

Lab Assignment - 4

Name: Ankur Prajapati

StudentID: N01324892

1. Step a to d:

The screenshot shows the SQL Developer interface. On the left, the 'Reports' pane is visible. The main window displays a PL/SQL procedure in the 'Worksheet' tab. The procedure is as follows:

```
1 SELECT * FROM countries;
2
3 SET SERVEROUTPUT ON;
4 SET VERIFY OFF;
5 DECLARE
6     v_countryid VARCHAR2(20);
7     v_country_record countries%ROWTYPE;
8 BEGIN
9     SELECT * INTO v_country_record FROM countries WHERE COUNTRY_ID = 'CA';
10    --SELECT * INTO v_country_record FROM countries WHERE COUNTRY_ID = 'DK';
11    --SELECT * INTO v_country_record FROM countries WHERE COUNTRY_ID = 'UK';
12    --SELECT * INTO v_country_record FROM countries WHERE COUNTRY_ID = 'USA';
13    DBMS_OUTPUT.PUT_LINE ('Country Id: ' || v_country_record.country_id || ' Country Name: '
14    || v_country_record.country_name || ' Region: ' || v_country_record.region_id);
15 END;
```

The 'Script Output' pane at the bottom shows the result of the execution:

```
Country Id: CA Country Name: Canada Region: 2

PL/SQL procedure successfully completed.
```

Here I have created variable `v_countryid` of type `VARCHAR2` to store the information of the county id. Using that variable, I am printing record of Country 'CA'. Note that the record `v_country_record` has the same type of table `countries` due to the `%ROWTYPE`.

Step e:

The screenshot shows the SQL Developer interface. The main window displays a PL/SQL procedure in the 'Worksheet' tab. The procedure is as follows:

```
1 SET SERVEROUTPUT ON;
2 SET VERIFY OFF;
3 DECLARE
4     v_countryid VARCHAR2(20) := 'CA';
5     v_countryid1 VARCHAR2(20) := 'DK';
6     v_countryid2 VARCHAR2(20) := 'UK';
7     v_countryid3 VARCHAR2(20) := 'US';
8     v_country_record countries%ROWTYPE;
9 BEGIN
10    SELECT * INTO v_country_record FROM countries WHERE COUNTRY_ID = v_countryid;
11    DBMS_OUTPUT.PUT_LINE ('Country Id: ' || v_country_record.country_id || ' Country Name: ' || v_country_record.country_name || ' Region: ' || v_country_record.region_id);
12    SELECT * INTO v_country_record FROM countries WHERE COUNTRY_ID = v_countryid1;
13    DBMS_OUTPUT.PUT_LINE ('Country Id: ' || v_country_record.country_id || ' Country Name: ' || v_country_record.country_name || ' Region: ' || v_country_record.region_id);
14    SELECT * INTO v_country_record FROM countries WHERE COUNTRY_ID = v_countryid2;
15    DBMS_OUTPUT.PUT_LINE ('Country Id: ' || v_country_record.country_id || ' Country Name: ' || v_country_record.country_name || ' Region: ' || v_country_record.region_id);
16    SELECT * INTO v_country_record FROM countries WHERE COUNTRY_ID = v_countryid3;
17    DBMS_OUTPUT.PUT_LINE ('Country Id: ' || v_country_record.country_id || ' Country Name: ' || v_country_record.country_name || ' Region: ' || v_country_record.region_id);
18 END;
```

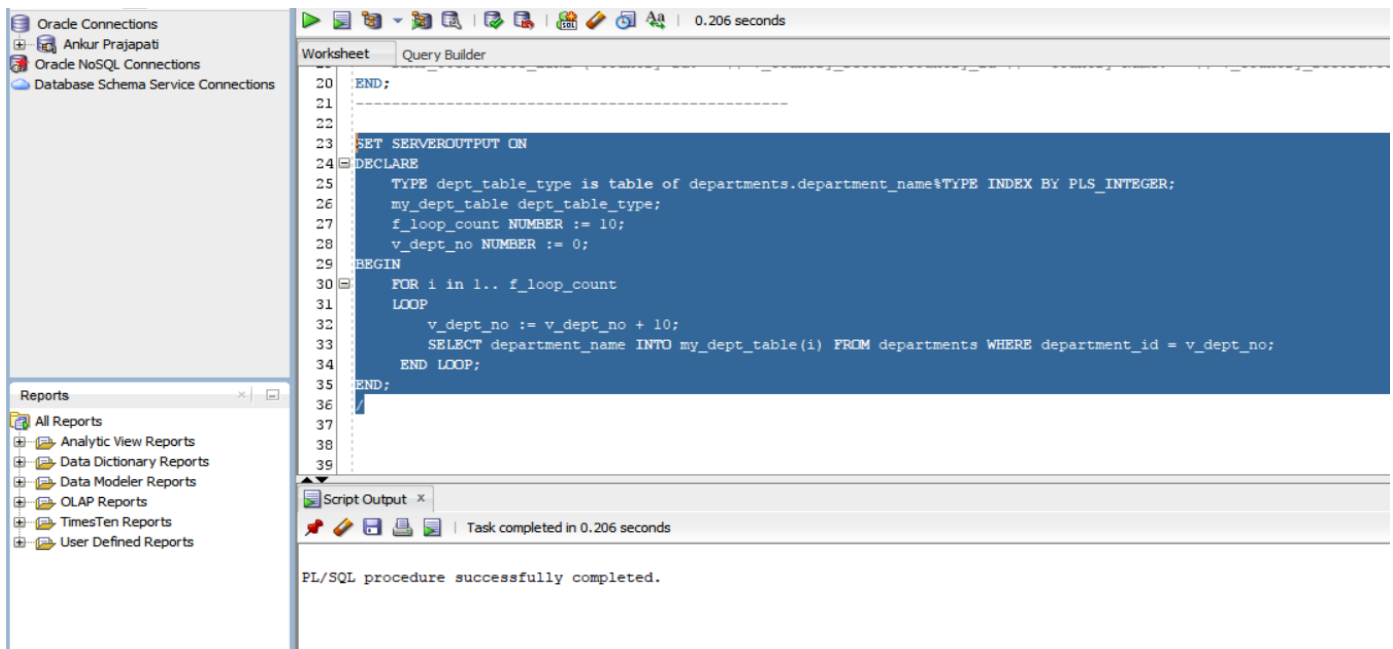
The 'Script Output' pane at the bottom shows the results of the execution:

```
Country Id: CA Country Name: Canada Region: 2
Country Id: DK Country Name: Denmark Region: 1
Country Id: UK Country Name: United Kingdom Region: 1
Country Id: US Country Name: United States of America Region: 2

PL/SQL procedure successfully completed.
```

In step e, I am printing record related to the 'ÇA', 'DA', 'UK', 'US'.

2. Step a to c:



The screenshot shows the SQL Developer interface. On the left, the 'Reports' pane lists various report types. The main area is the 'Query Builder' window, which contains a PL/SQL procedure. The procedure declares a table type, a table variable, and two numeric variables. It then enters a loop that iterates 10 times, incrementing a department number and selecting the department name into a table variable. The procedure ends with a 'PL/SQL procedure successfully completed.' message in the 'Script Output' pane at the bottom.

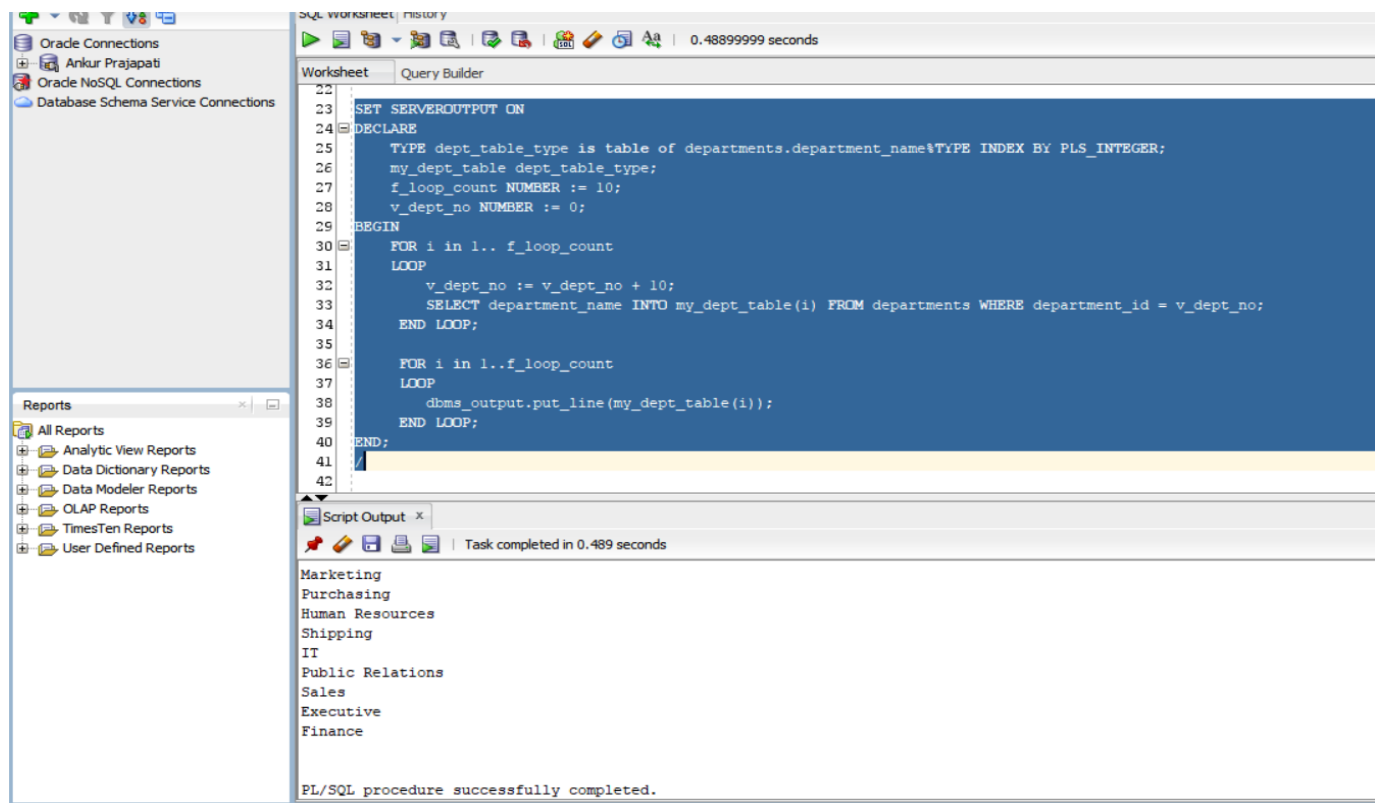
```
20 END;
21
22
23 SET SERVEROUTPUT ON
24 DECLARE
25     TYPE dept_table_type is table of departments.department_name%TYPE INDEX BY PLS_INTEGER;
26     my_dept_table dept_table_type;
27     f_loop_count NUMBER := 10;
28     v_dept_no NUMBER := 0;
29 BEGIN
30     FOR i in 1.. f_loop_count
31     LOOP
32         v_dept_no := v_dept_no + 10;
33         SELECT department_name INTO my_dept_table(i) FROM departments WHERE department_id = v_dept_no;
34     END LOOP;
35 END;
```

Script Output x
Task completed in 0.206 seconds
PL/SQL procedure successfully completed.

Here I have created dept_table_type and given the Index BY PLS_INTEGER and created two variables one for loop and other to increment the department number by 10.

In first loop it gets the value of department id and in second loop it gets the value of department name.

Step d:

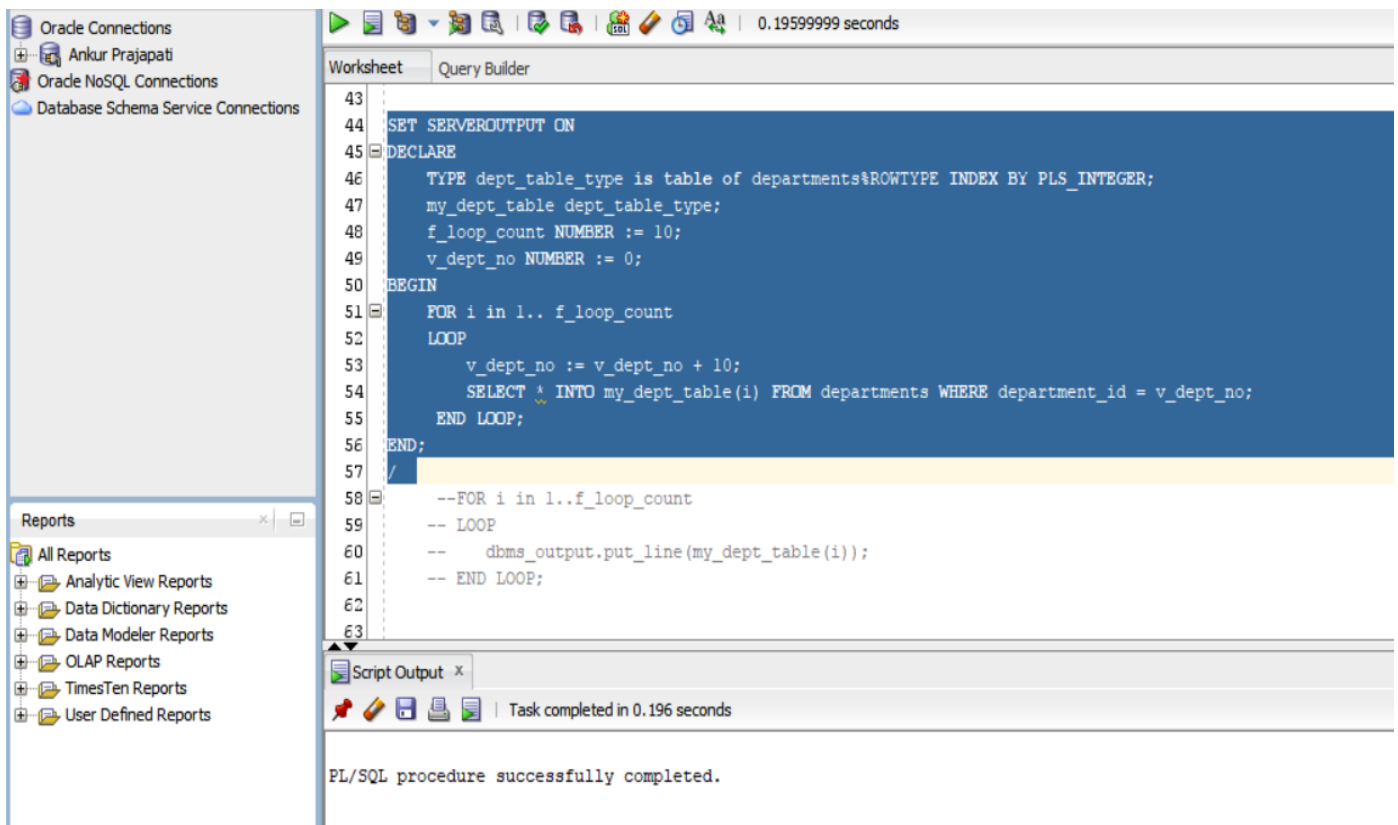


The screenshot shows the SQL Developer interface. The 'Query Builder' window contains a PL/SQL procedure that is similar to the one in the previous screenshot, but it includes a second loop that prints the department names using the 'dbms_output.put_line' function. The 'Script Output' pane at the bottom shows the list of department names: Marketing, Purchasing, Human Resources, Shipping, IT, Public Relations, Sales, Executive, and Finance. The procedure ends with a 'PL/SQL procedure successfully completed.' message.

```
22 SET SERVEROUTPUT ON
23 DECLARE
24     TYPE dept_table_type is table of departments.department_name%TYPE INDEX BY PLS_INTEGER;
25     my_dept_table dept_table_type;
26     f_loop_count NUMBER := 10;
27     v_dept_no NUMBER := 0;
28 BEGIN
29     FOR i in 1.. f_loop_count
30     LOOP
31         v_dept_no := v_dept_no + 10;
32         SELECT department_name INTO my_dept_table(i) FROM departments WHERE department_id = v_dept_no;
33     END LOOP;
34
35     FOR i in 1..f_loop_count
36     LOOP
37         dbms_output.put_line(my_dept_table(i));
38     END LOOP;
39 END;
```

Script Output x
Task completed in 0.489 seconds
Marketing
Purchasing
Human Resources
Shipping
IT
Public Relations
Sales
Executive
Finance
PL/SQL procedure successfully completed.

3. Step a to c:



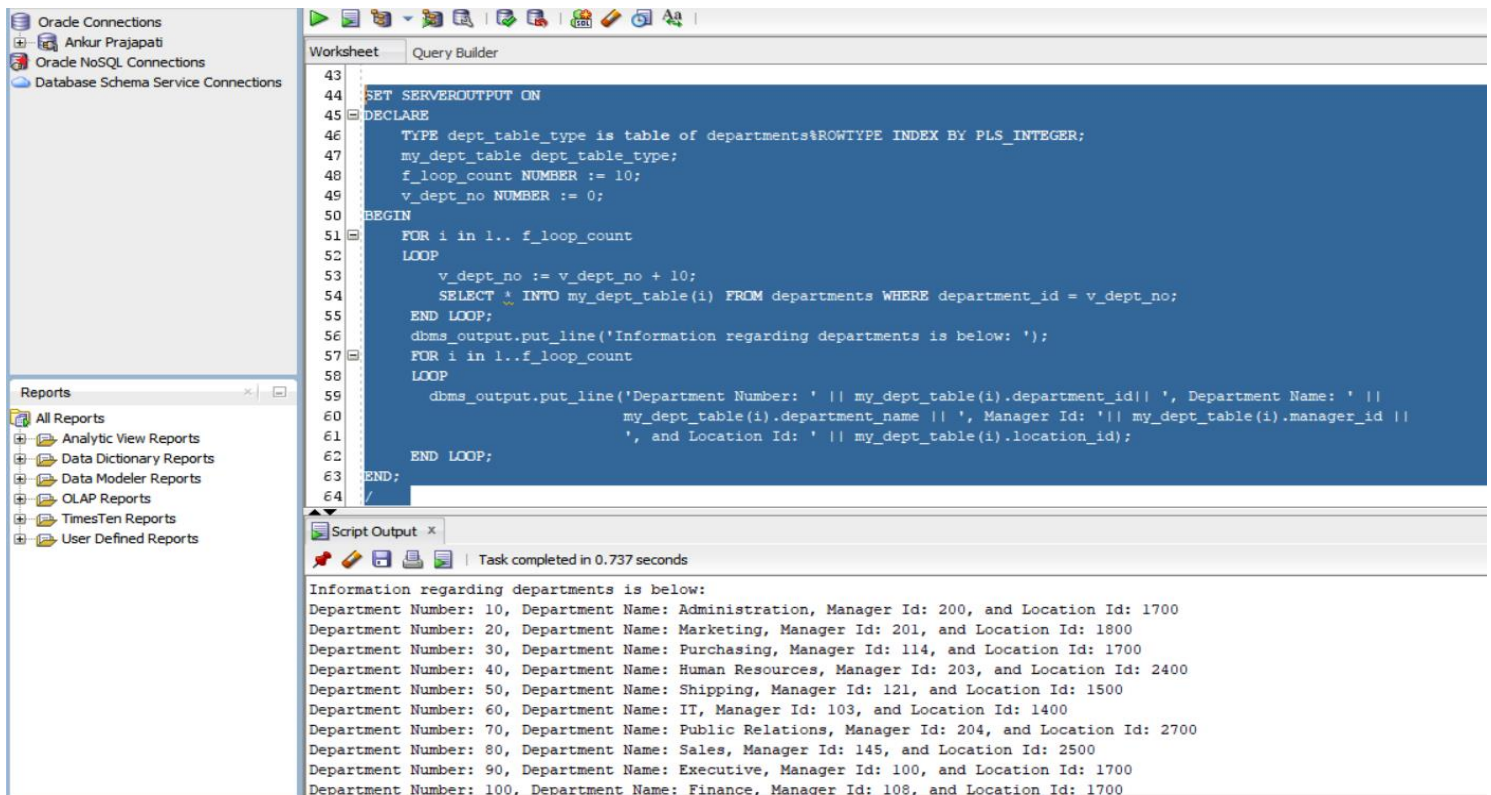
```
43
44 SET SERVEROUTPUT ON
45 DECLARE
46     TYPE dept_table_type is table of departments%ROWTYPE INDEX BY PLS_INTEGER;
47     my_dept_table dept_table_type;
48     f_loop_count NUMBER := 10;
49     v_dept_no NUMBER := 0;
50 BEGIN
51     FOR i in 1.. f_loop_count
52     LOOP
53         v_dept_no := v_dept_no + 10;
54         SELECT * INTO my_dept_table(i) FROM departments WHERE department_id = v_dept_no;
55     END LOOP;
56 END;
57 /
58 --FOR i in 1..f_loop_count
59 -- LOOP
60 --     dbms_output.put_line(my_dept_table(i));
61 -- END LOOP;
62
63
```

Script Output x

Task completed in 0.196 seconds

PL/SQL procedure successfully completed.

Here I have modified the declaration part of dept_table_type to keep it same as the departments table using %ROWTYPE. Now instead of selecting only the department id I am storing all the values from the department table to my_dept_table using for Loop.



```
43
44 SET SERVEROUTPUT ON
45 DECLARE
46     TYPE dept_table_type is table of departments%ROWTYPE INDEX BY PLS_INTEGER;
47     my_dept_table dept_table_type;
48     f_loop_count NUMBER := 10;
49     v_dept_no NUMBER := 0;
50 BEGIN
51     FOR i in 1.. f_loop_count
52     LOOP
53         v_dept_no := v_dept_no + 10;
54         SELECT * INTO my_dept_table(i) FROM departments WHERE department_id = v_dept_no;
55     END LOOP;
56     dbms_output.put_line('Information regarding departments is below: ');
57     FOR i in 1..f_loop_count
58     LOOP
59         dbms_output.put_line('Department Number: ' || my_dept_table(i).department_id || ', Department Name: ' ||
60             my_dept_table(i).department_name || ', Manager Id: ' || my_dept_table(i).manager_id ||
61             ', and Location Id: ' || my_dept_table(i).location_id);
62     END LOOP;
63 END;
64 /
```

Script Output x

Task completed in 0.737 seconds

Information regarding departments is below:
Department Number: 10, Department Name: Administration, Manager Id: 200, and Location Id: 1700
Department Number: 20, Department Name: Marketing, Manager Id: 201, and Location Id: 1800
Department Number: 30, Department Name: Purchasing, Manager Id: 114, and Location Id: 1700
Department Number: 40, Department Name: Human Resources, Manager Id: 203, and Location Id: 2400
Department Number: 50, Department Name: Shipping, Manager Id: 121, and Location Id: 1500
Department Number: 60, Department Name: IT, Manager Id: 103, and Location Id: 1400
Department Number: 70, Department Name: Public Relations, Manager Id: 204, and Location Id: 2700
Department Number: 80, Department Name: Sales, Manager Id: 145, and Location Id: 2500
Department Number: 90, Department Name: Executive, Manager Id: 100, and Location Id: 1700
Department Number: 100, Department Name: Finance, Manager Id: 108, and Location Id: 1700

Because we are storing all the column information from the departments table now, we have to modify the second loop to print the column data in order to the records we have.