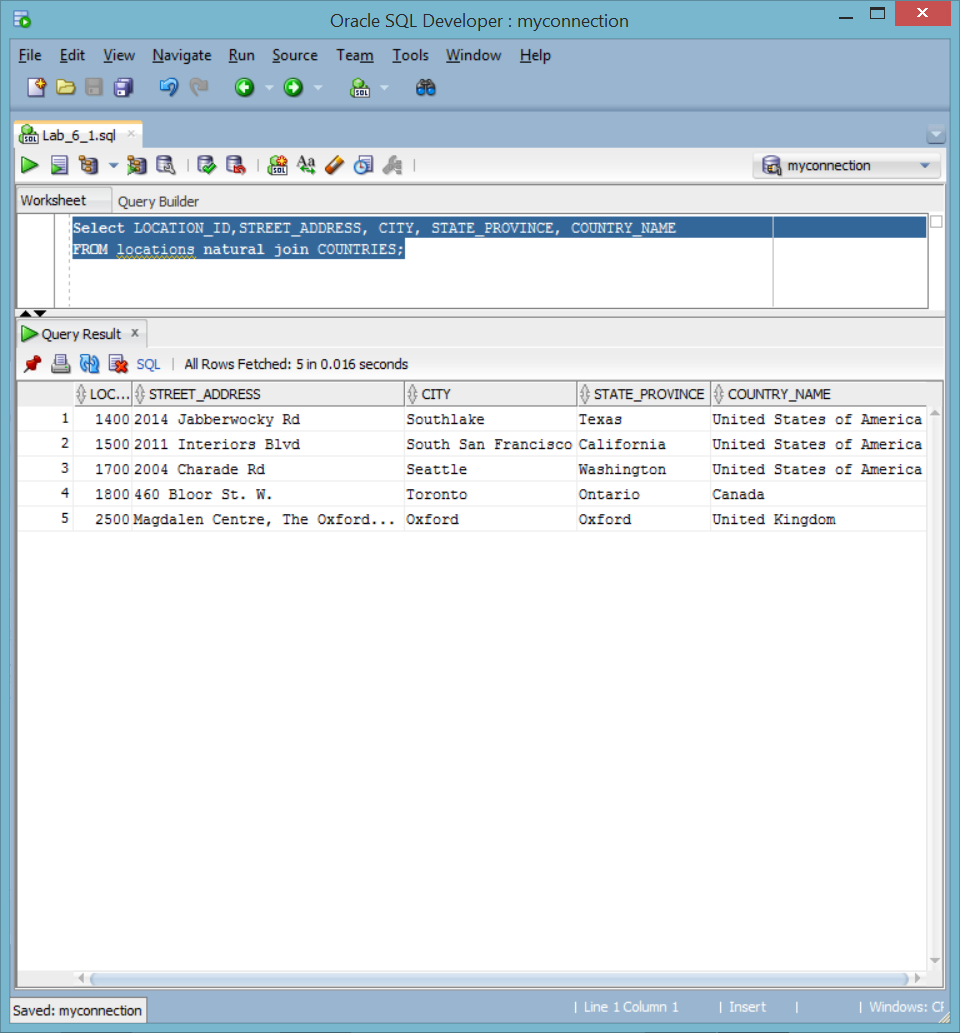
ASSIGNMENT 6: Practice 6-1: Displaying Data from Multiple Tables

1) Write a query for the HR department to produce the addresses of all the departments. Use the LOCATIONS and COUNTRIES tables. Show the location ID, street address, city, state or province, and country in the output. Use a NATURAL JOIN to produce the results.

Select LOCATION\_ID,STREET\_ADDRESS, CITY, STATE\_PROVINCE, COUNTRY\_NAME

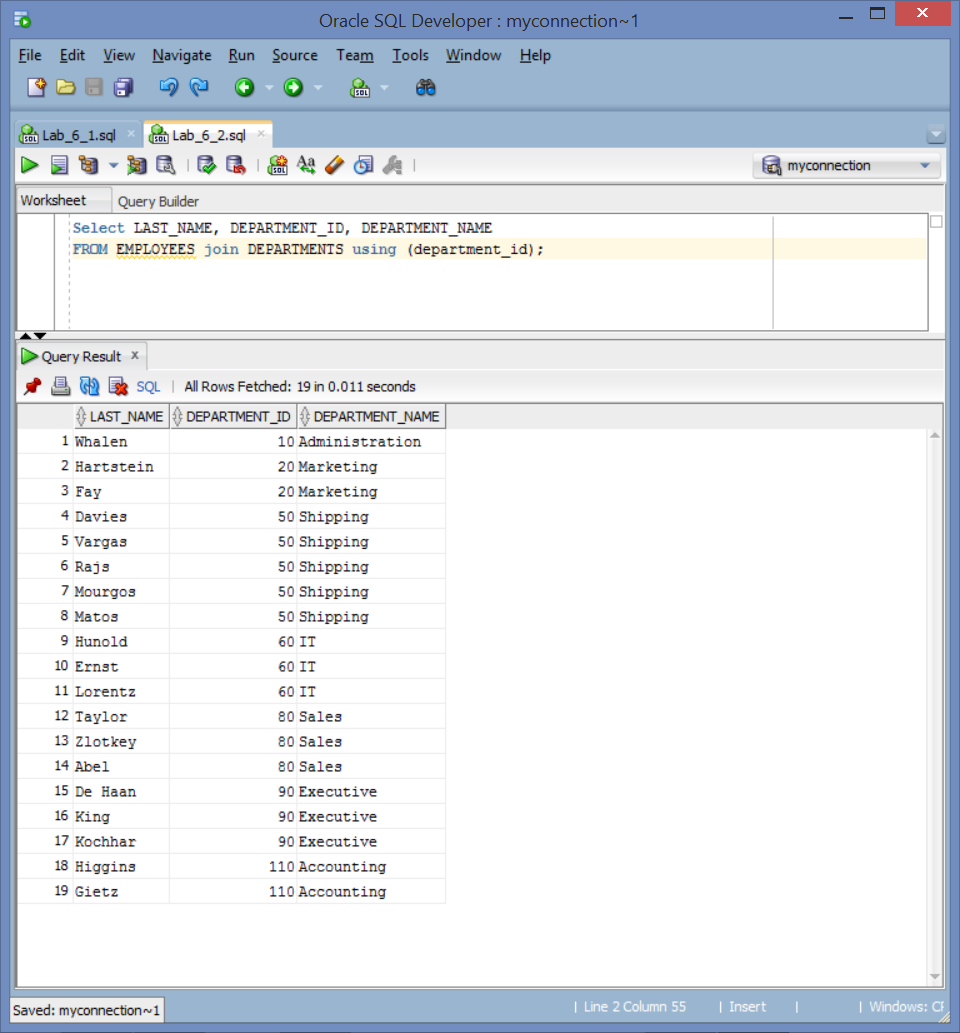
FROM locations natural join COUNTRIES;



2) The HR department needs a report of only those employees with corresponding departments. Write a query to display the last name, department number, and department name for these employees.

Select LAST\_NAME, DEPARTMENT\_ID, DEPARTMENT\_NAME

FROM EMPLOYEES join DEPARTMENTS using (department\_id);



3) The HR department needs a report of employees in Toronto. Display the last name, job, department number, and the department name for all employees who work in Toronto.

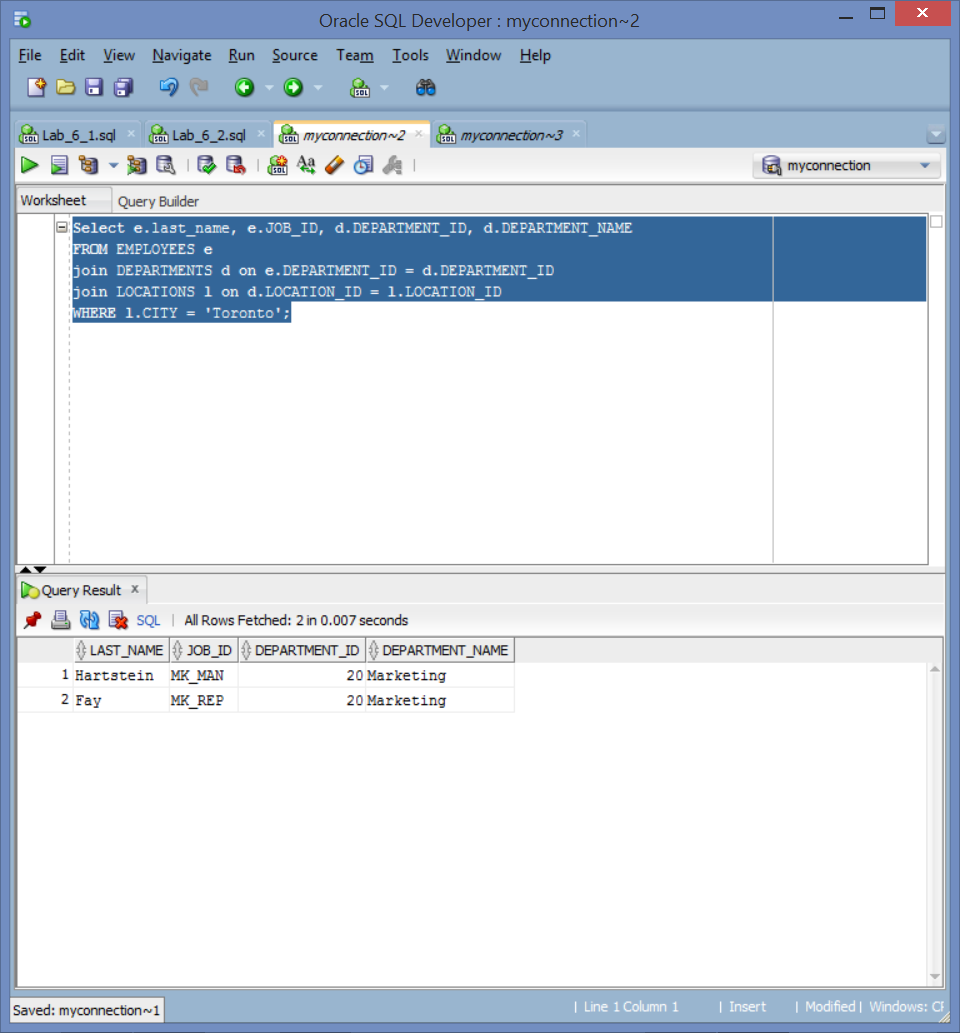
Select e.last\_name, e.JOB\_ID, d.DEPARTMENT\_ID, d.DEPARTMENT\_NAME

FROM EMPLOYEES e

join DEPARTMENTS d on e.DEPARTMENT\_ID = d.DEPARTMENT\_ID

join LOCATIONS l on d.LOCATION\_ID = l.LOCATION\_ID

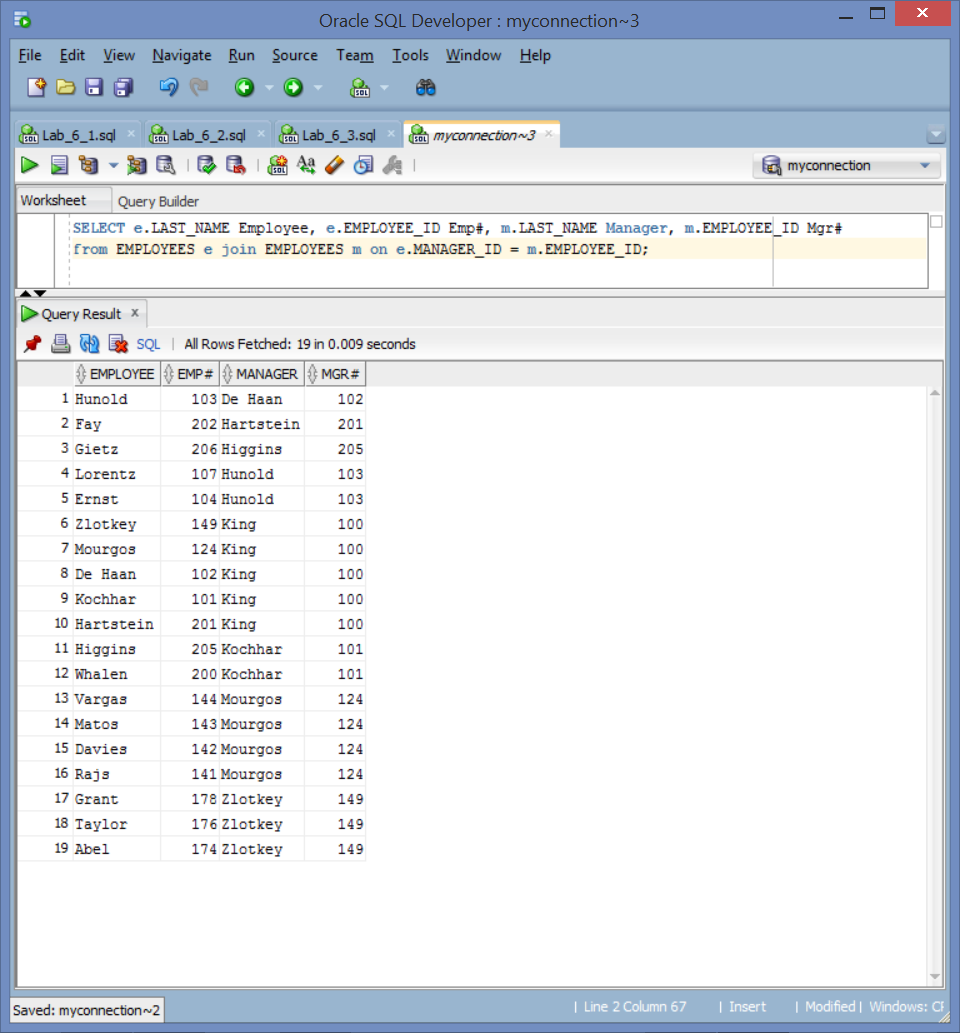
WHERE l.CITY = 'Toronto';



4) Create a report to display employees’ last name and employee number along with their manager’s last name and manager number. Label the columns Employee, Emp#, Manager, and Mgr#, respectively. Save your SQL statement as lab\_06\_04.sql. Run the query.

SELECT e.LAST\_NAME Employee, e.EMPLOYEE\_ID Emp#, m.LAST\_NAME Manager, m.EMPLOYEE\_ID Mgr#

from EMPLOYEES e join EMPLOYEES m on e.MANAGER\_ID = m.EMPLOYEE\_ID;

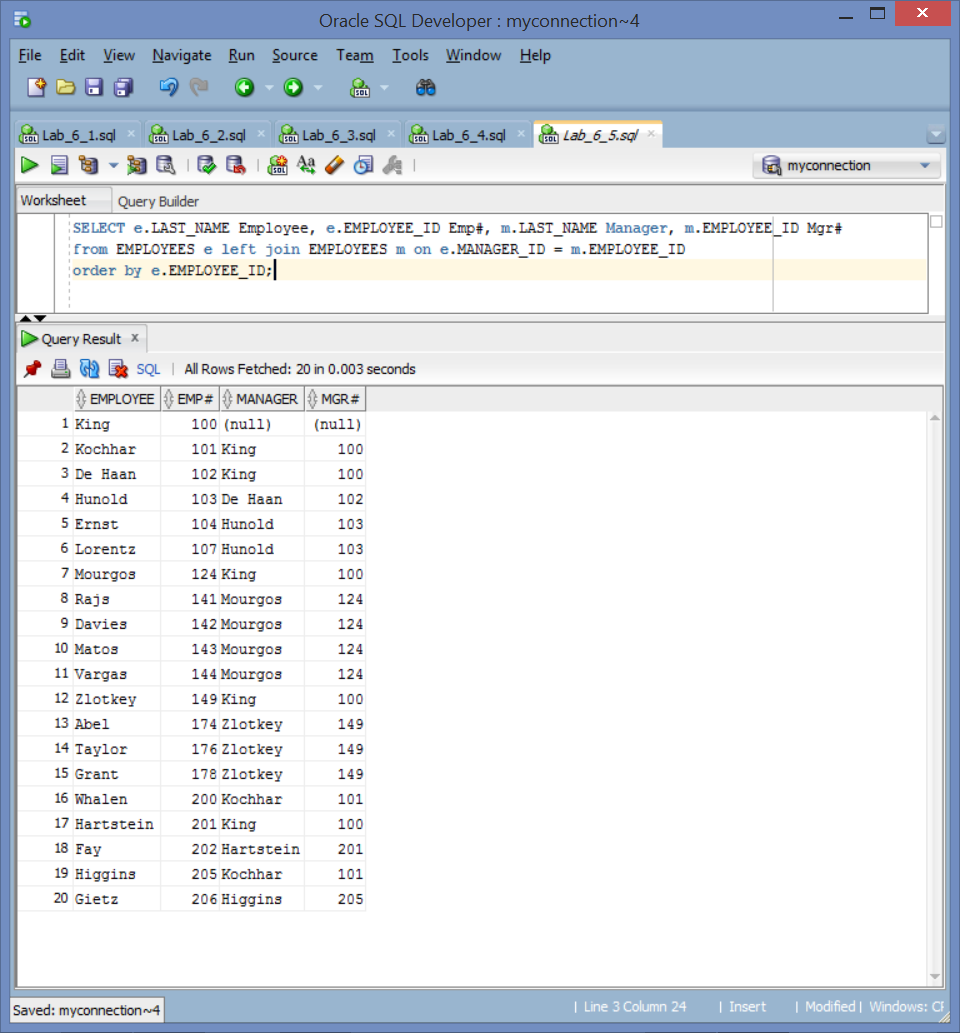


5) Modify lab\_06\_04.sql to display all employees including King, who has no manager. Order the results by the employee number. Save your SQL statement as lab\_06\_05.sql. Run the query in lab\_06\_05.sql.

SELECT e.LAST\_NAME Employee, e.EMPLOYEE\_ID Emp#, m.LAST\_NAME Manager, m.EMPLOYEE\_ID Mgr#

from EMPLOYEES e left join EMPLOYEES m on e.MANAGER\_ID = m.EMPLOYEE\_ID

order by e.EMPLOYEE\_ID;



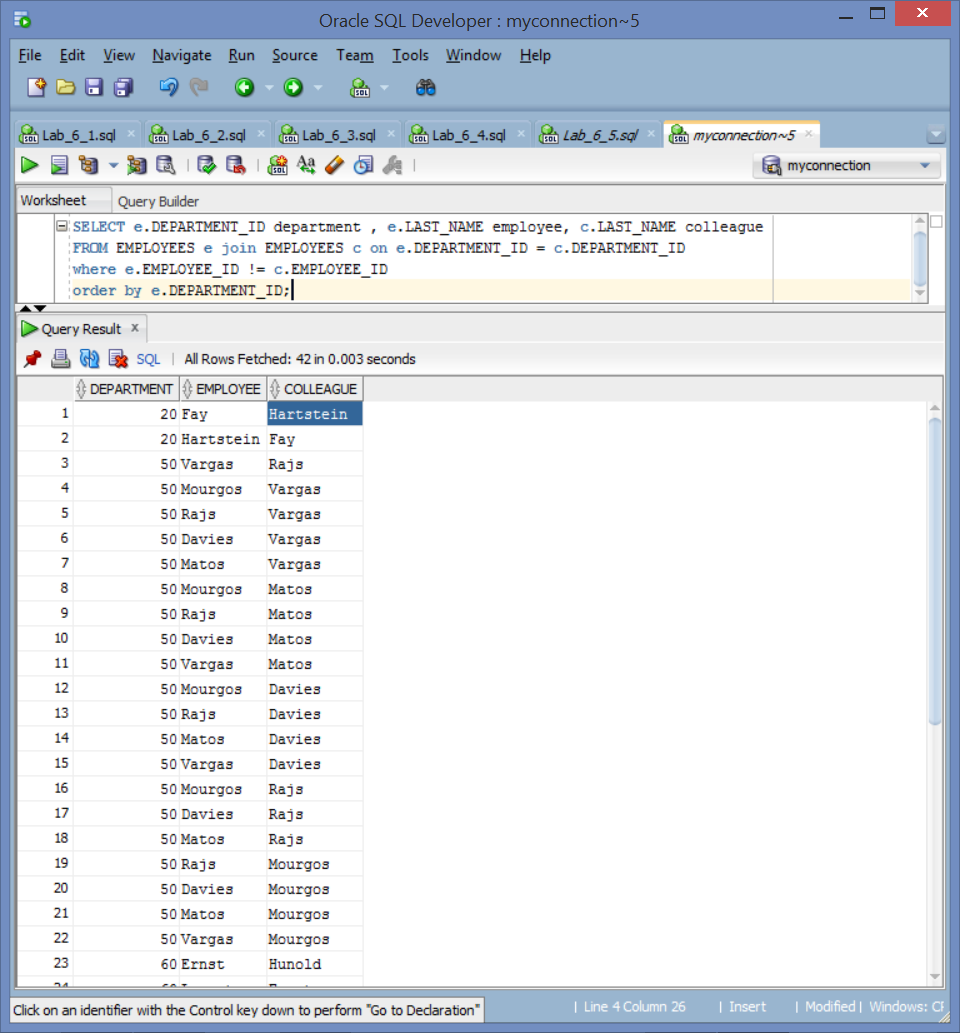
6) Create a report for the HR department that displays employee last names, department numbers, and all the employees who work in the same department as a given employee. Give each column an appropriate label. Save the script to a file named lab\_06\_06.sql.

SELECT e.DEPARTMENT\_ID department , e.LAST\_NAME employee, c.LAST\_NAME colleague

FROM EMPLOYEES e join EMPLOYEES c on e.DEPARTMENT\_ID = c.DEPARTMENT\_ID

where e.EMPLOYEE\_ID != c.EMPLOYEE\_ID

order by e.DEPARTMENT\_ID;



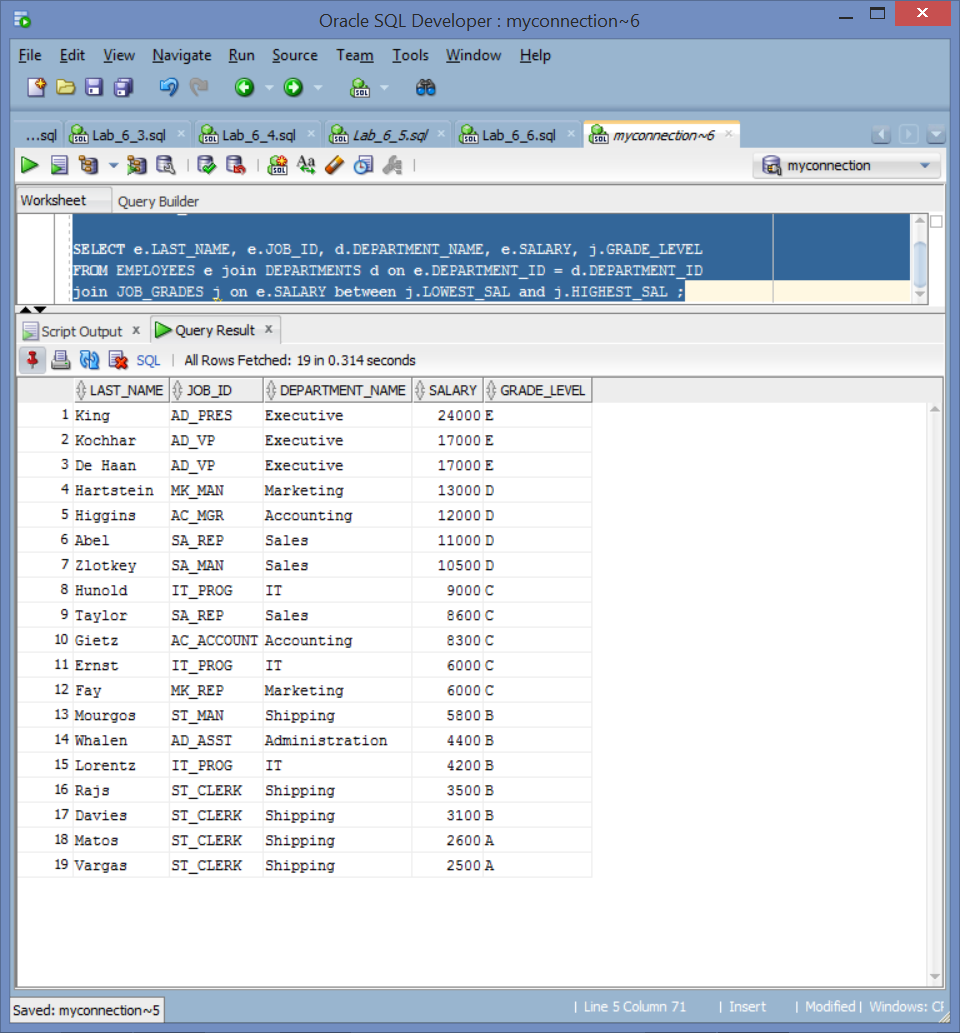
7) The HR department needs a report on job grades and salaries. To familiarize yourself with the JOB\_GRADES table, first show the structure of the JOB\_GRADES table. Then create a query that displays the name, job, department name, salary, and grade for all employees.

DESCRIBE JOB\_GRADES;

SELECT e.LAST\_NAME, e.JOB\_ID, d.DEPARTMENT\_NAME, e.SALARY, j.GRADE\_LEVEL

FROM EMPLOYEES e join DEPARTMENTS d on e.DEPARTMENT\_ID = d.DEPARTMENT\_ID

join JOB\_GRADES j on e.SALARY between j.LOWEST\_SAL and j.HIGHEST\_SAL ;

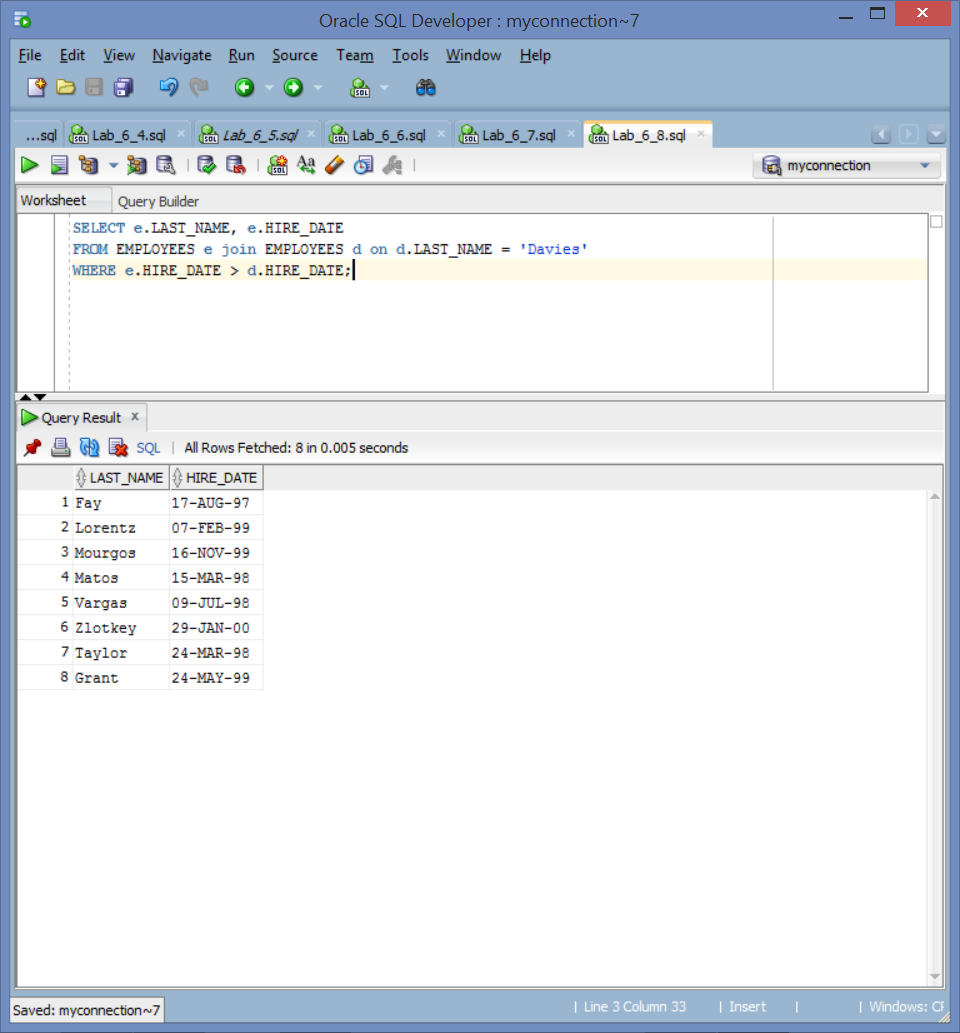


8) The HR department wants to determine the names of all the employees who were hired after Davies. Create a query to display the name and hire date of any employee hired after employee Davies.

SELECT e.LAST\_NAME, e.HIRE\_DATE

FROM EMPLOYEES e join EMPLOYEES d on d.LAST\_NAME = 'Davies'

WHERE e.HIRE\_DATE > d.HIRE\_DATE;

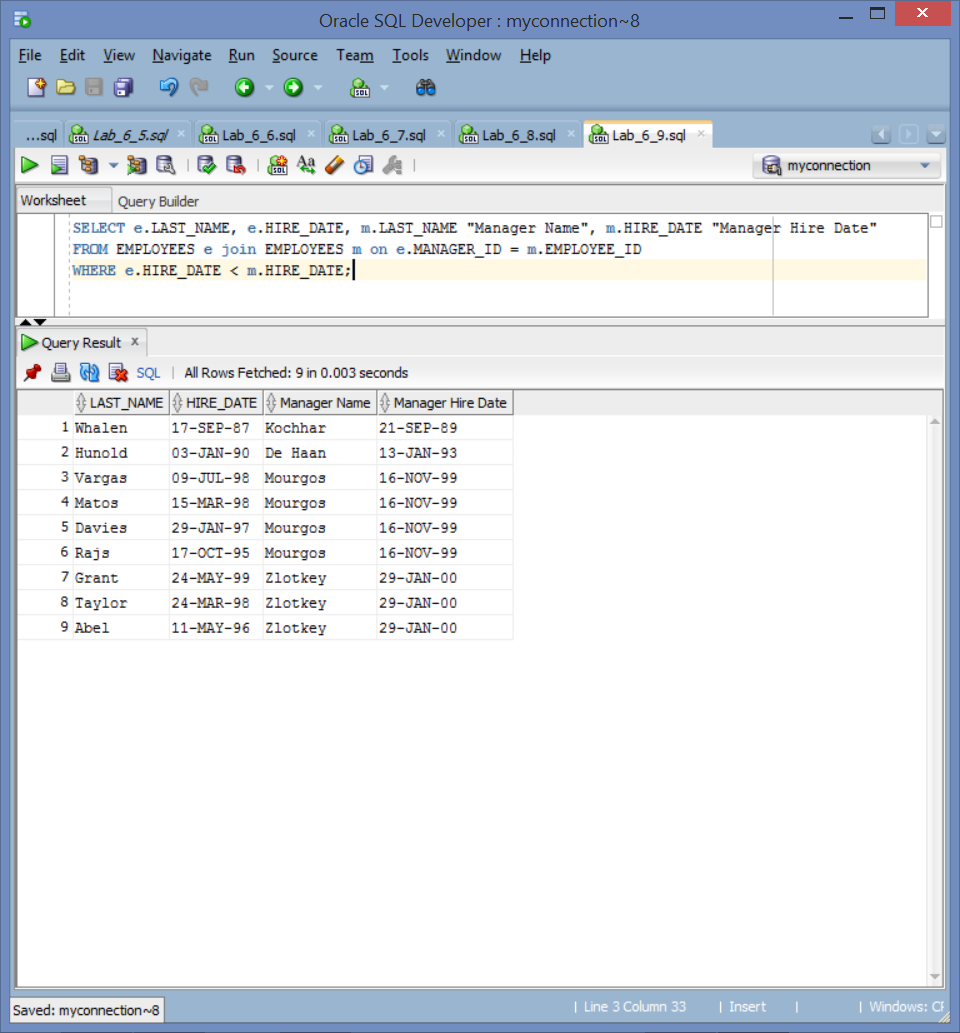


9) The HR department needs to find the names and hire dates of all the employees who were hired before their managers, along with their managers’ names and hire dates. Save the script to a file named lab\_06\_09.sql.

SELECT e.LAST\_NAME, e.HIRE\_DATE, m.LAST\_NAME "Manager Name", m.HIRE\_DATE "Manager Hire Date"

FROM EMPLOYEES e join EMPLOYEES m on e.MANAGER\_ID = m.EMPLOYEE\_ID

WHERE e.HIRE\_DATE < m.HIRE\_DATE;



10) Compare and contrast the primary joins compliant with the SQL 1999 standard.

INNER returns the records that match in both tables and does not return any records that are not linked between tables (the linking field is null in at least one table)

LEFT returns the full first table with linked columns from second table, if the second table lacks data null data is inserted. Data from the second table that cannot be liked is not displayed.

RIGHT is similar to left except inverse. Swapping the order of tables is equivalent to swapping left and right joins

FULL return all records from both tables linking data were possible and insering null where no link is made.

LEFT RIGHT and FULL are outer joins, because they join data outside of the tables’ intersection.

11) Using justification of a viable business report, create an equijoin, outer join, and natural join between at least four unique Oracle tables that would be valuable to executives to assess the performance of a company using the tables from the curriculum. At least two (2) of the queries should provide a unique calculation such as a sum or average. Provide three (3) screenshots that depict each join.

SELECT e.LAST\_NAME, d.DEPARTMENT\_NAME, j.JOB\_TITLE,

Round((CURRENT\_DATE - e.HIRE\_DATE)/365)-15 experience, jg.GRADE\_LEVEL

FROM EMPLOYEES e join DEPARTMENTS d on e.DEPARTMENT\_ID = d.DEPARTMENT\_ID

join JOBS j on e.JOB\_ID = j.JOB\_ID

join JOB\_GRADES jg on e.SALARY between jg.LOWEST\_SAL and jg.HIGHEST\_SAL

order by d.DEPARTMENT\_ID;

