1) The HR department needs a list of department IDs for departments that do not contain the job ID ST\_CLERK. Use the set operators to create this report.

**SELECT department\_id**

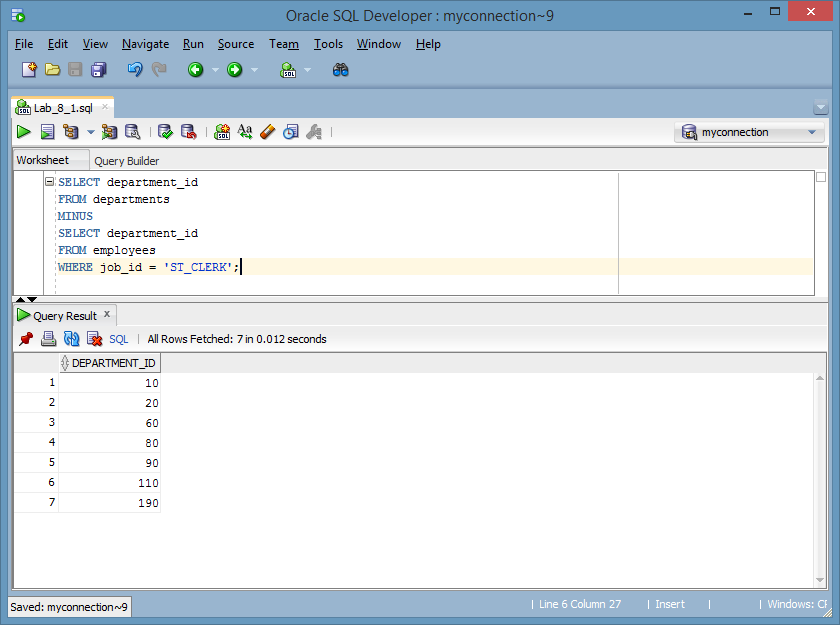
**FROM departments**

**MINUS**

**SELECT department\_id**

**FROM employees**

**WHERE job\_id = 'ST\_CLERK';**



2) The HR department needs a list of countries that have no departments located in them. Display the country ID and the name of the countries. Use the set operators to create this report.

**SELECT COUNTRY\_ID, COUNTRY\_NAME**

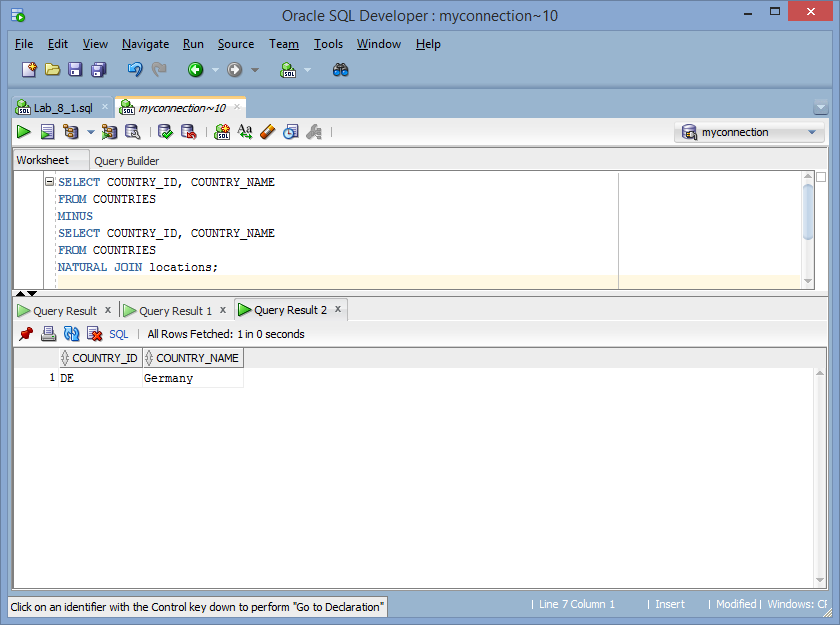
**FROM COUNTRIES**

**MINUS**

**SELECT COUNTRY\_ID, COUNTRY\_NAME**

**FROM COUNTRIES**

**NATURAL JOIN locations;**



3) Produce a list of jobs for departments 10, 50, and 20, in that order. Display the job ID and department ID by using the set operators.

**SELECT \* FROM**

**(SELECT job\_id, department\_id**

**FROM employees**

**minus**

**SELECT job\_id, department\_id**

**FROM employees**

**WHERE department\_id not in (20,50,10) or department\_id is null)**

**ORDER BY**

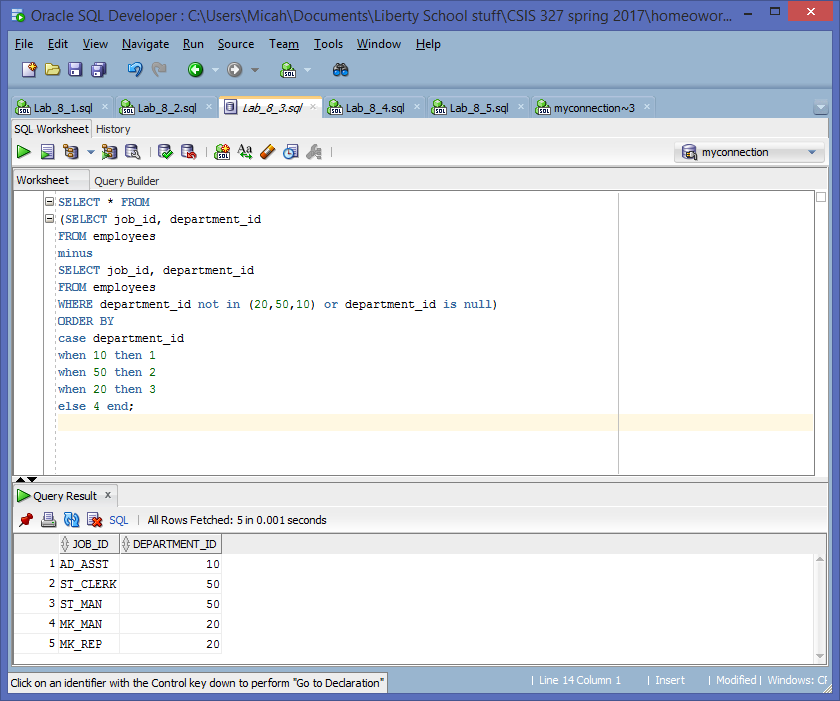
**case department\_id**

**when 10 then 1**

**when 50 then 2**

**when 20 then 3**

**else 4 end;**



4) Create a report that lists the employee IDs and job IDs of those employees who currently have a job title that is the same as their job title when they were initially hired by the company (that is, they changed jobs, but have now gone back to doing their original job).

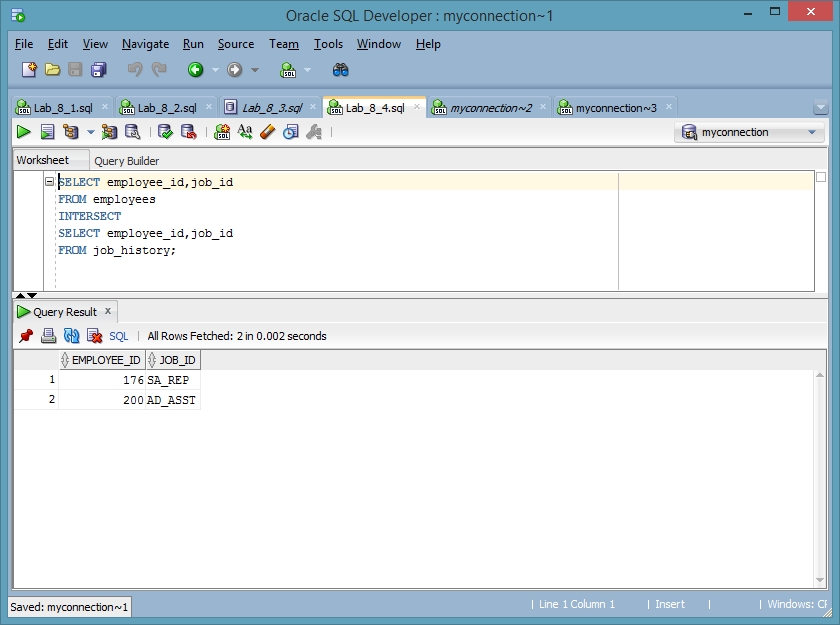
**SELECT employee\_id,job\_id**

**FROM employees**

**INTERSECT**

**SELECT employee\_id,job\_id**

**FROM job\_history;**



5) The HR department needs a report with the following specifications:

• Last name and department ID of all employees from the EMPLOYEES table, regardless of whether or not they belong to a department

• Department ID and department name of all departments from the DEPARTMENTS table, regardless of whether or not they have employees working in them. Write a compound query to accomplish this.

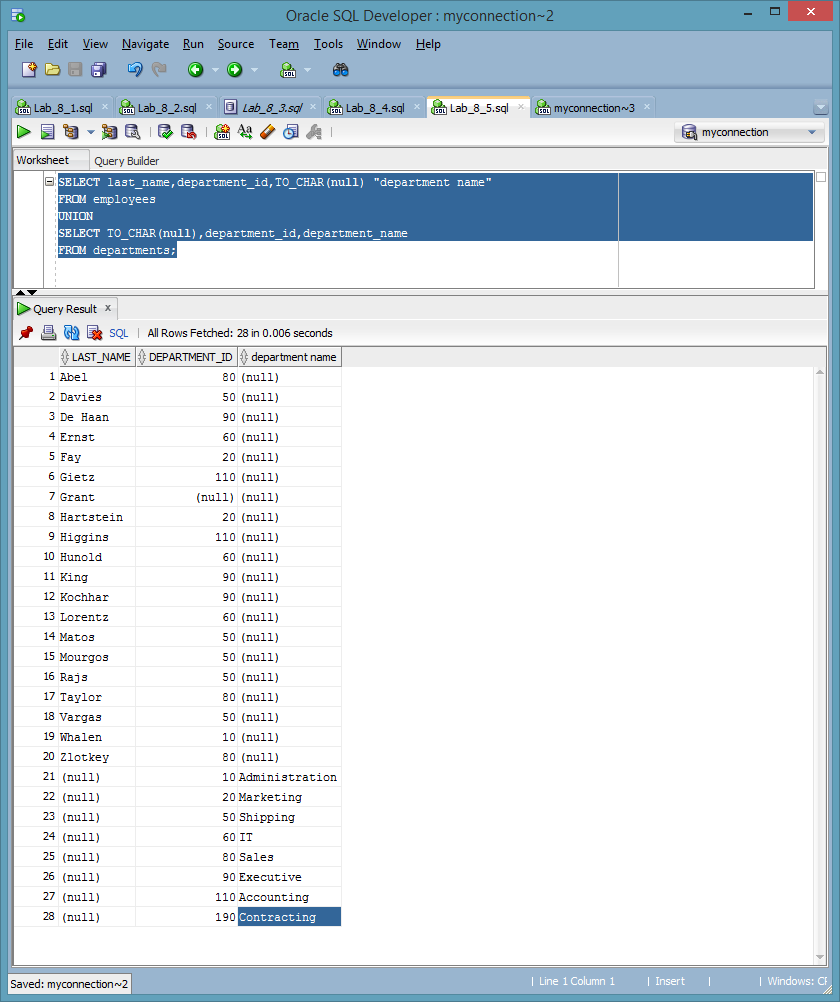
**SELECT last\_name,department\_id,TO\_CHAR(null) "department name"**

**FROM employees**

**UNION**

**SELECT TO\_CHAR(null),department\_id,department\_name**

**FROM departments;**



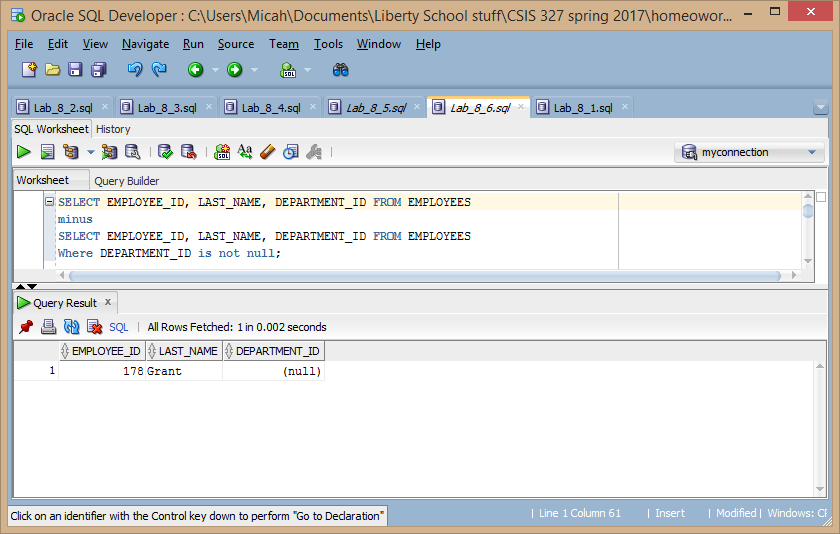
6) Using justification of a viable business report, create a union, minus, and intersect between at least five (5) unique Oracle tables that would be valuable to executives to assess the performance of a company using the tables from the curriculum. At least two (2) of the queries should include unique combinations such that there are useful differences between the combined tables. Provide three (3) screenshots that depict each set operator.

**SELECT EMPLOYEE\_ID, LAST\_NAME, DEPARTMENT\_ID FROM EMPLOYEES**

**minus**

**SELECT EMPLOYEE\_ID, LAST\_NAME, DEPARTMENT\_ID FROM EMPLOYEES**

**Where DEPARTMENT\_ID is not null;**

****

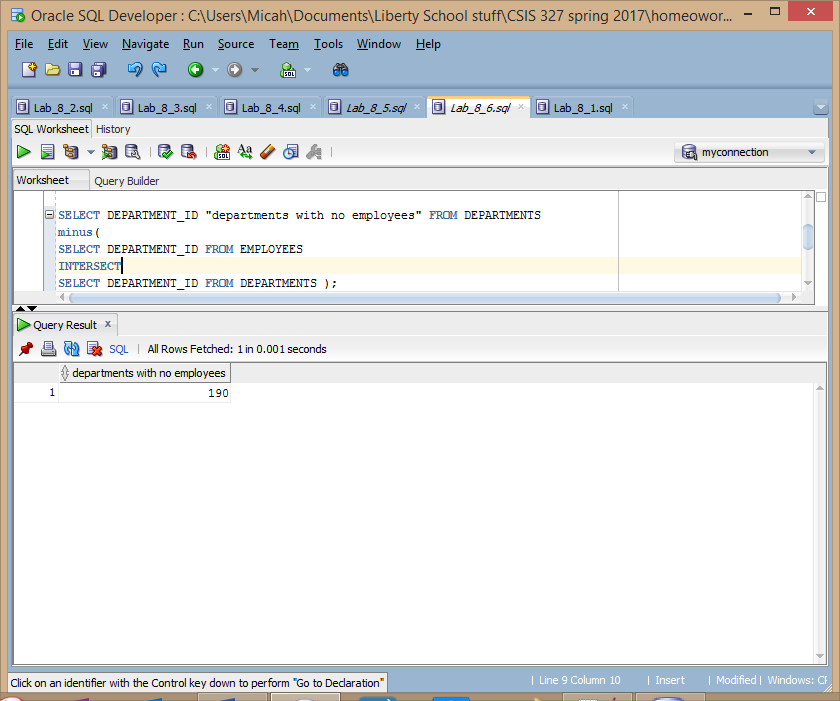
**SELECT DEPARTMENT\_ID "departments with no employees" FROM DEPARTMENTS**

**minus(**

**SELECT DEPARTMENT\_ID FROM EMPLOYEES**

**INTERSECT**

**SELECT DEPARTMENT\_ID FROM DEPARTMENTS );**

****

**SELECT EMPLOYEE\_ID, LAST\_NAME, TO\_DATE(null) "START\_DATE", TO\_DATE(null) "END\_DATE", JOB\_ID FROM EMPLOYEES**

**UNION**

**SELECT EMPLOYEE\_ID, TO\_CHAR(null), START\_DATE, END\_DATE, JOB\_ID FROM JOB\_HISTORY**

**ORDER BY EMPLOYEE\_ID;**

