Database Programming with SQL

6-1: Cross Joins and Natural Joins

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

```
SELECT E.LAST_NAME, D.DEPARTMENT_NAME
FROM EMPLOYEES E
CROSS JOIN DEPARTMENTS D;
```

Results | Explain | Describe | Saved SQL | History

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

```
SELECT DEPARTMENT_ID, DEPARTMENT_NAME, LOCATION_ID, CITY
FROM DEPARTMENTS
NATURAL JOIN LOCATIONS;
```

Results | Explain | Describe | Saved SQL | History

DEPARTMENT_ID	DEPARTMENT_NAME	LOCATION_ID	CITY
60	IT	1400	Southlake
50	Shipping	1500	South San Francisco
10	Administration	1700	Seattle
90	Executive	1700	Seattle
110	Accounting	1700	Seattle
190	Contracting	1700	Seattle
20	Marketing	1800	Toronto
85	Sales - Americas	2100	Rio de Janeiro
80	Sales - Europe	2500	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

```
SELECT DEPARTMENT_ID, DEPARTMENT_NAME, LOCATION_ID, CITY
FROM DEPARTMENTS
NATURAL JOIN LOCATIONS
WHERE DEPARTMENT_ID IN (20, 50);
```

Results | Explain | Describe | Saved SQL | History

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

6-2: Join Clauses

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

```
SELECT L.LOCATION_ID, L.CITY, D.DEPARTMENT_NAME
FROM LOCATIONS L
JOIN DEPARTMENTS D ON L.LOCATION_ID = D.LOCATION_ID
WHERE L.LOCATION_ID = 1400;
```

Results | Explain | Describe | Saved SQL | History

LOCATION_ID	CITY	DEPARTMENT_NAME
1400	Southlake	IT

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.

```
SELECT DPLI.SONG_ID, DCD.CD_NUMBER, DCD.TITLE, DPLI.COMMENTS
FROM D_PLAY_LIST_ITEMS DPLI
JOIN D_TRACK_LISTINGS DTL ON DPLI.SONG_ID = DTL.SONG_ID
JOIN D_CDS DCD ON DTL.CD_NUMBER = DCD.CD_NUMBER;
```

Results | Explain | Describe | Saved SQL | History

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.

```
SELECT L.CITY, D.DEPARTMENT_NAME, L.LOCATION_ID, D.DEPARTMENT_ID
FROM LOCATIONS L
JOIN DEPARTMENTS D ON L.LOCATION_ID = D.LOCATION_ID
WHERE D.DEPARTMENT_ID IN (10, 20, 30) AND L.CITY = 'Seattle';
```

Results | Explain | Describe | Saved SQL | History

CITY	DEPARTMENT_NAME	LOCATION_ID	DEPARTMENT_ID
Seattle	Administration	1700	10

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

```
SELECT C.COUNTRY_NAME, R.REGION_ID, R.REGION_NAME
FROM COUNTRIES C
JOIN REGIONS R ON C.REGION_ID = R.REGION_ID
WHERE R.REGION_NAME = 'Americas';
```

Results | Explain | Describe | Saved SQL | History

no data found

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.

```
SELECT E.FIRST_NAME, E.LAST_NAME, E.HIRE_DATE, J.JOB_ID, J.JOB_TITLE, J.MAX_SALARY
FROM EMPLOYEES E
JOIN JOBS J ON E.JOB_ID = J.JOB_ID
WHERE J.MAX_SALARY > 12000;
```

Results | Explain | Describe | Saved SQL | History

FIRST_NAME	LAST_NAME	HIRE_DATE	JOB_ID	JOB_TITLE	MAX_SALARY
Shelley	Higgins	07-Jun-2009	AC_MGR	Accounting Manager	16000
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
Michael	Hartstein	17-Feb-2011	MK_MAN	Marketing Manager	15000
Eleni	Zlotkey	29-Jan-2015	SA_MAN	Sales Manager	20000
Sophia	Barbosa Souza	12-Mar-2009	SR_SA_REP	Senior Sales Representative	16000
Nick	Hooper	01-Sep-2012	SR_SA_REP	Senior Sales Representative	16000

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

```
SELECT J.JOB_TITLE, E.FIRST_NAME, E.LAST_NAME, E.EMAIL
FROM EMPLOYEES E
JOIN JOBS J ON E.JOB_ID = J.JOB_ID
WHERE J.JOB_TITLE = 'Stock Clerk';
```

Results | Explain | Describe | Saved SQL | History

JOB_TITLE	FIRST_NAME	LAST_NAME	EMAIL
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.

```
1  SELECT E.EMPLOYEE_ID, E.FIRST_NAME, E.LAST_NAME, E.MANAGER_ID, M.FIRST_NAME AS MANAGER_FIRST_NAME, M.LAST_NAME AS MANAGER_LAST_NAME
2  FROM EMPLOYEES E
3  JOIN EMPLOYEES M ON E.MANAGER_ID = M.EMPLOYEE_ID;
```

Results | Explain | Describe | Saved SQL | History

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	MANAGER_ID	MANAGER_FIRST_NAME	MANAGER_LAST_NAME
101	Neena	Kochhar	100	Steven	King
102	Lex	De Haan	100	Steven	King
124	Kevin	Mourgos	100	Steven	King
149	Eleni	Zlotkey	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

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

```
1  SELECT L.LOCATION_ID, L.CITY, D.DEPARTMENT_NAME
2  FROM LOCATIONS L
3  JOIN DEPARTMENTS D ON L.LOCATION_ID = D.LOCATION_ID
4  WHERE L.COUNTRY_ID = 2;
```

Results | Explain | Describe | Saved SQL | History

LOCATION_ID	CITY	DEPARTMENT_NAME
1800	Toronto	Marketing

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.

```
1  SELECT E.MANAGER_ID, D.DEPARTMENT_ID, D.DEPARTMENT_NAME, E.FIRST_NAME, E.LAST_NAME
2  FROM EMPLOYEES E
3  JOIN DEPARTMENTS D ON E.DEPARTMENT_ID = D.DEPARTMENT_ID
4  WHERE D.DEPARTMENT_ID IN (80, 90, 110, 190);
```

Results | Explain | Describe | Saved SQL | History

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

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

```
SELECT E.EMPLOYEE_ID, E.LAST_NAME, E.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 = TO_DATE('07-JUN-1994', 'DD-MON-YYYY');
```

Results | Explain | Describe | Saved SQL | History

no data found

6-3: Inner vs Outer Joins

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

```
SELECT E.FIRST_NAME, E.LAST_NAME, D.DEPARTMENT_NAME
FROM EMPLOYEES E
LEFT JOIN DEPARTMENTS D ON E.DEPARTMENT_ID = D.DEPARTMENT_ID;
```

Results | Explain | Describe | Saved SQL | History

FIRST_NAME	LAST_NAME	DEPARTMENT_NAME
Jennifer	Whalen	Administration
Katia	Hernandez	Administration
Guido	Ricci	Administration
Mizuto	Saikawa	Administration
Michael	Hartstein	Marketing
Pat	Fay	Marketing
Donna	Steiner	Marketing
Lisa	TAYLOR	Marketing
Michael	Stocks	Marketing
Nabil	Safwah	Marketing

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.

```
SELECT E.FIRST_NAME, E.LAST_NAME, D.DEPARTMENT_NAME
FROM EMPLOYEES E
RIGHT JOIN DEPARTMENTS D ON E.DEPARTMENT_ID = D.DEPARTMENT_ID;
```

Results | Explain | Describe | Saved SQL | History

FIRST_NAME	LAST_NAME	DEPARTMENT_NAME
Guido	Ricci	Administration
Mizuto	Saikawa	Administration
Katia	Hernandez	Administration
Jennifer	Whalen	Administration
Nabil	Safwah	Marketing
Alice	Newton	Marketing
Donna	Steiner	Marketing
Michael	Stocks	Marketing
Lisa	TAYLOR	Marketing
Michael	Hartstein	Marketing

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.

```
1  SELECT E.FIRST_NAME, E.LAST_NAME, D.DEPARTMENT_NAME
2  FROM EMPLOYEES E
3  FULL OUTER JOIN DEPARTMENTS D ON E.DEPARTMENT_ID = D.DEPARTMENT_ID;
```

Results | Explain | Describe | Saved SQL | History

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
Ellen	Abel	Sales - Europe
Jonathon	Taylor	Sales - Europe
Kimberely	Grant	-

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.

```
1  SELECT C.FIRST_NAME, C.LAST_NAME, E.EVENT_DATE, E.DESCRIPTION
2  FROM D_CLIENTS C
3  LEFT JOIN D_EVENTS E ON C.CLIENT_NUMBER = E.CLIENT_NUMBER;
```

Results | Explain | Describe | Saved SQL | History

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.

```
1  SELECT S.DESCRIPTION AS SHIFT_DESCRIPTION,
2         SA.SHIFT_ASSIGN_DATE
3  FROM F_SHIFTS S
4  LEFT JOIN F_SHIFT_ASSIGNMENTS SA ON S.CODE = SA.CODE;
```

Results | Explain | Describe | Saved SQL | History

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

6-4: Self Joins and Hierarchical Queries

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.

```
SELECT E.LAST_NAME AS Employee,
       E.EMPLOYEE_ID AS "Emp#",
       M.LAST_NAME AS Manager,
       M.EMPLOYEE_ID AS "Mgr#"
FROM EMPLOYEES E
JOIN EMPLOYEES M ON E.MANAGER_ID = M.EMPLOYEE_ID;
```

Results | Explain | Describe | Saved SQL | History

EMPLOYEE	Emp#	MANAGER	Mgr#
Hunold	103	De Haan	102
Fay	202	Hartstein	201
Newton	235	Hartstein	201
Safwah	228	Hartstein	201
Steiner	215	Hartstein	201
Stocks	219	Hartstein	201
TAYLOR	217	Hartstein	201
Duric	231	Higgins	205
Gietz	206	Higgins	205

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.

```
SELECT E.LAST_NAME AS Employee,
       E.EMPLOYEE_ID AS "Emp#",
       M.LAST_NAME AS Manager,
       M.EMPLOYEE_ID AS "Mgr#"
FROM EMPLOYEES E
LEFT JOIN EMPLOYEES M ON E.MANAGER_ID = M.EMPLOYEE_ID
ORDER BY E.LAST_NAME;
```

Results | Explain | Describe | Saved SQL | History

EMPLOYEE	Emp#	MANAGER	Mgr#
Abel	174	Zlotkey	149
Almeida Castro	210	Zlotkey	149
Alves Rocha	209	Zlotkey	149
Barbosa Souza	207	Zlotkey	149
Bell	216	Mourgos	124
Davies	142	Mourgos	124
De Haan	102	King	100
Duric	231	Higgins	205

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.

```
SELECT E.LAST_NAME AS Employee,
       E.HIRE_DATE AS "Emp Hired",
       M.LAST_NAME AS Manager,
       M.HIRE_DATE AS "Mgr Hired"
FROM EMPLOYEES E
JOIN EMPLOYEES M ON E.MANAGER_ID = M.EMPLOYEE_ID
WHERE E.HIRE_DATE < M.HIRE_DATE;
```

Results | Explain | Describe | Saved SQL | History

EMPLOYEE	Emp Hired	MANAGER	Mgr Hired
Whalen	17-Sep-2002	Kochhar	21-Sep-2004
Hunold	03-Jan-2005	De Haan	13-Jan-2008
Rajs	17-Oct-2010	Mourgos	16-Nov-2014
Davies	29-Jan-2012	Mourgos	16-Nov-2014
Matos	15-Mar-2013	Mourgos	16-Nov-2014
Vargas	09-Jul-2013	Mourgos	16-Nov-2014
Bell	01-Apr-2014	Mourgos	16-Nov-2014
Abel	11-May-2011	Zlotkey	29-Jan-2015

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

```
SELECT LAST_NAME, SALARY, DEPARTMENT_ID
FROM EMPLOYEES
WHERE DEPARTMENT_ID = (SELECT DEPARTMENT_ID FROM EMPLOYEES WHERE LAST_NAME = 'De Haan');
```

Results | Explain | Describe | Saved SQL | History

LAST_NAME	SALARY	DEPARTMENT_ID
King	24000	90
Kochhar	17000	90
De Haan	17000	90

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;

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

Results | Explain | Describe | Saved SQL | History

LAST_NAME	DEPARTMENT_ID	SALARY
King	90	24000

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.

```
SELECT LPAD(' ', LEVEL * 2, '-') || LAST_NAME AS EMPLOYEE_NAME,
       EMPLOYEE_ID
FROM EMPLOYEES
START WITH MANAGER_ID IS NULL
CONNECT BY PRIOR EMPLOYEE ID = MANAGER ID;
```

Results | Explain | Describe | Saved SQL | History

EMPLOYEE_NAME	EMPLOYEE_ID
- King	100
--- Kochhar	101
----- Whalen	200
----- Higgins	205
------- Gietz	206
------- Reinhard	224
------- Duric	231
------- Loermans	232
----- Hernandez	225

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

```
SELECT LPAD(' ', LEVEL * 2, '-') || LAST_NAME AS EMPLOYEE_NAME,EMPLOYEE_ID
FROM EMPLOYEES
START WITH MANAGER_ID IS NULL
CONNECT BY PRIOR EMPLOYEE_ID = MANAGER_ID
AND EMPLOYEE_ID NOT IN (SELECT EMPLOYEE_ID FROM EMPLOYEES
                        CONNECT BY PRIOR EMPLOYEE_ID = MANAGER_ID START WITH LAST_NAME = 'De Haan');
```

Results | Explain | Describe | Saved SQL | History

EMPLOYEE_NAME	EMPLOYEE_ID
- King	100
--- Kochhar	101
----- Whalen	200
----- Higgins	205
------- Gietz	206
------- Reinhard	224
------- Duric	231
------- Loermans	232