

[JAVA TUTORIAL](#)[#INDEX POSTS](#)[#INTERVIEW QUESTIONS](#)[RESOURCES](#)[HIRE ME](#)[DOWNLOAD ANDROID APP](#)[CONTRIBUTE](#)**Subscribe to Download Java Design Patterns eBook****DOWNLOAD NOW**[HOME](#) » [INTERVIEW QUESTIONS](#) » [PL SQL INTERVIEW QUESTIONS AND ANSWERS](#)

PL SQL Interview Questions and Answers

APRIL 2, 2018 BY [PANKAJ](#) — [1 COMMENT](#)

If you have worked on Oracle database and going for an interview, you should go through PL [SQL interview questions and answers](#).

PLSQL stands for Procedural Language extension to Structured Query Language. As the definition suggests it's a Block Structured programming language that extends the capabilities of [SQL](#) to provide a more versatile query and update solutions regarding relational databases.

A few of the many key features that P/L SQL provides are procedures, functions, triggers, cursors etc. A portion of P/L SQL is framed from the SQL programming syntax with added procedural features. For prerequisites, it is highly recommended to have fundamental knowledge in SQL (Structured Query Language).

Table of Contents [\[hide\]](#)

[1 PL SQL Interview Questions](#)

- [1.1 What is PL SQL?](#)
- [1.2 What are the key differences between SQL and PL SQL?](#)
- [1.3 What is the basic structure of PL/SQL?](#)
- [1.4 What are triggers and its uses?](#)
- [1.5 How is a PL/SQL code compiled?](#)

- 1.6 What are few of the schema objects that are created using PL/SQL?
- 1.7 Define Commit, Rollback and Savepoint.
- 1.8 What are the different datatypes available in PL/SQL?
- 1.9 What are %TYPE and %ROWTYPE for?
- 1.10 What is an exception in PL/SQL? What are the two types of exceptions?
- 1.11 How are functions and procedures called in PL/SQL?
- 1.12 Cite the differences between execution of triggers and stored procedures?
- 1.13 What is the cause of mutating table error and how can we solve it?
- 1.14 What are the different types of Constraints?
- 1.15 Why does %ISOPEN return false for an implicit cursor?
- 1.16 What are the different parts of a package?
- 1.17 What are the differences between Implicit and Explicit Cursors?
- 1.18 Why is there a need for SQLCODE and SQLERRM variables?
- 1.19 What does the argument [OR REPLACE] do?
- 1.20 What is error ORA-12154: TNS:could not resolve the connect identifier specified?
- 1.21 Define Overloaded Procedure
- 1.22 What are Integrity Rules?
- 1.23 How can we make an IF Statement within a SELECT Statement?
- 1.24 What is Normalization?
- 1.25 What is error ORA-01000: maximum open cursors exceeded
- 1.26 What is error ORA-03113: end-of-file on communication channel?
- 1.27 What is an example of translating a date into Julian Format?
- 1.28 What is consistency?
- 1.29 What are the various restrictions imposed on View in terms of DML?
- 1.30 What is PL/SQL Records?
- 1.31 What is scope and visibility in PL/SQL?
- 1.32 Is it possible to read/write files to-and-from PL/SQL?
- 1.33 How to Declare Fixed Length String Value In PL SQL
- 1.34 How can we implement Rollback or Commit statement in a Trigger?
- 1.35 How to Fix Oracle Error ORA-00942: table or view does not exist
- 1.36 How can we debug in PL/SQL?
- 1.37 Is it possible to pass an object or table to a procedure as an argument?
- 1.38 How many times can we COMMIT in a loop?
- 1.39 How do we accept inputs from user during runtime?
- 1.40 How can we connect an Android App to an Oracle database and use the PL/SQL procedural code?

PL SQL Interview Questions



Listed below are various PL SQL interview questions that will help you in your upcoming interview. You should also go through [SQL Interview Questions](#).

1. What is PL SQL?

PL SQL is a Block Structured programming language created by Oracle in the 1990s in hoping to provide additional procedural programming solutions to SQL.

2. What are the key differences between SQL and PL SQL?

SQL is a Structured Query Language as opposed to P/L SQL which is a full procedural language. SQL code is processed one statement block at a time, while P/L SQL code is executed as a single program at one time. SQL can be within P/L SQL, but P/L SQL cannot be within SQL.

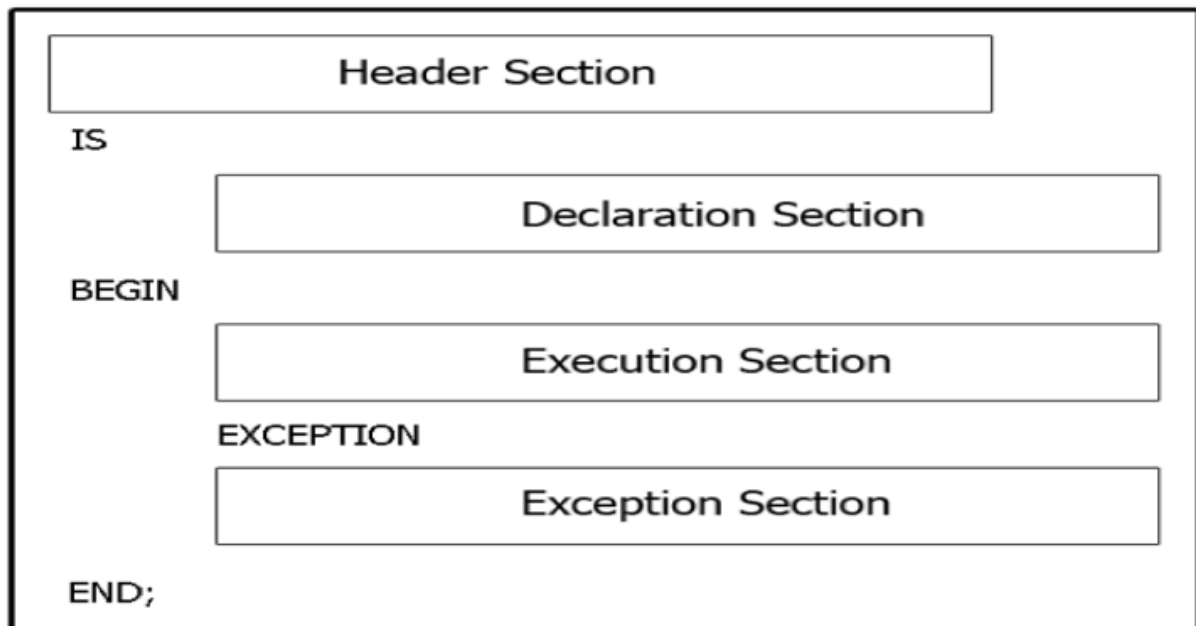
3. What is the basic structure of PL/SQL?

PL SQL, as much as any other procedural language, contains blocks. These blocks which are the basic unit of sensible code are primarily categorized by two types: anonymous blocks and named blocks.

```
[DECLARE]
    Declaration statements;
BEGIN
    Execution statements;
[EXCEPTION]
    Exception handling statements;
END;
```

They are called anonymous because they have no names and are not saved in an Oracle Database.

The figure below shows a detailed PL SQL block structure:



Header Section is for naming/labeling named blocks. This may or may not be used.

Declaration Section is for declaring variables, cursors, or sub-blocks that will be used in the Execution Section and Exception Section. This may or may not be used.

Execution Section block is where runtime code is placed. Statements in this section are required to exist for the structure to run.

Exception Section contains the exception and error handling of the code.

Essential keywords such as IS, BEGIN, EXCEPTION, and END are vital in the program for the runtime engine to distinguish each block sections.

4. What are triggers and its uses?

Triggers are blocks of code which are run whenever the criteria for a specific event is satisfied. They are hardcoded within the PL SQL program and listens to events such as: DML(database manipulation), DDL(database definition), and database operation. They can be coded within a view, table, database, or scheme for which the mentioned event belongs.

There are many uses of triggers. They can be used to generate column values upon activating. For event logging within the table activities, auditing, and creating table duplicates. For security, they can implement security authorization, and handle invalid transactions.

General Structure of creating a Trigger:

```
CREATE [OR REPLACE ] TRIGGER triggerName
{BEFORE | AFTER | INSTEAD OF }
{INSERT [OR] | UPDATE [OR] | DELETE}
[OF colName]
ON tableName
[REFERENCING OLD AS o NEW AS n]
[FOR EACH ROW]
WHEN (condition)
DECLARE
    Declaration-statements
BEGIN
    Executable-statements
EXCEPTION
    Exception-handling-statements
END;
```

5. How is a PL/SQL code compiled?

Firstly, PL/SQL code is transferred to the server and is compiled to byte code. This process takes place prior to program execution. To increase the performance of the procedural code, the procedures are converted to native code shared libraries which are conveniently linked to the kernel. Note that increase in performance still greatly depends on the code structure. This is independent to database calls, and affects performance only on loops, calculations, etc.

6. What are few of the schema objects that are created using PL/SQL?

A schema is a user-owned set of schema objects, or logical data structures. These schema objects types are as follows:

- Clusters
- Database links
- Database triggers
- Dimensions
- External procedure libraries
- Indexes and index types
- Java classes, Java resources, and Java sources
- Materialized views and materialized view logs
- Object tables, object types, and object views
- Operators
- Sequences

- Stored functions, procedures, and packages
- Synonyms
- Tables and index-organized tables
- Views

Among other objects which are not contained in a schema are:

- Contexts
- Directories
- Profiles
- Roles
- Tablespace
- Users
- Rollback segments

7. Define Commit, Rollback and Savepoint.

The COMMIT Statement finalizes to end your transaction and sets all changes to permanent. A transaction in SQL is any of statements that the Oracle Database treats as a single block. This also enables users to see the updates and changes made by the transaction. Finally, the COMMIT statement deletes all the savepoints prior to the transaction and releases transaction locks.

The ROLLBACK statement undoes the changes that the transaction has made. This is practically the opposite of the COMMIT Statement. Also, any locks made due to the transaction are released.

In conjunction, the SAVEPOINT statement is also used to set a restoration point when the ROLLBACK Statement is used. This limits the bounds of the ROLLBACK Statement by only reverting to the SAVEPOINT set point.

8. What are the different datatypes available in PL/SQL?

PL SQL data types can be broadly divided into following categories. There are many data types available in PL SQL but mostly you will be using some of the popular ones.

1. Numbers – INT, INTEGER, FLOAT, NUMBER, SMALLINT, REAL etc.
2. Character or String – CHAR, CHARACTER, RAW, VARCHAR, VARCHAR2, NCHAR, NVARCHAR2 etc.
3. Boolean – BOOLEAN
4. Date Time – DATE, TIMESTAMP etc.

Refer them in detail at [oracle Database Documentation](https://www.oracle.com/technetwork/database/enterprise-education/oracle-database-12c-education-2844761-01.pdf).

9. What are %TYPE and %ROWTYPE for?

The %ROWTYPE allows the coder to indirectly represent a full or partial row of a database table or view, whereas the %TYPE allows for the coder to indirectly represent the data type from a previously

declared variable or column. Basically, %ROWTYPE works on a full object whereas %TYPE works on a single column. The advantage to using either of these enables the coder to maintain data type declarations without ever having to know or change the data type for the items that use these. Below is an example of how the %TYPE allows for a layer of abstraction between names; allowing the coder to just change the first occurrence of the data type.

```
DECLARE
    name    VARCHAR(50);
    firstName name%TYPE;
    lastName name%TYPE;
    province name%TYPE;
    nationality name%TYPE;

    emp employees_table%ROWTYPE;
BEGIN
    Execution section;
END;
```

10. What is an exception in PL/SQL? What are the two types of exceptions?

Exceptions are manageable errors in a program. This means that errors handled by exceptions are within the bounds of the programmer to repair and PL/SQL provides catch features to encapsulate these errors to enable debugging and preventing the program to stop working.

There are two main types of exceptions – System Exceptions and User-Defined Exceptions. System Exceptions are such as no_data_found or too_many_rows and already defined by PL SQL. User-Defined Exceptions are exceptions defined by the user to handle particular errors.

11. How are functions and procedures called in PL/SQL?

For Procedures:

- CALL <procedure name> to call it directly
- EXECUTE <procedure name> from calling environment
- <Procedure name> from other procedures or functions or packages

Functions are called directly from other programs or procedures, no additional keyword is required.

12. Cite the differences between execution of triggers and stored procedures?

Triggers are activated automatically if the criteria for activation is met. This requires virtually no input or action from the user. On the other hand, a stored procedure requires to be explicitly called for it to be

activated.

13. **What is the cause of mutating table error and how can we solve it?**

This error occurs when an activated trigger causes a change or update in currently used table row.

This is fixed using views or temp tables instead of the actual table being used so that the database can provide use for the table row, while updating/changing the other.

14. **What are the different types of Constraints?**

Some of the popular constraints are:

- Check
- Not NULL
- Primary key
- Unique
- Foreign Key

15. **Why does %ISOPEN return false for an implicit cursor?**

Implicit cursors: SQL%ISOPEN always returns FALSE, indicating that the implicit cursor has been closed.

16. **What are the different parts of a package?**

(1) Package Specification – it holds the global package declaration

(2) Package Body – holds the functions, procedures (along with its local declarations), and cursor declarations.

17. **What are the differences between Implicit and Explicit Cursors?**

Explicit cursors are cursors created by the programmer to improve the manipulation of the context area. It is declared within the Declaration Section of the PL/SQL Block using the SELECT Statement. For example:

```
CURSOR cursorName [ parameter ] [ RETURN returnType ]  
IS SELECT STATEMENT;
```

On the other hand, Implicit Cursors are automatically generated by Oracle if there is no existing explicit cursor for the statement. This happens at the instant the SQL statement is ran. The developer has no control over the content and data within implicit cursors.

18. **Why is there a need for SQLCODE and SQLERRM variables?**

SQLCODE AND SQLERRM are globally-defined variables that handles the error description whenever a statement invokes an exception block. They are important due to their capability to trace exceptions that are highlighted by the OTHERS handler. The difference between the two is that SQLERRM returns the exact error message of the recently raised error, while SQLCODE returns the error code number of the last encountered error.

19. **What does the argument [OR REPLACE] do?**

The [OR REPLACE] argument grants the developer to reinvent an existing trigger. This means that with [OR REPLACE] it is possible to edit the specifics of a trigger without removing it.

20. **What is error ORA-12154: TNS:could not resolve the connect identifier specified?**

This error normally occurs when trying to establish a connection with your database. This is due to typographical error in database naming, rendering the Oracle unable to know what database you wish to connect to. This is a very popular error as this is one of the headscratchers faced by developers when trying to connect to a database. This maybe also an issue with the connection string within your tnsnames.ora file. Be sure to check that the tnsnames.ora is accessible and has the service name that you are currently using. But most commonly, syntax and typographical errors are the reason behind this error.

21. **Define Overloaded Procedure**

An overloaded procedure is a feature in PL/SQL that enables the developer to re-use an existing procedure with the exact name by varying the parameter structure such as the data type and parameter number. This is synonymous to the overloaded methods/functions in fundamental programming. This promotes uniformity and versatility of a PL/SQL block by giving a specific name procedure multiple ways to be called and multiple circumstances to operate.

An overloaded procedure is nothing more than a mechanism that allows the coder to reuse the same procedure name for different subprograms inside a PL/SQL block by varying the number of parameters or the parameter data type. Below is an example of where the same subprogram (callit) is reused but the data type for the input parameter is changed from INTEGER to VARCHAR2; Oracle is smart enough to know the input parameter type and call the proper subprogram.

```
SQL>
DECLARE
PROCEDURE ovlprocedure (num1 INTEGER) IS
BEGIN
DBMS_OUTPUT.PUT_LINE('Parameter is an INTEGER');
END ovlprocedure;

PROCEDURE ovlprocedure (character VARCHAR2) IS
```

```

BEGIN
DBMS_OUTPUT.PUT_LINE('Parameter is a VARCHAR2');
END ovlprocedure;

BEGIN
ovlprocedure (99);
ovlprocedure ('Hello');
* END;
SQL> /
Parameter is an INTEGER
Parameter is a VARCHAR2
PL/SQL procedure successfully completed.

```

22. What are Integrity Rules?

The Entity Integrity Rule(EIR) states that the Primary key must have value always or not Null.

The Foreign Key Integrity Rule(FKIR) states that if there exists a database relationship between a foreign key and a primary key, the Master Table cannot be deleted WHEN THERE IS DATA EXISTING IN THE CHILD TABLE.

The Business Integrity Rules encompasses the underlying processing not included in FKIR and EIR.

23. How can we make an IF Statement within a SELECT Statement?

We make use of the DECODE keyword. For example,

e.g. select DECODE (EMP_CAT,'3?','Third','4?','Fourth',Null);

24. What is Normalization?

Normalization is a neat feature where redundant tables and duplicate attributes are purposely removed to optimize the logical layout of the structure makes it easier to manage. This also improves data retrieval and nowadays, more and more servers implement normalized databases because of improvisations.

a) 1 Normal Form : This states that the database doesn't have duplicate attributes.

b) 2 Normal Form: This states that all the candidate keys are linked on the primary key. Problems related to 2 Normal Form occurs when a multiple columned primary key exists.

c) 3rd Normal Form : This states that if transitive dependency doesn't occur in a table, it is of 3rd Normal Form

25. What is error ORA-01000: maximum open cursors exceeded

There are two common scenarios in which this error might occur:

(1) When the value for OPEN_CURSORS is very low compared to the demand of cursors in your program. It is possible to increase the value of OPEN_CURSORS but discouraged, as it doesn't identify the problem in your program and only allocates more resources than it should, which is very inefficient and wasteful for a developer.

(2) A more recommended approach is considering your program and search for problems that are

concerning cursors. Take note of the implicit and explicit cursors as they also take up the available cursors available. Common issues of developers having this problem is they failed to close their cursor that is inside a loop, which in turn creates the error.

26. **What is error ORA-03113: end-of-file on communication channel?**

This is a common error which developers tend to overthink. A rather no-brainer cause for this is that the physical connection between the client's end and the database server might have been severed. Check your connection and make sure the cable properly connected to its intended port. Also make sure to check the status of the server if it is still functioning, some developers report an issue in their server that causes this error.

Most of the time, the problem lies on the physical layer and nothing more. But if you think the problem lies on the server itself, try accessing the alert log and take note of the catalog.

27. **What is an example of translating a date into Julian Format?**

We make use of the String Format: 'J'

For example, SQL > select to_char(to_date('5-Dec-2017','dd-mon-yyyy'),'J')

28. **What is consistency?**

Consistency simply means that each user sees the consistent view of the data.

Consider an example: there are two users A and B. A transfers money to B's account. Here the changes are updated in A's account (debit) but until it will be updated to B's account (credit), till then other users can't see the debit of A's account. After the debit of A and credit of B, one can see the updates. That's consistency.

29. **What are the various restrictions imposed on View in terms of DML?**

These are you are NOT ALLOWED to delete rows of Views containing: The Distinct keyword, Group Functions, The Pseudocolumn ROWNUM keyword, or a Group by Clause.

The other restriction is that you are NOT ALLOWED to change contents in a View that contains: The Distinct keyword, Group Functions, The Pseudocolumn ROWNUM keyword, a Group by Clause, or Expression-defined Columns (e.g. number_of_days * 7)

Few restrictions of DML operations on Views are:

You cannot DELETE a row if the View contains the following:

- 1) Group Functions
- 2) A Group By clause
- 3) The Distinct Keyword
- 4) The Pseudo column ROWNUM Keyword.

You cannot MODIFY data in a View if it contains the following:

- 1) Group Functions

- 2) A Group By clause
- 3) The Distinct Keyword
- 4) The Pseudo column ROWNUM Keyword.
- 5) Columns defined by expressions (Ex; Salary * 12)

30. What is PL/SQL Records?

PL/SQL Records is type of data structure that contain a set of data(can be of various types), or distinct information values that can be referenced with each other as fields. They are useful for classifying and retrieving data with common attributes or properties. With this, it is much easier to identify similar data by tracing the attributes.

PL/SQL can manage three types of records:

- Table based records
- Programmer based records
- Cursor based records

31. What is scope and visibility in PL/SQL?

The scope of a variable pertains to range within PL/SQL that it can be referenced. To expound, it is the block that contains the linked blocks along with the declared block.

The definition of scope and visibility for a variable is quite close with the only difference being if you have to qualify the variable. The scope of a variable refers to the region (breadth) of code where the variable can be referenced. The visibility refers to the region of code you can reference the variable without qualifying it. So, hopefully you can see, visibility is a subset of the scope and requires the variable to be qualified (told where it comes from) in order to use. An example is clearly the best option here to help explain. Consider the PL/SQL code:

```
SQL>
CREATE OR REPLACE PROCEDURE proceduretest IS
  var1 VARCHAR2(1); -- scope of proceduretest.var1 begins
PROCEDURE prodOne
IS
  var1 VARCHAR2(1); -- scope of prodOne.var1 begins
BEGIN -- visible prodOne.var1
  var1:= 'prodOne';
  DBMS_OUTPUT.PUT_LINE('In procedure prodOne, var1 = ' || var1);
  -- even though proceduretest.var1 is not visible it can still be
  qualified/referenced
  DBMS_OUTPUT.PUT_LINE('In procedure prodOne, proceduretest.var1 = ' ||
proceduretest.var1);
END; -- scope of prodOne.var1 ends
PROCEDURE prodTwo
IS
```

```

BEGIN -- visible proceduretest.var1
DBMS_OUTPUT.PUT_LINE('In procedure prodTwo, var1(proceduretest) = ' || var1);
DBMS_OUTPUT.PUT_LINE('In procedure prodOne, proceduretest.var1 = ' ||
proceduretest.var1);
END;
BEGIN -- visible proceduretest.var1

```

In the sample code above, we can observe the referenced variable, var1 is identical on the procedures. We can also see that var1 is not locally declared within procedure prodTwo. In this instance, the var1 used is the one declared from proceduretest which is visible on the perspective prodTwo. You can still be able to declare it locally like prodOne as well. This code tests the scope and visibility of variable var1 within different procedures. Oracle also provides debugging log to determine the scope of your variables.

32. Is it possible to read/write files to-and-from PL/SQL?

By making use of the UTL_FILE package, which was introduced in Oracle 7.3, we can make our PL/SQL code write and read files to and from our computer. However, you still need to receive an access grant by a DBA user to do such an activity. This promotes security and prevent intrusive coding.

For example:

```
CONNECT / AS SYSDBA
```

```
CREATE OR REPLACE DIRECTORY dir AS '/temp';
```

```
GRANT read, write ON DIRECTORY dir to test;
```

To grant the user to access to UTL_FILE:

```
GRANT EXECUTE ON UTL_FILE to test;
```

To write to a File:

```

DECLARE
handlerOne UTL_FILE.FILE_TYPE;
BEGIN
    handlerOne := UTL_FILE.FOPEN('DIR', 'tofile', 'w');
    UTL_FILE.PUTF(handlerOne, 'Write To File. \n');
    UTL_FILE.FCLOSE(handlerOne);
EXCEPTION
    WHEN utl_file.invalid_path THEN
        raise_application_error(-20000, 'Path is Invalid. Check
UTL_FILE_DIR');
END;

```

To read from a File:

```
DECLARE
handlerOne UTL_FILE.FILE_TYPE;
bufferOne varchar2(4000);
BEGIN
    handlerOne := UTL_FILE.FOPEN('DIR', 'tofile', 'r');
    UTL_FILE.GET_LINE(handlerOne, bufferOne);
    Dbms_output.put_line('The Contents from the File are: ' || bufferOne);
    UTL_FILE.FCLOSE(handlerOne);
EXCEPTION
    WHEN utl_file.invalid_path THEN
        raise_application_error(-20000, 'Path is Invalid. Check
UTL_FILE_DIR');
END;
```

33. How to Declare Fixed Length String Value In PL SQL

We can use CHAR to declare fixed length string value. Note that if the assigned string value is smaller then white spaces will be padded to end of it to create the value.

```
DECLARE
v1 VARCHAR2 (10) := 'PANKAJ';
v2 CHAR (10) := 'PANKAJ';
v3 CHAR (6) := 'PANKAJ';
BEGIN
    IF v1 = v2
    THEN
        DBMS_OUTPUT.put_line ('v1 = v2');
    ELSE
        DBMS_OUTPUT.put_line ('v1 != v2');
    END IF;

    IF v1 = v3
    THEN
        DBMS_OUTPUT.put_line ('v1 = v3');
    ELSE
        DBMS_OUTPUT.put_line ('v1 != v3');
    END IF;
END;
```

SQL Worksheet

```
1 DECLARE
2     v1 VARCHAR2 (10) := 'PANKAJ';
3     v2 CHAR (10) := 'PANKAJ';
4     v3 CHAR (6) := 'PANKAJ';
5 BEGIN
6     IF v1 = v2
7     THEN
8         DBMS_OUTPUT.put_line ('v1 = v2');
9     ELSE
10        DBMS_OUTPUT.put_line ('v1 != v2');
11    END IF;
12
13    IF v1 = v3
14    THEN
15        DBMS_OUTPUT.put_line ('v1 = v3');
16    ELSE
17        DBMS_OUTPUT.put_line ('v1 != v3');
18    END IF;
19
20 END;
```

```
v1 != v2
v1 = v3
```

34. How can we implement Rollback or Commit statement in a Trigger?

We cannot. It is not logical to put a Rollback or Commit within a Trigger because these statements impose a savepoint which affects the logical transaction processing.

35. How to Fix Oracle Error ORA-00942: table or view does not exist

There are two ways to check and fix this error.

- Check for typo error in your program, also check if the table and views are created or not and you are connecting to correct database.
- Check if your user has required permissions or not, this is also one of the reason where you can't see the table or view.

36. How can we debug in PL/SQL?

We can make use of the DBMS_OUTPUT for printing breakpoint activities. We can also use DBMS_DEBUG.

37. Is it possible to pass an object or table to a procedure as an argument?

Using database links, we can pass an object type of a certain database into another database. By this, we can pass these as arguments and be used by another procedure that contains the appropriate parameter argument type to receive this object. For example

```
-- Database One: receives the table as an argument and stored to TableTest
```

```
CREATE OR REPLACE PROCEDURE receiveTable(TableTest DBMS_SQL.VARCHAR2S) IS
BEGIN
    --
    null;
END;
/
```

```
-- Database Two: passes the table as an argument and is hoped to be received by
'receiveTable'
```

```
CREATE OR REPLACE PROCEDURE sendTable IS
    TableTest DBMS_SQL.VARCHAR2S@DBLINK2;
BEGIN
    receiveTable @DBLINK2(TableTest);
END;
/
```

38. How many times can we COMMIT in a loop?

It is advised to commit as least as possible especially within loops or iterative statements. This is to minimize resource over usage on instances that the loops were poorly constructed or no end keyword, resulting in a leak and infinite calls to the COMMIT block. This scenario results in a ORA-1555 error.

In declaring undo or rollback statements, having the least number COMMITS lessens the stress and code execution of these state, resulting in a faster and much more responsive rollbacks.

Also it is of good programming habit to allocate the least amount of memory resource as much as possible while keeping the same functionality of the code. This is done by minimizing the overuse of COMMIT within the code.

39. **How do we accept inputs from user during runtime?**

The ACCEPT keyword is used to receive inputs from the user. It also allows the received data to be buffered to a variable for storage.

For example:

```
> accept y number prompt 'Enter your desired number: '

>declare
    z number;
begin
    z := &y;
end;
```

40. **How can we connect an Android App to an Oracle database and use the PL/SQL procedural code?**

Before getting into the PL/SQL, we must create a REST API using JSON for establishing connection. You can import the Retrofit libraries along with dependencies to establish communication with your Android App. Then prepare the functions and procedures using PL/SQL, and then once your oracle database is connected you are good to go.

That's all for PL SQL interview questions and answers. It's a very vast programming language and it's not feasible to cover everything here, but I have tried to cover most of the important questions asked in the interview. If you think that I have missed any important interview questions, please let me know through comments and I will try to add them to the list.

« PREVIOUS

SQL Interview Questions and Answers

About Pankaj

If you have come this far, it means that you liked what you are reading. Why not reach little more and connect with me directly on [Google Plus](#), [Facebook](#) or [Twitter](#). I would love to hear your thoughts and opinions on my articles directly.

Recently I started creating video tutorials too, so do check out my videos on [Youtube](#).

FILED UNDER: [INTERVIEW QUESTIONS](#), [PL SQL](#)

Comments

Madhubanti says

JANUARY 16, 2018 AT 2:08 PM

Thanks Pankaj for this

[Reply](#)

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

☐

Save my name, email, and website in this browser for the next time I comment.

POST COMMENT

DOWNLOAD ANDROID APP



IMPORTANT INTERVIEW QUESTIONS

Java Interview Questions

- > [Core Java Interview Questions](#)
- > [String Interview Questions](#)
- > [Multithreading Interview Questions](#)
- > [Collections Interview Questions](#)
- > [Exception Interview Questions](#)
- > [Java Programming Interview Questions](#)
- > [Java 8 Interview Questions Part 1](#)
- > [Java 8 Interview Questions Part 2](#)
- > [Servlet Interview Questions](#)
- > [JSP Interview Questions](#)
- > [Struts 2 Interview Questions](#)
- > [JDBC Interview Questions](#)
- > [Spring Interview Questions](#)
- > [Hibernate Interview Questions](#)
- > [JSF Interview Questions](#)
- > [Web Services Interview Questions](#)
- > [Scala Basic Interview Questions](#)
- > [Scala Intermediate Interview Questions](#)
- > [Scala Advanced Interview Questions](#)
- > [Scala Interview Questions Summary](#)
- > [Common Job Interview Questions](#)

Miscellaneous

- > [Java ClassLoader](#)
- > [String StringBuffer StringBuilder](#)
- > [Java is Pass By Value](#)
- > [Java Heap vs Stack Memory](#)

RECOMMENDED TUTORIALS

Java Tutorials

- > [Java IO](#)
- > [Java Regular Expressions](#)
- > [Multithreading in Java](#)
- > [Java Logging](#)
- > [Java Annotations](#)
- > [Java XML](#)
- > [Collections in Java](#)
- > [Java Generics](#)
- > [Exception Handling in Java](#)
- > [Java Reflection](#)
- > [Java Design Patterns](#)
- > [JDBC Tutorial](#)

Java EE Tutorials

- > [Servlet JSP Tutorial](#)
- > [Struts2 Tutorial](#)

- > [Spring Tutorial](#)
- > [Hibernate Tutorial](#)
- > [Primefaces Tutorial](#)
- > [Apache Axis 2](#)
- > [JAX-RS](#)
- > [Memcached Tutorial](#)

Java Programming Courses

Beginners Java Script Tutorials

Java Books for Beginners

Java Game Development Tutorials

Basic Java Tutorials

Java Tutorials for Beginners

Free Interactive Java Tutorial

Download Java Development Kit

