Angela Zhang

Programming 6

6-1: Cross Joins and Natural Joins

Practice Activities

Objectives

- Construct and execute a natural join using ANSI-99 SQL join syntax
- Create a cross join using ANSI-99 SQL join syntax
- Explain the importance of having a standard for SQL as defined by ANSI
- Describe a business need for combining information from multiple data sources

Vocabulary

Identify the vocabulary word for each definition below.

Cross Join	Returns the Cartesian product from two tables.
Inner Join	Joins two tables based on the same column name.

Try It / Solve It

Use the Oracle database for problems 1-3.

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

```
SELECT last_name, department_name
FROM employees
CROSS JOIN departments;
```

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;
```

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);
```

6-2: Join Clauses

Practice Activities

Objectives

- Construct and execute a natural join using ANSI-99 SQL join syntax
- Create a cross join using ANSI-99 SQL join syntax
- Explain the importance of having a standard for SQL as defined by ANSI
- Describe a business need for combining information from multiple data sources

Vocabulary

Identify the vocabulary word for each definition below.

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

Try It / Solve It

Use the Oracle database for problems 1-6.

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

```
SELECT d.department_id, d.department_name, l.location_id, l.city
FROM departments d
JOIN locations l
ON d.location_id = l.location_id
WHERE l.location_id = 1400;
```

OR

```
SELECT departments.department_id, departments.department_name, locations.location_id, locations.city
FROM departments
JOIN locations
ON departments.location_id = locations.location_id
WHERE locations.location_id = 1400;
```

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

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

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

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

```
SELECT countries.country_name, countries.region_id, regions.region_name
FROM countries
INNER JOIN regions
ON countries.region_id = regions.region_id
WHERE regions.region_name = 'Americas';
```

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 employees.first_name, employees.last_name, employees.hire_date, employees.job_id,
jobs.job_title, jobs.max_salary
FROM employees
INNER JOIN jobs
ON employees.job_id = jobs.job_id
WHERE jobs.max_salary > 12000;
```

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

```
SELECT jobs.job_title, employees.first_name, employees.last_name, employees.email
FROM employees
JOIN jobs
ON employees.job_id = jobs.job_id
WHERE jobs.job_title = 'Stock Clerk';
```

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.

```
SELECT employee_id, first_name AS employee_first_name, last_name AS
employee_last_name, manager_id, first_name AS manager_first_name, last_name AS
manager_last_name
FROM employees
LEFT JOIN employees
ON e.manager_id = employee_id;
```

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

```
SELECT locations.location_id, locations.city, departments.department_name
FROM locations
JOIN departments
ON locations.location_id = departments.location_id
WHERE locations.country_id = 'CA'; -- Assuming 'CA' is the country ID for Canada
```

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.

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

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

```
SELECT employee_id, last_name, department_id, department_name, hire_date
FROM employees
JOIN departments ON department_id = department_id
WHERE hire_date = TO_DATE('1994-06-07', 'YYYY-MM-DD');
```

6-3: Inner versus Outer Joins

Practice Activities

Objectives

- Compare and contrast an inner and an outer join
- Construct and execute a query to use a left outer join
- Construct and execute a query to use a right outer join
- Construct and execute a query to use a full outer join

Vocabulary

Identify the vocabulary word for each definition below.

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

Try It / Solve It

Use the Oracle database for problems 1-7.

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
```

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;
```

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.

```
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;
```

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.

```
SELECT c.first_name, c.last_name, e.event_date, e.description
FROM clients c
LEFT JOIN events e
ON c.client_id = e.client_id;
```

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.

```
SELECT s.shift_description, sa.assignment_date
FROM shifts s
LEFT JOIN shift_assignments sa
ON s.shift_id = sa.shift_id;
```

6-4: Self Joins and Hierarchical Queries

Practice Activities

Objectives

- Construct and execute a SELECT statement to join a table to itself using a self-join
- Interpret the concept of a hierarchical query
- Create a tree-structured report
- Format hierarchical data
- Exclude branches from the tree structure

Vocabulary

Identify the vocabulary word for each definition below.

SELF JOIN	Joins a table to itself
HIERARCHY QUERY	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
ROOT ROW	Identifies the beginning row for a hierarchical query
CONNECT BY	Specifies the relationship between parent rows and child rows of a hierarchical query

Try It / Solve It

For each problem, use the Oracle database.

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 last_name AS Employee, employee_id AS Emp#, last_name AS Manager,employee_id
AS Mgr#
FROM employees
JOIN employees ON manager_id = employee_id;
```

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 last_name AS Employee, e.employee_id AS Emp#, last_name AS Manager,
employee_id AS Mgr#
FROM employees
LEFT JOIN employees ON manager_id = employee_id
ORDER BY last_name;
```

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 last_name AS Employee, hire_date AS "Emp Hired", last_name AS Manager,
hire_date AS "Mgr Hired"
FROM employees
JOIN employees ON manager_id = employee_id
WHERE hire_date < hire_date;
```

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
START WITH last_name = 'De Haan'
CONNECT BY PRIOR employee_id = manager_id;
```

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;
```

Change “CONNECT BY PRIOR manager_id = employee_id” to “CONNECT BY PRIOR employee_id = manager_id”

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('-', 2 * (LEVEL - 1)) || last_name AS Employee, employee_id, manager_id
FROM employees
START WITH manager_id IS NULL
CONNECT BY PRIOR employee_id = manager_id;
```

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

```
SELECT LPAD('-', 2 * (LEVEL - 1)) || last_name AS Employee, employee_id, manager_id
FROM employees
START WITH manager_id IS NULL
CONNECT BY PRIOR employee_id = manager_id
AND PRIOR employee_id NOT IN (SELECT employee_id FROM employees WHERE
last_name = 'De Haan');
```