Name: Danny Chen
Assignment:
Date: 10/14/24

6.1
Vocab

Words	Definitions
CROSS JOIN	Returns the Cartesian product from two tables.
NATURAL JOIN	Joins two tables based on the same column name.

1. Create a cross-join that displays the last name and department name from the employees and departments tables.

ANS:
SELECT last_name, department_name
FROM employees CROSS JOIN departments;

LAST_NAME	DEPARTMENT_NAME
Abel	Administration
Almeida Castro	Administration
Alves Rocha	Administration
Barbosa Souza	Administration
Bell	Administration
Davies	Administration
De Haan	Administration
Duric	Administration

2. Create a query that uses a natural join to join the departments table and the locations table. Display the department id, department name, location id, and city.

ANS:
SELECT *
FROM departments NATURAL JOIN locations;

LOCATION_ID	DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	STREET_ADDRESS	POSTAL_CODE	CITY
1400	60	IT	103	2014 Jabberwocky Rd	26192	Southlake
1500	50	Shipping	124	2011 Interiors Blvd	99236	South San Francisco
1700	10	Administration	200	2004 Charade Rd	98199	Seattle
1700	190	Contracting	-	2004 Charade Rd	98199	Seattle
1700	90	Executive	100	2004 Charade Rd	98199	Seattle
1700	110	Accounting	205	2004 Charade Rd	98199	Seattle
1800	20	Marketing	201	460 Bloor St. W.	ON M5S 1X8	Toronto
2100	85	Sales - Americas	149	Av. Rio Branco	20090-003	Rio de Janeiro
2500	80	Sales - Europe	149	Magdalen Centre, The Oxford Science Park	OX9 9ZB	Oxford

3. Create a query that uses a natural join to join the departments table and the locations table. Restrict the output to only department IDs of 20 and 50. Display the department id, department name, location id, and city.

ANS:

```
SELECT department_id, department_name, location_id, city
FROM departments NATURAL JOIN locations
WHERE department_id IN (20, 50);
```

DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID	CITY
20	Marketing	1800	Toronto
50	Shipping	1500	South San Francisco

6.2

Vocab

Words	Definitions
ON clause	Allows a natural join based on an arbitrary condition or two columns with different names.
USING clause	Performs an equijoin based on one specified column name

1. Join the Oracle database locations and departments table using the location_id column. Limit the results to location 1400 only

ANS:

```
SELECT *
FROM departments JOIN locations USING (location_id)
WHERE location_id = 1400;
```

LOCATION_ID	DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	STREET_ADDRESS	POSTAL_CODE	CITY	STATE_PROVINCE	COUNTRY_ID
1400	60	IT	103	2014 Jabberwocky Rd	26192	Southlake	Texas	1

1 rows returned in 0.00 seconds Download

2. Join DJs on Demand d_play_list_items, d_track_listings, and d_cds tables with the JOIN USING syntax. Include the song ID, CD number, title, and comments in the output.

ANS:

```
SELECT song_id, cd_number, title, comments
FROM d_play_list_items JOIN d_track_listings USING (song_id)
   JOIN d_cds USING (cd_number);
```

SONG_ID	CD_NUMBER	TITLE	COMMENTS
45	92	Back to the Shire	Play late
46	93	Songs from My Childhood	-
47	91	Party Music for All Occasions	Play early
48	95	Here Comes the Bride	Play after cake cutting
49	91	Party Music for All Occasions	Play first
47	91	Party Music for All Occasions	Play for the father

3. Display the city, department name, location ID, and department ID for departments 10, 20, and 30 for the city of Seattle.

ANS:

```
SELECT city, department_name, location_id, department_id
FROM departments JOIN locations USING (location_id)
WHERE department_id IN (10, 20, 30) AND city = 'Seattle';
```

CITY	DEPARTMENT_NAME	LOCATION_ID	DEPARTMENT_ID
Seattle	Administration	1700	10

4. Display country name, region ID, and region name for Americas.

ANS:

```
SELECT country_name, region_id, region_name
FROM countries JOIN regions USING (region_id)
WHERE region_name LIKE '%America';
```

COUNTRY_NAME	REGION_ID	REGION_NAME
South Georgia and the South Sandwich Islands	5	South America
Falkland Islands	5	South America
Co-operative Republic of Guyana	5	South America
Republic of Peru	5	South America
Republic of Paraguay	5	South America
Republic of Ecuador	5	South America
Oriental Republic of Uruguay	5	South America
Bolivarian Republic of Venezuela	5	South America
Republic of Chile	5	South America
Republic of Colombia	5	South America
Republic of Suriname	5	South America
Department of Guiana	5	South America
Argentine Republic	5	South America
Republic of Bolivia	5	South America
Federative Republic of Brazil	5	South America
Belize	13	Central America
Republic of Guatemala	13	Central America
United Mexican States	13	Central America
Republic of Nicaragua	13	Central America
Republic of El Salvador	13	Central America
Republic of Panama	13	Central America
Republic of Honduras	13	Central America
Republic of Costa Rica	13	Central America
United States of America	21	North America
Territorial Collectivity of Saint Pierre and Miquelon	21	North America
Canada	21	North America
Greenland	21	North America

5. Write a statement joining the employees and jobs tables. Display the first and last names, hire date, job id, job title, and maximum salary. Limit the query to those employees who are in jobs that can earn more than $12,000.

ANS:

```
SELECT first_name, last_name, hire_date, job_id, job_title, max_salary
FROM employees JOIN jobs USING (job_id)
WHERE min_salary > 12000;
```

FIRST_NAME	LAST_NAME	HIRE_DATE	JOB_ID	JOB_TITLE	MAX_SALARY
Steven	King	17-Jun-2002	AD_PRES	President	40000
Neena	Kochhar	21-Sep-2004	AD_VP	Administration Vice President	30000
Lex	De Haan	13-Jan-2008	AD_VP	Administration Vice President	30000

6. Display job title, employee first name, last name, and email for all employees who are stock clerks.

ANS:

```
SELECT job_title, first_name, last_name, email
FROM employees JOIN jobs USING (job_id)
WHERE LOWER(job_title) LIKE '%stock clerk%';
```

JOB_TITLE	FIRST_NAME	LAST_NAME	EMAIL
Senior Stock Clerk	George	Bell	GBELL
Stock Clerk	Trenna	Rajs	TRAJS
Stock Clerk	Curtis	Davies	CDAVIES
Stock Clerk	Tiffany	Heiden	THEIDEN
Stock Clerk	Peter	Vargas	PVARGAS
Stock Clerk	Randall	Matos	RMATOS

The following questions use the JOIN…ON syntax:

7. Write a statement that displays the employee ID, first name, last name, manager ID, manager first name, and manager last name for every employee in the employees table. Hint: this is a self-join.

ANS:

```
SELECT
   e.employee_id,
   e.first_name "Employee First Name",
   e.last_name "Employee Last Name",
   e.manager_id "Manager ID" ,
   m.first_name "Manager First Name",
   m.last_name "Manager Last Name"
FROM employees e LEFT JOIN employees m
   ON (e.manager_id = m.employee_id);
```

EMPLOYEE_ID	Employee First Name	Employee Last Name	Manager ID	Manager First Name	Manager Last Name
101	Neena	Kochhar	100	Steven	King
102	Lex	De Haan	100	Steven	King
149	Eleni	Zlotkey	100	Steven	King
124	Kevin	Mourgos	100	Steven	King
201	Michael	Hartstein	100	Steven	King
200	Jennifer	Whalen	101	Neena	Kochhar
205	Shelley	Higgins	101	Neena	Kochhar
225	Katia	Hernandez	101	Neena	Kochhar
226	Guido	Ricci	101	Neena	Kochhar

215	Donna	Steiner	201	Michael	Hartstein
217	Lisa	TAYLOR	201	Michael	Hartstein
219	Michael	Stocks	201	Michael	Hartstein
228	Nabil	Safwah	201	Michael	Hartstein
235	Alice	Newton	201	Michael	Hartstein
206	William	Gietz	205	Shelley	Higgins
224	Matthias	Reinhard	205	Shelley	Higgins
231	Jelena	Duric	205	Shelley	Higgins
232	Jennifer	Loermans	205	Shelley	Higgins
100	Steven	King	-	-	-

8. Use JOIN ON syntax to query and display the location ID, city, and department name for all Canadian locations.

ANS:

```
SELECT l.location_id, city, department_name
FROM locations l JOIN departments d
   ON (l.location_id = d.location_id)
WHERE country_id = 2;
```

OR

```
SELECT l.location_id, l.city, d.department_name
FROM locations l JOIN departments d
   ON (l.location_id = d.location_id)
WHERE l.country_id = 2;
```

9. Query and display manager ID, department ID, department name, first name, and last name for all employees in departments 80, 90, 110, and 190.

ANS:

```
SELECT e.manager_id, d.department_id, d.department_name, e.first_name, e.last_name
FROM employees e JOIN departments d
   ON (e.department_id = d.department_id)
WHERE e.department_id IN (80, 90, 110, 190);
```

MANAGER_ID	DEPARTMENT_ID	DEPARTMENT_NAME	FIRST_NAME	LAST_NAME
100	80	Sales - Europe	Eleni	Zlotkey
149	80	Sales - Europe	Ellen	Abel
149	80	Sales - Europe	Jonathon	Taylor
149	80	Sales - Europe	Nick	Hooper
-	90	Executive	Steven	King
100	90	Executive	Neena	Kochhar
100	90	Executive	Lex	De Haan
101	110	Accounting	Shelley	Higgins
205	110	Accounting	William	Gietz
205	110	Accounting	Matthias	Reinhard
205	110	Accounting	Jelena	Duric
205	110	Accounting	Jennifer	Loermans

10. Display employee ID, last name, department ID, department name, and hire date for those employees whose hire date was June 7, 1994.

ANS:

```
SELECT e.employee_id, e.last_name, d.department_id, d.department_name, e.hire_date
FROM employees e JOIN departments d
   ON (e.department_id = d.department_id)
WHERE e.hire_date = '07-Jun-1994';
```

NOTE: There is actually no one whose hire date was 07-Jun-1994 when I check the table data in the Oracle APEX's Object Browser, so no data was returned

no data found

6.3

Vocab

Words	Definitions
FULL OUTER JOIN	Performs a join on two tables, retrieves all the rows in the Left table, even if there is no match in the Right table. It also retrieves all the rows in the Right table, even if there is no match in the Left table.
OUTER JOIN	A join that returns the unmatched rows as well as matched rows

LEFT OUTER JOIN	Performs a join on two tables, retrieves all the rows in the Left table even if there is no match in the Right table
RIGHT OUTER JOIN	Performs a join on two tables, retrieves all the rows in the Right table even if there is no match in the Left table.
INNER JOIN	A join of two or more tables that returns only matched rows

1. Return the first name, last name, and department name for all employees including those employees not assigned to a department.

ANS:

```
SELECT e.first_name, e.last_name, d.department_name
FROM employees e LEFT OUTER JOIN departments d
ON (e.department_id = d.department_id);
```

FIRST_NAME	LAST_NAME	DEPARTMENT_NAME
Jennifer	Whalen	Administration
Katia	Hernandez	Administration
Guido	Ricci	Administration
Mizuto	Saikawa	Administration
Michael	Hartstein	Marketing
Pat	Fay	Marketing
William	Gietz	Accounting
Matthias	Reinhard	Accounting
Jelena	Duric	Accounting
Jennifer	Loermans	Accounting
Kimberely	Grant	-

2. Return the first name, last name, and department name for all employees including those departments that do not have an employee assigned to them.

ANS:

```
SELECT e.first_name, e.last_name, d.department_name
FROM employees e RIGHT OUTER JOIN departments d
ON (e.department_id = d.department_id);
```

FIRST_NAME	LAST_NAME	DEPARTMENT_NAME
Guido	Ricci	Administration
Mizuto	Saikawa	Administration
Katia	Hernandez	Administration
Jennifer	Whalen	Administration
Nabil	Safwah	Marketing
Shelley	Higgins	Accounting
William	Gietz	Accounting
Jelena	Duric	Accounting
Jennifer	Loermans	Accounting
Matthias	Reinhard	Accounting
-	-	Contracting

3. Return the first name, last name, and department name for all employees including those departments that do not have an employee assigned to them and those employees not assigned to a department.

ANS:

```
SELECT e.first_name, e.last_name, d.department_name
FROM employees e FULL OUTER JOIN departments d
ON (e.department_id = d.department_id);
```

FIRST_NAME	LAST_NAME	DEPARTMENT_NAME
Steven	King	Executive
Neena	Kochhar	Executive
Lex	De Haan	Executive
Jennifer	Whalen	Administration
Shelley	Higgins	Accounting
William	Gietz	Accounting
Eleni	Zlotkey	Sales - Europe
Kimberely	Grant	-
Kevin	Mourgos	Shipping
Trenna	Rajs	Shipping

Jelena	Duric	Accounting
Jennifer	Loermans	Accounting
Alice	Newton	Marketing
-	-	Contracting

4. Create a query of the DJs on Demand database to return the first name, last name, event date, and description of the event the client held. Include all the clients even if they have not had an event scheduled.

ANS:

```
SELECT c.first_name, c.last_name, e.event_date, e.description
FROM d_clients c LEFT OUTER JOIN d_events e
ON (c.client_number = e.client_number);
```

FIRST_NAME	LAST_NAME	EVENT_DATE	DESCRIPTION
Serena	Jones	-	-
Hiram	Peters	14-May-2004	Party for 200, red, white, blue motif
Lauren	Vigil	28-Apr-2004	Black tie at Four Season hotel

5. Using the Global Fast Foods database, show the shift description and shift assignment date even if there is no date assigned for each shift description.

ANS:

```
SELECT sh.description, sh_asg.shift_assign_date
FROM f_shifts sh LEFT OUTER JOIN f_shift_assignments sh_asg
ON (sh.code = sh_asg.code);
```

DESCRIPTION	SHIFT_ASSIGN_DATE
8am to 12pm	06-May-2004
6pm to 10pm	-

6.4
Vocab

Words	Definitions

Self join	Joins a table to itself
Hierarchical queries	Retrieves data based on a natural hierarchical relationship between rows in a table
LEVEL	Determines the number of steps down from the beginning row that should be returned by a hierarchical query
START WITH	Identifies the beginning row for a hierarchical query
CONNECT BY	Specifies the relationship between parent rows and child rows of a hierarchical query

1. Display the employee's last name and employee number along with the manager's last name and manager number. Label the columns: Employee, Emp#, Manager, and Mgr#, respectively.

ANS:

```
SELECT
   e.last_name "Employee",
   e.employee_id "Emp#",
   m.last_name "Manager",
   m.manager_id "Mgr#"
FROM employees e JOIN employees m
ON (e.manager_id = m.employee_id);
```

Employee	Emp#	Manager	Mgr#
Zlotkey	149	King	-
Mourgos	124	King	-
Kochhar	101	King	-
Hartstein	201	King	-
De Haan	102	King	-
Whalen	200	Kochhar	100
Saikawa	227	Kochhar	100
Ricci	226	Kochhar	100

Reinhard	224	Higgins	101
Loermans	232	Higgins	101
Gietz	206	Higgins	101
Duric	231	Higgins	101

39 rows returned in 0.01 seconds Download

2. Modify question 1 to display all employees and their managers, even if the employee does not have a manager. Order the list alphabetically by the last name of the employee.

ANS:

```
SELECT
   e.last_name "Employee",
   e.employee_id "Emp#",
   m.last_name "Manager",
   m.manager_id "Mgr#"
FROM employees e LEFT OUTER JOIN employees m
ON (e.manager_id = m.employee_id);
```

Employee	Emp#	Manager	Mgr#
Kochhar	101	King	-
De Haan	102	King	-
Zlotkey	149	King	-
Mourgos	124	King	-
Hartstein	201	King	-
Whalen	200	Kochhar	100
Higgins	205	Kochhar	100
Hernandez	225	Kochhar	100

Steiner	215	Hartstein	100
TAYLOR	217	Hartstein	100
Stocks	219	Hartstein	100
Safwah	228	Hartstein	100
Newton	235	Hartstein	100
King	100	-	-

40 rows returned in 0.01 seconds Download

3. Display the names and hire dates for all employees who were hired before their managers, along with their managers' names and hire dates. Label the columns Employee, Emp Hired, Manager and Mgr Hired, respectively.

ANS:

```
SELECT
   CONCAT(e.first_name || ' ', e.last_name) "Employee",
   e.hire_date "Emp Hired",
   CONCAT(m.first_name || ' ', m.last_name) "Manager",
   m.hire_date "Mgr Hired"
FROM employees e JOIN employees m
ON (e.manager_id = m.employee_id)
AND e.hire_date < m.hire_date;
```

Employee	Emp Hired	Manager	Mgr Hired
Jennifer Whalen	17-Sep-2002	Neena Kochhar	21-Sep-2004
Alexander Hunold	03-Jan-2005	Lex De Haan	13-Jan-2008
Trenna Rajs	17-Oct-2010	Kevin Mourgos	16-Nov-2014
Curtis Davies	29-Jan-2012	Kevin Mourgos	16-Nov-2014
Randall Matos	15-Mar-2013	Kevin Mourgos	16-Nov-2014
Peter Vargas	09-Jul-2013	Kevin Mourgos	16-Nov-2014

Nick Hooper	01-Sep-2012	Eleni Zlotkey	29-Jan-2015
Donna Steiner	02-Nov-2004	Michael Hartstein	17-Feb-2011
Nabil Safwah	06-Jan-1997	Michael Hartstein	17-Feb-2011
Matthias Reinhard	25-Jul-2007	Shelley Higgins	07-Jun-2009
Jelena Duric	11-May-2009	Shelley Higgins	07-Jun-2009

19 rows returned in 0.00 seconds Download

4. Write a report that shows the hierarchy for Lex De Haans department. Include last name, salary, and department id in the report

ANS:

```
SELECT last_name, salary, department_id
FROM employees
START WITH first_name = 'Lex' AND last_name = 'De Haan'
CONNECT BY PRIOR employee_id = manager_id;
```

LAST_NAME	SALARY	DEPARTMENT_ID
De Haan	17000	90
Hunold	9000	60
Ernst	6000	60
Lorentz	4200	60
Li	8000	60
Fontaine	7800	60

6 rows returned in 0.01 seconds Download

5. What is wrong in the following statement?

```
SELECT last_name, department_id, salary
FROM employees
START WITH last_name = 'King'
CONNECT BY PRIOR manager_id = employee_id;
```

ANS:

The last statement is incorrect as the PRIOR keyword should be put before the employee_id rather than manager_id. The corrected statement can be either CONNECT BY PRIOR employee_id = manager_id or CONNECT BY manager_id = PRIOR employee_id.

Incorrect statement in the question leads to this output:

LAST_NAME	DEPARTMENT_ID	SALARY
King	90	24000

Corrected statement and output:

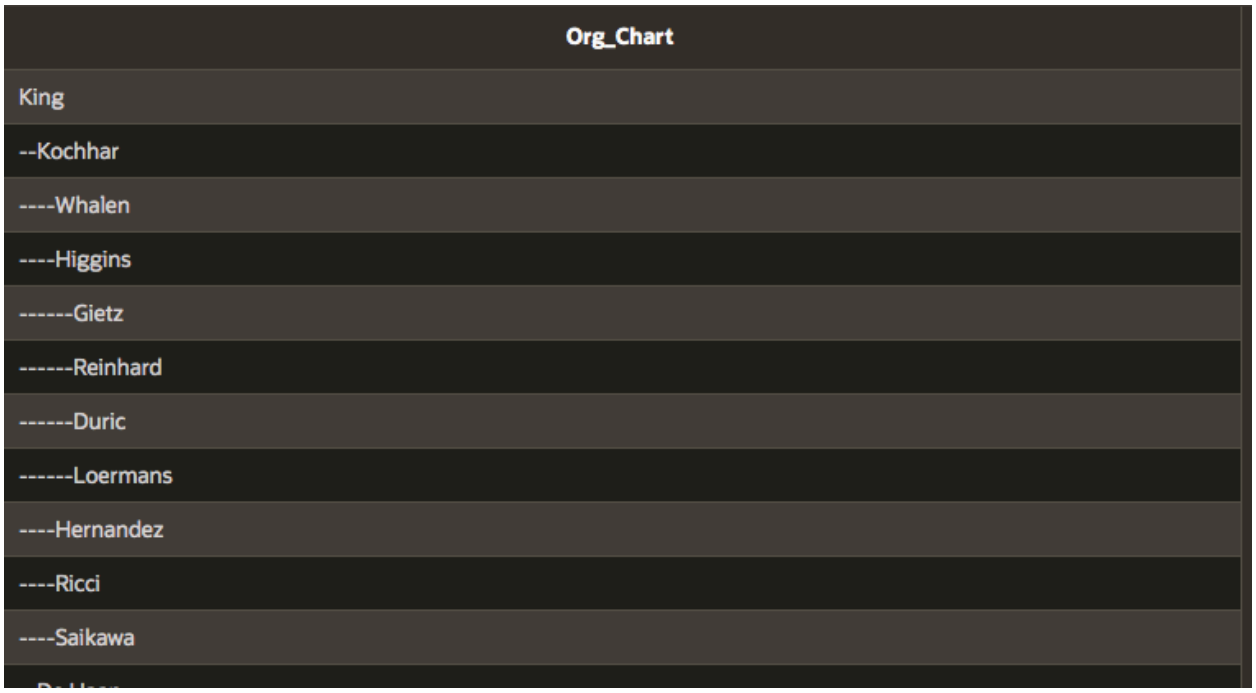
```
SELECT last_name, department_id, salary
FROM employees
START WITH last_name = 'King'
CONNECT BY PRIOR employee_id = manager_id;
```

LAST_NAME	DEPARTMENT_ID	SALARY
King	90	24000
Kochhar	90	17000
Whalen	10	4400
Higgins	110	12000
Gietz	110	8300
Reinhard	110	8100
Duric	110	5400
Loermans	110	5200
Hernandez	10	4300
Ricci	10	4100

6. Create a report that shows the organization chart for the entire employee table. Write the report so that each level will indent each employee 2 spaces. Since Oracle Application Express cannot display the spaces in front of the column, use - (minus) instead.

ANS:

```
SELECT LPAD(last_name, LENGTH(last_name) + (LEVEL * 2) - 2, '-') AS "Org_Chart"
FROM employees
START WITH last_name = (SELECT last_name FROM employees WHERE manager_id IS
NULL)
CONNECT BY PRIOR employee_id = manager_id;
```



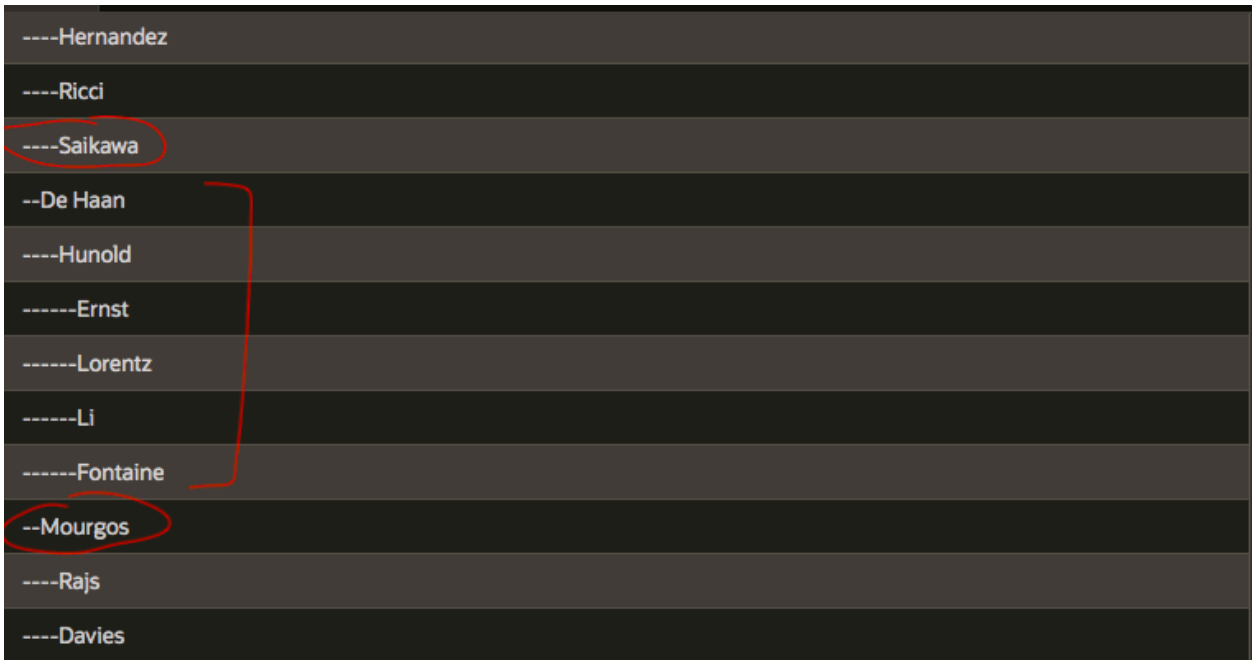
Org_Chart
King
--Kochhar
----Whalen
----Higgins
------Gietz
------Reinhard
------Duric
------Loermans
----Hernandez
----Ricci
----Saikawa

7. Re-write the report from 6 to exclude De Haan and all the people working for him.

ANS:

```
SELECT LPAD(last_name, LENGTH(last_name) + (LEVEL * 2) - 2, '-') AS "Org_Chart"
FROM employees
START WITH last_name = (SELECT last_name FROM employees WHERE manager_id IS
NULL)
CONNECT BY PRIOR employee_id = manager_id
AND last_name != 'De Haan';
```

With De Haan



Without De Haan