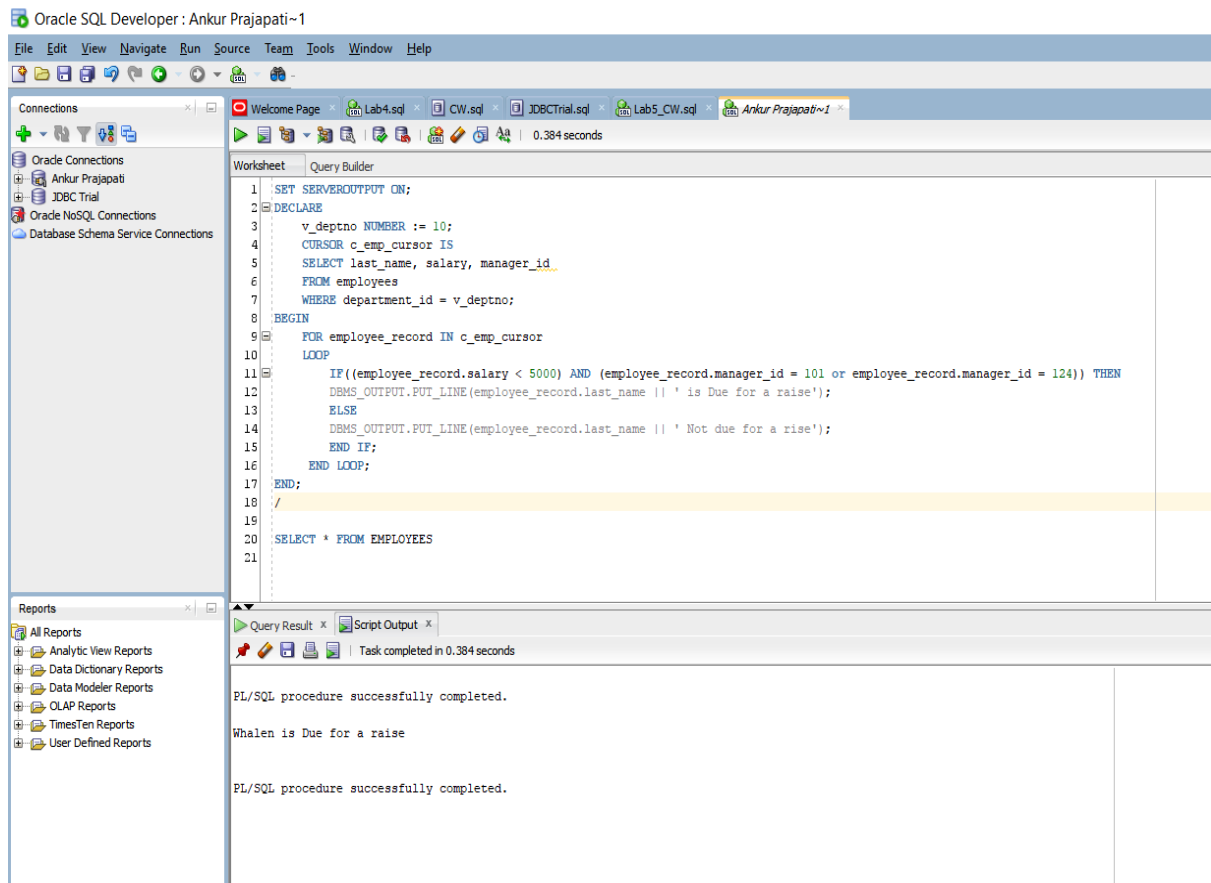


Practice 1:

Question 1:

Test Outputs:



The screenshot displays the Oracle SQL Developer interface. The main window shows a PL/SQL script in the 'Worksheet' tab. The script declares a cursor 'c_emp_cursor' to select last_name, salary, and manager_id from employees where department_id equals v_deptno. It then uses a loop to iterate through the cursor, printing a message for each employee if their salary is less than 5000 and their manager_id is 101 or 124. The script concludes with a 'SELECT * FROM EMPLOYEES' statement. The 'Script Output' tab at the bottom shows the execution results, indicating that the PL/SQL procedure completed successfully and printed the message 'Whalen is Due for a raise'.

```
1 SET SERVEROUTPUT ON;
2 DECLARE
3   v_deptno NUMBER := 10;
4   CURSOR c_emp_cursor IS
5     SELECT last_name, salary, manager_id
6     FROM employees
7     WHERE department_id = v_deptno;
8 BEGIN
9   FOR employee_record IN c_emp_cursor
10  LOOP
11    IF ((employee_record.salary < 5000) AND (employee_record.manager_id = 101 OR employee_record.manager_id = 124)) THEN
12      DBMS_OUTPUT.PUT_LINE(employee_record.last_name || ' is Due for a raise');
13    ELSE
14      DBMS_OUTPUT.PUT_LINE(employee_record.last_name || ' Not due for a raise');
15    END IF;
16  END LOOP;
17 END;
18 /
19
20 SELECT * FROM EMPLOYEES
21
```

PL/SQL procedure successfully completed.

Whalen is Due for a raise

PL/SQL procedure successfully completed.

Here I have created CURSOR named c_emp_cursor and it selects last name, salary and manager id from employees where cdepartment id is same as v_deptno.

In executable section I used LOOP for employee_record in c_emp_cursor to print the output.

Oracle SQL Developer : Ankur Prajapati~1

File Edit View Navigate Run Source Team Tools Window Help

Connections

- Oracle Connections
 - Ankur Prajapati
 - JDBC Trial
 - Oracle NoSQL Connections
 - Database Schema Service Connections

Worksheet

```

1 SET SERVEROUTPUT ON;
2 DECLARE
3   v_deptno NUMBER := 20;
4   CURSOR c_emp_cursor IS
5     SELECT last_name, salary, manager_id
6     FROM employees
7     WHERE department_id = v_deptno;
8 BEGIN
9   FOR employee_record IN c_emp_cursor
10  LOOP
11    IF((employee_record.salary < 5000) AND (employee_record.manager_id = 101 or employee_record.manager_id = 124)) THEN
12      DBMS_OUTPUT.PUT_LINE(employee_record.last_name || ' is Due for a raise');
13    ELSE
14      DBMS_OUTPUT.PUT_LINE(employee_record.last_name || ' Not due for a rise');
15    END IF;
16  END LOOP;
17 END;
18 /
19
20 SELECT * FROM EMPLOYEES
21

```

Query Result x Script Output x

Task completed in 0.198 seconds

PL/SQL procedure successfully completed.

Whalen is Due for a raise

PL/SQL procedure successfully completed.

Hartstein Not due for a rise
Fay Not due for a rise

PL/SQL procedure successfully completed.

Oracle SQL Developer : Ankur Prajapati~1

File Edit View Navigate Run Source Team Tools Window Help

Connections

- Oracle Connections
 - Ankur Prajapati
 - JDBC Trial
 - Oracle NoSQL Connections
 - Database Schema Service Connections

Worksheet

```

1 SET SERVEROUTPUT ON;
2 DECLARE
3   v_deptno NUMBER := 50;
4   CURSOR c_emp_cursor IS
5     SELECT last_name, salary, manager_id
6     FROM employees
7     WHERE department_id = v_deptno;
8 BEGIN
9   FOR employee_record IN c_emp_cursor
10  LOOP
11    IF((employee_record.salary < 5000) AND (employee_record.manager_id = 101 or employee_record.manager_id = 124)) THEN
12      DBMS_OUTPUT.PUT_LINE(employee_record.last_name || ' is Due for a raise');
13    ELSE
14      DBMS_OUTPUT.PUT_LINE(employee_record.last_name || ' Not due for a rise');
15    END IF;
16  END LOOP;
17 END;
18 /
19
20 SELECT * FROM EMPLOYEES
21

```

Query Result x Script Output x

Task completed in 0.181 seconds

Weiss Not due for a rise
Fripp Not due for a rise
Kaufling Not due for a rise
Vollman Not due for a rise
Mourgos Not due for a rise
Nayer Not due for a rise
Mikkilineni Not due for a rise
Landry Not due for a rise
Markle Not due for a rise
Bissot Not due for a rise
Atkinson Not due for a rise
Marlow Not due for a rise
Olson Not due for a rise
Mallin Not due for a rise
Rogers Not due for a rise
Gee Not due for a rise

The screenshot displays the Oracle SQL Developer interface. The main window shows a PL/SQL script in the 'Query Builder' tab. The script is as follows:

```
1 SET SERVEROUTPUT ON;
2 DECLARE
3   v_deptno NUMBER := 80;
4   CURSOR c_emp_cursor IS
5     SELECT last_name, salary, manager_id
6     FROM employees
7     WHERE department_id = v_deptno;
8 BEGIN
9   FOR employee_record IN c_emp_cursor
10  LOOP
11    IF (employee_record.salary < 5000) AND (employee_record.manager_id = 101 OR employee_record.manager_id = 124) THEN
12      DBMS_OUTPUT.PUT_LINE(employee_record.last_name || ' is Due for a raise');
13    ELSE
14      DBMS_OUTPUT.PUT_LINE(employee_record.last_name || ' Not due for a raise');
15    END IF;
16  END LOOP;
17 END;
```

Below the script, the 'Query Result' tab shows the output of the script, which is a list of employee names followed by a message indicating whether they are due for a raise. The output is as follows:

```
Russell Not due for a raise
Partners Not due for a raise
Errazuriz Not due for a raise
Cambrault Not due for a raise
Zlotkey Not due for a raise
Tucker Not due for a raise
Bernstein Not due for a raise
Hall Not due for a raise
Olsen Not due for a raise
Cambrault Not due for a raise
Tuvault Not due for a raise
King Not due for a raise
```

These above output shows for all the department id given in PDF.

Question 2:

Oracle SQL Developer : Ankur Prajapati~1

```

File Edit View Navigate Run Source Team Tools Window Help
Connections
  Oracle Connections
    Ankur Prajapati
    JDBC Trial
  Oracle NoSQL Connections
  Database Schema Service Connections
Reports
  All Reports
    Analytic View Reports
    Data Dictionary Reports
    Data Modeler Reports
    OLAP Reports
    TimesTen Reports
    User Defined Reports
Worksheet
  1 -----Question 2
  2 DECLARE
  3 -----first cursor
  4 CURSOR c_dept_cursor IS
  5   SELECT department_id, department_name..
  6   FROM departments
  7   WHERE department_id < 100
  8   ORDER BY department_id;
  9 -----second cursor
 10 CURSOR c_emp_cursor(v_dep_id departments.department_id%TYPE) IS
 11   SELECT last_name, job_id, hire_date, salary..
 12   FROM employees
 13   WHERE employee_id < 120 AND department_id = v_dep_id;
 14 -----declaring variables
 15 v_department_id departments.department_id%TYPE;
 16 v_department_name departments.department_name%TYPE;
 17 v_name employees.last_name%TYPE;
 18 v_job_id employees.job_id%TYPE;
 19 v_hire_date employees.hire_date%TYPE;
 20 v_salary employees.salary%TYPE;
 21 BEGIN
 22   OPEN c_dept_cursor;
 23   LOOP
 24     FETCH c_dept_cursor INTO v_department_id, v_department_name;
 25     EXIT WHEN c_dept_cursor%NOTFOUND;
 26     DBMS_OUTPUT.PUT_LINE('Department no is: ' || v_department_id || ' Name is: ' || v_department_name);
 27   END LOOP;
 28   CLOSE c_dept_cursor;
 29 END;
 30
Script Output
  Task completed in 0.131 seconds
  Department no is: 10 Name is: Administration
  Department no is: 20 Name is: Marketing
  Department no is: 30 Name is: Purchasing
  Department no is: 40 Name is: Human Resources
  Department no is: 50 Name is: Shipping
  Department no is: 60 Name is: IT
  Department no is: 70 Name is: Public Relations
  Department no is: 80 Name is: Sales
  Department no is: 90 Name is: Executive
  PL/SQL procedure successfully completed.

```

Oracle SQL Developer : Ankur Prajapati~1

```

File Edit View Navigate Run Source Team Tools Window Help
Connections
  Oracle Connections
    Ankur Prajapati
    JDBC Trial
  Oracle NoSQL Connections
  Database Schema Service Connections
Reports
  All Reports
    Analytic View Reports
    Data Dictionary Reports
    Data Modeler Reports
    OLAP Reports
    TimesTen Reports
    User Defined Reports
Worksheet
  1 -----Question 2
  2 DECLARE
  3 -----first cursor
  4 CURSOR c_dept_cursor IS
  5   SELECT department_id, department_name..
  6   FROM departments
  7   WHERE department_id < 100
  8   ORDER BY department_id;
  9 -----second cursor
 10 CURSOR c_emp_cursor(v_dep_id departments.department_id%TYPE) IS
 11   SELECT last_name, job_id, hire_date, salary..
 12   FROM employees
 13   WHERE employee_id < 120 AND department_id = v_dep_id;
 14 -----declaring variables
 15 v_department_id departments.department_id%TYPE;
 16 v_department_name departments.department_name%TYPE;
 17 v_name employees.last_name%TYPE;
 18 v_job_id employees.job_id%TYPE;
 19 v_hire_date employees.hire_date%TYPE;
 20 v_salary employees.salary%TYPE;
 21 BEGIN
 22   OPEN c_dept_cursor;
 23   LOOP
 24     FETCH c_dept_cursor INTO v_department_id, v_department_name;
 25     EXIT WHEN c_dept_cursor%NOTFOUND;
 26     DBMS_OUTPUT.PUT_LINE('Department no: ' || v_department_id || ' Name: ' || v_department_name);
 27     OPEN c_emp_cursor(v_department_id);
 28     LOOP
 29       FETCH c_emp_cursor INTO v_name, v_job_id, v_hire_date, v_salary;
 30       EXIT WHEN c_emp_cursor%NOTFOUND;
 31       DBMS_OUTPUT.PUT_LINE(v_name || ' ' || v_job_id || ' ' || v_hire_date || ' ' || v_salary);
 32     END LOOP;
 33     CLOSE c_emp_cursor;
 34     DBMS_OUTPUT.PUT_LINE('-----');
 35   END LOOP;
 36   CLOSE c_dept_cursor;
 37 END;
 38 /
 39

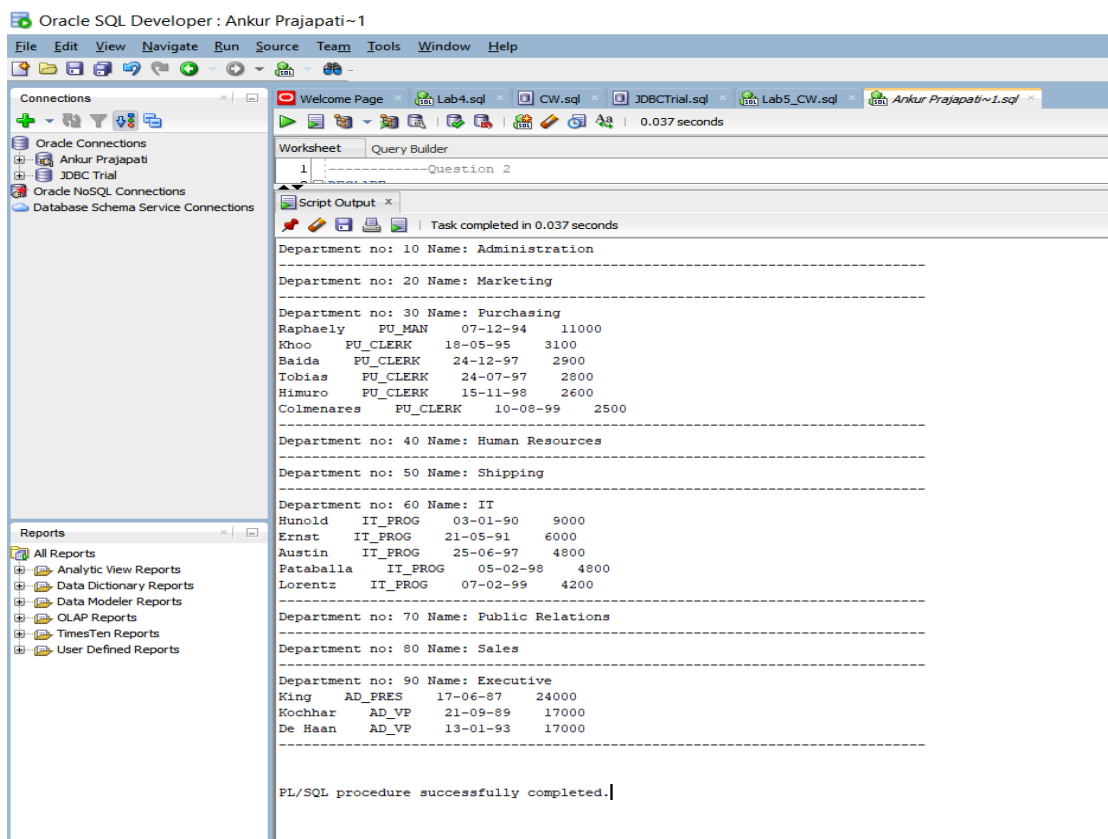
```

First, I declared cursor named c_dept_cursor which selects multiple rows where department id is less than 100. After that I declared second cursor named c_emp_cursor having parameter of v_dep_id which is having same data type of department_id column of department table.

After that I have declared variables as asked in PDF which will be used to print the output.

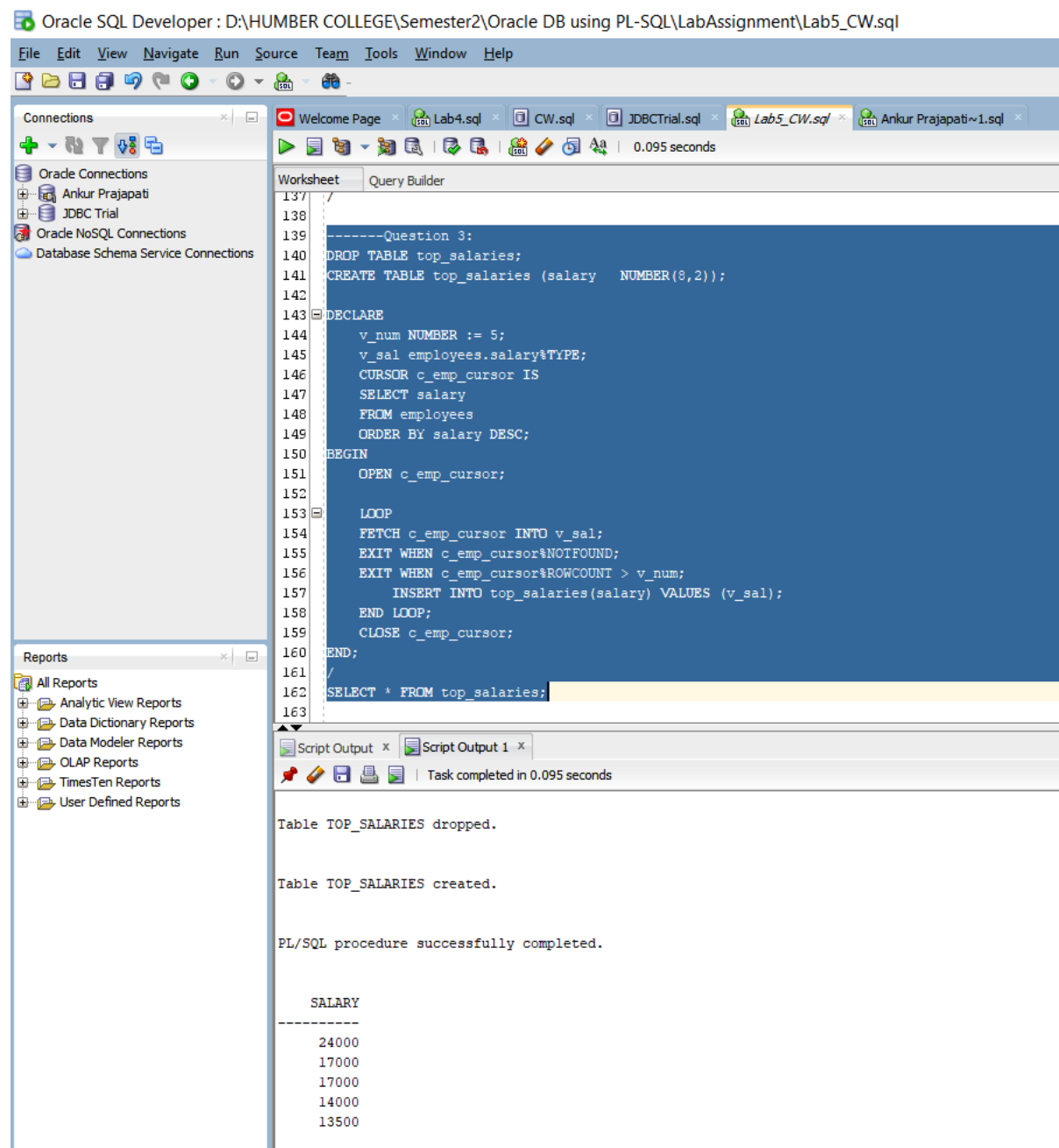
In Executable part I have opened c_dept_cursor and IN LOOP it fetches data from allocated data and if its not found then it will exit. When this cursor is opened, I opened another cursor c_emp_cursor with the parameter v_department_id and it fetches data and print it. If not found it will exit. It close Loop first and after that c_emp_cursor and after that it closes c_dept_cursor.

Output:



```
Oracle SQL Developer : Ankur Prajapati~1
File Edit View Navigate Run Source Team Tools Window Help
Connections
Oracle Connections
Ankur Prajapati
JDBC Trial
Oracle NoSQL Connections
Database Schema Service Connections
Worksheet
Query Builder
1
Script Output
Task completed in 0.037 seconds
Department no: 10 Name: Administration
-----
Department no: 20 Name: Marketing
-----
Department no: 30 Name: Purchasing
Raphaely PU_MAN 07-12-94 11000
Khoo PU_CLERK 18-05-95 3100
Baida PU_CLERK 24-12-97 2900
Tobias PU_CLERK 24-07-97 2800
Himuro PU_CLERK 15-11-98 2600
Colmenares PU_CLERK 10-08-99 2500
-----
Department no: 40 Name: Human Resources
-----
Department no: 50 Name: Shipping
-----
Department no: 60 Name: IT
Hunold IT_PROG 03-01-90 9000
Ernst IT_PROG 21-05-91 6000
Austin IT_PROG 25-06-97 4800
Pataballa IT_PROG 05-02-98 4800
Lorentz IT_PROG 07-02-99 4200
-----
Department no: 70 Name: Public Relations
-----
Department no: 80 Name: Sales
-----
Department no: 90 Name: Executive
King AD_PRES 17-06-87 24000
Kochhar AD_VP 21-09-89 17000
De Haan AD_VP 13-01-93 17000
-----
PL/SQL procedure successfully completed.
```

Practice 2:



Oracle SQL Developer : D:\HUMBER COLLEGE\Semester2\Oracle DB using PL-SQL\LabAssignment\Lab5_CW.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections

- Oracle Connections
 - Ankur Prajapati
 - JDBC Trial
- Oracle NoSQL Connections
- Database Schema Service Connections

Worksheet

```
137 /
138
139 -----Question 3:
140 DROP TABLE top_salaries;
141 CREATE TABLE top_salaries (salary NUMBER(8,2));
142
143 DECLARE
144     v_num NUMBER := 5;
145     v_sal employees.salary%TYPE;
146     CURSOR c_emp_cursor IS
147     SELECT salary
148     FROM employees
149     ORDER BY salary DESC;
150 BEGIN
151     OPEN c_emp_cursor;
152
153     LOOP
154     FETCH c_emp_cursor INTO v_sal;
155     EXIT WHEN c_emp_cursor%NOTFOUND;
156     EXIT WHEN c_emp_cursor%ROWCOUNT > v_num;
157     INSERT INTO top_salaries(salary) VALUES (v_sal);
158     END LOOP;
159     CLOSE c_emp_cursor;
160 END;
161 /
162 SELECT * FROM top_salaries;
163
```

Script Output

Task completed in 0.095 seconds

Table TOP_SALARIES dropped.

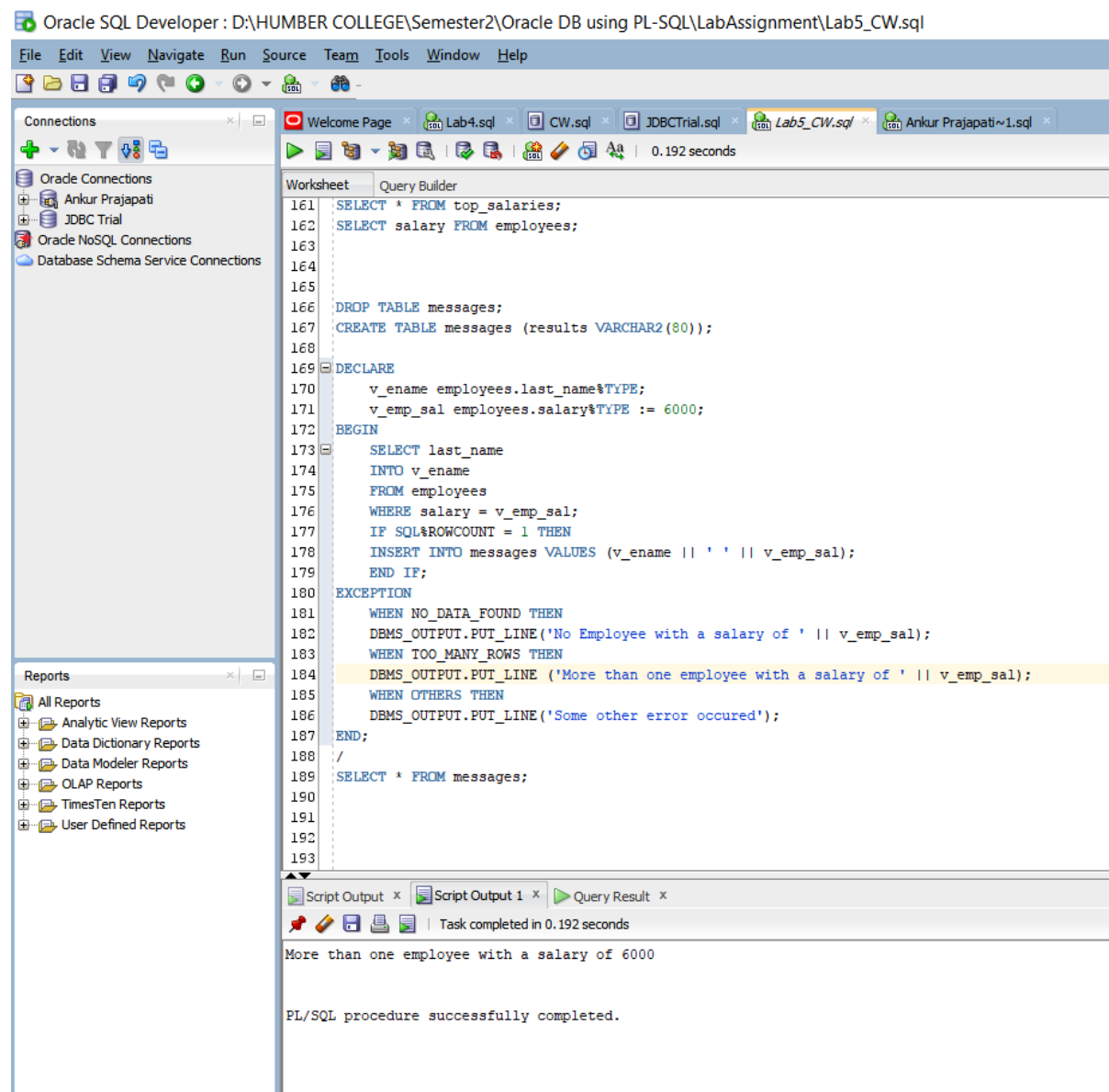
Table TOP_SALARIES created.

PL/SQL procedure successfully completed.

SALARY
24000
17000
17000
14000
13500

Here, I have created top_salaries table and declared cursor named c_emp_cursor which selects rows from employees table and orders it by salary in descending order. In executable section I am opening c_emp_cursor and in LOOP I am fetching it into v_sal variable where it stores the top 5 salaries. If c_emp_cursor is not found it will exit and if the %ROWCOUNT is grater than v_sal it will exit. At the end I closed the c_emp_cursor and used SELECT statement to print the top 5 salaries.

Practice 3:



The screenshot displays the Oracle SQL Developer interface. The main window shows a PL/SQL script in the 'Worksheet' tab. The script performs several operations: it selects data from 'top_salaries' and 'employees' tables, drops and creates a 'messages' table, declares variables 'v_ename' and 'v_emp_sal', and uses a BEGIN block to select data from 'employees' based on a salary condition. It includes an IF statement to check the row count and an INSERT statement to store data in the 'messages' table. An EXCEPTION block handles 'NO_DATA_FOUND', 'TOO_MANY_ROWS', and 'OTHERS' errors. The script concludes with a SELECT statement from the 'messages' table.

```
161 SELECT * FROM top_salaries;
162 SELECT salary FROM employees;
163
164
165
166 DROP TABLE messages;
167 CREATE TABLE messages (results VARCHAR2(80));
168
169 DECLARE
170     v_ename employees.last_name%TYPE;
171     v_emp_sal employees.salary%TYPE := 6000;
172 BEGIN
173     SELECT last_name
174     INTO v_ename
175     FROM employees
176     WHERE salary = v_emp_sal;
177     IF SQL%ROWCOUNT = 1 THEN
178         INSERT INTO messages VALUES (v_ename || ' ' || v_emp_sal);
179     END IF;
180 EXCEPTION
181     WHEN NO_DATA_FOUND THEN
182         DBMS_OUTPUT.PUT_LINE('No Employee with a salary of ' || v_emp_sal);
183     WHEN TOO_MANY_ROWS THEN
184         DBMS_OUTPUT.PUT_LINE('More than one employee with a salary of ' || v_emp_sal);
185     WHEN OTHERS THEN
186         DBMS_OUTPUT.PUT_LINE('Some other error occurred');
187 END;
188 /
189 SELECT * FROM messages;
190
191
192
193
```

The 'Script Output' window at the bottom shows the execution results. It indicates that the task was completed in 0.192 seconds and displays the message: 'More than one employee with a salary of 6000'. Below this, it states 'PL/SQL procedure successfully completed.'

In declaration section v_emp_salary of employees.salary type is declared and initialized with 6000. Where in executable section it selects the row as per select statement and IF SQL%ROWCOUNT is 1 it will insert the selected data into message table.

In Exception section, there are three exceptions if the data is not found it will print the message stored in NO_DATA_FOUND exception. If there are many rows then it will print the message stored in exception called TOO_MANY_ROWS. Suppose, anything else than these exceptions occurs it will show the exception message stored in OTHERS.

Note that here NO_DATA_FOUND and TOO_MANY_ROWS are predefined exceptions of Oracle.

Oracle SQL Developer : D:\HUMBER COLLEGE\Semester2\Oracle DB using PL-SQL\LabAssignment\Lab5_CW.sql

File Edit View Navigate Run Source Team Tools Window Help

Connections

- Oracle Connections
 - Ankur Prajapati
 - JDBC Trial
- Oracle NoSQL Connections
- Database Schema Service Connections

Worksheet Query Builder

```
161 SELECT * FROM top_salaries;
162 SELECT salary FROM employees;
163
164
165
166 DROP TABLE messages;
167 CREATE TABLE messages (results VARCHAR2(80));
168
169 DECLARE
170     v_ename employees.last_name%TYPE;
171     v_emp_sal employees.salary%TYPE := 2000;
172 BEGIN
173     SELECT last_name
174     INTO v_ename
175     FROM employees
176     WHERE salary = v_emp_sal;
177     IF SQL%ROWCOUNT = 1 THEN
178         INSERT INTO messages VALUES (v_ename || ' ' || v_emp_sal);
179     END IF;
180 EXCEPTION
181     WHEN NO_DATA_FOUND THEN
182         DBMS_OUTPUT.PUT_LINE('No Employee with a salary of ' || v_emp_sal);
183     WHEN TOO_MANY_ROWS THEN
184         DBMS_OUTPUT.PUT_LINE('More than one employee with a salary of ' || v_emp_sal);
185     WHEN OTHERS THEN
186         DBMS_OUTPUT.PUT_LINE('Some other error occurred');
187 END;
188 /
189 SELECT * FROM messages;
190
191
192
193
```

Script Output x Script Output 1 x Query Result x

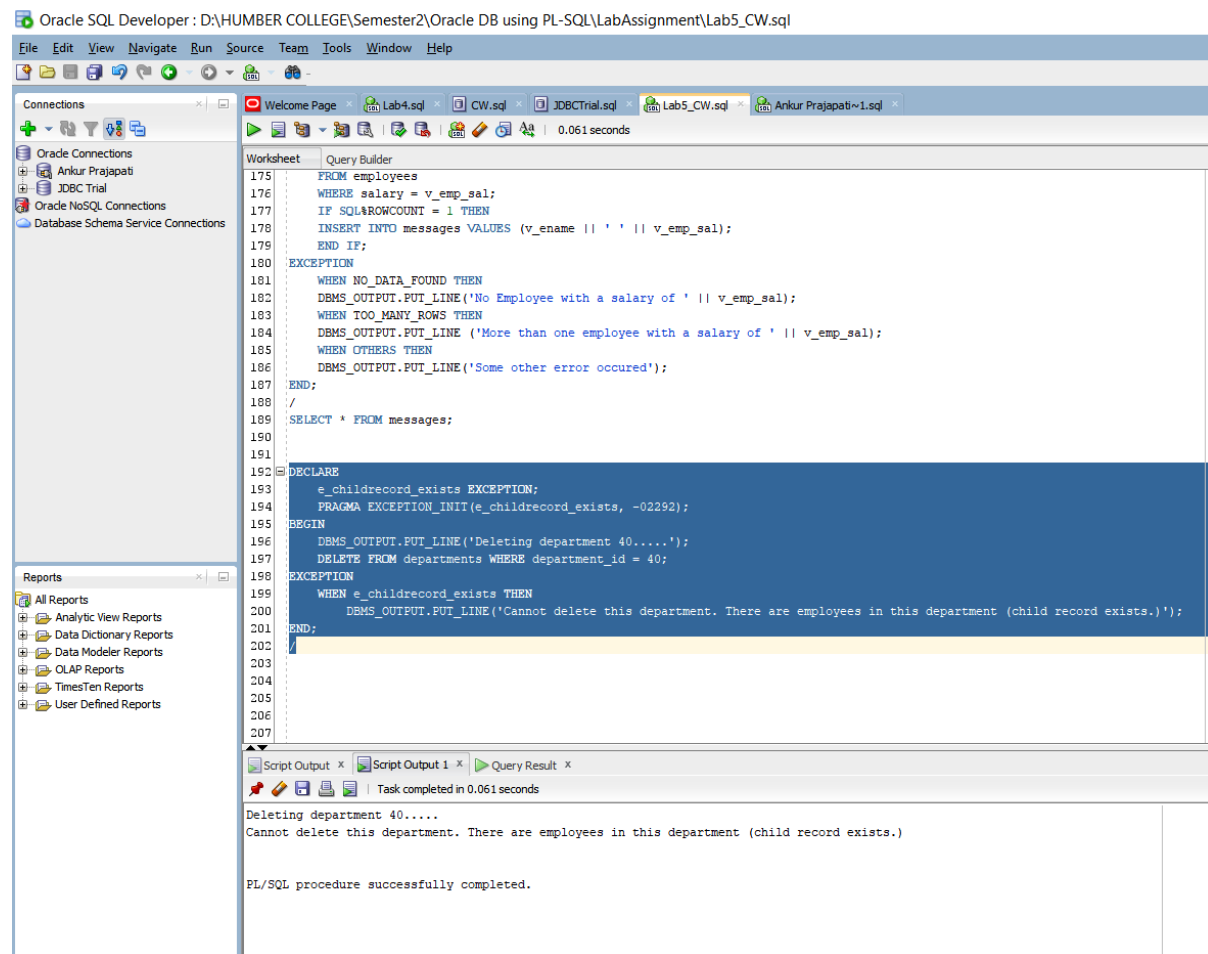
Task completed in 0.055 seconds

No Employee with a salary of 2000

PL/SQL procedure successfully completed.

In here it prints message stored in exception NO_DATA_FOUND. Because there are no data with the salary 2000.

Practice 4:



The screenshot displays the Oracle SQL Developer interface. The main window shows a PL/SQL script with the following code:

```
175 FROM employees
176 WHERE salary = v_emp_sal;
177 IF SQL%ROWCOUNT = 1 THEN
178 INSERT INTO messages VALUES (v_emp_name || ' ' || v_emp_sal);
179 END IF;
180 EXCEPTION
181 WHEN NO_DATA_FOUND THEN
182 DBMS_OUTPUT.PUT_LINE('No Employee with a salary of ' || v_emp_sal);
183 WHEN TOO_MANY_ROWS THEN
184 DBMS_OUTPUT.PUT_LINE('More than one employee with a salary of ' || v_emp_sal);
185 WHEN OTHERS THEN
186 DBMS_OUTPUT.PUT_LINE('Some other error occurred');
187 END;
188 /
189 SELECT * FROM messages;
190
191
192 DECLARE
193 e_childrecord_exists EXCEPTION;
194 PRAGMA EXCEPTION_INIT(e_childrecord_exists, -02292);
195 BEGIN
196 DBMS_OUTPUT.PUT_LINE('Deleting department 40.....');
197 DELETE FROM departments WHERE department_id = 40;
198 EXCEPTION
199 WHEN e_childrecord_exists THEN
200 DBMS_OUTPUT.PUT_LINE('Cannot delete this department. There are employees in this department (child record exists.)');
201 END;
202 /
203
204
205
206
207
```

The script is executed, and the output is shown in the Script Output window:

```
Task completed in 0.061 seconds

Deleting department 40.....
Cannot delete this department. There are employees in this department (child record exists.)

PL/SQL procedure successfully completed.
```

Here in declaration section I have declared `e_childrecord_exists` as an Exception. I used `PRAGMA` to link the declared exception with error code `-2292`. In executable section I am trying to delete data from departments table where department id is 40. In Exception section I am printing the output message associated with that exception.