DBMS Lab - 4

Academic year: 2020-2021 **Semester:** Long Sem

Faculty Name: Ms Beebi Naseeba **Date:** 18 / 6 / 2022

Student name: Taran Mamidala **Reg. no.:** 19BCE7346

Lab Exercise -4

Implement the following programs in PL/SQL

1. To create Basic PL/SQL code with exceptions.

```
CREATE TABLE CLASSROOM (
   ID INT NOT NULL,
   NAME VARCHAR (15) NOT NULL,
   AGE INT NOT NULL,
   PRIMARY KEY (ID)
);

INSERT INTO CLASSROOM VALUES (1,'Taran', 20);
INSERT INTO CLASSROOM VALUES (2,'surya', 19);
INSERT INTO CLASSROOM VALUES (5, 'mahesh', 56);

DECLARE
   CLASSROOM_id CLASSROOM.id%type;
   CLASSROOM_name CLASSROOM.name%type;
   CLASSROOM_age CLASSROOM.age%type := 9;
BEGIN
   SELECT  id, name INTO  CLASSROOM_id, CLASSROOM_name
   FROM CLASSROOM
   WHERE age = CLASSROOM_age;
   DBMS_OUTPUT.PUT_LINE ('CLASSROOM id is: '||  CLASSROOM_id);
   DBMS_OUTPUT.PUT_LINE ('CLASSROOM name is: '||  CLASSROOM_name);

EXCEPTION
   WHEN no_data_found THEN
      dbms_output.put_line ('No details found in CLASSROOM');
   WHEN others THEN
      dbms_output.put_line ('Errors!!');
END;
/
```

```
SQL> start with_exceptions.sql

Table created.


1 row created.


1 row created.


1 row created.

No details found in CLASSROOM

PL/SQL procedure successfully completed.
```

with_exceptions - Notepad

File Edit Format View Help

```
CREATE TABLE CLASSROOM (
   ID INT NOT NULL,
   NAME VARCHAR (15) NOT NULL,
   AGE INT NOT NULL,
   PRIMARY KEY (ID)
);

INSERT INTO CLASSROOM VALUES (1,'Taran',
20);
INSERT INTO CLASSROOM VALUES (2,'surya',
19);
INSERT INTO CLASSROOM VALUES (5,
'mahesh', 56);

DECLARE
   CLASSROOM_id CLASSROOM.id%type;
   CLASSROOM_name CLASSROOM.name%type;
   CLASSROOM_age CLASSROOM.age%type :=
9;
```

2. To create PL/SQL code using a control statement IF LOOP.

```
SET
  SERVEROUTPUT ON;
DECLARE emp_salary NUMBER := 500000;
Increment NUMBER(20, 3) := 0;
BEGIN IF emp_salary > 300000 THEN Increment := emp_salary * 0.2;
dbms_output.put_line(
  'Emp salary is incremented by 0.2 percentage'
);
ELSIF emp_salary <= 300000
AND emp_salary > 200000 THEN Increment := emp_salary * 0.04;
dbms_output.put_line(
  'Emp salary is incremented by 0.04 percentage'
);
ELSIF emp_salary <= 2100000
AND emp_salary > 40000 THEN Increment := emp_salary * 0.09;
dbms_output.put_line(
  'Emp salary is incremented by 0.09 percentage'
);
ELSE Increment := emp_salary * 0.02;
dbms_output.put_line(
  'Emp salary is incremented by 0.02 percentage'
);
END IF;
END;
/
```

3. To create PL/SQL code using a control statement WHILE LOOP.

```
CREATE OR REPLACE FUNCTION factorial(num integer) RETURNS integer AS $$
DECLARE
factorial integer = 1; i integer = 1; BEGIN
WHILE(i <= num) LOOP
factorial = factorial*i; i = i+1;
END LOOP;
RETURN factorial; END;
$$ LANGUAGE plpgsql;
```

```
SELECT public.factorial( 6 );
```

4. To create PL/SQL code using a control statement FOR LOOP.

```
DECLARE
   a number(2);
BEGIN
   FOR a in 10 .. 20 LOOP
      dbms_output.put_line('value of a: ' || a);
  END LOOP;
END;
/
```

5. To create a cursor and work on that.

Scenario: Retrieve the name, address and overall percentage of students in a table ‘student’ having columns ‘name’, ‘address’ and ‘percentage’

```
DECLARE stud_name student.name % type;
stud_address student.address % type;
stud_percentage student.percentage % type;
CURSOR stud IS
SELECT
  name,
  address,
  percentage
FROM
  student;
BEGIN Open stud;
LOOP FETCH stud into stud_name,
stud_address,
stud_percentage;
```

```
EXIT when stud % NOTFOUND;
dbms_ouput.put_line(
  stud_name || ' ' || stud_address || ' ' || stud_percentage
);
END LOOP;
CLOSE stud;
END /
```

7. To create a Function to find Factorial of a number.

```
Create function factor(@number int)
returns int
as begin
Declare @i int = 1,@result int=1
while (@i<=@number)
Begin
    Set @result = @result * @i
    Set @i += 1
End
return @result
End
```

8. To create a Producer to find Fibonacci of a number.

```
declare
first number:=0;
second number:=1;
third number;
n number:=&n;
i number;

begin
dbms_output.put_line('Fibonacci series is:');
dbms_output.put_line(first);
dbms_output.put_line(second);

for i in 2..n
loop
third:=first+second;
first:=second;
second:=third;
dbms_output.put_line(third);
```

```
end loop;
end;
/
```

9. To create Triggers on any table and work on that.

```
CREATE TABLE Employee
(
  Id int Primary Key,
  Name nvarchar(30),
  Salary int,
  Gender nvarchar(10),
  DepartmentId int
)
GO
-- Insert data into Employee table
INSERT INTO Employee VALUES (1,'Pranaya', 5000, 'Male', 3)
INSERT INTO Employee VALUES (2,'Priyanka', 5400, 'Female', 2)
INSERT INTO Employee VALUES (3,'Anurag', 6500, 'male', 1)
INSERT INTO Employee VALUES (4,'sambit', 4700, 'Male', 2)
INSERT INTO Employee VALUES (5,'Hina', 6600, 'Female', 3)

CREATE TRIGGER trInsertEmployee
ON Employee
FOR INSERT
AS
BEGIN
  PRINT 'YOU CANNOT PERFORM INSERT OPERATION'
  ROLLBACK TRANSACTION
END
```

```
INSERT INTO Employee VALUES (6, 'Saroj', 7600, 'Male', 1)
```