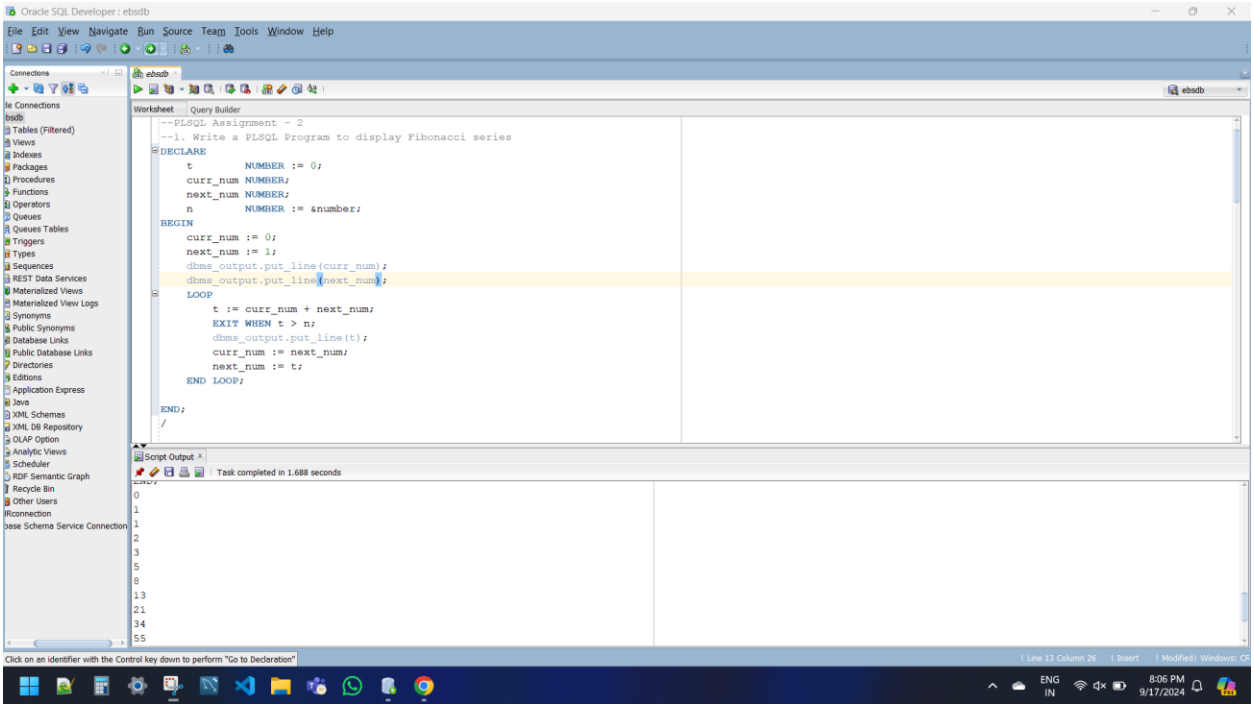
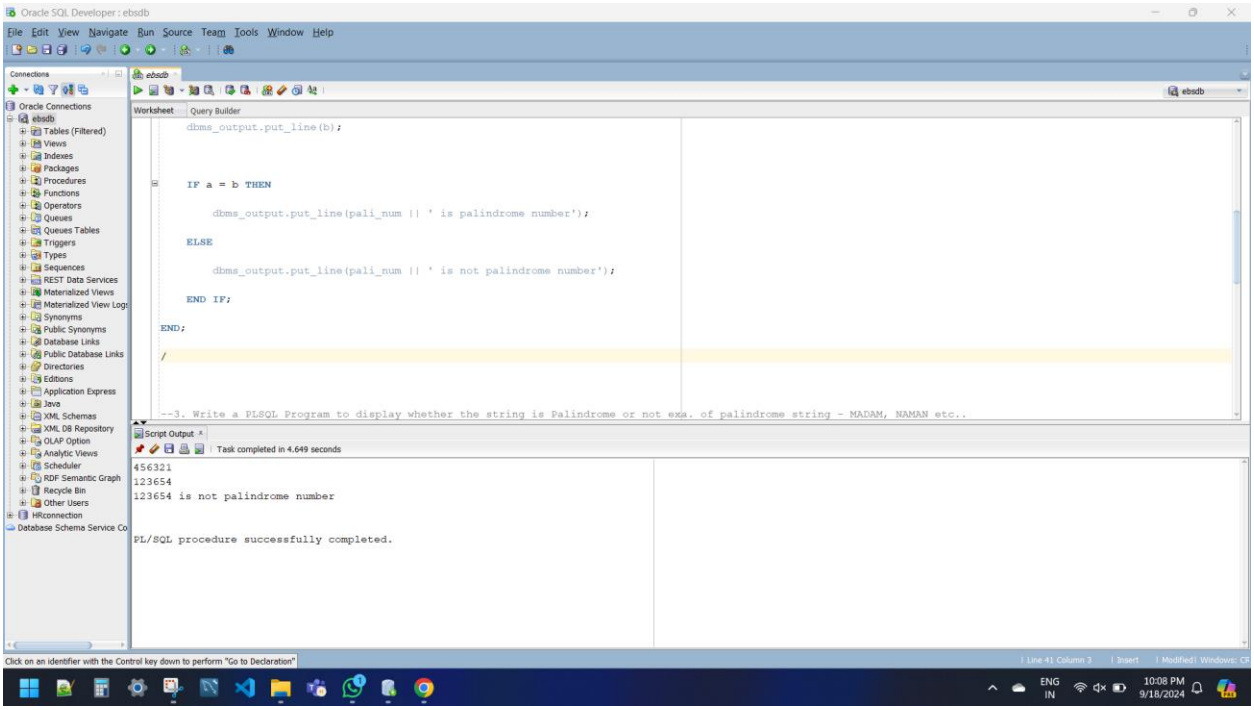
```
set serveroutput on;
--PLSQL Assignment - 2
--1. Write a PLSQL Program to display Fibonacci series
DECLARE
   t      NUMBER := 0;
   curr_num NUMBER;
   next_num NUMBER;
   n       NUMBER := &number;
BEGIN
   curr_num := 0;
   next_num := 1;
   dbms_output.put_line(curr_num);
   dbms_output.put_line(next_num);
   LOOP
      t := curr_num + next_num;
      EXIT WHEN t > n;
      dbms_output.put_line(t);
      curr_num := next_num;
      next_num := t;
   END LOOP;

END;
/
```



--2. Write a PLSQL Program to display whether the number is Palindrome or not exa. of palindrome number - 12321 , 56765 etc..

```
DECLARE
   pali_num NUMBER := '&number';
   a      VARCHAR2(20);
   b      VARCHAR2(20);
BEGIN
   FOR i IN 1..length(pali_num) LOOP
      a := concat(substr(pali_num, i, 1),a);
      b := concat(substr(pali_num, -i, 1),b);
   END LOOP;


   dbms_output.put_line(a);
   dbms_output.put_line(b);
```

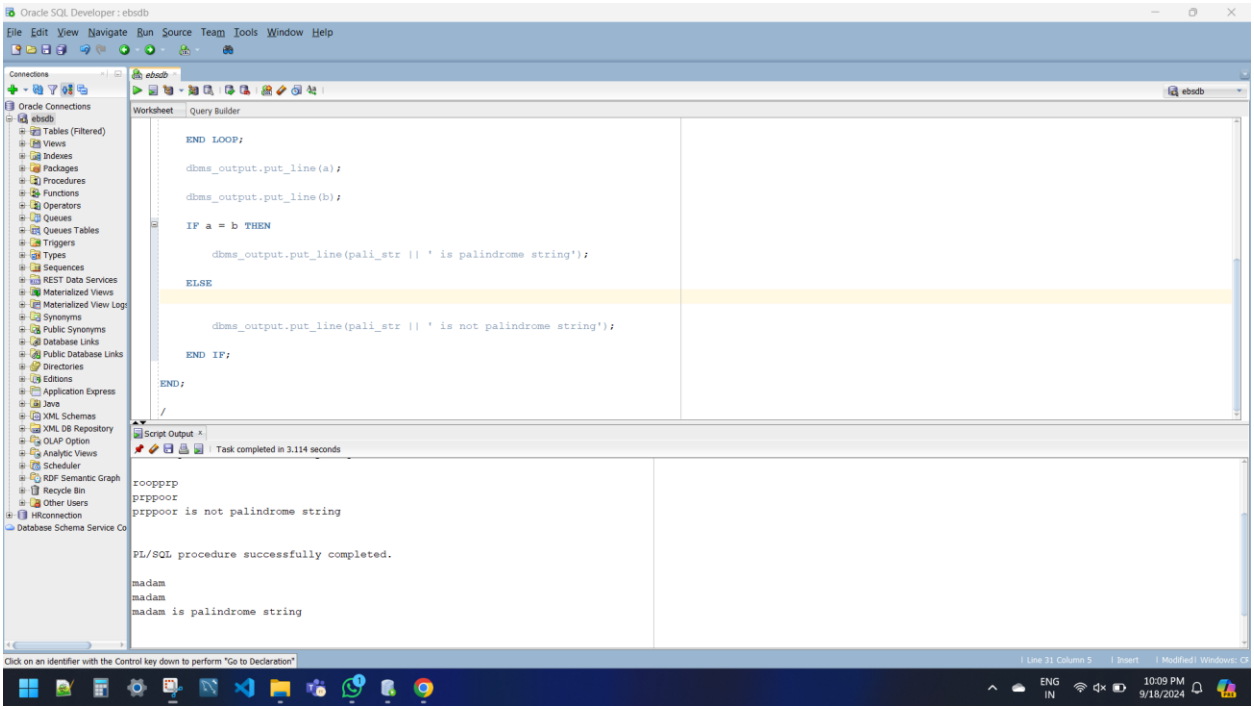
```
  IF a = b THEN
    dbms_output.put_line(pali_num || ' is palindrome number');
  ELSE
    dbms_output.put_line(pali_num || ' is not palindrome number');
  END IF;
END;
/
```



--3. Write a PLSQL Program to display whether the string is Palindrome or not exa. of palindrome string - MADAM, NAMAN etc..

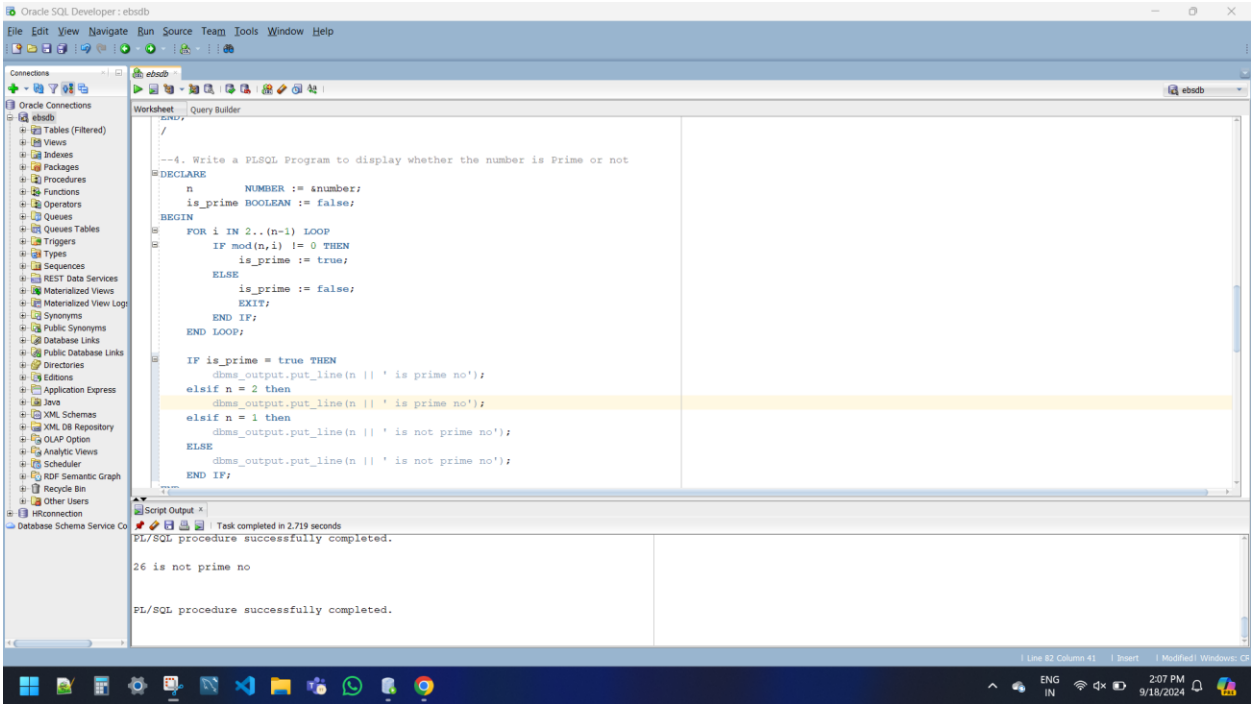
```
DECLARE
  pali_str varchar2(20) := '&string';
  a      VARCHAR2(20);
  b      VARCHAR2(20);
BEGIN
```

```
    FOR i IN 1..length(pali_str) LOOP
        a := concat(substr(pali_str, i, 1),a);
        b := concat(substr(pali_str, -i, 1),b);
    END LOOP;
    dbms_output.put_line(a);
    dbms_output.put_line(b);
    IF a = b THEN
        dbms_output.put_line(pali_str || ' is palindrome string');
    ELSE
        dbms_output.put_line(pali_str || ' is not palindrome string');
    END IF;
END;
/
```



--4. Write a PLSQL Program to display whether the number is Prime or not

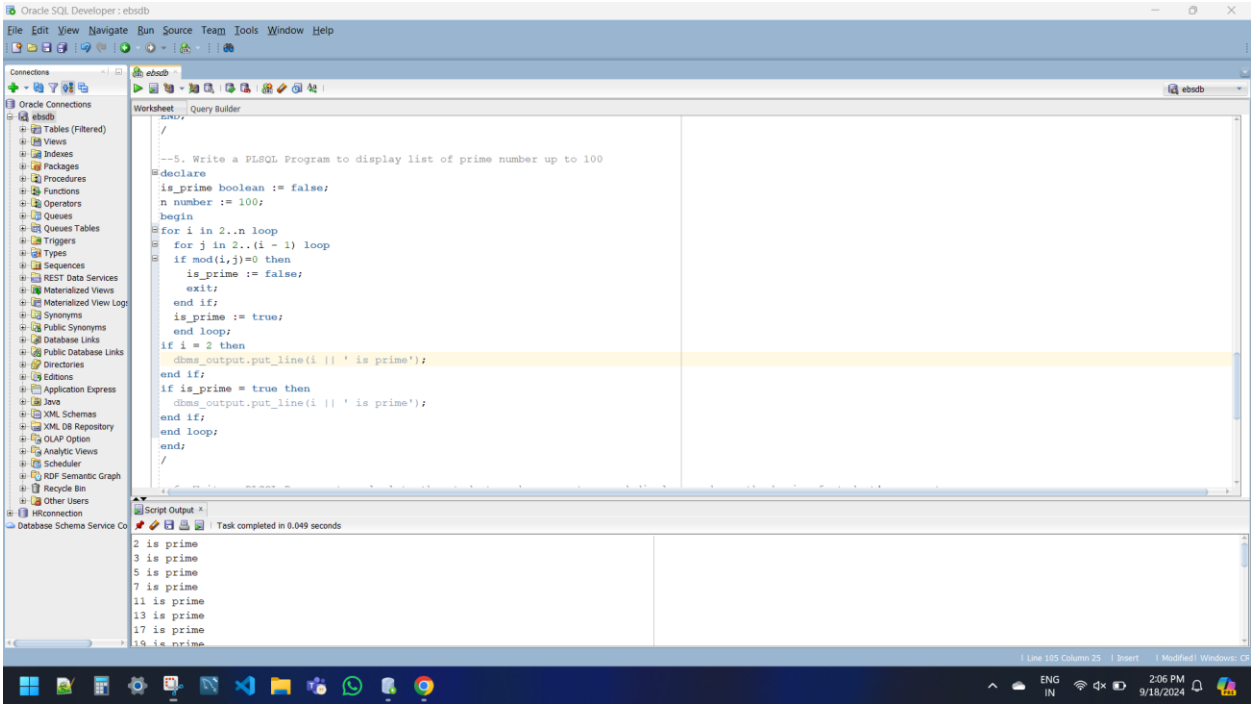
```
DECLARE
   n       NUMBER := &number;
   is_prime BOOLEAN := false;
BEGIN
   FOR i IN 2..(n-1) LOOP
      IF mod(n,i) != 0 THEN
         is_prime := true;
      ELSE
         is_prime := false;
         EXIT;
      END IF;
   END LOOP;

   IF is_prime = true THEN
      dbms_output.put_line(n || ' is prime no');
   elsif n = 2 then
      dbms_output.put_line(n || ' is prime no');
   elsif n = 1 then
      dbms_output.put_line(n || ' is not prime no');
   ELSE
      dbms_output.put_line(n || ' is not prime no');
   END IF;
END;
/
```



--5. Write a PLSQL Program to display list of prime number up to 100

```
declare
is_prime boolean := false;
n number := 100;
begin
for i in 2..n loop
 for j in 2..(i - 1) loop
 if mod(i,j)=0 then
  is_prime := false;
  exit;
 end if;
 is_prime := true;
 end loop;
if i = 2 then
```

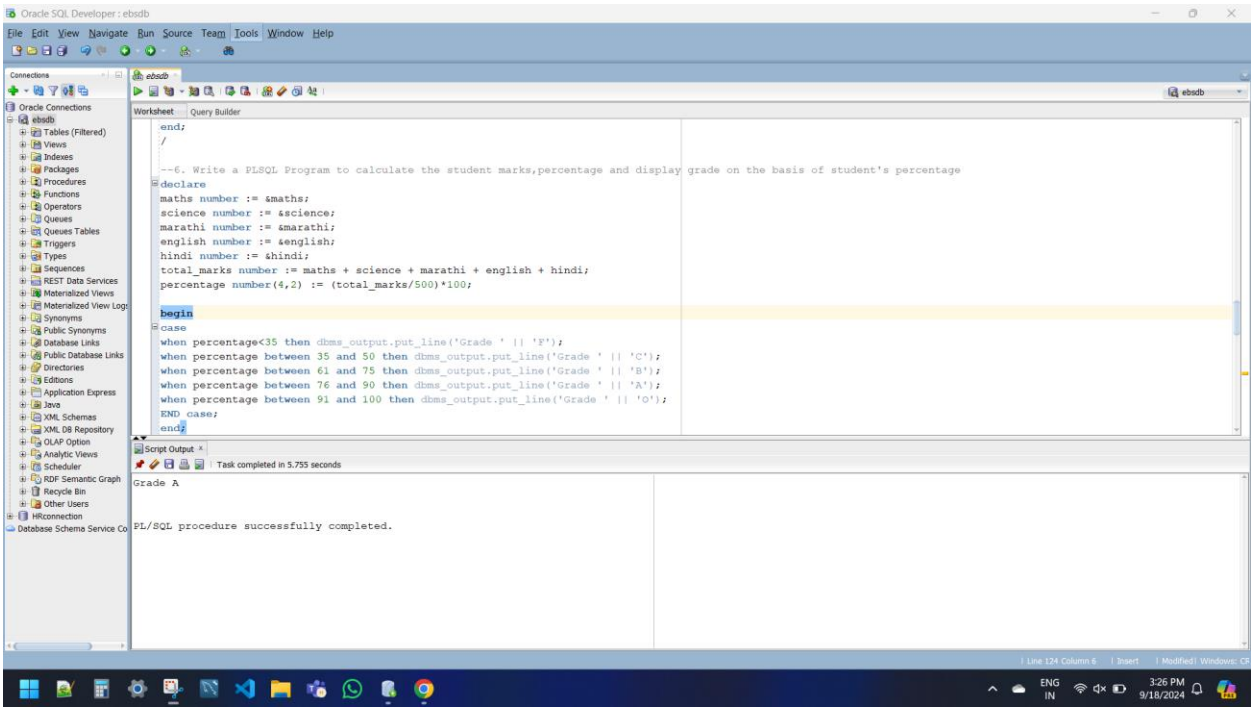
```
 dbms_output.put_line(i || ' is prime');
end if;
if is_prime = true then
 dbms_output.put_line(i || ' is prime');
end if;
end loop;
end;
/
```



--6. Write a PLSQL Program to calculate the student marks,percentage and display grade on the basis of student's percentage

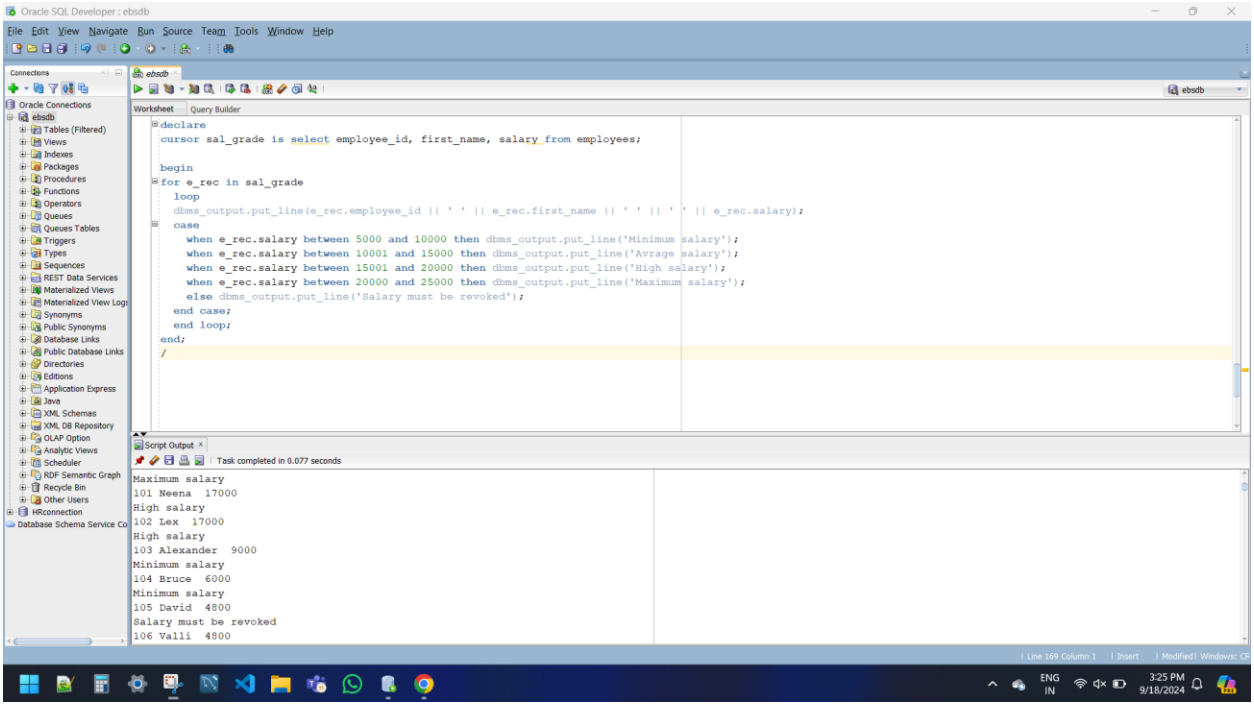
```
declare
maths number := &maths;
science number := &science;
marathi number := &marathi;
```

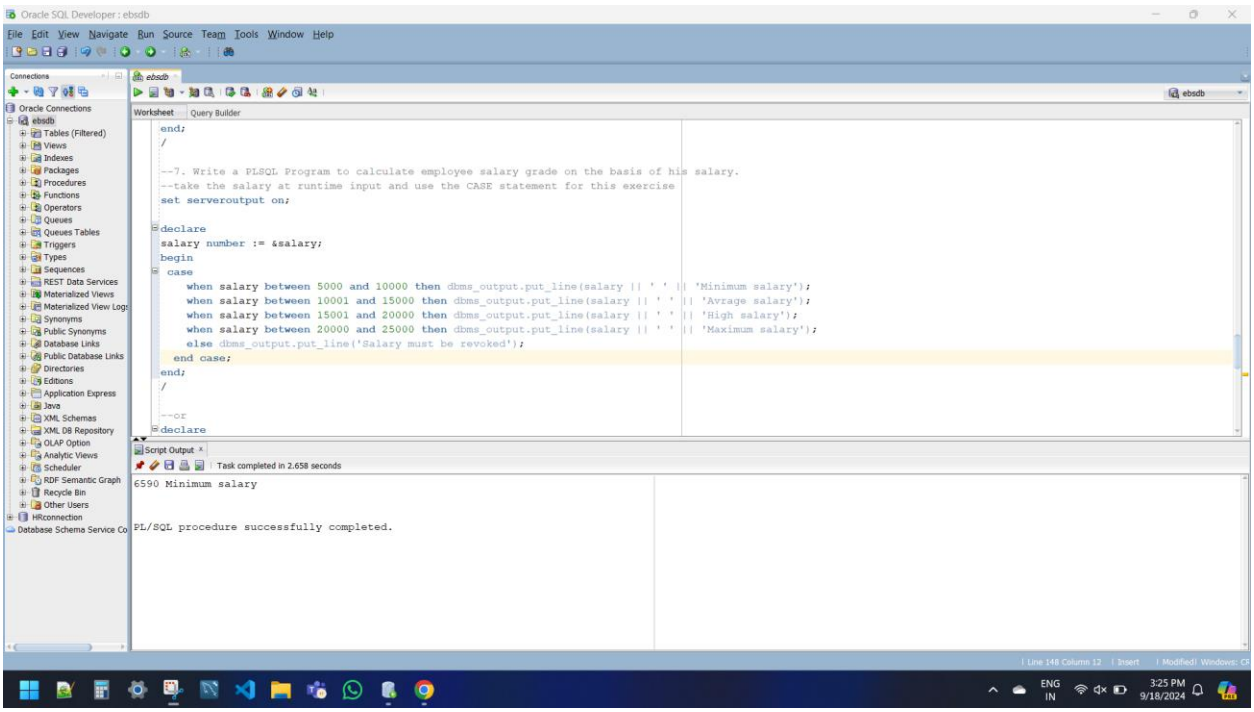
```
english number := &english;
hindi number := &hindi;
total_marks number := maths + science + marathi + english + hindi;
percentage number(4,2) := (total_marks/500)*100;

begin
case
when percentage<35 then dbms_output.put_line('Grade ' || 'F');
when percentage between 35 and 50 then dbms_output.put_line('Grade ' || 'C');
when percentage between 61 and 75 then dbms_output.put_line('Grade ' || 'B');
when percentage between 76 and 90 then dbms_output.put_line('Grade ' || 'A');
when percentage between 91 and 100 then dbms_output.put_line('Grade ' || 'O');
END case;
end;
/
```



```
--7. Write a PLSQL Program to calculate employee salary grade on the basis of his salary.

--take the salary at runtime input and use the CASE statement for this exercise

set serveroutput on;


declare

salary number := &salary;

begin

case

  when salary between 5000 and 10000 then dbms_output.put_line(salary || ' ' || 'Minimum salary');

  when salary between 10001 and 15000 then dbms_output.put_line(salary || ' ' || 'Avrage salary');

  when salary between 15001 and 20000 then dbms_output.put_line(salary || ' ' || 'High salary');

  when salary between 20000 and 25000 then dbms_output.put_line(salary || ' ' || 'Maximum salary');

  else dbms_output.put_line('Salary must be revoked');

 end case;

end;

/


--or

declare

cursor sal_grade is select employee_id, first_name, salary from employees;


begin
```

```
for e_rec in sal_grade
 loop
 dbms_output.put_line(e_rec.employee_id || ' ' || e_rec.first_name || ' ' || ' ' || e_rec.salary);
 case
  when e_rec.salary between 5000 and 10000 then dbms_output.put_line('Minimum salary');
  when e_rec.salary between 10001 and 15000 then dbms_output.put_line('Avrage salary');
  when e_rec.salary between 15001 and 20000 then dbms_output.put_line('High salary');
  when e_rec.salary between 20000 and 25000 then dbms_output.put_line('Maximum salary');
  else dbms_output.put_line('Salary must be revoked');
 end case;
 end loop;
end;
/
```


Oracle SQL Developer : ebsdb
File Edit View Navigate Run Source Team Tools Window Help
declare
cursor sal_grade is select employee_id, first_name, salary from employees;
begin
for e_rec in sal_grade
loop
dbms_output.put_line(e_rec.employee_id || ' ' || e_rec.first_name || ' ' || ' ' || e_rec.salary);
case
when e_rec.salary between 5000 and 10000 then dbms_output.put_line('Minimum salary');
when e_rec.salary between 10001 and 15000 then dbms_output.put_line('Avrage salary');
when e_rec.salary between 15001 and 20000 then dbms_output.put_line('High salary');
when e_rec.salary between 20000 and 25000 then dbms_output.put_line('Maximum salary');
else dbms_output.put_line('Salary must be revoked');
end case;
end loop;
end;
/
Task completed in 0.077 seconds
Maximum salary
101 Neena 17000
High salary
102 Lex 17000
High salary
103 Alexander 9000
Minimum salary
104 Bruce 6000
Minimum salary
105 David 4800
Salary must be revoked
106 Valli 4800


Oracle SQL Developer : ebsdb
File Edit View Navigate Run Source Team Tools Window Help
end;
/
--7. Write a PLSQL Program to calculate employee salary grade on the basis of his salary.
--take the salary at runtime input and use the CASE statement for this exercise
set serveroutput on;
declare
salary number := &salary;
begin
case
when salary between 5000 and 10000 then dbms_output.put_line(salary || ' ' || 'Minimum salary');
when salary between 10001 and 15000 then dbms_output.put_line(salary || ' ' || 'Avrage salary');
when salary between 15001 and 20000 then dbms_output.put_line(salary || ' ' || 'High salary');
when salary between 20000 and 25000 then dbms_output.put_line(salary || ' ' || 'Maximum salary');
else dbms_output.put_line('Salary must be revoked');
end case;
end;
/
--or
declare
Task completed in 2.658 seconds
6590 Minimum salary
PL/SQL procedure successfully completed.