**4. plsql radius area**

Create database plsql;

Use plsql;

CREATE TABLE areas (

radius INT,

area DECIMAL(10, 2)

);

DELIMITER //

CREATE PROCEDURE calculate\_circle\_areas()

BEGIN

DECLARE v\_radius INT DEFAULT 5;

DECLARE v\_area DECIMAL(10, 2);

WHILE v\_radius <= 9 DO

SET v\_area = 3.14159 \* v\_radius \* v\_radius;

INSERT INTO areas (radius, area) VALUES (v\_radius, v\_area);

SET v\_radius = v\_radius + 1;

END WHILE;

END //

CALL calculate\_circle\_areas();

SELECT \* FROM areas;

**Theory**

**1. Area of a Circle**

The formula for the area of a circle is:

Area= pie \* radius^2

Here, π (pi) is approximated as 3.14159.

Using this formula, we can compute the area for any given radius.

**2. Table Creation**

The table areas is created with two columns:

radius: Holds the radius value (as an integer).

area: Holds the calculated area (as a decimal with two decimal places).

SQL Syntax: Since MySQL does not support NUMBER as a data type, we use INT for radius and DECIMAL(10, 2) for area.

**3. PL/SQL Procedure to Calculate Area**

A PL/SQL procedure allows for sequential calculations, where we can compute and store multiple areas in a single run using a loop.

Looping Mechanism: A FOR loop iterates over the radius values from 5 to 9.

Each loop iteration:

Calculates the area for the current radius.

Inserts the radius and area as a new row in the areas table.

Code Walkthrough

Declare Variables:

v\_radius for holding the current radius.

v\_area for holding the computed area.

Looping from 5 to 9:

For each value of v\_radius, the area is calculated using 3.14159 \* v\_radius \* v\_radius.

This area is stored in v\_area.

Insert Statement:

The INSERT INTO areas statement within the loop stores each calculated radius and area into the areas table.

**Example Execution**

When the procedure is executed, it fills the areas table with radius values from 5 to 9 and their corresponding calculated areas, as shown below:

radius area

5 78.54

6 113.10

7 153.94

8 201.06

9 254.47