PL/SQL Interview Questions & Answers

1. What is PL/SQL?

PL/SQL is Oracle's procedural extension to SQL. It allows you to use programming constructs like variables, loops, and conditions within SQL.

2. Difference between SQL and PL/SQL

SQL is declarative and used for data manipulation. PL/SQL is procedural and can execute multiple SQL statements inside a block with procedural logic.

3. Structure of a PL/SQL block

DECLARE -- variable declarations
BEGIN -- executable code
EXCEPTION -- error handling
END;

4. Types of PL/SQL blocks

Anonymous blocks: unnamed, executed immediately.
Named blocks: procedures, functions, packages, triggers.

5. What is %TYPE?

%TYPE declares a variable with the same datatype as a table column.

6. What is %ROWTYPE?

%ROWTYPE declares a record with the same structure as a table row.

7. What are variables in PL/SQL?

Variables are storage locations for data in a PL/SQL block. Declared in the DECLARE section.

8. What are constants in PL/SQL?

Constants hold values that cannot be changed after initialization. Declared with the CONSTANT keyword.

9. What are control structures in PL/SQL?

Control structures include IF-THEN, CASE, LOOP, WHILE, FOR to control flow.

10. NULL handling in PL/SQL

Use NVL, COALESCE, or explicit checks to handle NULL values.

11. Difference between a procedure and a function

PL/SQL Interview Questions & Answers

Procedure: may or may not return a value, cannot be used in SQL statements.
Function: must return a value, can be used in SQL statements.

12. Syntax for creating a function

CREATE FUNCTION name(params) RETURN datatype IS BEGIN ... RETURN value; END;

13. Syntax for creating a procedure

CREATE PROCEDURE name(params) IS BEGIN ... END;

14. What are IN, OUT, and IN OUT parameters?

IN: pass value in.
OUT: return value out.
IN OUT: pass and return value.

15. Can functions return multiple values?

Not directly. Use OUT parameters, records, or collections to return multiple values.

16. What is a recursive function?

A function that calls itself until a base condition is met.

17. How do you call a function in PL/SQL?

SELECT function_name(args) FROM dual; or inside another PL/SQL block.

18. What is a stored function's deterministic property?

A deterministic function returns the same result for the same input. Marked with DETERMINISTIC keyword.

19. Can you commit inside a function?

Generally not recommended; it can cause unexpected transaction behavior.

20. Difference between local and global variables in packages

Local: visible only in package body.
Global: declared in package spec, visible outside.

21. What is a cursor?

A pointer to the result set of a SQL query.

22. Difference between implicit and explicit cursors

Implicit: created automatically for single SQL statements.
Explicit: declared and controlled manually.

23. Cursor attributes

%FOUND, %NOTFOUND, %ROWCOUNT, %ISOPEN

24. FOR loop cursor

A cursor FOR loop automatically opens, fetches, and closes a cursor.

25. Parameterized cursor

A cursor that accepts parameters to filter its result set.

26. What is a trigger?

A stored program that executes automatically when an event occurs on a table.

27. Types of triggers

BEFORE/AFTER INSERT, UPDATE, DELETE.

28. Row-level vs statement-level triggers

Row-level: fires for each row.
Statement-level: fires once per statement.

29. Mutating table error

Occurs when a row-level trigger queries/modifies the table that fired it. Use compound triggers or statement-level triggers to avoid.

30. Disabling and enabling triggers

ALTER TRIGGER name DISABLE; ALTER TRIGGER name ENABLE;

31. What is a package?

A collection of related procedures, functions, variables, and cursors stored together.

32. Package components

Specification (interface) and Body (implementation).

33. Advantages of packages

Encapsulation, better performance, easier maintenance.

PL/SQL Interview Questions & Answers

34. Overloading in packages

Multiple procedures/functions with the same name but different parameter lists.

35. Global variables in packages

Declared in the package specification, accessible to all procedures/functions in the package.

36. Types of exceptions

Predefined, user-defined, unnamed.

37. Predefined exceptions

Examples: NO_DATA_FOUND, ZERO_DIVIDE, TOO_MANY_ROWS.

38. Raising exceptions

Use RAISE or RAISE_APPLICATION_ERROR(-20001, 'message').

39. Handling multiple exceptions

Use multiple WHEN clauses in the EXCEPTION section.

40. Exception propagation

An exception propagates to the calling block if not handled locally.

41. What is dynamic SQL?

SQL that is built and executed at runtime using EXECUTE IMMEDIATE or DBMS_SQL.

42. Difference between EXECUTE IMMEDIATE and DBMS_SQL

EXECUTE IMMEDIATE: simpler, for most dynamic SQL needs.
DBMS_SQL: more flexible, used for complex cases.

43. Bulk collect

Fetches multiple rows into a collection in one go, improving performance.

44. FORALL statement

Executes DML for all elements in a collection in one go.

45. Autonomous transaction

Independent transaction using PRAGMA AUTONOMOUS_TRANSACTION.

PL/SQL Interview Questions & Answers

46. How to schedule jobs

Use DBMS_SCHEDULER or DBMS_JOB to schedule procedures/functions.

47. COMMIT, ROLLBACK, SAVEPOINT

COMMIT: save changes.
ROLLBACK: undo changes.
SAVEPOINT: rollback partially.

48. Performance tuning tips

Use indexes, bulk operations, minimize context switches, avoid unnecessary loops.

49. Security in PL/SQL

Use definer rights/invoker rights, limit privileges, validate inputs.

50. Real-time use cases of PL/SQL

Data validation, audit logging, automation, enforcing business rules.