Expno12:

SQL> set serveroutput on;

SQL> create or replace function factorial(n number) return number is fact number := 1;

2 begin

3 if n < 0 then

4 return null;

5 elsif n = 0 then

6 return 1;

7 else

8 for i in 1..n loop

9 fact := fact \* i;

10 end loop;

11 return fact;

12 end if;

13 end;

14 /

Function created.

SQL> select factorial(5) from dual;

FACTORIAL(5)

------------

120

SQL> create or replace function area\_of\_square(a number) return number is area number:=0;

2 begin

3 area:=a\*a;

4 return area;

5 end;

6 /

Function created.

SQL> select area\_of\_square(5) from dual;

AREA\_OF\_SQUARE(5)

-----------------

25

SQL> create or replace function pow(base number,exponent number) return number is

2 result number:=1;

3 begin

4 for i in 1..exponent loop

5 result:=result\*base;

6 end loop;

7 return result;

8 end;

9 /

Function created.

SQL> select pow(2,8) from dual;

POW(2,8)

----------

256

SQL> create or replace function oddeven(n number) return varchar is

2 begin

3 if mod(n,2)=0 then

4 return 'even';

5 else

6 return 'odd';

7 end if;

8 end;

9 /

Function created.

SQL> select oddeven(8) from dual;

ODDEVEN(8)

--------------------------------------------------------------------------------

even

SQL> select oddeven(11) from dual;

ODDEVEN(11)

--------------------------------------------------------------------------------

odd

SQL> create or replace procedure multi\_table(n in number, m in number) is

2 result number;

3 begin

4 for i in n..m loop

5 result := n \* i;

6 dbms\_output.put\_line(n || ' \* ' || i || ' = ' || result);

7 end loop;

8 end;

9 /

Procedure created.

SQL> begin

2 multi\_table(1,5);

3 end;

4 /

1 \* 1 = 1

1 \* 2 = 2

1 \* 3 = 3

1 \* 4 = 4

1 \* 5 = 5

PL/SQL procedure successfully completed.

SQL> select \* from customers;

CID ADDRESS SALARY AGE NAME

---------- ---------- ---------- ---------- ----------

1 erode 12000 19 Dinesh

2 dindigul 14500 18 Babu

3 Madurai 15500 20 kumar

SQL> create or replace function maxsalary return number is

2 maximum\_salary number;

3 begin

4 select max(salary) into maximum\_salary from customers;

5 return maximum\_salary;

6 end;

7 /

Function created.

SQL> select maxsalary() as maximum\_salary from dual;

MAXIMUM\_SALARY

--------------

15500

SQL> create or replace procedure max\_and\_min\_salary is

2 max\_salary number;

3 min\_salary number;

4 begin

5 select max(salary), min(salary) into max\_salary, min\_salary from customers;

6 dbms\_output.put\_line('Maximum Salary: ' || max\_salary);

7 dbms\_output.put\_line('Minimum Salary: ' || min\_salary);

8 end;

9 /

Procedure created.

SQL> begin

2 max\_and\_min\_salary;

3 end;

4 /

Maximum Salary: 15500

Minimum Salary: 12000

PL/SQL procedure successfully completed.

SQL> select \* from customers;

CID ADDRESS SALARY AGE NAME

---------- ---------- ---------- ---------- ----------

1 erode 12000 19 Dinesh

2 dindigul 14500 18 Babu

3 Madurai 15500 20 kumar

SQL> create or replace procedure increment\_price as

2 begin

3 update customers

4 set salary=salary+500

5 end;

6 /

Warning: Procedure created with compilation errors.

SQL> create or replace procedure increment\_price as

2 begin

3 update customers set salary=salary+500;

4 end;

5 /

Procedure created.

SQL> execute increment\_price;

PL/SQL procedure successfully completed.

SQL> select \* from customers;

CID ADDRESS SALARY AGE NAME

---------- ---------- ---------- ---------- ----------

1 erode 12500 19 Dinesh

2 dindigul 15000 18 Babu

3 Madurai 16000 20 kumar

SQL> create or replace function sumofsalary return number is

2 total\_salary number:=0;

3 begin

4 select sum(salary) into total\_salary from customers

5 where age>25;

6

7 return total\_salary;

8 end;

9 /

Function created.

SQL> create or replace function sumofsalary return number is

2 total\_salary number:=0;

3 begin

4 select sum(salary) into total\_salary from customers

5 where age>25;

6 /

Warning: Function created with compilation errors.

SQL> create or replace function sumofsalary return number is

2 total\_salary number:=0;

3 begin

4 select sum(salary) into total\_salary from customers

5 where age>18;

6 return total\_salary;

7 end;

8 /

Function created.

SQL> select sumofsalary as total\_salary from dual;

TOTAL\_SALARY

------------

28500

SQL> select \* from customers;

CID ADDRESS SALARY AGE NAME

---------- ---------- ---------- ---------- ----------

1 erode 12500 19 Dinesh

2 dindigul 15000 18 Babu

3 Madurai 16000 20 kumar

SQL> create or replace function increment\_salary(p\_salary number) return number is

2 begin

3 return p\_salary + 1000;

4 end;

5 /

Function created.