Suggested Solution

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

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.

   Solution:

   ```
   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.

   Solution:

   ```
   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.

   Solution:

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

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

Use the Oracle database for problems 1-7.

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

   Solution:

   SELECT l.city, d.department_name

   FROM locations l JOIN departments d USING (location_id)
   WHERE 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.

   Solution:
   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.

   Solution:
   SELECT l.city, d.department_name, location_id, d.department_id
   FROM locations l JOIN departments d USING (location_id)
   WHERE city = 'Seattle'
   AND department_id IN (10, 20, 30);

4. 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.

Solution:

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

The following questions use the JOIN…ON syntax:

5. 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.

Solution:

```
SELECT worker.employee_id, worker.first_name, worker.last_name, worker.manager_id,
manager.first_name AS "Manager First Name", manager.Last_name AS "Manager Last Name"
FROM employees worker JOIN employees manager
ON (worker.manager_id = manager.employee_id);
```

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

Solution 1:

```
SELECT l.location_ID, city, department_name

FROM locations l JOIN departments d ON (l.location_id = d.location_id)

JOIN countries c ON (l.country_id= c.country_id)

WHERE country_name='Canada';
```

Solution 2:

```
SELECT l.location_ID, l.city, d.department_name

FROM locations l JOIN departments d ON (l.location_id = d.location_id)

JOIN countries c ON (l.country_id= c.country_id)
WHERE c.country_name='Canada';
```

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

Solution:

```
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 hire_date = '07-Jun-1994';
```

ORACLE
Academy

academy.oracle.com

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

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.

   Solution:
   ```
   SELECT e.first_name, e.last_name, d.department_id
   FROM employees e
   LEFT OUTER 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.

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

3. 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.

   Solution:
   ```
   SELECT s.description, a.shift_assign_date
   FROM f_shifts s LEFT OUTER JOIN f_shift_assignments a ON (s.code = a.code);
   ```

ORACLE
Academy

academy.oracle.com

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

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.

Solution:

```
    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);
```

Database Programming with SQL
7-1: Oracle Equijoin and Cartesian Product Practice Solutions

Vocabulary

1. Write a query to display the title, type, description, and artist from the DJs on Demand database.

   Solution:
   ```
   SELECT s.title, s.artist, t.description
   FROM d_songs s, d_types t
   WHERE s.type_code = t.code;
   ```

2. Rewrite the query in question 2 to select only those titles with an ID of 47 or 48.

   Solution:
   ```
   SELECT s.title, s.artist, t.description
   FROM d_songs s, d_types t
   WHERE s.type_code = t.code AND s.ID IN(47,48);
   ```

Database Programming with SQL
8-1: Group Functions
Practice Activities

1. Create a query that will return the average order total for all Global Fast Foods orders from January 1, 2002, to December 21, 2002.

   Solution:

   ```
   SELECT AVG(order_total)
   FROM f_orders
   WHERE order_date BETWEEN '01-Jan-2002' AND '21-Dec-2002';
   ```

2. What was the hire date of the last Oracle employee hired?

   Solution:

   ```
   SELECT MAX(hire
   FROM employees;
   ```

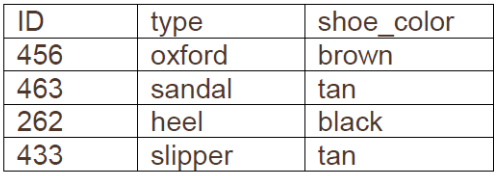
Oracle Academy

Database Programming with SQL
8-2: COUNT, DISTINCT, NVL
Practice Activities

1. How many songs are listed in the DJs on Demand D_SONGS table?

   Solution:
   SELECT COUNT(*) FROM d_songs;

2. What values will be returned when the statement below is issued?



ID	type	shoe_color
456	oxford	brown
463	sandal	tan
262	heel	black
433	slipper	tan

```
SELECT COUNT(shoe_color),
COUNT(DISTINCT shoe_color)
FROM shoes;
```

Solution:

COUNT = 4 DISTINCT = 3