NAME :- Manish Shashikant Jadhav
UID :- 2023301005.
BRANCH :- Comps -B. **BATCH:** B.
EXPERIMENT 10: Implement Pl/SQL procedures and functions.
SUBJECT :- DBMS (DATABASE MANAGEMENT SYSTEM)

1. **Write the PL/Sql anonymous block to calculate the area of circle for value of radius varying from 3 to 7. Store the radius and corresponding values of calculated areas in a table called Areas and display the table areas.**

```
1  CREATE TABLE Areas (
2  radius INT,
3  area FLOAT(5)
4  );

Table created.
```

SQL Worksheet

```
5  DECLARE
6  r NUMBER;
7  a NUMBER;
8  BEGIN
9  FOR r IN 3..7 LOOP
10 a := 3.14 * r * r;
11 INSERT INTO Areas VALUES (r, a);
12 END LOOP;
13 END;
14 /

Statement processed.
```

```
15  SELECT * FROM Areas;
16
```

RADIUS	AREA
3	28
4	50
5	79
6	110
7	150

2. **Create a table Book(ID, Title, Author). Write a function to insert the values of all attributes in Book table .**

```
1  CREATE TABLE Book (
2  ID INT,
3  Title VARCHAR2(50),
4  Author VARCHAR2(50)
5  );
```

```
Table created.
```

```
6  CREATE OR REPLACE FUNCTION InsertBook(
7  p_ID INT,
8  p_Title VARCHAR2,
9  p_Author VARCHAR2
10 ) RETURN VARCHAR2
11 AS
12 BEGIN
13 INSERT INTO Book VALUES (p_ID, p_Title, p_Author);
14 COMMIT;
15 RETURN 'Book inserted successfully';
16 END InsertBook;
17 /
18
```

```
Function created.
```

```
DECLARE
result VARCHAR2(50);
BEGIN
result := InsertBook(1, 'The Psychology of Monet', 'Morgan Housel');
DBMS_OUTPUT.PUT_LINE(result);
END;
/
```

```
Statement processed.
Book inserted successfully
```

3. Create a function to update the values of Book table

SQL Worksheet

```
CREATE OR REPLACE FUNCTION UpdateBook(
p_ID INT,
p_Title VARCHAR2,
p_Author VARCHAR2
) RETURN VARCHAR2
AS
BEGIN

UPDATE Book
SET Title = p_Title, Author = p_Author
WHERE ID = p_ID;
COMMIT; -- Save the changes
RETURN 'Book updated successfully';
END UpdateBook;
/
```

```
Function created.
```

```
DECLARE
result VARCHAR2(50);
BEGIN
result := (1, 'Ikigai: Japanese Secret for Happy Life', 'Hector Gracia');
DBMS_OUTPUT.PUT_LINE(result);
END;
/
```

```
Statement processed.
Book updated successfully
```

4. Create a function to display the details of Book table:

```
CREATE OR REPLACE FUNCTION DisplayBookDetails RETURN SYS_REFCURSOR
AS
v_cursor SYS_REFCURSOR;
BEGIN

OPEN v_cursor FOR
SELECT * FROM Book;
RETURN v_cursor;
END DisplayBookDetails;A
/

```

```
Function created.
```

```
DECLARE
result_cursor SYS_REFCURSOR;
id Book.ID%TYPE;
title Book.Title%TYPE;
author Book.Author%TYPE;
BEGIN
result_cursor := DisplayBookDetails;
LOOP
FETCH result_cursor INTO id, title, author;
EXIT WHEN result_cursor%NOTFOUND;
DBMS_OUTPUT.PUT_LINE('ID: ' || id || ', Title: ' || title || ',
Author: ' || author);
END LOOP;
CLOSE result_cursor;
END;
/
```

```
Statement processed.
ID: 1, Title: Ikigai: Japanese Secret for Happy Life,
Author: Hector Gracia
ID: 1, Title: Ikigai: Japanese Secret for Happy Life,
Author: Hector Gracia
```

5. **Write a function to calculate the total marks of all students in Subj1 and Subj2. Student1 table contains regn_no, sub1, sub2.**

```
CREATE TABLE Student1 (
Reg_no INT,
Name CHAR(15),
Sub1 INT,
Sub2 INT,
Total INT
);
```

```
Table created.
```

```
INSERT INTO Student1 (Reg_no, Name, Sub1, Sub2, Total)
VALUES (1, 'Student1', 85, 90, NULL);
```

```
1 row(s) inserted.
```

```
CREATE OR REPLACE FUNCTION CalculateTotalMarks RETURN VARCHAR2
AS
BEGIN

UPDATE Student1
SET Total = Sub1 + Sub2;
COMMIT; -- Save the changes
RETURN 'Total marks calculated and updated successfully';
END CalculateTotalMarks;
/
```

```
Function created.
```

```
23  DECLARE
24  result VARCHAR2(50);
25  BEGIN
26  result := CalculateTotalMarks;
27  DBMS_OUTPUT.PUT_LINE(result);
28  END;
29  /


Statement processed.
Total marks calculated and updated successfully
```

```
30
31  SELECT * FROM Student1;
32
```

REG_NO	NAME	SUB1	SUB2	TOTAL
1	Student1	85	90	175

Conclusion: Hence by completing this experiment I came to know about implementing Pl/SQL procedures and functions.