

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

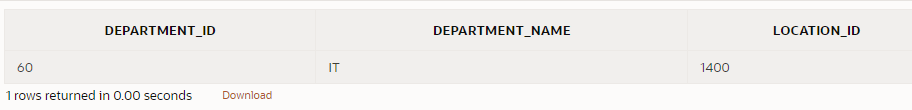
Use the Oracle database for problems 1-6.

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**

**FROM departments JOIN locations USING (location\_id)**

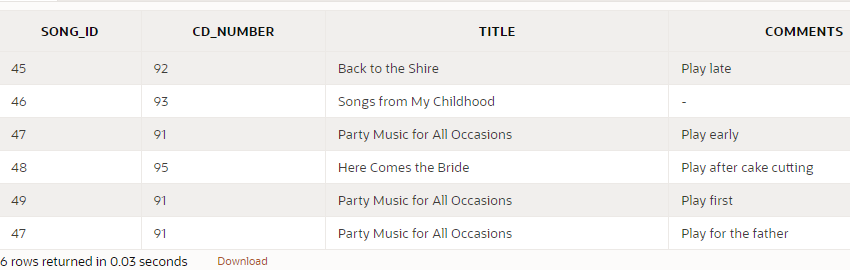
**WHERE location\_id = 1400;**

****

1. 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);**

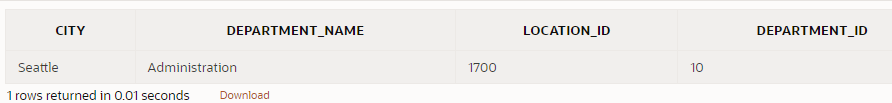


1. Display the city, department name, location ID, and department ID for departments 10, 20, and 30 for the city of Seattle.

**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';**

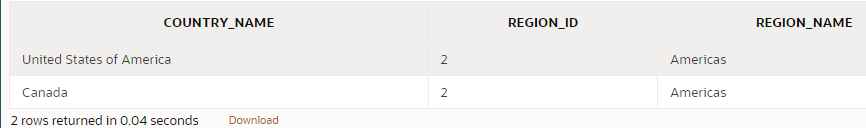
****

1. 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';**

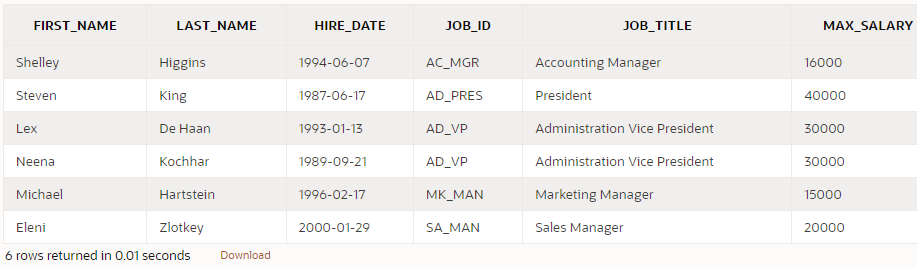
****

1. 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, job\_id, job\_title, max\_salary**

**FROM employees JOIN jobs USING (job\_id)**

**WHERE max\_salary > 12000;**

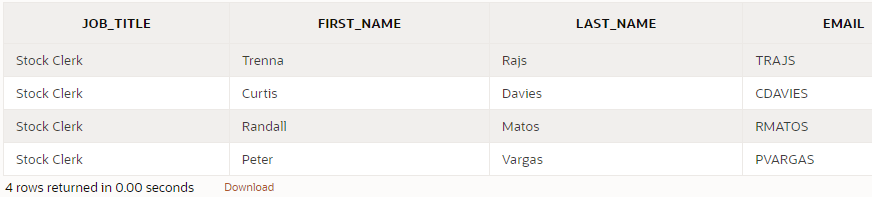
****

1. 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 JOIN jobs USING(job\_id)**

**WHERE job\_title = 'Stock Clerk' ;**

****

The following questions use the JOIN…ON syntax:

1. 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 emp.employee\_id AS "employee ID", emp.first\_name AS "first name", emp.last\_name "last name", emp.manager\_id "manager ID", mgr.first\_name "manager first name", mgr.last\_name "manager last name"**

**FROM employees emp JOIN employees mgr ON emp.manager\_id = mgr.employee\_id;**



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

**SELECT dp.location\_id, loc.city, dp.department\_name**

**FROM departments dp INNER JOIN locations loc ON dp.location\_id = loc.location\_id INNER JOIN countries ct ON loc.country\_id = ct.country\_id**

**WHERE ct.country\_name = 'Canada';**

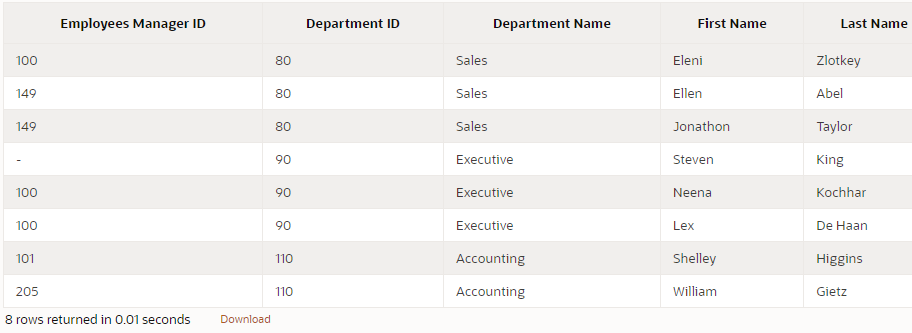
****

1. 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 emp.manager\_id "Employees Manager ID", emp.department\_id "Department ID", dpt.department\_name "Department Name", emp.first\_name "First Name", emp.last\_name "Last Name"**

**FROM employees emp INNER JOIN departments dpt ON emp.department\_id = dpt.department\_id**

**WHERE emp.department\_id in (80, 90, 110, 190);**

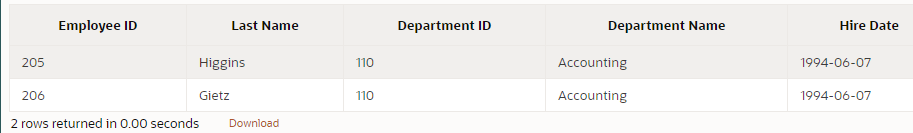
****

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

**SELECT emp.Employee\_id "Employee ID", emp.last\_name "Last Name" , emp.department\_id "Department ID", dpt.department\_name "Department Name", emp.hire\_date "Hire Date"**

**FROM employees emp LEFT JOIN departments dpt ON emp.department\_id = dpt.department\_id**

**WHERE emp.hire\_date = TO\_DATE('June 7, 1994', 'Month DD, YYYY');**

****

Copyright © 2020, Oracle and/or its affiliates. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.