1. What is PL/SQL?

PL/SQL is a procedural language designed specifically to embrace SQL statements within its syntax. PL/SQL program units are compiled by the Oracle Database server and stored inside the database. And at run-time, both PL/SQL and SQL run within the same server process, bringing optimal efficiency. PL/SQL automatically inherits the robustness, security, and portability of the Oracle Database.

1. What is the purpose of using PL/SQL?

In Oracle database management, PL/SQL is a procedural language extension to Structured Query Language (SQL). The purpose of PL/SQL is to combine database language and procedural programming language. The basic unit in PL/SQL is called a block, which is made up of three parts: a declarative part, an executable part, and an exception-building part.

Because PL/SQL allows you to mix SQL statements with procedural constructs, it is possible to use PL/SQL blocks and subprograms to group SQL statements before sending them to Oracle for execution. Without PL/SQL, Oracle must process SQL statements one at a time and, in a network environment, this can affect traffic flow and slow down response time. PL/SQL blocks can be compiled once and stored in executable form to improve response time.

1. What is PL/SQL table? Why it is used?

Objects of type TABLE are called *PL/SQL tables*, which are modeled as (but not the same as) database tables. For example, a PL/SQL table of employee names is modeled as a database table with two columns, which store a primary key and character data, respectively.

PL/SQL table is an ordered collection of elements of the same type. Each element has a unique index number that determines its position in the ordered collection.

PL/SQL tables help you move bulk data. They can store columns or rows of Oracle data, and they can be passed as parameters. So, PL/SQL tables make it easy to move collections of data into and out of database tables or between client-side applications and stored subprograms.

1. What are the datatypes available in PL/SQL?

There are two types of datatypes in PL/SQL:

1. **Scalar datatypes** Example are NUMBER, VARCHAR2, DATE, CHAR, LONG, BOOLEAN etc.
2. **Composite datatypes** Example are RECORD, TABLE etc.
3. What is the basic structure of PL/SQL?

PL/SQL uses BLOCK structure as its basic structure. Each PL/SQL program consists of SQL and PL/SQL statement which form a PL/SQL block.

PL/SQL block contains 3 sections.

1. The Declaration Section (optional)
2. The Execution Section (mandatory)
3. The Exception handling Section (optional)

6) What is the difference between FUNCTION, PROCEDURE AND PACKAGE in PL/SQL?

**Function**: The main purpose of a PL/SQL function is generally to compute and return a single value. A function has a return type in its specification and must return a value specified in that type.

**Procedure**: A procedure does not have a return type and should not return any value but it can have a return statement that simply stops its execution and returns to the caller. A procedure is used to return multiple values otherwise it is generally similar to a function.

**Package**: A package is schema object which groups logically related PL/SQL types , items and subprograms. You can also say that it is a group of functions, procedure, variables and record type statement. It provides modularity, due to this facility it aids application development. It is used to hide information from unauthorized users.

7) How to write a single statement that concatenates the words ?Hello? and ?World? and assign it in a variable named Greeting?

Greeting := 'Hello' || 'World';

8) Does PL/SQL support CREATE command?

No. PL/SQL doesn't support the data definition commands like CREATE.

9) How exception is different from error?

Whenever an Error occurs Exception arises. Error is a bug whereas exception is a warning or error condition.

10) What is the main reason behind using an index?

Faster access of data blocks in the table.

11) What are PL/SQL exceptions? Tell me any three.

1. Too\_many\_rows
2. No\_Data\_Found
3. Value\_error
4. Zero\_error etc.

12) What are some predefined exceptions in PL/SQL?

A list of predefined exceptions in PL/SQL:

* DUP\_VAL\_ON\_INDEX
* ZERO\_DIVIDE
* NO\_DATA\_FOUND
* TOO\_MANY\_ROWS
* CURSOR\_ALREADY\_OPEN
* INVALID\_NUMBER
* INVALID\_CURSOR
* PROGRAM\_ERROR
* TIMEOUT \_ON\_RESOURCE
* STORAGE\_ERROR
* LOGON\_DENIED
* VALUE\_ERROR
* etc.

13) What is a trigger in PL/SQL?

A trigger is a PL/SQL program which is stored in the database. It is executed immediately before or after the execution of INSERT, UPDATE, and DELETE commands.

14) What is the maximum number of triggers, you can apply on a single table?

12 triggers.

15) What is the difference between execution of triggers and stored procedures?

A trigger is automatically executed without any action required by the user, while, a stored procedure is explicitly invoked by the user.

16) How to disable a trigger name update\_salary?

ALTER TRIGGER update\_salary DISABLE;

17) Which command is used to delete a trigger?

DROP TRIGGER command.

18) What is stored Procedure?

A stored procedure is a sequence of statement or a named PL/SQL block which performs one or more specific functions. It is similar to a procedure in other programming languages. It is stored in the database and can be repeatedly executed. It is stored as schema object. It can be nested, invoked and parameterized.

19) What are the different schemas objects that can be created using PL/SQL?

* Stored procedures and functions
* Packages
* Triggers
* Cursors

20) What will you get by the cursor attribute SQL%ROWCOUNT?

The cursor attribute SQL%ROWCOUNT will return the number of rows that are processed by a SQL statement.

21) How to execute a stored procedure?

There are two way to execute a stored procedure.

From the SQL prompt, write EXECUTE or EXEC followed by procedure\_name.

1. **EXECUTE** or [**EXEC**] procedure\_name;

Simply use the procedure name

1. procedure\_name;

22) What are the cursor attributes used in PL/SQL?

**%ISOPEN**: it checks whether the cursor is open or not.

**%ROWCOUNT**: returns the number of rows affected by DML operations: INSERT,DELETE,UPDATE,SELECT.

**%FOUND**: it checks whether cursor has fetched any row. If yes - TRUE.

**%NOTFOUND**: it checks whether cursor has fetched any row. If no - TRUE.