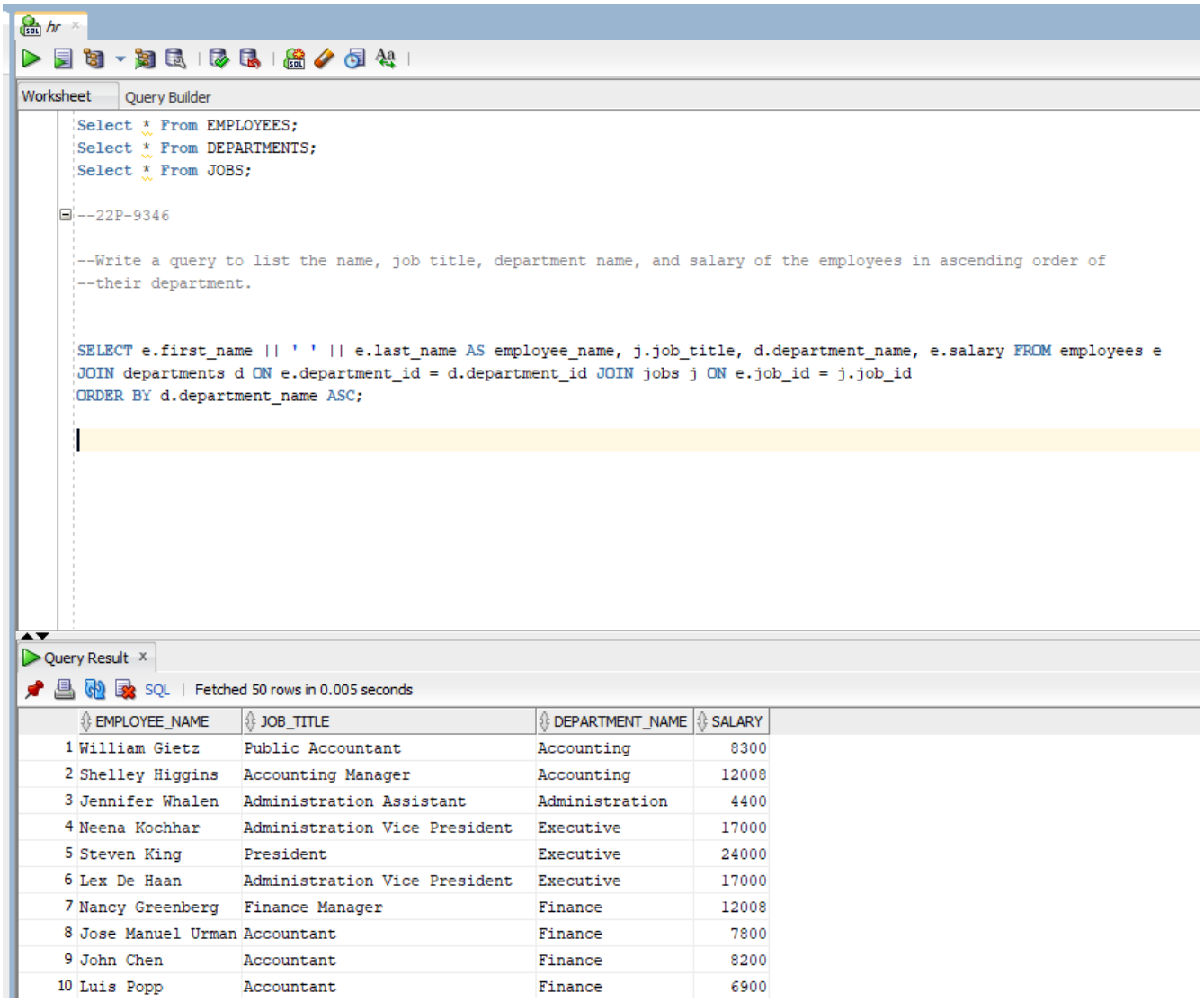
Lab 5

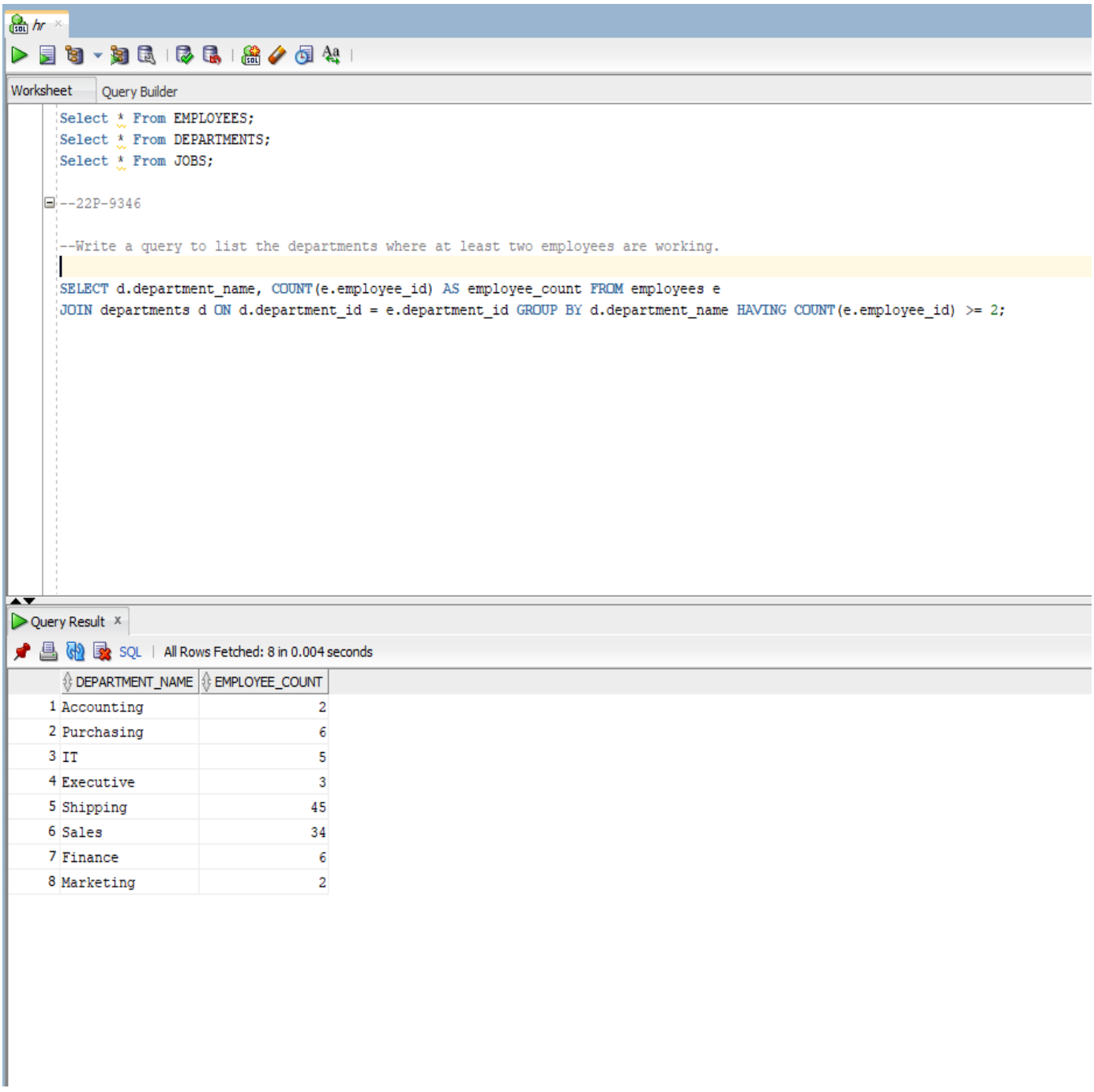


```
Select * From EMPLOYEES;
Select * From DEPARTMENTS;
Select * From JOBS;

--22P-9346

--Write a query to list the name, job title, department name, and salary of the employees in ascending order of
--their department.


SELECT e.first_name || ' ' || e.last_name AS employee_name, j.job_title, d.department_name, e.salary FROM employees e
JOIN departments d ON e.department_id = d.department_id JOIN jobs j ON e.job_id = j.job_id
ORDER BY d.department_name ASC;
```

Query Result

SQL | Fetched 50 rows in 0.005 seconds

	EMPLOYEE_NAME	JOB_TITLE	DEPARTMENT_NAME	SALARY
1	William Gietz	Public Accountant	Accounting	8300
2	Shelley Higgins	Accounting Manager	Accounting	12008
3	Jennifer Whalen	Administration Assistant	Administration	4400
4	Neena Kochhar	Administration Vice President	Executive	17000
5	Steven King	President	Executive	24000
6	Lex De Haan	Administration Vice President	Executive	17000
7	Nancy Greenberg	Finance Manager	Finance	12008
8	Jose Manuel Urman	Accountant	Finance	7800
9	John Chen	Accountant	Finance	8200
10	Luis Popp	Accountant	Finance	6900



hr

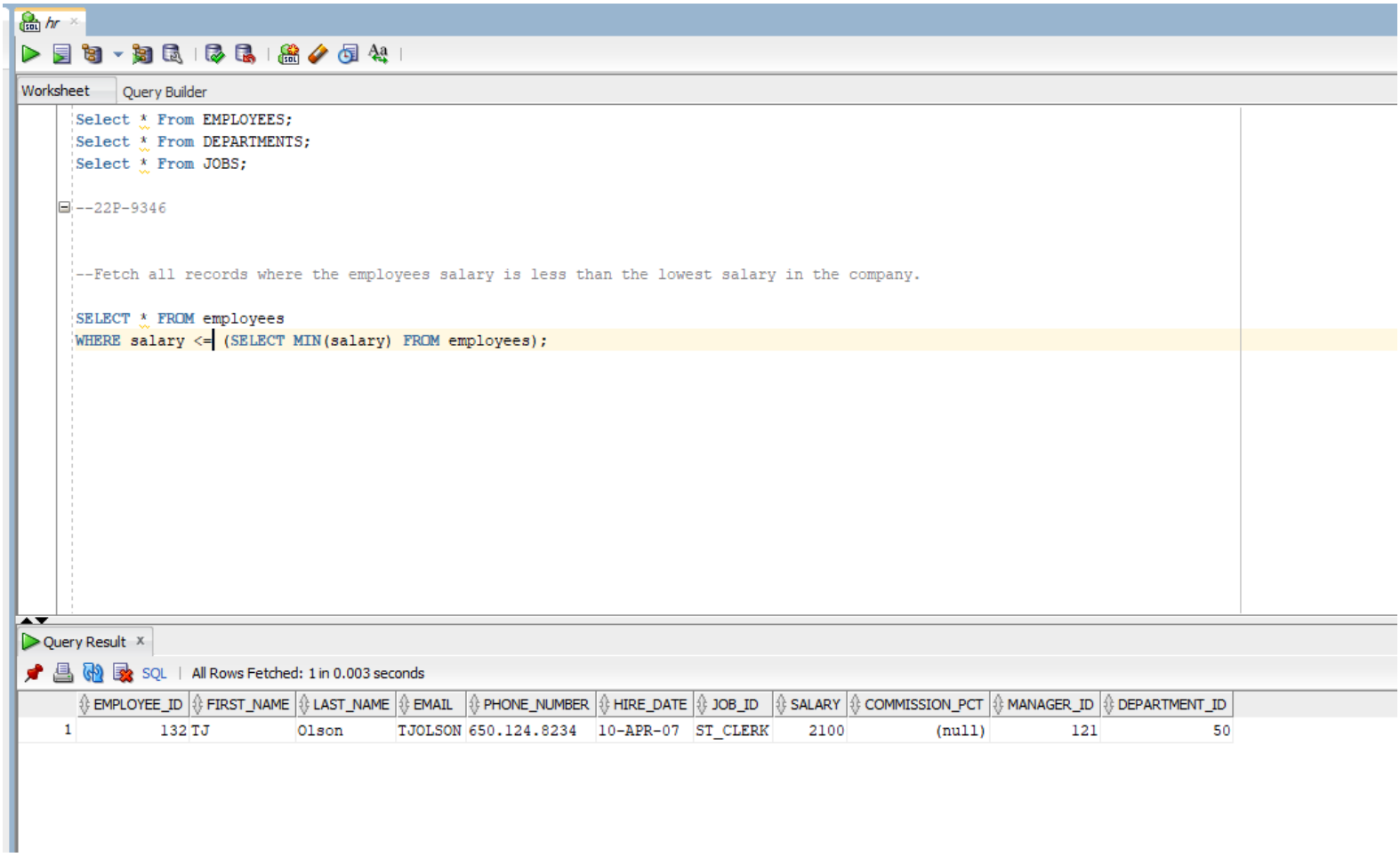
Worksheet | Query Builder

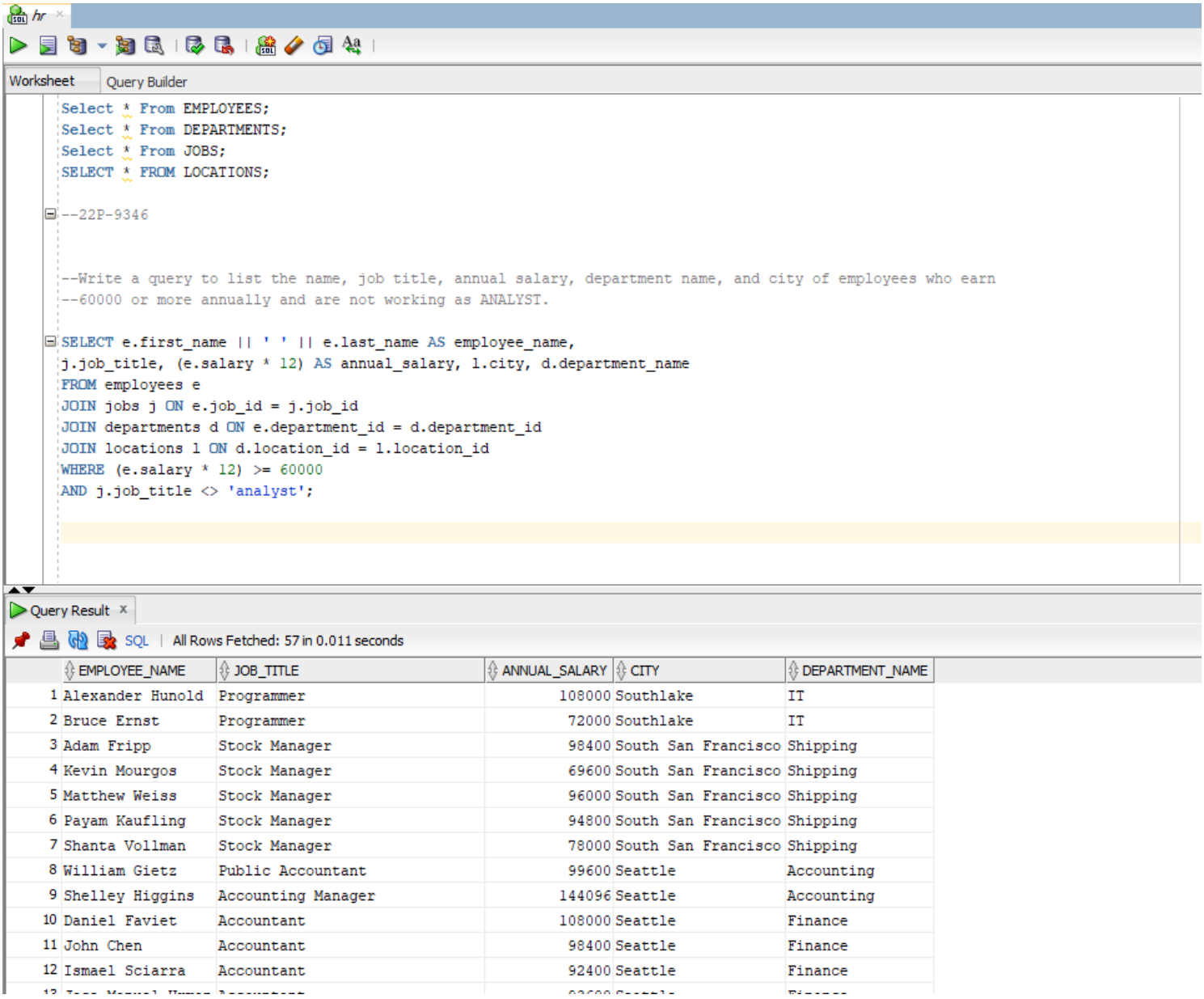
```
Select * From EMPLOYEES;
Select * From DEPARTMENTS;
Select * From JOBS;

--22P-9346

--Write a query to list the departments where at least two employees are working.

SELECT d.department_name, COUNT(e.employee_id) AS employee_count FROM employees e
JOIN departments d ON d.department_id = e.department_id GROUP BY d.department_name HAVING COUNT(e.employee_id) >= 2;
```

Query Result

SQL | All Rows Fetched: 8 in 0.004 seconds

	DEPARTMENT_NAME	EMPLOYEE_COUNT
1	Accounting	2
2	Purchasing	6
3	IT	5
4	Executive	3
5	Shipping	45
6	Sales	34
7	Finance	6
8	Marketing	2


hr
Worksheet
Query Builder
Select * From EMPLOYEES;
Select * From DEPARTMENTS;
Select * From JOBS;
--22P-9346
--Fetch all records where the employees salary is less than the lowest salary in the company.
SELECT * FROM employees
WHERE salary <= (SELECT MIN(salary) FROM employees);
Query Result
SQL
All Rows Fetched: 1 in 0.003 seconds
EMPLOYEE_ID FIRST_NAME LAST_NAME EMAIL PHONE_NUMBER HIRE_DATE JOB_ID SALARY COMMISSION_PCT MANAGER_ID DEPARTMENT_ID
1 132 TJ Olson TJOLSON 650.124.8234 10-APR-07 ST_CLERK 2100 (null) 121 50



hr

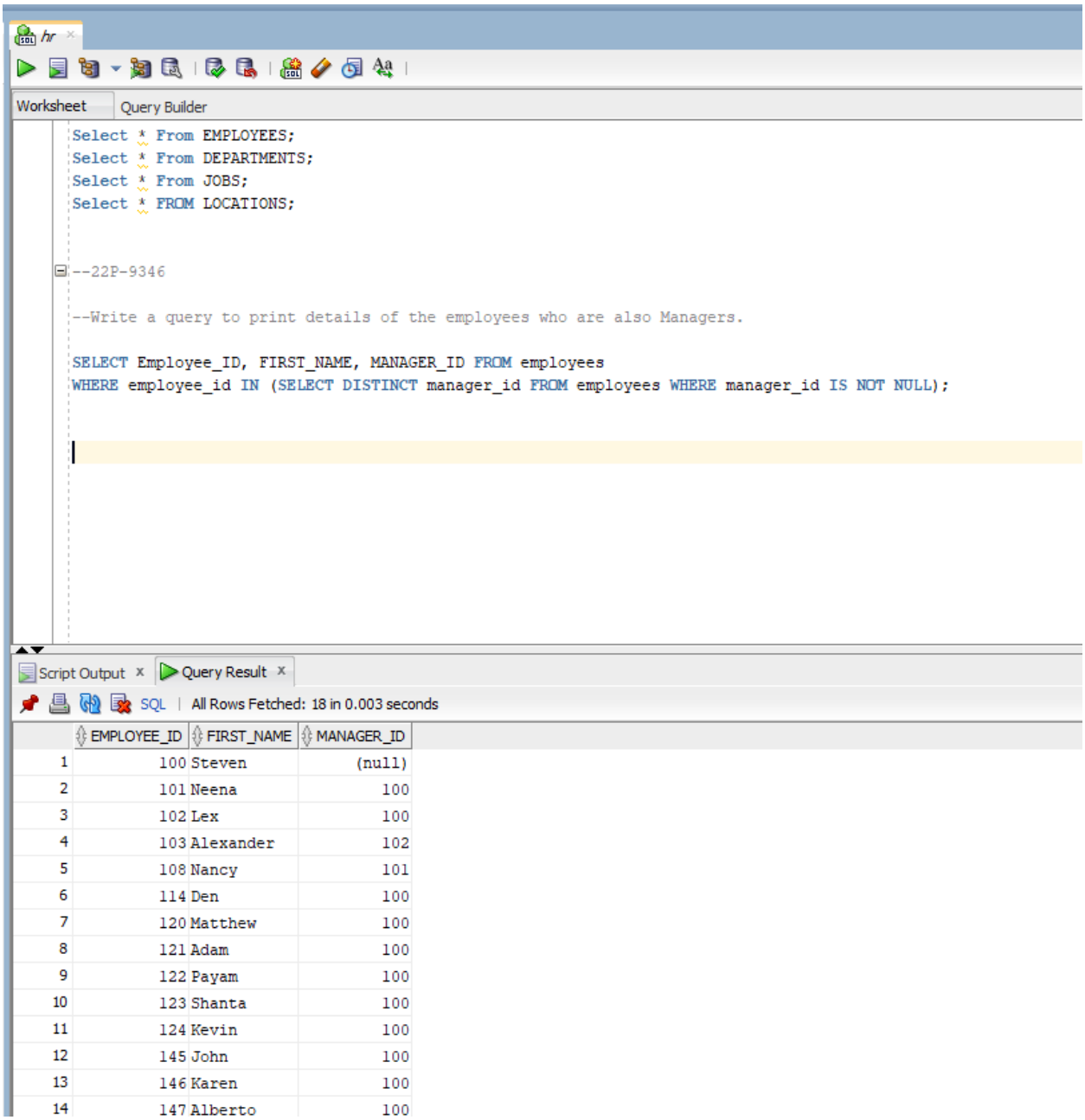
Worksheet | Query Builder

```
Select * From EMPLOYEES;
Select * From DEPARTMENTS;
Select * From JOBS;
SELECT * FROM LOCATIONS;

--22P-9346


--Write a query to list the name, job title, annual salary, department name, and city of employees who earn
--60000 or more annually and are not working as ANALYST.

SELECT e.first_name || ' ' || e.last_name AS employee_name,
j.job_title, (e.salary * 12) AS annual_salary, l.city, d.department_name
FROM employees e
JOIN jobs j ON e.job_id = j.job_id
JOIN departments d ON e.department_id = d.department_id
JOIN locations l ON d.location_id = l.location_id
WHERE (e.salary * 12) >= 60000
AND j.job_title <> 'analyst';
```

Query Result

SQL | All Rows Fetched: 57 in 0.011 seconds

	EMPLOYEE_NAME	JOB_TITLE	ANNUAL_SALARY	CITY	DEPARTMENT_NAME
1	Alexander Hunold	Programmer	108000	Southlake	IT
2	Bruce Ernst	Programmer	72000	Southlake	IT
3	Adam Fripp	Stock Manager	98400	South San Francisco	Shipping
4	Kevin Mourgos	Stock Manager	69600	South San Francisco	Shipping
5	Matthew Weiss	Stock Manager	96000	South San Francisco	Shipping
6	Payam Kaufling	Stock Manager	94800	South San Francisco	Shipping
7	Shanta Vollman	Stock Manager	78000	South San Francisco	Shipping
8	William Gietz	Public Accountant	99600	Seattle	Accounting
9	Shelley Higgins	Accounting Manager	144096	Seattle	Accounting
10	Daniel Faviet	Accountant	108000	Seattle	Finance
11	John Chen	Accountant	98400	Seattle	Finance
12	Ismael Sciarra	Accountant	92400	Seattle	Finance



hr

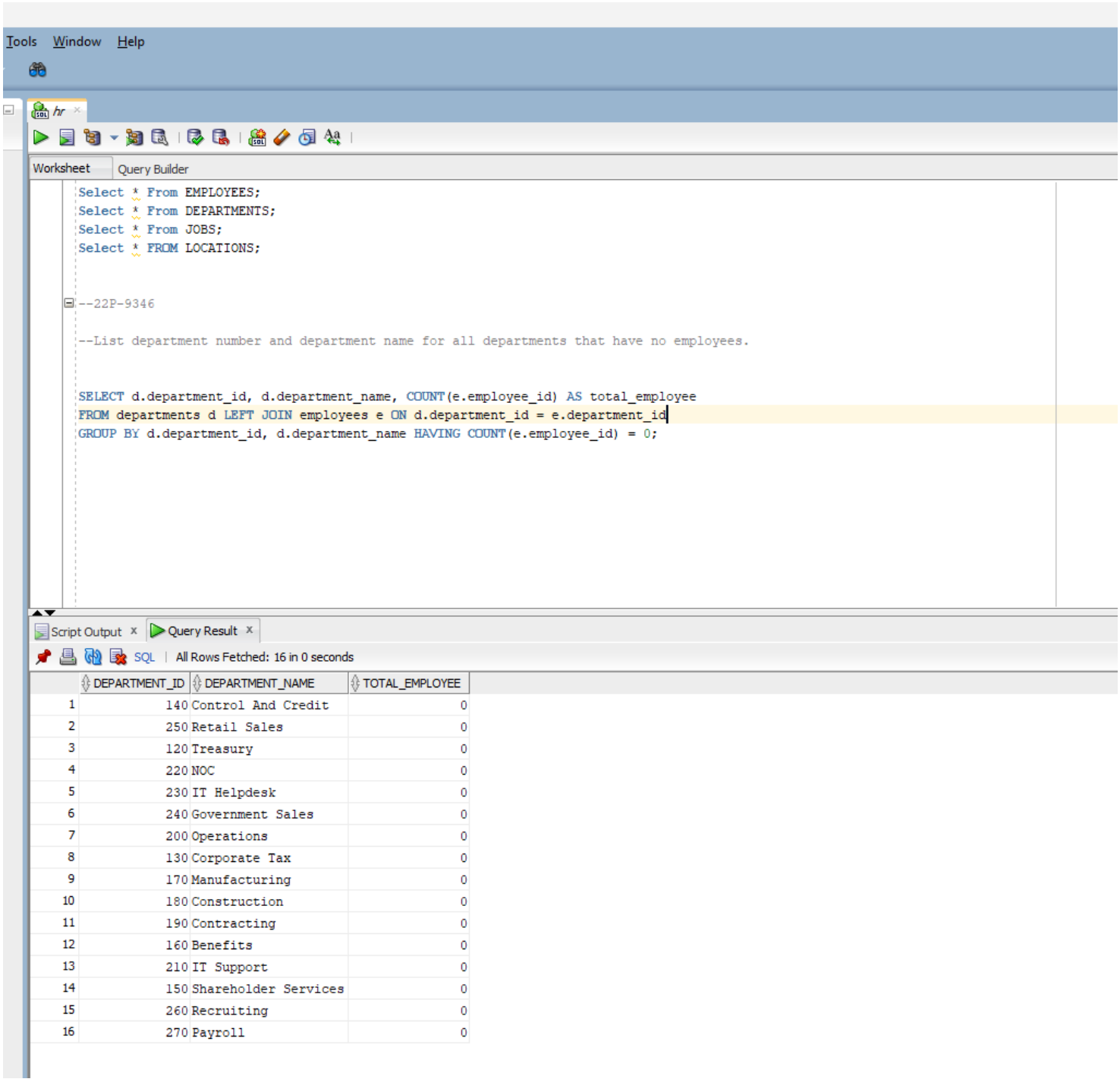
Worksheet | Query Builder

```
Select * From EMPLOYEES;
Select * From DEPARTMENTS;
Select * From JOBS;
Select * FROM LOCATIONS;


--22P-9346

--Write a query to print details of the employees who are also Managers.

SELECT Employee_ID, FIRST_NAME, MANAGER_ID FROM employees
WHERE employee_id IN (SELECT DISTINCT manager_id FROM employees WHERE manager_id IS NOT NULL);
```

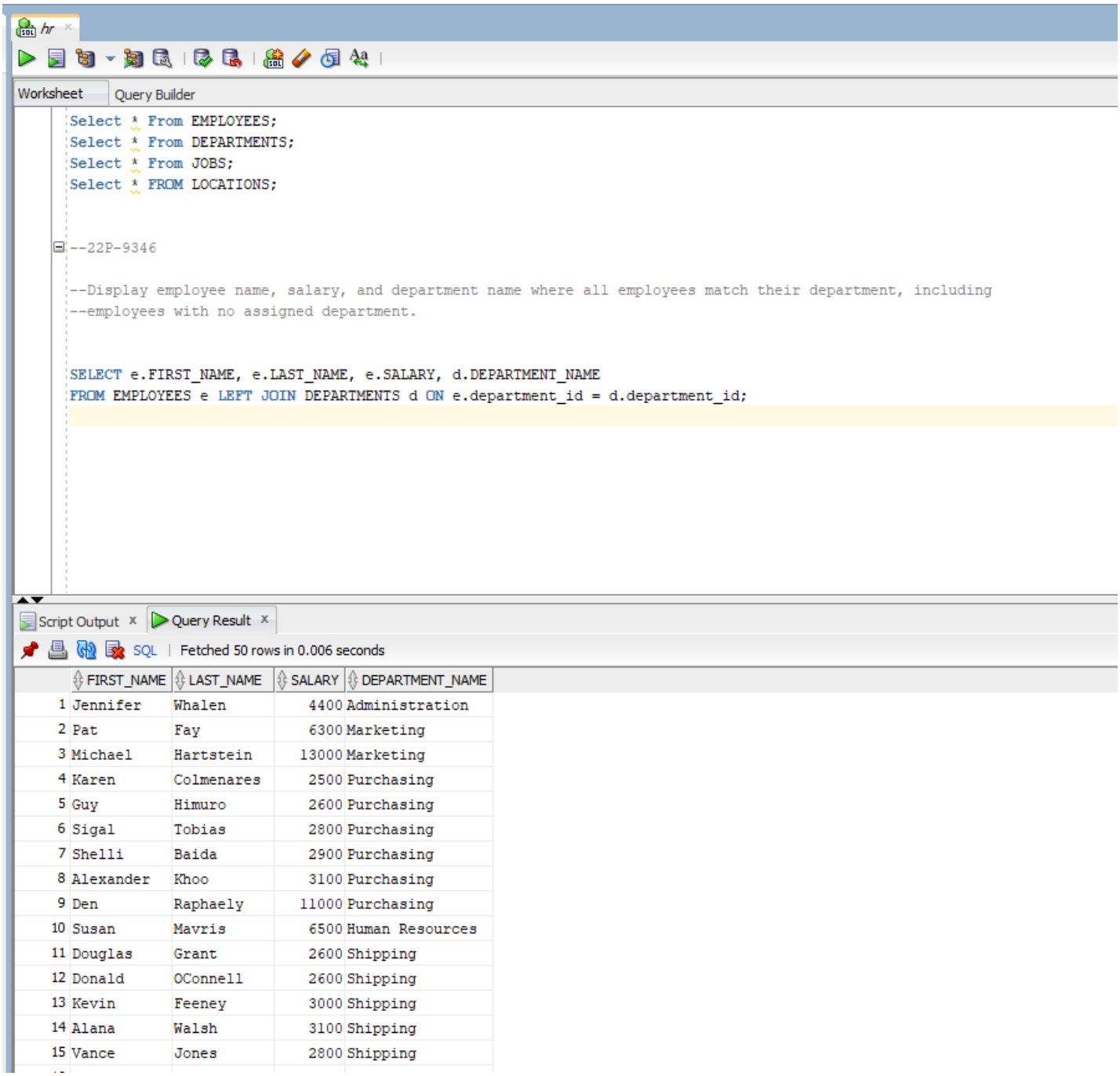
Script Output | Query Result

SQL | All Rows Fetched: 18 in 0.003 seconds

	EMPLOYEE_ID	FIRST_NAME	MANAGER_ID
1	100	Steven	(null)
2	101	Neena	100
3	102	Lex	100
4	103	Alexander	102
5	108	Nancy	101
6	114	Den	100
7	120	Matthew	100
8	121	Adam	100
9	122	Payam	100
10	123	Shanta	100
11	124	Kevin	100
12	145	John	100
13	146	Karen	100
14	147	Alberto	100



	DEPARTMENT_ID	DEPARTMENT_NAME	TOTAL_EMPLOYEE
1	140	Control And Credit	0
2	250	Retail Sales	0
3	120	Treasury	0
4	220	NOC	0
5	230	IT Helpdesk	0
6	240	Government Sales	0
7	200	Operations	0
8	130	Corporate Tax	0
9	170	Manufacturing	0
10	180	Construction	0
11	190	Contracting	0
12	160	Benefits	0
13	210	IT Support	0
14	150	Shareholder Services	0
15	260	Recruiting	0
16	270	Payroll	0



hr

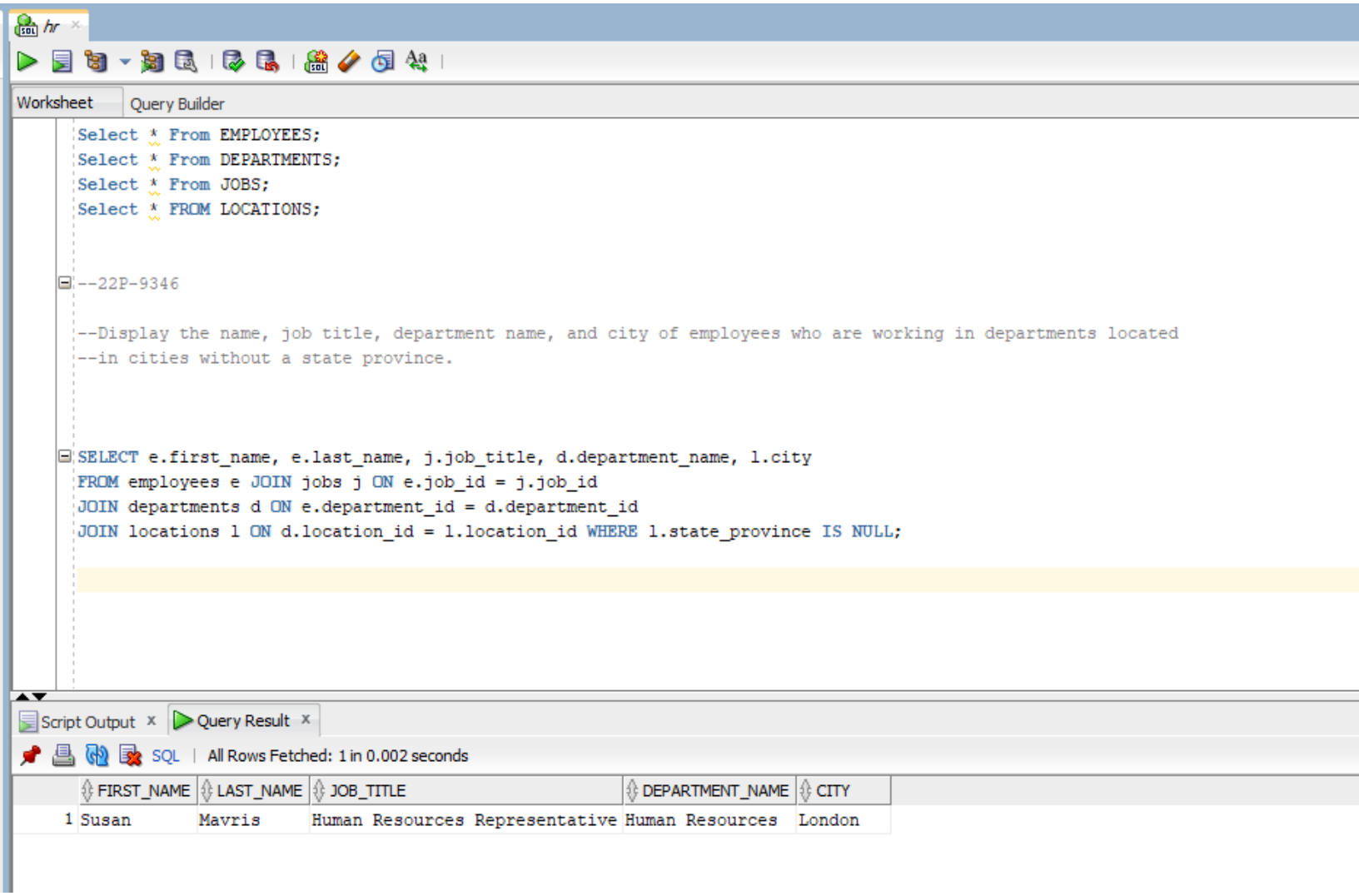
Worksheet | Query Builder

```
Select * From EMPLOYEES;
Select * From DEPARTMENTS;
Select * From JOBS;
Select * FROM LOCATIONS;


--22P-9346

--Display employee name, salary, and department name where all employees match their department, including
--employees with no assigned department.


SELECT e.FIRST_NAME, e.LAST_NAME, e.SALARY, d.DEPARTMENT_NAME
FROM EMPLOYEES e LEFT JOIN DEPARTMENTS d ON e.department_id = d.department_id;
```

Script Output | Query Result

SQL | Fetched 50 rows in 0.006 seconds

	FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_NAME
1	Jennifer	Whalen	4400	Administration
2	Pat	Fay	6300	Marketing
3	Michael	Hartstein	13000	Marketing
4	Karen	Colmenares	2500	Purchasing
5	Guy	Himuro	2600	Purchasing
6	Sigal	Tobias	2800	Purchasing
7	Shelli	Baida	2900	Purchasing
8	Alexander	Khoo	3100	Purchasing
9	Den	Raphaely	11000	Purchasing
10	Susan	Mavris	6500	Human Resources
11	Douglas	Grant	2600	Shipping
12	Donald	OConnell	2600	Shipping
13	Kevin	Feeney	3000	Shipping
14	Alana	Walsh	3100	Shipping
15	Vance	Jones	2800	Shipping


hr
Worksheet
Query Builder
Select * From EMPLOYEES;
Select * From DEPARTMENTS;
Select * From JOBS;
Select * FROM LOCATIONS;
--22P-9346
--Display the name, job title, department name, and city of employees who are working in departments located
--in cities without a state province.
SELECT e.first_name, e.last_name, j.job_title, d.department_name, l.city
FROM employees e JOIN jobs j ON e.job_id = j.job_id
JOIN departments d ON e.department_id = d.department_id
JOIN locations l ON d.location_id = l.location_id WHERE l.state_province IS NULL;
Script Output
Query Result
SQL
All Rows Fetched: 1 in 0.002 seconds
FIRST_NAME
LAST_NAME
JOB_TITLE
DEPARTMENT_NAME
CITY
1
Susan
Mavris
Human Resources Representative
Human Resources
London

hr

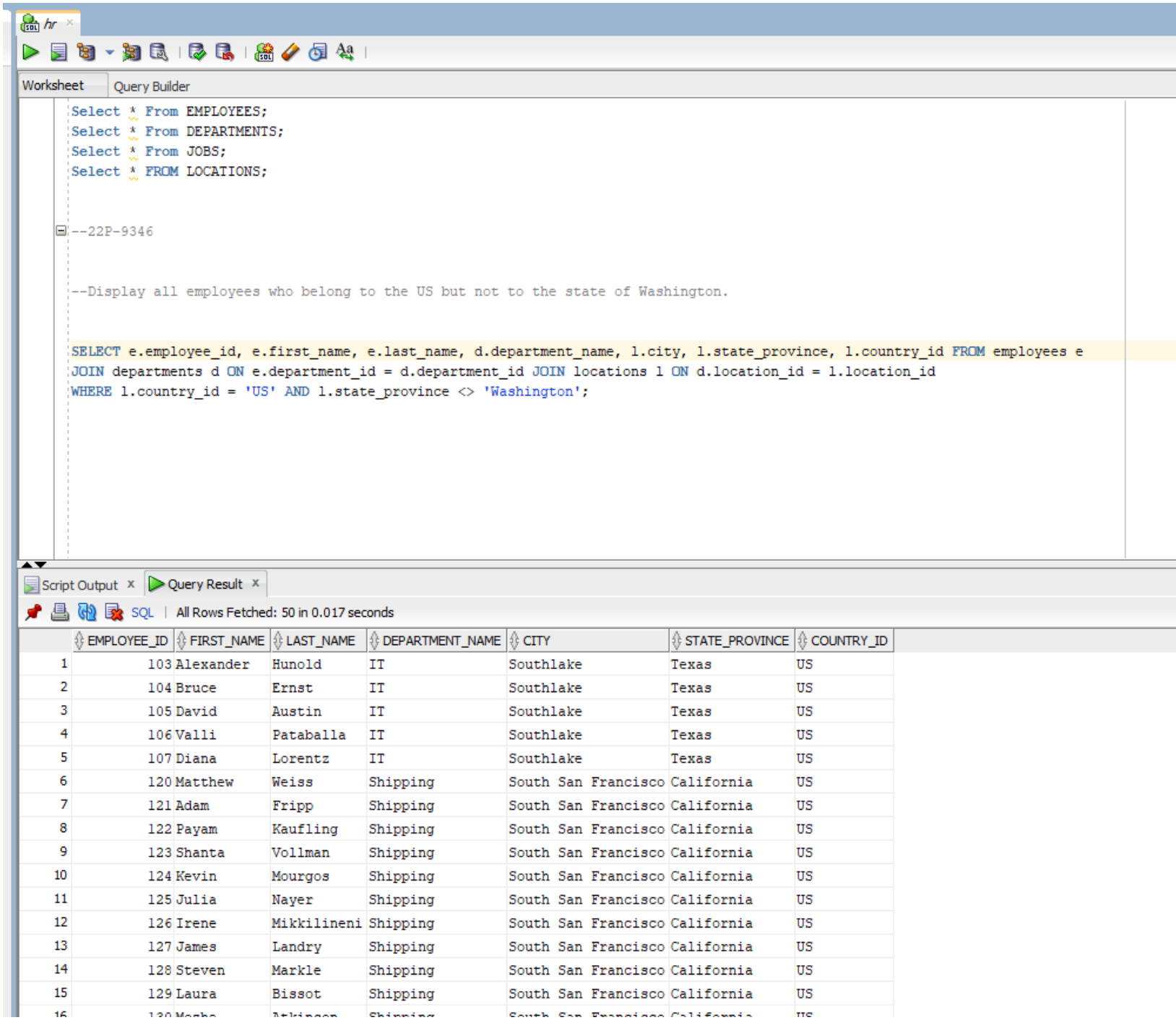
Worksheet | Query Builder

```
Select * From EMPLOYEES;
Select * From DEPARTMENTS;
Select * From JOBS;
Select * FROM LOCATIONS;


--22P-9346


--Write an SQL query to show records from one table that do not exist in another table.



SELECT * FROM employees e LEFT JOIN departments d ON e.department_id = d.department_id
WHERE d.MANAGER_ID IS NULL;
```

Script Output | Query Result

SQL | All Rows Fetched: 1 in 0.002 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID	DEPARTMENT_ID_1	DEPARTMENT_NAME	MAN
1	178	Kimberely	Grant	KGRANT	011.44.1644.429263	24-MAY-07	SA_REP	7000	0.15	149	(null)	(null)	(null)	

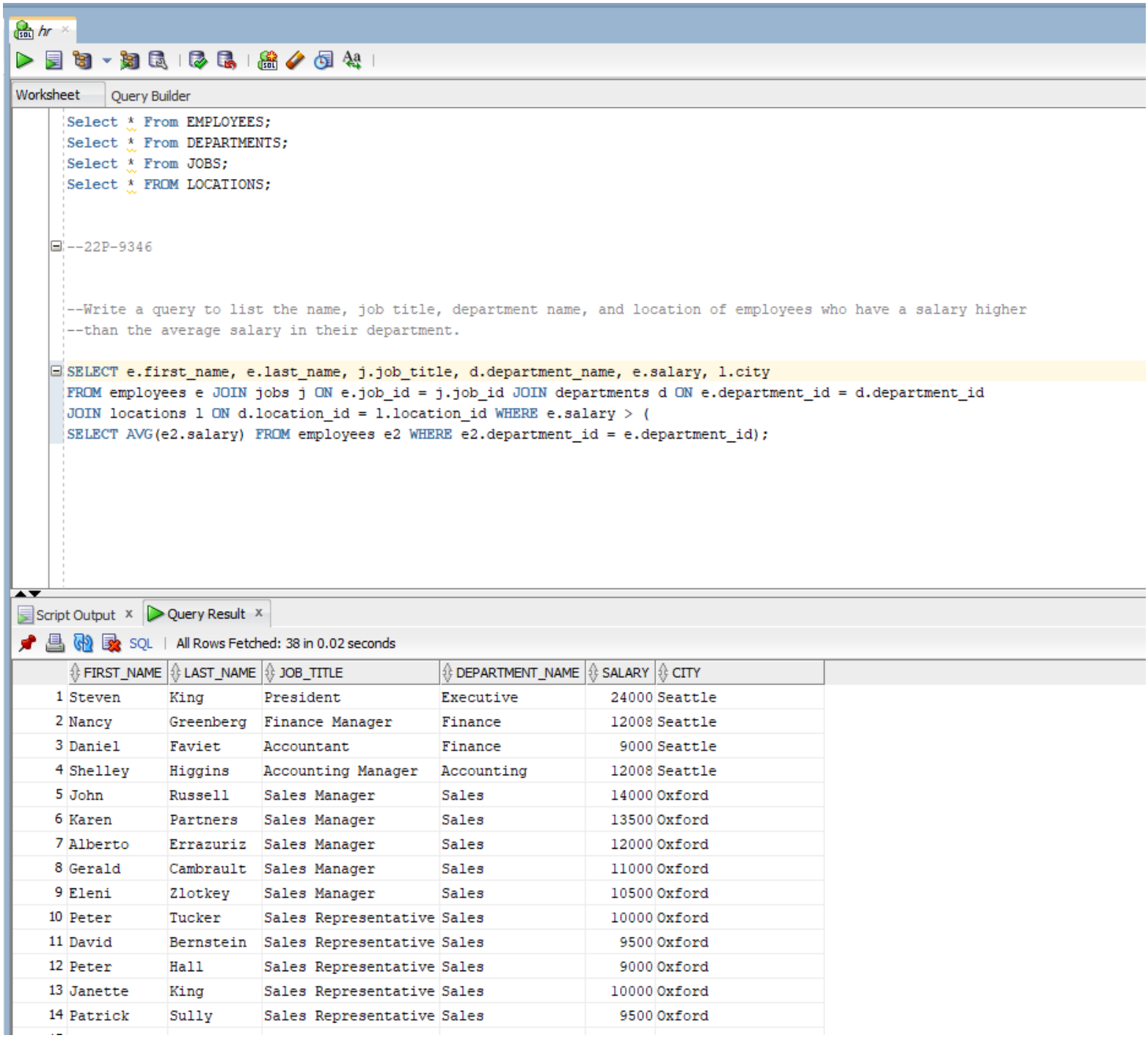


```
Select * From EMPLOYEES;
Select * From DEPARTMENTS;
Select * From JOBS;
Select * FROM LOCATIONS;


--22P-9346


--Display all employees who belong to the US but not to the state of Washington.


SELECT e.employee_id, e.first_name, e.last_name, d.department_name, l.city, l.state_province, l.country_id FROM employees e
JOIN departments d ON e.department_id = d.department_id JOIN locations l ON d.location_id = l.location_id
WHERE l.country_id = 'US' AND l.state_province <> 'Washington';
```

Script Output | Query Result

All Rows Fetched: 50 in 0.017 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_NAME	CITY	STATE_PROVINCE	COUNTRY_ID
1	103	Alexander	Hunold	IT	Southlake	Texas	US
2	104	Bruce	Ernst	IT	Southlake	Texas	US
3	105	David	Austin	IT	Southlake	Texas	US
4	106	Valli	Pataballa	IT	Southlake	Texas	US
5	107	Diana	Lorentz	IT	Southlake	Texas	US
6	120	Matthew	Weiss	Shipping	South San Francisco	California	US
7	121	Adam	Fripp	Shipping	South San Francisco	California	US
8	122	Payam	Kaufling	Shipping	South San Francisco	California	US
9	123	Shanta	Vollman	Shipping	South San Francisco	California	US
10	124	Kevin	Mourgos	Shipping	South San Francisco	California	US
11	125	Julia	Nayer	Shipping	South San Francisco	California	US
12	126	Irene	Mikkilineni	Shipping	South San Francisco	California	US
13	127	James	Landry	Shipping	South San Francisco	California	US
14	128	Steven	Markle	Shipping	South San Francisco	California	US
15	129	Laura	Bissot	Shipping	South San Francisco	California	US



hr

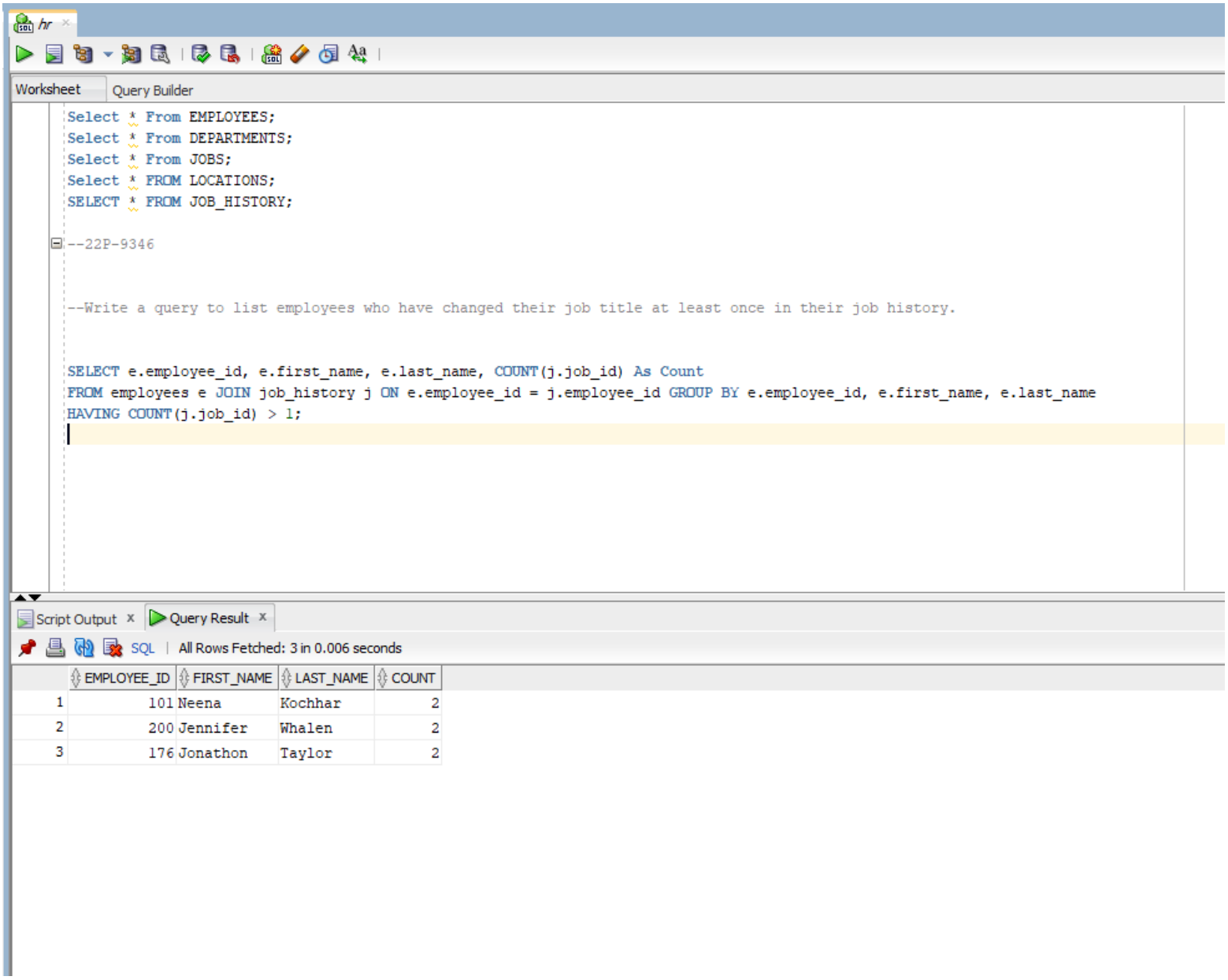
Worksheet | Query Builder

```
Select * From EMPLOYEES;
Select * From DEPARTMENTS;
Select * From JOBS;
Select * FROM LOCATIONS;


--22P-9346


--Write a query to list the name, job title, department name, and location of employees who have a salary higher
--than the average salary in their department.

SELECT e.first_name, e.last_name, j.job_title, d.department_name, e.salary, l.city
FROM employees e JOIN jobs j ON e.job_id = j.job_id JOIN departments d ON e.department_id = d.department_id
JOIN locations l ON d.location_id = l.location_id WHERE e.salary > (
SELECT AVG(e2.salary) FROM employees e2 WHERE e2.department_id = e.department_id);
```

Script Output | Query Result

SQL | All Rows Fetched: 38 in 0.02 seconds

	FIRST_NAME	LAST_NAME	JOB_TITLE	DEPARTMENT_NAME	SALARY	CITY
1	Steven	King	President	Executive	24000	Seattle
2	Nancy	Greenberg	Finance Manager	Finance	12008	Seattle
3	Daniel	Faviet	Accountant	Finance	9000	Seattle
4	Shelley	Higgins	Accounting Manager	Accounting	12008	Seattle
5	John	Russell	Sales Manager	Sales	14000	Oxford
6	Karen	Partners	Sales Manager	Sales	13500	Oxford
7	Alberto	Errazuriz	Sales Manager	Sales	12000	Oxford
8	Gerald	Cambrault	Sales Manager	Sales	11000	Oxford
9	Eleni	Zlotkey	Sales Manager	Sales	10500	Oxford
10	Peter	Tucker	Sales Representative	Sales	10000	Oxford
11	David	Bernstein	Sales Representative	Sales	9500	Oxford
12	Peter	Hall	Sales Representative	Sales	9000	Oxford
13	Janette	King	Sales Representative	Sales	10000	Oxford
14	Patrick	Sully	Sales Representative	Sales	9500	Oxford


hr
Worksheet
Query Builder
Select * From EMPLOYEES;
Select * From DEPARTMENTS;
Select * From JOBS;
Select * FROM LOCATIONS;
SELECT * FROM JOB_HISTORY;
--22P-9346
--Write a query to list employees who have changed their job title at least once in their job history.
SELECT e.employee_id, e.first_name, e.last_name, COUNT(j.job_id) As Count
FROM employees e JOIN job_history j ON e.employee_id = j.employee_id GROUP BY e.employee_id, e.first_name, e.last_name
HAVING COUNT(j.job_id) > 1;
Script Output
Query Result
SQL
All Rows Fetched: 3 in 0.006 seconds
EMPLOYEE_ID
FIRST_NAME
LAST_NAME
COUNT
1 101 Neena Kochhar 2
2 200 Jennifer Whalen 2
3 176 Jonathon Taylor 2

hr

Worksheet | Query Builder

```
Select * From EMPLOYEES;
Select * From DEPARTMENTS;
Select * From JOBS;
Select * FROM LOCATIONS;
SELECT * FROM JOB_HISTORY;

--22P-9346


--List employees who work in the same department as their managers.

SELECT e.employee_id, e.first_name || ' ' || e.last_name AS employee_name,
e.department_id, d.department_name, e.manager_id, m.first_name || ' ' || m.last_name AS manager_name
FROM employees e JOIN employees m ON e.manager_id = m.employee_id JOIN departments d ON e.department_id = d.department_id
WHERE e.department_id = m.department_id;
```

Script Output | Query Result

SQL | Fetched 50 rows in 0.016 seconds

	EMPLOYEE_ID	EMPLOYEE_NAME	DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	MANAGER_NAME
1	102	Lex De Haan	90	Executive	100	Steven King
2	101	Neena Kochhar	90	Executive	100	Steven King
3	107	Diana Lorentz	60	IT	103	Alexander Hunold
4	106	Valli Pataballa	60	IT	103	Alexander Hunold
5	105	David Austin	60	IT	103	Alexander Hunold
6	104	Bruce Ernst	60	IT	103	Alexander Hunold
7	113	Luis Popp	100	Finance	108	Nancy Greenberg
8	112	Jose Manuel Urman	100	Finance	108	Nancy Greenberg
9	111	Ismael Sciarra	100	Finance	108	Nancy Greenberg
10	110	John Chen	100	Finance	108	Nancy Greenberg
11	109	Daniel Faviet	100	Finance	108	Nancy Greenberg
12	119	Karen Colmenares	30	Purchasing	114	Den Raphaely
13	118	Guy Himuro	30	Purchasing	114	Den Raphaely
14	117	Sigal Tobias	30	Purchasing	114	Den Raphaely

Welcome Page | hr.sql

SQL Worksheet | History

Worksheet | Query Builder

```
Select * From EMPLOYEES;
Select * From DEPARTMENTS;
Select * From JOBS;
Select * FROM LOCATIONS;
SELECT * FROM JOB_HISTORY;

--22P-9346

--Write a query to list the name, department name, and location of employees who work in the same country as
--their department location.

SELECT e.first_name, e.last_name, d.department_name, l.city, l.country_id
FROM employees e JOIN departments d ON e.department_id = d.department_id
JOIN locations l ON d.location_id = l.location_id WHERE e.employee_id IN ( SELECT e2.employee_id FROM employees e2
JOIN departments d2 ON e2.department_id = d2.department_id JOIN locations l2 ON d2.location_id = l2.location_id
WHERE l2.country_id = l.country_id);
```

Query Result

SQL | Fetched 50 rows in 0.02 seconds

	FIRST_NAME	LAST_NAME	DEPARTMENT_NAME	CITY	COUNTRY_ID
1	Steven	King	Executive	Seattle	US
2	Neena	Kochhar	Executive	Seattle	US
3	Lex	De Haan	Executive	Seattle	US
4	Alexander	Hunold	IT	Southlake	US
5	Bruce	Ernst	IT	Southlake	US
6	David	Austin	IT	Southlake	US
7	Valli	Pataballa	IT	Southlake	US
8	Diana	Lorentz	IT	Southlake	US
9	Nancy	Greenberg	Finance	Seattle	US
10	Daniel	Faviet	Finance	Seattle	US
11	John	Chen	Finance	Seattle	US
12	Ismael	Sciarra	Finance	Seattle	US
13	Jose Manuel	Urman	Finance	Seattle	US
14	Luis	Popp	Finance	Seattle	US
15	Den	Raphaely	Purchasing	Seattle	US
16	Alexander	Khoo	Purchasing	Seattle	US
17	Shelli	Baida	Purchasing	Seattle	US

Welcome Page | hr.sql

SQL Worksheet | History

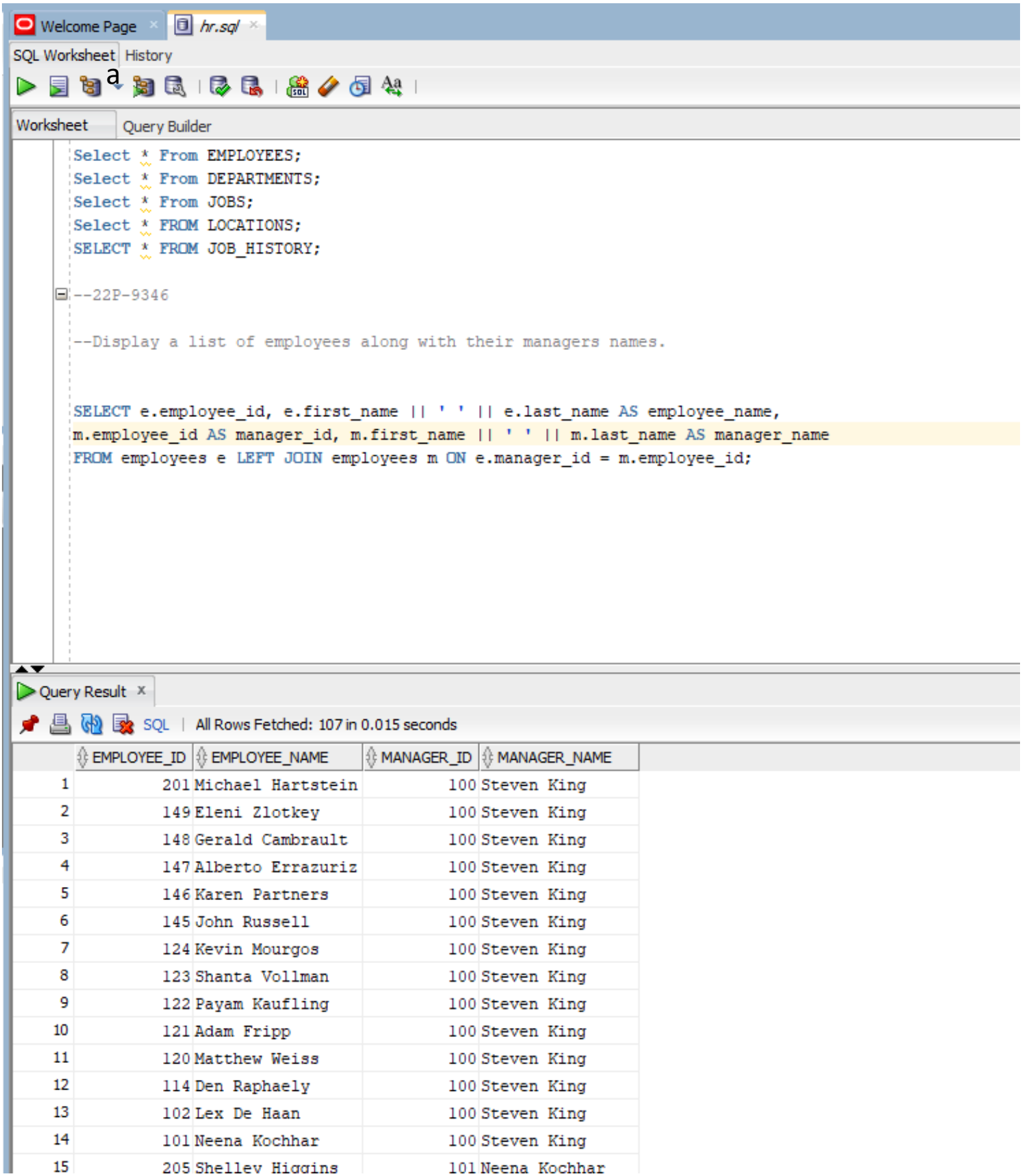
Worksheet | Query Builder

```
Select * From EMPLOYEES;
Select * From DEPARTMENTS;
Select * From JOBS;
Select * FROM LOCATIONS;
SELECT * FROM JOB_HISTORY;

--22P-9346

--Write a query to find employees who work in departments with more than 5 employees.


SELECT e.employee_id, e.first_name, e.last_name, e.department_id, d.department_name FROM employees e
JOIN departments d ON e.department_id = d.department_id WHERE e.department_id IN
( SELECT department_id FROM employees GROUP BY department_id HAVING COUNT(*) > 5);
```

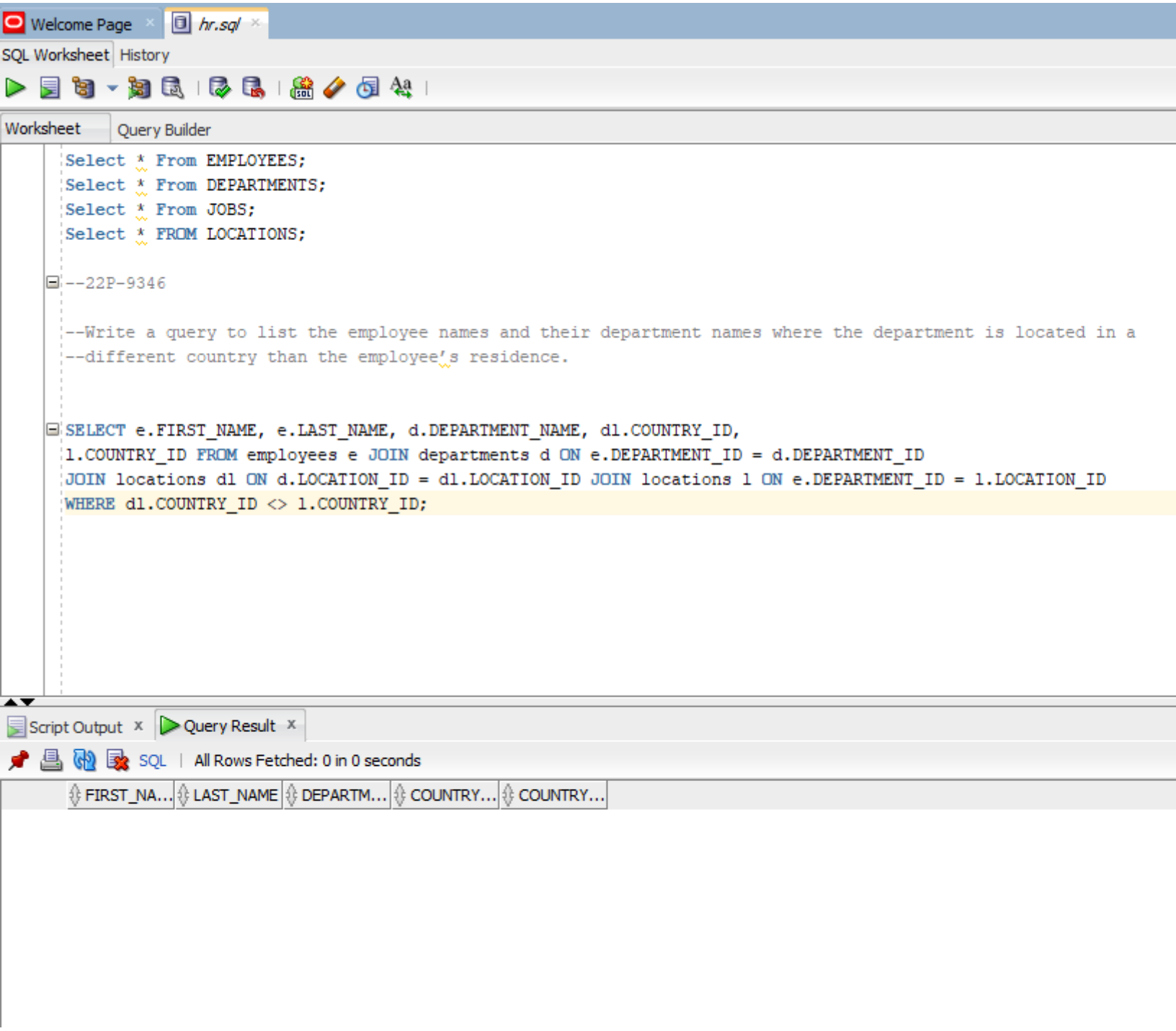
Query Result

SQL | All Rows Fetched: 91 in 0.013 seconds

	EMPLOYEE_ID	EMPLOYEE_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
74	144	Peter Vargas	50	Shipping
75	184	Nandita Sarchand	50	Shipping
76	188	Kelly Chung	50	Shipping
77	189	Jennifer Dilly	50	Shipping
78	199	Douglas Grant	50	Shipping
79	146	Karen Partners	80	Sales
80	152	Peter Hall	80	Sales
81	154	Nanette Cambrault	80	Sales
82	123	Shanta Vollman	50	Shipping
83	136	Hazel Philtanker	50	Shipping
84	138	Stephen Stiles	50	Shipping



	EMPLOYEE_ID	EMPLOYEE_NAME	MANAGER_ID	MANAGER_NAME
1	201	Michael Hartstein	100	Steven King
2	149	Eleni Zlotkey	100	Steven King
3	148	Gerald Cambrault	100	Steven King
4	147	Alberto Errazuriz	100	Steven King
5	146	Karen Partners	100	Steven King
6	145	John Russell	100	Steven King
7	124	Kevin Mourgos	100	Steven King
8	123	Shanta Vollman	100	Steven King
9	122	Payam Kaufling	100	Steven King
10	121	Adam Fripp	100	Steven King
11	120	Matthew Weiss	100	Steven King
12	114	Den Raphaely	100	Steven King
13	102	Lex De Haan	100	Steven King
14	101	Neena Kochhar	100	Steven King
15	205	Shelley Higgins	101	Neena Kochhar


Welcome Page
hr.sql
SQL Worksheet
History
Worksheet
Query Builder
Select * From EMPLOYEES;
Select * From DEPARTMENTS;
Select * From JOBS;
Select * FROM LOCATIONS;
--22P-9346
--Write a query to list the employee names and their department names where the department is located in a
--different country than the employee's residence.
SELECT e.FIRST_NAME, e.LAST_NAME, d.DEPARTMENT_NAME, dl.COUNTRY_ID,
l.COUNTRY_ID FROM employees e JOIN departments d ON e.DEPARTMENT_ID = d.DEPARTMENT_ID
JOIN locations dl ON d.LOCATION_ID = dl.LOCATION_ID JOIN locations l ON e.DEPARTMENT_ID = l.LOCATION_ID
WHERE dl.COUNTRY_ID <> l.COUNTRY_ID;
Script Output
Query Result
SQL
All Rows Fetched: 0 in 0 seconds
FIRST_NA...
LAST_NAME
DEPARTM...
COUNTRY...
COUNTRY...

Welcome Page | hr.sql

SQL Worksheet | History

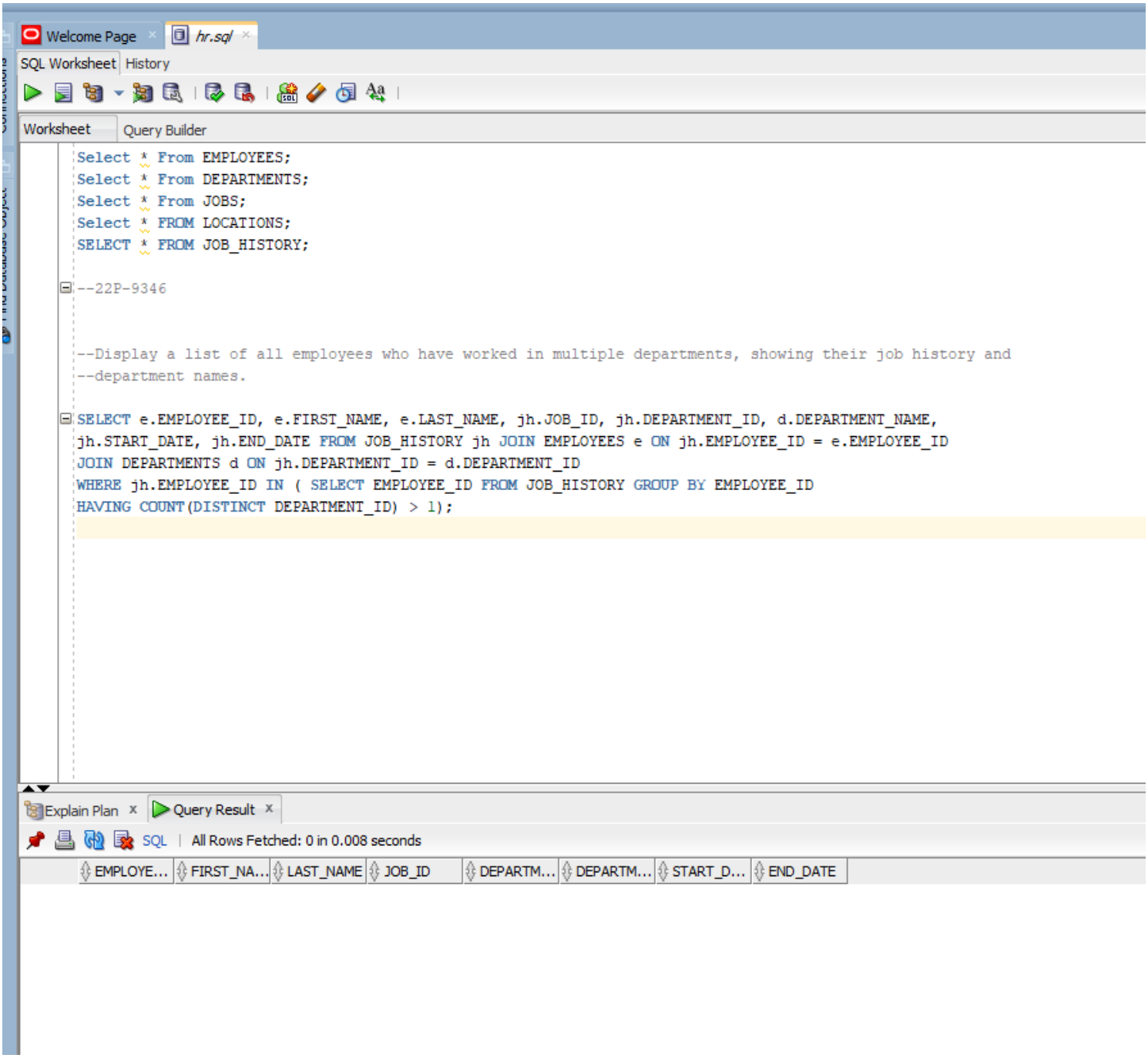
Worksheet | Query Builder

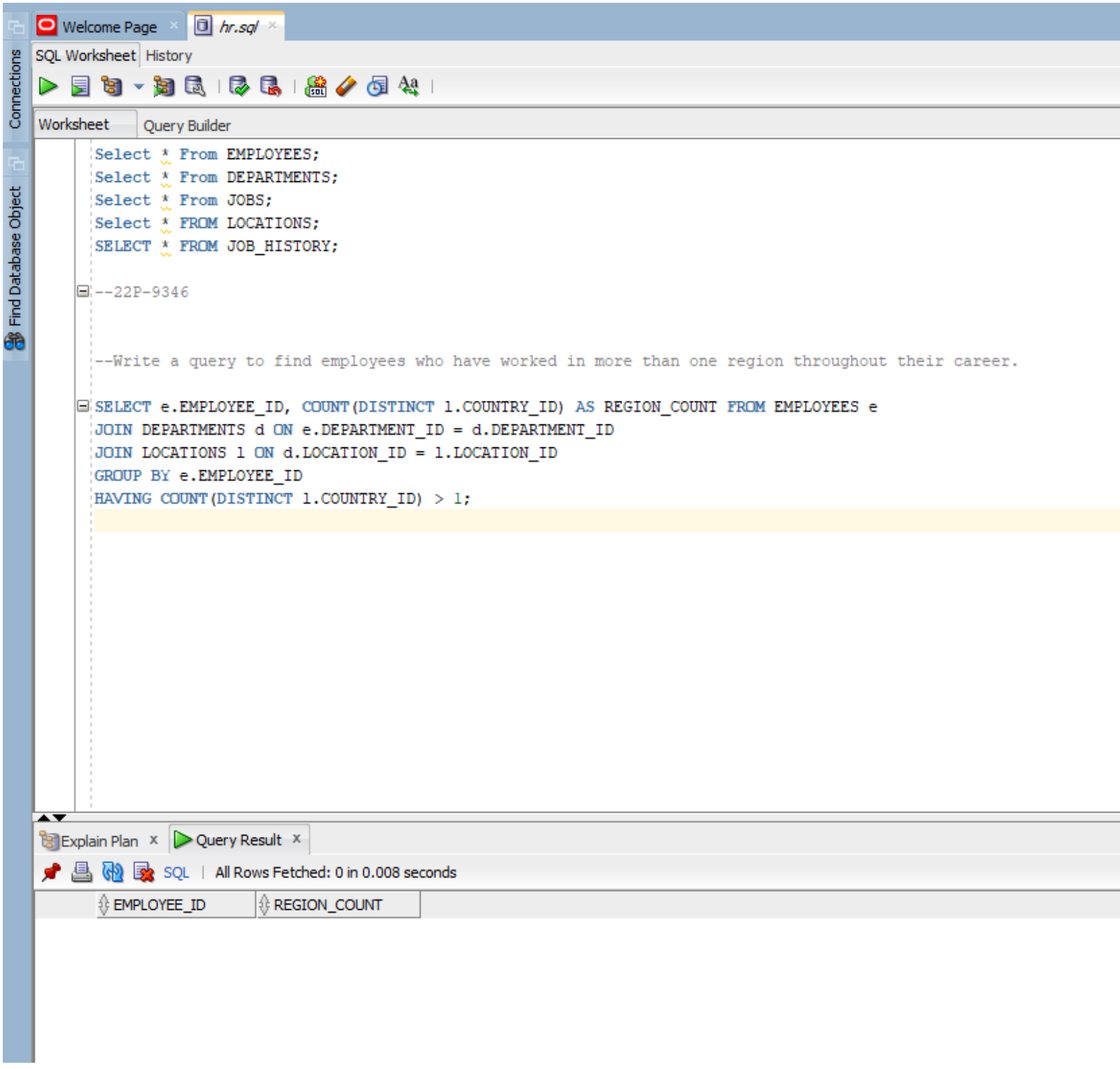
```
Select * From EMPLOYEES;
Select * From DEPARTMENTS;
Select * From JOBS;
Select * FROM LOCATIONS;
SELECT * FROM JOB_HISTORY;

--22P-9346

--Write a query to find employees who earn more than their department average salary but less than the
--highest salary in the company.


SELECT e.FIRST_NAME, e.LAST_NAME, e.SALARY, e.DEPARTMENT_ID, d.DEPARTMENT_NAME
FROM employees e JOIN departments d ON e.DEPARTMENT_ID = d.DEPARTMENT_ID
WHERE e.SALARY > (SELECT AVG(e2.SALARY) FROM employees e2  WHERE e2.DEPARTMENT_ID = e.DEPARTMENT_ID)
AND e.SALARY < (SELECT MAX(SALARY) FROM employees);
```

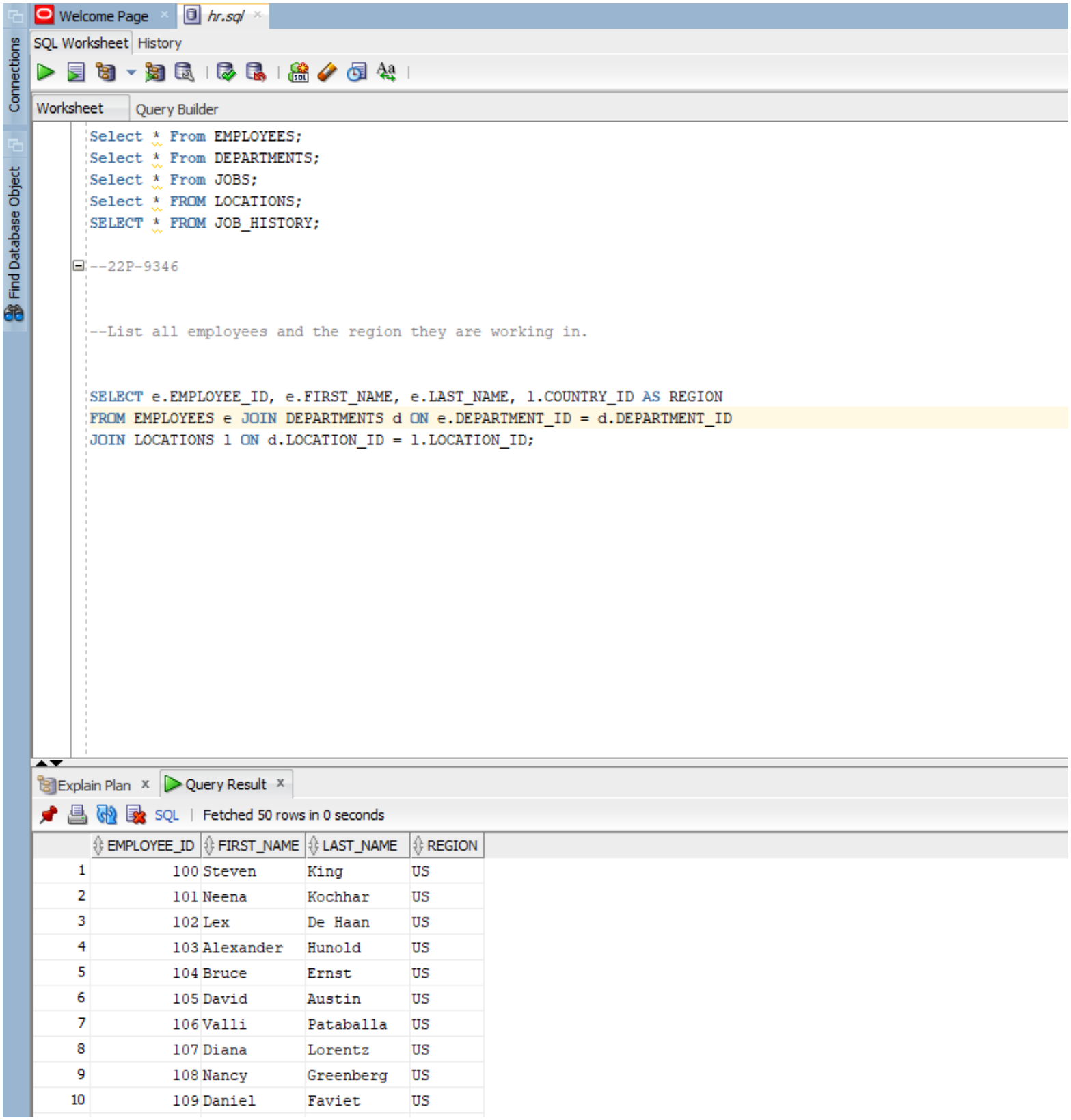
Script Output | Query Result

SQL | All Rows Fetched: 37 in 0.011 seconds

	FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_ID	DEPARTMENT_NAME
1	Daniel	Faviet	9000	100	Finance
2	Nancy	Greenberg	12008	100	Finance
3	Den	Raphaely	11000	30	Purchasing
4	Michael	Hartstein	13000	20	Marketing
5	Shelley	Higgins	12008	110	Accounting
6	Britney	Everett	3900	50	Shipping
7	Sarah	Bell	4000	50	Shipping
8	Jennifer	Dilly	3600	50	Shipping
9	Kelly	Chung	3800	50	Shipping
10	Alexis	Bull	4100	50	Shipping
11	Nandita	Sarchand	4200	50	Shipping
12	Trenna	Rajs	3500	50	Shipping
13	Renske	Ladwig	3600	50	Shipping
14	Kevin	Mourgos	5800	50	Shipping


Welcome Page
hr.sql
SQL Worksheet
History
Worksheet
Query Builder
Select * From EMPLOYEES;
Select * From DEPARTMENTS;
Select * From JOBS;
Select * FROM LOCATIONS;
SELECT * FROM JOB_HISTORY;
--22P-9346
--Display a list of all employees who have worked in multiple departments, showing their job history and
--department names.
SELECT e.EMPLOYEE_ID, e.FIRST_NAME, e.LAST_NAME, jh.JOB_ID, jh.DEPARTMENT_ID, d.DEPARTMENT_NAME,
jh.START_DATE, jh.END_DATE FROM JOB_HISTORY jh JOIN EMPLOYEES e ON jh.EMPLOYEE_ID = e.EMPLOYEE_ID
JOIN DEPARTMENTS d ON jh.DEPARTMENT_ID = d.DEPARTMENT_ID
WHERE jh.EMPLOYEE_ID IN (SELECT EMPLOYEE_ID FROM JOB_HISTORY GROUP BY EMPLOYEE_ID
HAVING COUNT(DISTINCT DEPARTMENT_ID) > 1);
Explain Plan
Query Result
SQL
All Rows Fetched: 0 in 0.008 seconds
EMPLOYE...
FIRST_NA...
LAST_NAME
JOB_ID
DEPARTM...
DEPARTM...
START_D...
END_DATE


Welcome Page
hr.sql
SQL Worksheet
History
Worksheet
Query Builder
Connections
Find Database Object
Select * From EMPLOYEES;
Select * From DEPARTMENTS;
Select * From JOBS;
Select * FROM LOCATIONS;
SELECT * FROM JOB_HISTORY;
--22P-9346
--Write a query to find employees who have worked in more than one region throughout their career.
SELECT e.EMPLOYEE_ID, COUNT(DISTINCT l.COUNTRY_ID) AS REGION_COUNT FROM EMPLOYEES e
JOIN DEPARTMENTS d ON e.DEPARTMENT_ID = d.DEPARTMENT_ID
JOIN LOCATIONS l ON d.LOCATION_ID = l.LOCATION_ID
GROUP BY e.EMPLOYEE_ID
HAVING COUNT(DISTINCT l.COUNTRY_ID) > 1;
Explain Plan
Query Result
SQL
All Rows Fetched: 0 in 0.008 seconds
EMPLOYEE_ID
REGION_COUNT


Welcome Page
hr.sql
SQL Worksheet
History
Worksheet
Query Builder
Connections
Find Database Object
Select * From EMPLOYEES;
Select * From DEPARTMENTS;
Select * From JOBS;
Select * FROM LOCATIONS;
SELECT * FROM JOB_HISTORY;
--22P-9346
--List all employees and the region they are working in.
SELECT e.EMPLOYEE_ID, e.FIRST_NAME, e.LAST_NAME, l.COUNTRY_ID AS REGION
FROM EMPLOYEES e JOIN DEPARTMENTS d ON e.DEPARTMENT_ID = d.DEPARTMENT_ID
JOIN LOCATIONS l ON d.LOCATION_ID = l.LOCATION_ID;
Explain Plan
Query Result
SQL
Fetched 50 rows in 0 seconds
EMPLOYEE_ID FIRST_NAME LAST_NAME REGION
1 100 Steven King US
2 101 Neena Kochhar US
3 102 Lex De Haan US
4 103 Alexander Hunold US
5 104 Bruce Ernst US
6 105 David Austin US
7 106 Valli Pataballa US
8 107 Diana Lorentz US
9 108 Nancy Greenberg US
10 109 Daniel Faviet US

Welcome Page | hr.sql

SQL Worksheet | History

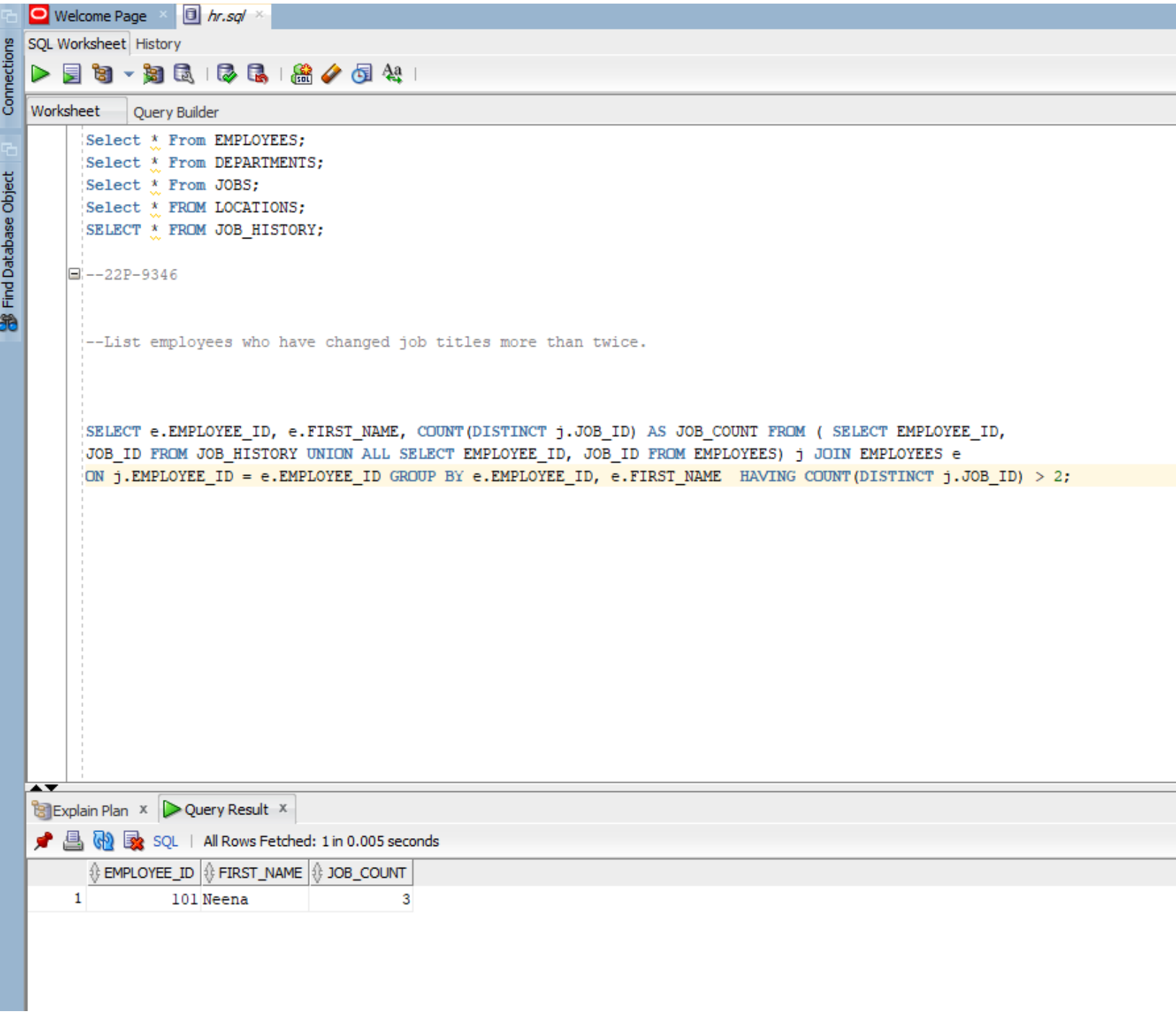
Worksheet | Query Builder

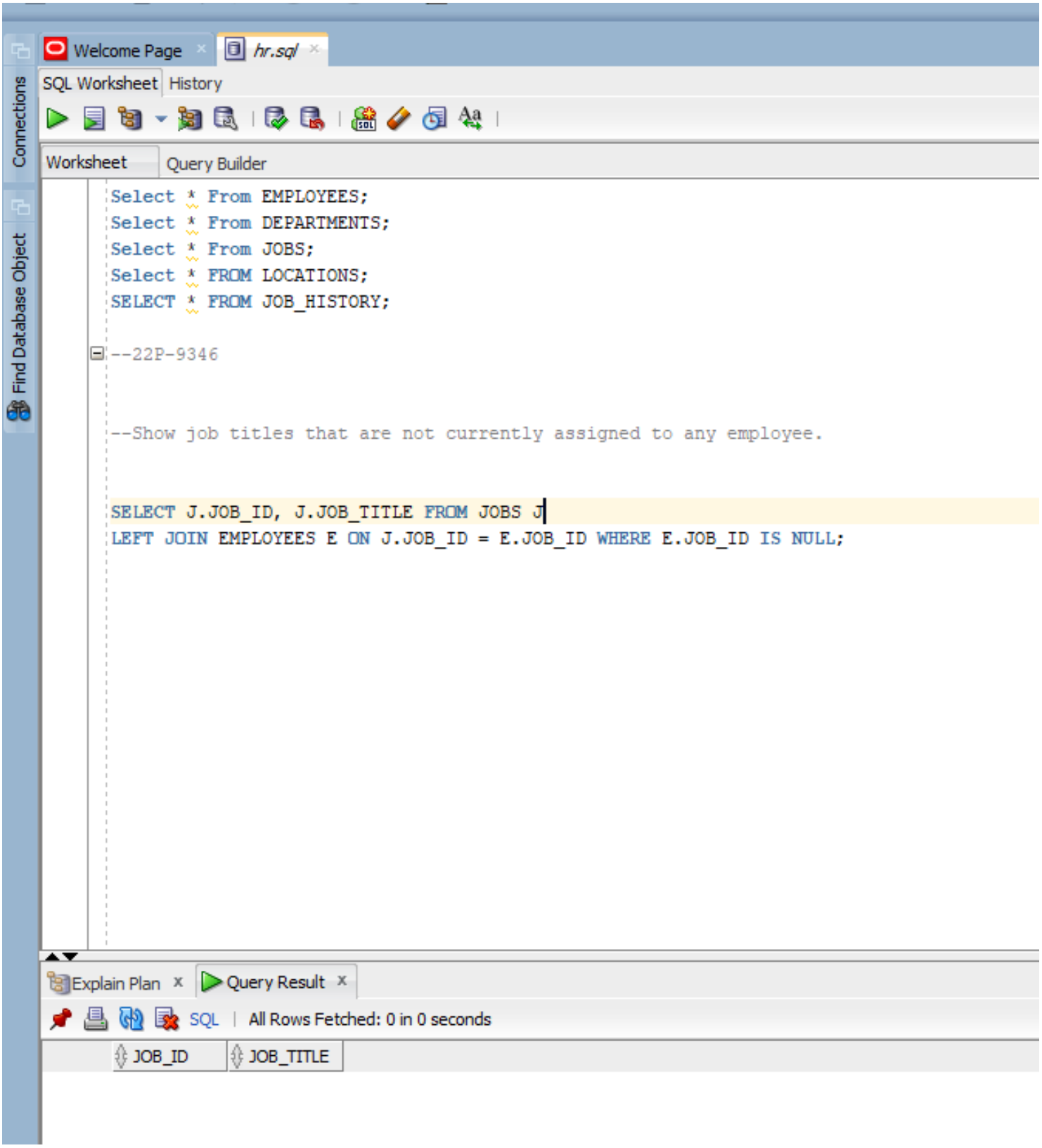
```
Select * From EMPLOYEES;
Select * From DEPARTMENTS;
Select * From JOBS;
Select * FROM LOCATIONS;
SELECT * FROM JOB_HISTORY;

--22P-9346


--Find employees who have the same last name but work in different departments.


SELECT e1.EMPLOYEE_ID, e1.FIRST_NAME, e1.LAST_NAME, e1.DEPARTMENT_ID, e2.EMPLOYEE_ID AS OTHER_EMPLOYEE_ID,
e2.FIRST_NAME AS OTHER_FIRST_NAME, e1.LAST_NAME AS OTHER_LAST_NAME, e2.DEPARTMENT_ID AS OTHER_DEPARTMENT_ID
FROM EMPLOYEES e1 JOIN EMPLOYEES e2 ON e1.LAST_NAME = e2.LAST_NAME AND e1.DEPARTMENT_ID <> e2.DEPARTMENT_ID;
```

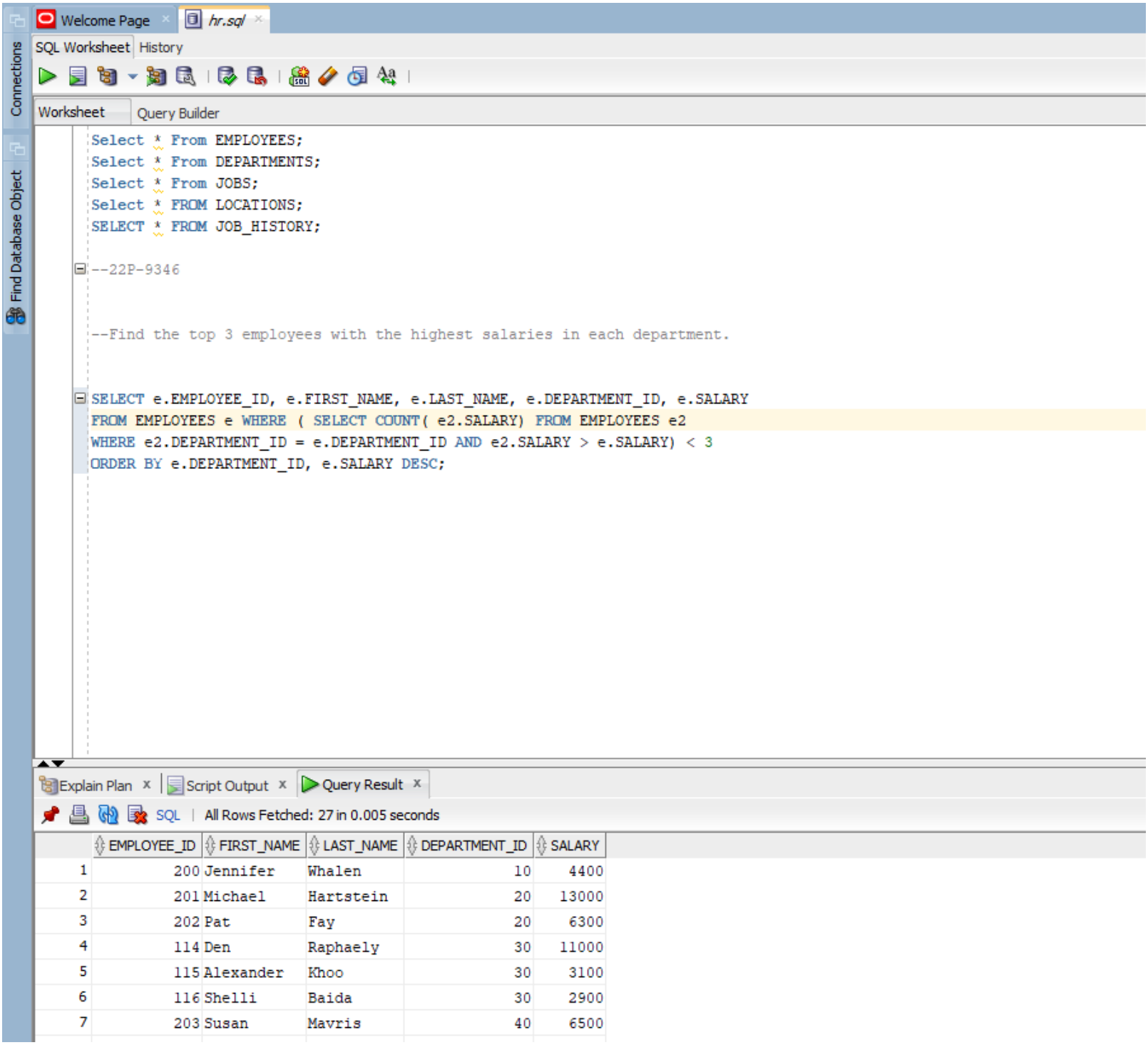
Explain Plan | Query Result

SQL | All Rows Fetched: 4 in 0 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID	OTHER_EMPLOYEE_ID	OTHER_FIRST_NAME	OTHER_LAST_NAME	OTHER_DEPARTMENT_ID
1	156	Janette	King	80	100	Steven	King	90
2	100	Steven	King	90	156	Janette	King	80
3	180	Winston	Taylor	50	176	Jonathon	Taylor	80
4	176	Jonathon	Taylor	80	180	Winston	Taylor	50


Welcome Page
hr.sql
SQL Worksheet
History
Worksheet
Query Builder
Connections
Find Database Object
Select * From EMPLOYEES;
Select * From DEPARTMENTS;
Select * From JOBS;
Select * FROM LOCATIONS;
SELECT * FROM JOB_HISTORY;
--22P-9346
--List employees who have changed job titles more than twice.
SELECT e.EMPLOYEE_ID, e.FIRST_NAME, COUNT(DISTINCT j.JOB_ID) AS JOB_COUNT FROM (SELECT EMPLOYEE_ID,
JOB_ID FROM JOB_HISTORY UNION ALL SELECT EMPLOYEE_ID, JOB_ID FROM EMPLOYEES) j JOIN EMPLOYEES e
ON j.EMPLOYEE_ID = e.EMPLOYEE_ID GROUP BY e.EMPLOYEE_ID, e.FIRST_NAME HAVING COUNT(DISTINCT j.JOB_ID) > 2;
Explain Plan
Query Result
SQL
All Rows Fetched: 1 in 0.005 seconds
EMPLOYEE_ID
FIRST_NAME
JOB_COUNT
1
101
Neena
3


Welcome Page
hr.sql
SQL Worksheet
History
Worksheet
Query Builder
Connections
Find Database Object
Select * From EMPLOYEES;
Select * From DEPARTMENTS;
Select * From JOBS;
Select * FROM LOCATIONS;
SELECT * FROM JOB_HISTORY;
--22P-9346
--Show job titles that are not currently assigned to any employee.
SELECT J.JOB_ID, J.JOB_TITLE FROM JOBS J
LEFT JOIN EMPLOYEES E ON J.JOB_ID = E.JOB_ID WHERE E.JOB_ID IS NULL;
Explain Plan
Query Result
SQL
All Rows Fetched: 0 in 0 seconds
JOB_ID
JOB_TITLE


Welcome Page
hr.sql
SQL Worksheet
History
Worksheet
Query Builder
Connections
Find Database Object
Select * From EMPLOYEES;
Select * From DEPARTMENTS;
Select * From JOBS;
Select * FROM LOCATIONS;
SELECT * FROM JOB_HISTORY;
--22P-9346
--Find the top 3 employees with the highest salaries in each department.
SELECT e.EMPLOYEE_ID, e.FIRST_NAME, e.LAST_NAME, e.DEPARTMENT_ID, e.SALARY
FROM EMPLOYEES e WHERE (SELECT COUNT(e2.SALARY) FROM EMPLOYEES e2
WHERE e2.DEPARTMENT_ID = e.DEPARTMENT_ID AND e2.SALARY > e.SALARY) < 3
ORDER BY e.DEPARTMENT_ID, e.SALARY DESC;
Explain Plan
Script Output
Query Result
SQL
All Rows Fetched: 27 in 0.005 seconds
EMPLOYEE_ID FIRST_NAME LAST_NAME DEPARTMENT_ID SALARY
1 200 Jennifer Whalen 10 4400
2 201 Michael Hartstein 20 13000
3 202 Pat Fay 20 6300
4 114 Den Raphaely 30 11000
5 115 Alexander Khoo 30 3100
6 116 Shelli Baida 30 2900
7 203 Susan Mavris 40 6500