errors encountered when running a code snippet in PL/SQL?
When running a code snippet written in PL/SQL, runtime errors may arise and cause the program to crash or behave unexpectedly. To prevent this, developers must take some precautions when coding their logic.

First of all, they should use proper error handling techniques like using try-catch blocks and exception handlers where appropriate; also, consider explicit transactions management statements such as COMMIT & ROLLBACK for data integrity if modifying records directly within database tables;

frequently test both during development by stepping through instructions one line at a time;
pay attention to any suspicious messages or warnings Oracle's engine provides while debugging, preferably including specific IDs clearly shown near each issue.

20. What is a database server, and how is it used in PL/SQL?
A computer system known as a database server processes and saves data from one or more databases. In PL/SQL, the database server is a fundamental component to process SQL commands sent via application programs to store and retrieve data.

The primary functions of the database server include managing connections between clients and databases, processing user requests for information retrieval by translating them into queries acceptable by relational databases, executing those queries on behalf of connected clients, returning results to requesting applications, ensuring successful transaction commits when requested, etc.

Database Server in PL/SQL.