Lab on Schema objects

Solve the following:

1. Create the EMPLOYEES2 table based on the structure of the EMPLOYEES table. Include only the EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, SALARY, and DEPARTMENT\_ID columns. Name the columns in your new table ID, FIRST\_NAME, LAST\_NAME, SALARY, and DEPT\_ID, respectively.

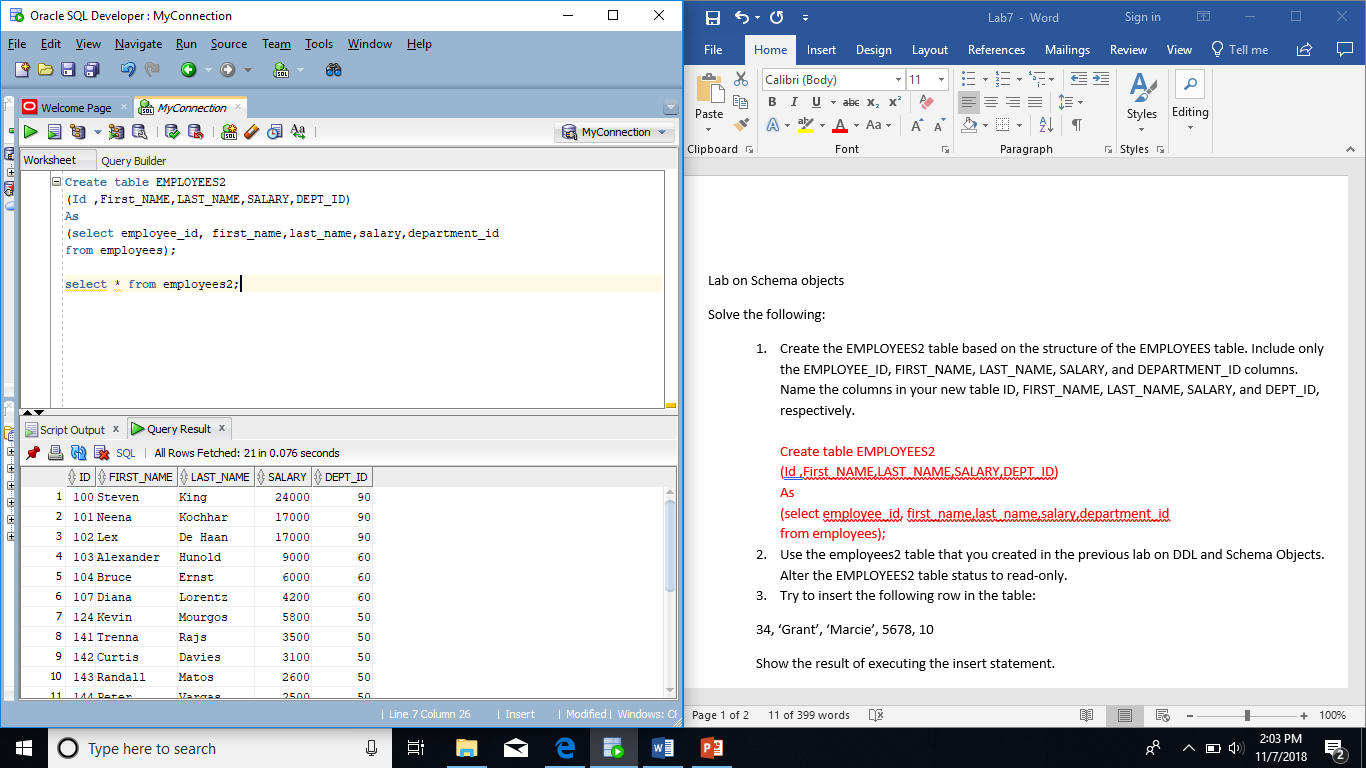
Create table EMPLOYEES2

(Id ,First\_NAME,LAST\_NAME,SALARY,DEPT\_ID)

As

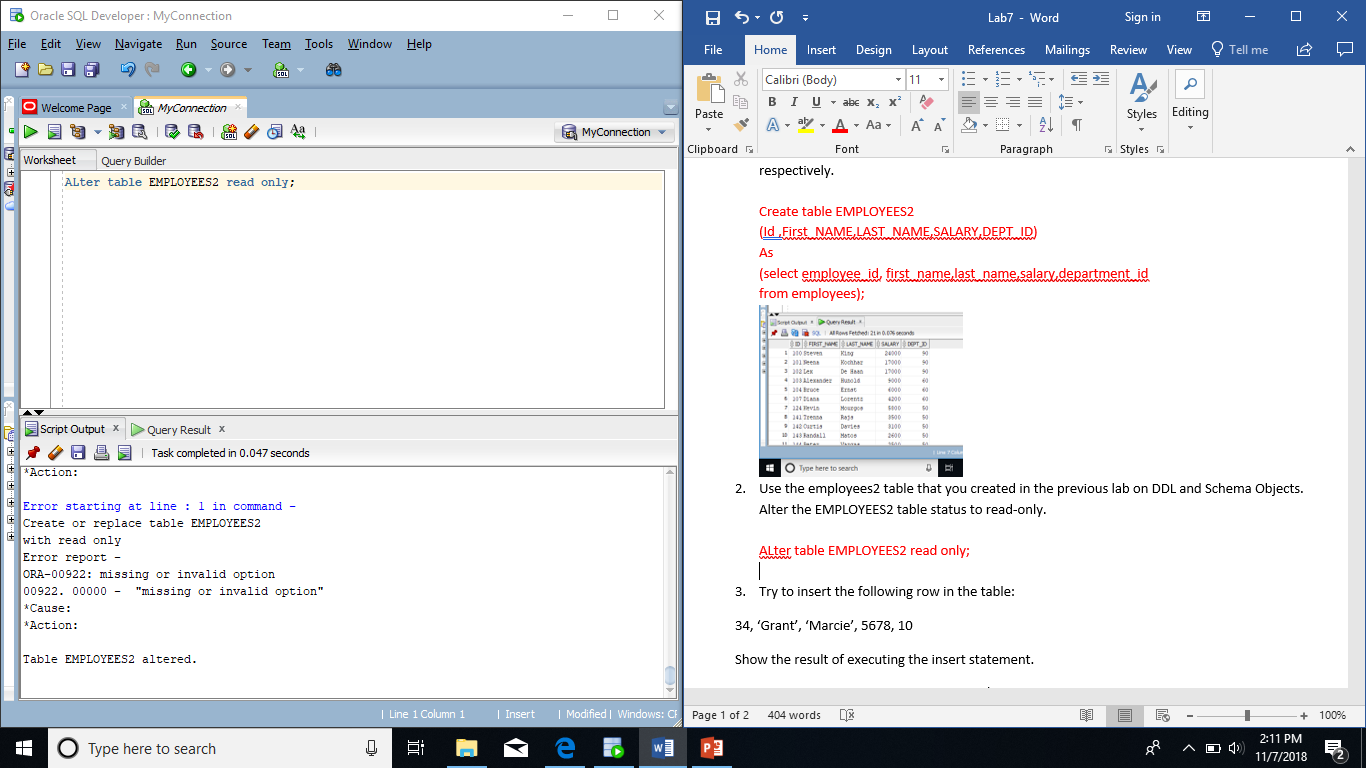
(select employee\_id, first\_name,last\_name,salary,department\_id

from employees);



1. Use the employees2 table that you created in the previous lab on DDL and Schema Objects. Alter the EMPLOYEES2 table status to read-only.

ALter table EMPLOYEES2 read only;



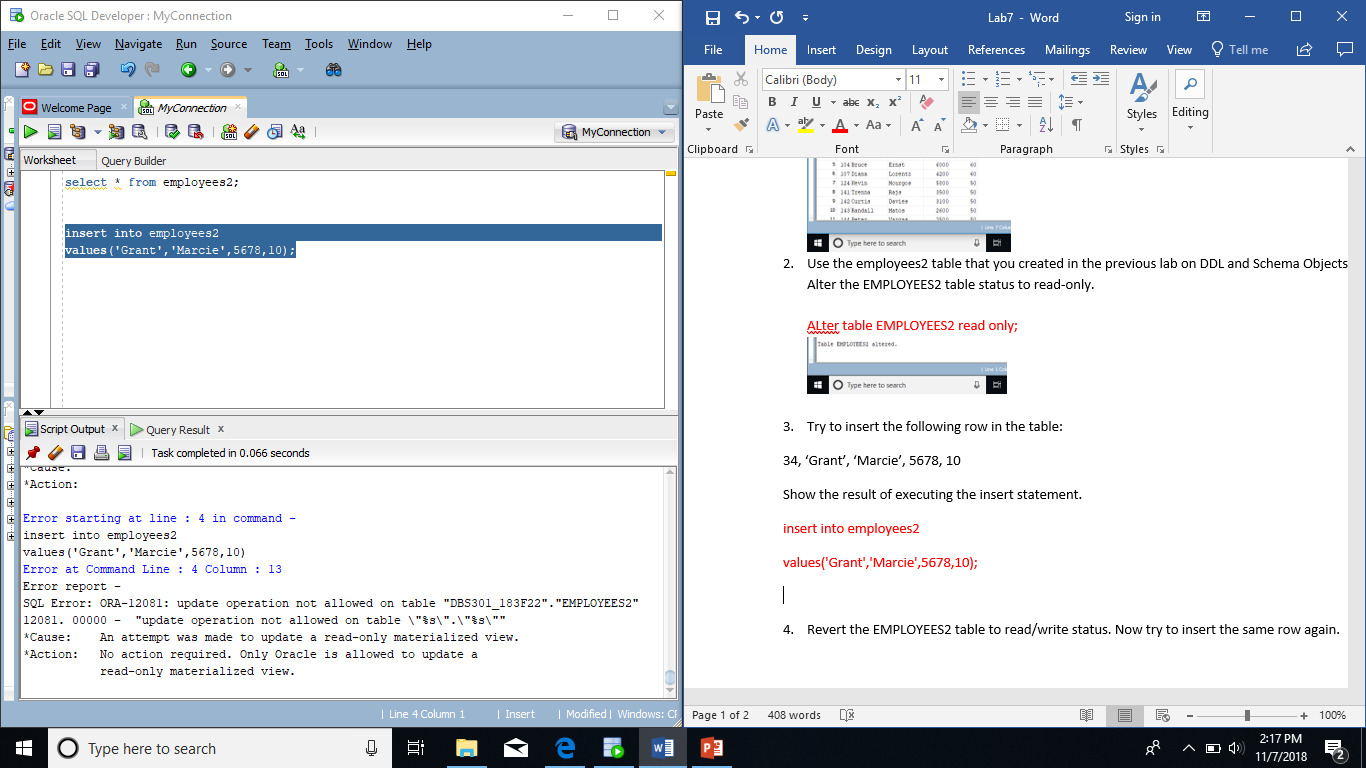
1. Try to insert the following row in the table:

34, ‘Grant’, ‘Marcie’, 5678, 10

Show the result of executing the insert statement.

insert into employees2

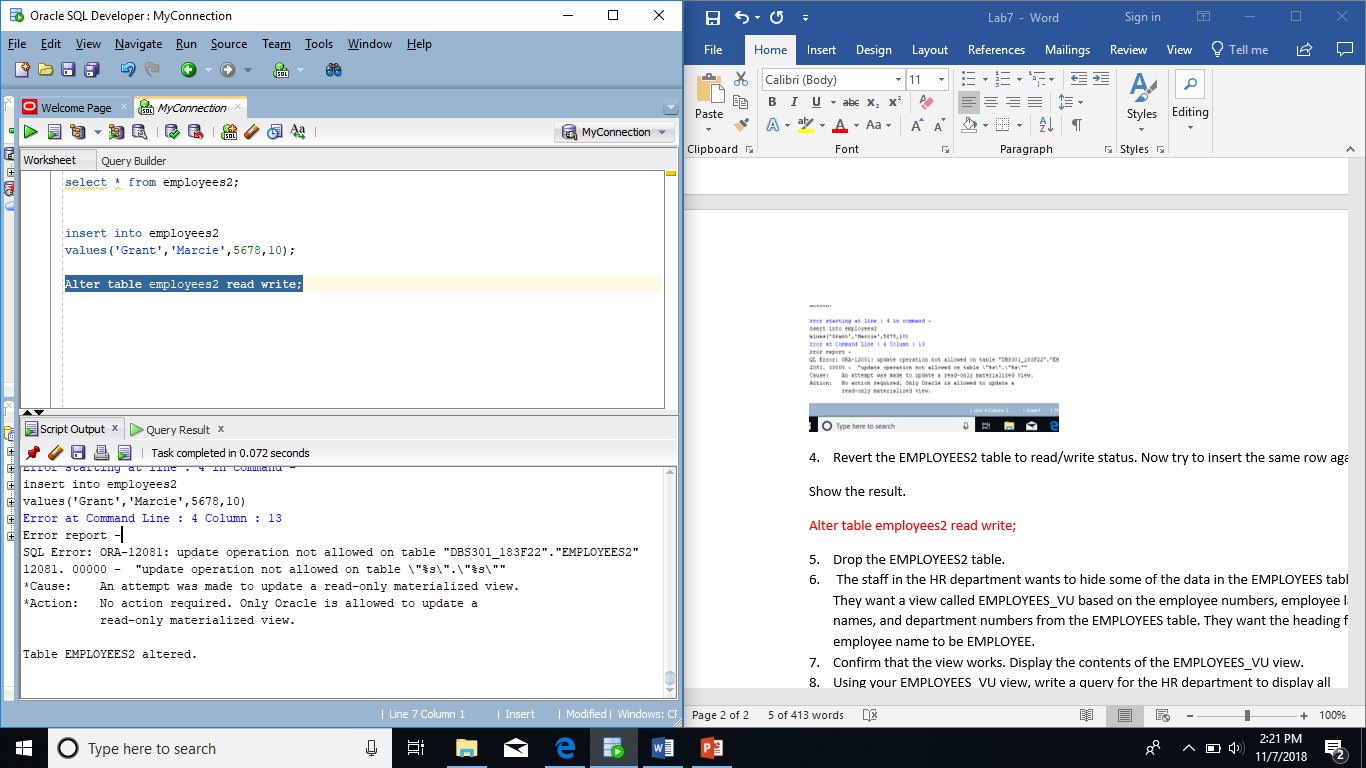
values(34,'Grant','Marcie',5678,10);



1. Revert the EMPLOYEES2 table to read/write status. Now try to insert the same row again.

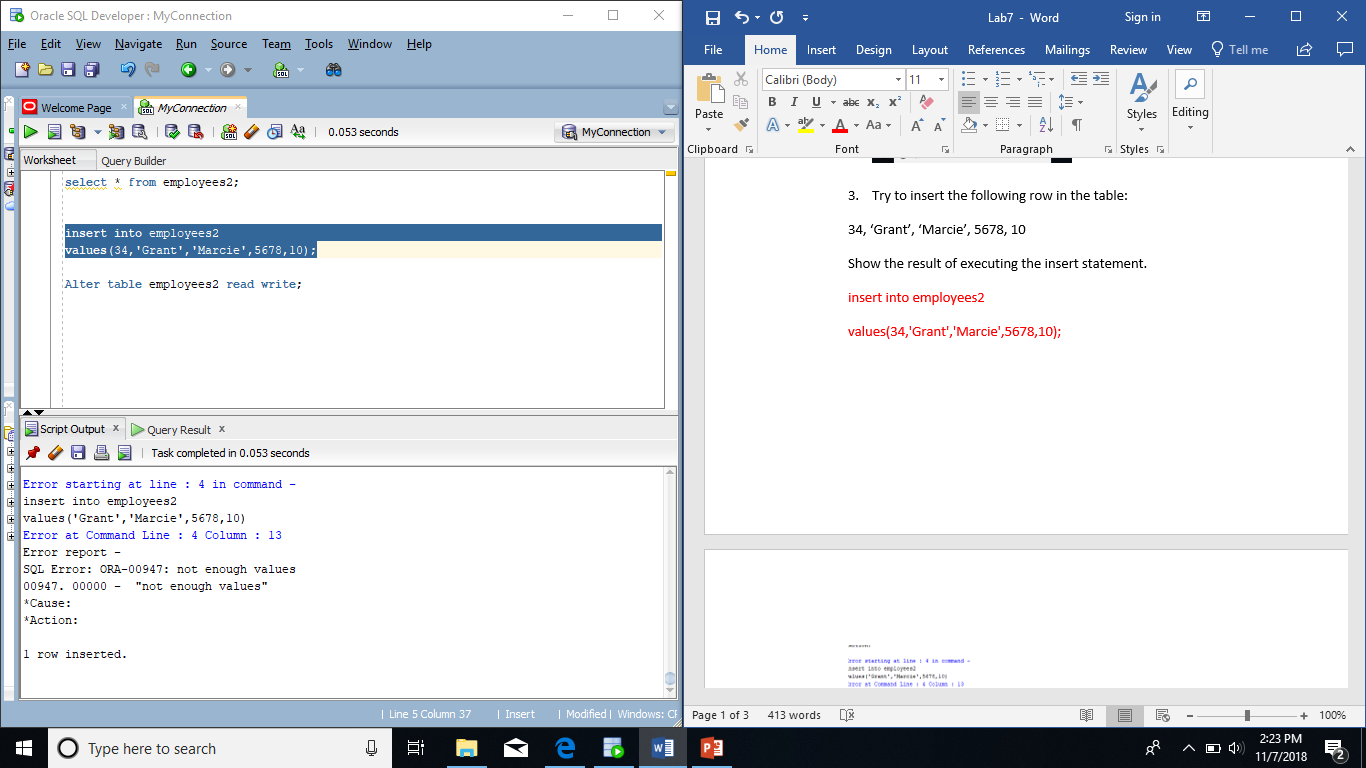
Show the result.

Alter table employees2 read write;



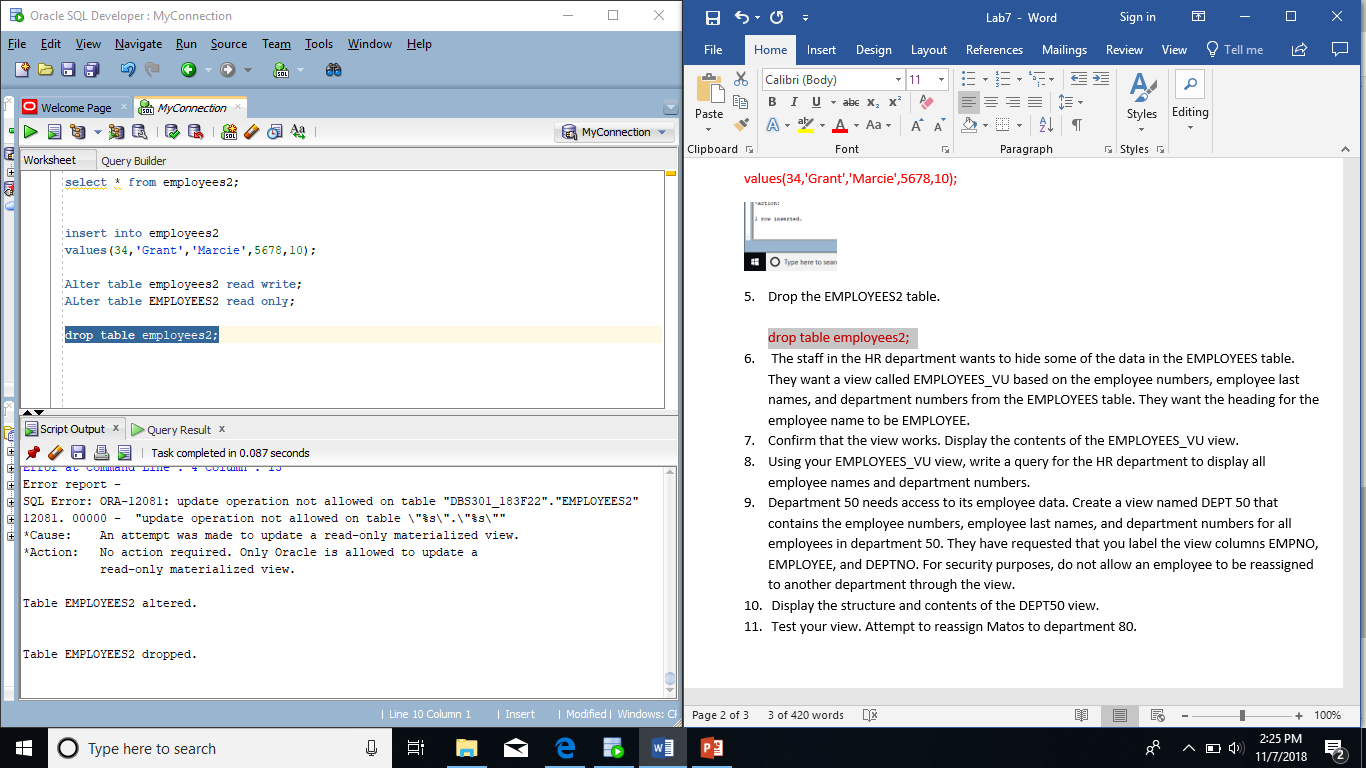
insert into employees2

values(34,'Grant','Marcie',5678,10);



1. Drop the EMPLOYEES2 table.

drop table employees2;

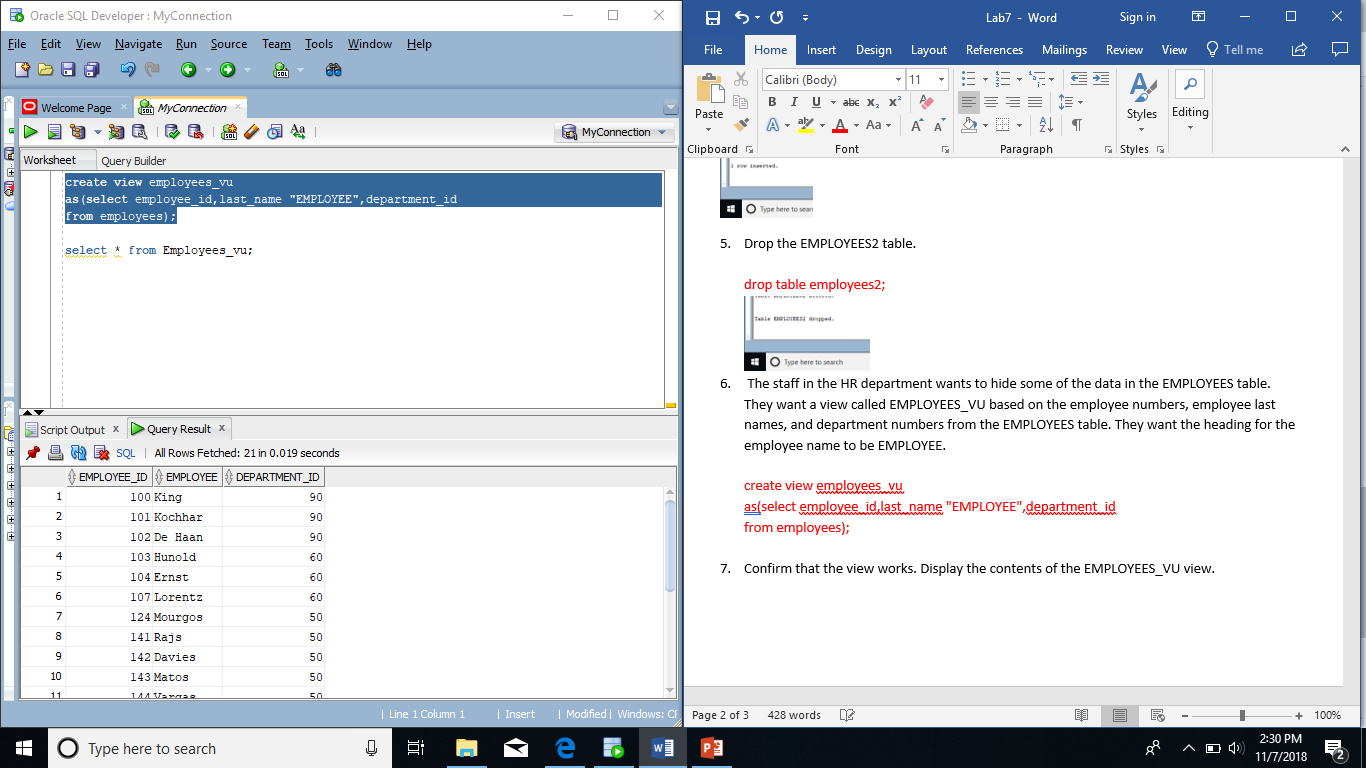


1. The staff in the HR department wants to hide some of the data in the EMPLOYEES table. They want a view called EMPLOYEES\_VU based on the employee numbers, employee last names, and department numbers from the EMPLOYEES table. They want the heading for the employee name to be EMPLOYEE.

create view employees\_vu

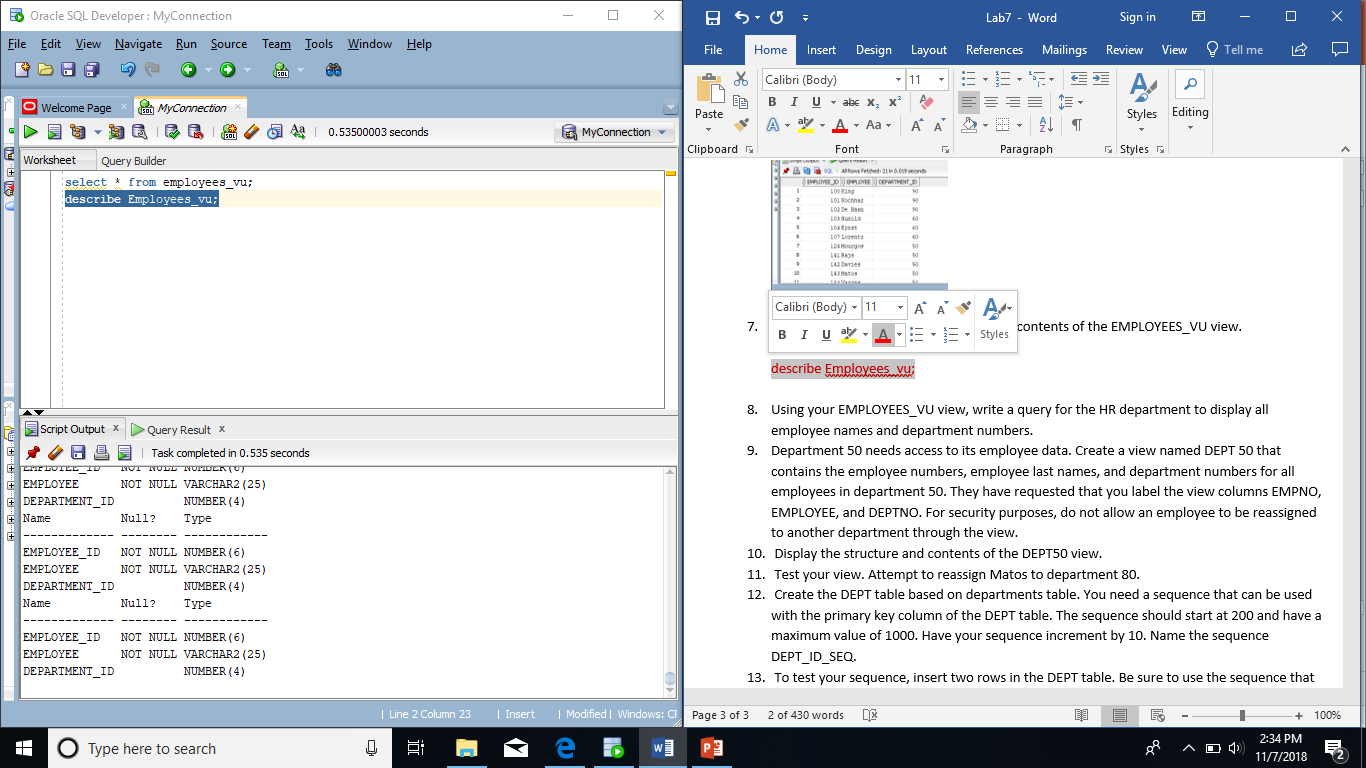
as(select employee\_id,last\_name "EMPLOYEE",department\_id

from employees);



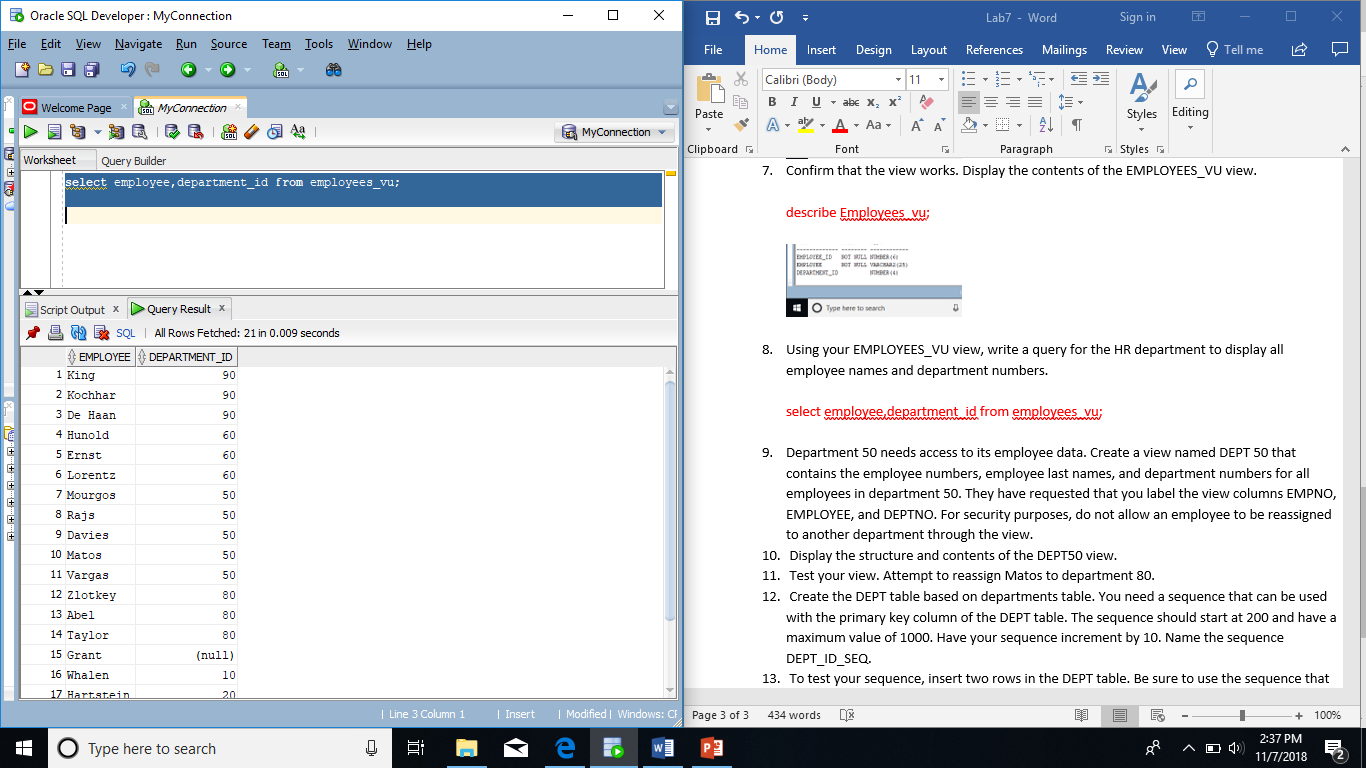
1. Confirm that the view works. Display the contents of the EMPLOYEES\_VU view.

describe Employees\_vu;



1. Using your EMPLOYEES\_VU view, write a query for the HR department to display all employee names and department numbers.

select employee,department\_id from employees\_vu;



1. Department 50 needs access to its employee data. Create a view named DEPT 50 that contains the employee numbers, employee last names, and department numbers for all employees in department 50. They have requested that you label the view columns EMPNO, EMPLOYEE, and DEPTNO. For security purposes, do not allow an employee to be reassigned to another department through the view.

create view dept50

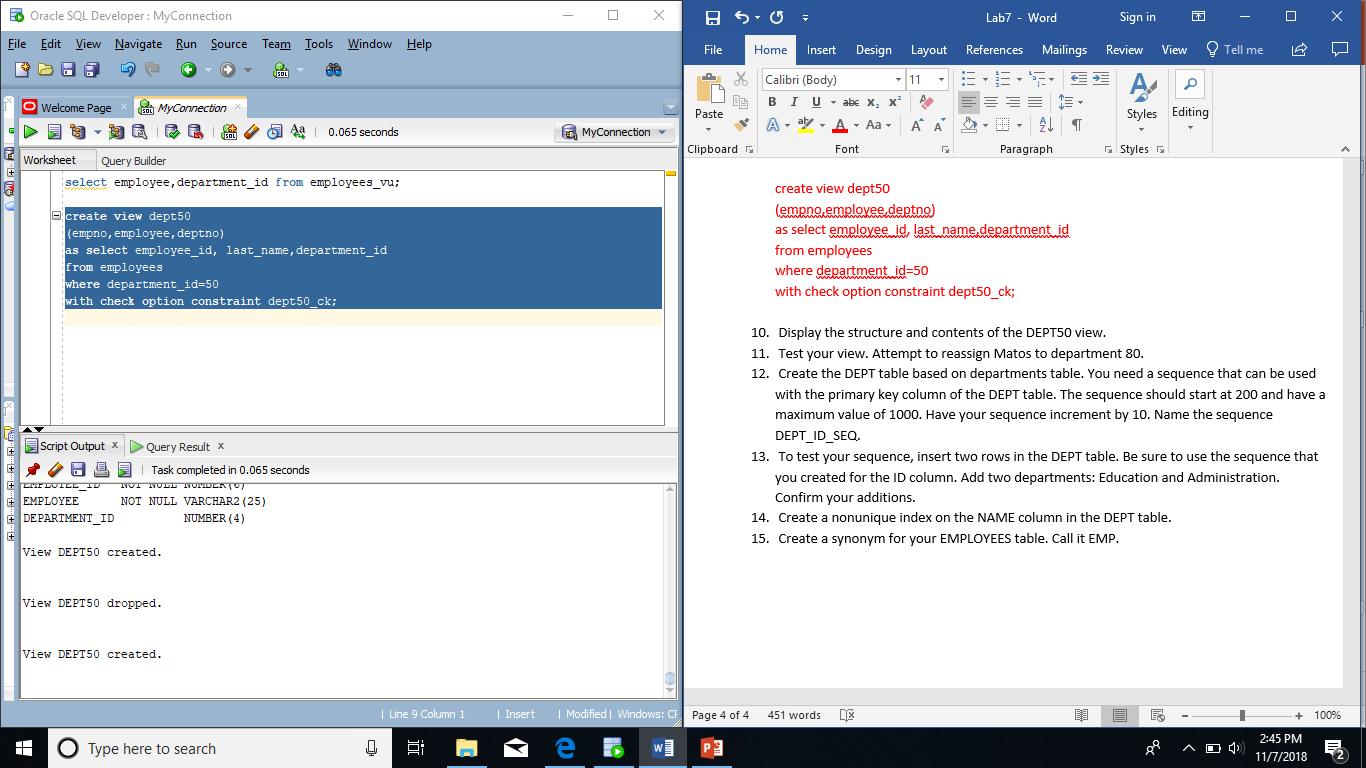
(empno,employee,deptno)

as select employee\_id, last\_name,department\_id

from employees

where department\_id=50

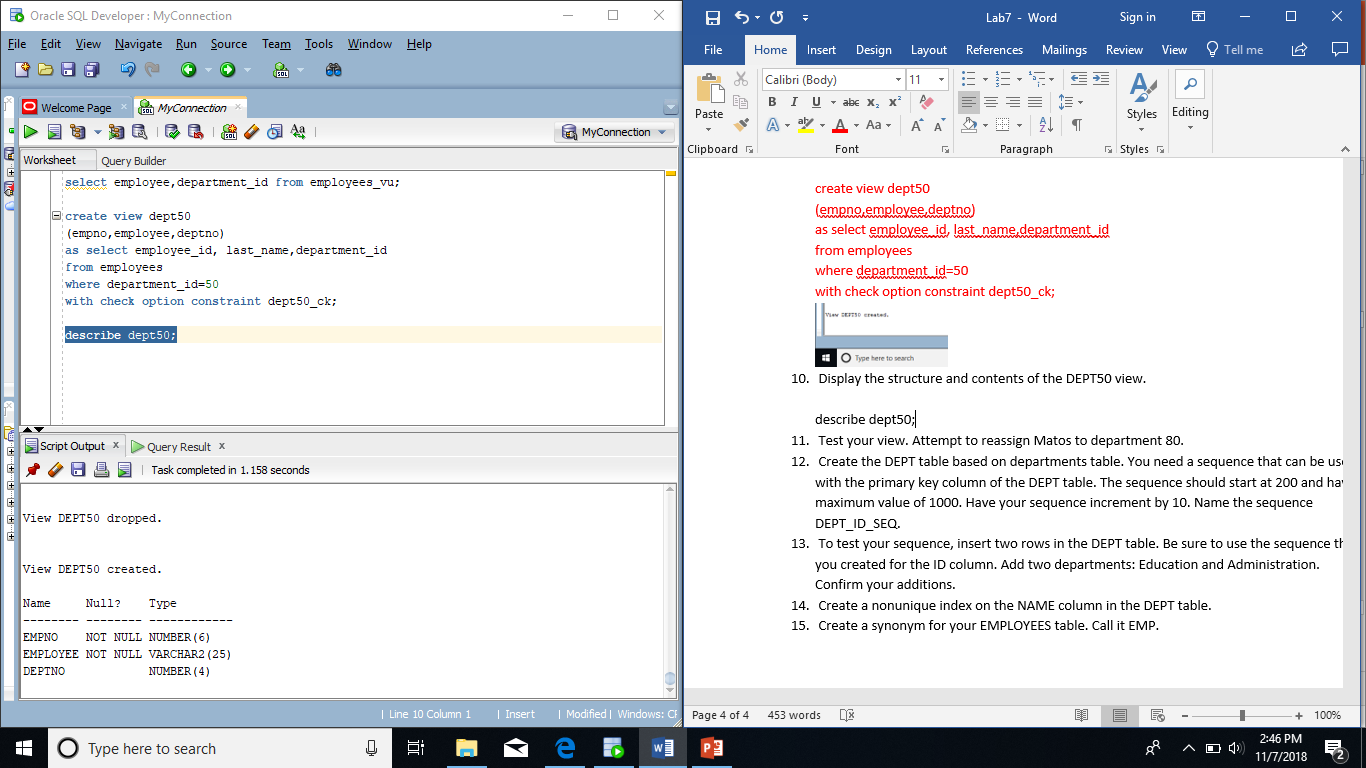
with check option constraint dept50\_ck;



??????????????

1. Display the structure and contents of the DEPT50 view.

describe dept50;

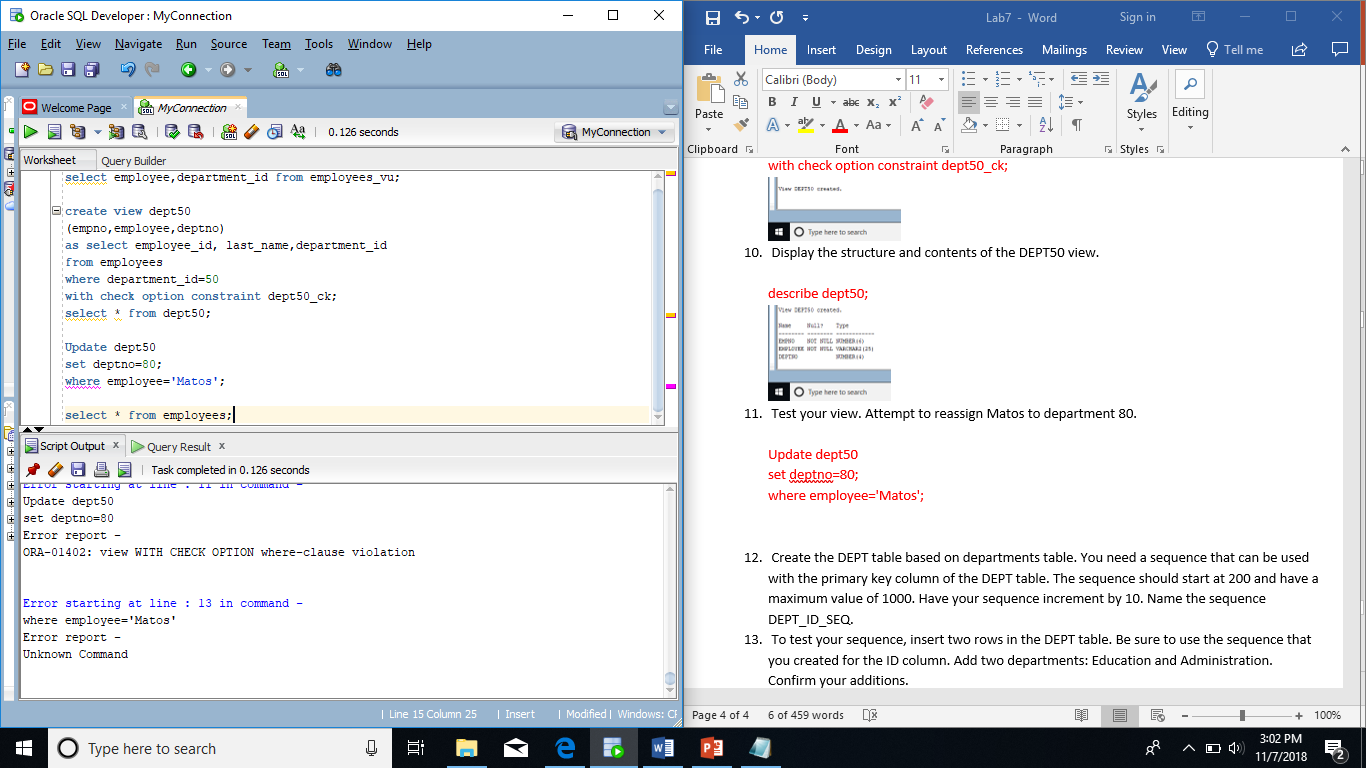


1. Test your view. Attempt to reassign Matos to department 80.

Update dept50

set deptno=80;

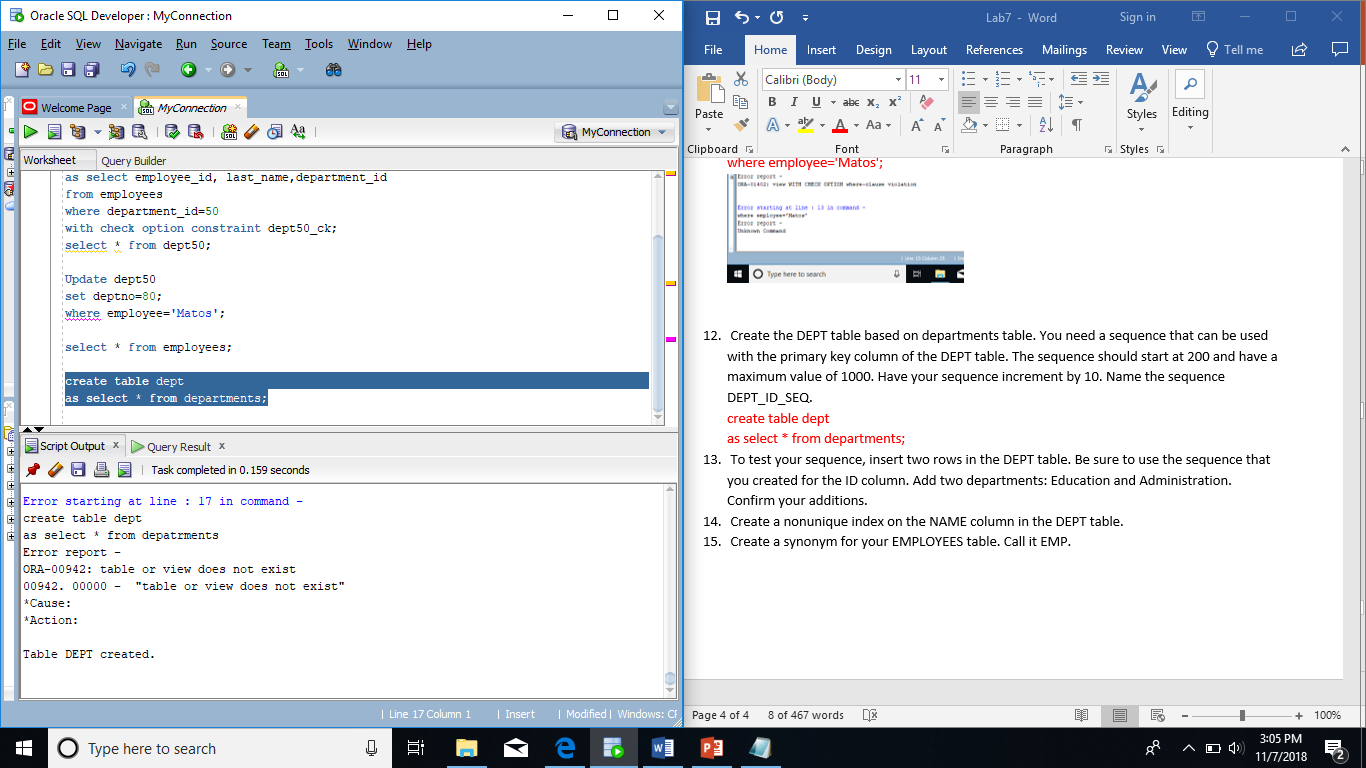
where employee='Matos';



1. Create the DEPT table based on departments table. You need a sequence that can be used with the primary key column of the DEPT table. The sequence should start at 200 and have a maximum value of 1000. Have your sequence increment by 10. Name the sequence DEPT\_ID\_SEQ.

create table dept

as select \* from departments;

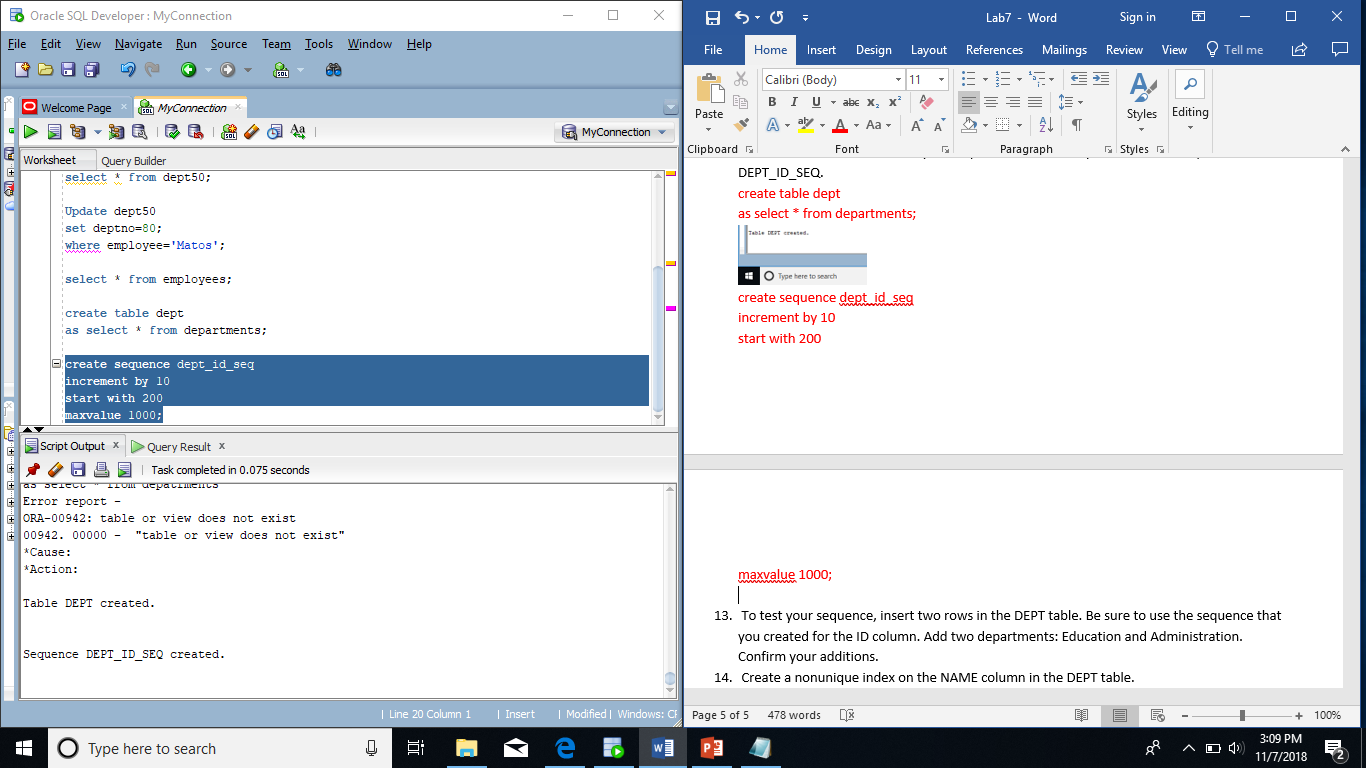


create sequence dept\_id\_seq

increment by 10

start with 200

maxvalue 1000;



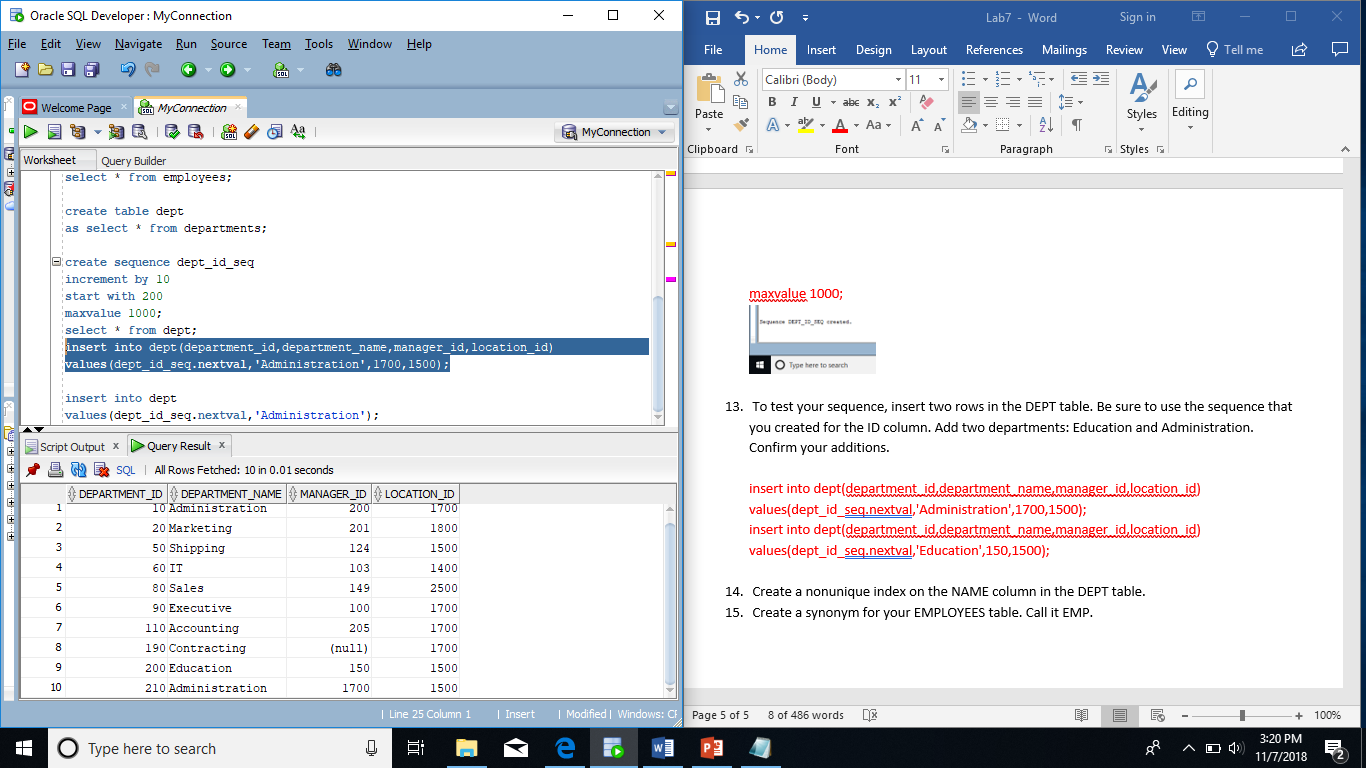
1. To test your sequence, insert two rows in the DEPT table. Be sure to use the sequence that you created for the ID column. Add two departments: Education and Administration. Confirm your additions.

insert into dept(department\_id,department\_name,manager\_id,location\_id)

values(dept\_id\_seq.nextval,'Administration',1700,1500);

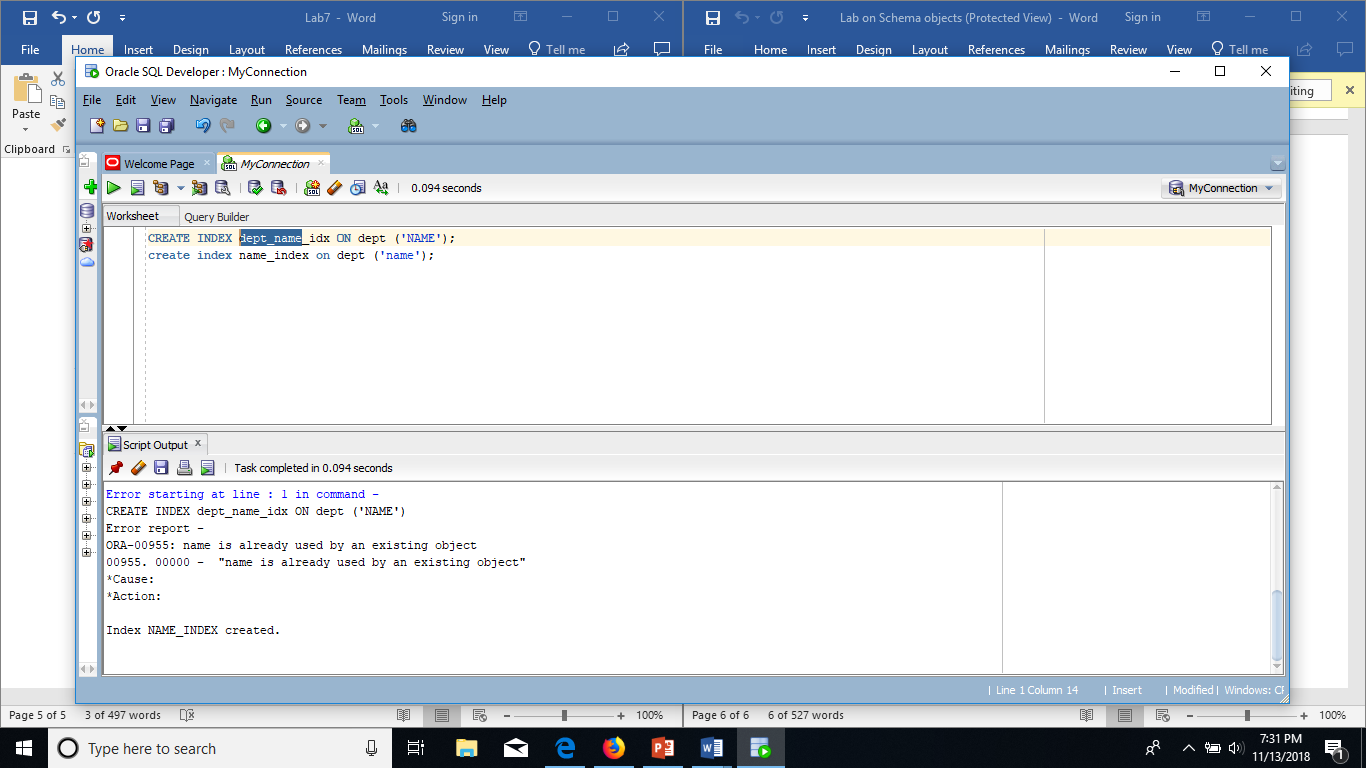
insert into dept(department\_id,department\_name,manager\_id,location\_id)

values(dept\_id\_seq.nextval,'Education',150,1500);



1. Create a nonunique index on the NAME column in the DEPT table.

create index name\_index on dept ('name');



1. Create a synonym for your EMPLOYEES table. Call it EMP.

CREATE SYNONYM EMP FOR EMPLOYEES;

