



# **MARRI LAXMAN REDDY** **INSTITUTE OF TECHNOLOGY AND MANAGEMENT**

(AN AUTONOMOUS INSTITUTION)

(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)

Accredited by NBA and NAAC with 'A' Grade & Recognized Under Section 2(f) & 12(B) of the UGC act, 1956

---

## **INDEX**

<b>S.No</b>	<b>Content</b>	<b>Page No</b>
1	Preface	4
2	Acknowledgement	5
3	General Instructions	6
4	Institute Vision and Mission	7
5	Department Vision and Mission	8
6	Programme Outcomes	9-11
7	Programme Educational Objectives	12
8	Programme Specific Outcomes	13
9	Course Structure	14
10	Course Objectives and Outcomes	15
11	Course Syllabus	16
12	Course Experiments	17-34



# **MARRI LAXMAN REDDY** **INSTITUTE OF TECHNOLOGY AND MANAGEMENT**

(AN AUTONOMOUS INSTITUTION)

(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)

Accredited by NBA and NAAC with 'A' Grade & Recognized Under Section 2(f) & 12(B) of the UGC act, 1956

---

## **PREFACE**

This book, “**Database Programming with PL/SQL Lab Manual**”, is designed to provide students with a clear understanding of database programming concepts using PL/SQL. The manual focuses on practical implementation of SQL and PL/SQL concepts through well-structured laboratory exercises.

The primary objective of this lab manual is to help students develop hands-on skills in writing PL/SQL programs, creating procedures, functions, triggers, cursors, and packages, and understanding exception handling techniques. Each experiment is presented in a simple and systematic manner, enabling learners to grasp both theoretical concepts and their practical applications effectively.

This manual is prepared in accordance with the prescribed syllabus and aims to bridge the gap between classroom learning and real-time database development practices. It will serve as a valuable resource for students pursuing courses in Computer Science, Information Technology, and related disciplines.

I hope this lab manual will enhance students’ analytical thinking, programming skills, and confidence in working with database systems.



# **MARRI LAXMAN REDDY** **INSTITUTE OF TECHNOLOGY AND MANAGEMENT**

(AN AUTONOMOUS INSTITUTION)

(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)

Accredited by NBA and NAAC with 'A' Grade & Recognized Under Section 2(f) & 12(B) of the UGC act, 1956

---

## **ACKNOWLEDGEMENT**

It has been a rewarding experience working on the Database Programming with PL\SQL Laboratory Manual.

We would like to express our sincere gratitude to **Dr.K.Abdul Basith**, Professor and Head of the Department of Computer Science and Engineering, Marri Laxman Reddy Institute of Technology & Management, for his constant encouragement, valuable guidance, and continuous support in the preparation of this manual.

We are deeply indebted and gratefully acknowledge the support and motivation provided by the **Dr.P.Sridhar** Director, Marri Laxman Reddy Institute of Technology & Management, for granting us the opportunity and necessary resources to develop this laboratory manual.

Our heartfelt thanks to **Dr. R. Murali Prasad**, Principal, Marri Laxman Reddy Institute of Technology & Management, for his insightful suggestions, timely feedback, and academic guidance throughout the preparation of this document.

Finally, we extend our sincere appreciation to all the faculty members of the CSE Department whose encouragement, cooperation, and constructive inputs have played a significant role in helping us accomplish this work successfully.



# **MARRI LAXMAN REDDY** **INSTITUTE OF TECHNOLOGY AND MANAGEMENT**

(AN AUTONOMOUS INSTITUTION)

(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)

Accredited by NBA and NAAC with 'A' Grade & Recognized Under Section 2(f) & 12(B) of the UGC act, 1956

---

## **COMPUTER SCIENCE AND ENGINEERING**

### **DATABASE PROGRAMMING WITH PL\SQL LAB MANUAL**

#### **GENERAL INSTRUCTIONS**

- Students are instructed to attend the Database Programming with PL\SQL laboratory on time. Late comers will not be entertained in the lab.
- Students must be punctual. Experiments conducted during the session will not be repeated for those who are absent or late.
- Students are expected to come prepared with the theory and procedure of the experiment scheduled for the day.
- The use of mobile phones inside the lab is strictly prohibited.
- Any damage or loss of system components such as keyboard, mouse, or other peripherals during the lab session will be the responsibility of the student, and a fine or penalty will be imposed accordingly.
- Students must update their lab records and observation books session-wise. Before leaving the lab, the student should get their observation book signed by the concerned faculty member.
- Lab records must be submitted in the next lab session to the respective faculty members in the staffroom for correction and return.
- Students should not move around or disturb others during the lab session.
- In case of any emergency, students must obtain written permission from the concerned faculty member before leaving the laboratory.
- Faculty members reserve the right to suspend any student from the lab session on disciplinary grounds



# **MARRI LAXMAN REDDY** **INSTITUTE OF TECHNOLOGY AND MANAGEMENT**

(AN AUTONOMOUS INSTITUTION)

(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)

Accredited by NBA and NAAC with 'A' Grade & Recognized Under Section 2(f) & 12(B) of the UGC act, 1956

---

## **COMPUTER SCIENCE AND ENGINEERING** **DATABASE PROGRAMMING WITH PL\SQL LAB MANUAL**

### **INSTITUTE VISION AND MISSION**

#### **Vision:**

To be as an ideal academic institution by graduating talented engineers to be ethically strong, competent with quality research and technologies.

#### **Mission:**

- Utilize rigorous educational experiences to produce talented engineers
- Create an atmosphere that facilitates the success of students
- Programs that integrate global awareness, communication skills and Leadership qualities
- Education and Research partnership with institutions and industries to prepare the students for interdisciplinary research



# **MARRI LAXMAN REDDY** **INSTITUTE OF TECHNOLOGY AND MANAGEMENT**

(AN AUTONOMOUS INSTITUTION)

(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)

Accredited by NBA and NAAC with 'A' Grade & Recognized Under Section 2(f) & 12(B) of the UGC act, 1956

## **COMPUTER SCIENCE AND ENGINEERING**

### **DATABASE PROGRAMMING WITH PL\SQL LAB MANUAL**

#### **DEPARTMENT VISION AND MISSION**

##### **Vision:**

To empower the students to be technologically adept, innovative, self-motivated and responsible global citizen possessing human values and contribute significantly towards high quality technical education with ever changing world.

##### **Mission:**

- To offer high-quality education in the computing fields by providing an environment where the knowledge is gained and applied to participate in research, for both students and faculty.
- To develop the problem solving skills in the students to be ready to deal with cutting edge technologies of the industry.
- To make the students and faculty excel in their professional fields by inculcating the communication skills, leadership skills, team building skills with the organization of various co-curricular and extra-curricular programmes.
- To provide the students with theoretical and applied knowledge, and adopt an education approach that promotes lifelong learning and ethical growth.



**MARRI LAXMAN REDDY**  
**INSTITUTE OF TECHNOLOGY AND MANAGEMENT**

(AN AUTONOMOUS INSTITUTION)

(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)

Accredited by NBA and NAAC with 'A' Grade & Recognized Under Section 2(f) & 12(B) of the UGC act, 1956

**COMPUTER SCIENCE AND ENGINEERING**  
**DATABASE PROGRAMMING WITH PL\SQL LAB**

**MANUAL**

**PROGRAM OUTCOMES**

<b>PO No.</b>	<b>NBA Statement / Vital Features</b>	<b>No. of Vital Features</b>
<b>PO1.</b>	<p><b>An ability to independently carry out research /investigation and development work to solve practical problems.</b></p> <ol style="list-style-type: none"> <li>1. Research problems in Computer Science and Engineering are clearly identified and defined.</li> <li>2. Literature review highlights research gaps and suitable methods.</li> <li>3. Experiments or simulations are conducted using appropriate tools.</li> <li>4. Data collection, analyses, and interpretation systematically.</li> <li>5. Innovative approaches are applied to engineering problem-solving.</li> <li>6. Results are validated against established theories and standards.</li> </ol>	<b>6</b>
<b>PO 2.</b>	<p><b>An ability to write and present a substantial technical report/document</b></p> <ol style="list-style-type: none"> <li>1. Technical reports, dissertations, and papers are well-structured.</li> <li>2. Referencing and academic integrity practices are properly maintained.</li> <li>3. Content is presented with clarity, precision, and logical flow.</li> <li>4. Oral communication and presentation skills are effectively demonstrated.</li> <li>5. Digital tools are used for documentation and visualization.</li> <li>6. Research findings are communicated to both technical and non-technical audience.</li> </ol>	<b>6</b>
<b>PO 3.</b>	<p><b>Students should be able to demonstrate advanced proficiency in Computer Science and allied emerging areas of Engineering.</b></p> <ol style="list-style-type: none"> <li>i. In-Depth Technical Knowledge</li> <li>ii. Advanced Problem-Solving and Algorithmic Skills.</li> <li>iii. Hands-On Practical Expertise</li> <li>iv. Research and Innovation Aptitude</li> <li>v. Interdisciplinary Integration</li> </ol>	<b>5</b>
<b>PO 4.</b>	<p><b>Students should be able to identify, analyze, and effectively solve complex real-world problems by applying advanced computing concepts, while considering</b></p>	

	<p><b>solutions from a global perspective</b></p> <ul style="list-style-type: none"> <li>i. Ability to break down multifaceted problems into manageable parts and develop effective solutions using advanced computing techniques.</li> <li>ii. Analytical Thinking and Critical Reasoning</li> <li>iii. Advanced Computing Knowledge.</li> <li>iv. Capability to understand and model complex systems, considering interdependencies and dynamic behaviors within global contexts.</li> <li>v. Global and Societal Awareness.</li> <li>vi. Aptitude for developing novel approaches and innovative solutions to address complex challenges.</li> <li>vii. Ability to work effectively in diverse teams and integrate knowledge from multiple disciplines to solve global problems.</li> <li>viii. Skill in clearly presenting problem analyses, solutions, and their implications to technical and non-technical global audience.</li> </ul>	<b>8</b>
<p><b>PO 5.</b></p>	<p><b>An ability to acquire and apply advanced technical knowledge, professional skills, and modern computing tools to develop sustainable solutions</b></p> <ul style="list-style-type: none"> <li>i. Ability to continuously learn and apply cutting-edge technical knowledge in computing and related fields.</li> <li>ii. Proficiency with Modern Computing Tools.</li> <li>iii. Capability to design and implement solutions that are environmentally, and economically feasible.</li> <li>iv. Understanding of professional ethics and responsibility in creating technology solutions with long-term positive impact.</li> <li>v. Integration of Multidisciplinary Knowledge.</li> <li>vi. Adaptability to sustainable practices.</li> </ul>	<b>6</b>
<p><b>PO 6.</b></p>	<p><b>An Ability to recognize the significance of lifelong learning and actively pursue continuous professional development by adapting technologies in emerging areas.</b></p> <ul style="list-style-type: none"> <li>i. Self-Directed Learning.</li> <li>ii. Skill in quickly learning and integrating new technologies and tools as they emerge in the field.</li> <li>iii. Capability to assess the relevance and impact of emerging technologies and decide on their applicability.</li> <li>iv. Continuous Professional Development Planning.</li> <li>v. Ability to engage with professional communities, attend workshops, and collaborate to stay updated.</li> <li>vi. Habit of regularly reflecting on personal growth, learning experiences, and professional competencies to improve continuously.</li> </ul>	<b>6</b>



# **MARRI LAXMAN REDDY** **INSTITUTE OF TECHNOLOGY AND MANAGEMENT**

(AN AUTONOMOUS INSTITUTION)

(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)

Accredited by NBA and NAAC with 'A' Grade & Recognized Under Section 2(f) & 12(B) of the UGC act, 1956

## **COMPUTER SCIENCE AND ENGINEERING**

### **DATABASE PROGRAMMING WITH PL\SQL LAB MANUAL**

#### **PROGRAM EDUCATIONAL OBJECTIVES**

<b>Sl. No.</b>	<b>PEOs Name</b>	<b>Program Education Objective Statements</b>
<b>1</b>	<b>PEO – 1</b>	Graduates will achieve professional excellence and success in the field of Computer Science and Engineering by applying strong technical foundations and problem-solving skills to contribute effectively to industry, academia, and entrepreneurship.
<b>2</b>	<b>PEO – 2</b>	Graduates will demonstrate a commitment to lifelong learning by continuously enhancing their knowledge and skills through professional development and self-directed learning to effectively adapt to evolving global challenges.
<b>3</b>	<b>PEO – 3</b>	Graduates of the Computer Science and Engineering program will actively pursue advanced research, contributing to the development of solutions for complex problems and the generation of new knowledge to effectively address real-world challenges.
<b>4</b>	<b>PEO – 4</b>	Graduates will exhibit professionalism, effective communication, leadership skills, and ethical responsibility while working in multidisciplinary teams to deliver computing solutions that address societal needs and contribute to sustainable development.



**MARRI LAXMAN REDDY**  
**INSTITUTE OF TECHNOLOGY AND MANAGEMENT**

(AN AUTONOMOUS INSTITUTION)

(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)

Accredited by NBA and NAAC with 'A' Grade & Recognized Under Section 2(f) & 12(B) of the UGC act, 1956

---

**COMPUTER SCIENCE AND ENGINEERING**

**DATABASE PROGRAMMING WITH PL\SQL LAB MANUAL**

**PROGRAM SPECIFIC OUTCOMES**

**PSO1:** Applications of Computing: Ability to use knowledge in various domains to provide solution to new ideas and innovations.

**PSO2:** Programming Skills: Identify required data structures, design suitable algorithms, develop and maintain software for real world problems.



# **MARRI LAXMAN REDDY** **INSTITUTE OF TECHNOLOGY AND MANAGEMENT**

(AN AUTONOMOUS INSTITUTION)

(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)

Accredited by NBA and NAAC with 'A' Grade & Recognized Under Section 2(f) & 12(B) of the UGC act, 1956

---

## **COMPUTER SCIENCE AND ENGINEERING** **DATABASE PROGRAMMING WITH PL\SQL LAB MANUAL**

### **Course Structure**

Database Programming with PL\SQL Lab will have a continuous evaluation during II semester for 40 sessional marks and 60 end semester examination marks.

Out of the 40 marks for internal evaluation, day-to-day work in the laboratory shall be evaluated for 20 marks and internal practical examination shall be evaluated for 10 marks conducted by the laboratory teacher concerned.

The end semester examination shall be conducted with an external examiner and internal examiner. The external examiner shall be appointed by the principal / Chief Controller of examinations



# **MARRI LAXMAN REDDY** **INSTITUTE OF TECHNOLOGY AND MANAGEMENT**

(AN AUTONOMOUS INSTITUTION)

(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)

Accredited by NBA and NAAC with 'A' Grade & Recognized Under Section 2(f) & 12(B) of the UGC act, 1956

---

## **COMPUTER SCIENCE AND ENGINEERING**

### **DATABASE PROGRAMMING WITH PL\SQL LAB MANUAL**

#### **OBJECTIVE:**

- To be able to introduce core programming basics and program design with functions using Python programming language.
- To understand a range of Object-Oriented Programming, as well as in-depth data and information processing techniques.
- To understand the high-performance programs designed to strengthen the practical expertise.

#### **OUTCOMES:**

Upon the completion of Operating Systems practical course, the student will be able to:

- Student should be able to understand the basic concepts scripting and the contributions of scripting language
- Ability to explore python especially the object oriented concepts, and the built in objects of Python.



# MARRI LAXMAN REDDY INSTITUTE OF TECHNOLOGY AND MANAGEMENT

(AN AUTONOMOUS INSTITUTION)

(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)

Accredited by NBA and NAAC with 'A' Grade & Recognized Under Section 2(f) & 12(B) of the UGC act, 1956

---

## 2415832: DATABASE PROGRAMMING WITH PL/SQL LAB (Lab - II)

M.Tech CSE I Year I Sem.

L	T	P	C
0	0	4	2

### Course Objectives:

1. Knowledge on significance of SQL fundamentals.
2. Evaluate functions and triggers of PL/SQL
3. Knowledge on control structures, packages in PL/SQL and its applications

### Course Outcomes:

1. Understand importance of PL/SQL basics
2. Implement functions and procedures using PL/SQL
3. Understand the importance of triggers in database

### List of Experiments:

1. Write a PL/SQL program using FOR loop to insert ten rows into a database table.
2. Given the table EMPLOYEE (EmpNo, Name, Salary, Designation, DeptID), write a cursor to select the five highest paid employees from the table.
3. Illustrate how you can embed PL/SQL in a high-level host language such as C/Java And demonstrates how a banking debit transaction might be done.
4. Given an integer i, write a PL/SQL procedure to insert the tuple (i, 'xxx') into a given relation.
5. Write a PL/SQL program to demonstrate Exceptions.
6. Write a PL/SQL program to demonstrate Cursors.
7. Write a PL/SQL program to demonstrate Functions.
8. Write a PL/SQL program to demonstrate Packages.
9. Write PL/SQL queries to create Procedures.
10. Write PL/SQL queries to create Triggers

## **Experiment-1**

**Write a PL/SQL program using FOR loop to insert ten rows into a database table.**

```
DECLARE
```

```
-- Declare a variable for the loop index
```

```
v_counter NUMBER;
```

```
BEGIN
```

```
-- FOR loop to insert 10 rows
```

```
FOR v_counter IN 1..10 LOOP
```

```
-- Insert a row into your target table (replace `your_table` with the actual table name)
```

```
INSERT INTO your_table (column1, column2)
```

```
VALUES (v_counter, 'Value ' || v_counter); -- Replace with your actual column values
```

```
END LOOP;
```

```
COMMIT;
```

```
DBMS_OUTPUT.PUT_LINE ('10 rows have been inserted successfully.');
```

```
END;
```

### **OUTPUT:**

10 rows have been inserted successfully.

## **Viva Questions**

1. What is a FOR loop in PL/SQL?
2. What is the syntax of a numeric FOR loop?
3. How many times does a FOR loop from 1..10 execute?
4. Is the loop variable explicitly declared?
5. Can we modify the loop counter inside the FOR loop?
6. What is the default step/increment value in a FOR loop?
7. What is the purpose of the INSERT statement?
8. What type of SQL command is INSERT (DDL/DML)?
9. What is the use of COMMIT in the program?
10. What happens if COMMIT is not used?
11. What is string concatenation operator in PL/SQL?
12. What will happen if the table has a primary key constraint and duplicate values are inserted?
13. What is the difference between FOR loop and WHILE loop?
14. What is an anonymous block in PL/SQL?
15. What is the difference between SQL and PL/SQL?

## Experiment-2

Given the table EMPLOYEE (EmpNo, Name, Salary, Designation, DeptID) write a cursor to select the five highest paid employees from the table in database programming with PL/SQL.

```
DECLARE
-- Declare a cursor to fetch the five highest-paid employees
CURSOR emp_cursor IS
SELECT EmpNo, Name, Salary, Designation, DeptID
FROM EMPLOYEE
ORDER BY Salary DESC
FETCH FIRST 5 ROWS ONLY; -- This limits the result to the top 5 highest paid employees
-- Declare variables to hold the values fetched from the cursor
v_empNo EMPLOYEE.EmpNo%TYPE;
v_name EMPLOYEE.Name%TYPE;
v_salary EMPLOYEE.Salary%TYPE;
v_designation EMPLOYEE.Designation%TYPE;
v_deptID EMPLOYEE.DeptID%TYPE;
BEGIN
OPEN emp_cursor;
-- Loop through the cursor and fetch each row
LOOP
FETCH emp_cursor INTO v_empNo, v_name, v_salary, v_designation, v_deptID;
-- Exit the loop if no more rows are fetched
EXIT WHEN emp_cursor%NOTFOUND;
-- Print the details of each employee
DBMS_OUTPUT.PUT_LINE ('EmpNo: ' || v_empNo || ', Name: ' || v_name || ', Salary: ' || v_salary ||
', Designation: ' || v_designation || ', DeptID: ' || v_deptID);
END LOOP;
CLOSE emp_cursor;
END;
```

## OUTPUT:

EmpNo: 101, Name: John Doe, Salary: 120000, Designation: Manager, DeptID: 10

EmpNo: 102, Name: Jane Smith, Salary: 110000, Designation: Senior Developer, DeptID: 20

## Viva Questions

16

1. What is a cursor in PL/SQL?
2. Why do we use a cursor in database programming?

3. What are the two types of cursors in PL/SQL?
4. What is an explicit cursor?
5. What is the syntax to declare a cursor?
6. Why is ORDER BY Salary DESC used in this query?
7. What does FETCH FIRST 5 ROWS ONLY do?
8. What are the steps involved in using an explicit cursor?
9. What is the purpose of the OPEN statement?
10. What does %NOTFOUND cursor attribute indicate?
11. What are cursor attributes? Name any four.
12. What happens if you try to FETCH from a closed cursor?
13. What is the difference between implicit and explicit cursor?
14. Can we use a cursor FOR loop instead of OPEN-FETCH-CLOSE?
15. What is the use of DBMS\_OUTPUT.PUT\_LINE in cursor programs?

### **Experiment-3**

**Illustrate how you can embed PL/SQL in a high-level host language such as C/JAVA and demonstrates how a banking debit transaction might be done.**

```
import java.sql.*;
```



```

p_debit_amount IN NUMBER,
p_error_message OUT VARCHAR2
) AS
v_balance NUMBER;
BEGIN
SELECT balance INTO v_balance FROM accounts WHERE account_number = p_account_number;
IF v_balance >= p_debit_amount THEN
UPDATE accounts SET balance = balance - p_debit_amount WHERE account_number =
p_account_number;
COMMIT;
p_error_message: = NULL; -- Success
ELSE
p_error_message: = 'Insufficient funds.';
END IF;
EXCEPTION
WHEN NO_DATA_FOUND THEN
p_error_message: = 'Account not found.';
WHEN OTHERS THEN
p_error_message: = SQLERRM;
ROLLBACK;
END;
-- Sample accounts table (for testing):
CREATE TABLE accounts (
account_number NUMBER PRIMARY KEY,
balance NUMBER
);
INSERT INTO accounts (account_number, balance) VALUES (12345, 1000.00);
INSERT INTO accounts (account_number, balance) VALUES (67890, 50.00);
COMMIT;

```

**OUTPUT:**

Debit successful for account: 12345 New Balance: 900.0  
Debit failed: Insufficient funds.  
Debit failed: Account not found.  
Database error: [Specific error message from SQL Exception]

**Viva Questions**

1. What is meant by embedding PL/SQL in a host language?
2. What is a host language? Give examples. <sup>19</sup>
3. What is JDBC in Java?

4. What is Callable Statement in JDBC?
5. Why do we use stored procedures in banking applications?
6. What is the purpose of the FOR UPDATE clause in the debit transaction?
7. What is transaction management?
8. What are ACID properties in DBMS?
9. What is the role of COMMIT and ROLLBACK in a debit transaction?
10. What happens if the account balance is insufficient?
11. What is RAISE\_APPLICATION\_ERROR in PL/SQL?
12. What is the difference between Prepared Statement and Callable Statement?
13. What is exception handling in Java database programs?
14. What is a bind variable and why is it important?
15. Why is embedding PL/SQL better than writing raw SQL in the application code?

#### **Experiment-4**

**Given an integer i, write a PL/SQL procedure to insert the tuple (i, 'xxx') into a given relation.**

20

```
CREATE OR REPLACE PROCEDURE insert_tuple (i IN INT) AS
```

```

BEGIN
INSERT INTO your_table (column1, column2)
VALUES (i, 'xxx'); -- Assuming column1 is an integer and column2 is a string.
COMMIT;
DBMS_OUTPUT.PUT_LINE ('Tuple (' || i || ', "xxx") inserted successfully. ');
EXCEPTION
WHEN OTHERS THEN
-- In case of an error, rollback the transaction and output the error message.
ROLLBACK;
DBMS_OUTPUT.PUT_LINE ('Error: ' || SQLERRM);
END insert_tuple;

```

**OUTPUT:**

Tuple (123, 'xxx') inserted successfully.  
Error: [error\_message]

**Viva Questions**

1. What is a stored procedure in PL/SQL?
2. What is the syntax to create a procedure?
3. What is the purpose of CREATE OR REPLACE PROCEDURE?
4. What type of parameter is i in this procedure?
5. What are the different parameter modes in PL/SQL?
6. What is the difference between formal and actual parameters?
7. What SQL command is used to insert a tuple into a relation?
8. Is INSERT a DDL or DML statement?
9. Why do we use COMMIT after INSERT?
10. What happens if the table has a primary key constraint and i is duplicated?
11. How do you execute (call) a stored procedure?
12. What is the difference between a procedure and a function?
13. Can a procedure return a value?
14. What privileges are required to create a procedure?
15. What happens if an exception occurs inside the procedure?

**Experiment-5**

**Write a PL/SQL program to demonstrate Exceptions.**

```

DECLARE
-- Declare a variable

```

```

v_number NUMBER := 10;
-- Declare a user-defined exception
e_invalid_number EXCEPTION;
BEGIN
DBMS_OUTPUT.PUT_LINE ('Starting the program...');
BEGIN
DBMS_OUTPUT.PUT_LINE ('Attempting division by zero...');
v_number := v_number / 0; -- This will raise a ZERO_DIVIDE exception
EXCEPTION
WHEN ZERO_DIVIDE THEN
DBMS_OUTPUT.PUT_LINE ('Error: Division by zero occurred. ');
END;
BEGIN
DBMS_OUTPUT.PUT_LINE ('Attempting to check invalid number...');
IF v_number > 5 THEN
RAISE e_invalid_number; -- Raise user-defined exception
END IF;
EXCEPTION
WHEN e_invalid_number THEN
DBMS_OUTPUT.PUT_LINE ('Error: Invalid number, value is greater than 5. ');
END;
BEGIN
DBMS_OUTPUT.PUT_LINE ('Performing normal operation...');
v_number := v_number + 5;
DBMS_OUTPUT.PUT_LINE ('Result of operation: ' || v_number);
END;
DBMS_OUTPUT.PUT_LINE ('End of program. ');
EXCEPTION
WHEN OTHERS THEN
DBMS_OUTPUT.PUT_LINE ('Unhandled exception: ' || SQLERRM);
END;

```

### **OUTPUT:**

Starting the program...  
Attempting division by zero...  
Error: Division by zero occurred.

### **Viva Questions**

1. What is an exception in PL/SQL?
2. What are the types of exceptions in PL/SQL?
3. What is a predefined exception? Give examples.
4. What is a user-defined exception? 22
5. What is the structure of an exception handling block?

6. What is the purpose of the EXCEPTION block?
7. What is the WHEN OTHERS clause?
8. What is the ZERO\_DIVIDE exception?
9. What is the use of the RAISE statement?
10. What is RAISE\_APPLICATION\_ERROR?
11. What are SQLCODE and SQLERRM functions?
12. What happens if an exception is not handled?
13. What is exception propagation?
14. What is PRAGMA EXCEPTION\_INIT used for?
15. What is the difference between an error and an exception in PL/SQL?

### **Experiment-6**

**Write a PL/SQL program to demonstrate Cursors.**

```
DECLARE
CURSOR emp_cursor IS
SELECT employee_id, first_name, last_name, salary
FROM employees
WHERE department_id = 10;
v_employee_id employees.employee_id%TYPE;
v_first_name employees.first_name%TYPE;
v_last_name employees.last_name%TYPE;
v_salary employees.salary%TYPE;
BEGIN
OPEN emp_cursor;
```

```

LOOP
FETCH emp_cursor INTO v_employee_id, v_first_name, v_last_name, v_salary;
EXIT WHEN emp_cursor%NOTFOUND;
DBMS_OUTPUT.PUT_LINE ('Employee ID: ' || v_employee_id ||
', Name: ' || v_first_name || ' ' || v_last_name ||
', Salary: ' || v_salary);
END LOOP;
CLOSE emp_cursor;
FOR emp_rec IN (SELECT employee_id, first_name, last_name, salary FROM employees WHERE
department_id = 20) LOOP
DBMS_OUTPUT.PUT_LINE ('Implicit Cursor - Employee ID: ' || emp_rec.employee_id ||
', Name: ' || emp_rec.first_name || ' ' || emp_rec.last_name ||
', Salary: ' || emp_rec.salary);
END LOOP;
END;
OUTPUT:
Employee ID: 100, Name: John Doe, Salary: 5000
Employee ID: 101, Name: Jane Smith, Salary: 5500
Implicit Cursor - Employee ID: 200, Name: Alice Johnson, Salary: 6000
Implicit Cursor - Employee ID: 201, Name: Bob Brown, Salary: 6500

```

### Viva Questions

1. What is a cursor in PL/SQL?
2. Why are cursors used in PL/SQL?
3. What are the two types of cursors?
4. What is an implicit cursor?
5. What is an explicit cursor?
6. What are the steps to use an explicit cursor?
7. What is the purpose of the OPEN statement?
8. What does the FETCH statement do?
9. What is the role of the CLOSE statement?
10. What are cursor attributes?
11. Name four cursor attributes in PL/SQL.
12. What is %NOTFOUND used for?
13. What is a cursor FOR loop?
14. What is the difference between cursor and cursor FOR loop?
15. What happens if a cursor is not closed properly?

### Experiment-7

**Write a PL/SQL program to demonstrate Functions.**

```

CREATE OR REPLACE FUNCTION calculate_annual_salary (p_monthly_salary IN NUMBER)
RETURN NUMBER
IS
v_annual_salary NUMBER;
BEGIN
v_annual_salary := p_monthly_salary * 12;
RETURN v_annual_salary;
END;
DECLARE
v_monthly_salary NUMBER := 5000;
v_annual_salary NUMBER;

```

```
BEGIN
v_annual_salary := calculate_annual_salary(v_monthly_salary);
DBMS_OUTPUT.PUT_LINE('Monthly Salary: ' || v_monthly_salary);
DBMS_OUTPUT.PUT_LINE('Calculated Annual Salary: ' || v_annual_salary);
END;
```

**OUTPUT:**

Monthly Salary: 5000

Calculated Annual Salary: 60000

**Viva Questions**

1. What is a function in PL/SQL?
2. How is a function different from a procedure?
3. What is the syntax to create a function?
4. What is the purpose of the RETURN statement?
5. Is the RETURN statement mandatory in a function?
6. What data types can a function return?
7. Can a function return multiple values?
8. Where can a PL/SQL function be called?
9. Can a function be called inside an SQL statement?
10. What are the parameter modes allowed in functions?
11. What is function overloading?
12. What is a recursive function?
13. What happens if a function does not return a value?
14. What privileges are required to create a function?
15. What is the difference between a stored function and an anonymous block?

### Experiment-8

**Write a PL/SQL program to demonstrate Packages.**

```
CREATE OR REPLACE PACKAGE emp_salary_pkg AS
v_bonus_percentage NUMBER := 10;
FUNCTION calculate_salary_after_bonus(p_salary IN NUMBER) RETURN NUMBER;
PROCEDURE print_employee_details(p_employee_id IN NUMBER);
END emp_salary_pkg;
-- Create Package Body
CREATE OR REPLACE PACKAGE BODY emp_salary_pkg AS
FUNCTION calculate_salary_after_bonus(p_salary IN NUMBER) RETURN NUMBER IS
BEGIN
RETURN p_salary + (p_salary * v_bonus_percentage / 100);
END calculate_salary_after_bonus;
PROCEDURE print_employee_details(p_employee_id IN NUMBER) IS
v_employee_name VARCHAR2(50);
v_employee_salary NUMBER;
BEGIN
SELECT first_name || ' ' || last_name, salary
INTO v_employee_name, v_employee_salary
FROM employees
WHERE employee_id = p_employee_id;
DBMS_OUTPUT.PUT_LINE('Employee Name: ' || v_employee_name);
DBMS_OUTPUT.PUT_LINE('Employee Salary: ' || v_employee_salary);

-- Calculate and print salary after bonus
DBMS_OUTPUT.PUT_LINE('Salary After Bonus: ' ||
calculate_salary_after_bonus(v_employee_salary));
END print_employee_details;

END emp_salary_pkg;
```

### OUTPUT:

```
Employee Name: John Doe
Employee Salary: 5000
Salary After Bonus: 5500
```

### Viva Questions

1. What is a package in PL/SQL?
2. What are the two main parts of a package?
3. What is a package specification?
4. What is a package body?
5. Is it mandatory to create both specification and body?
6. What is the advantage of using packages?
7. What is encapsulation in PL/SQL packages?
8. Can we overload procedures/functions inside a package?
9. What is a package variable?
10. What is the difference between public and private components in a package?
11. What happens if the package body is modified?
12. What is package initialization?
13. What is a stateful package?
14. What privileges are required to create a package?
15. What is the difference between a standalone procedure and a packaged procedure?

### Experiment-9

**Write PL/SQL queries to create Procedures.**

```
CREATE OR REPLACE PROCEDURE display_employee_details(p_employee_id IN NUMBER)
IS
v_employee_name VARCHAR2(100);
v_employee_salary NUMBER;
BEGIN
-- Query to retrieve employee details
```

```

SELECT first_name || ' ' || last_name, salary
INTO v_employee_name, v_employee_salary
FROM employees
WHERE employee_id = p_employee_id;
-- Display employee details using DBMS_OUTPUT
DBMS_OUTPUT.PUT_LINE('Employee Name: ' || v_employee_name);
DBMS_OUTPUT.PUT_LINE('Employee Salary: ' || v_employee_salary);
EXCEPTION
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('Error: No employee found with employee_id ' || p_employee_id);
WHEN OTHERS THEN
DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
END display_employee_details;
-- Calling the procedure to display details of an employee
BEGIN
-- Example: Displaying employee details for employee_id 100
display_employee_details(100);
END;
```

**OUTPUT:**

Employee Name: John Doe

Employee Salary: 5000

**Viva Questions**

1. What is a procedure in PL/SQL?
2. What is the difference between a procedure and a function?
3. What are the parameter modes in PL/SQL?
4. What is the purpose of the IN parameter?
5. What is the purpose of the OUT parameter?
6. What is the purpose of the IN OUT parameter?
7. What is DBMS\_OUTPUT.PUT\_LINE?
8. What is CREATE OR REPLACE PROCEDURE?
9. How do you call a procedure in PL/SQL?
10. What is exception handling in PL/SQL?
11. What is a cursor?
12. What is %ROWTYPE?
13. What is the use of COMMIT in a procedure?
14. Can a procedure return a value?

15. What is the difference between implicit and explicit cursors?

### **Experiment-10**

**Write PL/SQL queries to create Triggers.**

```
CREATE TABLE salary_audit (  
audit_id NUMBER PRIMARY KEY,  
employee_id NUMBER,  
old_salary NUMBER,  
new_salary NUMBER,  
change_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
changed_by VARCHAR2(50)  
);  
  
-- Create a trigger to log salary updates  
CREATE OR REPLACE TRIGGER salary_update_trigger  
AFTER UPDATE OF salary  
ON employees  
FOR EACH ROW  
BEGIN
```

```

-- Insert into salary_audit table to log the salary change
INSERT INTO salary_audit (employee_id, old_salary, new_salary, changed_by)
VALUES (:OLD.employee_id, :OLD.salary, :NEW.salary, USER);
-- Display the change in DBMS_OUTPUT
DBMS_OUTPUT.PUT_LINE ('Salary updated for Employee ID: ' || :NEW.employee_id);
DBMS_OUTPUT.PUT_LINE ('Old Salary: ' || :OLD.salary || ', New Salary: ' || :NEW.salary);
DBMS_OUTPUT.PUT_LINE ('Changed by: ' || USER);
END salary_update_trigger;
-- Update the salary of an employee (for example, employee_id = 100)
UPDATE employees
SET salary = 6000
WHERE employee_id = 100;
-- Query the salary_audit table to see the logged changes
SELECT * FROM salary_audit;

```

**OUTPUT:**

```

Salary updated for Employee ID: 100
Old Salary: 5000, New Salary: 6000
Changed by: HR_USER

```

**Viva Questions**

1. What is a trigger in PL/SQL?
2. What are the types of triggers?
3. What is the difference between BEFORE and AFTER triggers?
4. What is a row-level trigger?
5. What is a statement-level trigger?
6. What is the use of :NEW keyword?
7. What is the use of :OLD keyword?
8. What is an INSTEAD OF trigger?
9. Can a trigger commit or rollback transactions?
10. What is RAISE\_APPLICATION\_ERROR?
11. What is a WHEN clause in trigger?
12. How do you disable a trigger?

13. How do you drop a trigger?
14. What is a mutating table error?
15. What are the advantages of using triggers?