KG REDDY
College of Engineering
& Technology
AN AUTONOMOUS INSTITUTION

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

DBMS with PL/SQL LAB MANUAL

Subject Code: KGR2358112

with PL/ SQL(Lab-II)

Regulation: KGR23

Academic Year: 2023-24

M .Tech CSE/CS I Year /I Sem

COMPUTER SCIENCE AND ENGINEERING

KG REDDY COLLEGE OF ENGINEERING AND TECHNOLOGY

Affiliated To JNTUH, Chilkur,(V), Moinabad(M) R. R Dist, TS-501504

VISION AND MISSION OF THE INSTITUTION

VISION:

To become an institution which is internationally recognized for its holistic approach to engineering, innovative teaching and learning culture, research and entrepreneurial ecosystem, and sustainable social impact in the community.

MISSION:

• To offer undergraduate and post-graduate programs which are supported through industry relevant curriculum and innovative teaching and learning processes that would help students succeed in their professional careers.
• To provide faculty and students with an ecosystem that fosters innovation, research, entrepreneurship, and international exposure through strategic partnerships with government organizations and collaboration with industries.
• To provide holistic learning environment to students which will contribute to their personal and professional growth and enable them to become leaders in their respective fields.
• To contribute to the development of the region by using our technological expertise to work with nearby communities and support them in their social and economic development.

\

KG REDDY College of Engineering & Technology
AN AUTONOMOUS INSTITUTION

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

VISION:

To be recognized as a department of excellence by stimulating a learning environment in which students and faculty will thrive and grow to achieve their professional, institutional and societal goals.

MISSION:

• To provide high quality technical education to students that will enable life-long learning and build expertise in advanced technologies in Computer Science and Engineering.
• To promote research and development by providing opportunities to solve complex engineering problems in collaboration with industry and government agencies.
• To encourage professional development of students that will inculcate ethical values and leadership skills through entrepreneurship while working with the community to address societalissues.

PROGRAM EDUCATIONAL OBJECTIVES

PEO 1: Graduates will provide solutions to difficult and challenging issues in their profession by applying computer science and engineering theory and principles.
PEO 2: Graduates have successful careers in computer science and engineering fields or will be able to successfully pursue advanced degrees.
PEO 3: Graduates will communicate effectively, work collaboratively and exhibit high levels of professionalism, moral and ethical responsibility.
PEO 4: Graduates will develop the ability to understand and analyze engineering issues in a broader perspective with ethical responsibility towards sustainable development.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PROGRAM OUTCOMES

PO I: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO II: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO III: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO IV: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO V: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
PO VI: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO VII: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of and need for sustainable development.
PO VIII: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO IX: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO X: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO XI: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO XII: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

KG REDDY
College of Engineering
& Technology
AN AUTONOMOUS INSTITUTION

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

PROGRAM SPECIFIC OUTCOMES

PSO1: The Computer Science and Engineering graduates are able to analyze, design, develop, test and apply management principles, mathematical foundations in the development of intelligent systems with computational solutions, make them to expert in designing the secure application and hardware prototype
PSO2: The graduating student will be analyze the contemporary research issues in different areasof computer science & engineering and explore research gaps, analyze and carry out research in the specialized/emerging areas.
PSO3: Develop their skills to solve problems in the broad area of programming concepts and appraise environmental and social issues with ethics and manage different projects in multi-disciplinary field to conducive in cultivating skills for successful career, entrepreneurship and higher studies.

\

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DBMS with PL/SQL Lab

Course Objectives: The course will help students

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: Upon completion of this course, the students will beable to

CO1: Understand importance of PL/SQL basics.

CO2: Implement functions and procedures using PL/SQL .

CO3: Understand the importance of triggers in database.

s.no	**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(Emp-No, Name, Salary, Designation, Dept-ID) write a cursor to select the five highest paid employees from the table.
3.	Ilustrate how you can embed plsql in a high level host language such as java and demonstrate 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 a Pl / Sql queries to create procedures
10.	Write a Pl / Sql queries to create Triggers

1.Write a PL / SQL Program using FOR Loop to insert ten rows into a database table.

```
SQL> create table t ( x int, y char(50) );
SQL>
SQL> begin
    for i in 1 .. 10
    loop
        insert into t values ( i, 'x' );
    end loop;
    commit;
end;
```

2. Given the Table Employee
(Emp No, Name, Salary, Designation, Dept.-ID)
write a cursor to select the five highest paid employees from the table.

```
declare
cursor emp is
select empnm,salary from emp_cur order by salary desc;
begin
dbms_output.put_line(‘ EMP NAME ‘ || ‘ SALARY ‘);
dbms_output.put_line(‘========================’);
for i in emp
loop
if emp%rowcount <=5 then
dbms_output.put_line(i.empnm||’ ‘ || ‘  ‘||’ ‘||i.salary);
end if;
end loop;
end;
```

3. Illustrate how you can embed Pl/ Sql in a high level host language such as java and demonstrate how a banking debit transaction might be done.

The following example shows how you can embed PL/SQL in a high-level host language such as C and demonstrates how a banking debit transaction might be done.

Input Table

```
SQL> SELECT * FROM accounts ORDER BY account_id;

ACCOUNT_ID      BAL
---------- --------
         1     1000
         2     2000
         3     1500
         4     6500
         5      500
```

PL/SQL Block in a C Program

```
/* available online in file 'sample5' */
#include <stdio.h>
  char    buf[20];
EXEC SQL BEGIN DECLARE SECTION;
  int     acct;
  double  debit;
  double  new_bal;
  VARCHAR status[65];
  VARCHAR uid[20];
  VARCHAR pwd[20];
EXEC SQL END DECLARE SECTION;

EXEC SQL INCLUDE SQLCA;

main()
{
```

```
   extern double atof();

   strcpy (uid.arr,"scott");
   uid.len=strlen(uid.arr);
   strcpy (pwd.arr,"tiger");
   pwd.len=strlen(pwd.arr);

   printf("\n\n\tEmbedded PL/SQL Debit Transaction Demo\n\n");
   printf("Trying to connect...");
   EXEC SQL WHENEVER SQLERROR GOTO errprint;
   EXEC SQL CONNECT :uid IDENTIFIED BY :pwd;
   printf(" connected.\n");
 for (;;)       /*  Loop infinitely */
 {
   printf("\n** Debit which account number? (-1 to end) ");
   gets(buf);
   acct = atoi(buf);
   if (acct == -1)  /* Need to disconnect from Oracle */
   {               /* and exit loop if account is -1 */
      EXEC SQL COMMIT RELEASE;
      exit(0);
   }

   printf("   What is the debit amount? ");
   gets(buf);
   debit = atof(buf);

   /* ---------------------------------- */
   /* ----- Begin the PL/SQL block ----- */
   /* ---------------------------------- */
   EXEC SQL EXECUTE
```

```
DECLARE
  insufficient_funds EXCEPTION;
  old_bal          NUMBER;
  min_bal          CONSTANT NUMBER := 500;
BEGIN
  SELECT bal INTO old_bal FROM accounts
    WHERE account_id = :acct;
    -- If the account doesn't exist, the NO_DATA_FOUND
    -- exception will be automatically raised.
  :new_bal := old_bal - :debit;
  IF :new_bal >= min_bal THEN
    UPDATE accounts SET bal = :new_bal
      WHERE account_id = :acct;
    INSERT INTO journal
      VALUES (:acct, 'Debit', :debit, SYSDATE);
    :status := 'Transaction completed.';
  ELSE
    RAISE insufficient_funds;
  END IF;
  COMMIT;
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    :status := 'Account not found.';
    :new_bal := -1;
  WHEN insufficient_funds THEN
    :status := 'Insufficient funds.';
    :new_bal := old_bal;
  WHEN OTHERS THEN
    ROLLBACK;
    :status := 'Error: ' || SQLERRM(SQLCODE);
    :new_bal := -1;
END;
```

```
  END-EXEC;
  /* -------------------------------- */
  /* ----- End the PL/SQL block ----- */
  /* -------------------------------- */

  status.arr[status.len] = '\0';  /* null-terminate */
                                  /* the string     */
  printf("\n\n   Status:  %s\n", status.arr);
  if (new_bal >= 0)
     printf("   Balance is now:  $%.2f\n", new_bal);
}  /* End of loop */

errprint:
  EXEC SQL WHENEVER SQLERROR CONTINUE;
  printf("\n\n>>>>> Error during execution:\n");
  printf("%s\n",sqlca.sqlerrm.sqlerrmc);
  EXEC SQL ROLLBACK RELEASE;
  exit(1);
}
```

4.Given an integer i write a Pl/ Sql procedure to insert the tuple(i,'xxx') into a given relation

```
CREATE TABLE T2 (
a INTEGER,
b CHAR(10)
);
CREATE PROCEDURE addtuple1(i IN NUMBER) AS
BEGIN
INSERT INTO T2 VALUES(i, 'xxx');
END addtuple1;
run;
```

5. Write a pl/sql program to demonstrate exceptions

```
DECLARE
   c_id customers.id%type := 8;
   c_name  customers.name%type;
   c_addr customers.address%type;
BEGIN
   SELECT  name, address INTO  c_name, c_addr
   FROM customers
   WHERE id = c_id;
DBMS_OUTPUT.PUT_LINE ('Name: '||  c_name);
 DBMS_OUTPUT.PUT_LINE ('Address: ' || c_addr);
EXCEPTION
   WHEN no_data_found THEN
     dbms_output.put_line('No such customer!');
   WHEN others THEN
    dbms_output.put_line('Error!');
END;
```

6. Write a pl/sql program to demonstrate Cursors

The following program will update the table and increase the salary of each customer by 500 and use the SQL%ROWCOUNT attribute to determine the number of rows affected

```
DECLARE
   total_rows number(2);
BEGIN
   UPDATE customers
   SET salary = salary + 500;
   IF sql%notfound THEN
      dbms_output.put_line('no customers selected');
   ELSIF sql%found THEN
      total_rows := sql%rowcount;
      dbms_output.put_line( total_rows || ' customers selected ');
   END IF;  END;
```

7.Write a pl/sql program to demonstrate Functions

```
// first lets create a table
create table section(s_id int, s_name varchar(20), strength int );
 // Inserting values into table
insert into section values(1, 'computer science', 20);
insert into section values(2, 'portal', 45);
insert into section values(3, 'geeksforgeeks', 60);
 // Defining function
create or replace function totalStrength
 // Defining return type
return integer
as
total integer:=0;
 begin
 // calculating the sum and storing it in total
select sum(strength) into total from section;
return total;
 // closing function
end totalStrength;
 set serveroutput on;
 declare
answer integer;
 begin
answer:=totalstrength();
   dbms_output.put_line('Total strength of students is  ' || answer);
end;
```

8. Write a Pl /Sql program to demonstrate Packages

Code implementation:

```
CREATE PACKAGE citi AS
   FUNCTION p_strng RETURN VARCHAR2;
END citi;
/
```

Code implementation with package body.

```
CREATE OR REPLACE PACKAGE BODY citi AS
 --function implemented
  FUNCTION p_strng RETURN VARCHAR2 IS
    BEGIN
      RETURN 'Software Testing Help!';
    END p_strng;
END citi;
/
```

Coding implementation for accessing elements:

```
BEGIN
  DBMS_OUTPUT.PUT_LINE (citi.p_strng);
END;
```

9. Write a Pl/ Sql queries to create procedures

```
CREATE OR REPLACE PROCEDURE welcome_msg (p_name IN VARCHAR2)
IS
BEGIN
dbms_output.put_line (‘Welcome '|| p_name);
END;
/
EXEC welcome_msg (‘Welcome to PL SQL Lab’);
```

10. Write a Pl/ Sql queries to create Triggers

Let's look at an example of how to create an BEFORE INSERT trigger using the CREATE TRIGGER statement.

If you had a table created as follows:

```
CREATE TABLE orders
( order_id number(5),
  quantity number(4),
  cost_per_item number(6,2),
  total_cost number(8,2),
  create_date date,
  created_by varchar2(10)
);
```