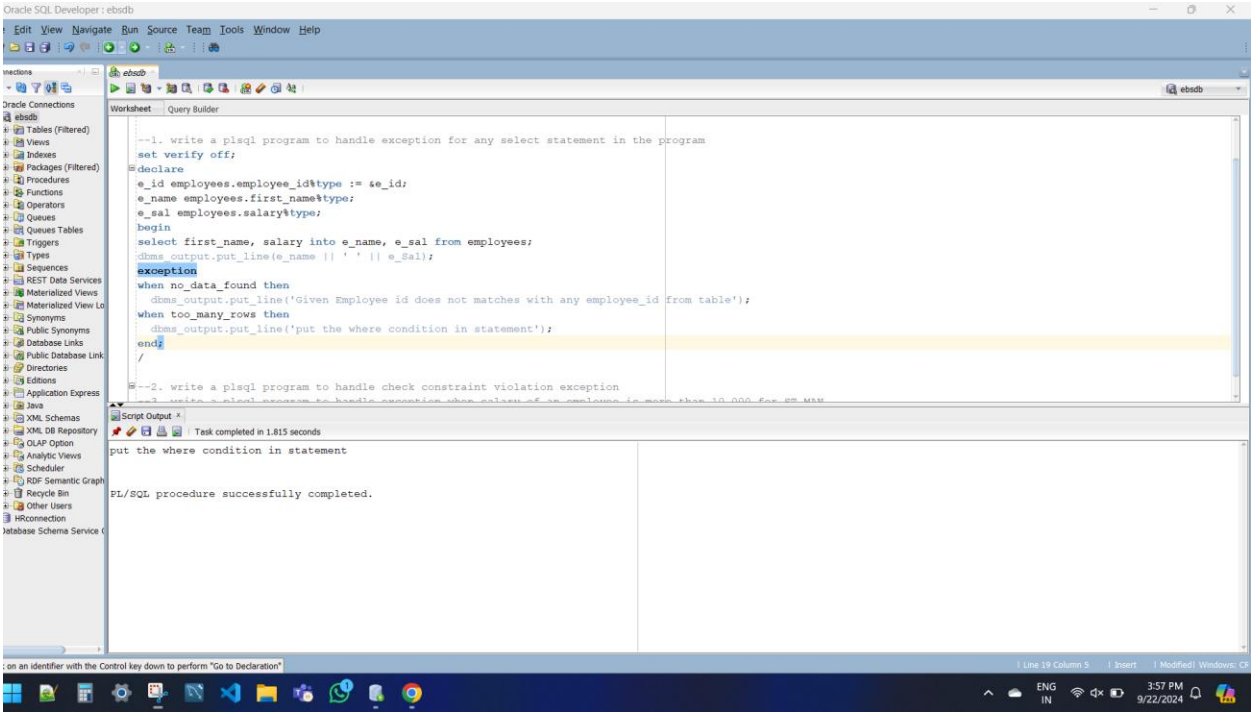
```
set serveroutput on;


---------- Exception ----------


--1. write a plsql program to handle exception for any select statement in the program

set verify off;

declare

e_id employees.employee_id%type := &e_id;

e_name employees.first_name%type;

e_sal employees.salary%type;

begin

select first_name, salary into e_name, e_sal from employees;

dbms_output.put_line(e_name || ' ' || e_Sal);

exception

when no_data_found then

  dbms_output.put_line('Given Employee id does not matches with any employee_id from table');

when too_many_rows then

  dbms_output.put_line('put the where condition in statement');

end;

/
```



--2. write a plsql program to handle check constraint violation exception

create table exc_table_onkar(id number(4) primary key, name varchar2(20));

select * from exc_table_onkar;

declare

e_id number(4):= &id;

e_name varchar2(20) := '&name';

check_error exception;

pragma EXCEPTION_INIT (check_error, -00001);

begin

insert into exc_table_onkar values(e_id, e_name);

exception

when check_error then

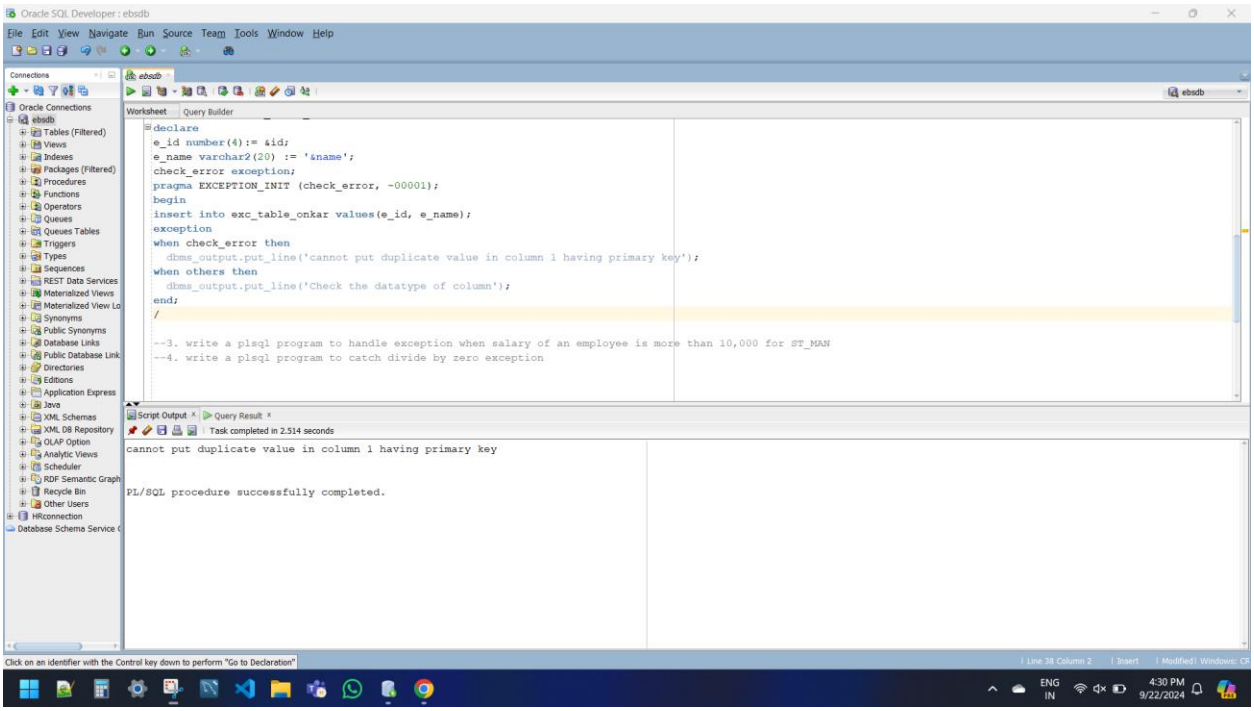
dbms_output.put_line('cannot put duplicate value in column 1 having primary key');

when others then

dbms_output.put_line('Check the datatype of column');

end;

/



--3. write a plsql program to handle exception when salary of an employee is more than 10,000 for ST_MAN

select * from employees where job_id = 'ST_MAN' and salary>10000;

declare

e_id number(4):= &emp_id;

e_sal number(8);

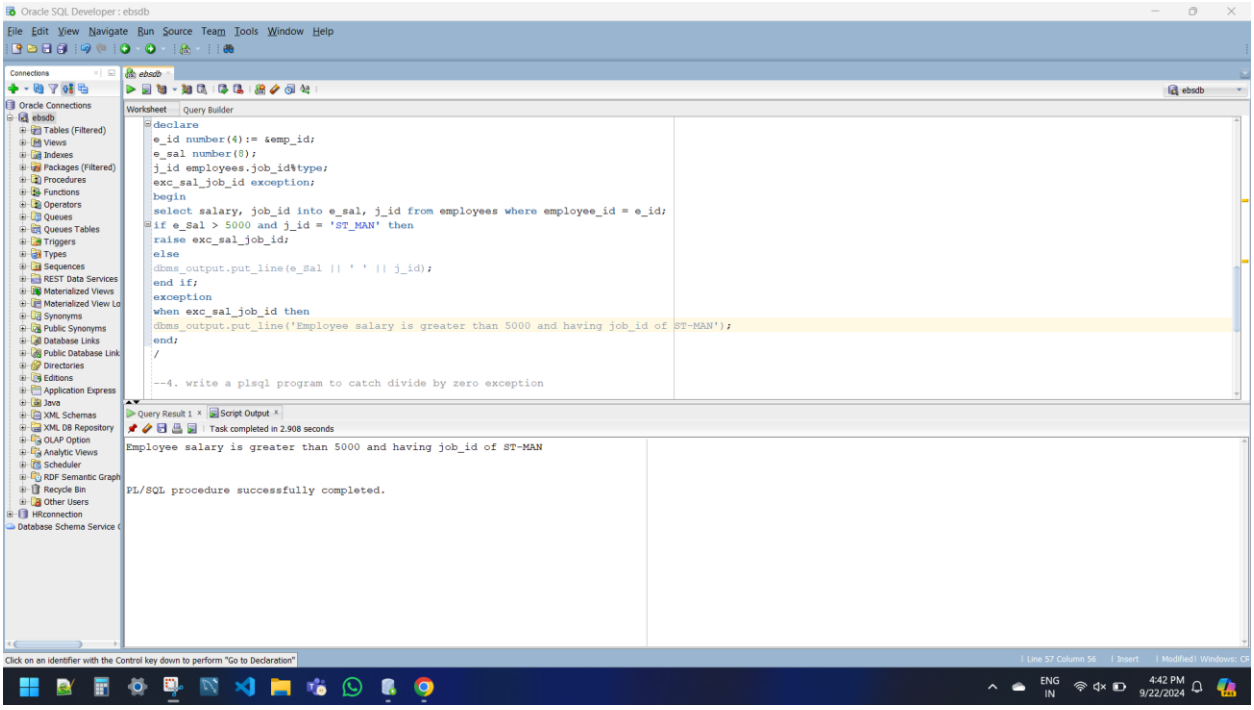
j_id employees.job_id%type;

exc_sal_job_id exception;

begin

select salary, job_id into e_sal, j_id from employees where employee_id = e_id;

```
if e_Sal > 10000 and j_id = 'ST_MAN' then
raise exc_sal_job_id;
else
dbms_output.put_line(e_Sal || ' ' || j_id);
end if;
exception
when exc_sal_job_id then
dbms_output.put_line('Employee salary is greater than 10000 and having job_id of ST-MAN');
end;
/
```




```
--4. write a plsql program to catch divide by zero exception
declare
number1 number(4) := &numerator;
number2 number(4) := &denominator;
```

```
answer number(8,4);
begin
answer := number1/number2;
dbms_output.put_line(answer);
exception
when zero_divide then
dbms_output.put_line('Denominator cannot be zero');
end;
/
```