ORACLE
Academy

academy.oracle.com

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.

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

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 department_id,department_name, location_id, city      .
FROM departments JOIN locations 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.

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

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 departments JOIN locations USING (location_id) WHERE department_id in (10, 20 , 30) AND city = 'Seattle'
```

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.

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

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.

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

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.

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

5.

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

6.

```
SELECT  job_title,  first_name,  last_name, LOWER(email) || 'somecomname.sometld' as email
FROM employees JOIN jobs USING(job_id)
WHERE  job_title = 'Stock Clerk' ;
```

7.

```
SELECT a.employee_id,a.first_name,a.last_name,b.manager_id,b.first_name,b.last_name
FROM employees a JOIN employees b ON (b.manager_id=a.employee_id);
```

8.

```
SELECT a.location_id, b.city, a.department_name
FROM departments a JOIN locations  b ON a.location_id = b.location_id JOIN countries c ON b.country_id =
c.country_id
WHERE c.country_name = 'Canada';
```

9.

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

10.

```
SELECT a.Employee_id, a.last_name , a.department_id , b.department_name , a.hire_date
FROM  employees a JOIN departments b ON a.department_id = b.department_id
WHERE a.hire_date = TO_DATE('June 7, 1994', 'Month DD, YYYY');
```