

Database Programming with SQL

* 1. : 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 |
| **hierarchical 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 |
| **START WITH** | Identifies the beginning row for a hierarchical query |
| **CONNECT BY** | Specifies the relationship between parent rows and child rows of a hierarchical query |

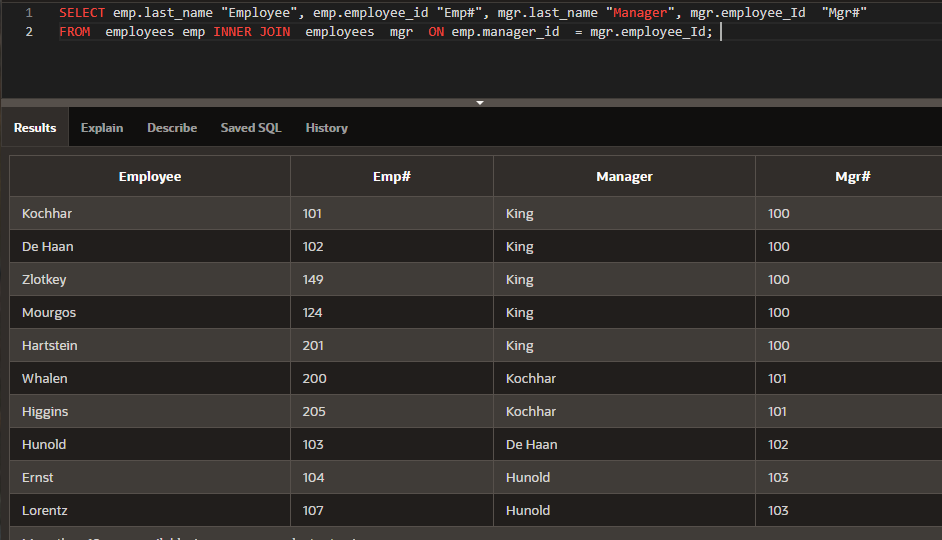
# Try It / Solve It

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 emp.last\_name "Employee", emp.employee\_id "Emp#", mgr.last\_name "Manager", mgr.employee\_Id "Mgr#"

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

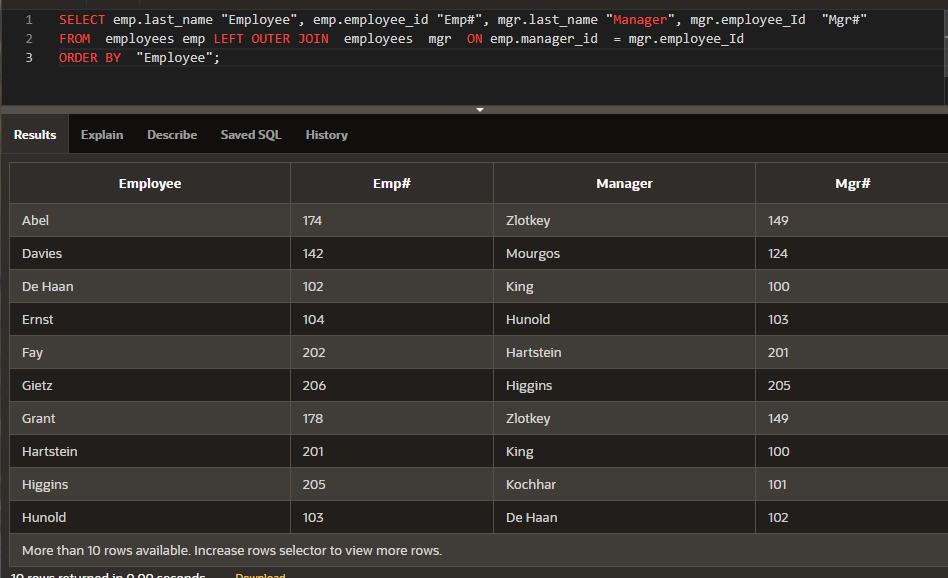


1. Modify question 1 to display all employees and their managers, even if the employee does not have a manager. Order the list alphabetically by the last name of the employee.

**SELECT emp.last\_name "Employee", emp.employee\_id "Emp#", mgr.last\_name "Manager", mgr.employee\_Id  "Mgr#"**

**FROM  employees emp LEFT OUTER JOIN  employees  mgr  ON emp.manager\_id  = mgr.employee\_Id**

**ORDER BY  "Employee";**



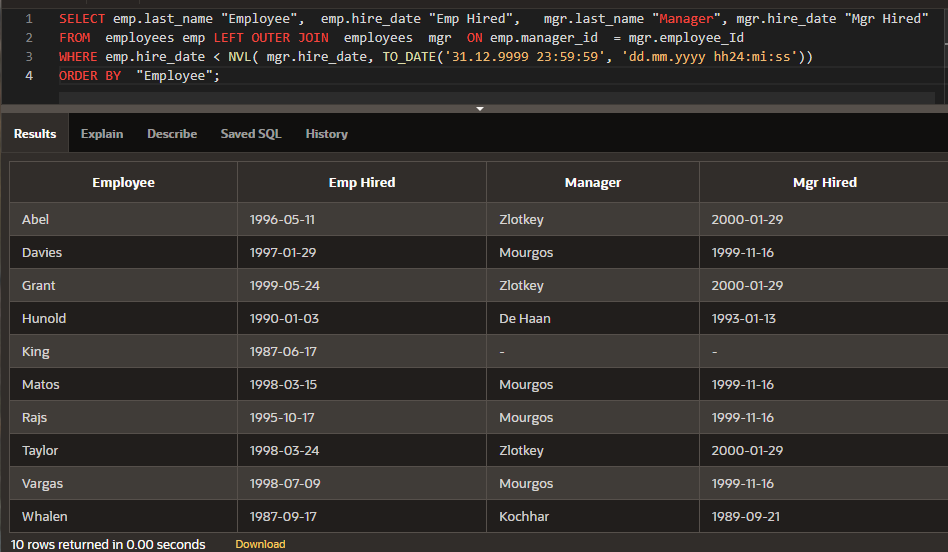
1. Display the names and hire dates for all employees who were hired before their managers, along with their managers’ names and hire dates. Label the columns Employee, Emp Hired, Manager and Mgr Hired, respectively.

**SELECT emp.last\_name "Employee",  emp.hire\_date "Emp Hired",   mgr.last\_name "Manager", mgr.hire\_date "Mgr Hired"**

**FROM  employees emp LEFT OUTER JOIN  employees  mgr  ON emp.manager\_id  = mgr.employee\_Id**

**WHERE emp.hire\_date < NVL( mgr.hire\_date, TO\_DATE('31.12.9999 23:59:59', 'dd.mm.yyyy hh24:mi:ss'))**

**ORDER BY  "Employee";**



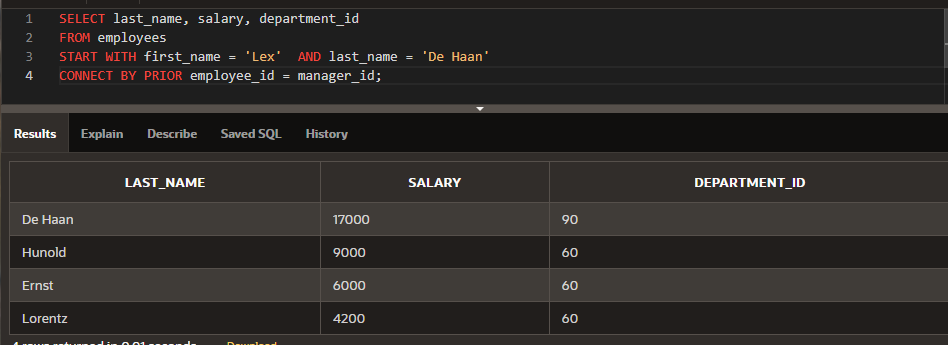
1. Write a report that shows the hierarchy for Lex De Haans department. Include last name, salary, and department id in the report.

**SELECT last\_name, salary, department\_id**

**FROM employees**

**START WITH first\_name = 'Lex'  AND last\_name = 'De Haan'**

**CONNECT BY PRIOR employee\_id = manager\_id;**



1. 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;

С точки зрения SQL Все в порядке

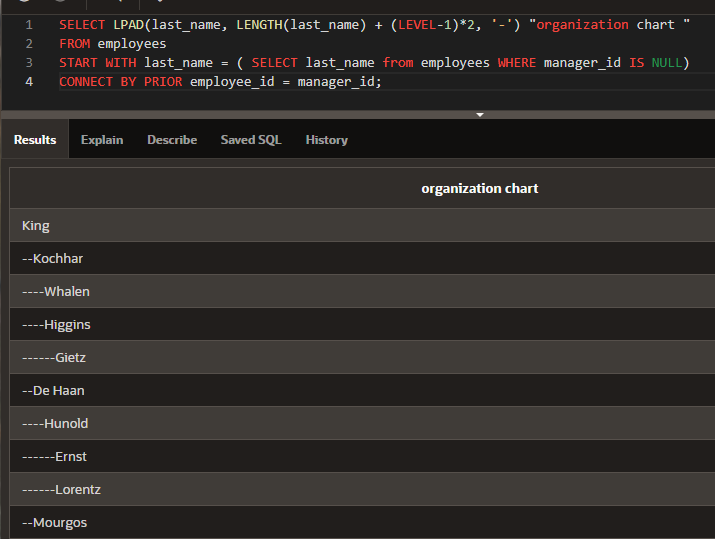
1. Create a report that shows the organization chart for the entire employee table. Write the report so that each level will indent each employee 2 spaces. Since Oracle Application Express cannot display the spaces in front of the column, use - (minus) instead.

**SELECT LPAD(last\_name, LENGTH(last\_name) + (LEVEL-1)\*2, '-') "organization chart "**

**FROM employees**

**START WITH last\_name = ( SELECT last\_name from employees WHERE manager\_id IS NULL)**

**CONNECT BY PRIOR employee\_id = manager\_id;**



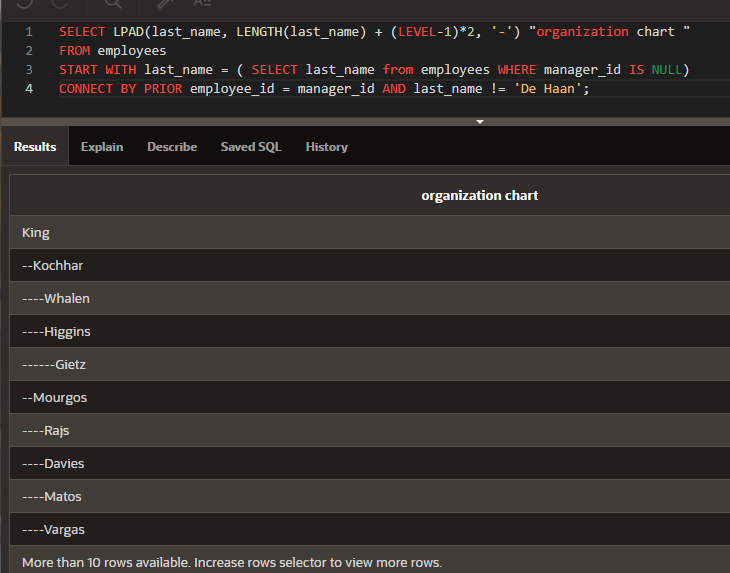
1. Re-write the report from 6 to exclude De Haan and all the people working for him.

**SELECT LPAD(last\_name, LENGTH(last\_name) + (LEVEL-1)\*2, '-') "organization chart "**

**FROM employees**

**START WITH last\_name = ( SELECT last\_name from employees WHERE manager\_id IS NULL)**

**CONNECT BY PRIOR employee\_id = manager\_id AND last\_name != 'De Haan';**



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.