1. INTRODUCTION

a. What is the difference b/w sql and plsql?

- Sql: sql is a query lang that allows you to issue a single query or execute a single inser/update/delete
- Pl/sql: it is oracles programmng lang which allows you to write a full program (loops, variables, etc) to accomplish multiple selects/inserts/ updates/ deletes.

2. DATATYPES

a. What is the diff b/w %rowtype and type records

- %ROWTYPE: is used when you need to work with complete record
- TYPERECORD: is used to create your own datatype with specified number of values to hold

  Suppose if a table has 20 columns and we need to work with 7 columns

3. BLOCKS

4. CONTROL STATEMENTS

a. What are the conditional statements?

5. LOOP

6. CURSORS

a. If a cursor is open, how can we find a pl/sql program?

- Use the %isopen cursor status variable

b. Tell me about cursors?

- A cursor is a temporary work area created in the system memory when a sql statement is executed. A cursor contains information on a select statement and the rows of data accessed by it
- This temporary work area is used to store the data retrieved from database and manipulate this data. A cursor can hold more than one row, but can process only one row at a time. The set of rows that cursor holds is called active set

c. What is diff b/w cursor and ref cursor?

- Ref cursor: ref cursor is basically a datatype, it is normally declared as type r_cursor is REF CURSOR. It supports dynamic change of query
- Cursor: it is a static cursor in which query is assigned at designing time and cannot changed at run time

d. What is diff b/w weak and strong ref cursor?

e. What is diff b/w select and cursor?

- A select statement make used of cursor implicitly to fetch data from data base
- The cursor follow a three stage process of getting data from the database. The process is

  Open cursor
  Fetch cursor

Close cursor

7. EXCEPTIONS

a. Tell me something about exceptions?

b. What are exception? Tell me some seeded exceptions?

- Exceptions are notifications that identify issues that need resolution. Oracle collaborative planning generates an exception whenever an actual process does not match the required process

c. What is difference b/w raise and Raise_application_error?

- Raise statement is used to explicitly raise an exception with in a pl/sql block. It immediately stops normal execution of a pl/sql block and transfers control to an exception handler. It can be used to raise user defined and predefined exception

d. What is SQLCODE & SQLERRM?

- SQLCODE shows the error code
- SQLERRM shows the error message

```
 Begin
    Dbms_output.put_line (‘error code:’||SQLCODE);
    Dbms_output.put_line (‘error message:’||SQLERRM);
End;
```

e. What is exception of prorogation?

- A PL/SQL program is an anonymous block, a procedure or a function. This program or highest level block can call other procedures or functions or nest an anonymous block with in block . each pl/sql block can have its own exception section, or it can be totally void of exception handler
- To determine the appropriate exception-handling behavior, pl/sql fallows rules
- Scope: the pl/sql blocks in which an exception can be raised and hadled
- Propagation- the way in which an exception is passed back through enclosing blocks until it is handled or is resolved to be an unhandled exception.

8. SUBPROGRAMS

a. What are advantages of procedure and function?

- A function will return a value, a value can be one of many things including ref cursor, pl/sql tables, etc. adding to that , it is possible to use a function in sql statements, whereas procedures cannot be used

- Procedures are used to execute business logic, where we can return multiple values from procedure using OUT or INOUT parameter

b. **What are formal and actual parameters?**

- Formal parameters: it is a term used to refer to a parameter defined in the procedure or function declaration section
  Ex: procedure raise_sal(emp_id integer, increase real) is

- Actual parameter: it is a term used to refer to a parameter provided by calling statement to a procedure or a function
  Ex: raise_sal (emp_num, amount)

9. PACKAGES

a. **Explain polymorphism in pl/sql**

- Polymorphism is a feature of object-oriented programming, is the ability to create variable, function, or an object that has more than one form
  Consider the below package

```
Create or replace package addition
Is
Function adding (n integer, m integer) return integer;
Function adding (n date, m integer) return date;
End addition
```

In oracle procedureal programming also supports polymorphism in the form of program unit overloading inside a package, member function type etc.

b. **What are the advantages of packages?**

- It offers several advantages modularity, easier application design, information hiding, added functionality and better performance
- Modularity: modularity packages let you encapsulate logically related types, items, and subprograms in named pl/sql module.
- Easy application design: each package is easy to understand, and the interfaces b/w packages are simple clear and well defined. This aids application developer easier application design

When designing an application, all you need initially is the interface information in the package specs. You can code and compile a spec without its body. Then, stored subprograms that reference the package can be compiles as well.

- Information hiding with packages, you can specify which types, items, and subprograms are public (visible and accessible) or private (hidden and inaccessible).

  For example if a package contains four subprograms, three might be public and one private. The package hides the implementation of the private subprogram so that only the package is affected if the implementation changes. This simplifies maintenance and enhancement. Also, by hiding implementation details from users, you protect the integrity of the package.

c. **What is the diff b/w procedure and package?**

- Procedure: it can be called standalone and process list of action (although function can also do the same) to perform. The only diff with respect to function is that it do not require any placeholder column to hold the return value.
- Package: if any package component called the whole package gets compiled and remain loaded in the memory until flushed out (restart or using pragma serially_reusable). Therefore whenever any function/procedure called the response time is fast
- Draw backs
    1. Procedure can’t be called inside select/where statement
    2. If package procedure/function needs to be used only once, unnecessary the whole package gets loaded in memory.

d. **What is forward declaration and where can we use this?**

10.TRIGGERS

a. **How may triggers can be applied on a table?**

- 12 types of triggers we can apply on a table

  Insert/update/delete--------3
  Before/after--------------------2
  Rowlevel/statement level—2

  Hence 3*2*2

b. **Diff b/w procedure and trigger?**

Procedure	Trigger

1.we execute a procedure whenever we want with the help of exec comman	1.it can only be executed whenever an even (insert, delete and update) is fired on table on which trigger is defined
2.we can call a procedure from inside another procedure	2.we can't call another trigger within a trigger.
3.it can be scheduled	3.it can't be scheduled
4. we can pass input parameter to procedures	4.we can't pass parameters to triggers
5.it can return a value	5.it can't return a value
6.we can use print commands inside procedure to debug purpose	6.we can't use print command in triggers
7.we can use transaction statements like (commit, rollback)	7.we can't use the transaction statements inside a trigger
8.we can call a stored procedure from front end (.asp files, aspx files etc)	8.we can't call a trigger from these files.

c. What are mutating errors? How to avoid?

- Don't break the rule, don't use the same table for insertion and selection at the same time, use temporary tables or views
- Don't use triggers, modern developers don't use triggers
- If you want to break it use autonomous transaction

11.COLLECTIONS

a. What are collection types? Explain briefly?

b. Diff b/w nested tables and varrays?

c. Diff b/w for and forall loop?

- For is an actual loop which will go through records one by one and do some processing
- Forall is not an actual loop, its just a notation for a bulk DML operation. It will not go through rows one by one.

  For example, you can do some row processing in FOR loop, but you won't be able to do it in FORALL

12.BULK

a. Bulk collect? Give an ex?

- Select statements that retrieve multiple rows with a single fetch, improving the speed of the data retrieval.

13.PRAGMA

a. **Diff b/w pragma exception_init and Raise_application_error?**

- Pragma exception_init- allows to associate an oracle error number with the name of a user defined error. Here you need to define user-defined exception before it is used in 'pragma exception_init'.
  There are two parameter: exception name and error code.
  Used in un named exception
- Raise_application_error- allows to create meaningful error msg. it works with un-name user defined exceptions. It associates the number of the error with the text of the error. Therefore, the user-defined exception does not have a name associated with it.
  There are three parameters: err_number, err_msg, keep_err

b. **Pragma autonomous_transaction?**

14.What is NOCOPY? Where we can use them?

15.Where current of clause?

16.What are who columns? Why we can use in the table?

- ➢ WHO columns are used to track the information updated or inserted by users against the tables
- ➢ FND_STANDARD package is used for this purpose.
- ➢ FND_STANDARD.SET_WHO procedure is used to update the WHO columns in a table when insert, update are performed
  - Created_by
  - Creation_date
  - Last_update_by
  - Last_update_date
  - Last_update_login

- ➢ Use FND_PROFILE.VALUE ('USER_ID') for retrieving the user_id which will be used by created_by column
- ➢ Creation_date and last_update_date will be normally SYSDATE.
- ➢ Last_update_by is same as created_by
- ➢ Use USERENV ('SESSIONID') for getting last_update_login id