**General Instructions:**

* **Follow the instructions given in each section.**
* **Make sure that you attempt the questions in order.**

**SECTION-A (10\*1 mark=10 marks)**

***(All questions are compulsory)***

1. Which of the following is true about a "REPEAT" loop in programming?
   1. **The loop body is executed at least once, even if the condition is false initially**
   2. The loop body is executed only if the condition is true
   3. The loop body is executed until the condition becomes true
   4. The loop body is executed a fixed number of times
2. In SQL, the "HAVING" clause is used to:
   1. **Filter rows based on a condition in a SELECT statement**
   2. Specify the order of result rows
   3. Join multiple tables based on a common column
   4. Define constraints on table columns
3. Which of the following is true about BEFORE triggers?
   1. They are fired after the triggering event occurs
   2. **They are fired before the triggering event occurs**
   3. They are fired instead of the triggering event
   4. They are not supported in DBMS
4. Which keyword is used to define a procedure in SQL?
   1. **PROCEDURE**
   2. FUNCTION
   3. METHOD
   4. DECLARE
5. Which recovery technique requires that the transaction redo all its changes during the recovery process?
   1. Undo logging
   2. **Redo logging**
   3. Deferred update
   4. Shadow paging
6. Which of the following is a disadvantage of the shadow paging recovery technique?
   1. **Increased disk space requirement**
   2. Slower recovery process
   3. Limited support for concurrent transactions
   4. Difficulty in implementing locking mechanisms
7. Which of the following is a property of a Serializable schedule?
   1. Recoverability
   2. **Cascadeless**
   3. Starvation-free
   4. Deadlock-free
8. Which of the following is an example of a schedule that is not recoverable?
   1. S: r1(A), w1(B), r2(A), w2(B)
   2. **S: r1(A), r2(A), w2(B), w1(B)**
   3. S: r1(A), w1(B), w2(A), r2(B)
   4. S: r1(A), w1(B), r2(A), r2(B)
9. Which of the following recovery techniques ensures that all the changes made by a transaction are either committed or rolled back?
   1. Undo logging
   2. Redo logging
   3. Deferred update
   4. **Immediate update**
10. Which recovery technique uses a log file to record all the changes made to the database?
    1. Checkpointing
    2. Shadow paging
    3. Deferred update
    4. **Log-based recovery**

**SECTION-B (5\*2 mark=10 marks)**

***(All questions are compulsory)***

1. What is the output of the following program?

DECLARE

i NUMBER := 1;

BEGIN

LOOP

IF i > 5 THEN

EXIT;

END IF;

DBMS\_OUTPUT.PUT\_LINE(i);

i := i + 1;

END LOOP;

END;

1. **1 2 3 4 5**
2. 1 2 3 4
3. 1 2 3
4. 1 2
5. DECLARE

CURSOR emp\_cursor IS

SELECT employee\_id, first\_name FROM employees;

What happens if the cursor returns no rows when it is opened and fetched in PL/SQL?

1. An exception is raised, and the program terminates.
2. The cursor remains open but does not fetch any rows.
3. The cursor is automatically closed.
4. **The program continues execution without any error.**

13) In PL/SQL, how can you drop a view named "my\_view" from the database?

1. **DROP VIEW my\_view;**
2. DELETE VIEW my\_view;
3. REMOVE VIEW my\_view;
4. DESTROY VIEW my\_view;

14) Which statement is used to define a cursor inside a package specification in PL/SQL?

1. CURSOR
2. DECLARE CURSOR
3. **CURSOR IS**
4. CURSOR DECLARE

15) What is the purpose of the :OLD and :NEW keywords in a trigger?

1. To reference the parent table of the trigger.
2. To reference variables declared within the trigger body.
3. To reference the records in the trigger body.
4. **To reference the old and new values of a column during an update or delete operation.**

**SECTION-C(Coding Question) (4x5 marks=20 marks)**

16) Write a PL/SQL program to generate the Fibonacci series up to a given number of terms.

Solution:

**DECLARE**

**num\_terms NUMBER;**

**first\_term NUMBER := 0;**

**second\_term NUMBER := 1;**

**next\_term NUMBER;**

**BEGIN**

**num\_terms := &num\_terms\_input;**

**DBMS\_OUTPUT.PUT\_LINE('Fibonacci Series:');**

**DBMS\_OUTPUT.PUT(first\_term || ' ' || second\_term);**

**FOR i IN 3..num\_terms LOOP**

**next\_term := first\_term + second\_term;**

**DBMS\_OUTPUT.PUT(' ' || next\_term);**

**first\_term := second\_term;**

**second\_term := next\_term;**

**END LOOP;**

**DBMS\_OUTPUT.PUT\_LINE(' ');**

**END;**

17) Create a PL/SQL procedure to determine whether a given number is even or odd.

Solution:

**CREATE OR REPLACE PROCEDURE check\_even\_odd(num IN NUMBER) IS**

**BEGIN**

**IF MOD(num, 2) = 0 THEN**

**DBMS\_OUTPUT.PUT\_LINE(num || ' is even');**

**ELSE**

**DBMS\_OUTPUT.PUT\_LINE(num || ' is odd');**

**END IF;**

**END;**

**/**

18) Create a PL/SQL Package to Calculate Factorial of a Number.

Solution:

**CREATE OR REPLACE PACKAGE Factorial\_Pkg AS**

**FUNCTION Calculate\_Factorial(number NUMBER) RETURN NUMBER;**

**END Factorial\_Pkg;**

**/**

**CREATE OR REPLACE PACKAGE BODY Factorial\_Pkg AS**

**FUNCTION Calculate\_Factorial(number NUMBER) RETURN NUMBER IS**

**result NUMBER := 1;**

**BEGIN**

**IF number = 0 OR number = 1 THEN**

**RETURN 1;**

**ELSE**

**FOR i IN 2..number LOOP**

**result := result \* i;**

**END LOOP;**

**RETURN result;**

**END IF;**

**END;**

**END Factorial\_Pkg;**

**/**

19) Create an "INSTEAD OF INSERT" trigger that converts an uppercase string into a lowercase string before inserting it into the "names" table.

Solution:

**CREATE OR REPLACE TRIGGER tr\_convert\_to\_lowercase**

**INSTEAD OF INSERT ON names**

**FOR EACH ROW**

**BEGIN**

**:NEW.name := LOWER(:NEW.name);**

**INSERT INTO names (name) VALUES (:NEW.name);**

**END;**

**/**