**DBMS LAB - PL/SQL Procedure, Function, Cursor and Triggers**

1. Write a procedure to declare a character & constant number and do the following:

1. If the character is ‘R’ assuming that the constant number is a radius, calculate the area of the circle, sphere and the volume of the sphere it will form.
2. If the character is ‘L’ assuming that the constant number is the length of the side of a square, calculate the area of the square it will form and the volume of the cube it will form.
3. If the character is not ‘R’ or ‘L’ then print “invalid choice”.
4. After calculation, insert the values into the “Areas” table.

**create or replace procedure area(N in number, CH in char ) as**

Length numeric;

Rad numeric;

Ca numeric;

Sa numeric

Spa numeric;

Spa numeric;

Cv numeric;

Begin

If CH=’R’ then

Rad: =N;

Ca:= 3.14 \* Rad \* Rad;

Spa := 4\*3.14 \*Rad \*Rad;

Spv:= 4/3\*3.14 \* Rad \* Rad \* Rad;

DBMS\_OUTPUT.PUT\_LINE(‘CIRCLE-AREA= =’|| Ca);

DBMS\_OUTPUT.PUT\_LINE(‘SPHERE-AREA= =’||Spa);

DBMS\_OUTPUT.PUT\_LINE(SPHERE-VOLUME = =’||Spv);

INSERT INTO AREA(INPUT,CIRCLE AREA,SPHERE\_AREA,SPHERE\_VOL) VALUES (Rad,Ca,Spa,Spv);

DBMS\_OUTPUT.PUT\_LINE(‘Values Inserted’);

Else

If CH= ‘L’ then

Length := N;

Sa := Length \* Length;

Cv := Length \* Length \* Length;

DBMS\_OUTPUT.PUT\_LINE(‘CIRCLE-AREA= =’|| sa);

DBMS\_OUTPUT.PUT\_LINE(‘SPHERE-AREA= =’||cv);

INSERT INTO AREA(INPUT,SQUARE\_AREA,CUBE\_VOL) VALUES (Length,Sa,Cv);

End if;

End if;

End;

2. **Creates a procedure ‘employer\_details’ which gives the details of the employee.**

CREATE OR REPLACE PROCEDURE employer\_details

IS

CURSOR emp\_cur IS

SELECT first\_name, last\_name, salary FROM emp\_tbl;

emp\_rec emp\_cur%rowtype;

BEGIN

FOR emp\_rec in sales\_cur

LOOP

dbms\_output.put\_line(emp\_cur.first\_name || ' ' ||emp\_cur.last\_name

|| ' ' ||emp\_cur.salary);

END LOOP;

END;

3.**Sales\_Transaction**

|  |  |  |  |
| --- | --- | --- | --- |
| Prod\_Code | Date\_Of\_Sale | Qty | Price |
| P001 | 01-Jan-2007 | 10 | 1000 |
| P002 | 01-Jan-2007 | 10 | 1000 |
| P003 | 01-Jan-2007 | 10 | 1000 |
| P004 | 02-Jan-2007 | 10 | 1000 |
| P001 | 02-Jan-2007 | 10 | 1000 |

**GST\_Calculation**

|  |  |  |
| --- | --- | --- |
| Date\_Of\_Sale | Prod\_Code | GST |
| 01-Jan-2007 | P001 | 10% of the price sold on this date |
| 01-Jan-2007 | P002 | 10% of the price sold on this date |

1. Whenever an insert happens to the Sales\_Transaction table, the GST for the price should be calculated and inserted into the GST\_Calculation table.
2. If the price is updated in the Sales\_Transaction table, the GST in the GST\_Calculation table should be updated accordingly.

3. If a row is deleted in the Sales\_Transaction table, the corresponding row in the

GST\_Calculation table should be deleted as well

create trigger tin

after insert on sales\_transaction

for each row

begin

insert into vat\_calculation (date\_of\_sales,prod\_code,VAT) values(:new.date\_of\_sale,:new.prod\_code,:new.price\*0.1);

end;

create triggers tup

after update on sales\_transaction

for each row

begin

update vat\_calculation set VAT:=new.price\*0.1 where sales\_transaction

prod\_code = VAT\_calculation.prod\_code;

end;

create trigger tdel

after delete on sales\_transaction

for each row

begin

delete from vat\_calculation where prod\_code:=old;

end;