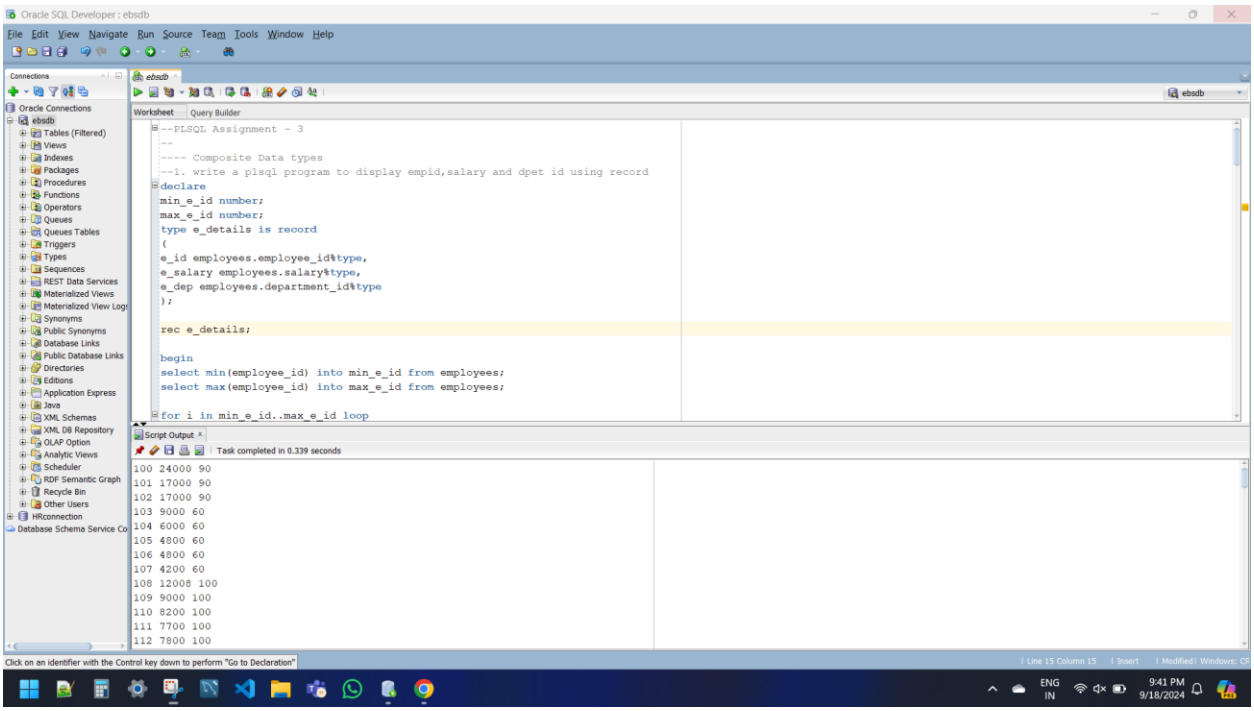
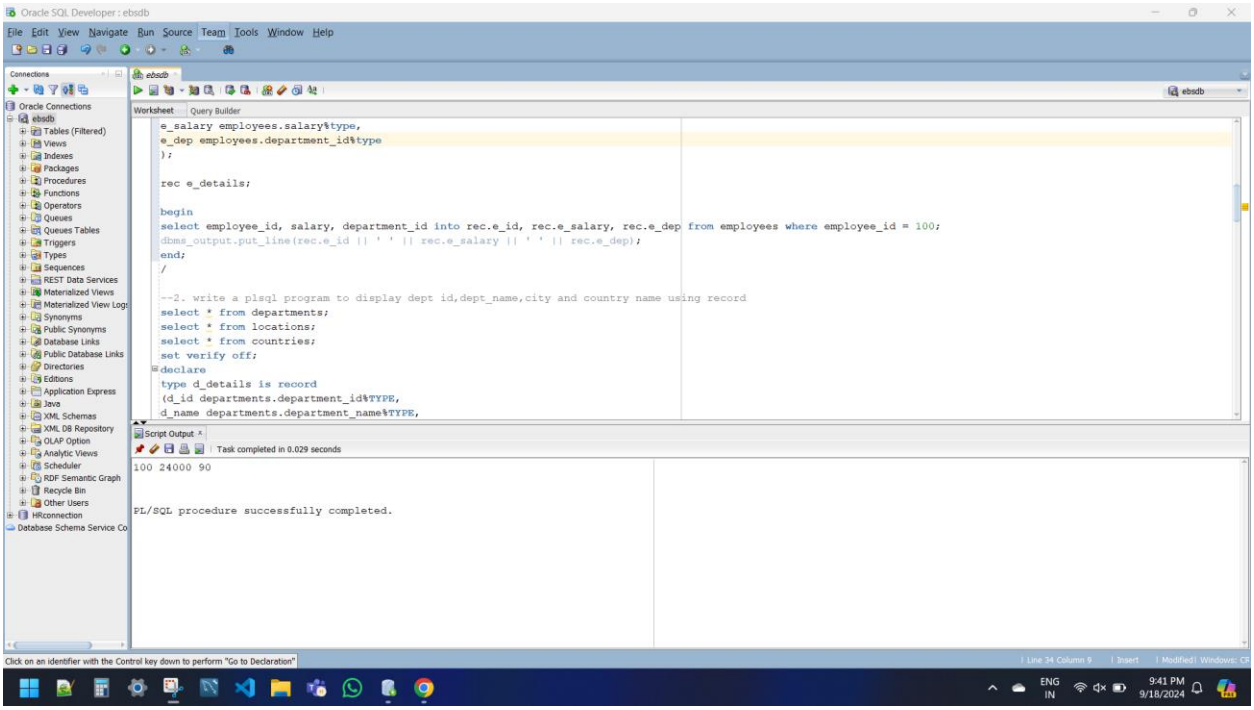
```
--PLSQL Assignment - 3
--
---- Composite Data types
--1. write a plsql program to display empid,salary and dpet id using record
declare
min_e_id number;
max_e_id number;
type e_details is record
(
e_id employees.employee_id%type,
e_salary employees.salary%type,
e_dep employees.department_id%type
);

rec e_details;

begin
select min(employee_id) into min_e_id from employees;
select max(employee_id) into max_e_id from employees;

for i in min_e_id..max_e_id loop
select employee_id, salary, department_id into rec.e_id, rec.e_salary, rec.e_dep from
employees where employee_id = i;
dbms_output.put_line(rec.e_id || ' ' || rec.e_salary || ' ' || rec.e_dep);
end loop;
end;
```

```
/
--or

declare
type e_details is record
(
e_id employees.employee_id%type,
e_salary employees.salary%type,
e_dep employees.department_id%type
);

rec e_details;

begin
select employee_id, salary, department_id into rec.e_id, rec.e_salary, rec.e_dep from
employees where employee_id = 100;
dbms_output.put_line(rec.e_id || ' ' || rec.e_salary || ' ' || rec.e_dep);
end;
/
```



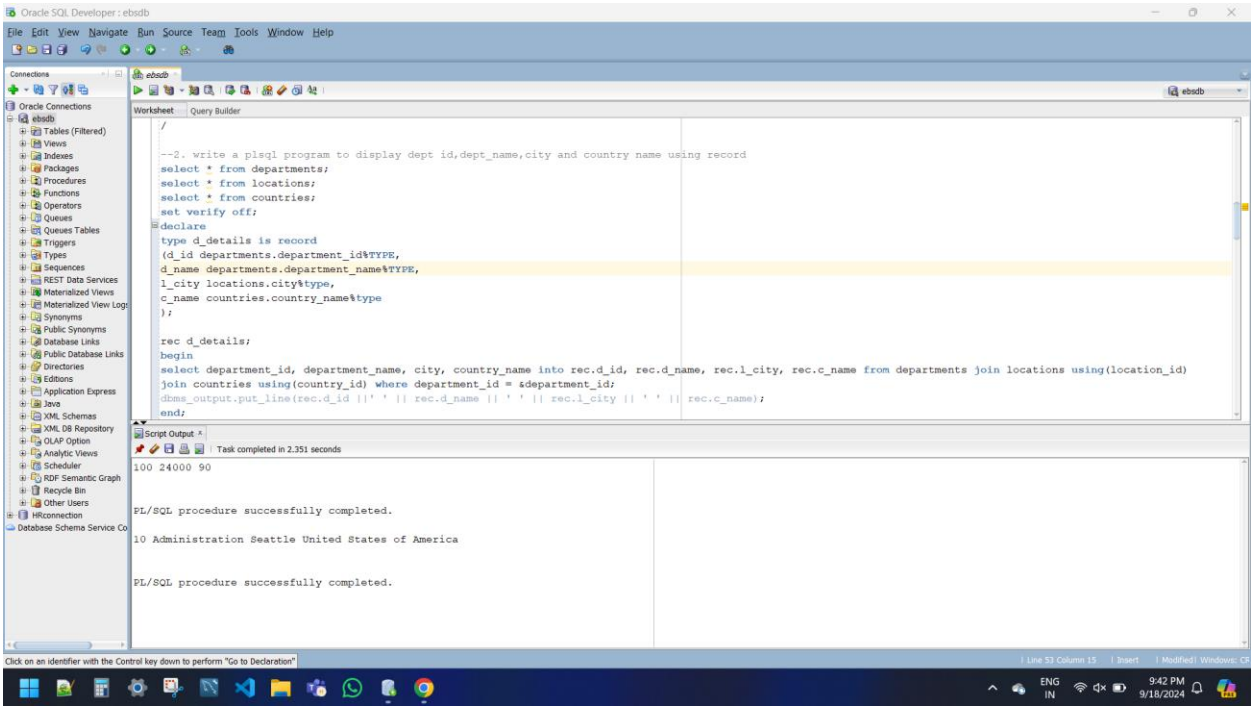


--2. write a plsql program to display dept id,dept_name,city and country name using record

select * from departments;

select * from locations;

```
select * from countries;

set verify off;

declare

type d_details is record

(d_id departments.department_id%TYPE,

d_name departments.department_name%TYPE,

l_city locations.city%type,

c_name countries.country_name%type

);


rec d_details;

begin

select department_id, department_name, city, country_name into rec.d_id, rec.d_name,
rec.l_city, rec.c_name from departments join locations using(location_id)

join countries using(country_id) where department_id = &department_id;

dbms_output.put_line(rec.d_id ||' ' || rec.d_name || ' ' || rec.l_city || ' ' || rec.c_name);

end;

/
```



--3. write a plsql program to display location details like loc id and city using %ROWTYPE

declare

rec locations%rowtype;

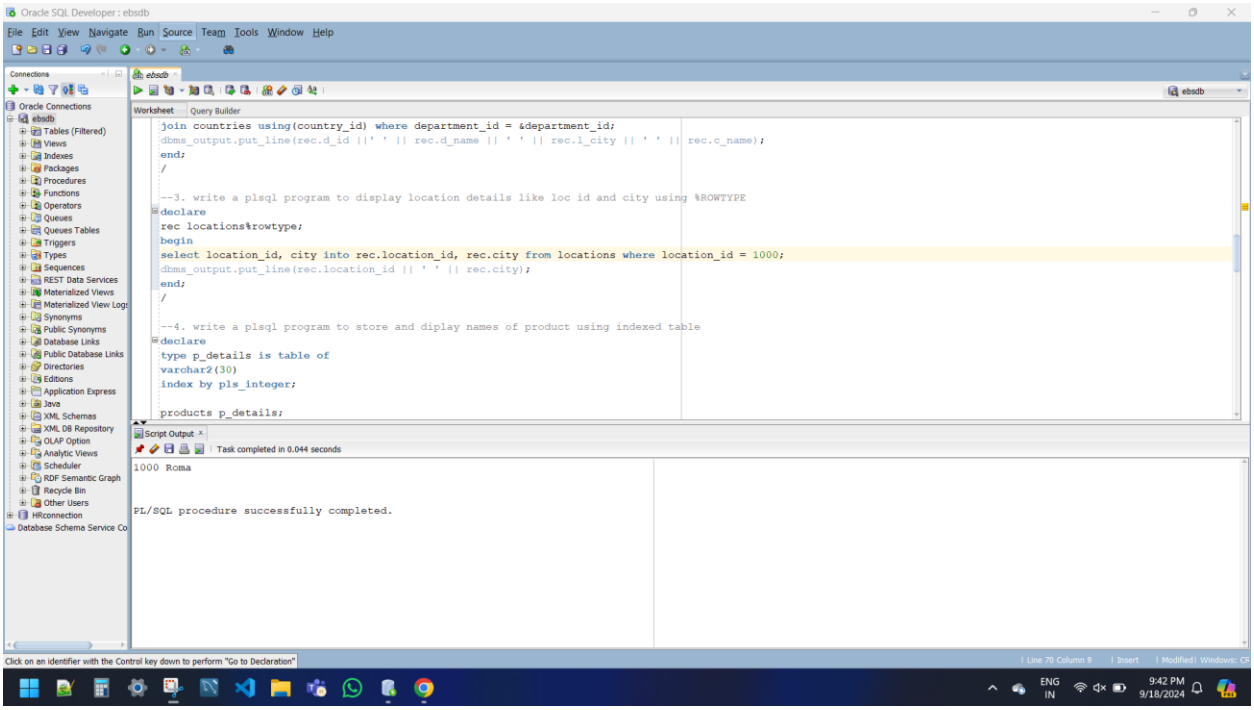
begin

select location_id, city into rec.location_id, rec.city from locations where location_id = 1000;

dbms_output.put_line(rec.location_id || ' ' || rec.city);

end;

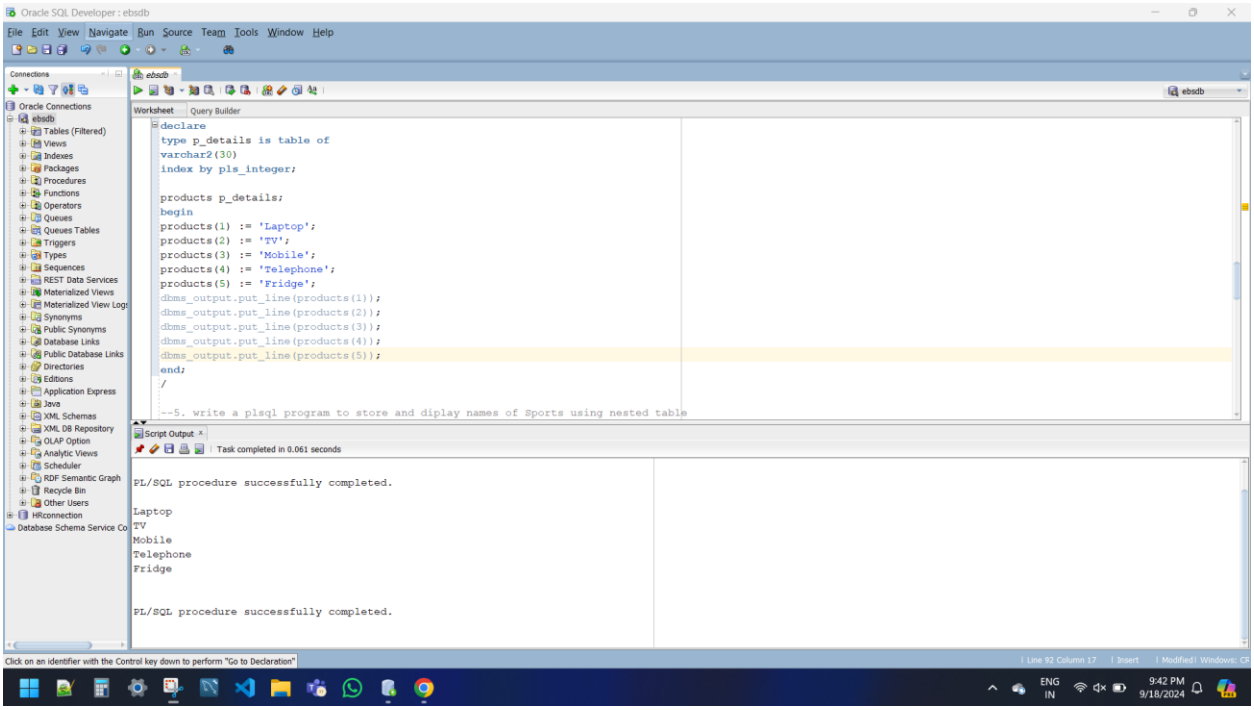
/



--4. write a plsql program to store and diplay names of product using indexed table

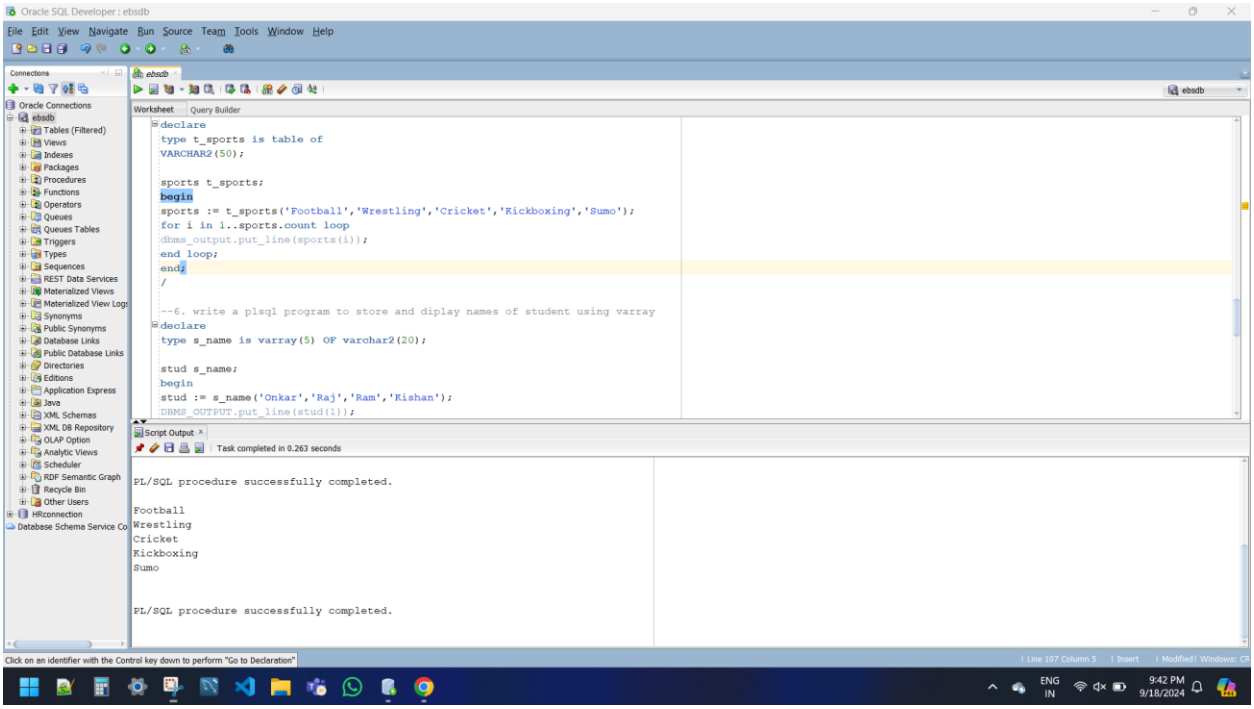
```
declare
type p_details is table of
varchar2(30)
index by pls_integer;

products p_details;
begin
products(1) := 'Laptop';
products(2) := 'TV';
products(3) := 'Mobile';
products(4) := 'Telephone';
products(5) := 'Fridge';
dbms_output.put_line(products(1));
```

```
dbms_output.put_line(products(2));
dbms_output.put_line(products(3));
dbms_output.put_line(products(4));
dbms_output.put_line(products(5));
end;
/
```




```
--5. write a plsql program to store and diplay names of Sports using nested table
declare
type t_sports is table of
VARCHAR2(50);

sports t_sports;
begin
sports := t_sports('Football','Wrestling','Cricket','Kickboxing','Sumo');
```

```
for i in 1..sports.count loop
dbms_output.put_line(sports(i));
end loop;
end;
/
```



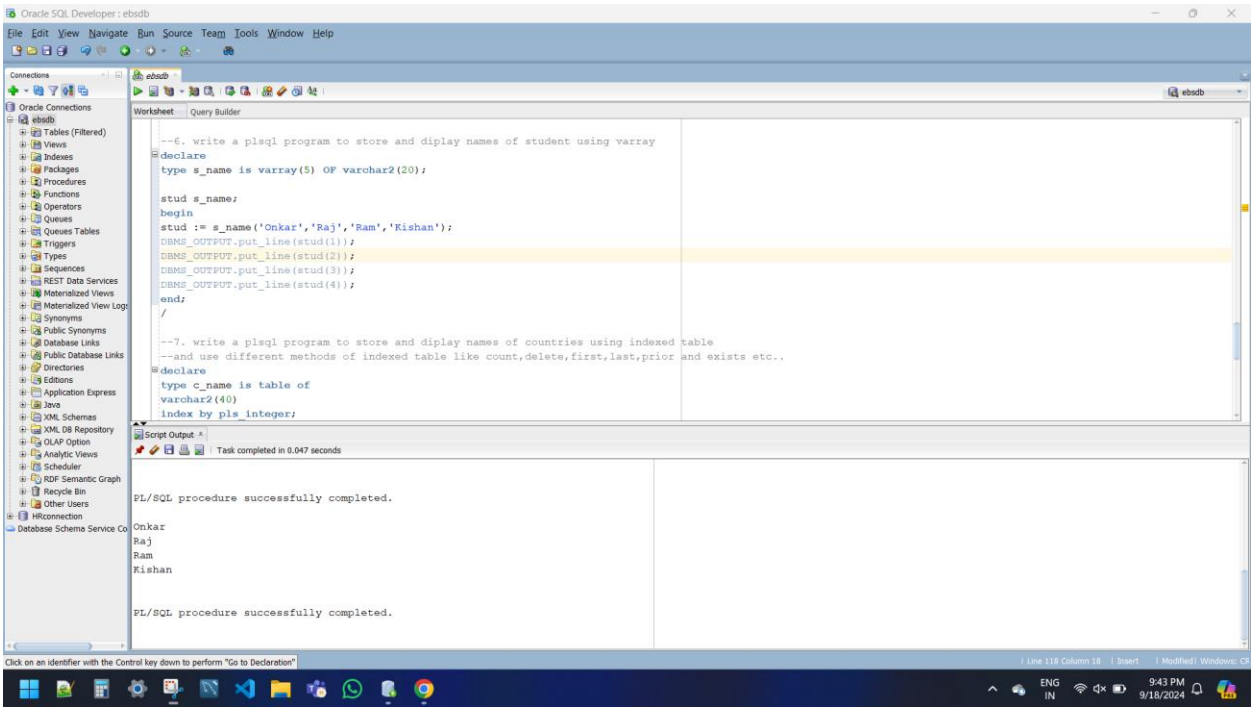

```
--6. write a plsql program to store and diplay names of student using varray
declare
type s_name is varray(5) OF varchar2(20);

stud s_name;
begin
stud := s_name('Onkar','Raj','Ram','Kishan');
DBMS_OUTPUT.put_line(stud(1));
DBMS_OUTPUT.put_line(stud(2));
```

DBMS_OUTPUT.put_line(stud(3));

DBMS_OUTPUT.put_line(stud(4));

end;

/



--7. write a plsql program to store and diplay names of countries using indexed table

--and use different methods of indexed table like count,delete,first,last,prior and exists etc..

declare

type c_name is table of

varchar2(40)

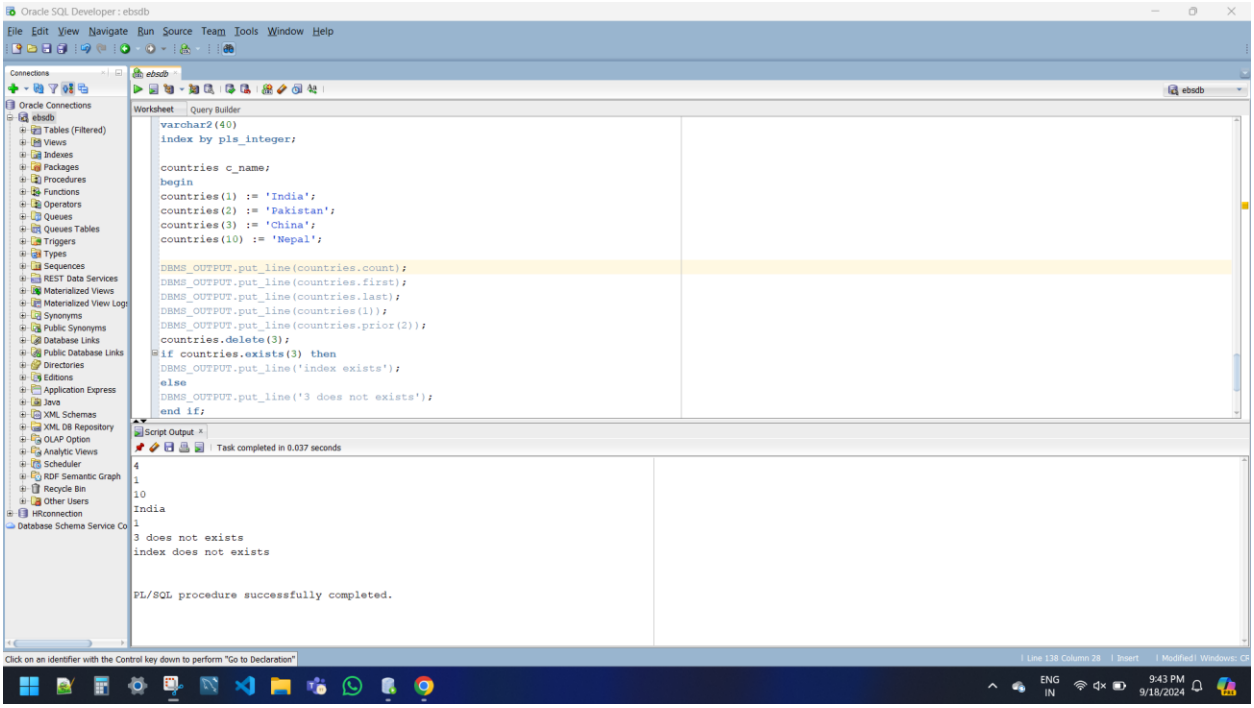
index by pls_integer;

countries c_name;

begin

countries(1) := 'India';

```
countries(2) := 'Pakistan';
countries(3) := 'China';
countries(10) := 'Nepal';

DBMS_OUTPUT.put_line(countries.count);
DBMS_OUTPUT.put_line(countries.first);
DBMS_OUTPUT.put_line(countries.last);
DBMS_OUTPUT.put_line(countries(1));
DBMS_OUTPUT.put_line(countries.prior(2));
countries.delete(3);
if countries.exists(3) then
DBMS_OUTPUT.put_line('index exists');
else
DBMS_OUTPUT.put_line('3 does not exists');
end if;

if countries.exists(110) then
DBMS_OUTPUT.put_line('index exists');
else
DBMS_OUTPUT.put_line('index does not exists');
end if;
end;
/
```


Oracle SQL Developer : ebsdb
File Edit View Navigate Run Source Team Tools Window Help
Connections
Oracle Connections
ebsdb
Tables (Filtered)
Views
Indexes
Packages
Procedures
Functions
Operators
Queues
Queues Tables
Triggers
Types
Sequences
REST Data Services
Materialized Views
Materialized View Logs
Synonyms
Public Synonyms
Database Links
Public Database Links
Directories
Editions
Application Express
Java
XML Schemas
XML DB Repository
OLAP Option
Analytic Views
Scheduler
RDF Semantic Graph
Recycle Bin
Other Users
HRconnection
Database Schema Service Co
ebsdb
Worksheet
Query Builder
varchar2(40)
index by pls_integer;
countries c_name;
begin
countries(1) := 'India';
countries(2) := 'Pakistan';
countries(3) := 'China';
countries(10) := 'Nepal';
DBMS_OUTPUT.put_line(countries.count);
DBMS_OUTPUT.put_line(countries.first);
DBMS_OUTPUT.put_line(countries.last);
DBMS_OUTPUT.put_line(countries(1));
DBMS_OUTPUT.put_line(countries.prior(2));
countries.delete(3);
if countries.exists(3) then
DBMS_OUTPUT.put_line('index exists');
else
DBMS_OUTPUT.put_line('3 does not exists');
end if;
Script Output
Task completed in 0.037 seconds
4
1
10
India
1
3 does not exists
index does not exists
PL/SQL procedure successfully completed.
Click on an identifier with the Control key down to perform "Go to Declaration"
Line 138 Column 28
Insert
Modified
Windows: CR
ENG
IN
9:43 PM
9/18/2024