CO527 – Advanced Database Systems

Lab 05 – Database Security

Karunachandra R.H.I.O.

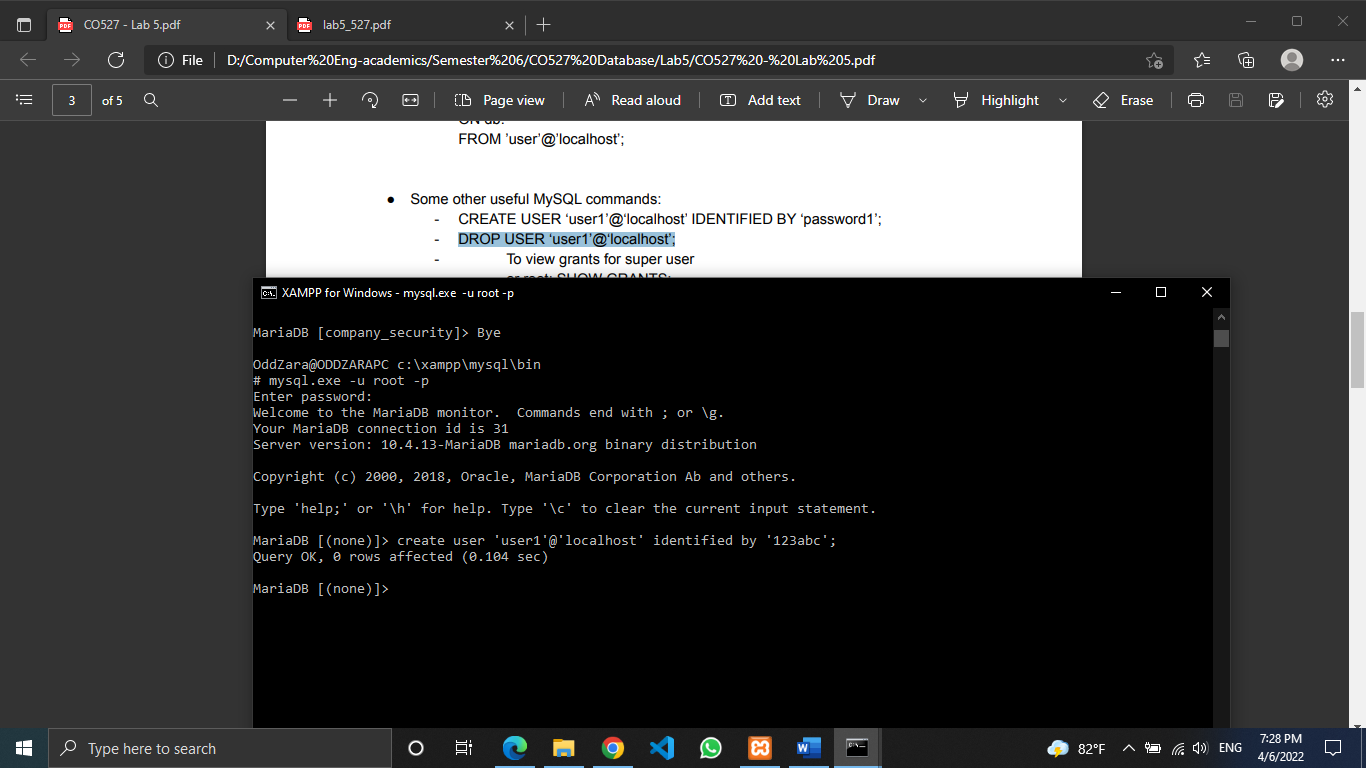
E/17/153

4.Exercise

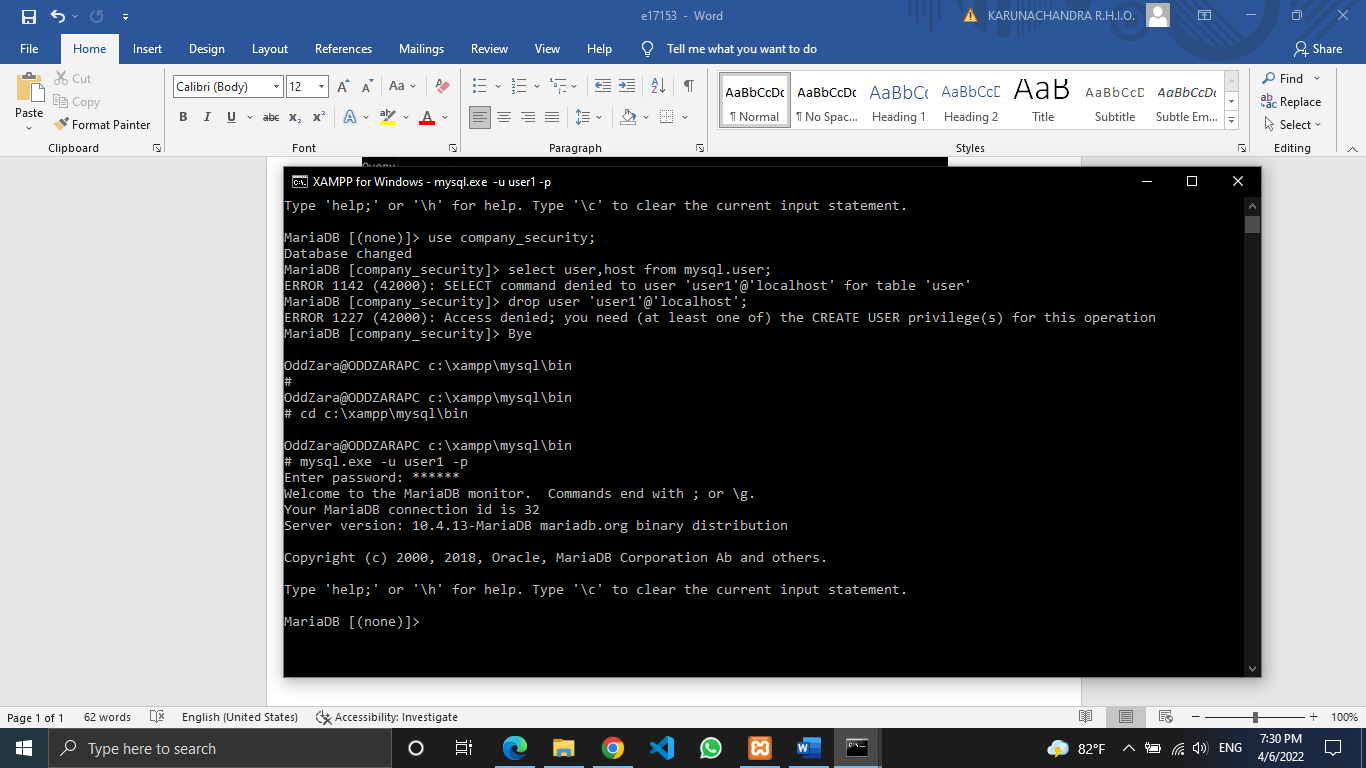
1.Create database company security

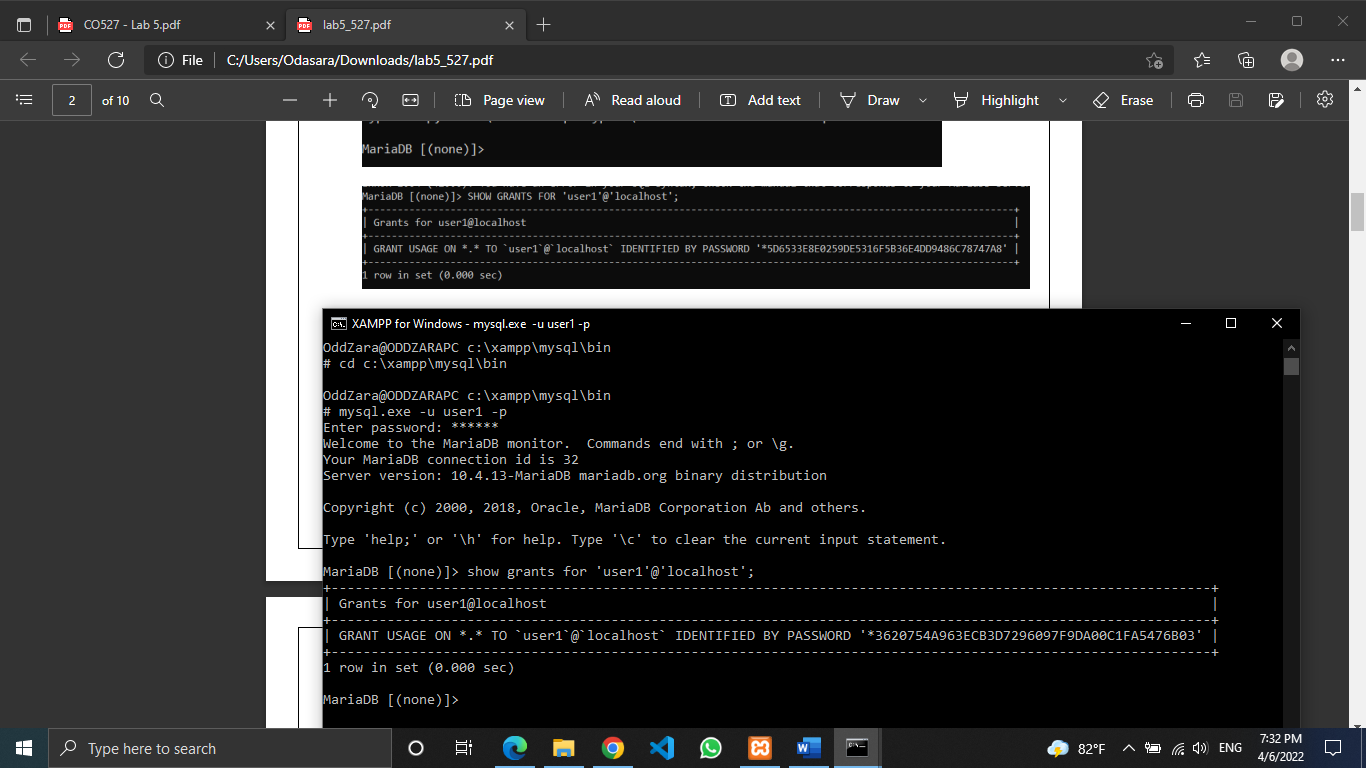
2.Load the given company security.sql file to the company security database.

3. Create a new user ‘user1’ within the MySQL shell.



4.Login to MySQL with a new user account and password and see if the new user has any authorities or privileges to the database

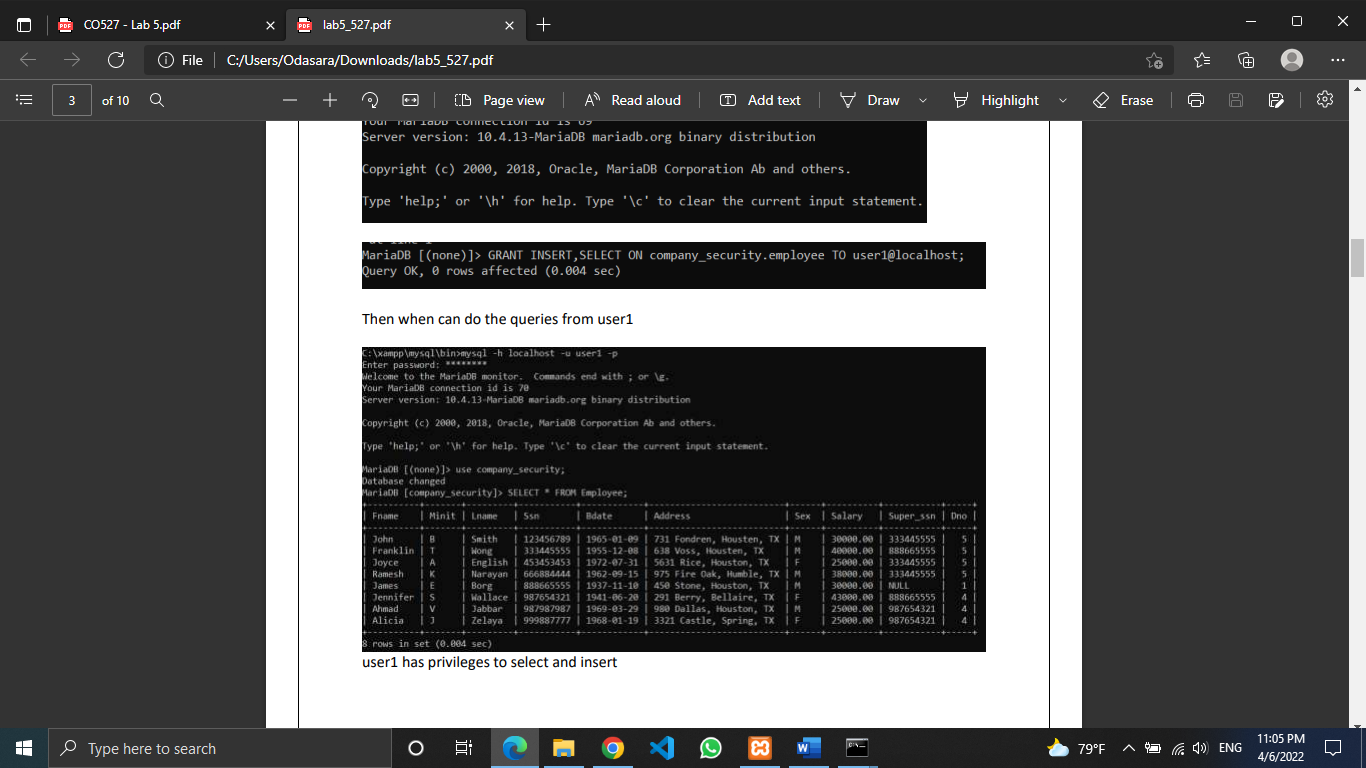




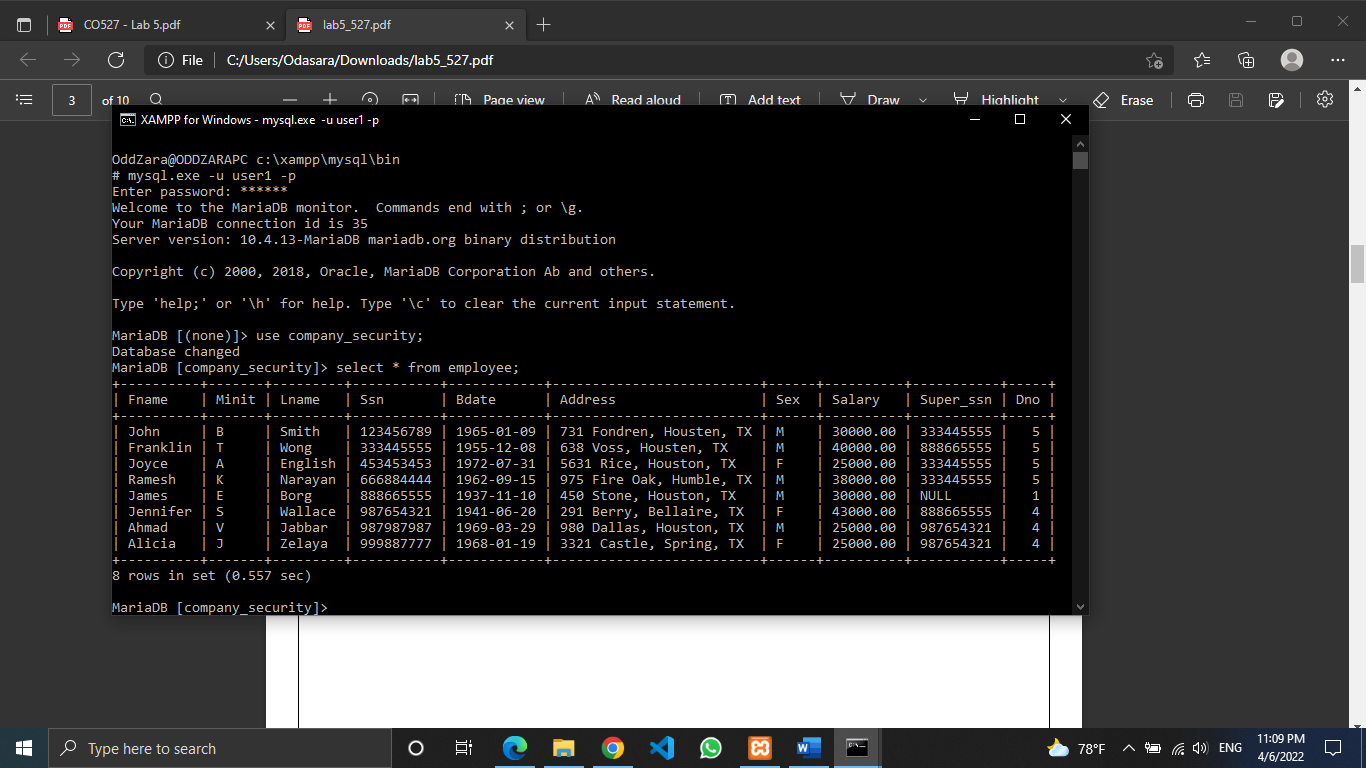
5.Make sure the new user has only read only permission to ‘Employee’ table.



6.Now allow ‘user1’ to query the followings: SELECT \* FROM Employee; INSERT into Employee(...)VALUES(...). What happens? Fix the problem.

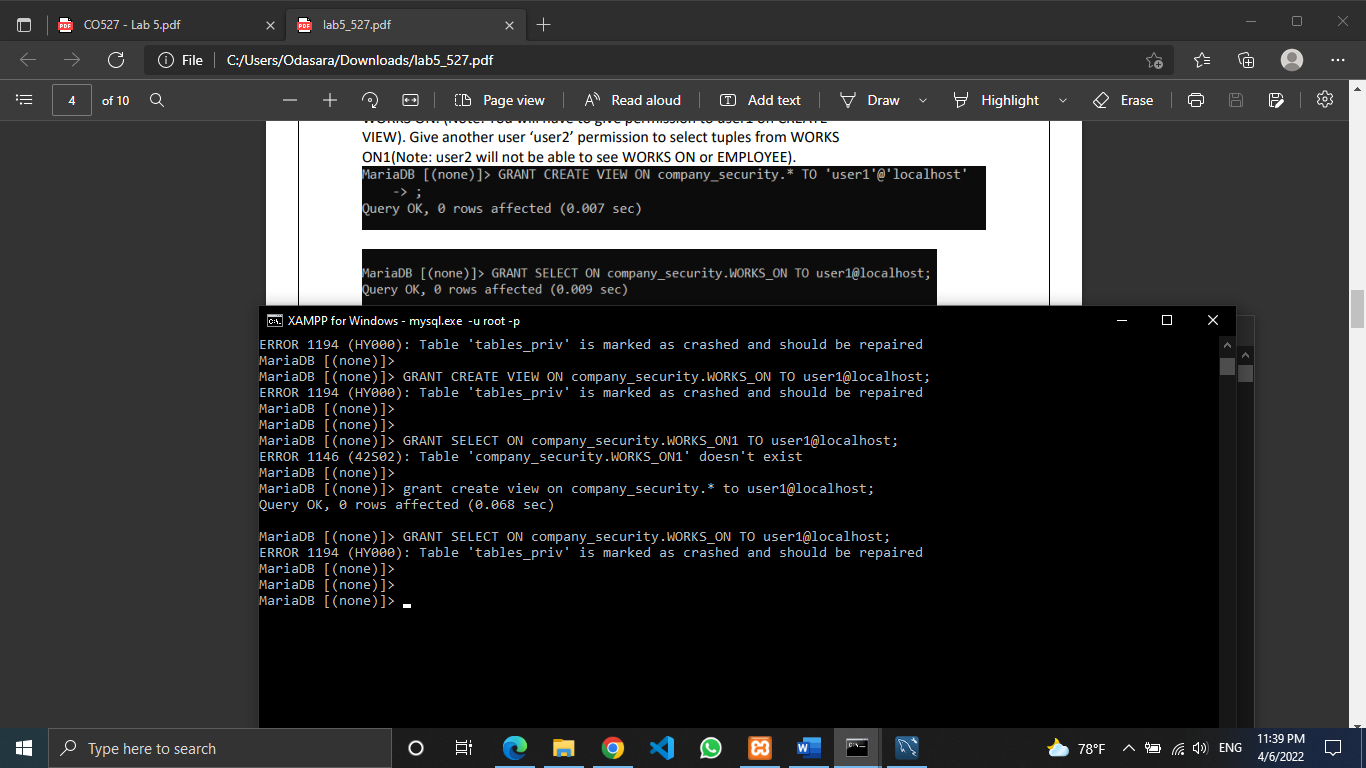


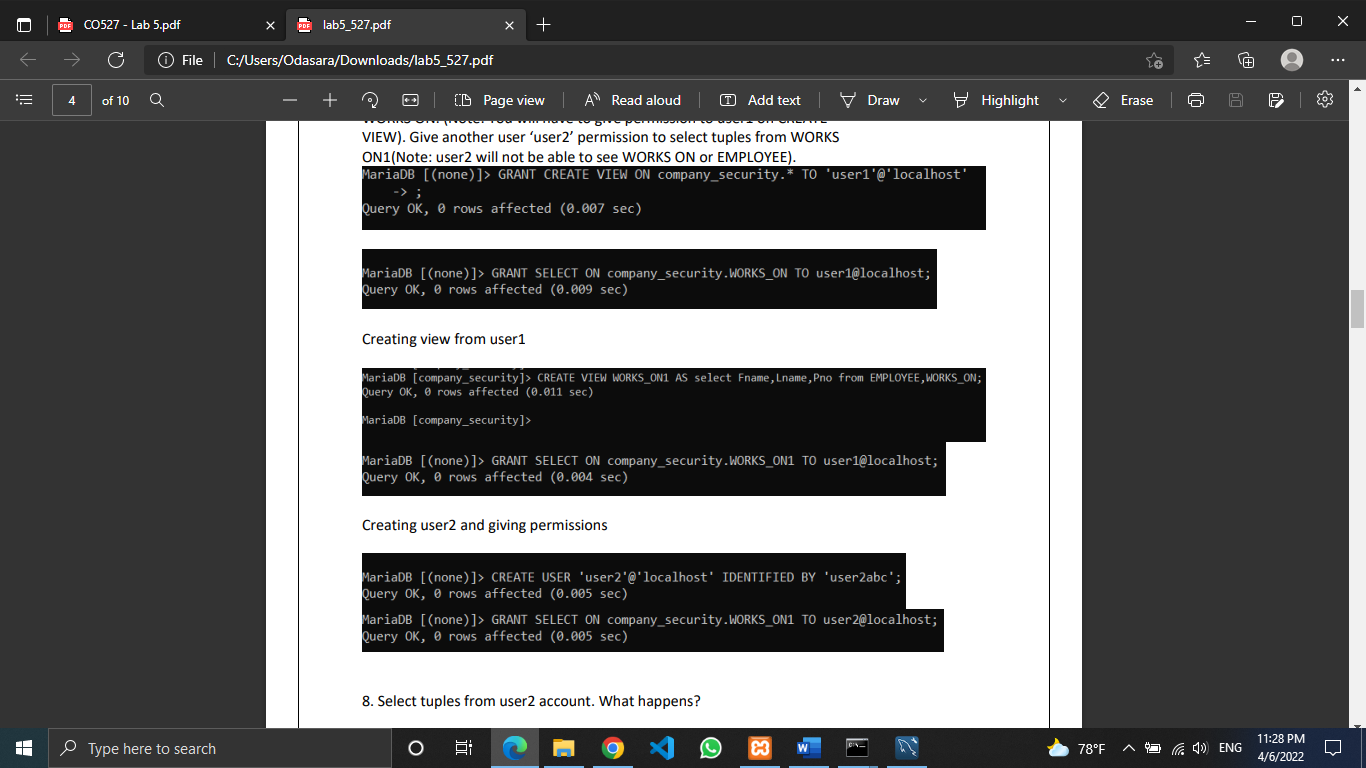
Now user1 can use the database.



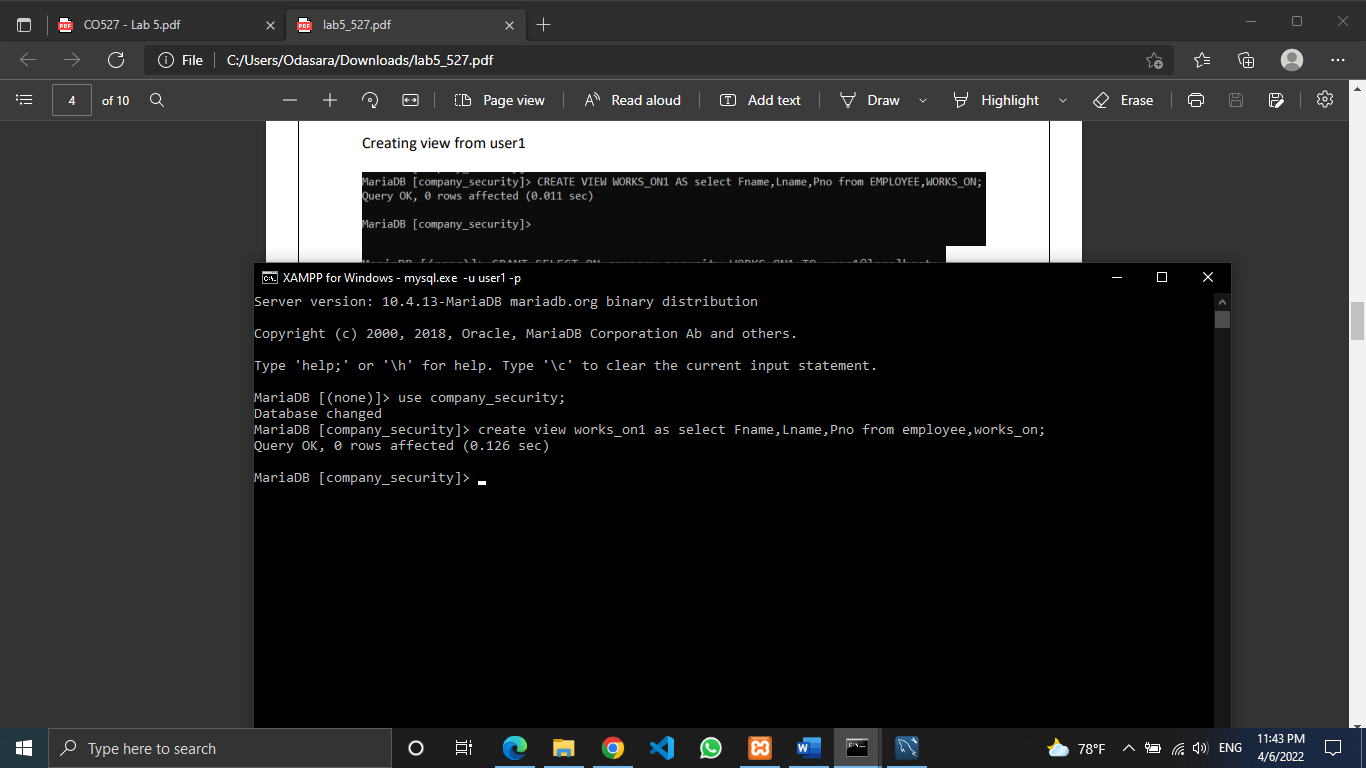
7.From user1 create a view WORKS ON1(Fname,Lname,Pno) on EMPLOYEE and WORKS ON. (Note: You will have to give permission to user1 on CREATE VIEW). Give another user ‘user2’ permission to select tuples from WORKS ON1(Note: user2 will not be able to see WORKS ON or EMPLOYEE)

Giving permission to user1

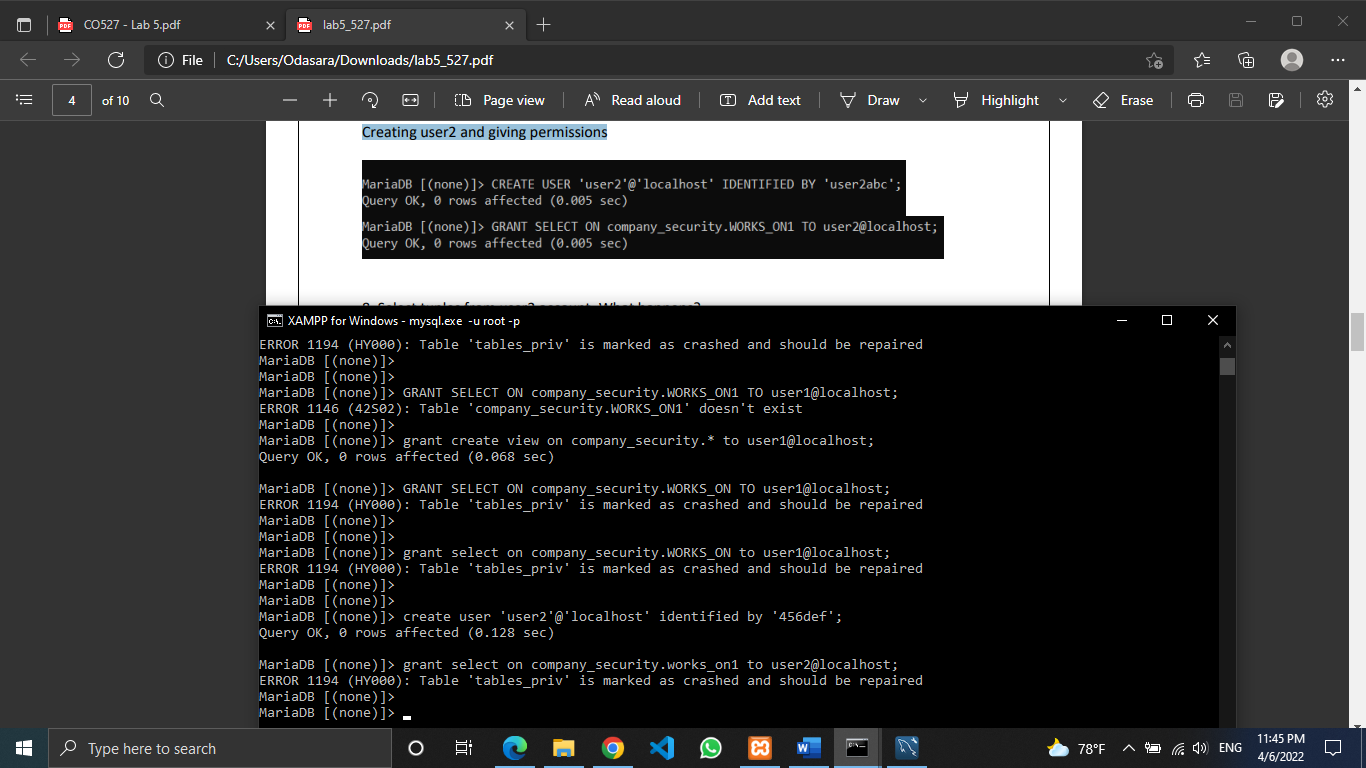


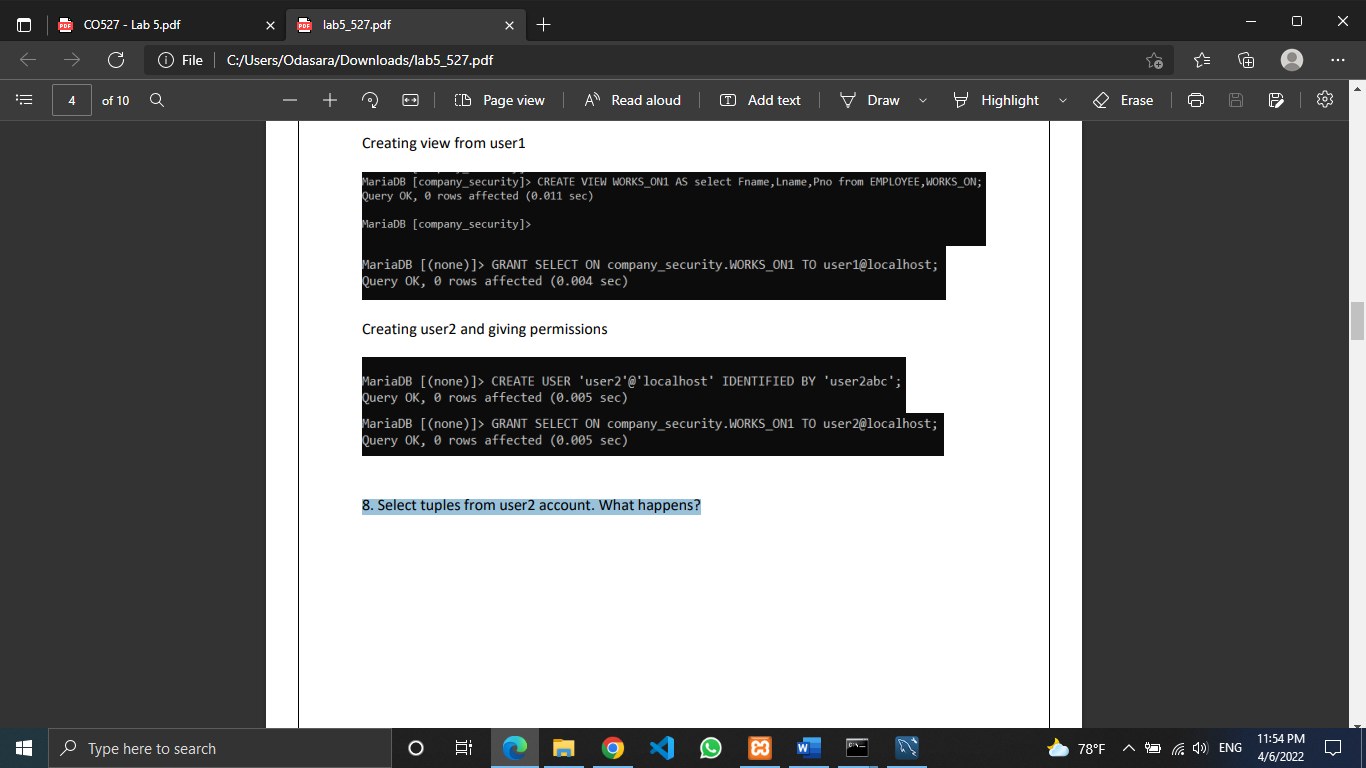


Creating a view from user1

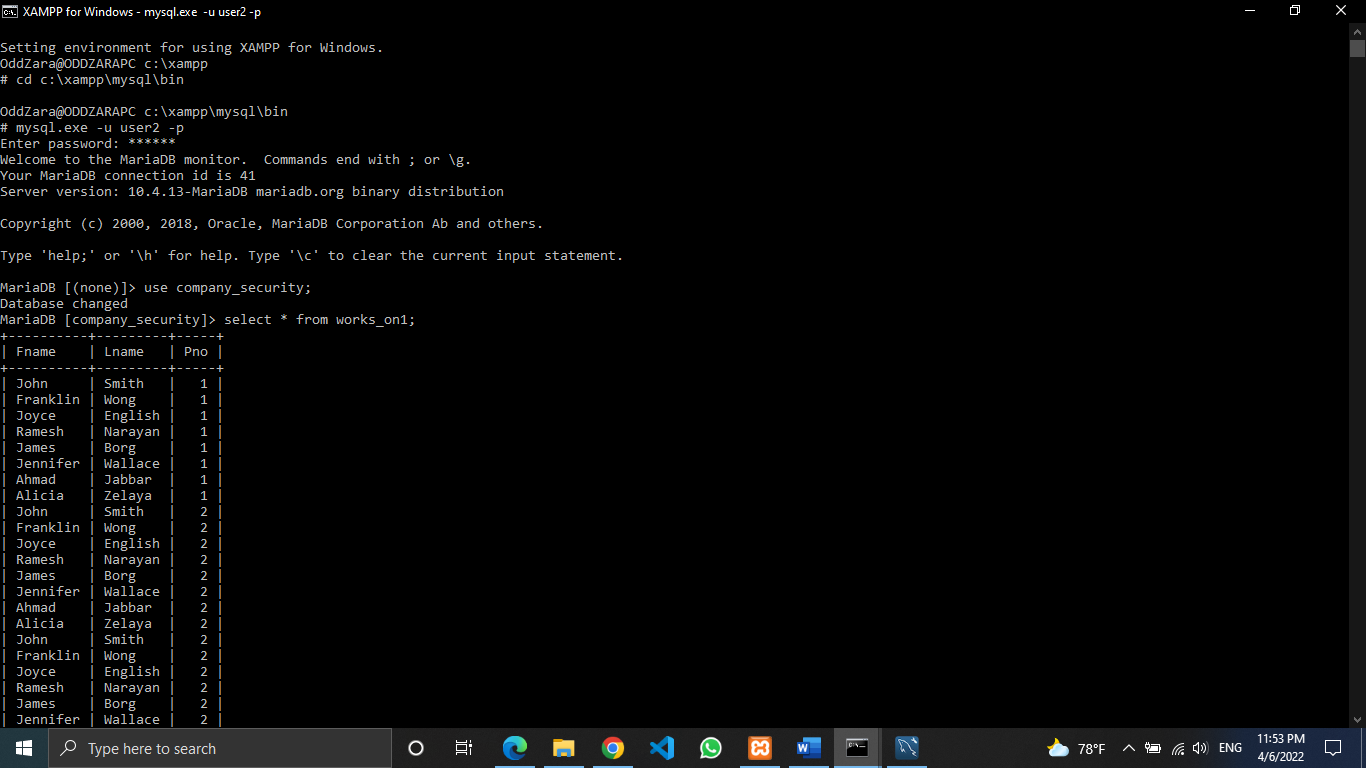


Creating user2 and giving permissions

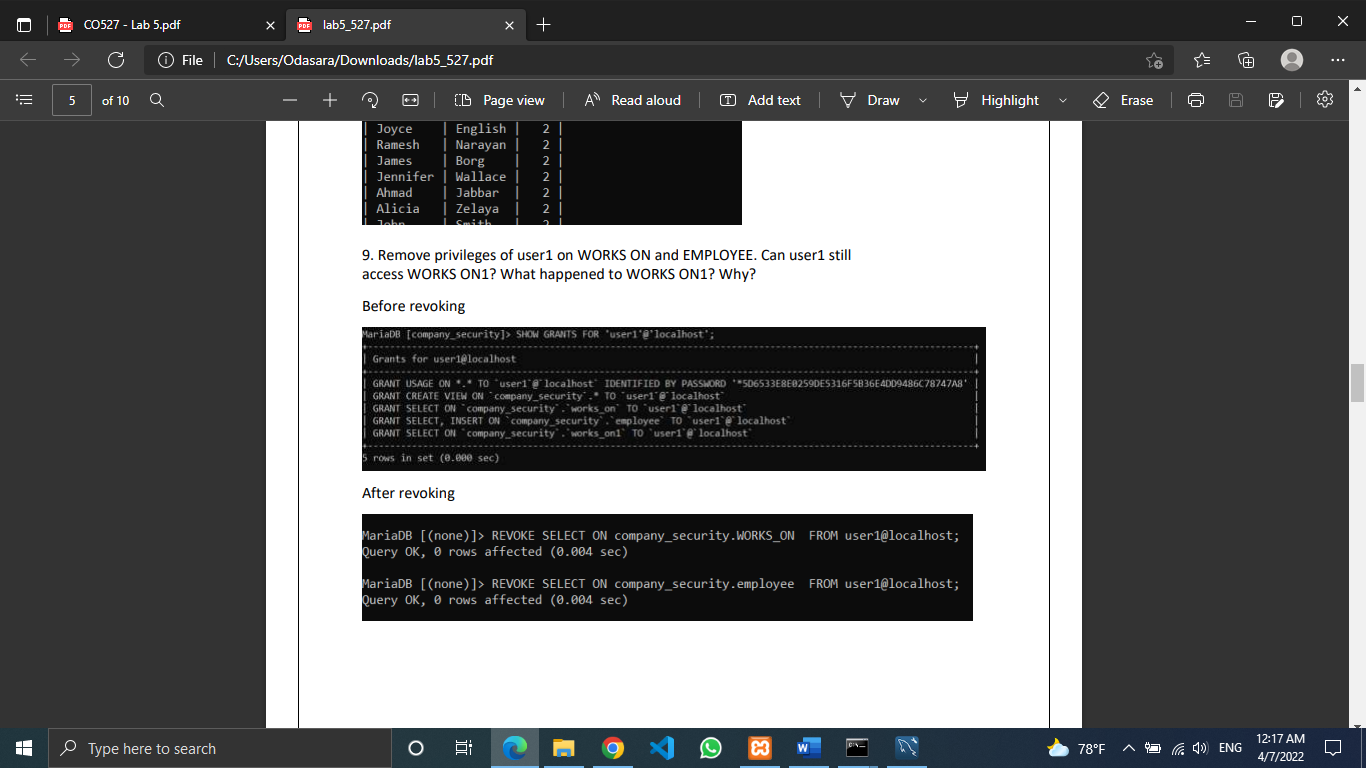




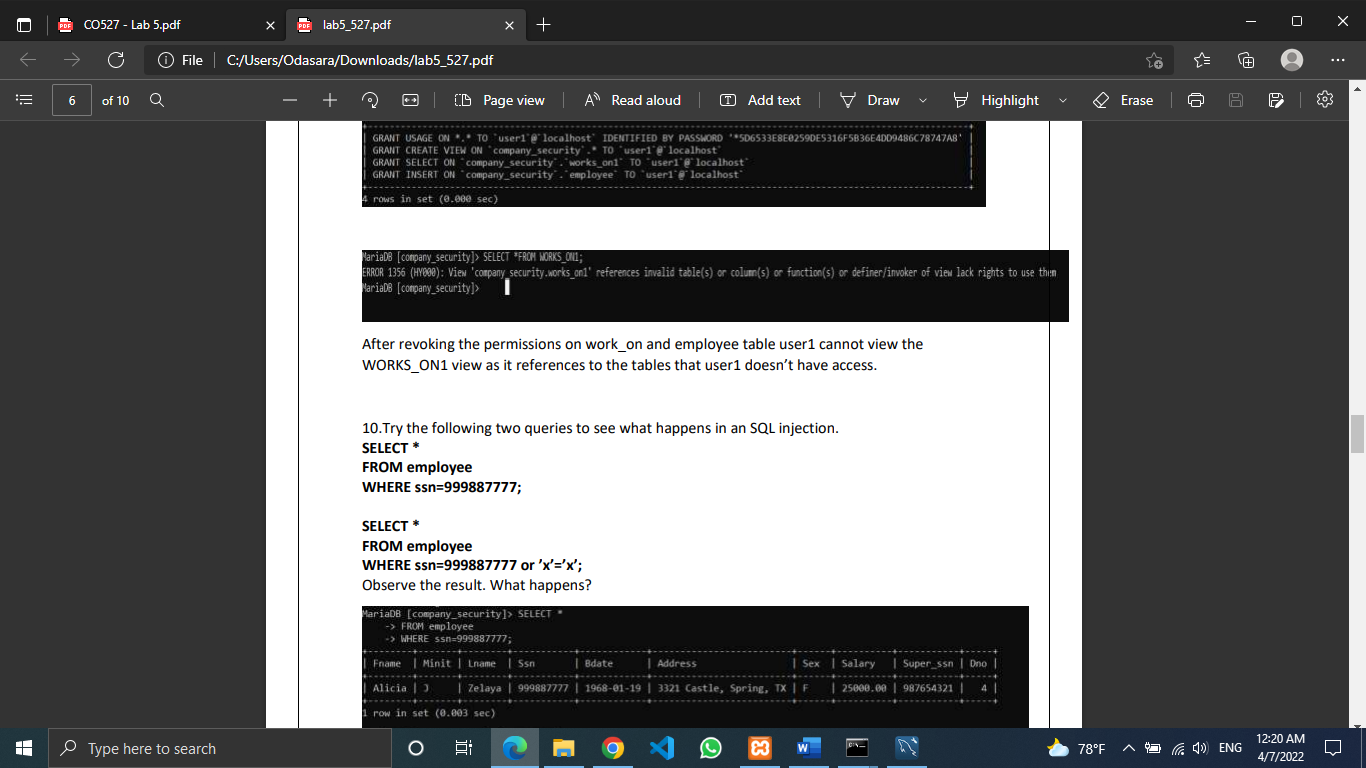
8. Select tuples from user2 account. What happens?



9.Remove privileges of user1 on WORKS ON and EMPLOYEE. Can user1 still access WORKS ON1? What happened to WORKS ON1? Why?



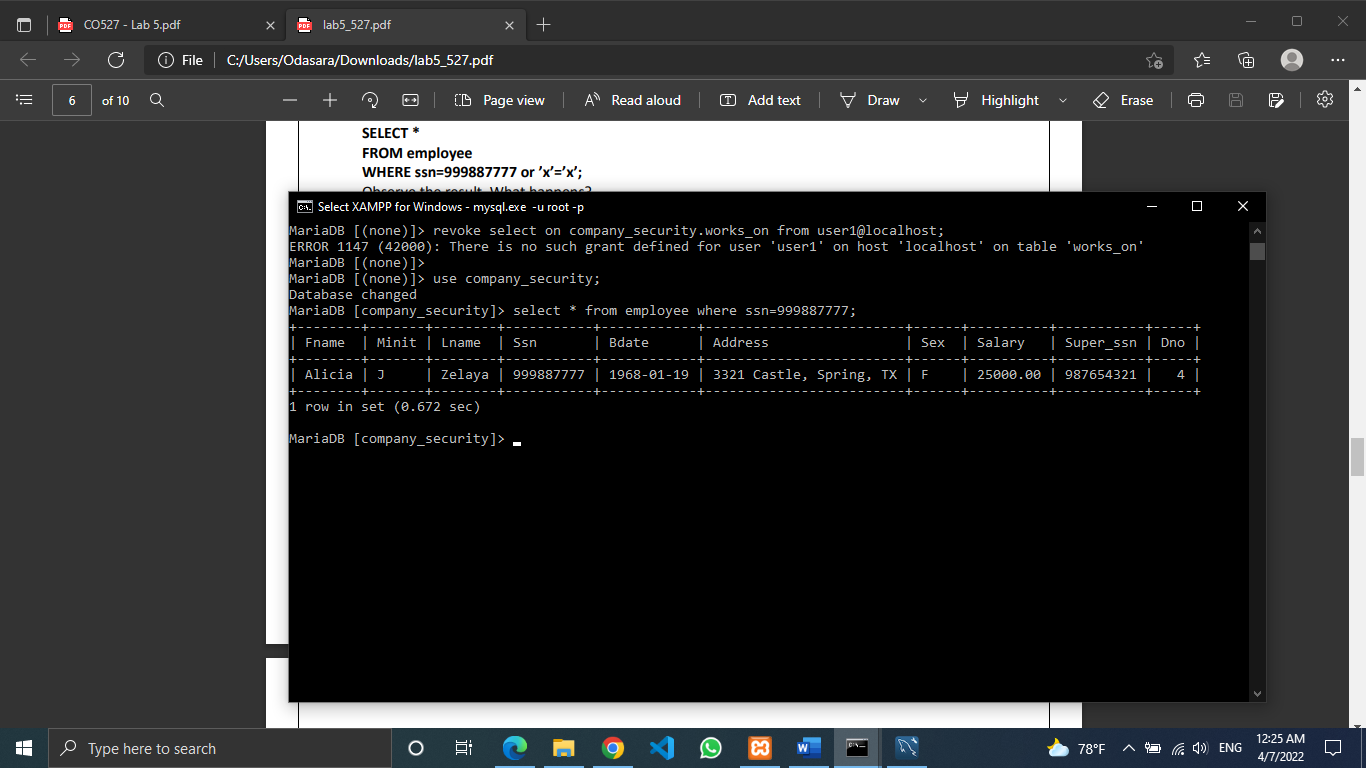
After revoking the permissions on works\_on and employee table user1 cannot view the WORKS\_ON1 view as it references to the tables that user1 doesn’t have access



10.Try the following two queries to see what happens in an SQL injection

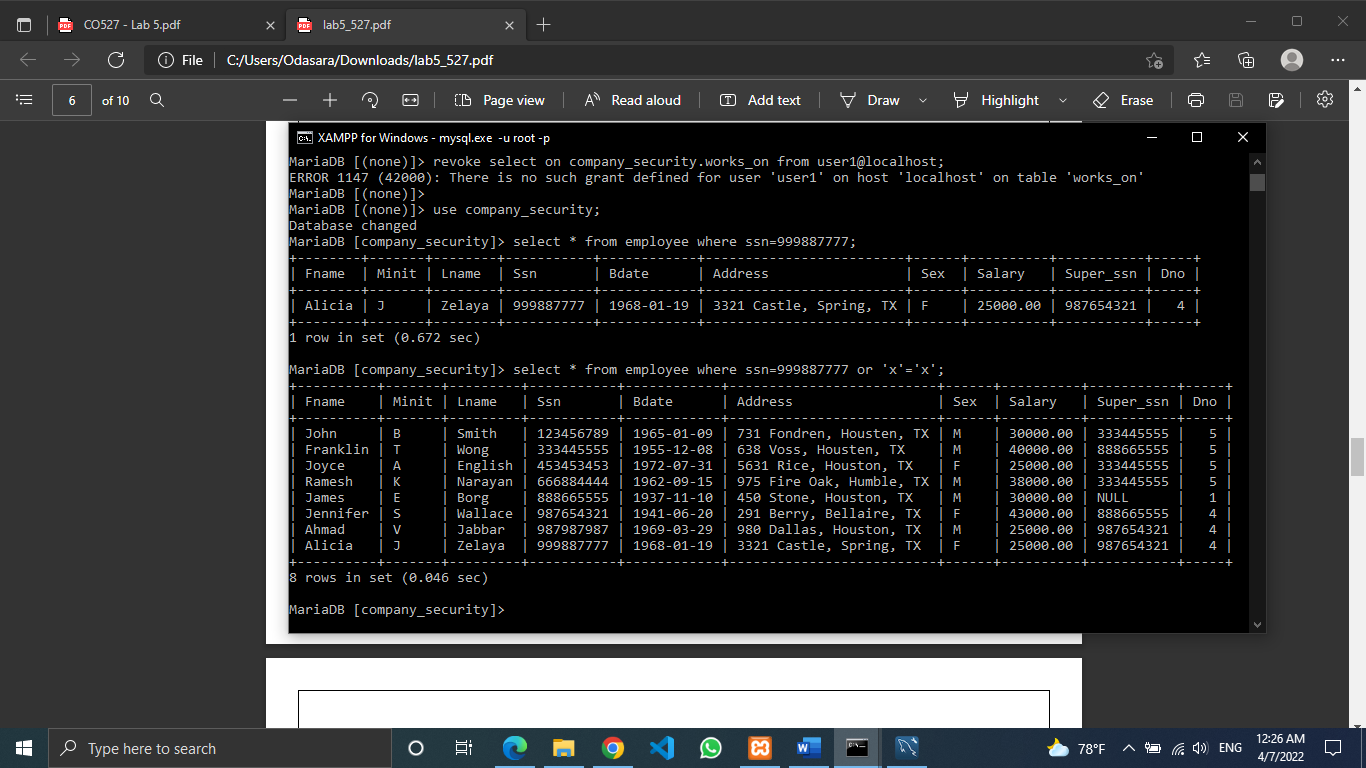
Query 1:

SELECT \* FROM employee WHERE ssn=999887777;



Query 2:

SELECT \* FROM employee WHERE ssn=999887777 or ’x’=’x’;

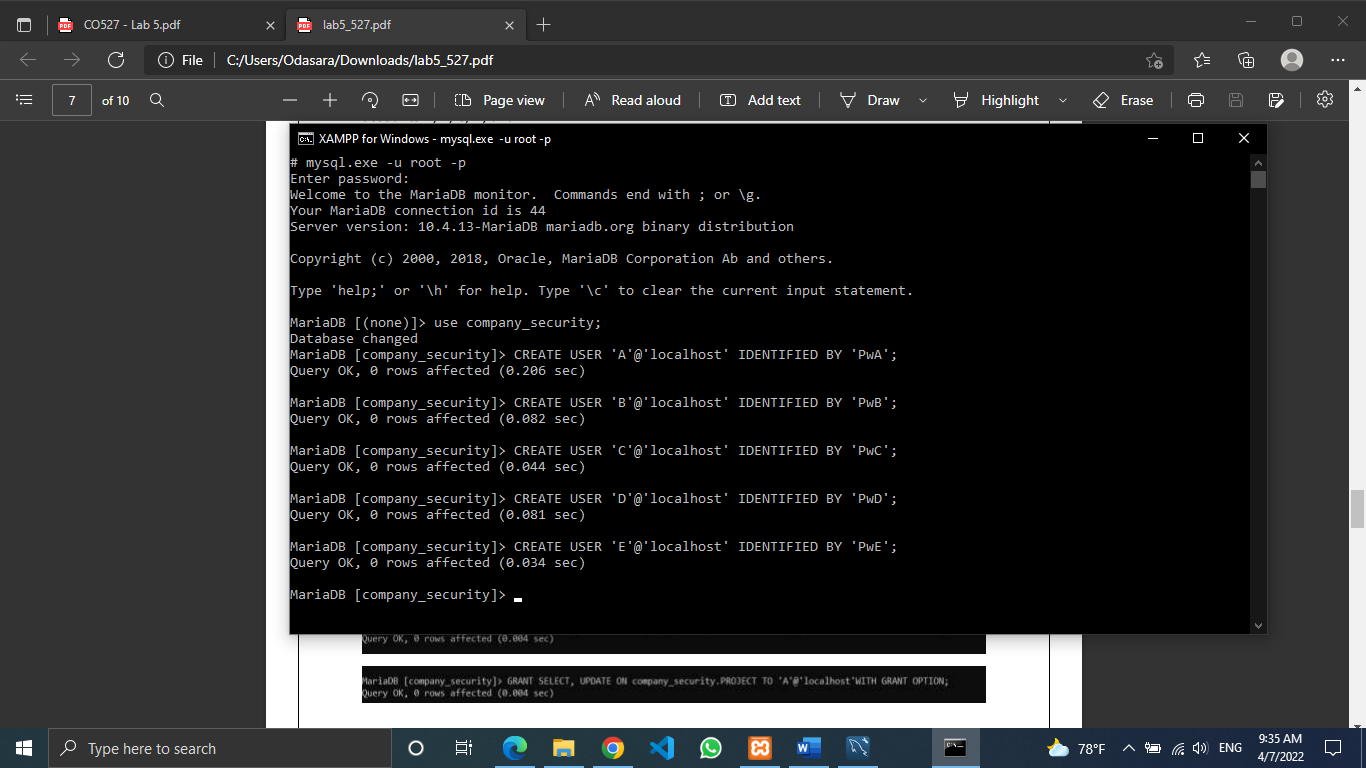


Query 1 outputs only the details of employee who has ssn=999887777.

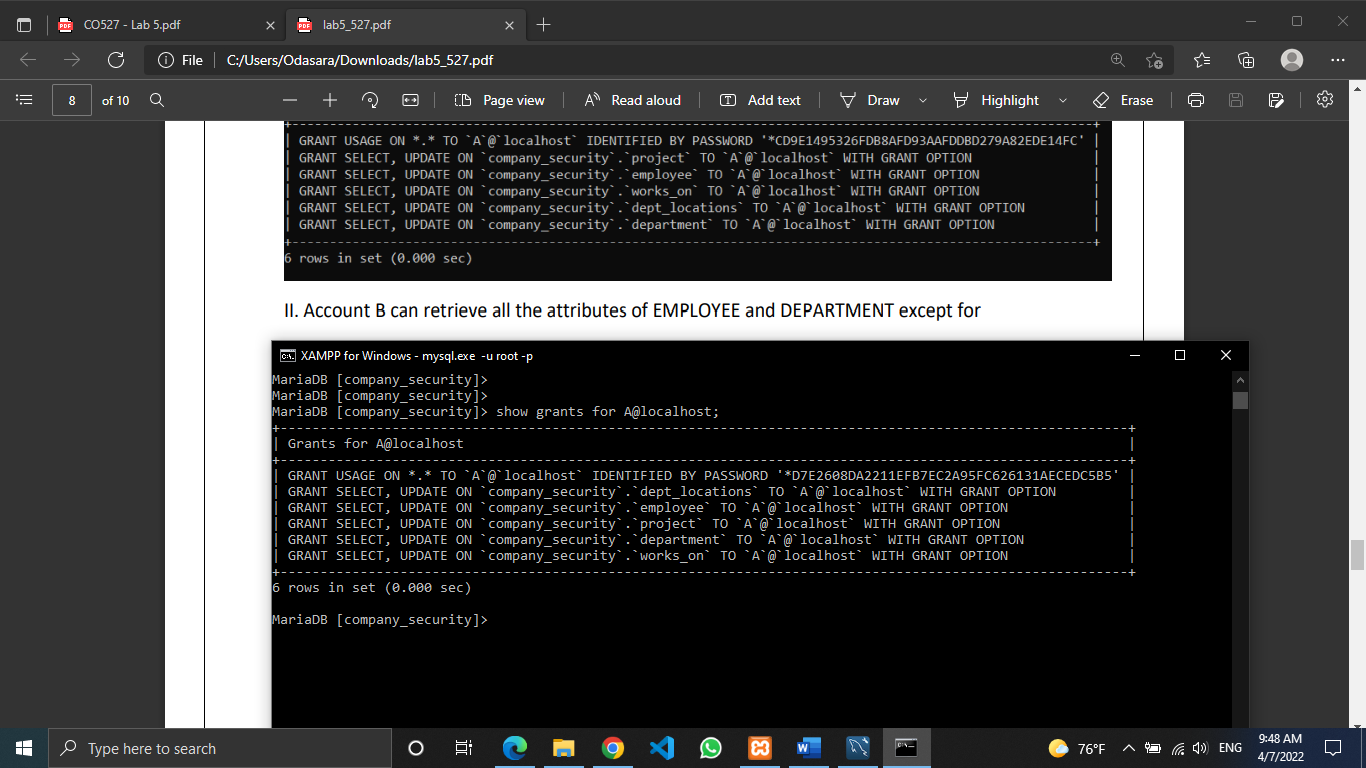
Query 2 outputs details of all employees. This conveys that the attacker who knows there is a valid login for ssn=999887777 can login to the database system as an authorized user without knowing his password and is able to do everything that who has ssn=999887777 may be authorized to do to the database system.

5.Assignment

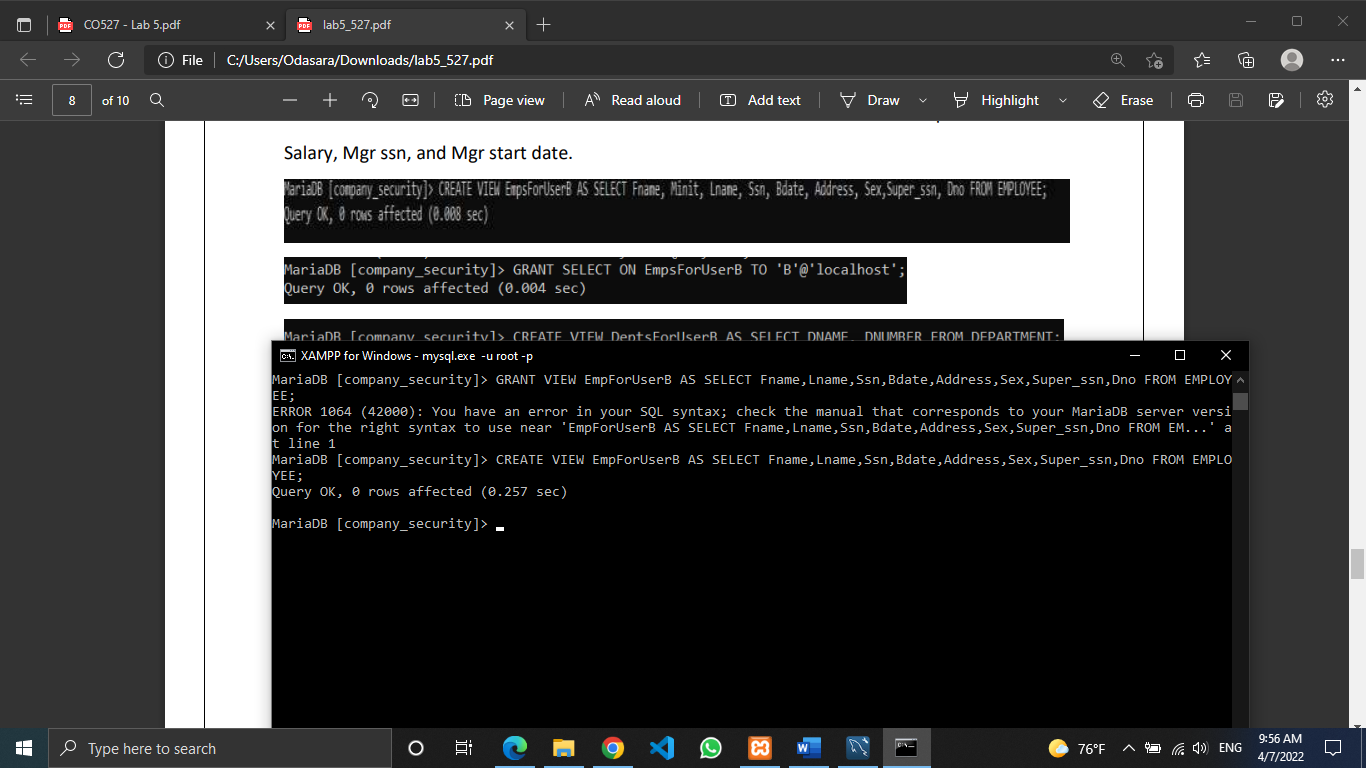
Consider the relational database schema provided. Suppose that all the relations were created by (and hence are owned by) user X, who wants to grant the following privileges to user accounts A, B, C, D, and E:

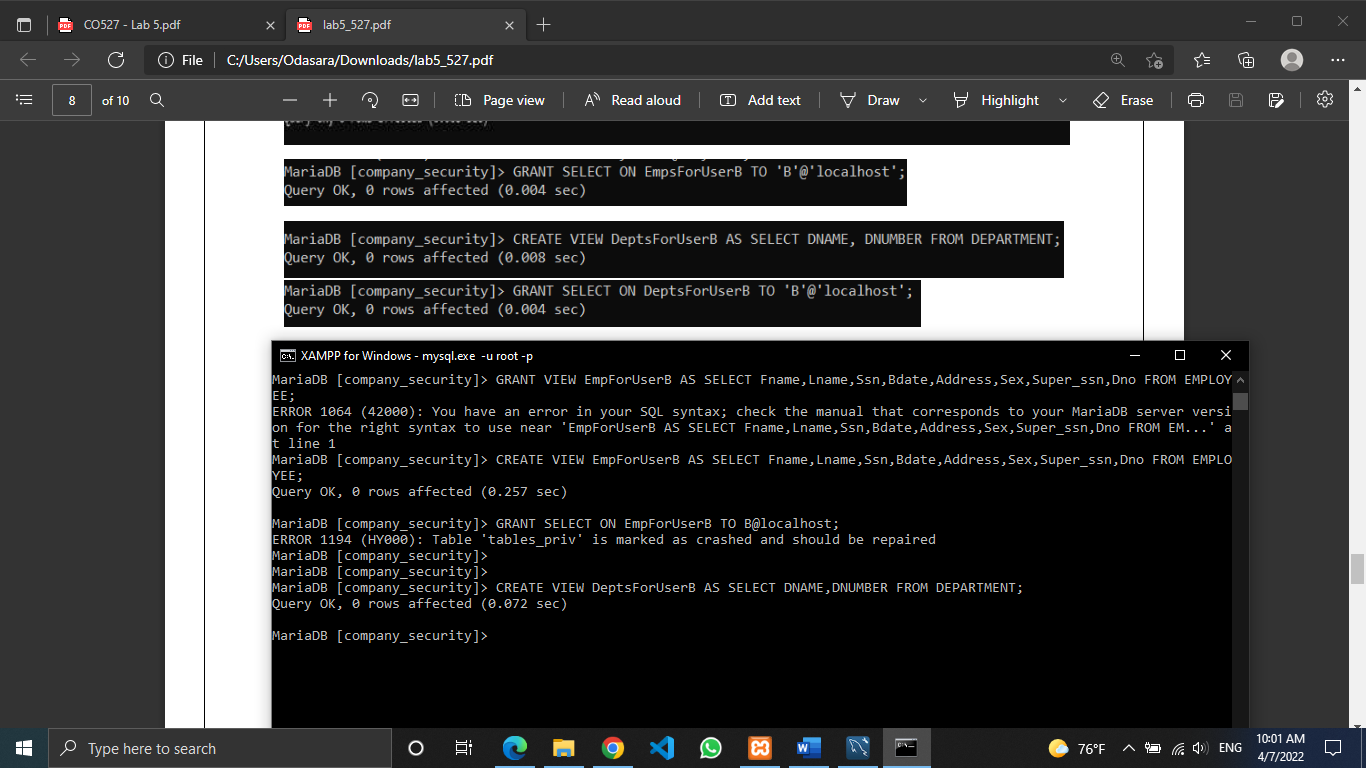


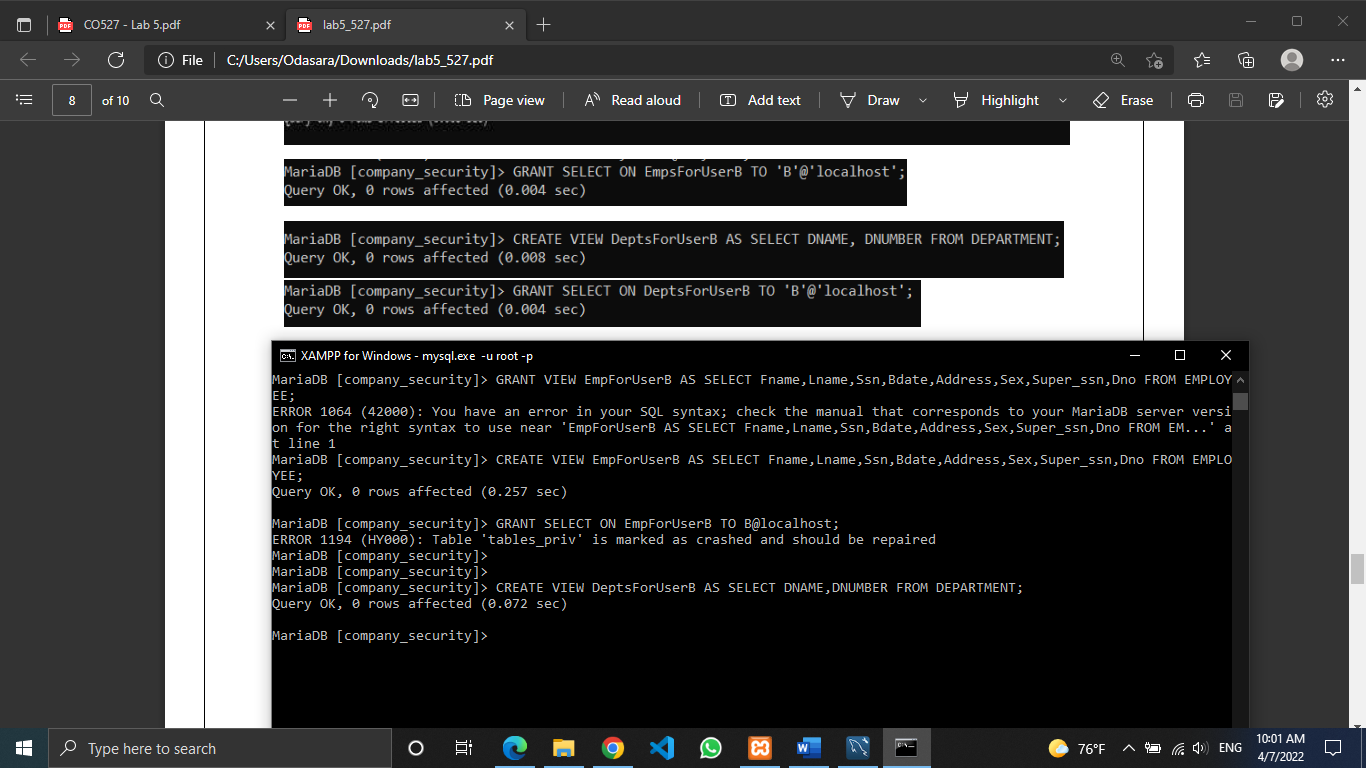
I.Account A can retrieve or modify any relation except DEPENDENT and can grant any of these privileges to other users.

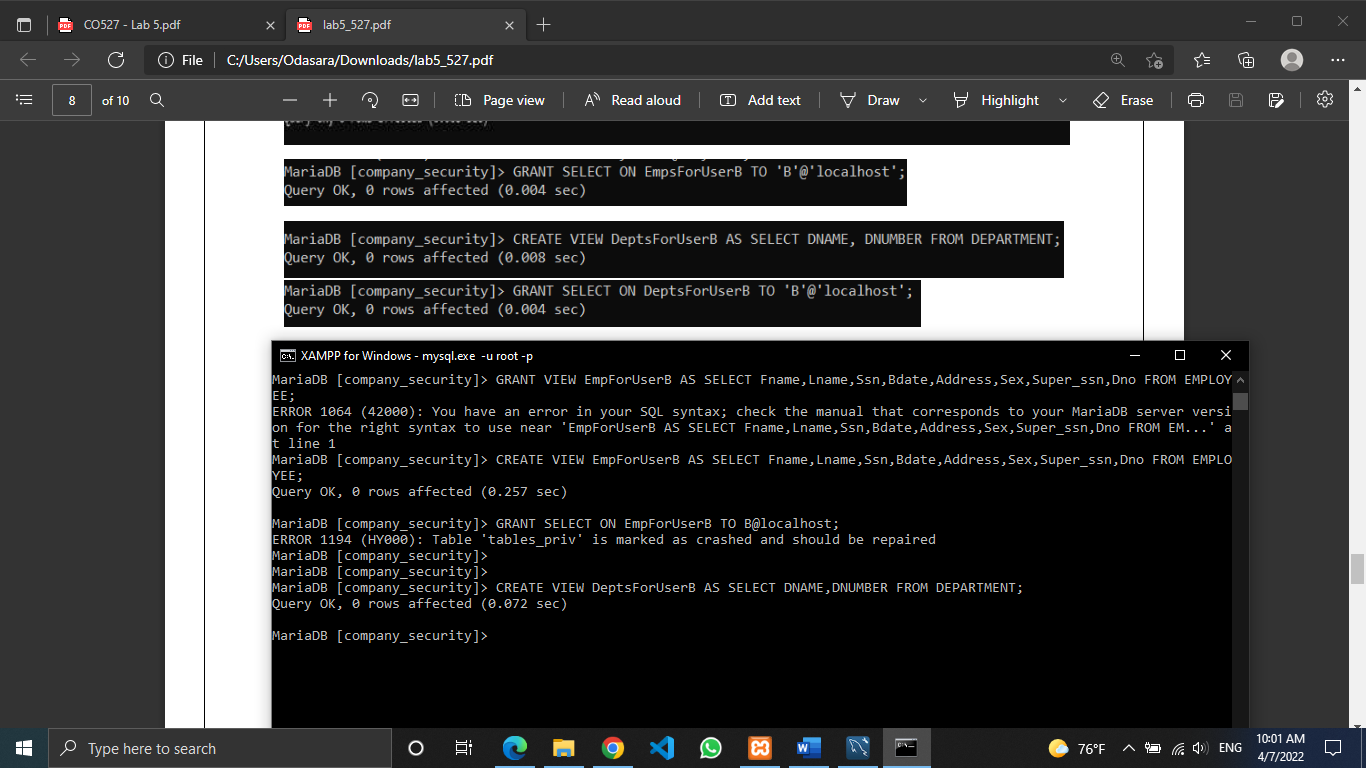


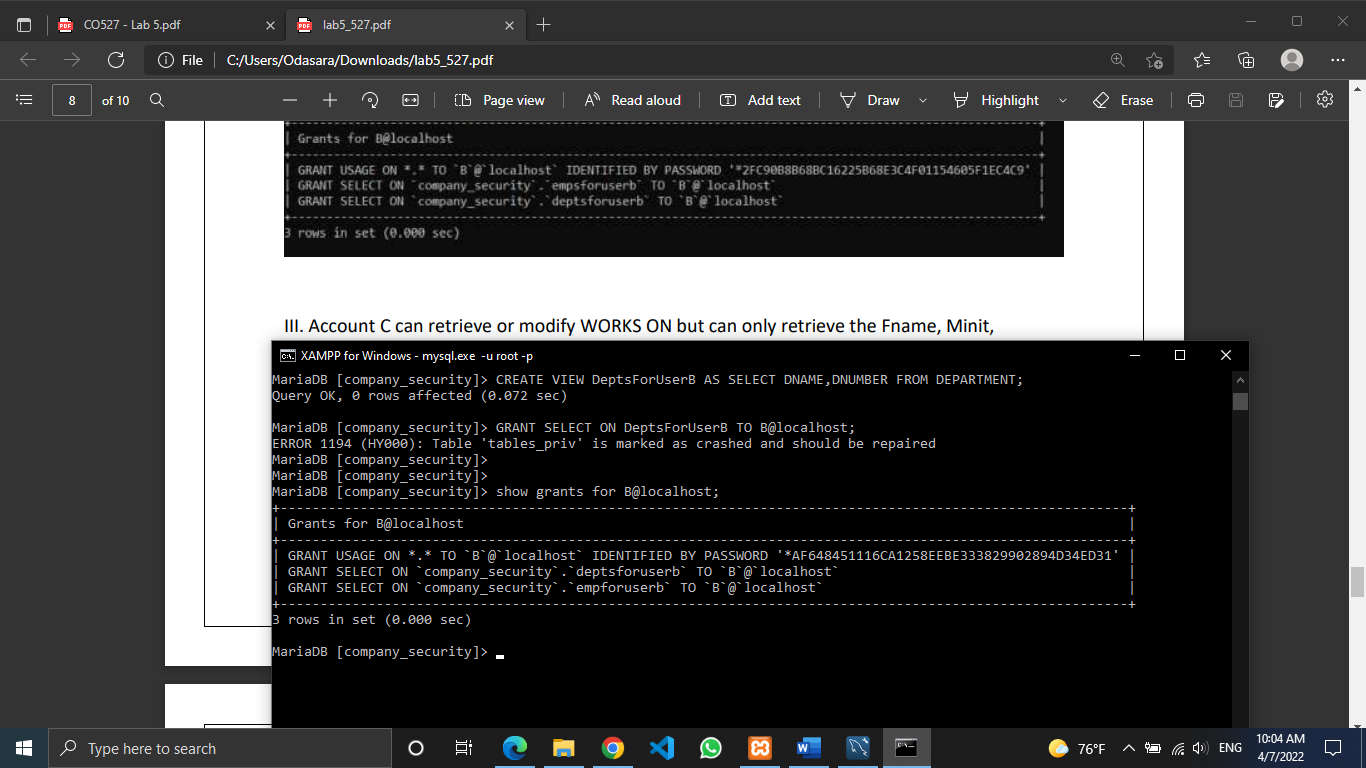
II. Account B can retrieve all the attributes of EMPLOYEE and DEPARTMENT except for Salary, Mgr ssn, and Mgr start date.



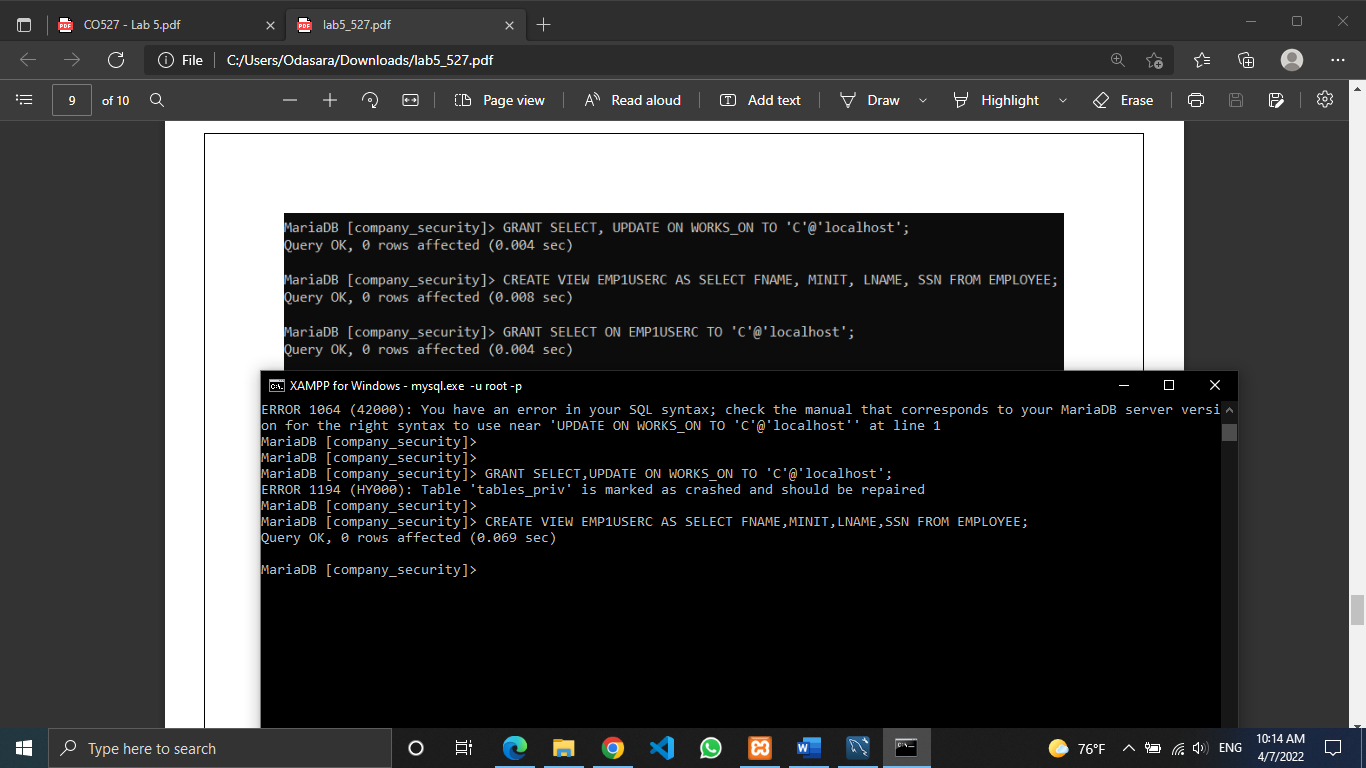


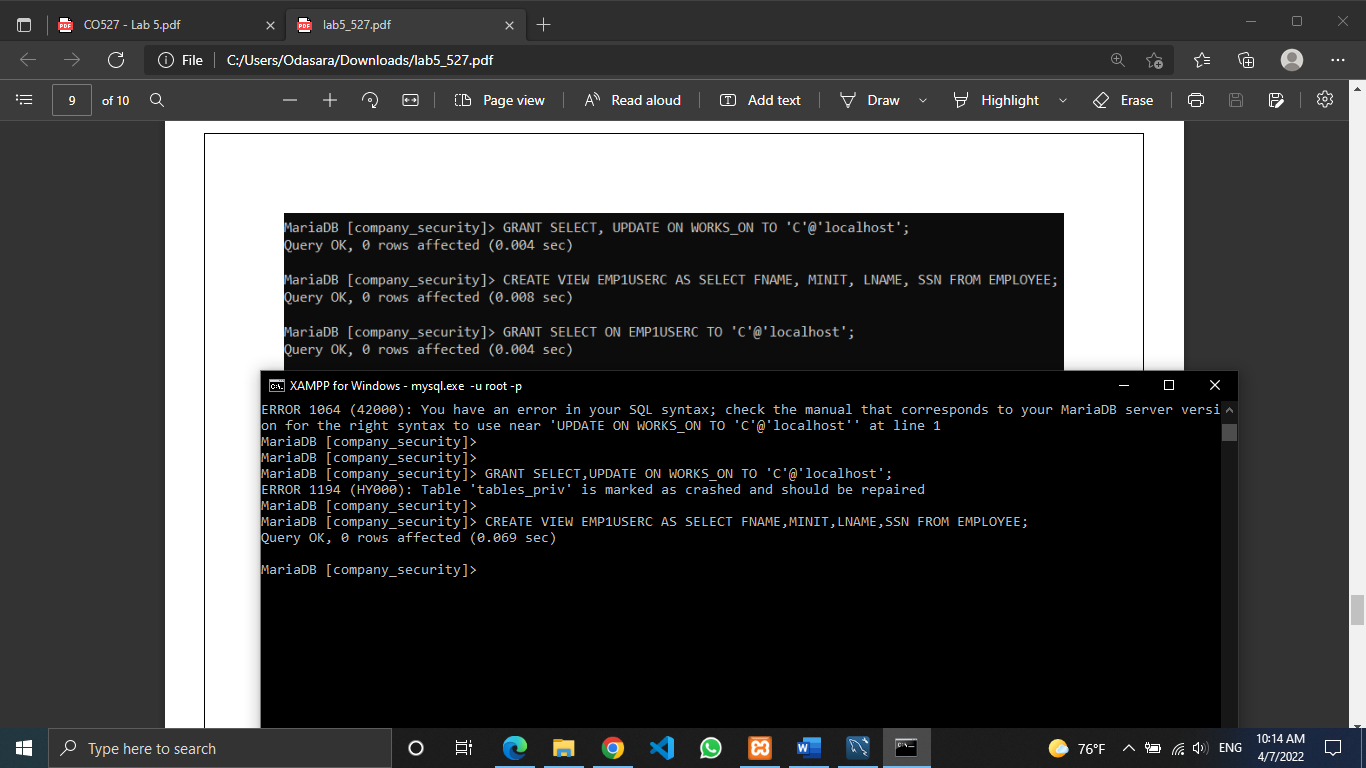


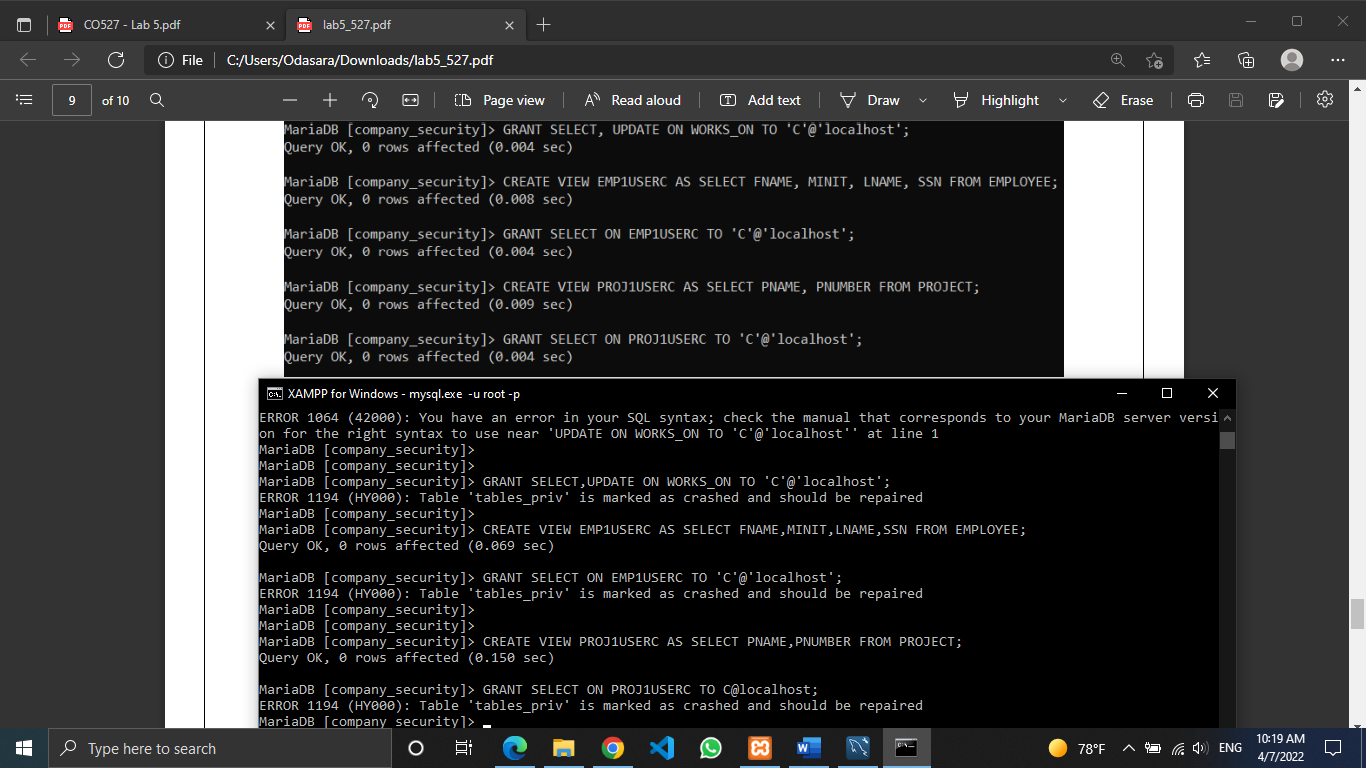


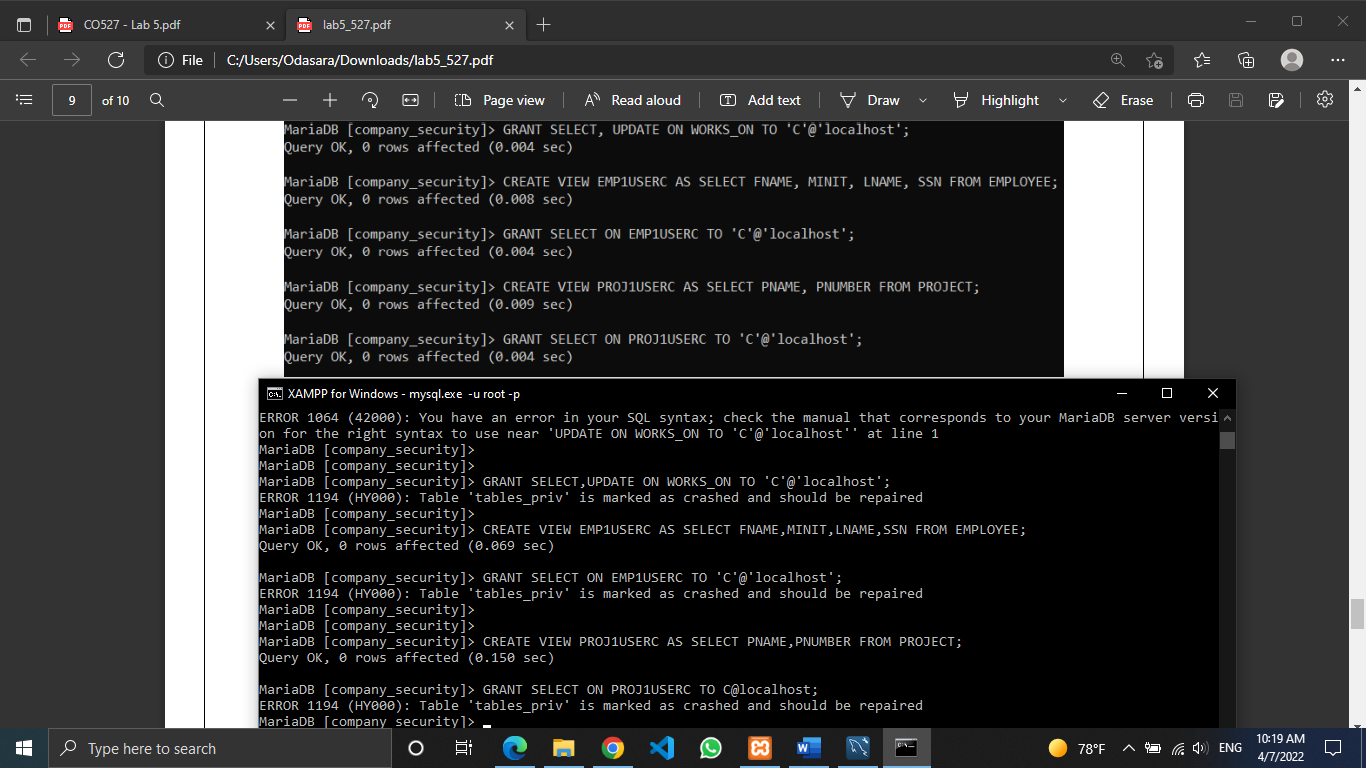


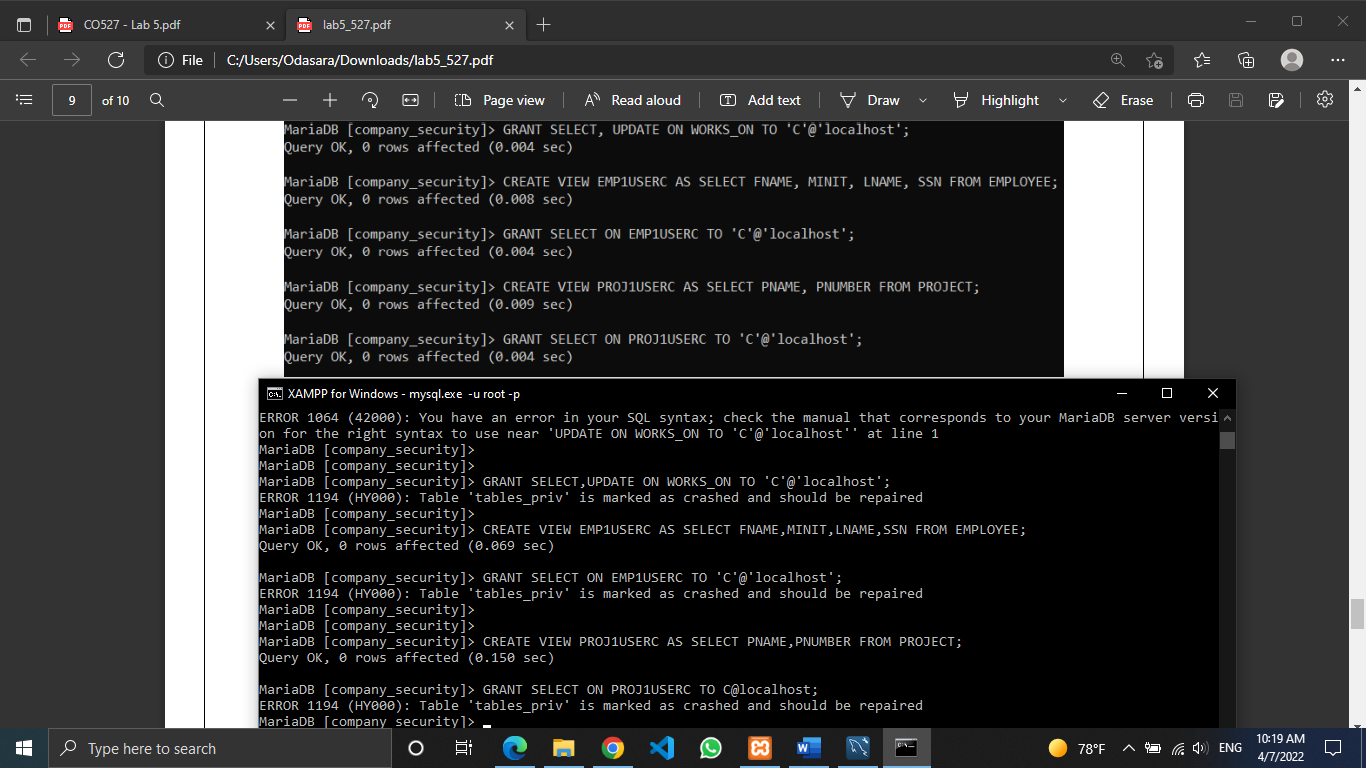
III. Account C can retrieve or modify WORKS ON but can only retrieve the Fname, Minit, Lname, and Ssn attributes of EMPLOYEE and the Pname and Pnumber attributes of PROJECT.



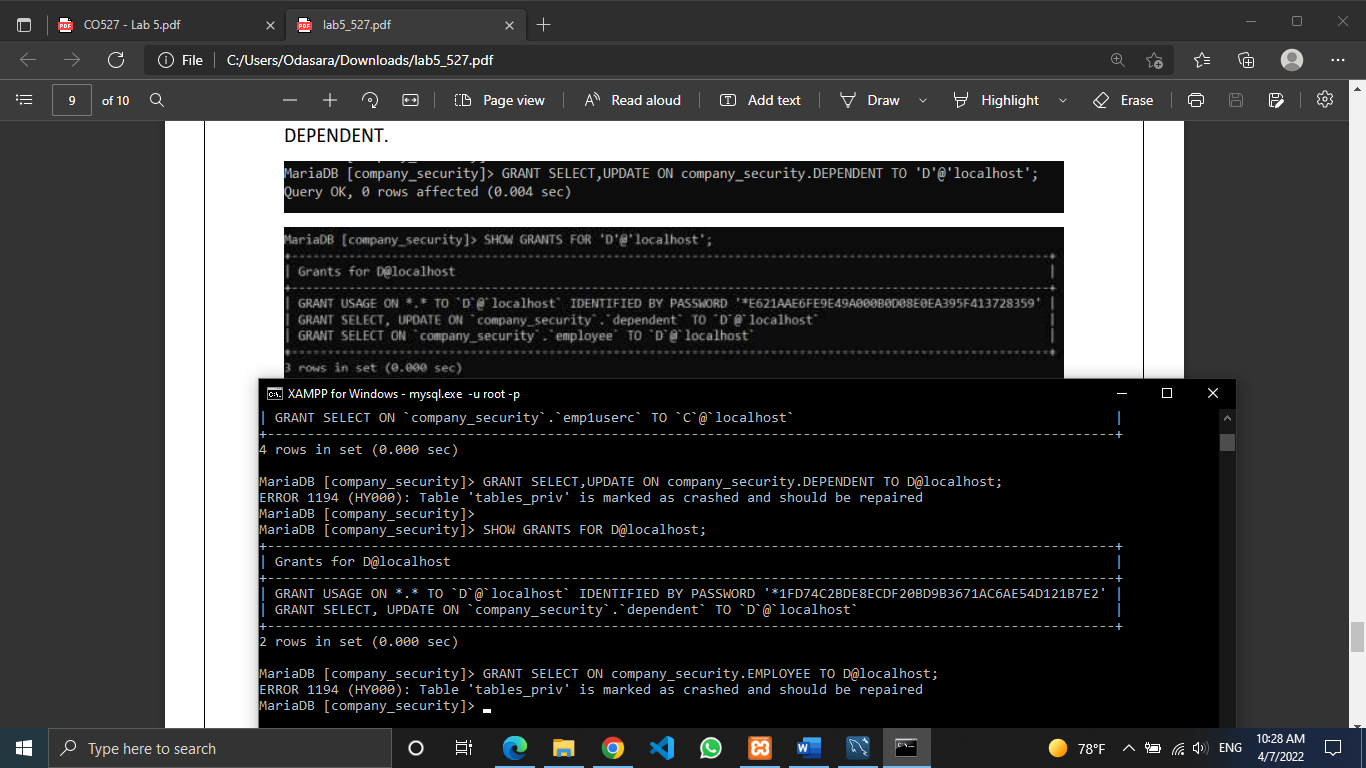


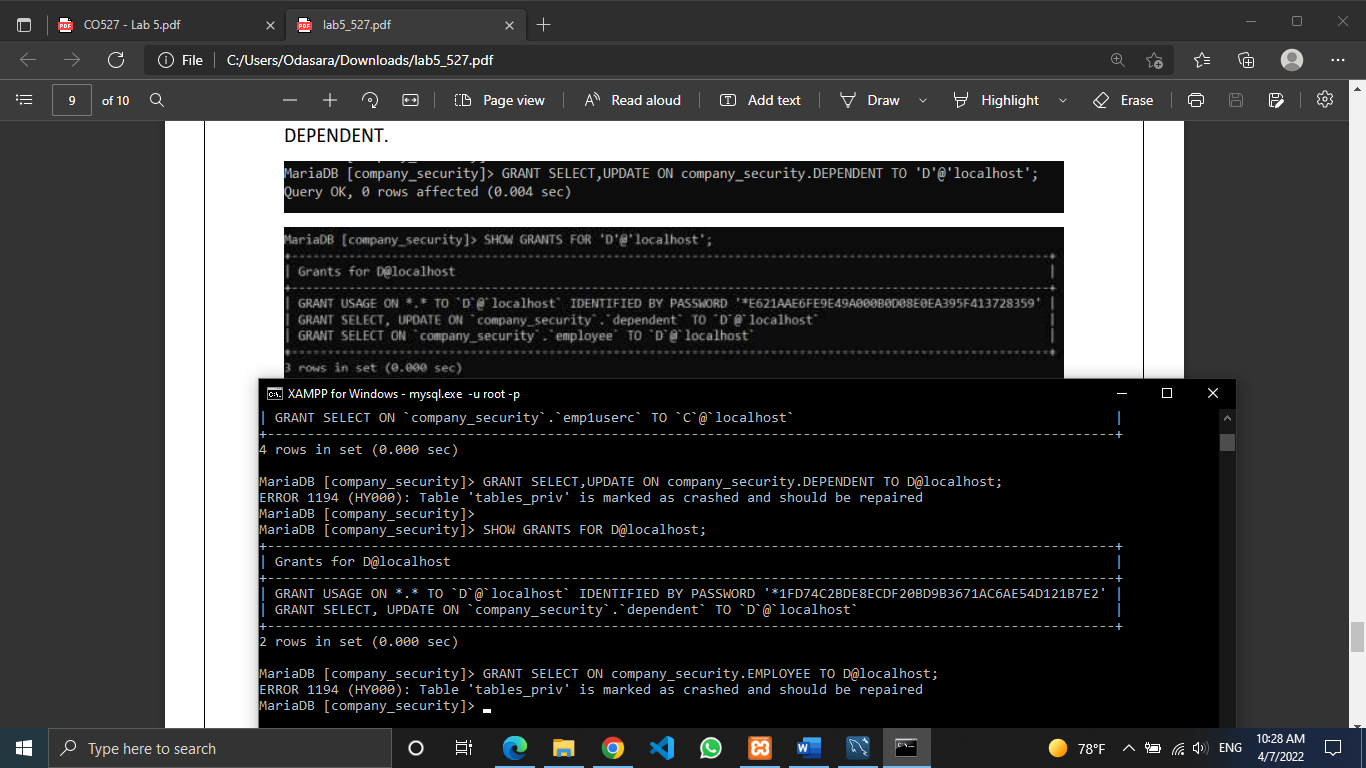


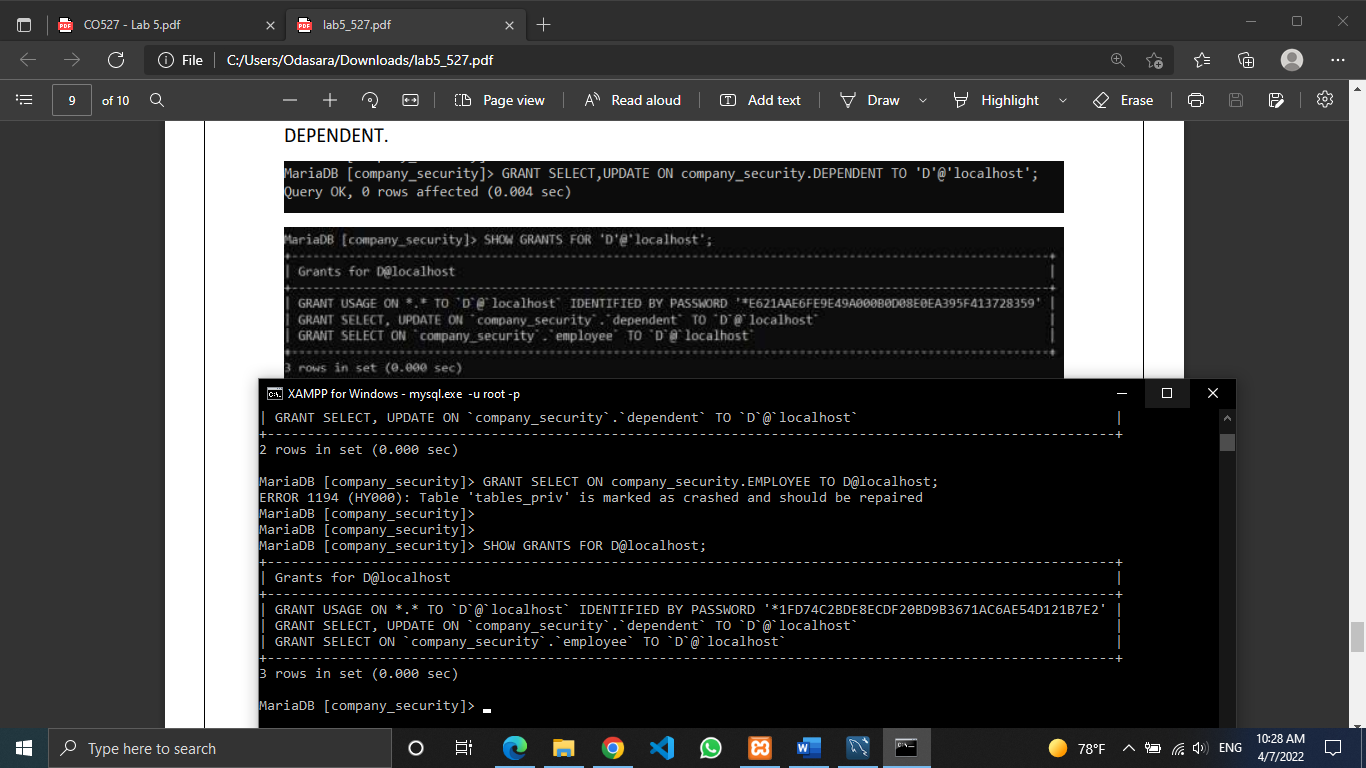




IV. Account D can retrieve any attribute of EMPLOYEE or DEPENDENT and can modify DEPENDENT.







V. Account E can retrieve any attribute of EMPLOYEE but only for EMPLOYEE tuples that have Dno = 3.

