

Project3 Report

Ruojia Zhang

Figure 1

Executing yaml file to create the required stacks. Filled with my initials -> roger-project3 there are blanks that requires filling in the beginning.

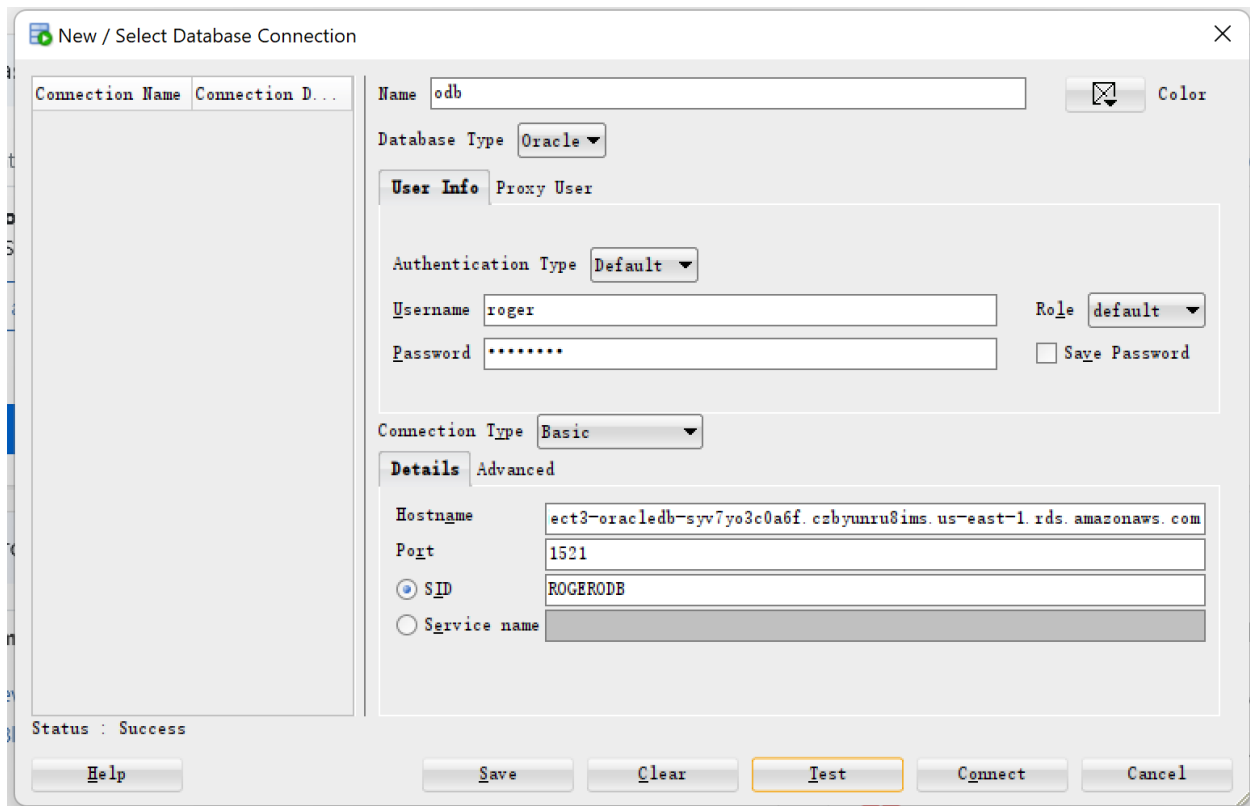
The screenshot shows the AWS CloudFormation console interface. On the left, a sidebar lists two stacks: 'Roger-project3' (status: CREATE_IN_PROGRESS) and 'c66759a130077313227762t1w062690179031' (status: CREATE_COMPLETE). The main panel displays the details for 'Roger-project3', including tabs for Stack info, Events, Resources, Outputs, Parameters, Template, and Change sets. The 'Events' tab is active, showing a single event with the status 'CREATE_IN_PROGRESS' and the reason 'User Initiated'.

Timestamp	Logical ID	Status	Status reason
2022-12-01 18:26:57 UTC-0500	Roger-project3	CREATE_IN_PROGRESS	User Initiated

Note: Run several file create, policy operation and finished the creating process. You can see the stack's name is roger-project3 by me.

Figure2

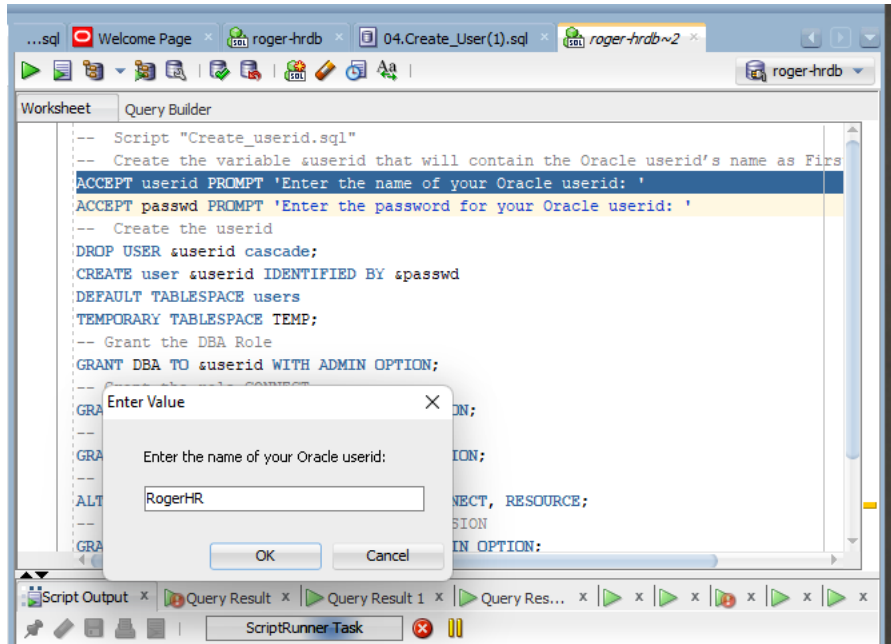
Connecting to oracle sql database using the sql developer as is suggested in the project3 task.



Note: this is the connection panel of sql developer. As all information need are inserted and you can see the test Status: Success in the left bottom. Indicating that I will be able to connect to this db at anytime.

Figure3

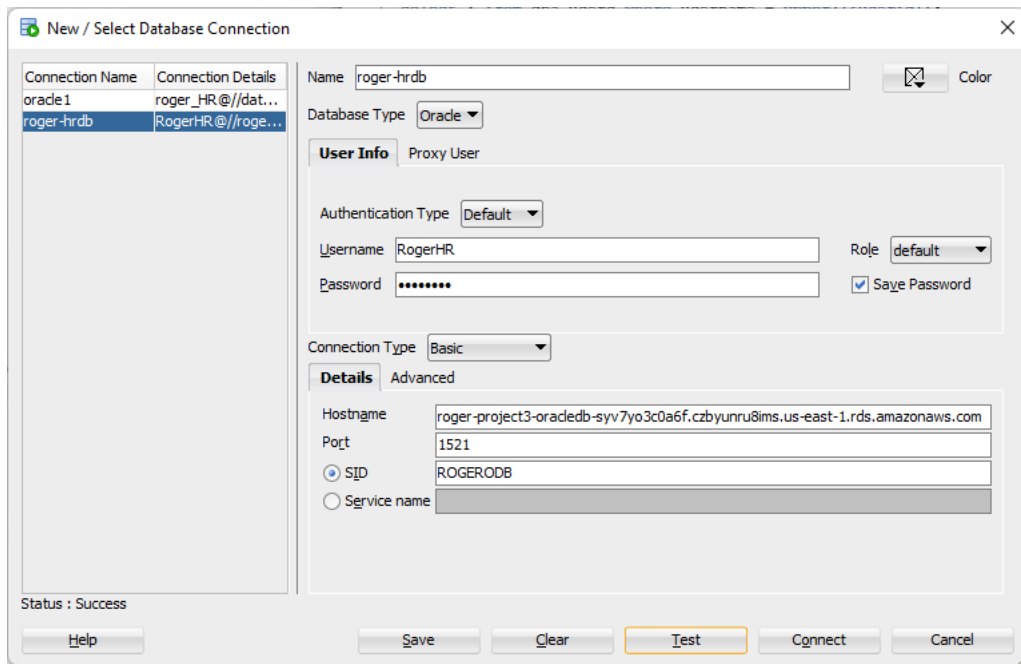
Connecting to oracle sql database using the sql developer and create user



Note: this screenshot shows that I'm creating the RogerHR user using the create user sql

Figure4

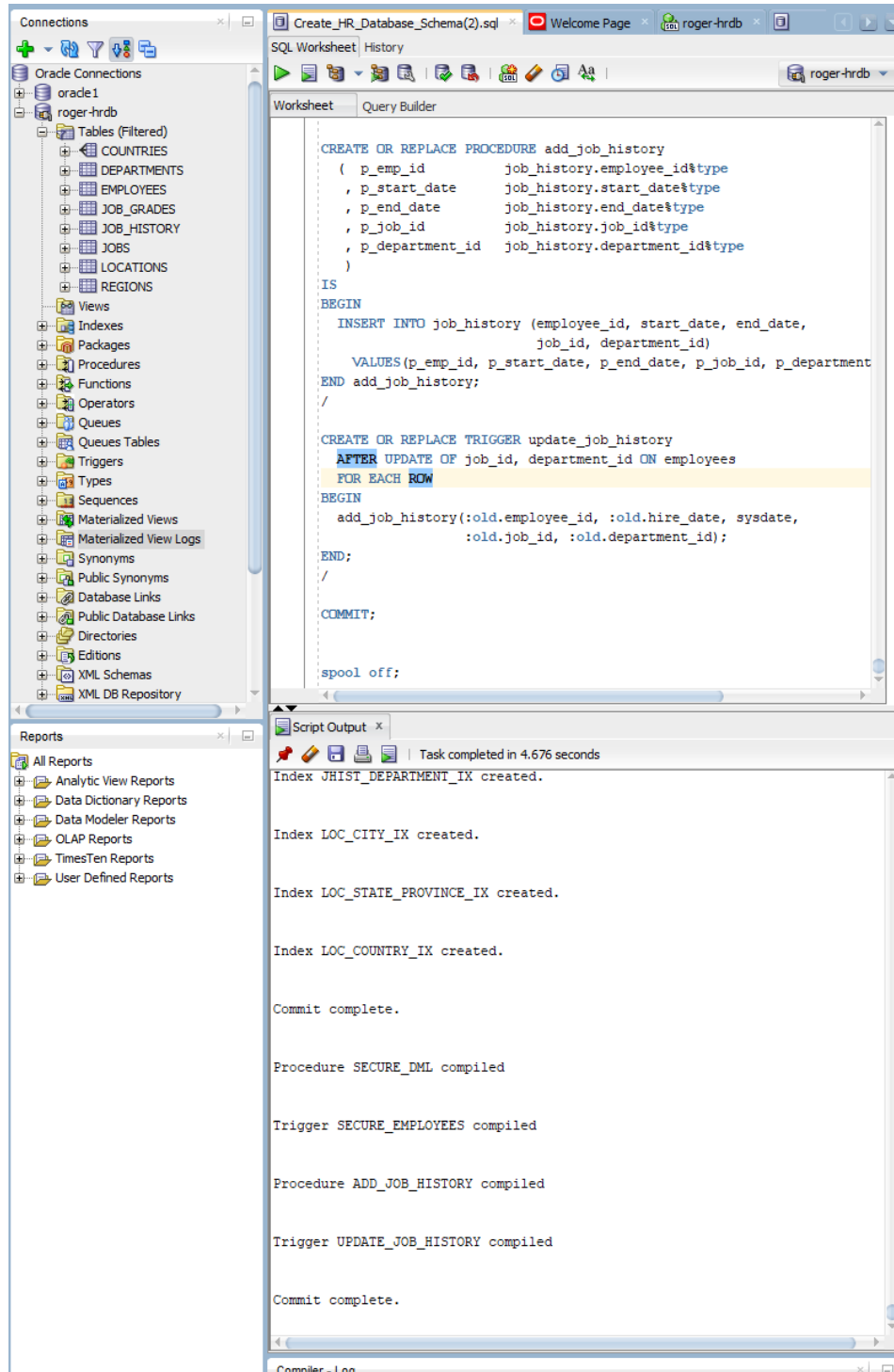
Connecting to oracle sql database using the sql developer with new user



Note: this screenshot shows that I'm reconnecting to db using the RogerHR user that just created

Figure5

Connecting to oracle sql database and run create_hr_database_schema.sql

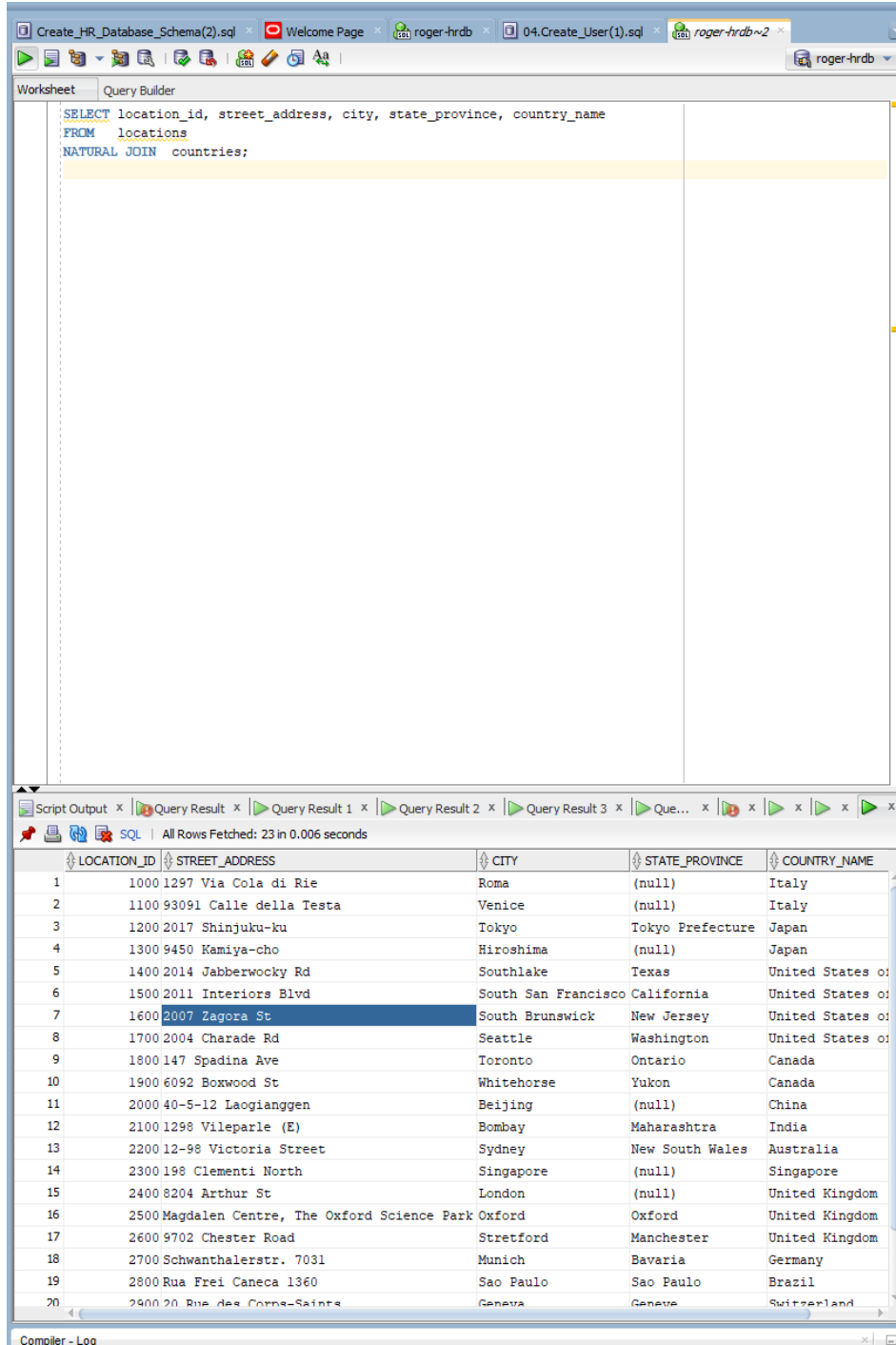


Note: this screenshot shows that I've done creating the tables that need for the project.

Oracle Practice 7

Figure 6

The SQL Developer output of oracle sql practice 7-1



The screenshot shows the Oracle SQL Developer interface. The top toolbar includes icons for running queries, saving, and other standard database operations. The main window is divided into a 'Worksheet' area at the top and a 'Query Result' area at the bottom. The 'Worksheet' area contains the following SQL query:

```
SELECT location_id, street_address, city, state_province, country_name
FROM locations
NATURAL JOIN countries;
```

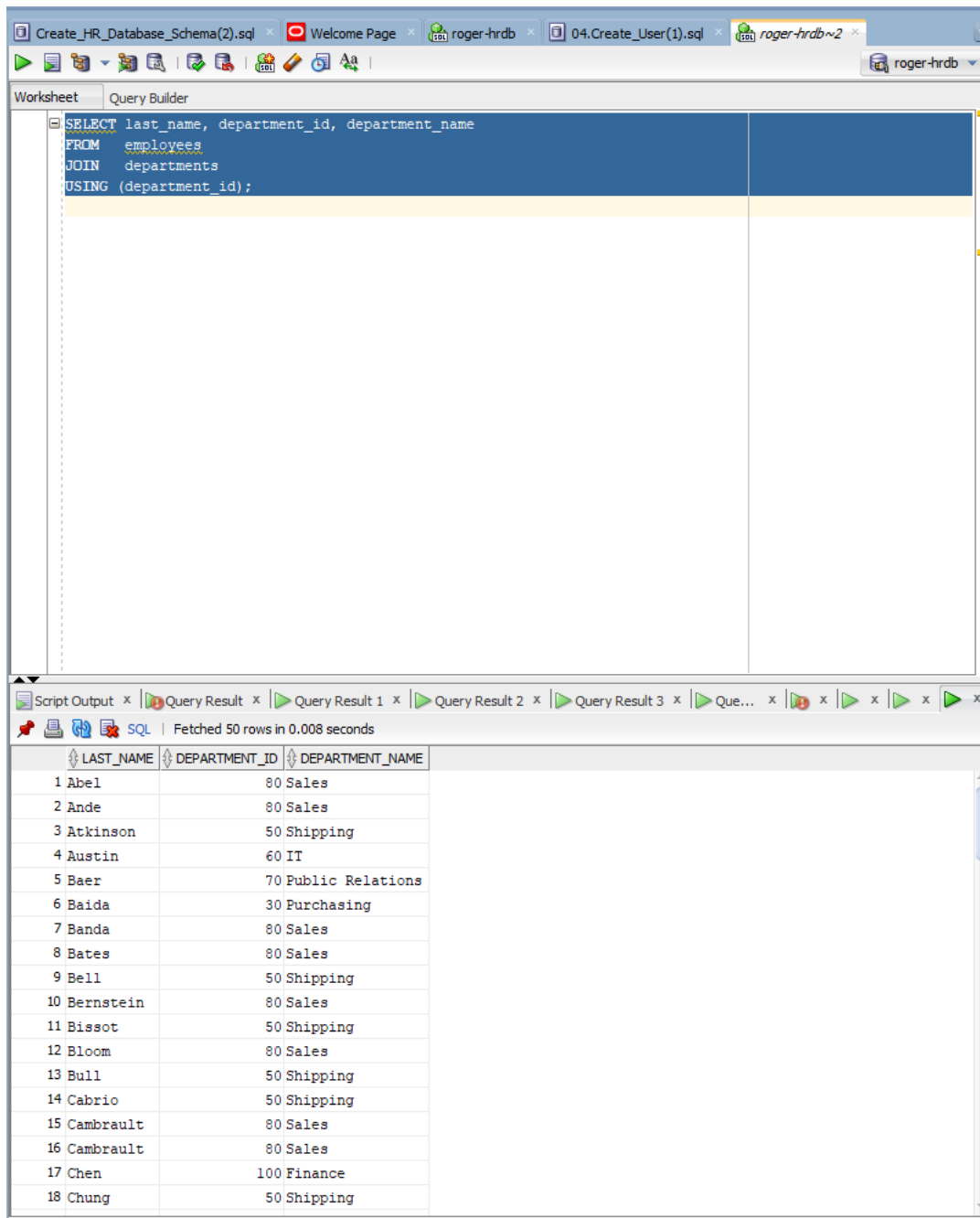
The 'Query Result' area displays the output of the query as a table with 20 rows. The columns are labeled LOCATION_ID, STREET_ADDRESS, CITY, STATE_PROVINCE, and COUNTRY_NAME. The data includes various locations from different countries, such as Italy, Japan, United States, Canada, China, India, Australia, Singapore, United Kingdom, Germany, Brazil, and Switzerland.

LOCATION_ID	STREET_ADDRESS	CITY	STATE_PROVINCE	COUNTRY_NAME
1	1000 1297 Via Cola di Rie	Roma	(null)	Italy
2	1100 93091 Calle della Testa	Venice	(null)	Italy
3	1200 2017 Shinjuku-ku	Tokyo	Tokyo Prefecture	Japan
4	1300 9450 Kamiya-cho	Hiroshima	(null)	Japan
5	1400 2014 Jabberwocky Rd	Southlake	Texas	United States of America
6	1500 2011 Interiors Blvd	South San Francisco	California	United States of America
7	1600 2007 Zagora St	South Brunswick	New Jersey	United States of America
8	1700 2004 Charade Rd	Seattle	Washington	United States of America
9	1800 147 Spadina Ave	Toronto	Ontario	Canada
10	1900 6092 Boxwood St	Whitehorse	Yukon	Canada
11	2000 40-5-12 Laogianggen	Beijing	(null)	China
12	2100 1298 Vileparle (E)	Bombay	Maharashtra	India
13	2200 12-98 Victoria Street	Sydney	New South Wales	Australia
14	2300 198 Clementi North	Singapore	(null)	Singapore
15	2400 8204 Arthur St	London	(null)	United Kingdom
16	2500 Magdalen Centre, The Oxford Science Park	Oxford	Oxford	United Kingdom
17	2600 9702 Chester Road	Stretford	Manchester	United Kingdom
18	2700 Schwanthalerstr. 7031	Munich	Bavaria	Germany
19	2800 Rua Frei Caneca 1360	Sao Paulo	Sao Paulo	Brazil
20	2900 20 Rue des Corps-Saints	Geneva	Geneve	Switzerland

Note: this screenshot shows that I've output all fields required for the practice by using join countries

Figure7

The SQL Developer output of oracle sql practice 7-2



The screenshot displays the Oracle SQL Developer interface. The top toolbar includes icons for running queries, saving, and other standard database operations. The main window is divided into two panes: the left pane shows the SQL query, and the right pane shows the query results.

The SQL query in the left pane is:

```
SELECT last_name, department_id, department_name
FROM employees
JOIN departments
USING (department_id);
```

The right pane shows the query results, which are displayed in a table format. The table has three columns: LAST_NAME, DEPARTMENT_ID, and DEPARTMENT_NAME. The results show 18 rows of data, representing employees and their respective departments.

	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
1	Abel	80	Sales
2	Ande	80	Sales
3	Atkinson	50	Shipping
4	Austin	60	IT
5	Baer	70	Public Relations
6	Baida	30	Purchasing
7	Banda	80	Sales
8	Bates	80	Sales
9	Bell	50	Shipping
10	Bernstein	80	Sales
11	Bissot	50	Shipping
12	Bloom	80	Sales
13	Bull	50	Shipping
14	Cabrio	50	Shipping
15	Cambrault	80	Sales
16	Cambrault	80	Sales
17	Chen	100	Finance
18	Chung	50	Shipping

Note: this screenshot shows that I've output all fields required for the practice by using join departments

Figure8

The SQL Developer output of oracle sql practice 7-3

The screenshot displays the Oracle SQL Developer interface. The top pane shows a SQL query in the 'Query Builder' tab. The query is as follows:

```
SELECT e.last_name, e.job_id, e.department_id, d.department_name
FROM   employees e JOIN departments d
ON     (e.department_id = d.department_id)
JOIN   locations l
ON     (d.location_id = l.location_id)
WHERE  LOWER(l.city) = 'toronto';
```

The bottom pane shows the 'Query Result' tab with the following data:

	LAST_NAME	JOB_ID	DEPARTMENT_ID	DEPARTMENT_NAME
1	Hartstein	MK_MAN	20	Marketing
2	Fay	MK_REP	20	Marketing

The status bar indicates 'All Rows Fetched: 2 in 0.006 seconds'.

Note: this screenshot shows that I've output all fields required for the practice by using join location

Figure9

The SQL Developer output of oracle sql practice 7-4

The screenshot displays the Oracle SQL Developer interface. The main window shows a SQL query in the Worksheet tab:

```
SELECT w.last_name "Employee", w.employee_id "EMP#",  
       m.last_name "Manager", m.employee_id "Mgr#"   
FROM   employees w join employees m   
ON     (w.manager_id = m.employee_id);
```

Below the query, the Query Result tab shows the output of the query. The results are displayed in a table with 4 columns: Employee, EMP#, Manager, and Mgr#. The table contains 18 rows of data, representing the hierarchy of employees and their managers.

Employee	EMP#	Manager	Mgr#
1 Kochhar	101	King	100
2 De Haan	102	King	100
3 Hunold	103	De Haan	102
4 Ernst	104	Hunold	103
5 Austin	105	Hunold	103
6 Pataballa	106	Hunold	103
7 Lorentz	107	Hunold	103
8 Greenberg	108	Kochhar	101
9 Faviet	109	Greenberg	108
10 Chen	110	Greenberg	108
11 Sciarra	111	Greenberg	108
12 Urman	112	Greenberg	108
13 Popp	113	Greenberg	108
14 Raphaely	114	King	100
15 Khoo	115	Raphaely	114
16 Baida	116	Raphaely	114
17 Tobias	117	Raphaely	114
18 Himuro	118	Raphaely	114

Note: this screenshot shows that I've output all fields required for the practice by using self join

Figure10

The SQL Developer output of oracle sql practice 7-5

The screenshot displays the Oracle SQL Developer interface. The top pane shows a SQL query in the 'Query Builder' tab. The query is a self-join on the 'employees' table, selecting employee and manager details, ordered by the employee ID.

```
SELECT w.last_name "Employee", w.employee_id "EMP#",  
       m.last_name "Manager", m.employee_id "Mgr#"   
FROM   employees w  
LEFT OUTER JOIN employees m  
ON     (w.manager_id = m.employee_id)  
ORDER BY 2;
```

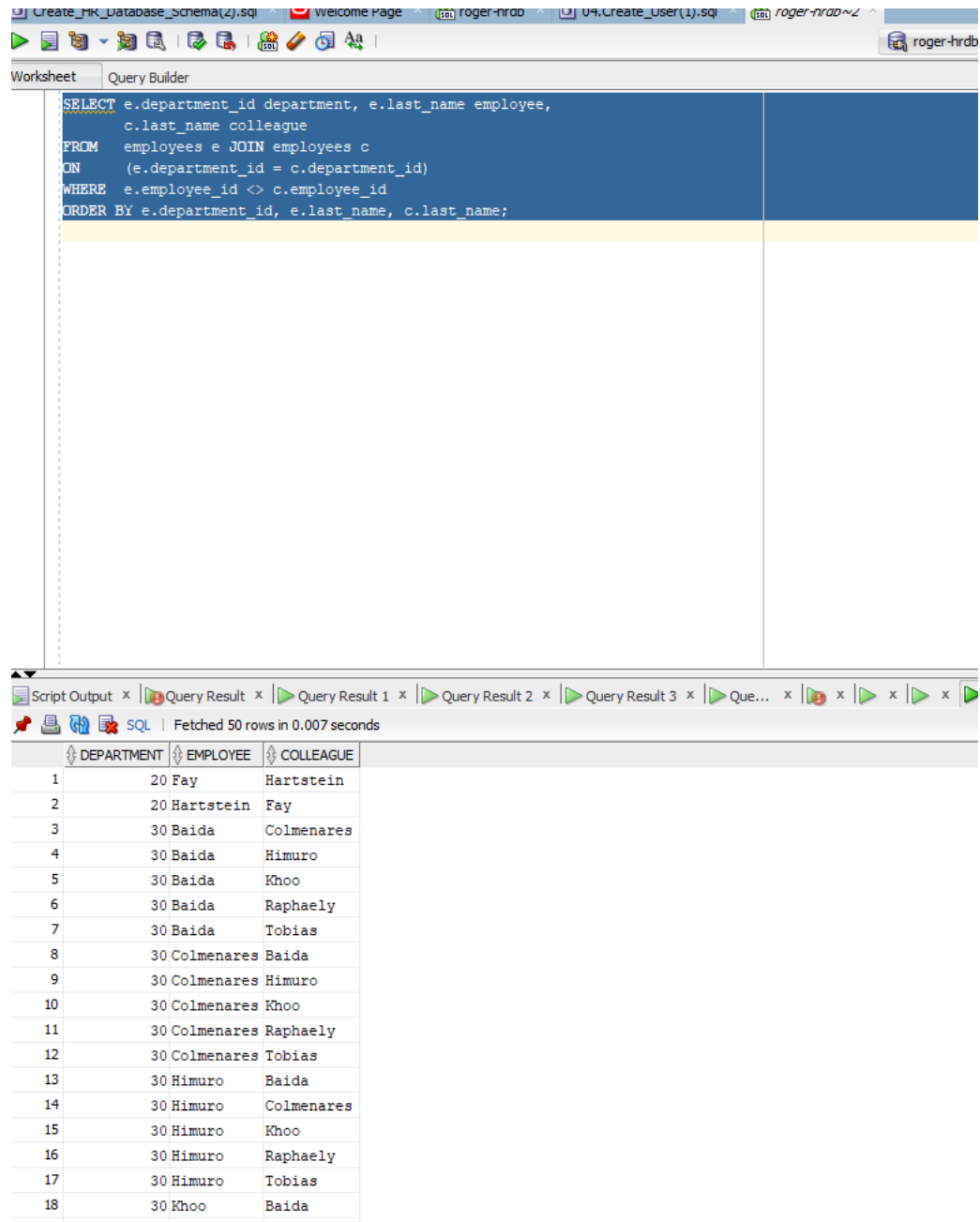
The bottom pane shows the 'Query Result' tab with 50 rows fetched in 0.007 seconds. The results are displayed in a table with four columns: Employee, EMP#, Manager, and Mgr#.

	Employee	EMP#	Manager	Mgr#
1	King	100	(null)	(null)
2	Kochhar	101	King	100
3	De Haan	102	King	100
4	Hunold	103	De Haan	102
5	Ernst	104	Hunold	103
6	Austin	105	Hunold	103
7	Pataballa	106	Hunold	103
8	Lorentz	107	Hunold	103
9	Greenberg	108	Kochhar	101
10	Faviet	109	Greenberg	108
11	Chen	110	Greenberg	108
12	Sciarra	111	Greenberg	108
13	Urman	112	Greenberg	108
14	Popp	113	Greenberg	108
15	Raphaely	114	King	100
16	Khoo	115	Raphaely	114
17	Baida	116	Raphaely	114
18	Tobias	117	Raphaely	114

Note: this screenshot shows that I've output all fields required for the practice by using self join and order by second column

Figure11

The SQL Developer output of oracle sql practice 7-6



The screenshot displays the Oracle SQL Developer interface. The top pane shows a SQL query in the 'Query Builder' tab. The query is a self-join on the 'employees' table, selecting department and last names for both the employee and their colleague. The bottom pane shows the 'Query Result' tab with 50 rows of data, sorted by department ID, employee last name, and colleague last name.

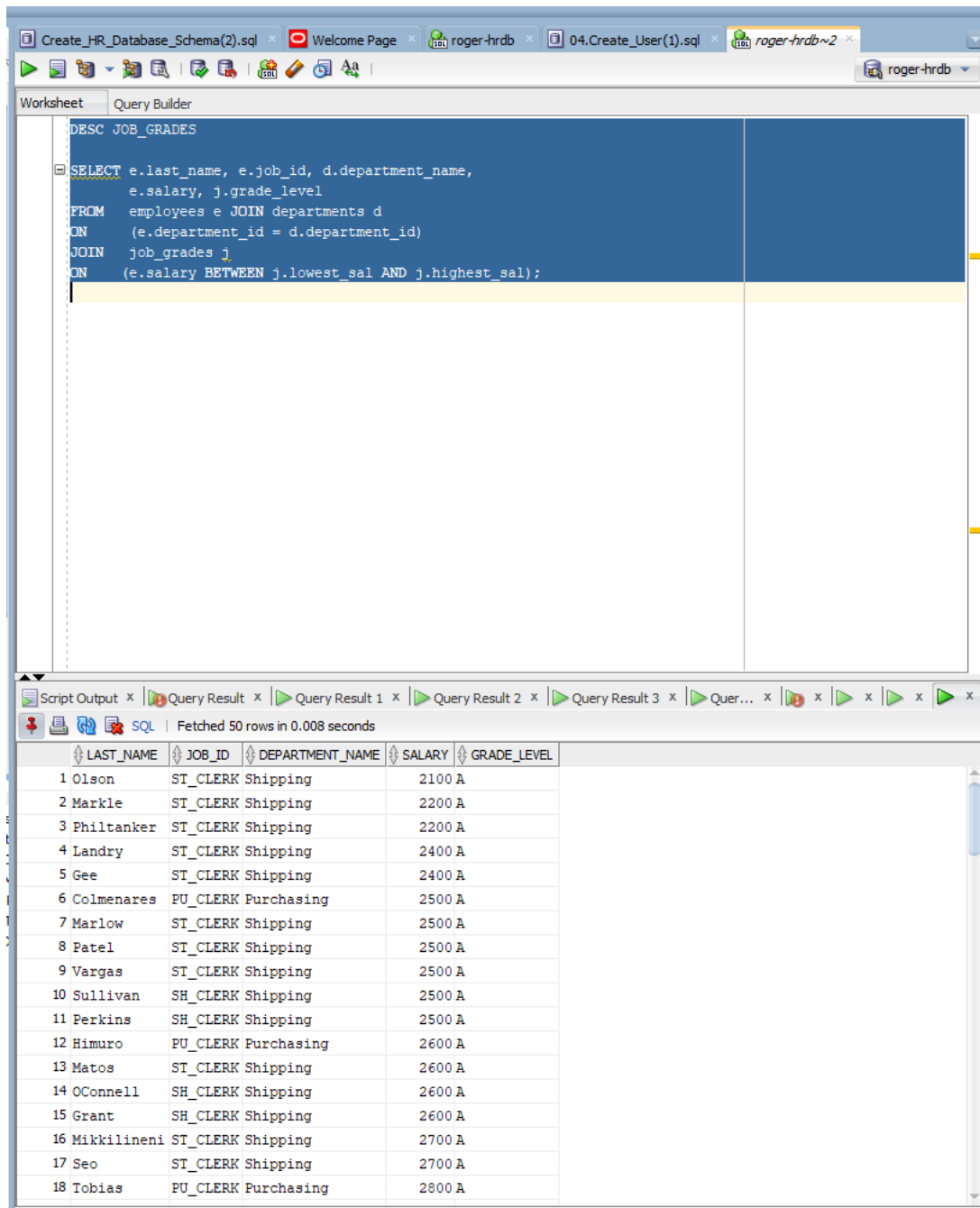
```
SELECT e.department_id department, e.last_name employee,  
       c.last_name colleague  
FROM   employees e JOIN employees c  
ON      (e.department_id = c.department_id)  
WHERE  e.employee_id <> c.employee_id  
ORDER BY e.department_id, e.last_name, c.last_name;
```

	DEPARTMENT	EMPLOYEE	COLLEAGUE
1	20	Fay	Hartstein
2	20	Hartstein	Fay
3	30	Baida	Colmenares
4	30	Baida	Himuro
5	30	Baida	Khoo
6	30	Baida	Raphaely
7	30	Baida	Tobias
8	30	Colmenares	Baida
9	30	Colmenares	Himuro
10	30	Colmenares	Khoo
11	30	Colmenares	Raphaely
12	30	Colmenares	Tobias
13	30	Himuro	Baida
14	30	Himuro	Colmenares
15	30	Himuro	Khoo
16	30	Himuro	Raphaely
17	30	Himuro	Tobias
18	30	Khoo	Baida

Note: this screenshot shows that I've output all fields required for the practice by using self join and order by e.department_id, e.last_name, c.last_name

Figure12

The SQL Developer output of oracle sql practice 7-7



The screenshot displays the Oracle SQL Developer interface. The top pane shows a SQL query in the 'Query Builder' tab. The query is a SELECT statement that joins the 'employees' table with the 'departments' and 'job_grades' tables, filtering for employees whose salary falls within a specific grade's range. The bottom pane shows the 'Query Result' tab, which displays 50 rows of data fetched in 0.008 seconds. The data is presented in a table with five columns: LAST_NAME, JOB_ID, DEPARTMENT_NAME, SALARY, and GRADE_LEVEL. The results show employees from various departments like Shipping, Purchasing, and HR, with their respective job titles, salaries, and grade levels.

	LAST_NAME	JOB_ID	DEPARTMENT_NAME	SALARY	GRADE_LEVEL
1	Olson	ST_CLERK	Shipping	2100	A
2	Markle	ST_CLERK	Shipping	2200	A
3	Philtanker	ST_CLERK	Shipping	2200	A
4	Landry	ST_CLERK	Shipping	2400	A
5	Gee	ST_CLERK	Shipping	2400	A
6	Colmenares	PU_CLERK	Purchasing	2500	A
7	Marlow	ST_CLERK	Shipping	2500	A
8	Patel	ST_CLERK	Shipping	2500	A
9	Vargas	ST_CLERK	Shipping	2500	A
10	Sullivan	SH_CLERK	Shipping	2500	A
11	Perkins	SH_CLERK	Shipping	2500	A
12	Himuro	PU_CLERK	Purchasing	2600	A
13	Matos	ST_CLERK	Shipping	2600	A
14	OConnell	SH_CLERK	Shipping	2600	A
15	Grant	SH_CLERK	Shipping	2600	A
16	Mikkilineni	ST_CLERK	Shipping	2700	A
17	Seo	ST_CLERK	Shipping	2700	A
18	Tobias	PU_CLERK	Purchasing	2800	A

Note: this screenshot shows that I've output all fields required for the practice by filtering salary in between

Figure13

The SQL Developer output of oracle sql practice 7-8

The screenshot displays the Oracle SQL Developer interface. The main window shows a SQL query in the Query Builder:

```
SELECT e.last_name, e.hire_date
FROM   employees e JOIN employees davies
ON     (davies.last_name = 'Davies')
WHERE  davies.hire_date < e.hire_date;
```

Below the query, the results are displayed in a table with columns LAST_NAME and HIRE_DATE. The table contains 18 rows of data, numbered 1 through 18.

	LAST_NAME	HIRE_DATE
1	Austin	25-JUN-97
2	Pataballa	05-FEB-98
3	Lorentz	07-FEB-99
4	Chen	28-SEP-97
5	Sciarra	30-SEP-97
6	Urman	07-MAR-98
7	Popp	07-DEC-99
8	Baida	24-DEC-97
9	Tobias	24-JUL-97
10	Himuro	15-NOV-98
11	Colmenares	10-AUG-99
12	Fripp	10-APR-97
13	Vollman	10-OCT-97
14	Mourgos	16-NOV-99
15	Nayer	16-JUL-97
16	Mikkilineni	28-SEP-98
17	Landry	14-JAN-99
18	Markle	08-MAR-00

Note: this screenshot shows that I've output all fields required for the practice by filtering hire_data

Figure14

The SQL Developer output of oracle sql practice 7-9

The screenshot displays the Oracle SQL Developer interface. The main window shows a SQL query in the Worksheet tab:

```
SELECT w.last_name, w.hire_date, m.last_name, m.hire_date
FROM employees w JOIN employees m
ON (w.manager_id = m.employee_id)
WHERE w.hire_date < m.hire_date;
```

Below the query, the Query Result tab shows the output of the query. The results are displayed in a table with four columns: LAST_NAME, HIRE_DATE, LAST_NAME_1, and HIRE_DATE_1. The table contains 18 rows of data, representing employees who were hired before their manager.

	LAST_NAME	HIRE_DATE	LAST_NAME_1	HIRE_DATE_1
1	Hunold	03-JAN-90	De Haan	13-JAN-93
2	Faviet	16-AUG-94	Greenberg	17-AUG-94
3	Marlow	16-FEB-97	Fripp	10-APR-97
4	Ladwig	14-JUL-95	Vollman	10-OCT-97
5	Rajs	17-OCT-95	Mourgos	16-NOV-99
6	Davies	29-JAN-97	Mourgos	16-NOV-99
7	Matos	15-MAR-98	Mourgos	16-NOV-99
8	Vargas	09-JUL-98	Mourgos	16-NOV-99
9	King	30-JAN-96	Partners	05-JAN-97
10	Sully	04-MAR-96	Partners	05-JAN-97
11	McEwen	01-AUG-96	Partners	05-JAN-97
12	Ozer	11-MAR-97	Cambrault	15-OCT-99
13	Bloom	23-MAR-98	Cambrault	15-OCT-99
14	Fox	24-JAN-98	Cambrault	15-OCT-99
15	Smith	23-FEB-99	Cambrault	15-OCT-99
16	Bates	24-MAR-99	Cambrault	15-OCT-99
17	Abel	11-MAY-96	Zlotkey	29-JAN-00
18	Hutton	19-MAR-97	Zlotkey	29-JAN-00

Note: this screenshot shows that I've output all fields required for the practice by filtering employees that are hired before their manager.

Oracle Practice 8

Figure15

The SQL Developer output of oracle sql practice 8-1

The screenshot displays the Oracle SQL Developer interface. The top pane shows a SQL query in the 'Query Builder' tab. The query is as follows:

```
UNDEFINE Enter_name  
  
SELECT last_name, hire_date  
FROM employees  
WHERE department_id = (SELECT department_id  
                       FROM employees  
                       WHERE last_name = '&Enter_name')  
AND last_name <> '&Enter_name';
```

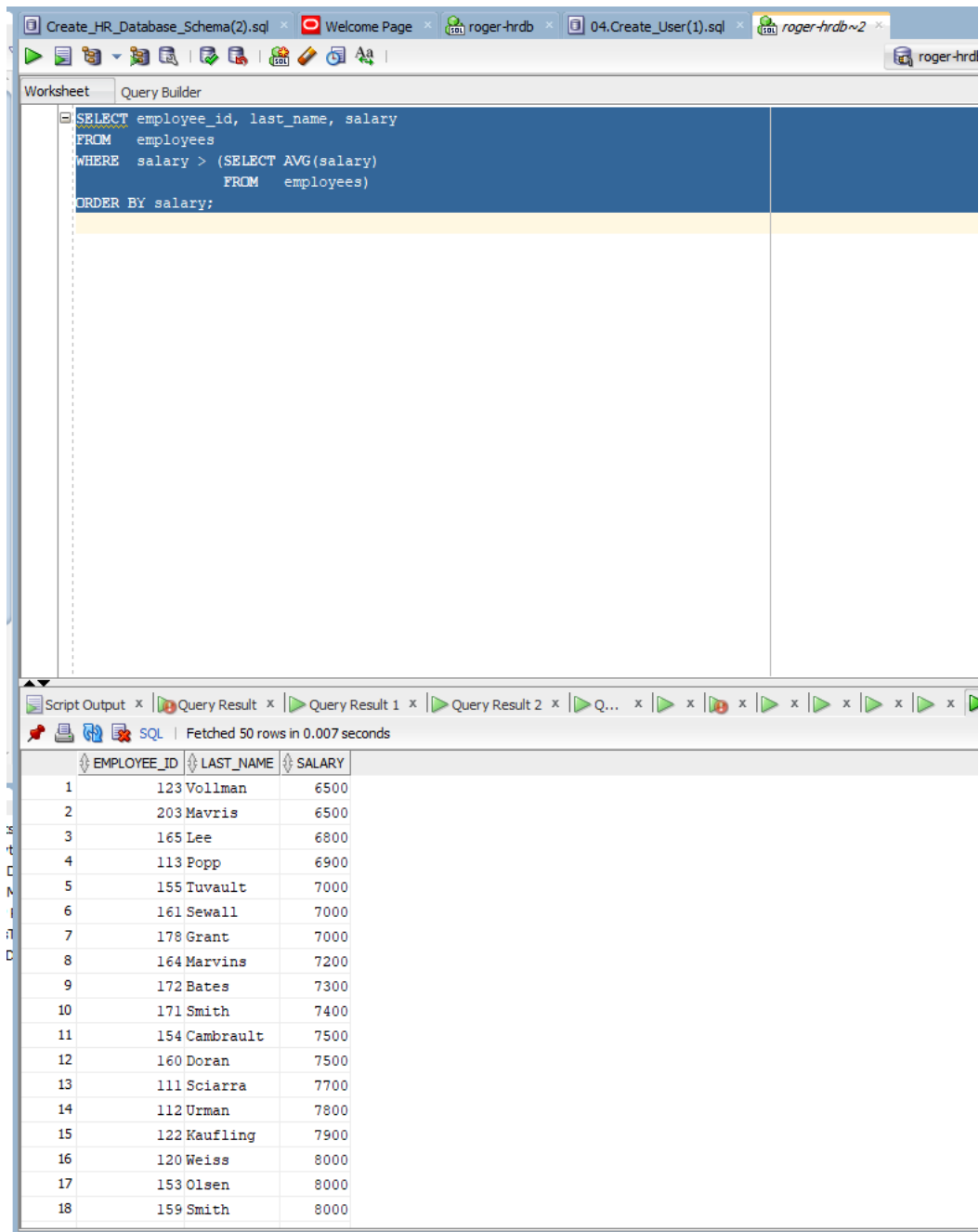
The bottom pane shows the 'Query Result' tab with the following data:

	LAST_NAME	HIRE_DATE
1	Russell	01-OCT-96
2	Partners	05-JAN-97
3	Errazuriz	10-MAR-97
4	Cambrault	15-OCT-99
5	Tucker	30-JAN-97
6	Bernstein	24-MAR-97
7	Hall	20-AUG-97
8	Olsen	30-MAR-98
9	Cambrault	09-DEC-98
10	Tuvault	23-NOV-99
11	King	30-JAN-96
12	Sully	04-MAR-96
13	McEwen	01-AUG-96
14	Smith	10-MAR-97
15	Doran	15-DEC-97
16	Sewall	03-NOV-98
17	Vishney	11-NOV-97
18	Greene	19-MAR-99

Note: this screenshot shows that I've output all fields required for the practice by using filtering all employee work with zlotkey

Figure16

The SQL Developer output of oracle sql practice 8-2



The screenshot displays the Oracle SQL Developer interface. The top toolbar includes icons for running queries, saving, and other standard database operations. The main window is divided into two panes: the left pane shows the SQL script, and the right pane shows the query results. The SQL script is as follows:

```
SELECT employee_id, last_name, salary
FROM employees
WHERE salary > (SELECT AVG(salary)
                FROM employees)
ORDER BY salary;
```

The query results pane shows a table with 50 rows (only the first 18 are visible in the screenshot). The table has three columns: EMPLOYEE_ID, LAST_NAME, and SALARY. The data is sorted by salary in ascending order.

EMPLOYEE_ID	LAST_NAME	SALARY
1	123 Vollman	6500
2	203 Mavris	6500
3	165 Lee	6800
4	113 Popp	6900
5	155 Tuvault	7000
6	161 Sewall	7000
7	178 Grant	7000
8	164 Marvins	7200
9	172 Bates	7300
10	171 Smith	7400
11	154 Cambrault	7500
12	160 Doran	7500
13	111 Sciarra	7700
14	112 Urman	7800
15	122 Kaufling	7900
16	120 Weiss	8000
17	153 Olsen	8000
18	159 Smith	8000

Note: this screenshot shows that I've output all fields required for the practice by using filtering all employee order by salary

Figure17

The SQL Developer output of oracle sql practice 8-3

The screenshot displays the Oracle SQL Developer interface. The top pane shows a SQL query in the 'Query Builder' tab. The query is as follows:

```
SELECT employee_id, last_name
FROM employees
WHERE department_id IN (SELECT department_id
                        FROM employees
                        WHERE last_name like '%u%');
```

The bottom pane shows the 'Query Result' tab, which displays the output of the query. The results are presented in a table with two columns: 'EMPLOYEE_ID' and 'LAST_NAME'. The table contains 18 rows of data, all of which have a last name containing the letter 'u'.

EMPLOYEE_ID	LAST_NAME
1	103 Hunold
2	104 Ernst
3	105 Austin
4	106 Pataballa
5	107 Lorentz
6	120 Weiss
7	121 Fripp
8	122 Kaufling
9	123 Vollman
10	124 Mourgos
11	125 Nayer
12	126 Mikkilineni
13	127 Landry
14	128 Markle
15	129 Bissot
16	130 Atkinson
17	131 Marlow
18	132 Olson

Note: this screenshot shows that I've output all fields required for the practice by using filtering all employee with u in their name

Figure18

The SQL Developer output of oracle sql practice 8-4

The screenshot displays the Oracle SQL Developer interface. The top pane shows a SQL query in the Query Builder:

```
SELECT last_name, department_id, job_id
FROM employees
WHERE department_id IN (SELECT department_id
                        FROM departments
                        WHERE location_id = 1700);
```

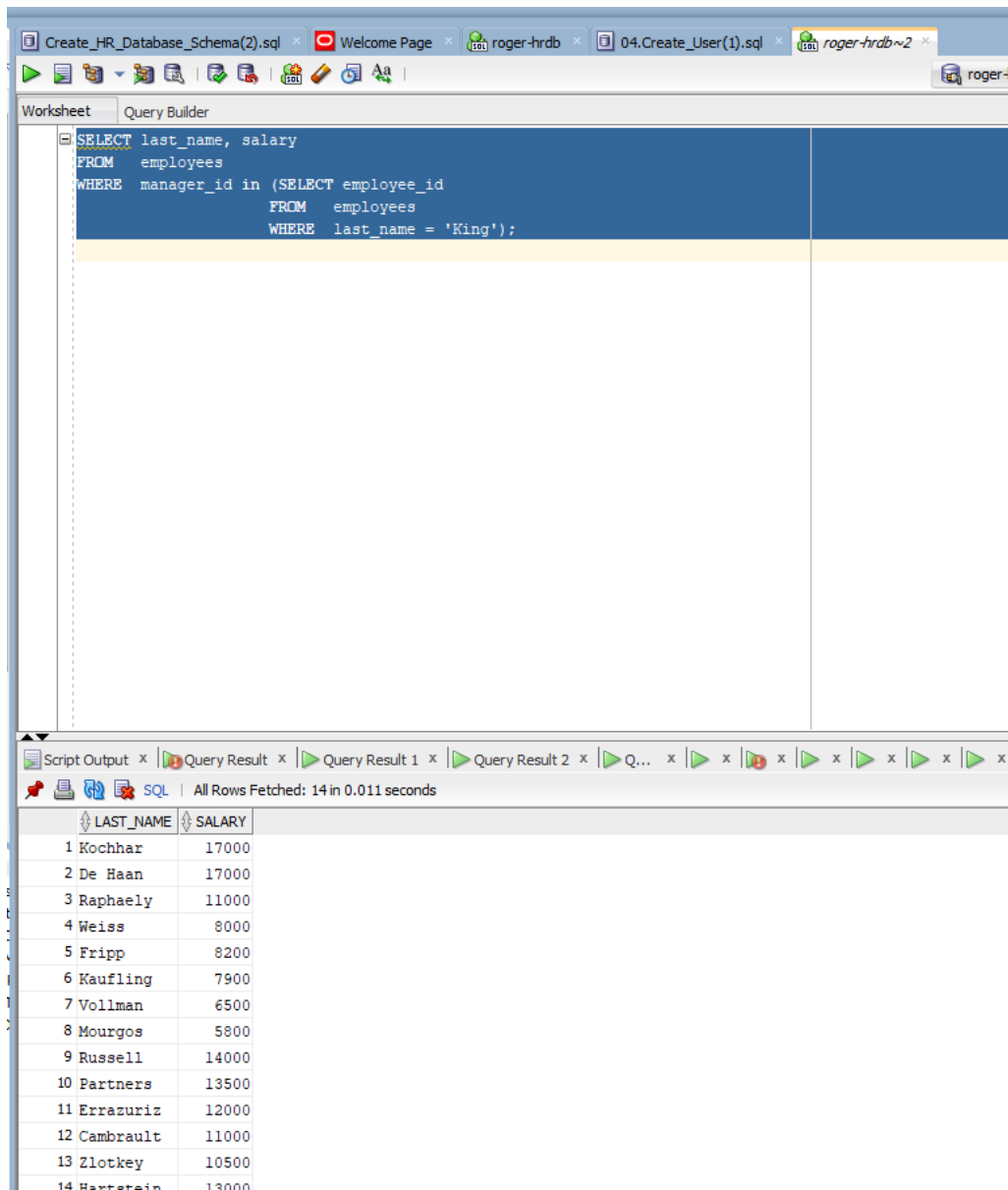
The bottom pane shows the query results in a table format. The status bar indicates "All Rows Fetched: 18 in 0.006 seconds".

	LAST_NAME	DEPARTMENT_ID	JOB_ID
1	King	90	AD_PRES
2	Kochhar	90	AD_VP
3	De Haan	90	AD_VP
4	Greenberg	100	FI_MGR
5	Faviet	100	FI_ACCOUNT
6	Chen	100	FI_ACCOUNT
7	Sciarra	100	FI_ACCOUNT
8	Urman	100	FI_ACCOUNT
9	Popp	100	FI_ACCOUNT
10	Raphaely	30	PU_MAN
11	Khoo	30	PU_CLERK
12	Baida	30	PU_CLERK
13	Tobias	30	PU_CLERK
14	Himuro	30	PU_CLERK
15	Colmenares	30	PU_CLERK
16	Whalen	10	AD_ASST
17	Higgins	110	AC_MGR
18	Gietz	110	AC_ACCOUNT

Note: this screenshot shows that I've output all fields required for the practice by using filtering all employee with location id=1700

Figure19

The SQL Developer output of oracle sql practice 8-5



The screenshot displays the Oracle SQL Developer interface. The top window shows a SQL script with the following query:

```
SELECT last_name, salary
FROM employees
WHERE manager_id in (SELECT employee_id
                     FROM employees
                     WHERE last_name = 'King');
```

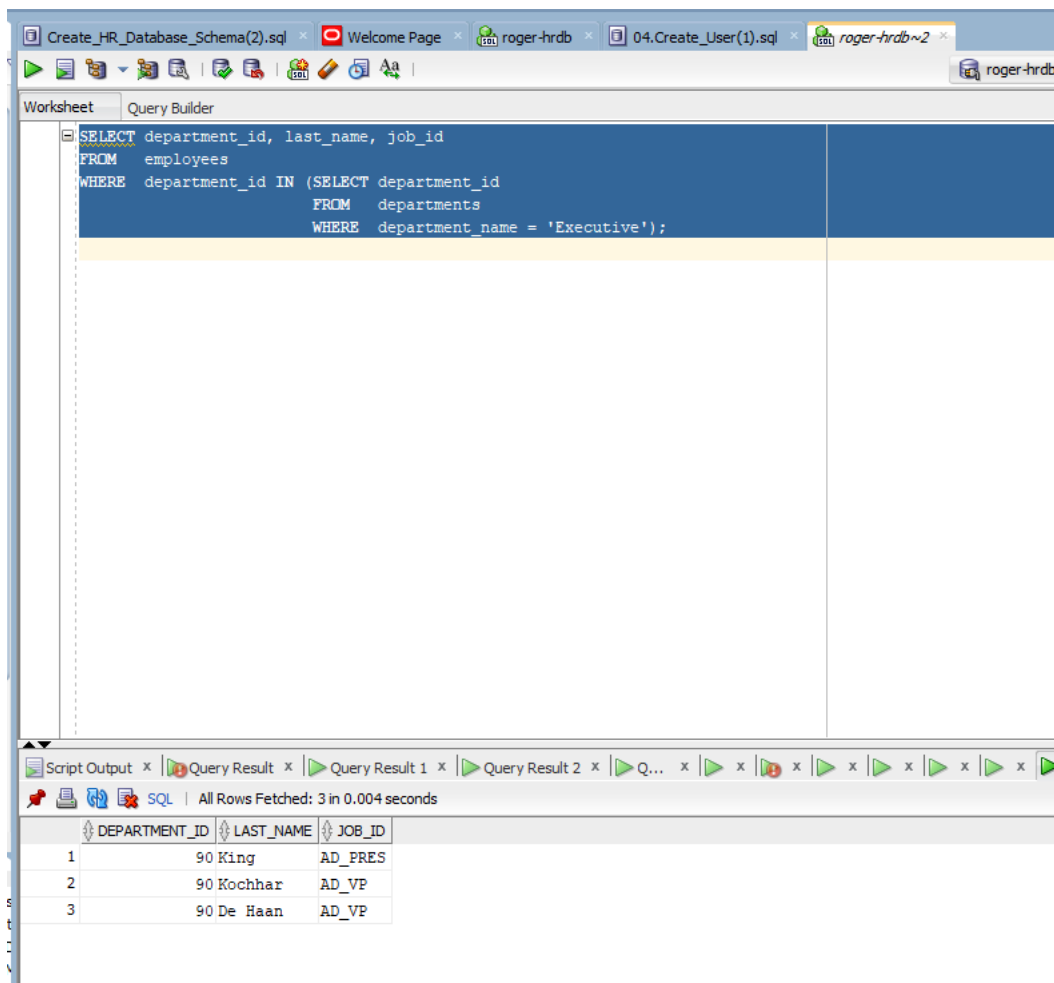
The bottom window shows the query results in a table format. The status bar indicates "All Rows Fetched: 14 in 0.011 seconds".

	LAST_NAME	SALARY
1	Kochhar	17000
2	De Haan	17000
3	Raphaely	11000
4	Weiss	8000
5	Fripp	8200
6	Kaufling	7900
7	Vollman	6500
8	Mourgos	5800
9	Russell	14000
10	Partners	13500
11	Errazuriz	12000
12	Cambrault	11000
13	Zlotkey	10500
14	Wattstein	13000

Note: this screenshot shows that I've output all fields required for the practice by using filtering all employee with king as their manager

Figure20

The SQL Developer output of oracle sql practice 8-6



The screenshot displays the Oracle SQL Developer interface. The top toolbar includes icons for file operations, execution, and formatting. The 'Worksheet' tab is active, showing a SQL query in the 'Query Builder' pane. The query is as follows:

```
SELECT department_id, last_name, job_id
FROM employees
WHERE department_id IN (SELECT department_id
                        FROM departments
                        WHERE department_name = 'Executive');
```

Below the query, the 'Query Result' pane shows the output of the query. It indicates that all rows were fetched in 0.004 seconds. The results are displayed in a table with three columns: DEPARTMENT_ID, LAST_NAME, and JOB_ID.

DEPARTMENT_ID	LAST_NAME	JOB_ID
1	90 King	AD_PRES
2	90 Kochhar	AD_VP
3	90 De Haan	AD_VP

Note: this screenshot shows that I've output all fields required for the practice by using filtering all employee with the executive department

Figure21

The SQL Developer output of oracle sql practice 8-7

The screenshot displays the Oracle SQL Developer interface. The top toolbar includes icons for file operations, execution, and formatting. The main window is divided into a 'Worksheet' and a 'Query Builder' tab. The 'Worksheet' tab contains the following SQL query:

```
SELECT department_id, last_name, job_id
FROM employees
WHERE department_id IN (SELECT department_id
                        FROM departments
                        WHERE department_name = 'Executive');
```

Below the query editor, the 'Query Result' tab shows the output of the query. It indicates that all rows were fetched in 0.004 seconds. The results are displayed in a table with three columns: DEPARTMENT_ID, LAST_NAME, and JOB_ID.

DEPARTMENT_ID	LAST_NAME	JOB_ID
1	90 King	AD_PRES
2	90 Kochhar	AD_VP
3	90 De Haan	AD_VP

Note: this screenshot shows that I've output all fields required for the practice by using filtering all employee with the executive department

Figure22

The SQL Developer output of oracle sql practice 8-8

The screenshot displays the Oracle SQL Developer interface. The main window shows a SQL query in the Query Builder:

```
SELECT employee_id, last_name, salary
FROM employees
WHERE department_id IN (SELECT department_id
                        FROM employees
                        WHERE last_name like '%u%')
AND salary > (SELECT AVG(salary)
              FROM employees);
```

Below the query, the Query Result pane shows the output of the query. The results are displayed in a table with columns EMPLOYEE_ID, LAST_NAME, and SALARY. The table contains 18 rows of data, representing employees whose last names contain the letter 'u' and whose salaries are greater than the average salary of all employees.

EMPLOYEE_ID	LAST_NAME	SALARY
1	103 Hunold	9000
2	120 Weiss	8000
3	121 Fripp	8200
4	122 Kaufling	7900
5	123 Vollman	6500
6	145 Russell	14000
7	146 Partners	13500
8	147 Errazuriz	12000
9	148 Cambrault	11000
10	149 Zlotkey	10500
11	150 Tucker	10000
12	151 Bernstein	9500
13	152 Hall	9000
14	153 Olsen	8000
15	154 Cambrault	7500
16	155 Tuvault	7000
17	156 King	10000
18	157 Sully	9500

Note: this screenshot shows that I've output all fields required for the practice by using filtering all employee with the more than avg salary

Oracle Practice 9

Figure23

The SQL Developer output of oracle sql practice 9-1

The screenshot displays the Oracle SQL Developer interface. The main window shows a SQL query in the Worksheet tab:

```
SELECT department_id  
FROM departments  
MINUS  
SELECT department_id  
FROM employees  
WHERE job_id = 'ST_CLERK';
```

Below the query, the Query Result tab shows the output of the query. The results are displayed in a table with two columns: DEPARTMENT_ID and an unnamed column. The data is as follows:

DEPARTMENT_ID	
1	10
2	20
3	30
4	40
5	60
6	70
7	80
8	90
9	100
10	110
11	120
12	130
13	140
14	150
15	160
16	170
17	180
18	190

The status bar at the bottom indicates "All Rows Fetched: 26 in 0.007 seconds".

Note: this screenshot shows that I've output all fields required for the practice by using filtering all department with employee id st_clerk

Figure24

The SQL Developer output of oracle sql practice 9-2

The screenshot displays the Oracle SQL Developer interface. The top toolbar includes icons for file operations, execution, and debugging. The 'Worksheet' tab is active, showing a SQL query in the 'Query Builder' pane. The query is as follows:

```
SELECT country_id, country_name
FROM countries
MINUS
SELECT l.country_id, c.country_name
FROM locations l JOIN countries c
ON (l.country_id = c.country_id)
JOIN departments d
ON d.location_id=l.location_id;
```

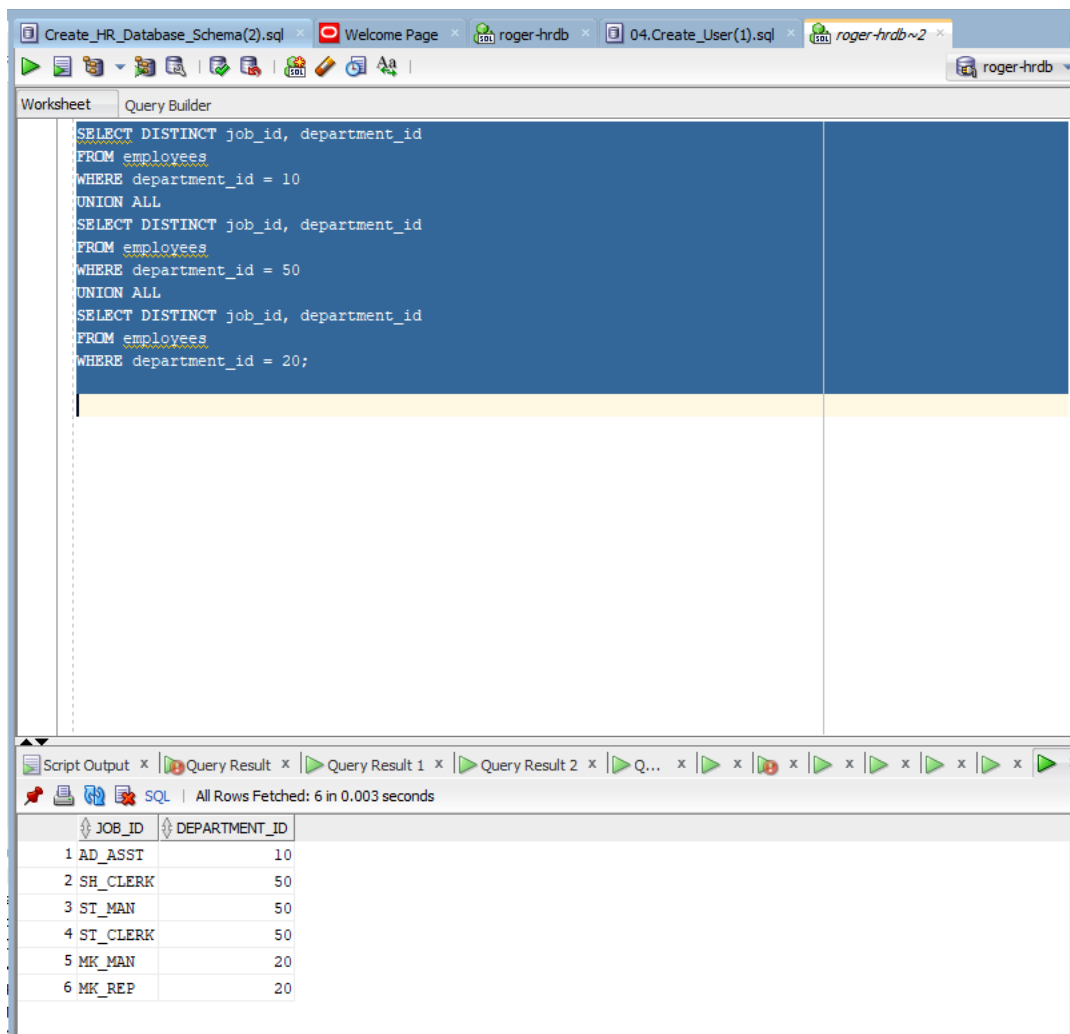
Below the query, the 'Query Result' pane shows the output of the query. The status bar indicates 'All Rows Fetched: 21 in 0.005 seconds'. The results are displayed in a table with two columns: COUNTRY_ID and COUNTRY_NAME.

COUNTRY_ID	COUNTRY_NAME
1 AR	Argentina
2 AU	Australia
3 BE	Belgium
4 BR	Brazil
5 CH	Switzerland
6 CN	China
7 DK	Denmark
8 EG	Egypt
9 FR	France
10 HK	HongKong
11 IL	Israel
12 IN	India
13 IT	Italy
14 JP	Japan
15 KW	Kuwait
16 MX	Mexico
17 NG	Nigeria
18 NL	Netherlands

Note: this screenshot shows that I've output all fields required for the practice by using filtering all country without department

Figure25

The SQL Developer output of oracle sql practice 9-3



The screenshot displays the Oracle SQL Developer interface. The top toolbar includes icons for running queries, saving, and other standard database operations. The main window is divided into two panes: the left pane shows the SQL query, and the right pane shows the query results.

The SQL query in the left pane is as follows:

```
SELECT DISTINCT job_id, department_id
FROM employees
WHERE department_id = 10
UNION ALL
SELECT DISTINCT job_id, department_id
FROM employees
WHERE department_id = 50
UNION ALL
SELECT DISTINCT job_id, department_id
FROM employees
WHERE department_id = 20;
```

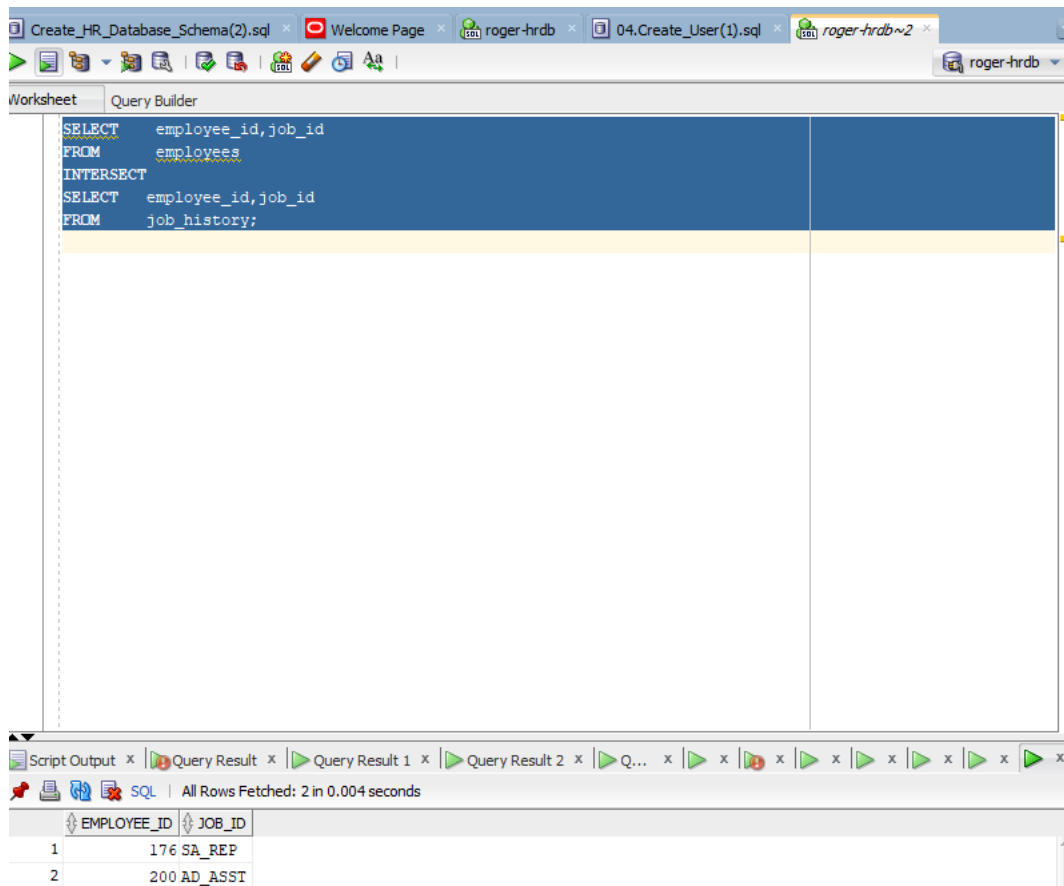
The query results are displayed in the right pane, showing a table with two columns: JOB_ID and DEPARTMENT_ID. The results are as follows:

JOB_ID	DEPARTMENT_ID
1 AD_ASST	10
2 SH_CLERK	50
3 ST_MAN	50
4 ST_CLERK	50
5 MK_MAN	20
6 MK_REP	20

Note: this screenshot shows that I've output all fields required for the practice by using filtering all job_id in department id

Figure26

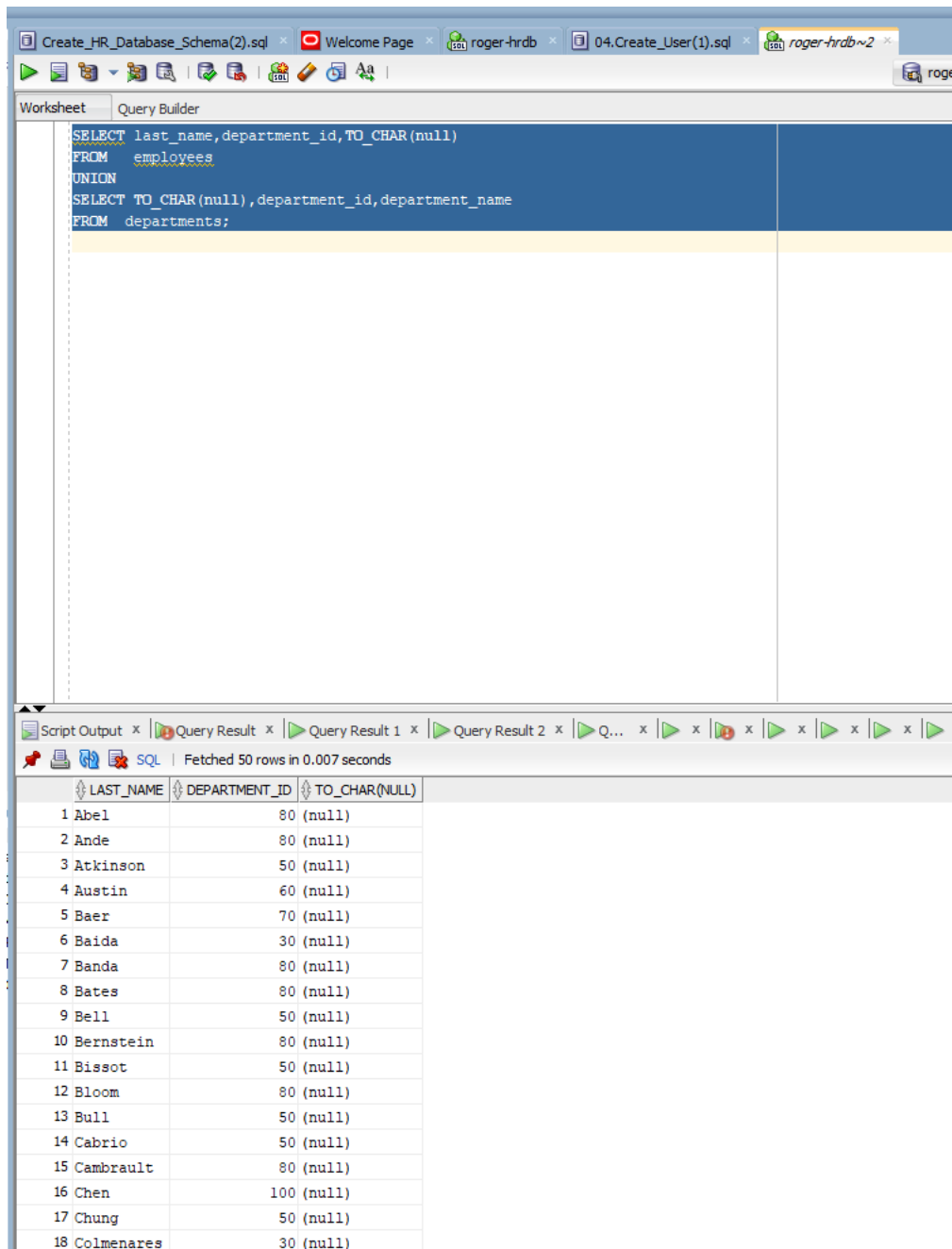
The SQL Developer output of oracle sql practice 9-4



Note: this screenshot shows that I've output all fields required for the practice by using filtering all employee same with their previous jobs

Figure27

The SQL Developer output of oracle sql practice 9-5



The screenshot displays the Oracle SQL Developer interface. The top pane shows a SQL query in the 'Query Builder' tab. The query is a UNION of two SELECT statements. The first SELECT statement retrieves the last name and department ID from the 'employees' table, with the department ID converted to a CHAR type. The second SELECT statement retrieves the department ID and department name from the 'departments' table, with the department name converted to a CHAR type. The bottom pane shows the 'Query Result' tab, which displays the results of the query. The results are fetched in 0.007 seconds and consist of 50 rows. The columns are LAST_NAME, DEPARTMENT_ID, and TO_CHAR(NULL). The data is presented in a table with 18 rows visible, each containing a row number, a last name, a department ID, and a null value for the TO_CHAR(NULL) column.

```
SELECT last_name,department_id,TO_CHAR(null)
FROM employees
UNION
SELECT TO_CHAR(null),department_id,department_name
FROM departments;
```

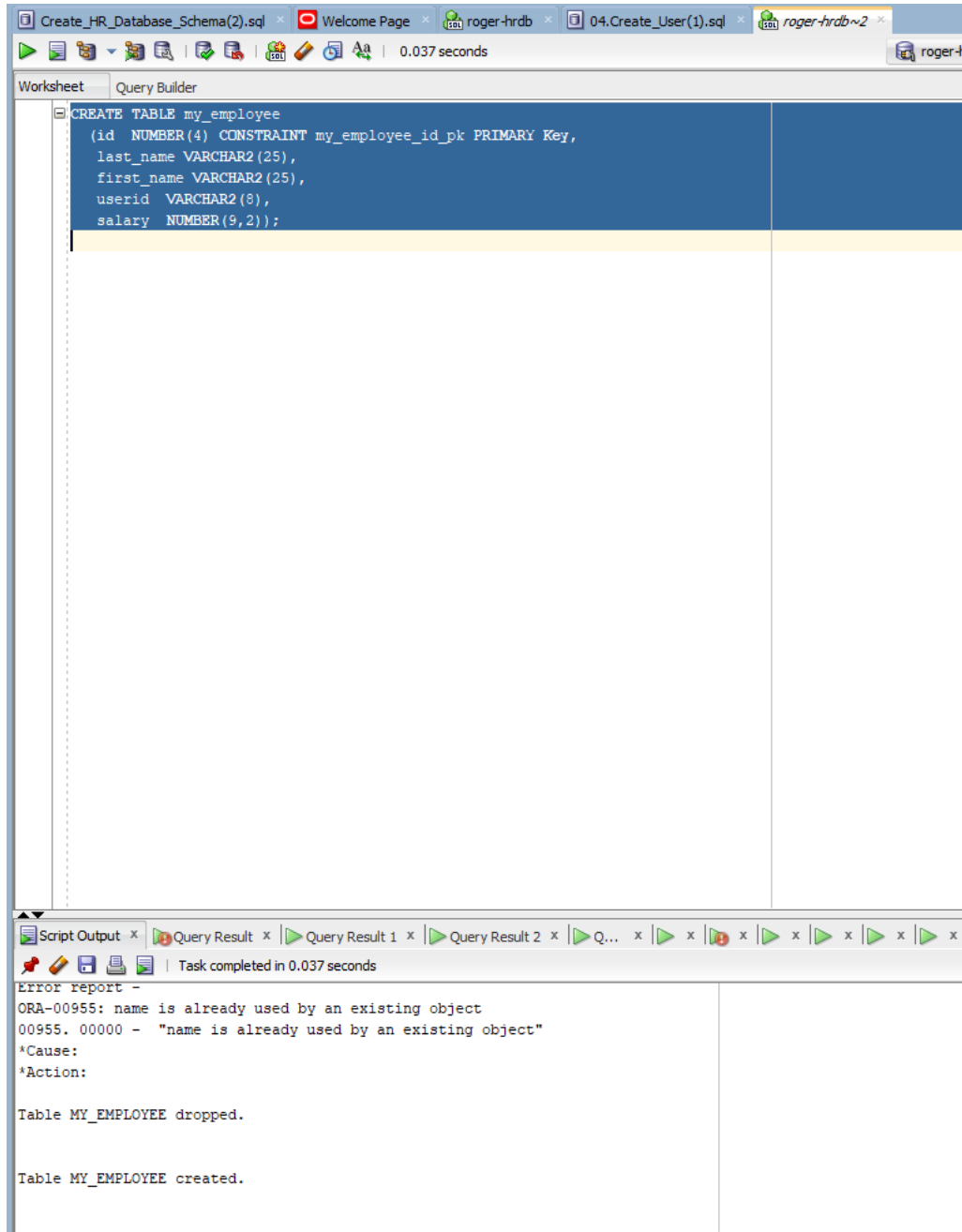
	LAST_NAME	DEPARTMENT_ID	TO_CHAR(NULL)
1	Abel	80	(null)
2	Ande	80	(null)
3	Atkinson	50	(null)
4	Austin	60	(null)
5	Baer	70	(null)
6	Baida	30	(null)
7	Banda	80	(null)
8	Bates	80	(null)
9	Bell	50	(null)
10	Bernstein	80	(null)
11	Bissot	50	(null)
12	Bloom	80	(null)
13	Bull	50	(null)
14	Cabrio	50	(null)
15	Cambrault	80	(null)
16	Chen	100	(null)
17	Chung	50	(null)
18	Colmenares	30	(null)

Note: this screenshot shows that I've output all fields required for the practice by using filtering all employee last name and department number

Oracle practice 10

Figure28

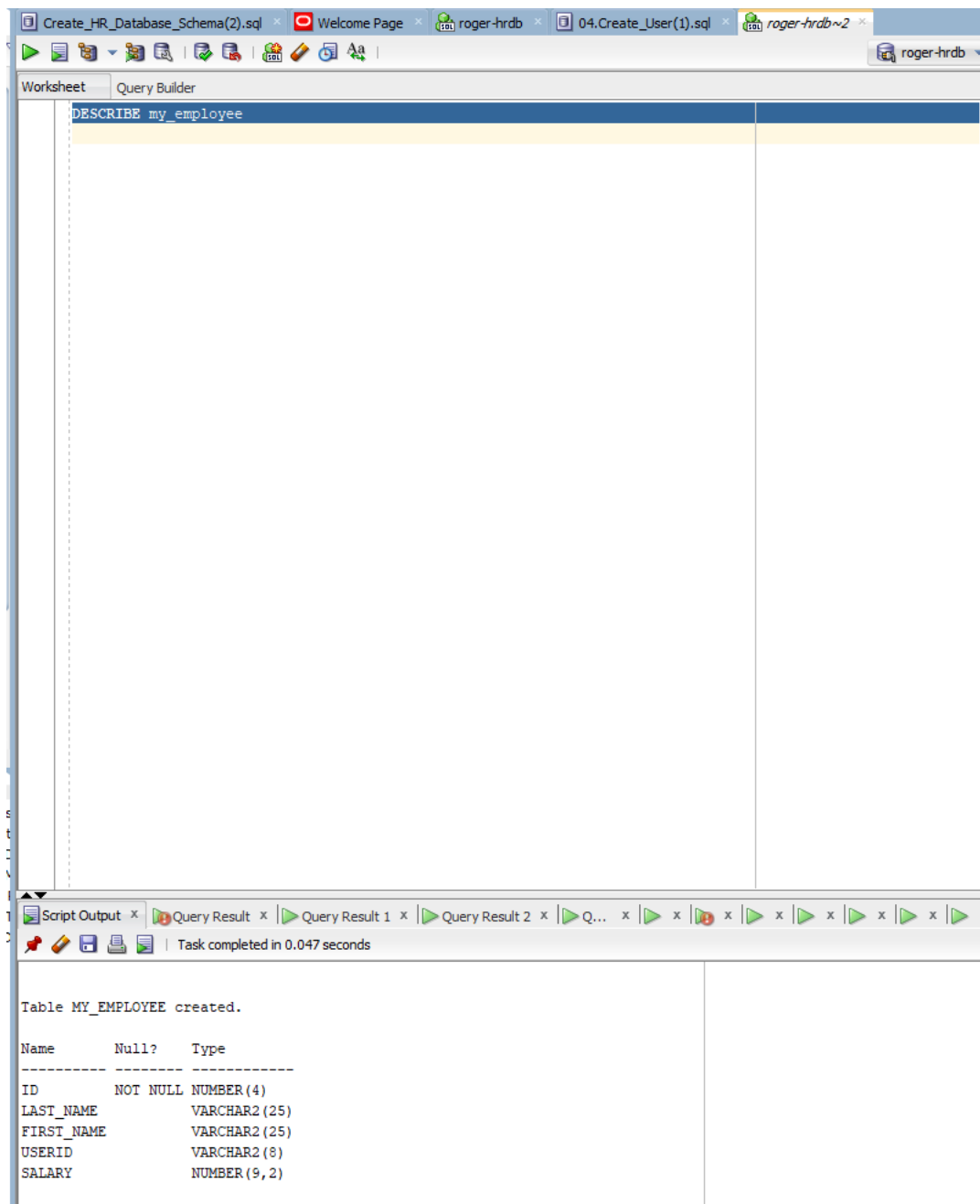
The SQL Developer output of oracle sql practice 10-1



Note: this screenshot shows that I've output the successful creation of the table needed for the practice

Figure29

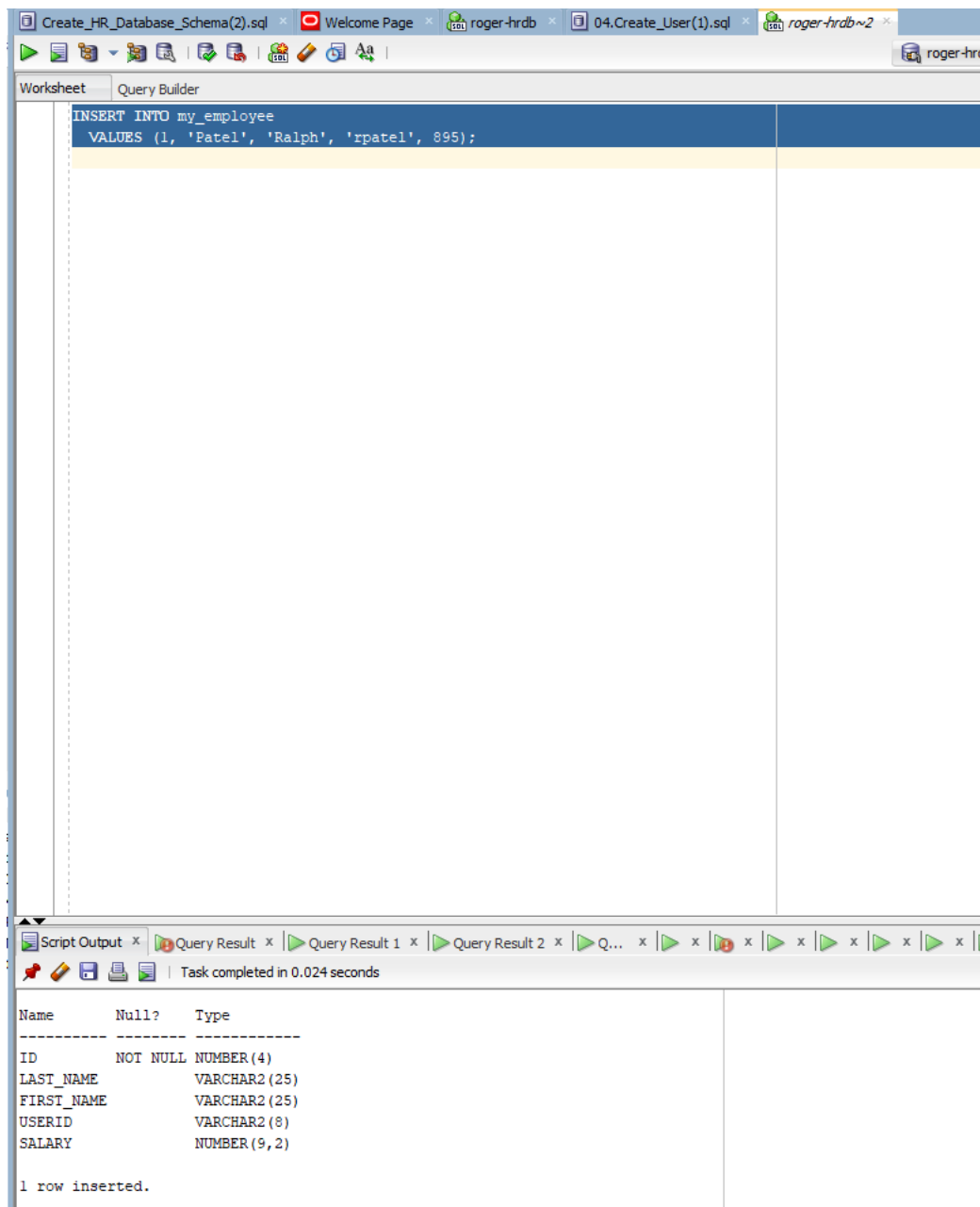
The SQL Developer output of oracle sql practice 10-2



Note: this screenshot shows that I've output the description of the table needed for the practice

Figure30

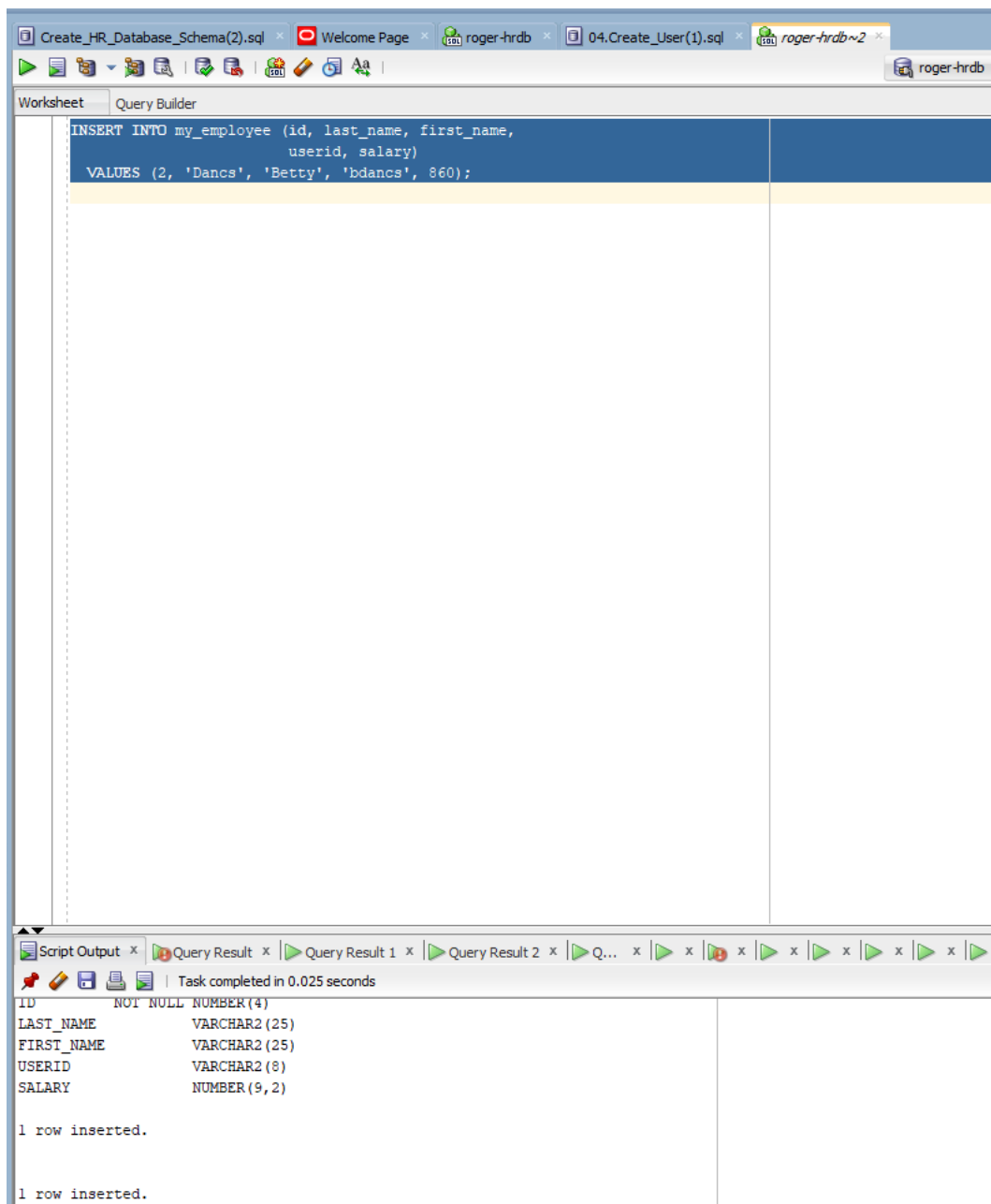
The SQL Developer output of oracle sql practice 10-3



Note: this screenshot shows that I've output the successful insert of the record.

Figure31

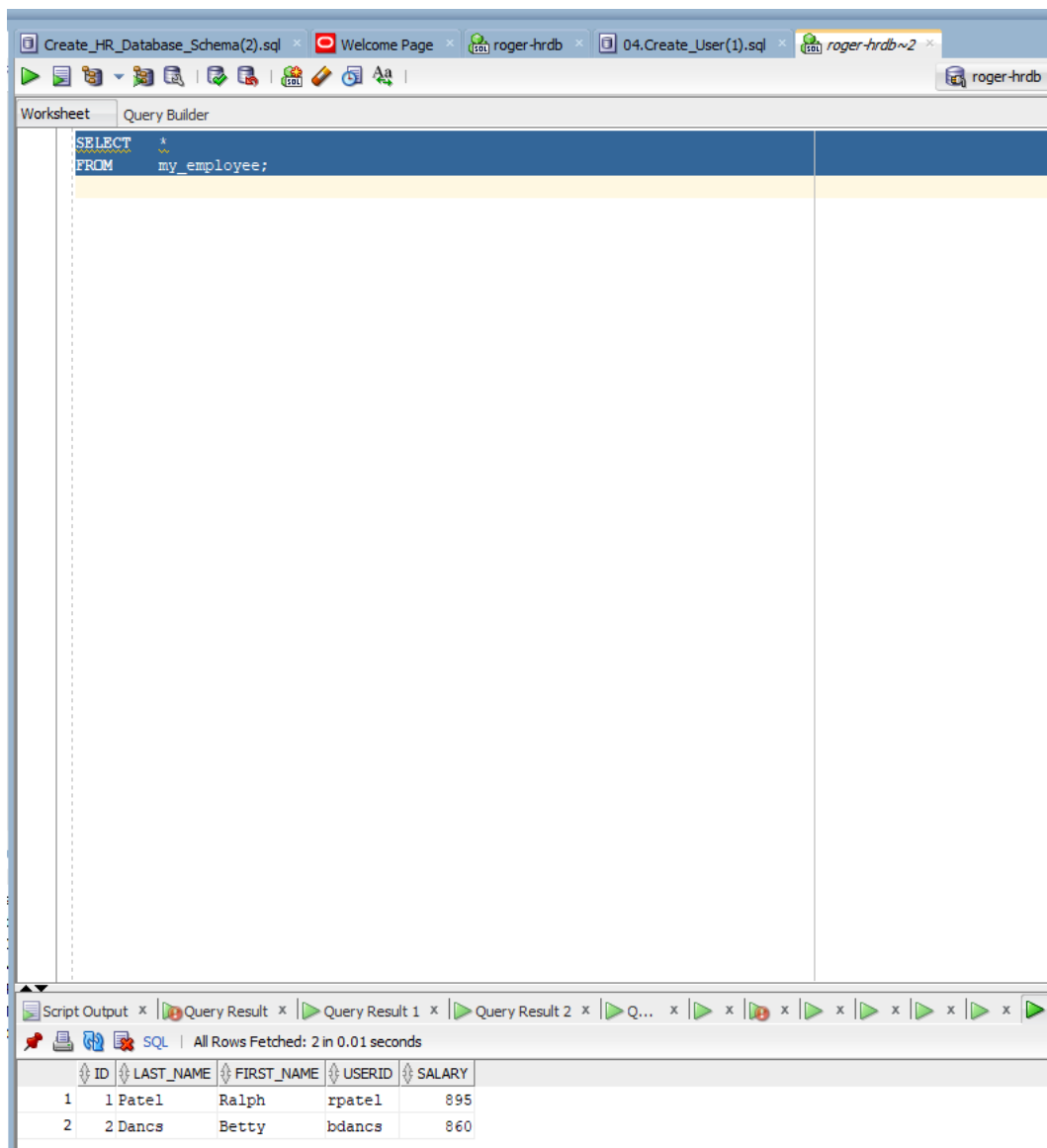
The SQL Developer output of oracle sql practice 10-4



Note: this screenshot shows that I've output the successful insert of the record.

Figure32

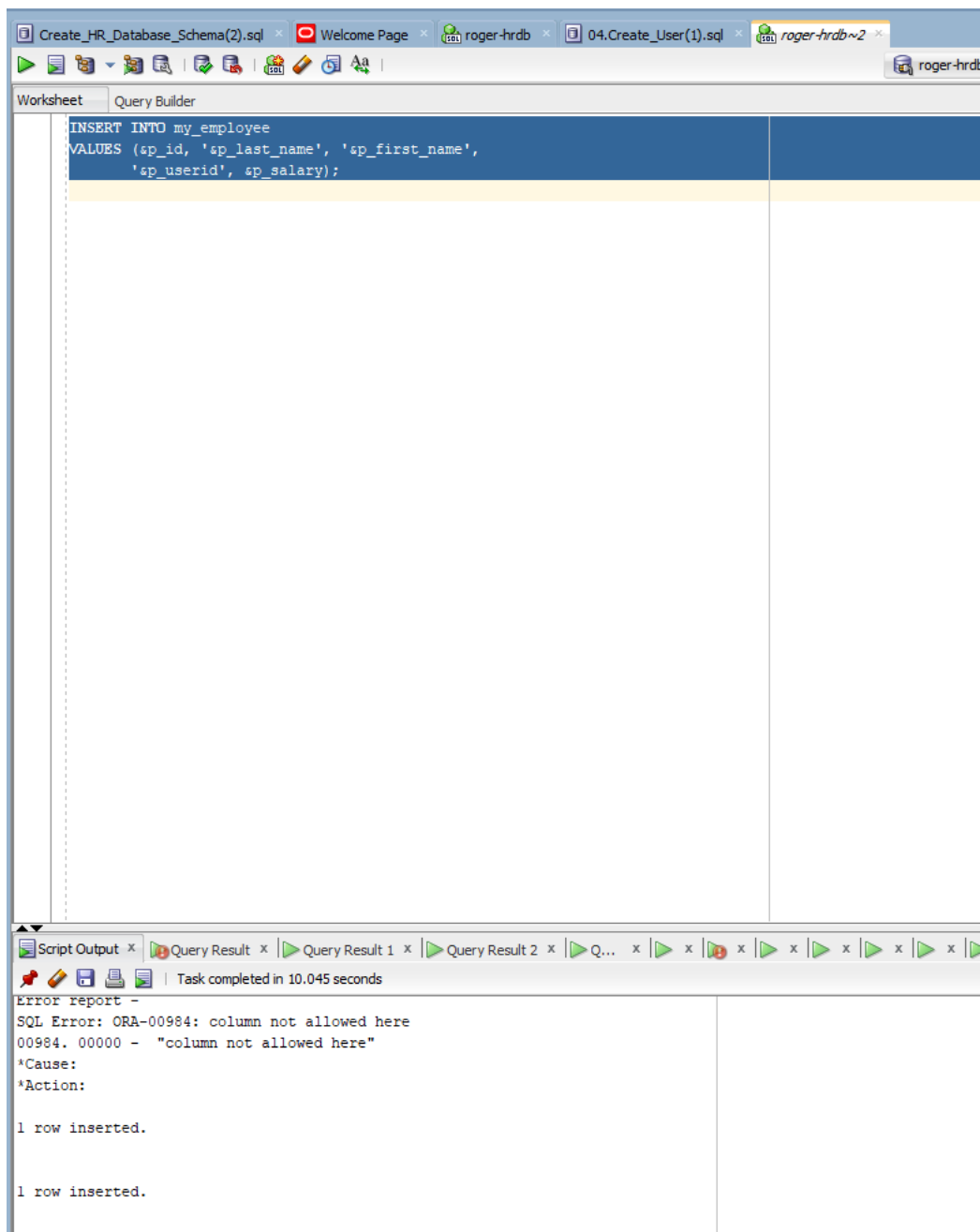
The SQL Developer output of oracle sql practice 10-5



Note: this screenshot shows that I've output the previous rows that inserted

Figure33

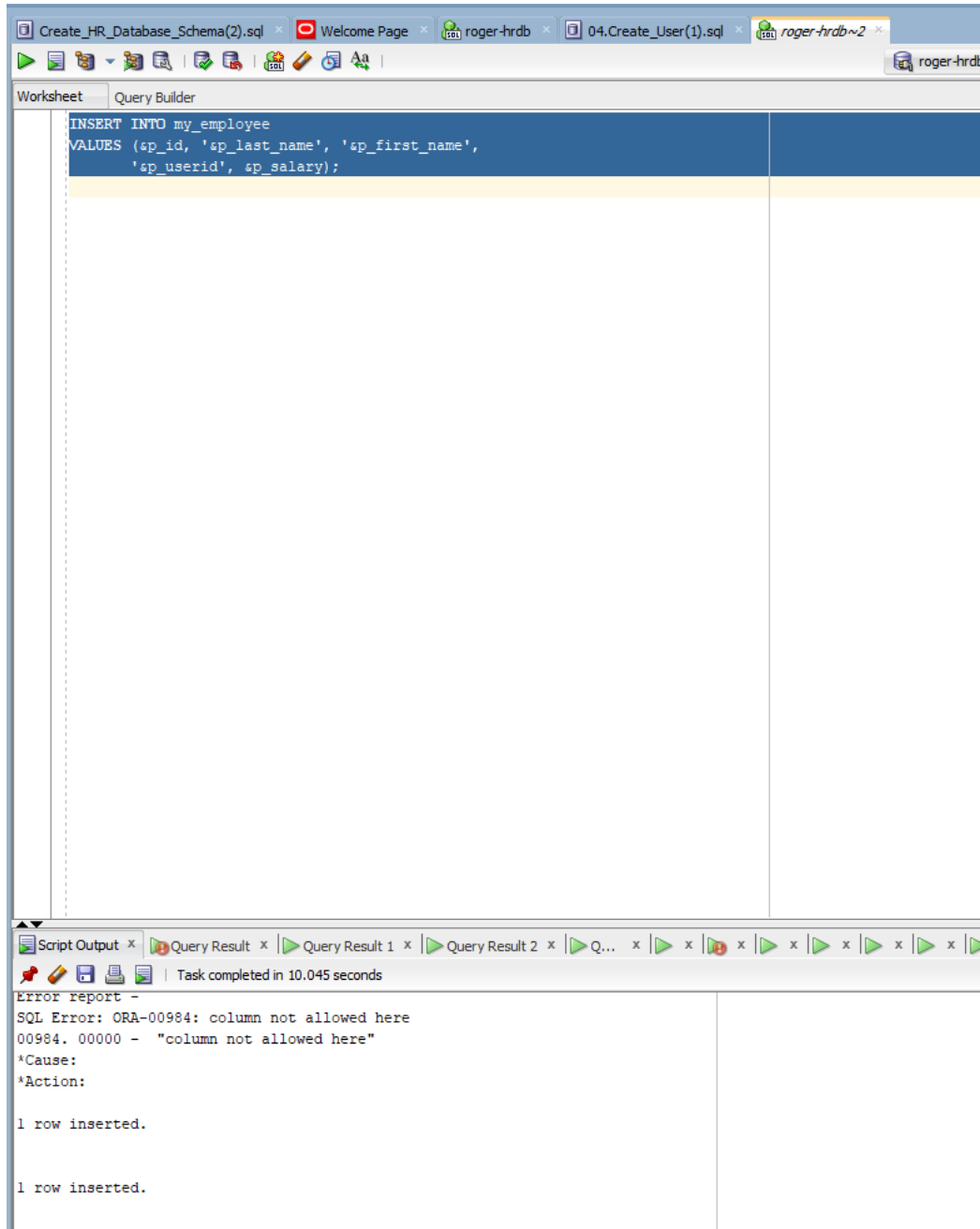
The SQL Developer output of oracle sql practice 10-6



Note: this screenshot shows that I've output the previous rows that inserted

Figure34

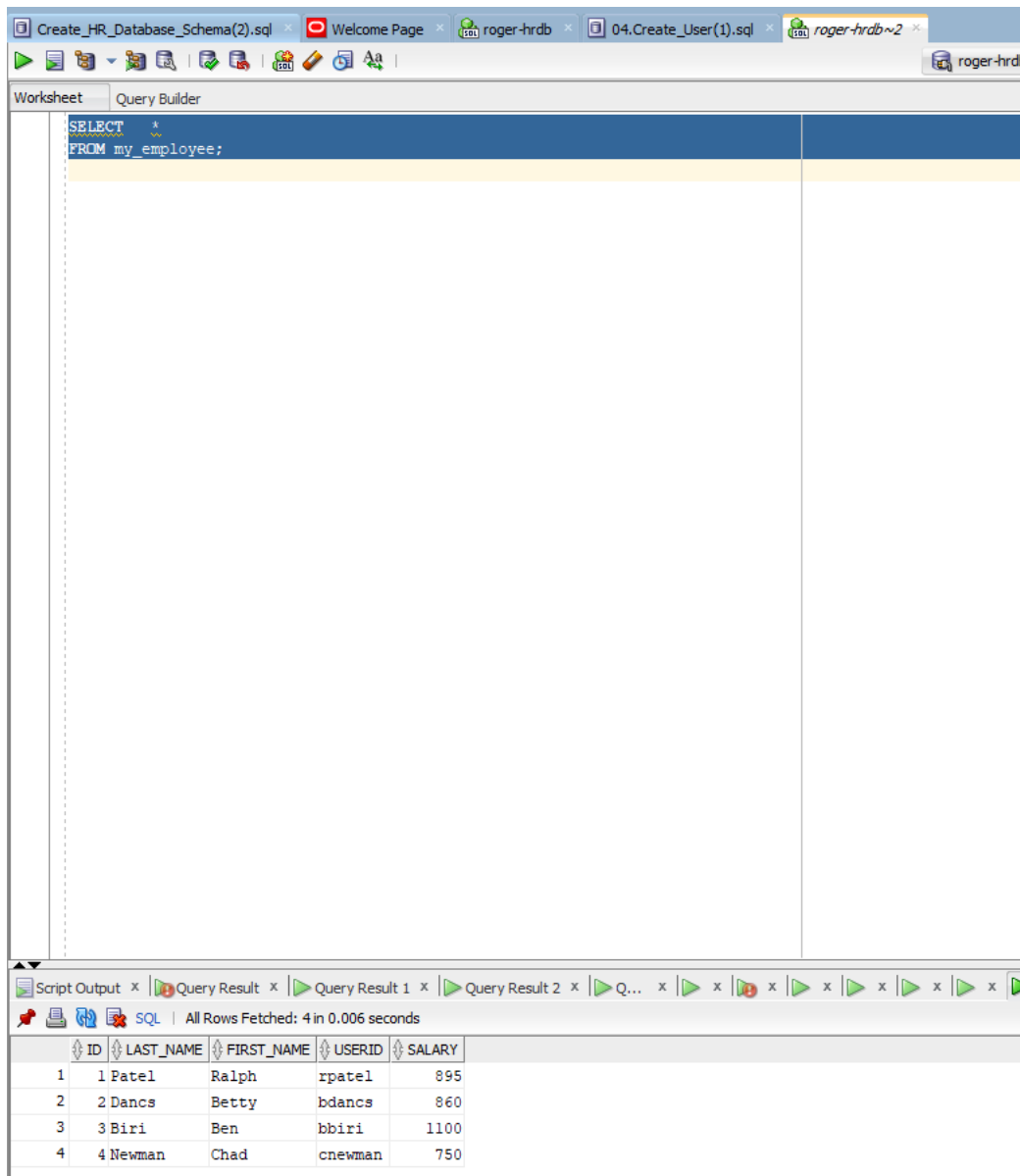
The SQL Developer output of oracle sql practice 10-7



Note: this screenshot shows that I've output the two required rows are inserted

Figure35

The SQL Developer output of oracle sql practice 10-8



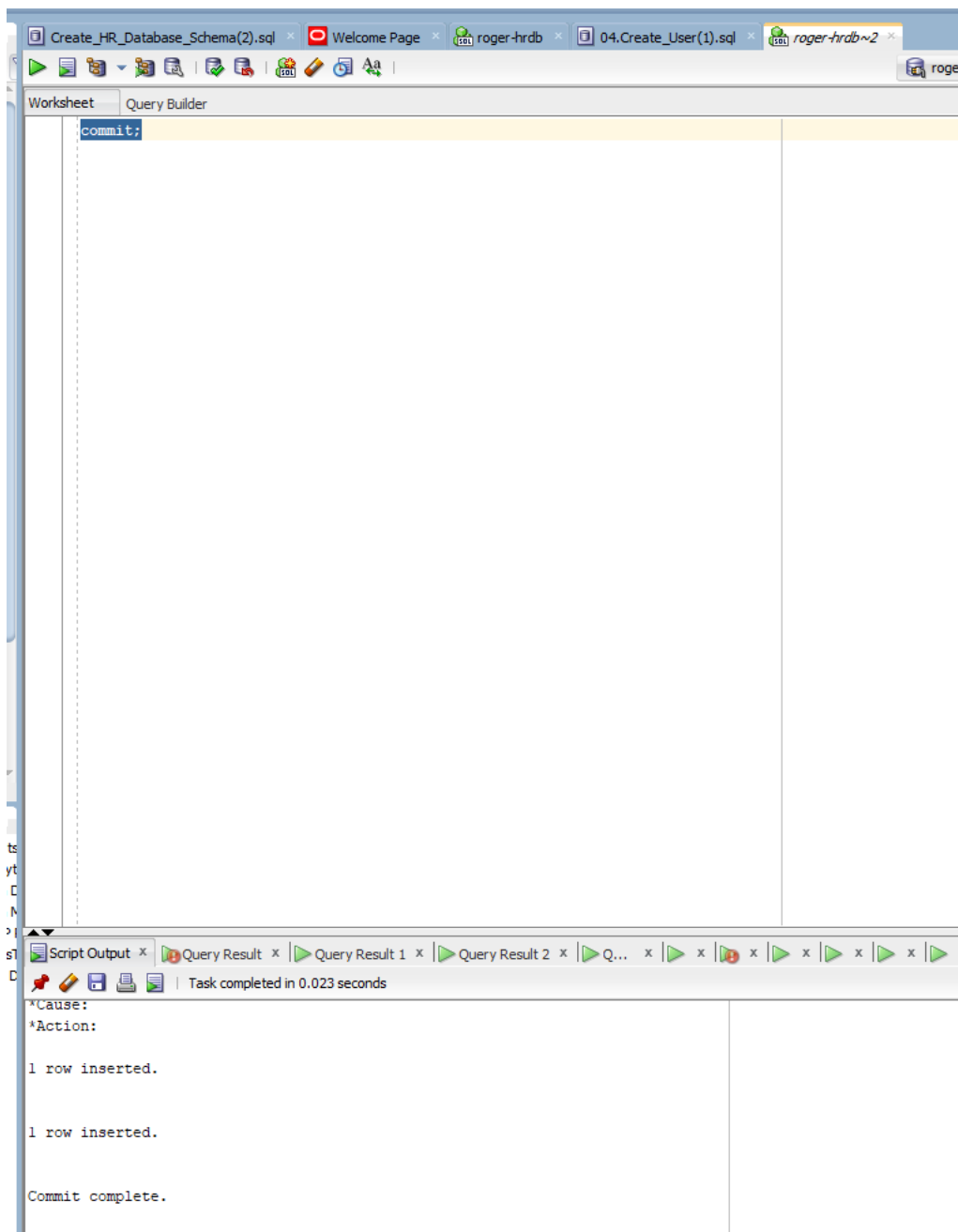
The screenshot displays the Oracle SQL Developer environment. The top toolbar includes icons for running queries, saving, and other standard database operations. The main workspace is divided into a 'Worksheet' and a 'Query Builder' tab. The 'Worksheet' tab is active, showing a SQL query: `SELECT * FROM my_employee;`. Below the query, the 'Query Result' tab is visible, displaying the output of the query. The output is a table with 5 columns: ID, LAST_NAME, FIRST_NAME, USERID, and SALARY. It contains 4 rows of data, representing the four required rows inserted into the database. The status bar at the bottom indicates 'All Rows Fetched: 4 in 0.006 seconds'.

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	895
2	Dancs	Betty	bdancs	860
3	Biri	Ben	bbiri	1100
4	Newman	Chad	cnewman	750

Note: this screenshot shows that I've output the four required rows are inserted

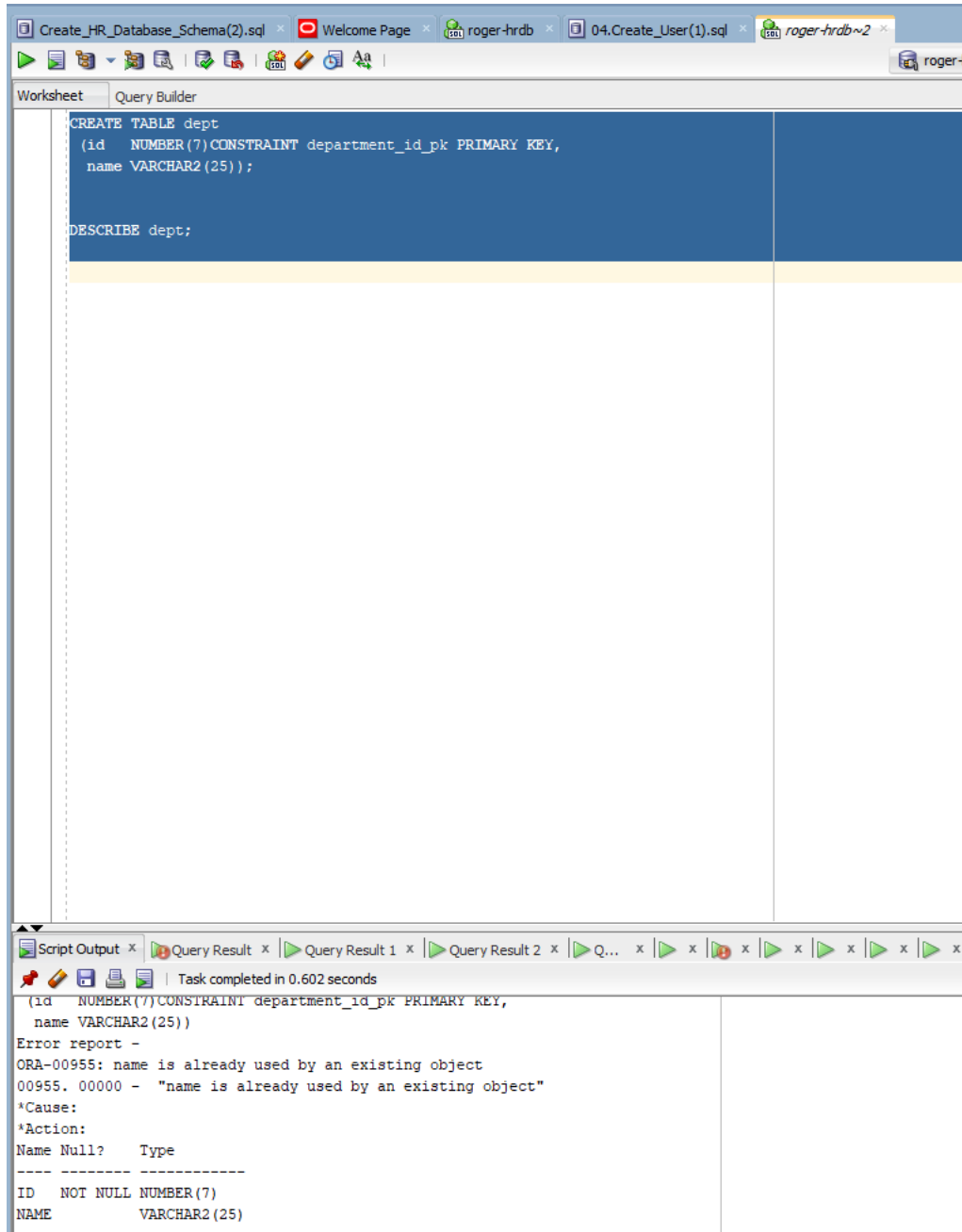
Figure35

The SQL Developer output of oracle sql practice 10-9



Note: this screenshot shows that I've output the four required rows are committed

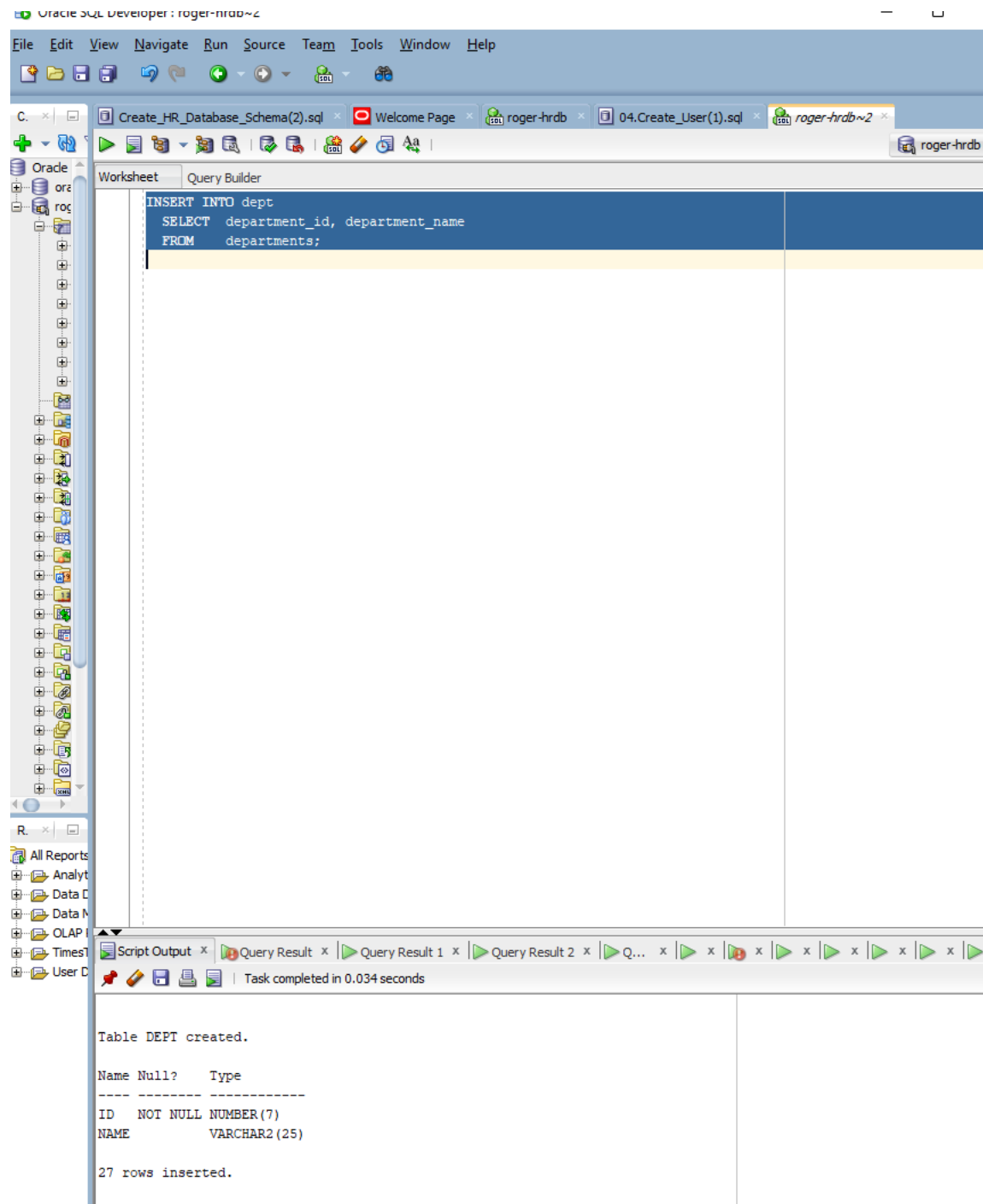
The SQL Developer output of oracle sql practice 11-1



Note: this screenshot shows that I've output the successful creation of the table needed for the practice

Figure37

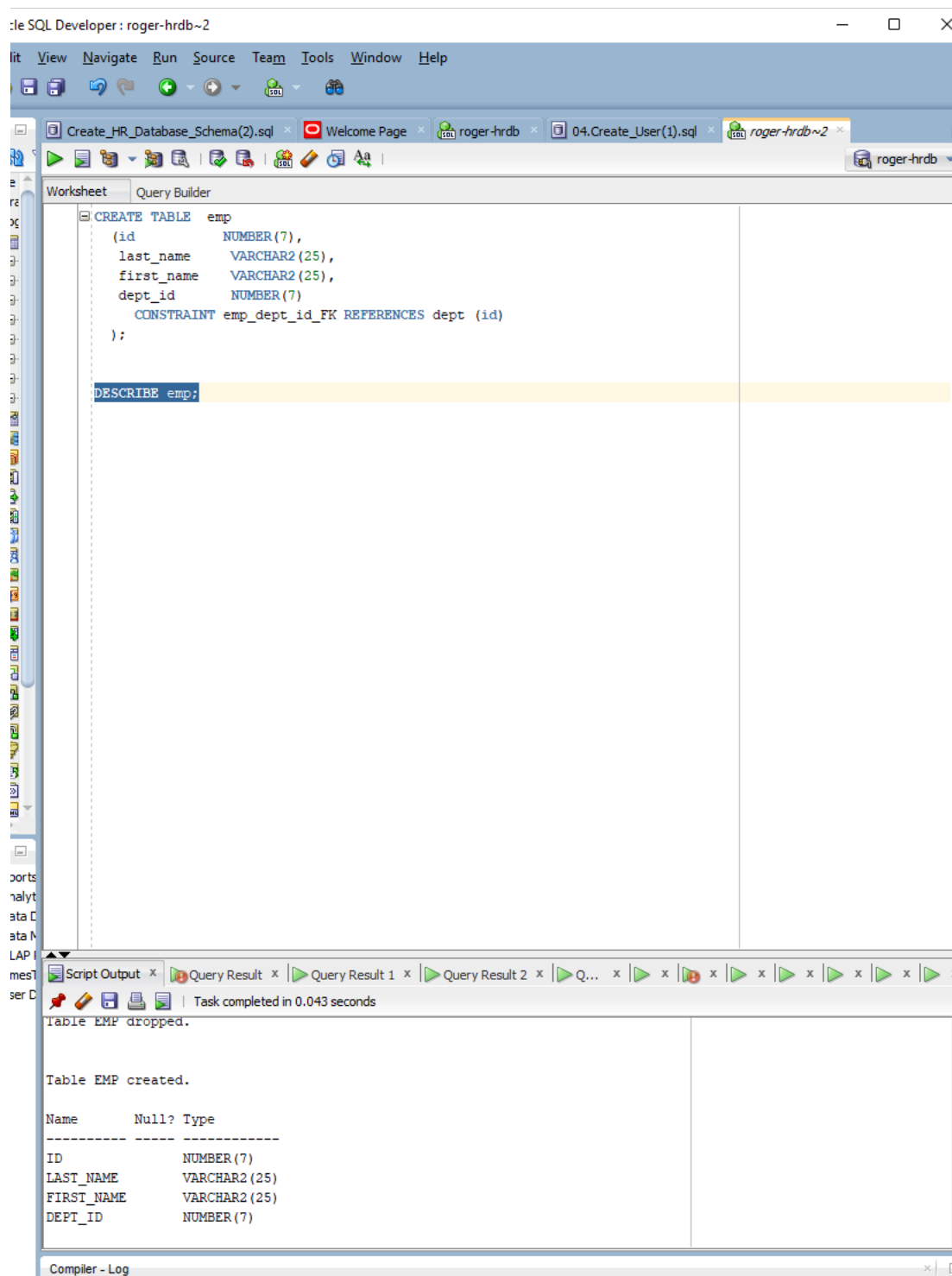
The SQL Developer output of oracle sql practice 11-2



Note: this screenshot shows that I've output the successful population of new created dept table

Figure38

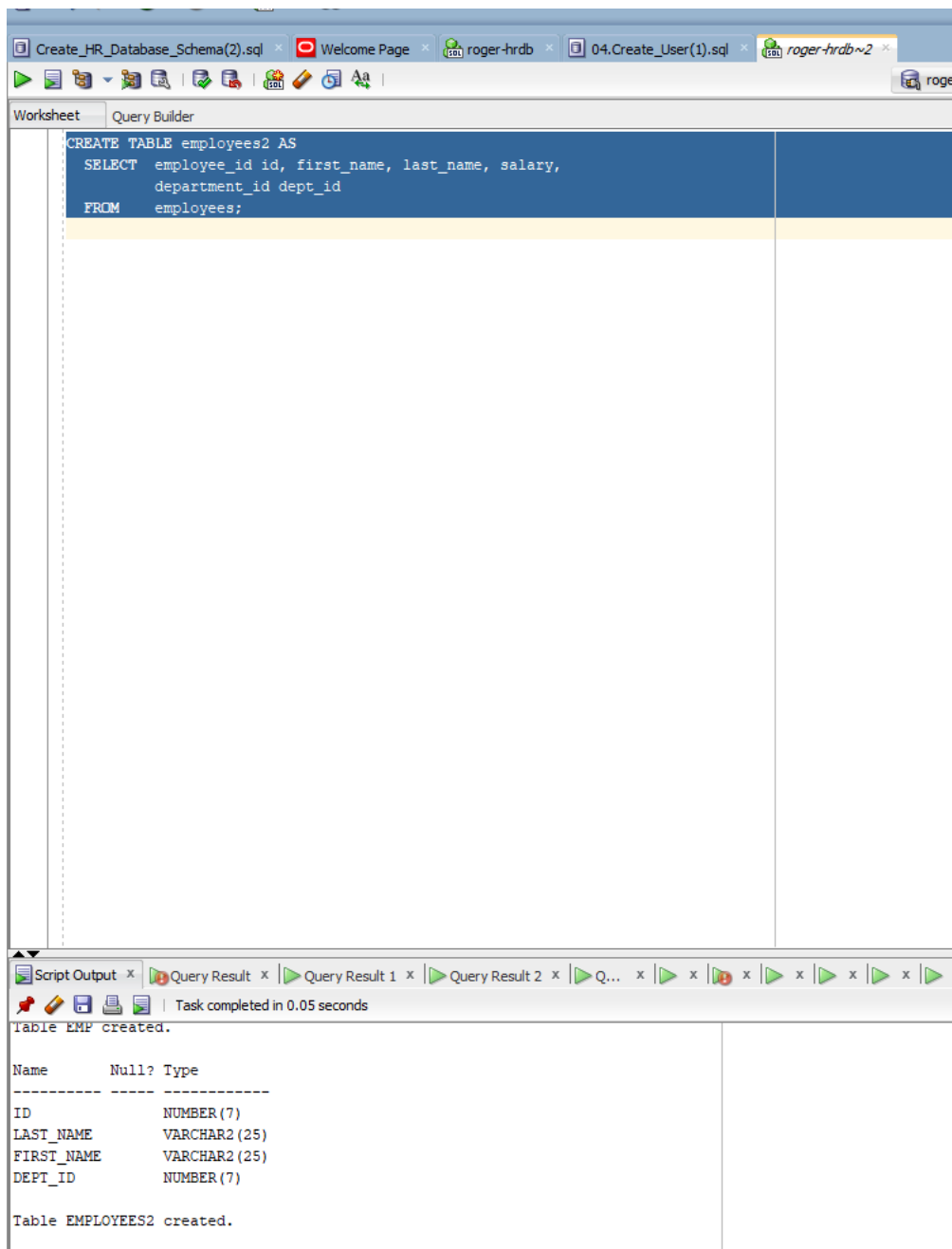
The SQL Developer output of oracle sql practice 11-3



Note: this screenshot shows that I've output the successful creation of the table emp

Figure39

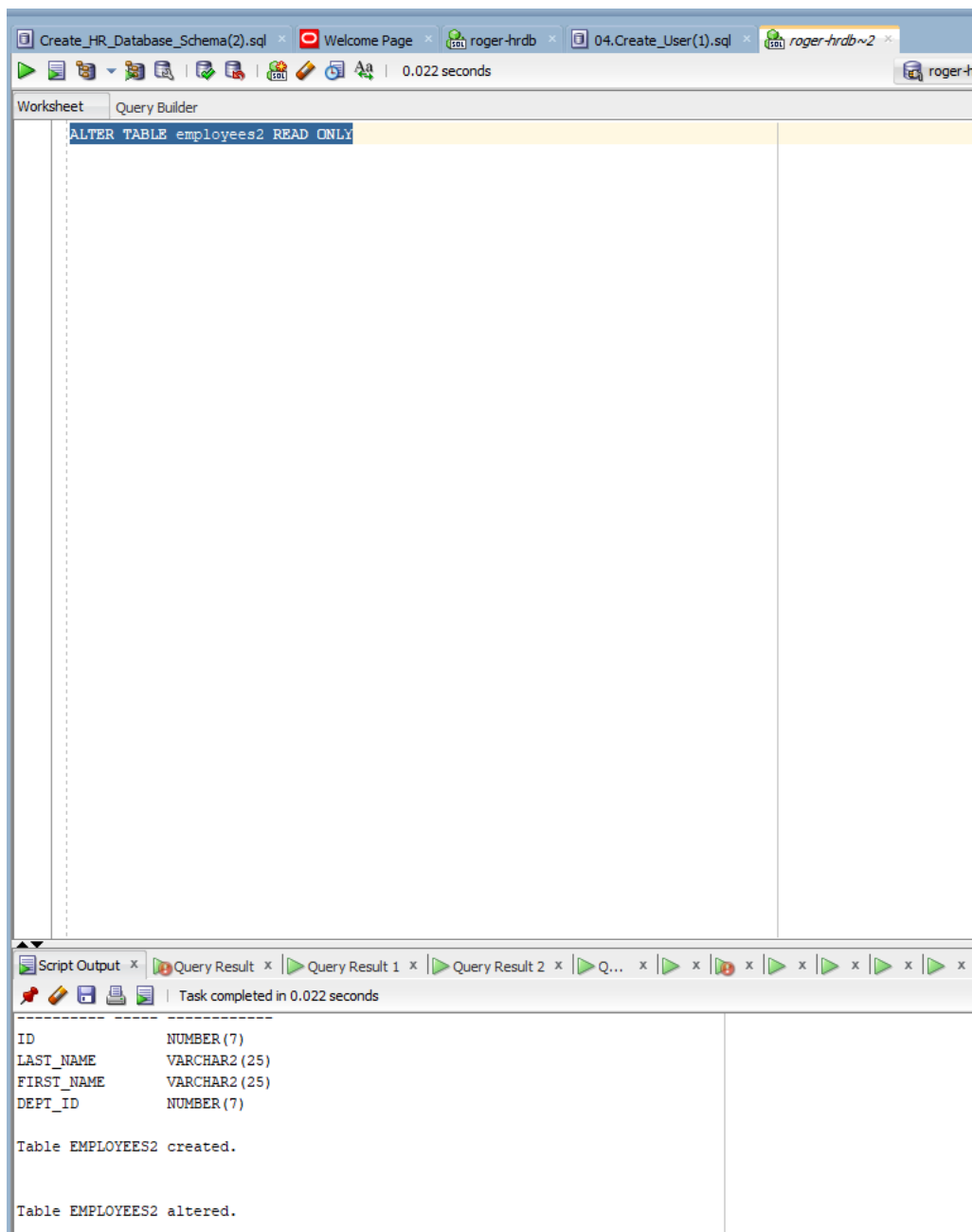
The SQL Developer output of oracle sql practice 11-4



Note: this screenshot shows that I've output the successful creation of the table employee2 and populate with employee table

Figure40

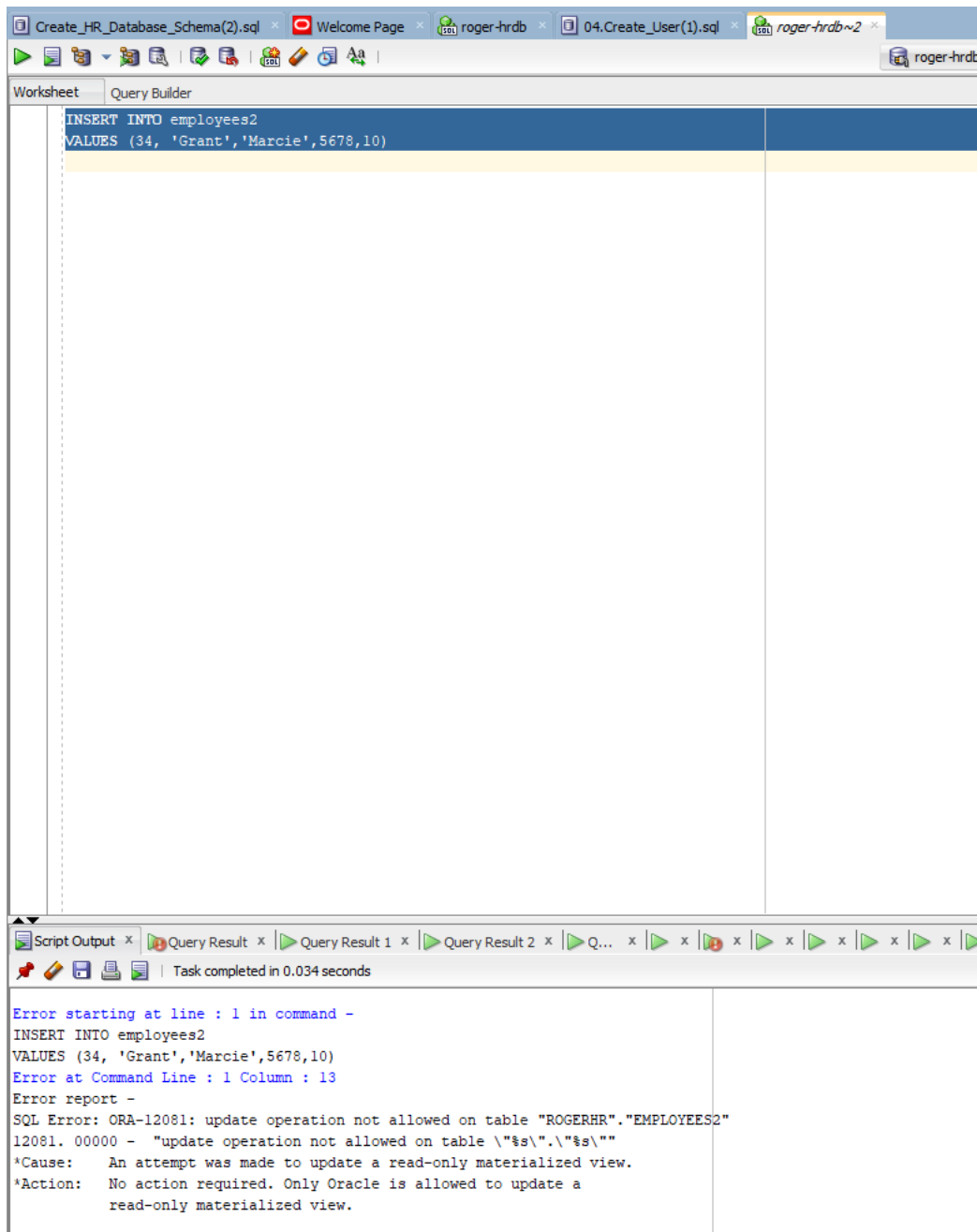
The SQL Developer output of oracle sql practice 11-5



Note: this screenshot shows that I've output the successful creation of the table employee2 and change it to read only

Figure41

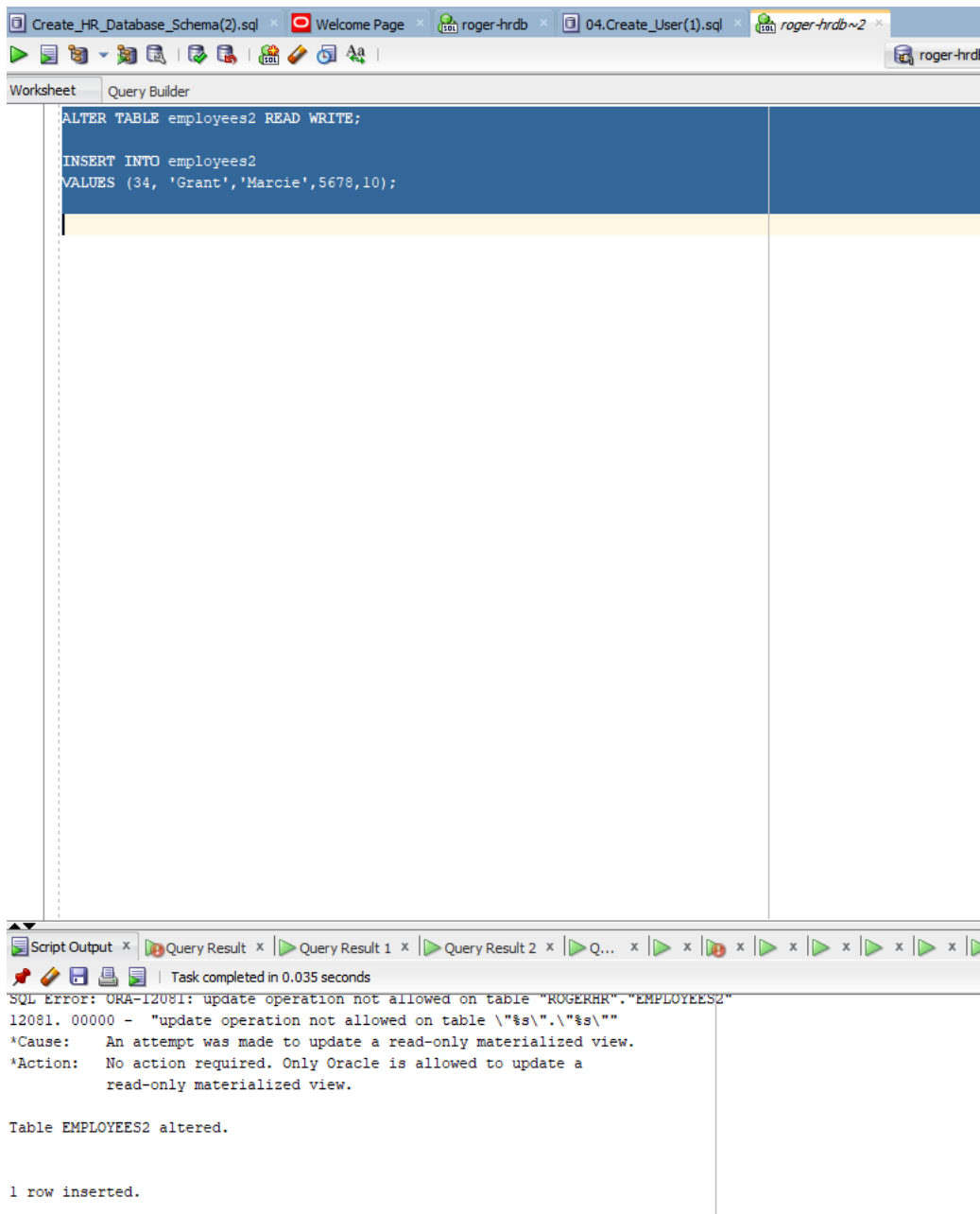
The SQL Developer output of oracle sql practice 11-6



Note: this screenshot shows that I've output the failed insert of record into the read only table

Figure41

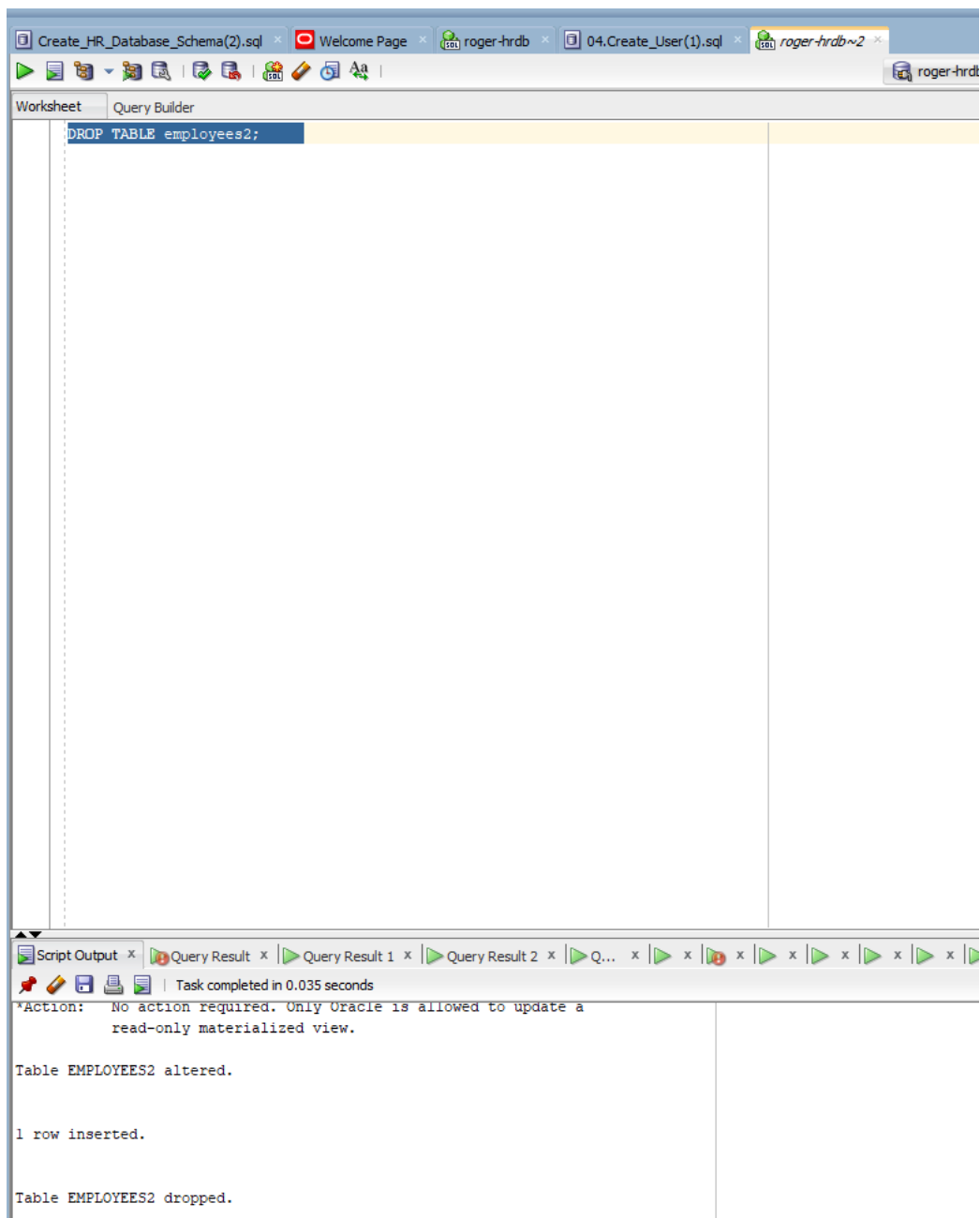
The SQL Developer output of oracle sql practice 11-7



Note: this screenshot shows that I've output the successful insert into the table after setting it read write again

Figure42

The SQL Developer output of oracle sql practice 11-8

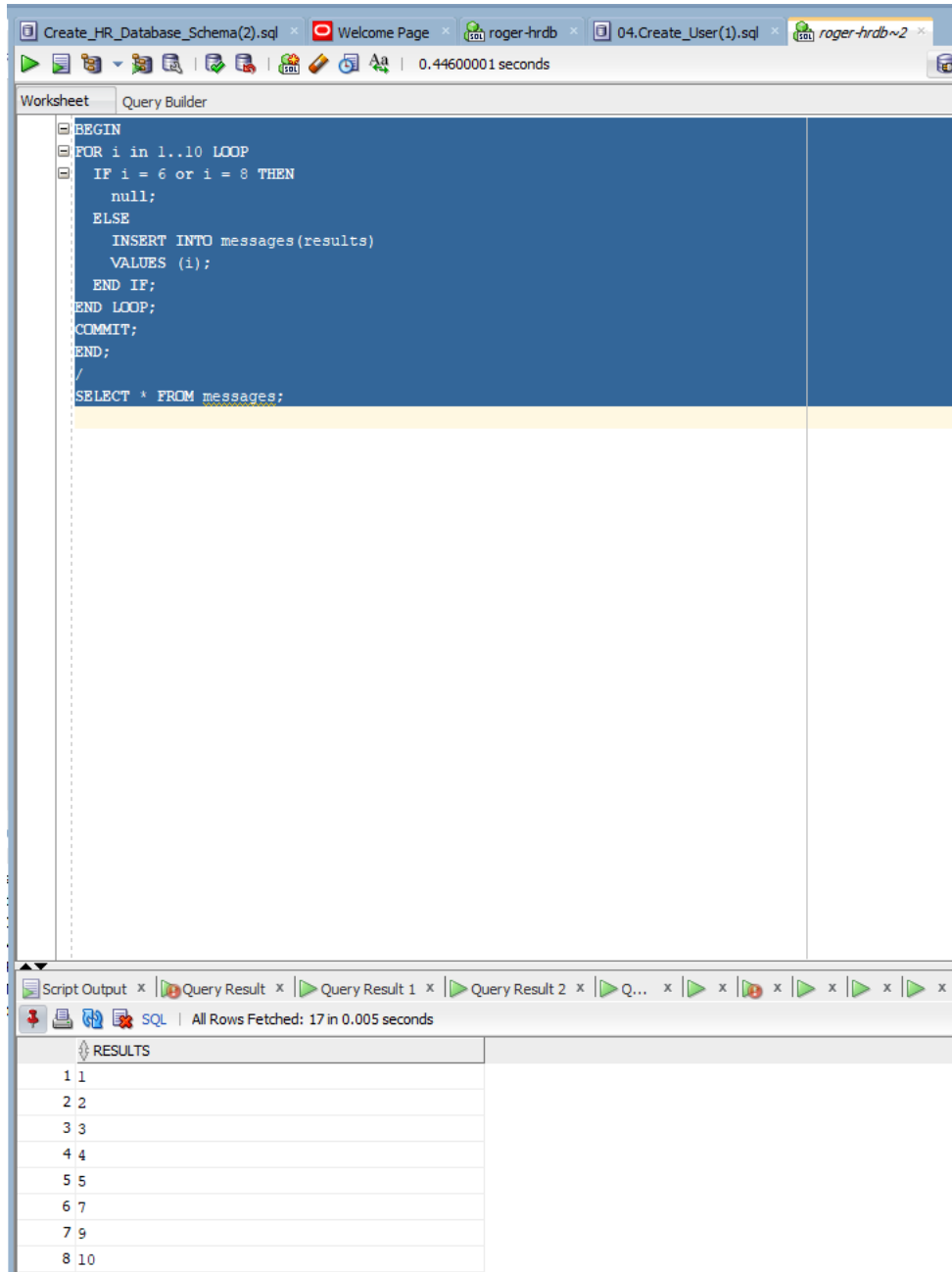


Note: this screenshot shows that I've output the successful drop of the table employee2

PL/SQL practice 6

Figure43

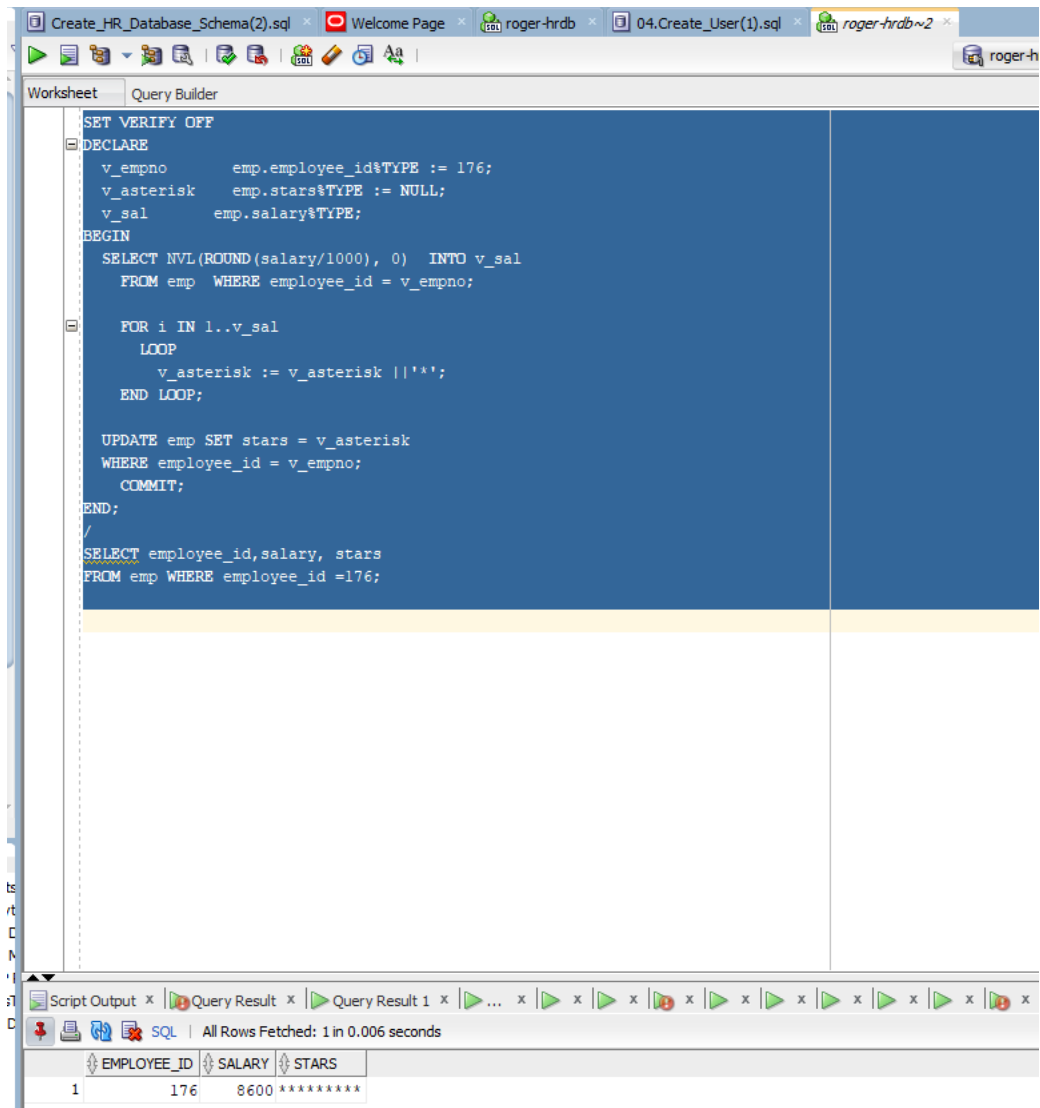
The SQL Developer output of pl/sql sql practice 6-1



Note: this screenshot shows that I've output the required number from 1-10 except 6,8

Figure44

The SQL Developer output of pl/sql sql practice 6-2

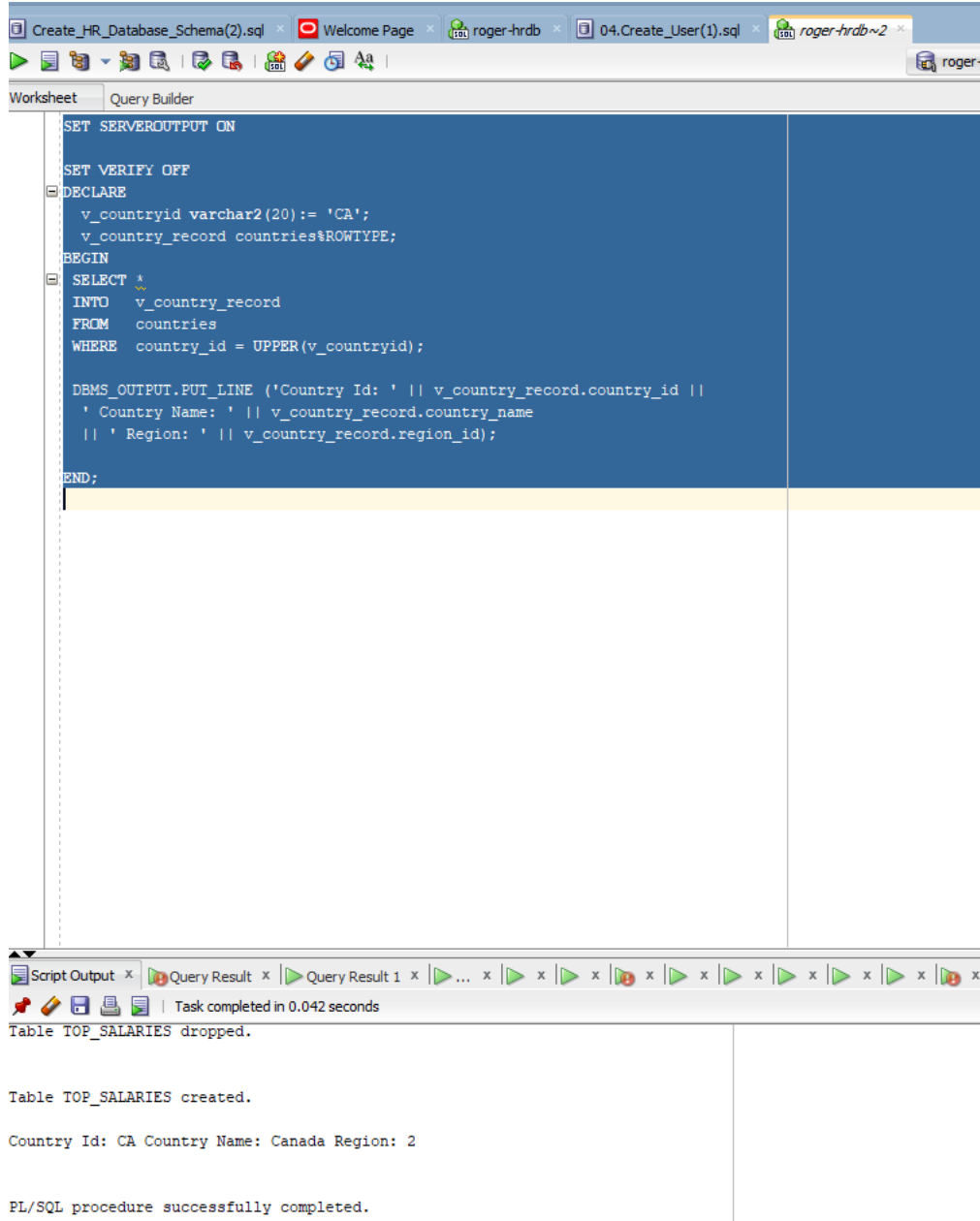


Note: this screenshot shows that I've output the employee id, salary and starts

PL/SQL practice 7

Figure45

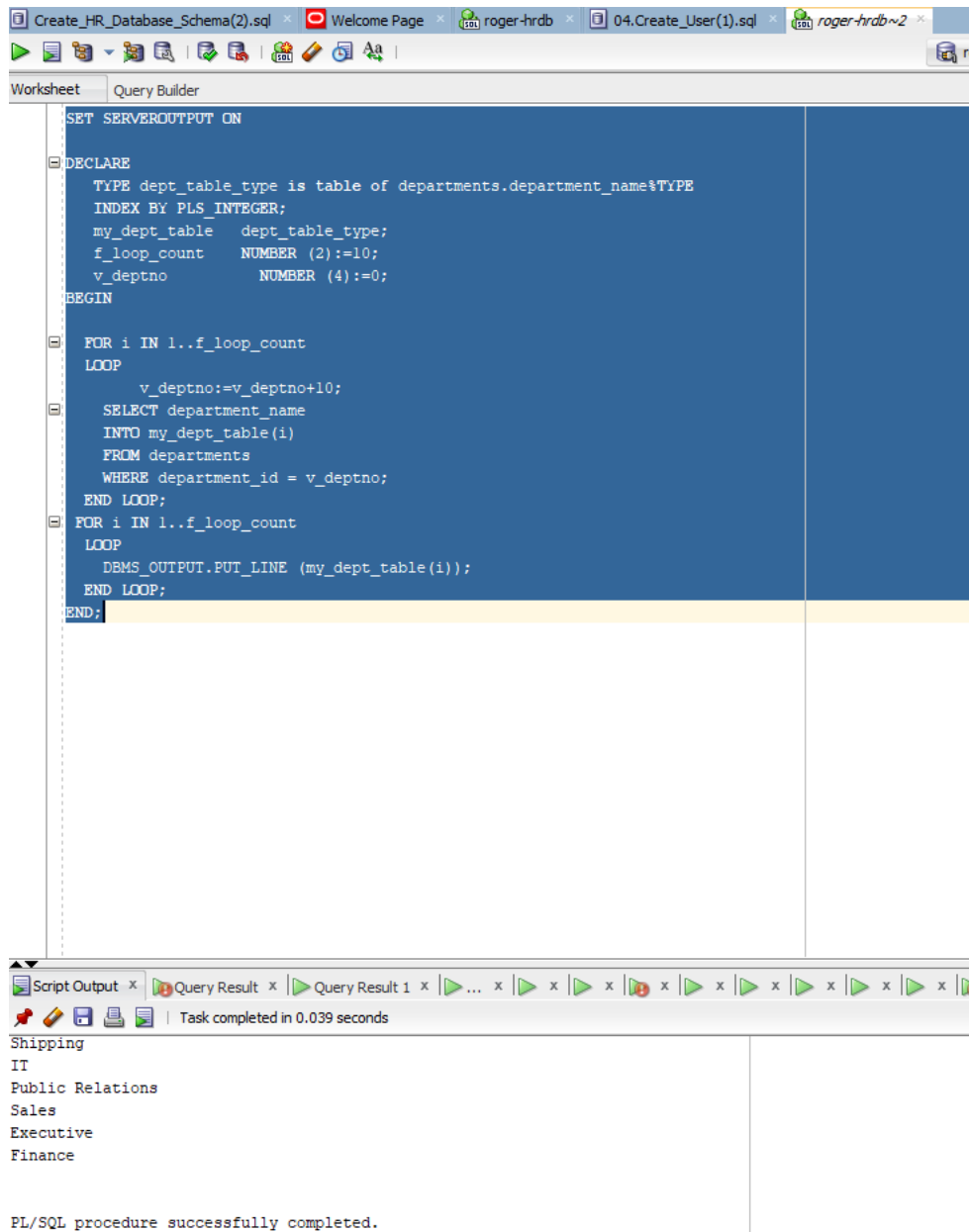
The SQL Developer output of pl/sql sql practice 7-1



Note: this screenshot shows that I've output the required country information

Figure46

The SQL Developer output of pl/sql sql practice 7-2



The screenshot displays the SQL Developer interface. The top toolbar includes icons for running and saving scripts. The main workspace is divided into two panes. The left pane, titled 'Worksheet', contains a PL/SQL procedure named 'my_dept_table' with the following code:

```
SET SERVEROUTPUT ON

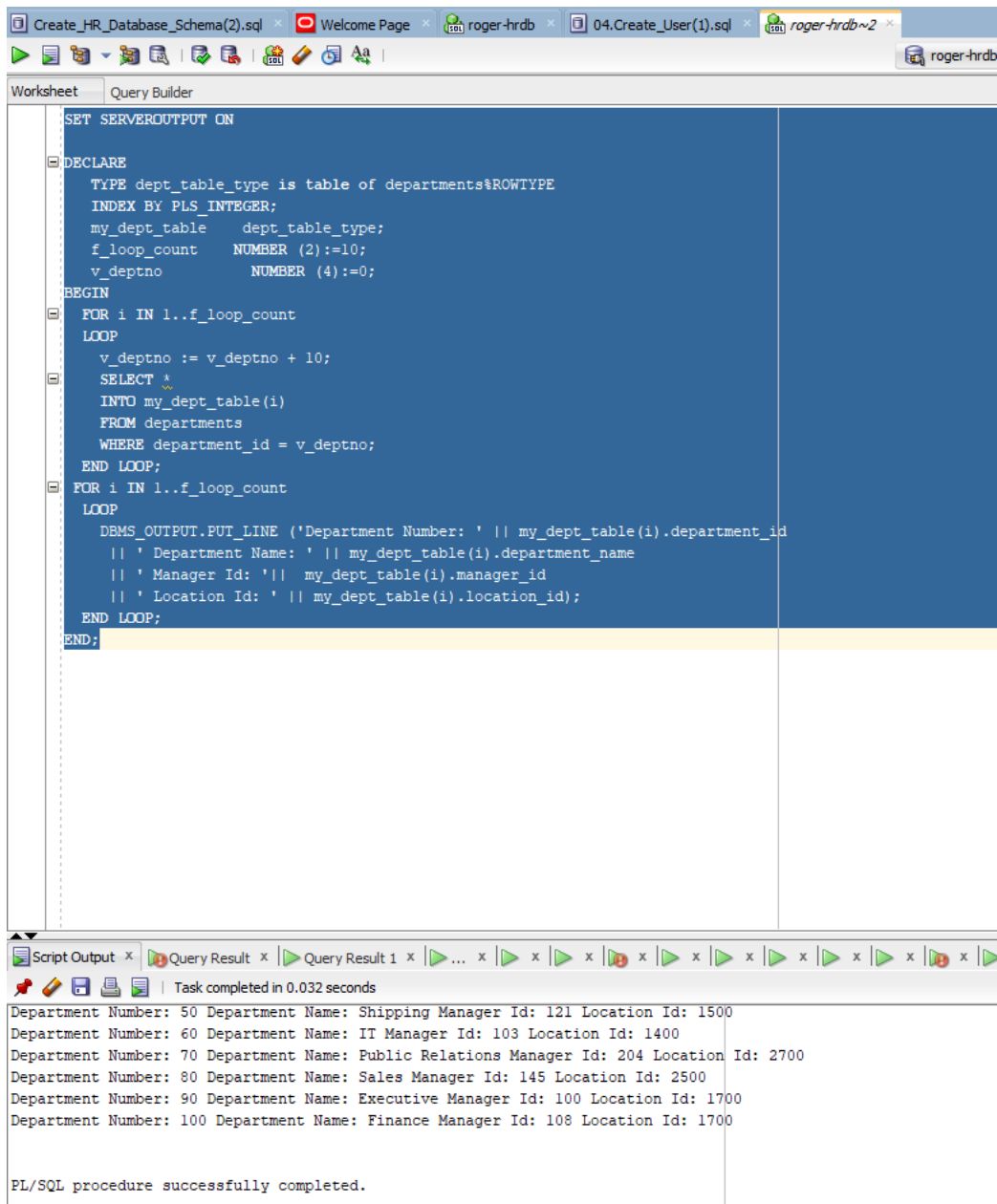
DECLARE
  TYPE dept_table_type is table of departments.department_name%TYPE
  INDEX BY PLS_INTEGER;
  my_dept_table    dept_table_type;
  f_loop_count     NUMBER (2):=10;
  v_deptno         NUMBER (4):=0;
BEGIN
  FOR i IN 1..f_loop_count
  LOOP
    v_deptno:=v_deptno+10;
    SELECT department_name
    INTO my_dept_table(i)
    FROM departments
    WHERE department_id = v_deptno;
  END LOOP;
  FOR i IN 1..f_loop_count
  LOOP
    DBMS_OUTPUT.PUT_LINE (my_dept_table(i));
  END LOOP;
END;
```

The right pane, titled 'Query Result', shows the output of the procedure. It lists five department names: Shipping, IT, Public Relations, Sales, and Executive. Below the list, a status message reads: 'PL/SQL procedure successfully completed.'

Note: this screenshot shows that I've output the required country information

Figure47

The SQL Developer output of pl/sql sql practice 7-3



```
SET SERVEROUTPUT ON

DECLARE
  TYPE dept_table_type is table of departments%ROWTYPE
  INDEX BY PLS_INTEGER;
  my_dept_table      dept_table_type;
  f_loop_count       NUMBER (2):=10;
  v_deptno           NUMBER (4):=0;
BEGIN
  FOR i IN 1..f_loop_count
  LOOP
    v_deptno := v_deptno + 10;
    SELECT *
    INTO my_dept_table(i)
    FROM departments
    WHERE department_id = v_deptno;
  END LOOP;
  FOR i IN 1..f_loop_count
  LOOP
    DBMS_OUTPUT.PUT_LINE ('Department Number: ' || my_dept_table(i).department_id
      || ' Department Name: ' || my_dept_table(i).department_name
      || ' Manager Id: ' || my_dept_table(i).manager_id
      || ' Location Id: ' || my_dept_table(i).location_id);
  END LOOP;
END;
```

Task completed in 0.032 seconds

Department Number: 50 Department Name: Shipping Manager Id: 121 Location Id: 1500
Department Number: 60 Department Name: IT Manager Id: 103 Location Id: 1400
Department Number: 70 Department Name: Public Relations Manager Id: 204 Location Id: 2700
Department Number: 80 Department Name: Sales Manager Id: 145 Location Id: 2500
Department Number: 90 Department Name: Executive Manager Id: 100 Location Id: 1700
Department Number: 100 Department Name: Finance Manager Id: 108 Location Id: 1700

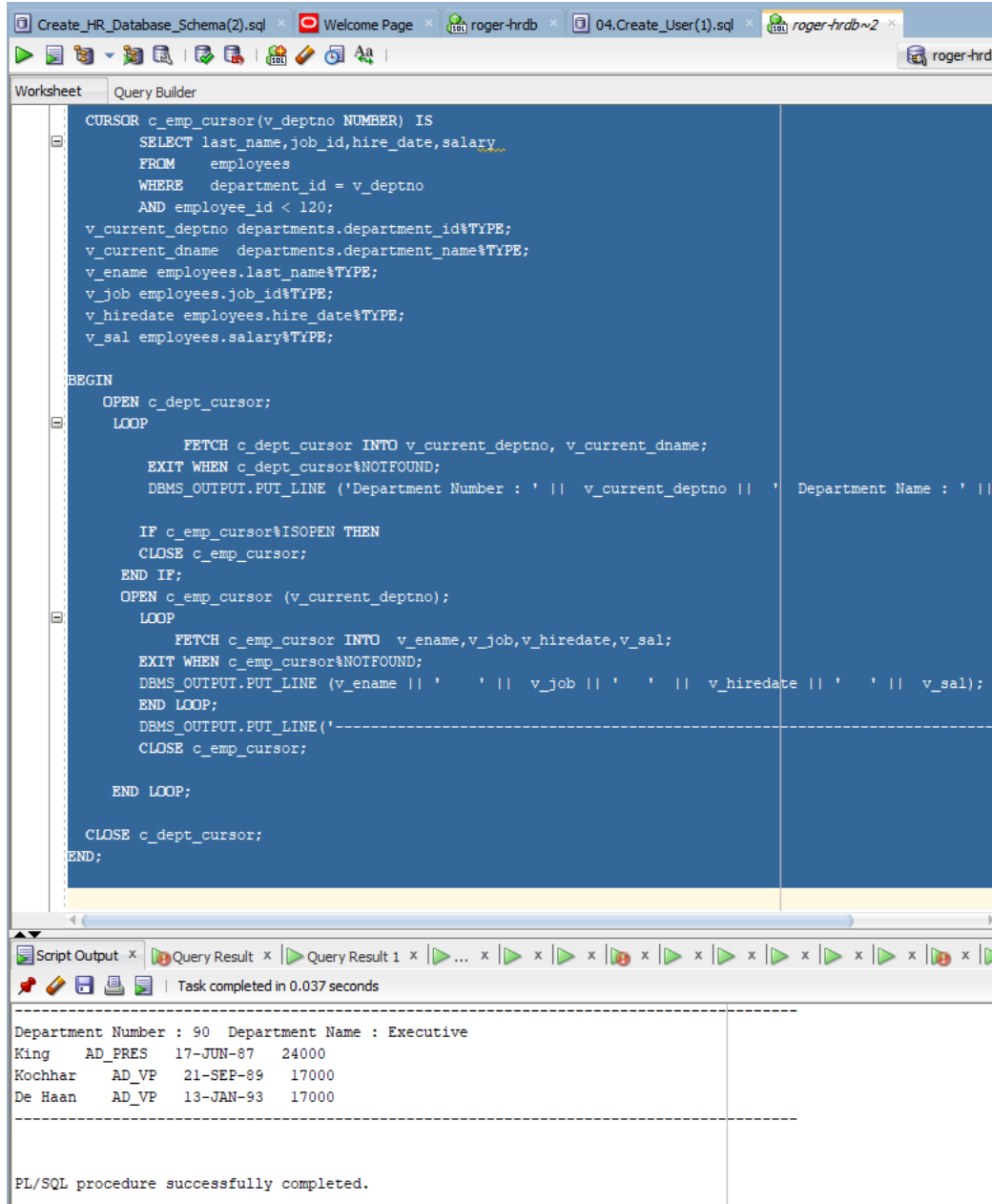
PL/SQL procedure successfully completed.

Note: this screenshot shows that I've output the required department information

PL/SQL practice 8

Figure48

The SQL Developer output of pl/sql sql practice 8-1



The screenshot displays the SQL Developer interface with a PL/SQL procedure named `c_emp_cursor` defined and executed. The procedure iterates through departments and employees, outputting their details. The output window shows the results for Department 90 (Executive), listing employees King, Kochhar, and De Haan with their respective job titles, hire dates, and salaries.

```
CURSOR c_emp_cursor(v_deptno NUMBER) IS
    SELECT last_name,job_id,hire_date,salary
    FROM    employees
    WHERE   department_id = v_deptno
    AND employee_id < 120;
v_current_deptno departments.department_id%TYPE;
v_current_dname  departments.department_name%TYPE;
v_name employees.last_name%TYPE;
v_job employees.job_id%TYPE;
v_hiredate employees.hire_date%TYPE;
v_sal employees.salary%TYPE;

BEGIN
    OPEN c_dept_cursor;
    LOOP
        FETCH c_dept_cursor INTO v_current_deptno, v_current_dname;
        EXIT WHEN c_dept_cursor%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE ('Department Number : ' || v_current_deptno || ' Department Name : ' || v_current_dname);

        IF c_emp_cursor%ISOPEN THEN
            CLOSE c_emp_cursor;
        END IF;
        OPEN c_emp_cursor (v_current_deptno);
        LOOP
            FETCH c_emp_cursor INTO v_name,v_job,v_hiredate,v_sal;
            EXIT WHEN c_emp_cursor%NOTFOUND;
            DBMS_OUTPUT.PUT_LINE (v_name || ' ' || v_job || ' ' || v_hiredate || ' ' || v_sal);
        END LOOP;
        DBMS_OUTPUT.PUT_LINE('-----');
        CLOSE c_emp_cursor;

    END LOOP;

    CLOSE c_dept_cursor;
END;
```

Task completed in 0.037 seconds

Department Number : 90 Department Name : Executive				
King	AD_PRES	17-JUN-87	24000	
Kochhar	AD_VP	21-SEP-89	17000	
De Haan	AD_VP	13-JAN-93	17000	

PL/SQL procedure successfully completed.

Note: this screenshot shows that I've output the required department information

Figure49

The SQL Developer output of pl/sql sql practice 8-2

The screenshot displays the SQL Developer interface. The main window shows a PL/SQL script in the 'Query Builder' tab. The script performs the following actions:

- SET VERIFY OFF
- DELETE FROM top_salaries;
- DECLARE
- v_num NUMBER(3) := 5;
- v_sal employees.salary%TYPE;
- CURSOR c_emp_cursor IS
- SELECT salary
- FROM employees
- ORDER BY salary DESC;
- BEGIN
- OPEN c_emp_cursor;
- FETCH c_emp_cursor INTO v_sal;
- WHILE c_emp_cursor%ROWCOUNT <= v_num AND c_emp_cursor%FOUND LOOP
- INSERT INTO top_salaries (salary)
- VALUES (v_sal);
- FETCH c_emp_cursor INTO v_sal;
- END LOOP;
- CLOSE c_emp_cursor;
- END;
- /
- SELECT * FROM top_salaries;

The bottom of the window shows the 'Script Output' and 'Query Result' tabs. The 'Query Result' tab displays the output of the final SELECT statement, showing 5 rows of salary data:

	SALARY
1	24000
2	17000
3	17000
4	14000
5	13500

Note: this screenshot shows that I've output the required top salary

PL/SQL practice 9

Figure50

The SQL Developer output of pl/sql sql practice 9-1

The screenshot displays the SQL Developer interface with a PL/SQL script in the main editor and its execution results in the bottom pane.

Script Content:

```
SET VERIFY OFF
DECLARE
  v_ename employees.last_name%TYPE;
  v_emp_sal employees.salary%TYPE := 6000;
BEGIN
  SELECT last_name
  INTO v_ename
  FROM employees
  WHERE salary = v_emp_sal;

  INSERT INTO messages (results)
  VALUES (v_ename || ' - ' || v_emp_sal);

EXCEPTION
  WHEN no_data_found THEN
    INSERT INTO messages (results)
    VALUES ('No employee with a salary of ' || TO_CHAR(v_emp_sal));
  WHEN too_many_rows THEN
    INSERT INTO messages (results)
    VALUES ('More than one employee with a salary of ' ||
            TO_CHAR(v_emp_sal));
  WHEN others THEN
    INSERT INTO messages (results)
    VALUES ('Some other error occurred.');
```

Execution Results:

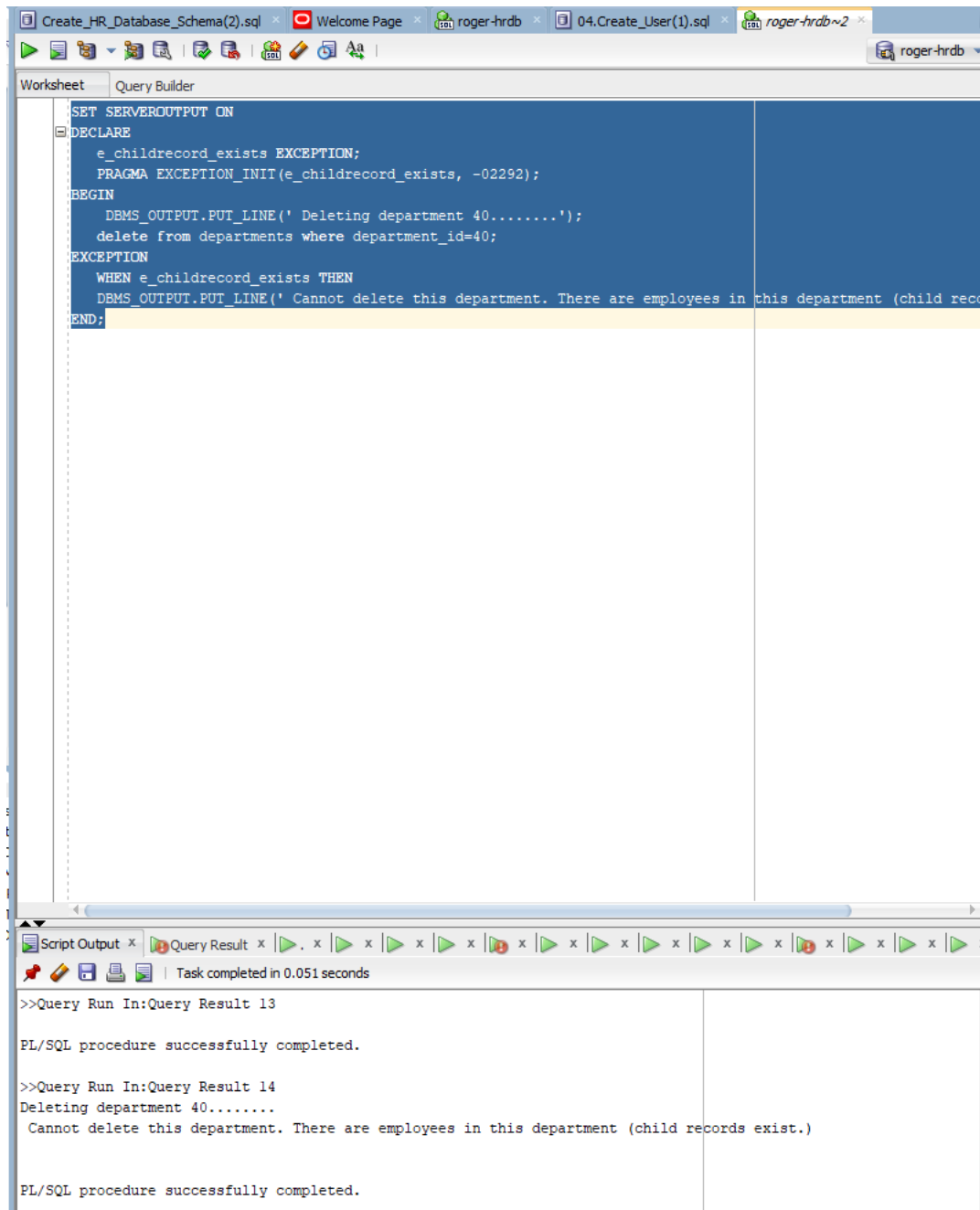
Script Output x Query Result x All Rows Fetched: 18 in 0.004 seconds

RESULTS
1 1
2 2
3 3
4 4
5 5
6 7
7 9
8 10
9 More than one employee with a salary of 6000

Note: this screenshot shows that I've output the required exception

Figure51

The SQL Developer output of pl/sql sql practice 9-2

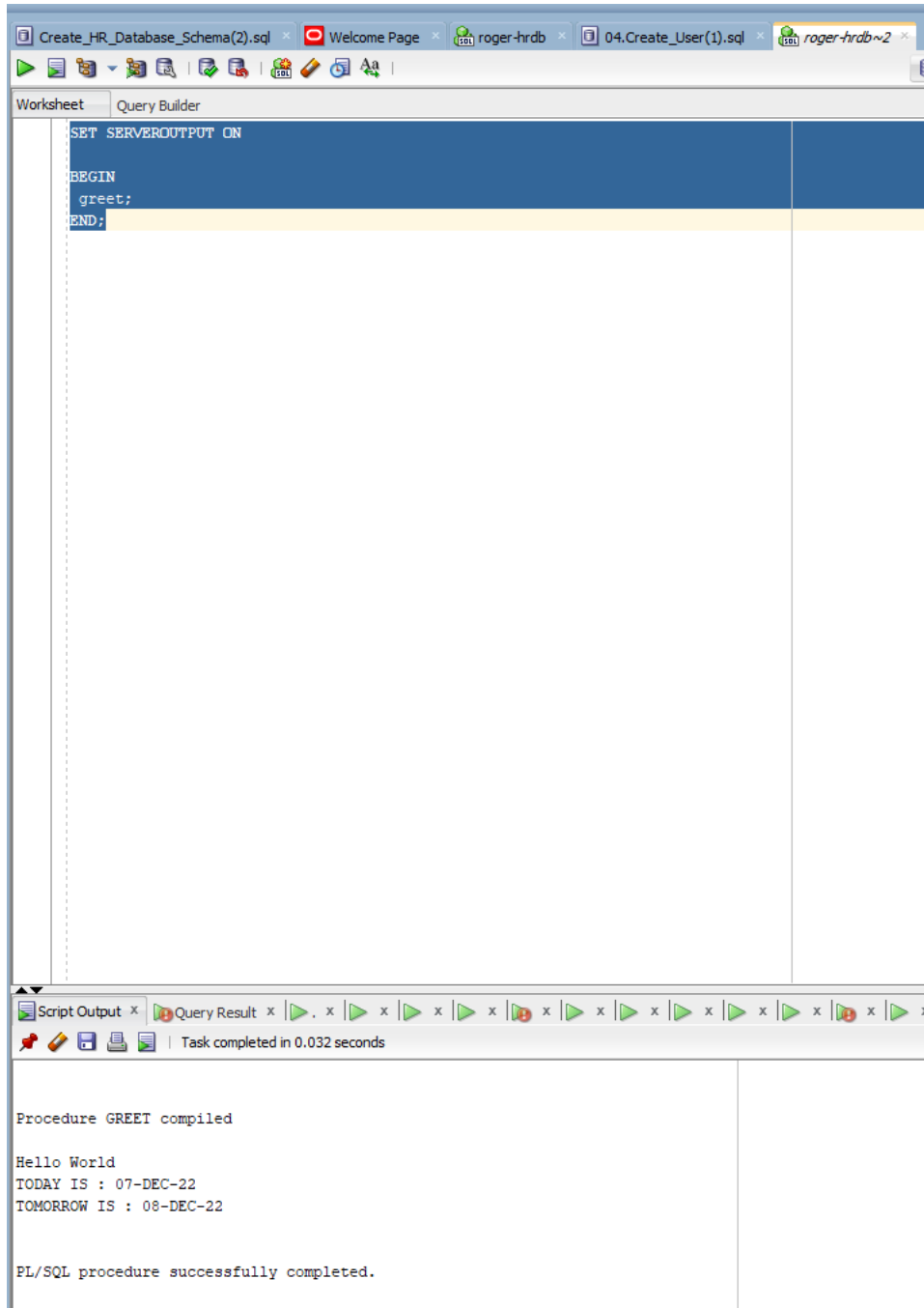


Note: this screenshot shows that I've output the department can't be deleted because it has a child table

PL/SQL practice 10

Figure52

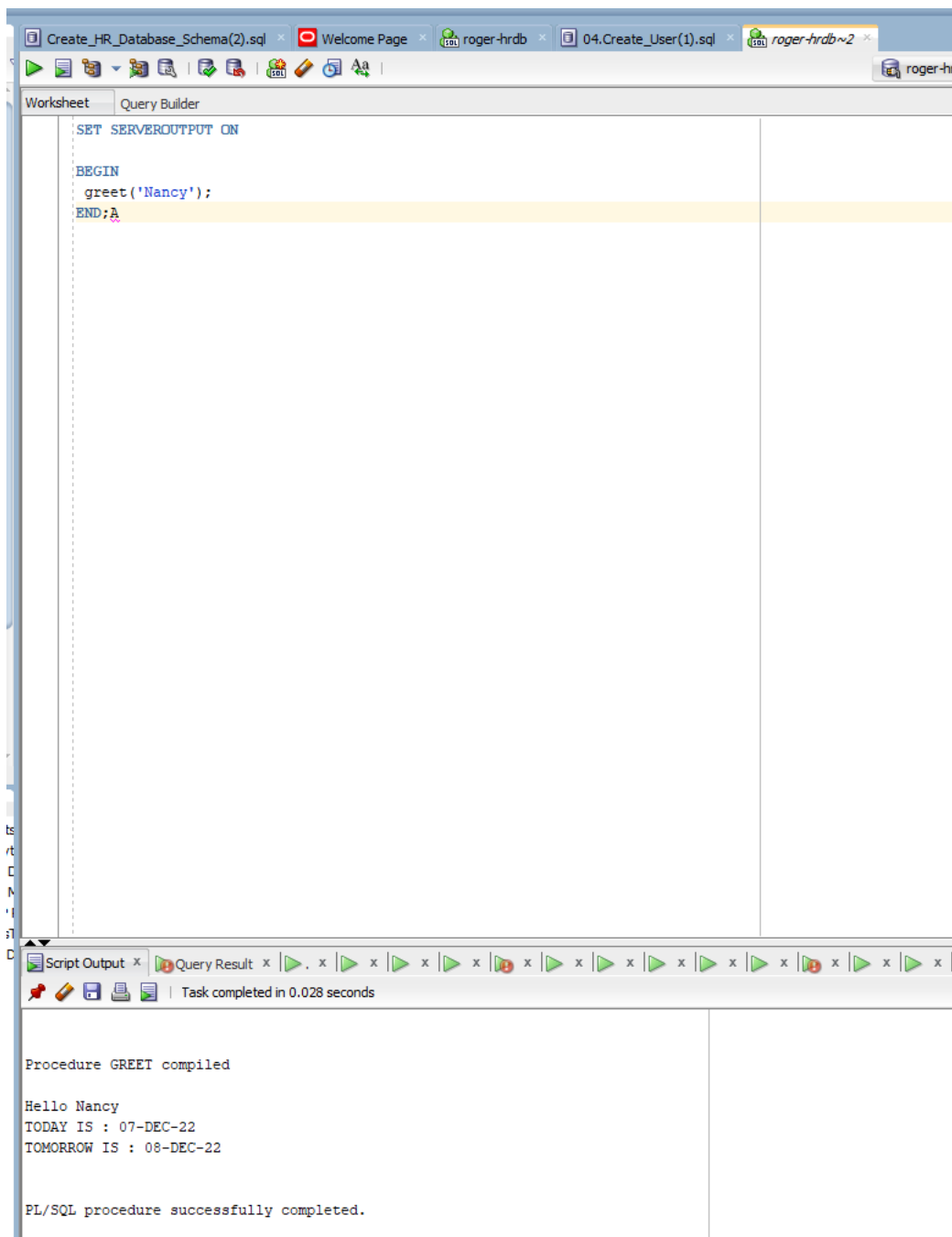
The SQL Developer output of pl/sql sql practice 10-1



Note: this screenshot shows that I've created the greet procedure and output the greet message

Figure53

The SQL Developer output of pl/sql sql practice 10-2



Note: this screenshot shows that I've recreated the greet procedure and output the greet message for nancy