**Database Programming with SQL  
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.

* Returns the Cartesian product from two tables.  
  cross join
* Joins two tables based on the same column name.

Natural join

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 e.last\_name, d.department\_name

FROM employees e

CROSS JOIN departments d;

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 = 20 OR department\_id = 50;

**Database Programming with SQL  
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.

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

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

FROM locations l

JOIN departments d ON l.location\_id = d.location\_id

WHERE l.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 song\_id, cd\_number, title, 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 city, department\_name, location\_id, department\_id

FROM locations

JOIN departments USING (location\_id)

WHERE city = 'Seattle'

AND department\_id IN (10, 20, 30);

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

SELECT country\_name, region\_id, region\_name

FROM countries

JOIN regions USING (region\_id)

WHERE 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 first\_name, last\_name, hire\_date, employees.job\_id, job\_title, maximum\_salary

FROM employees

JOIN jobs ON employees.job\_id = jobs.job\_id

WHERE maximum\_salary > 12000;

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

SELECT job\_title, first\_name, last\_name, email

FROM employees

WHERE 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 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

FROM employees e

JOIN employees m ON e.manager\_id = m.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 = '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 employees.manager\_id, employees.department\_id, departments.department\_name, employees.first\_name, employees.last\_name

FROM employees

JOIN departments USING (department\_id)

WHERE employees.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 employees.employee\_id, employees.last\_name, employees.department\_id, departments.department\_name, employees.hire\_date

FROM employees

JOIN departments ON employees.department\_id = departments.department\_id

WHERE employees.hire\_date = '1994-06-07';

**Database Programming with SQL  
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.

* 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.  
  full outer join
* A join that returns the unmatched rows as well as matched rows  
  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.  
  left 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.  
  right outer join
* A join of two or more tables that returns only matched rows  
  inner join

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 first\_name, last\_name, department\_name

FROM employees

LEFT JOIN departments ONdepartment\_id = 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 first\_name, last\_name, department\_name

FROM employees

FULL OUTER JOIN departments ON department\_id =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 first\_name, last\_name, department\_name

FROM employees

FULL OUTER JOIN departments ON department\_id = 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 first\_name, last\_name, event\_date, description

FROM clients

LEFT JOIN events ON client\_id = 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 shift\_description, assignment\_date

FROM shifts

LEFT JOIN shift\_assignments ON shift\_id = shift\_assignments\_id;

**Database Programming with SQL  
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.

* Joins a table to itself  
  self join
* Retrieves data based on a natural hierarchical relationship  
  between rows in a table  
  hierarchical query
* Determines the number of steps down from the beginning row  
  that should be returned by a hierarchical query  
  level
* Identifies the beginning row for a hierarchical query  
  start with
* Specifies the relationship between parent rows and child rows of  
  a hierarchical query  
  connect by

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 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;

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.

ELECT 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;

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;

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 employee\_id = (SELECT employee\_id FROM employees WHERE last\_name = 'De Haan' AND first\_name = 'Lex')

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;

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", salary, department\_id

FROM employees

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('-', LEVEL \* 2, '-') || last\_name AS "Employee Name", salary, department\_id

FROM employees