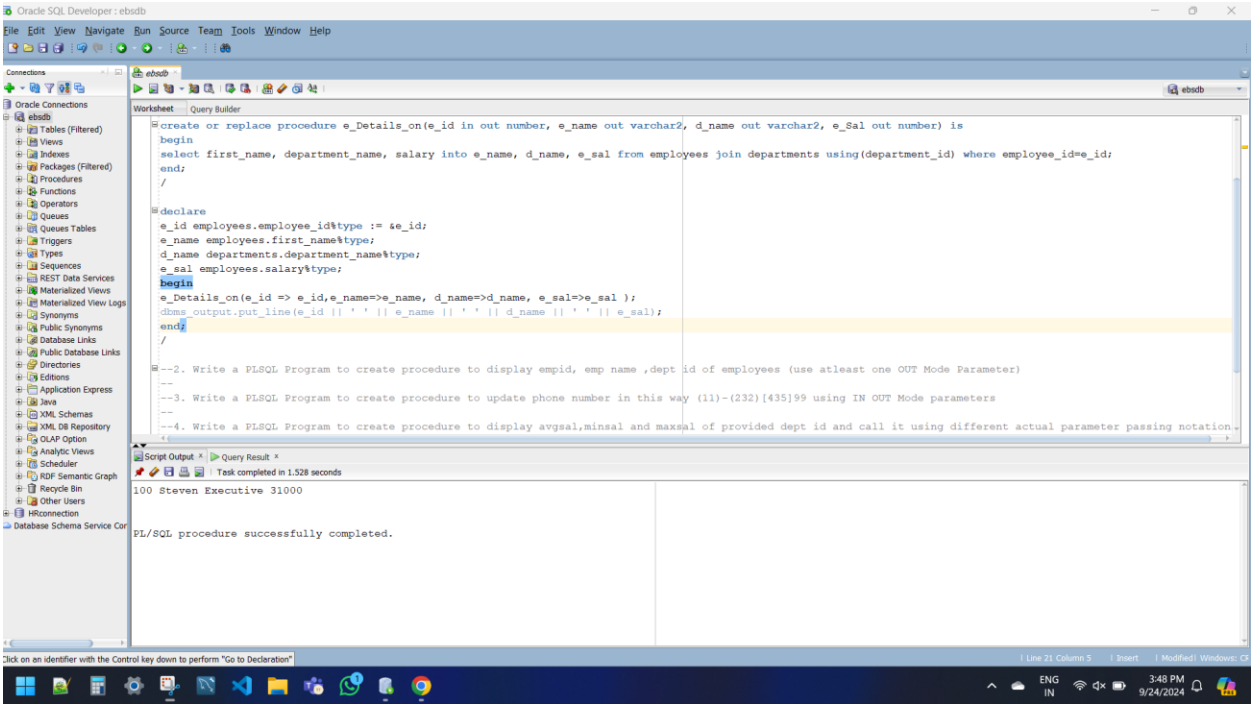
---------- Procedures ----------

```
set serveroutput on;

--1. Write a PLSQL Program to create a procedure which will return Emp id , Emp
name ,Dept name ,Salary of an employee.

select * from employees;

set verify off;


create or replace procedure e_Details_on(e_id in out number, e_name out varchar2,
d_name out varchar2, e_Sal out number) is

begin

select first_name, department_name, salary into e_name, d_name, e_sal from employees
join departments using(department_id) where employee_id=e_id;

end;

/


declare

e_id employees.employee_id%type := &e_id;

e_name employees.first_name%type;

d_name departments.department_name%type;

e_sal employees.salary%type;

begin

e_Details_on(e_id => e_id,e_name=>e_name, d_name=>d_name, e_sal=>e_sal );

dbms_output.put_line(e_id || ' ' || e_name || ' ' || d_name || ' ' || e_sal);

end;

/
```



--2. Write a PLSQL Program to create procedure to display empid, emp name ,dept id of employees (use atleast one OUT Mode Parameter)

create or replace procedure emp_Details_on2(e_id in out number, e_name out varchar2, d_id out number) is

begin

select first_name, department_id into e_name, d_id from employees join departments using(department_id) where employee_id=e_id;

end;

/

declare

e_id employees.employee_id%type := &e_id;

e_name employees.first_name%type;

d_id departments.department_id%type;

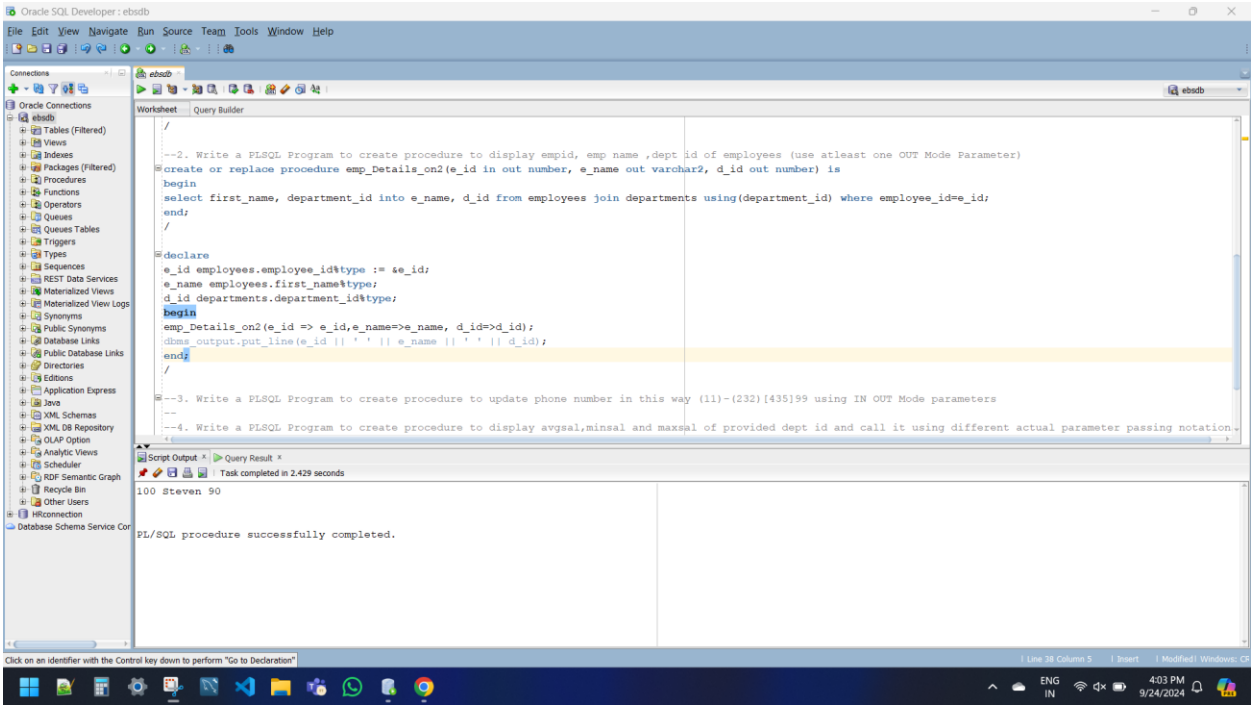
begin

emp_Details_on2(e_id => e_id,e_name=>e_name, d_id=>d_id);

dbms_output.put_line(e_id || ' ' || e_name || ' ' || d_id);

end;

/



--3. Write a PLSQL Program to create procedure to update phone number in this way (11)-(232)[435]99 using IN OUT Mode parameters

create or replace procedure convert_phone_num(p_no in out varchar2) is

begin

p_no := to_Char(p_no);

p_no := '(' || substr(p_no,1,2) || ')-(' || substr(p_no,3,3)|| ')[' || substr(p_no,6,3) || ']' || substr(p_no,9,2);

end;

/

declare

p_no varchar2(100) := '&e_id';

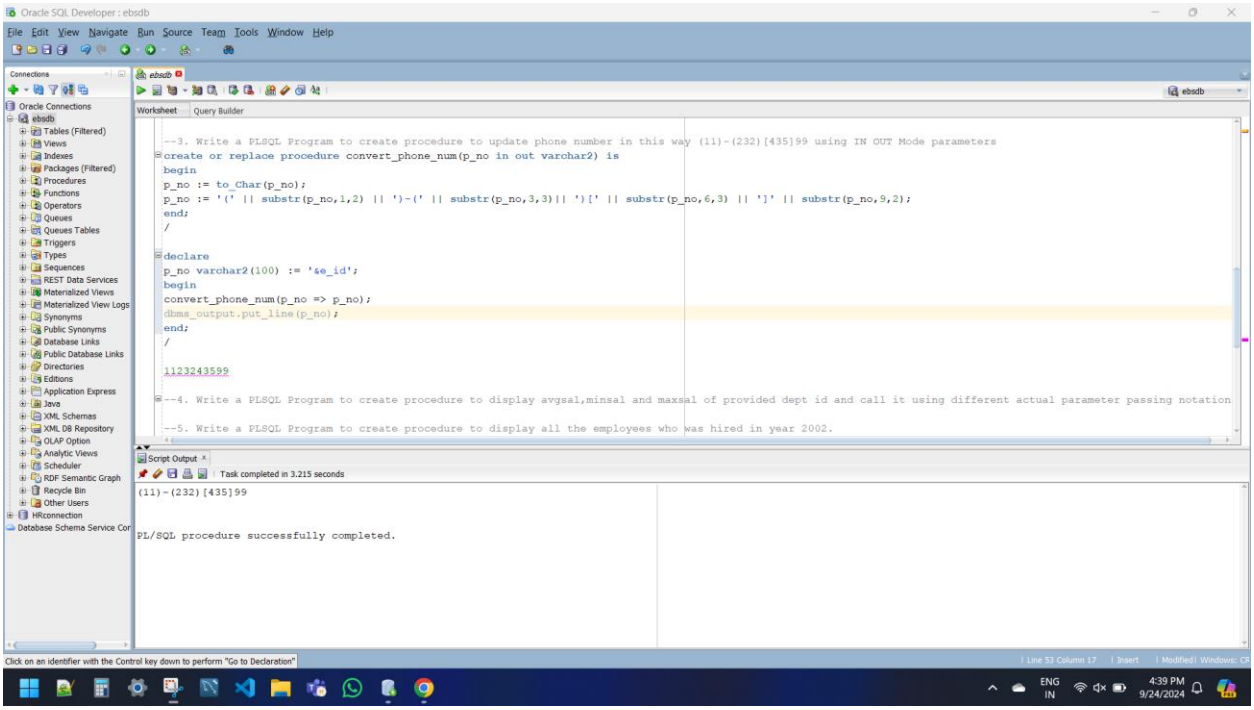
begin

convert_phone_num(p_no => p_no);

dbms_output.put_line(p_no);

end;

/



--4. Write a PLSQL Program to create procedure to display avgsal,minsal and maxsal of provided dept id

--and call it using different actual parameter passing notations.

select * from employees;

create or replace procedure avg_min_max_Sal(d_id in number, avg_sal out number, min_Sal out number, max_sal out number) is

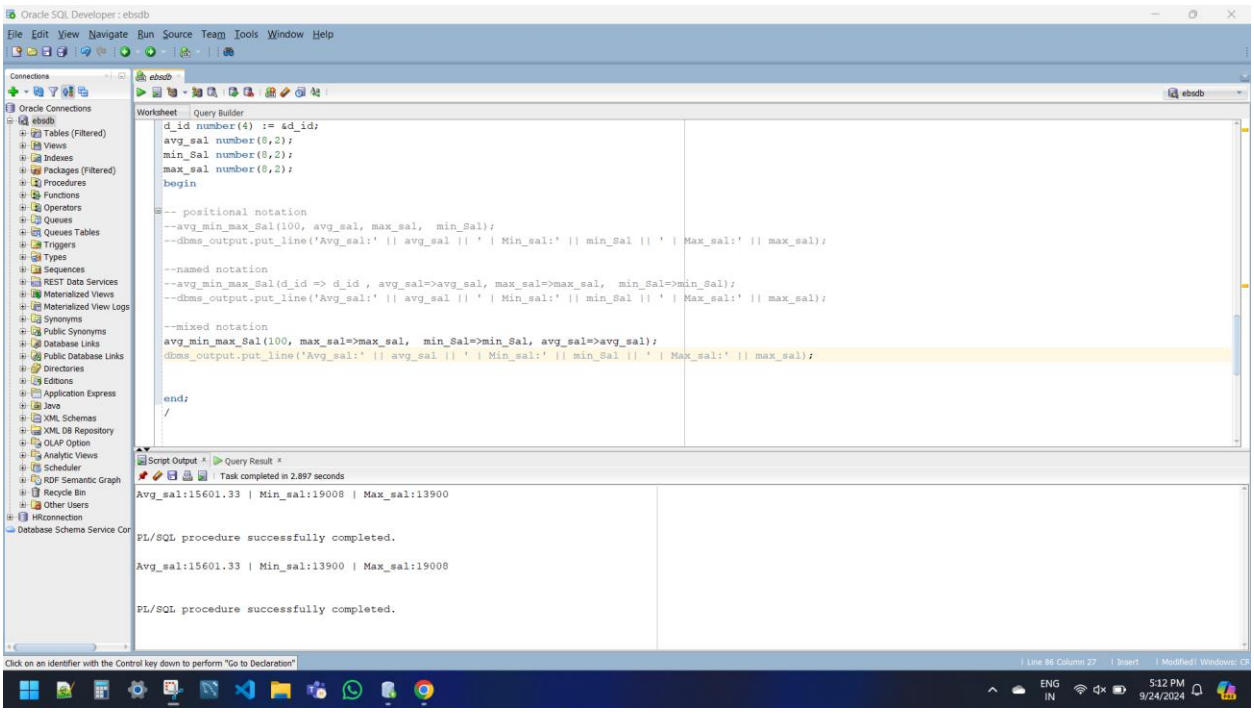
begin

```
select min(salary),max(salary),AVG(salary) into min_Sal,max_sal,avg_sal from employees
join departments

using(department_id) group by department_id having department_id = d_id;

end;

/


declare

result_sal number;

d_id number(4) := &d_id;

avg_sal number(8,2);

min_Sal number(8,2);

max_sal number(8,2);

begin


-- positional notation

--avg_min_max_Sal(100, avg_sal, max_sal,  min_Sal);

--dbms_output.put_line('Avg_sal:' || avg_sal || ' | Min_sal:' || min_Sal || ' | Max_sal:' ||
max_sal);


--named notation

--avg_min_max_Sal(d_id => d_id , avg_sal=>avg_sal, max_sal=>max_sal,
min_Sal=>min_Sal);

--dbms_output.put_line('Avg_sal:' || avg_sal || ' | Min_sal:' || min_Sal || ' | Max_sal:' ||
max_sal);


--mixed notation

avg_min_max_Sal(100, max_sal=>max_sal,  min_Sal=>min_Sal, avg_sal=>avg_sal);
```

dbms_output.put_line('Avg_sal:' || avg_sal || ' | Min_sal:' || min_Sal || ' | Max_sal:' || max_sal);

end;

/



--5. Write a PLSQL Program to create procedure to display all the employees who was hired in year 2002.

select * from employees;

create or replace procedure emp_hire_Date(h_year in varchar2) is

cursor e_details is select * from employees where to_char(hire_date,'YYYY') = h_year;

e_rec employees%rowtype;

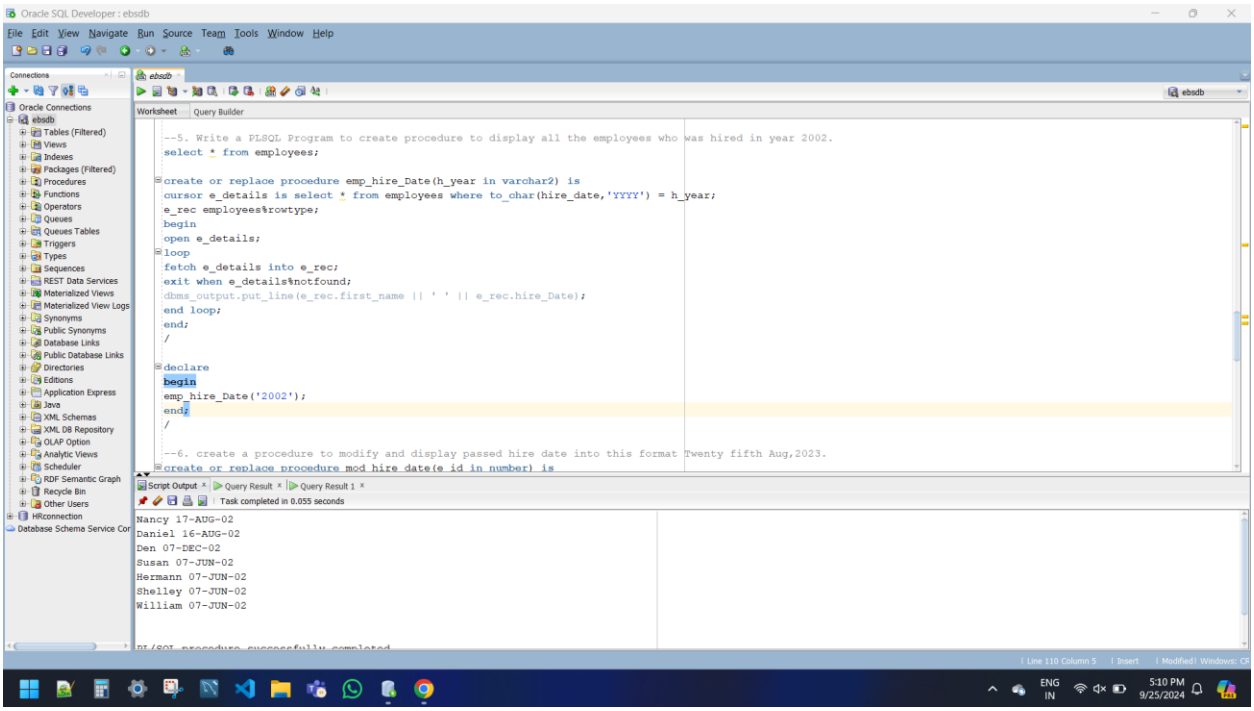
begin

open e_details;

loop

fetch e_details into e_rec;

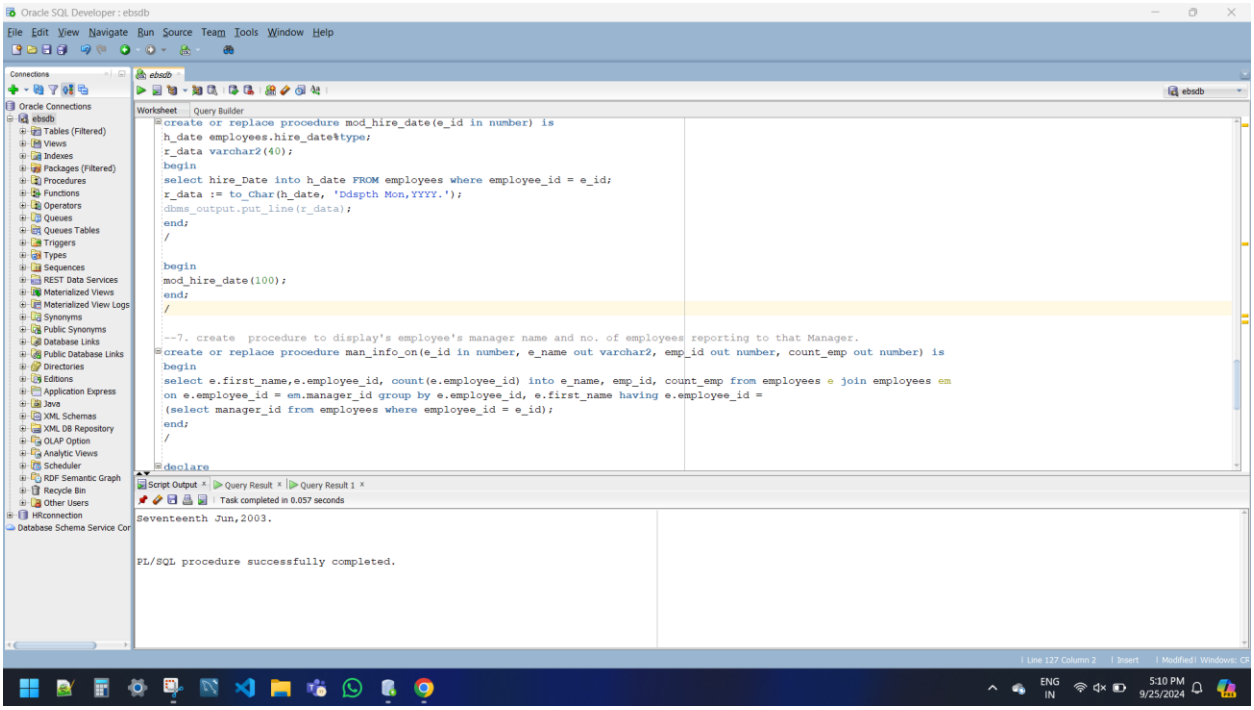
```
exit when e_details%notfound;
dbms_output.put_line(e_rec.first_name || ' ' || e_rec.hire_Date);
end loop;
end;
/

declare
begin
emp_hire_Date('2002');
end;
/
```



--6. create a procedure to modify and display passed hire date into this format Twenty fifth Aug,2023.

```
create or replace procedure mod_hire_date(e_id in number) is
```

```
h_date employees.hire_date%type;
r_data varchar2(40);
begin
select hire_Date into h_date FROM employees where employee_id = e_id;
r_data := to_Char(h_date, 'Ddspth Mon,YYYY.');
dbms_output.put_line(r_data);
end;
/

begin
mod_hire_date(100);
end;
/
```



--7. create procedure to display's employee's manager name and no. of employees reporting to that Manager.

```
create or replace procedure man_info_on(e_id in number, e_name out varchar2, emp_id out number, count_emp out number) is

begin

select e.first_name,e.employee_id, count(e.employee_id) into e_name, emp_id, count_emp from employees e join employees em

on e.employee_id = em.manager_id group by e.employee_id, e.first_name having e.employee_id =

(select manager_id from employees where employee_id = e_id);

end;

/


declare

e_id employees.employee_id%type := &emp_id;

e_name employees.first_name%type;

emp_id employees.employee_id%type;

count_emp number(10);

begin

man_info_on(e_id => e_id, e_name=>e_name, emp_id=>emp_id, count_emp=>count_emp );

dbms_output.put_line('Manager name : ' || e_name );

dbms_output.put_line('Manager Id : ' || emp_id );

dbms_output.put_line('No of emp reporting : ' || count_emp );


end;

/
```

select e.first_name,e.employee_id, count(e.employee_id) from employees e join employees em on e.employee_id = em.manager_id group by

e.employee_id, e.first_name having e.employee_id = (select manager_id from employees where employee_id = &e_id);