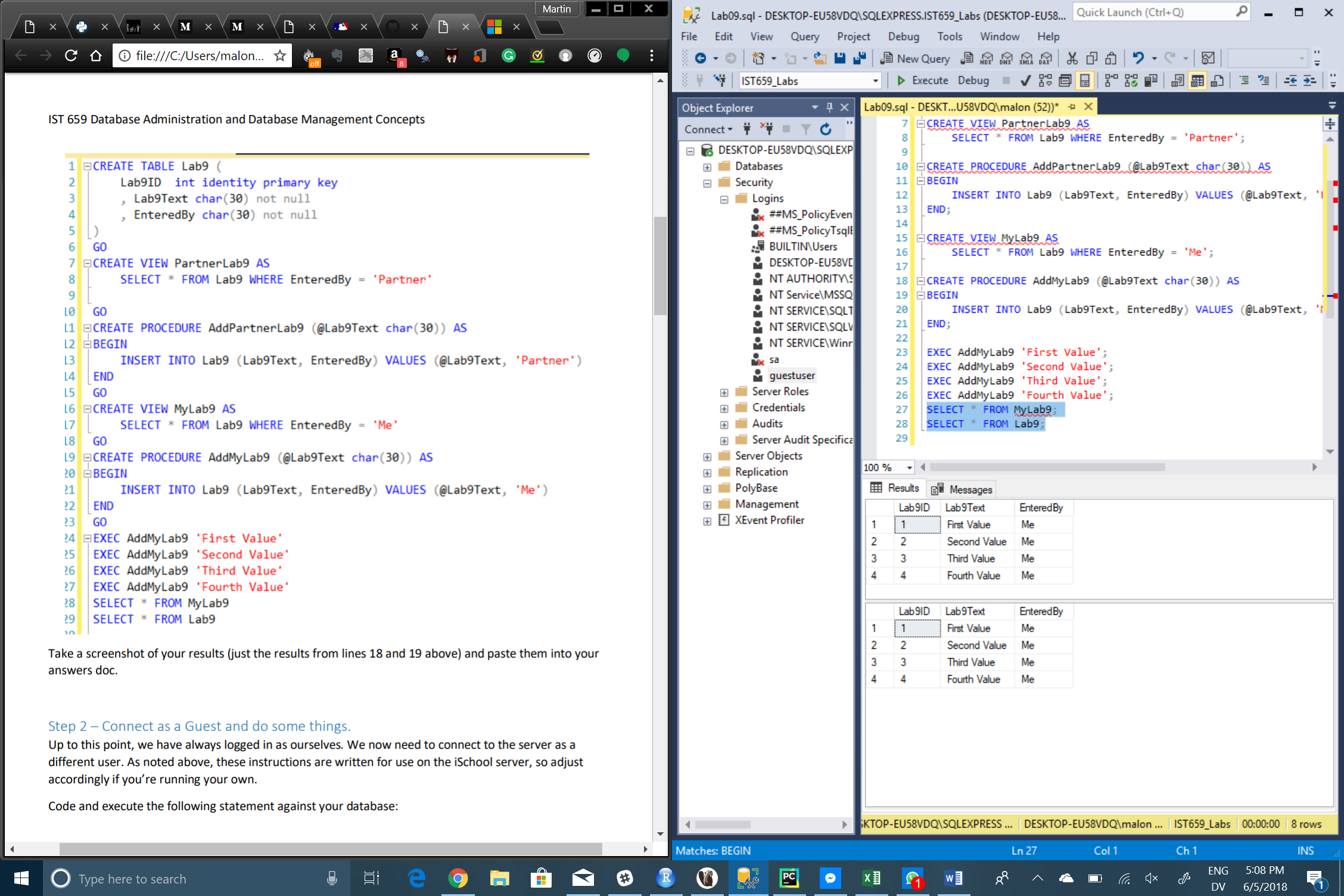
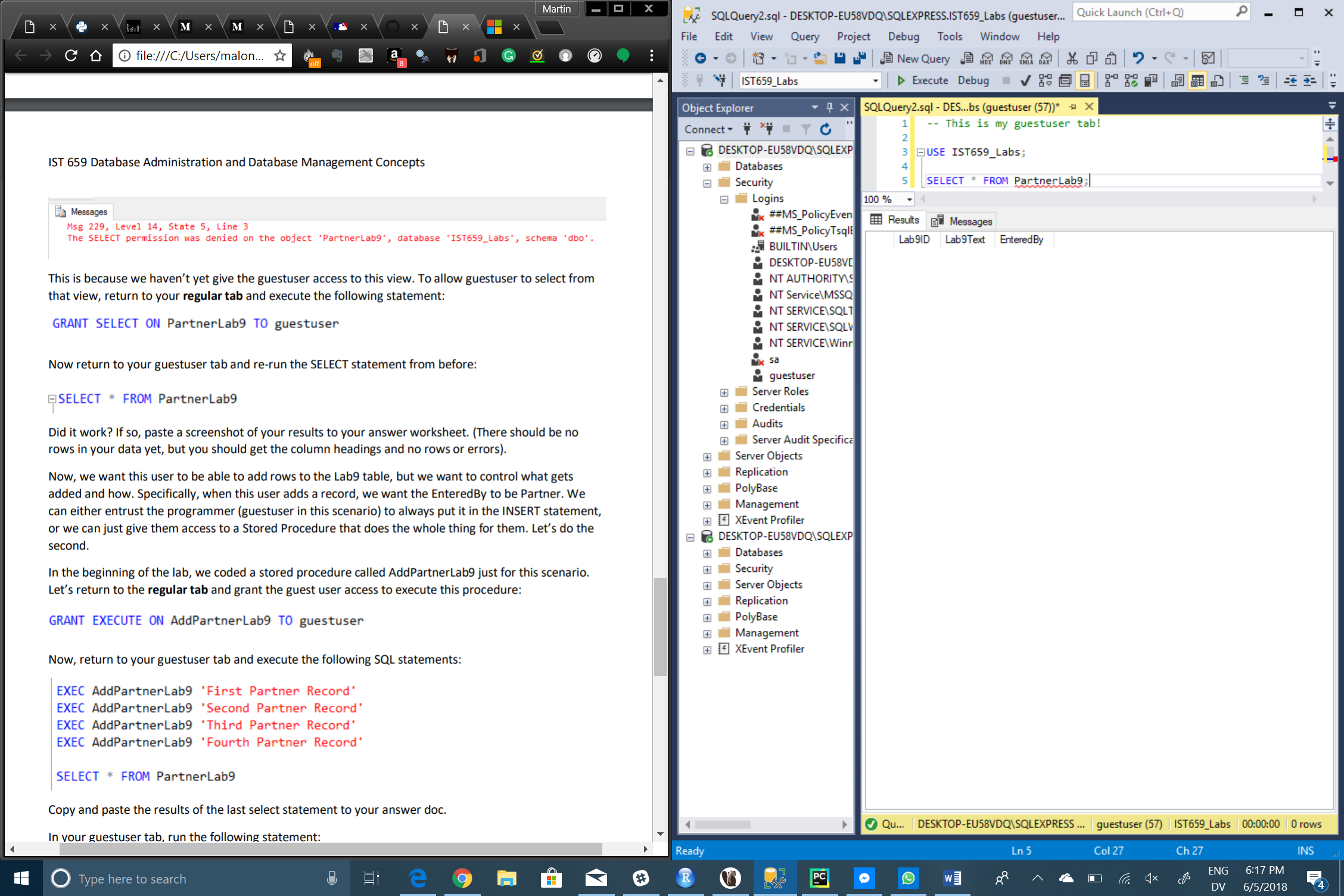
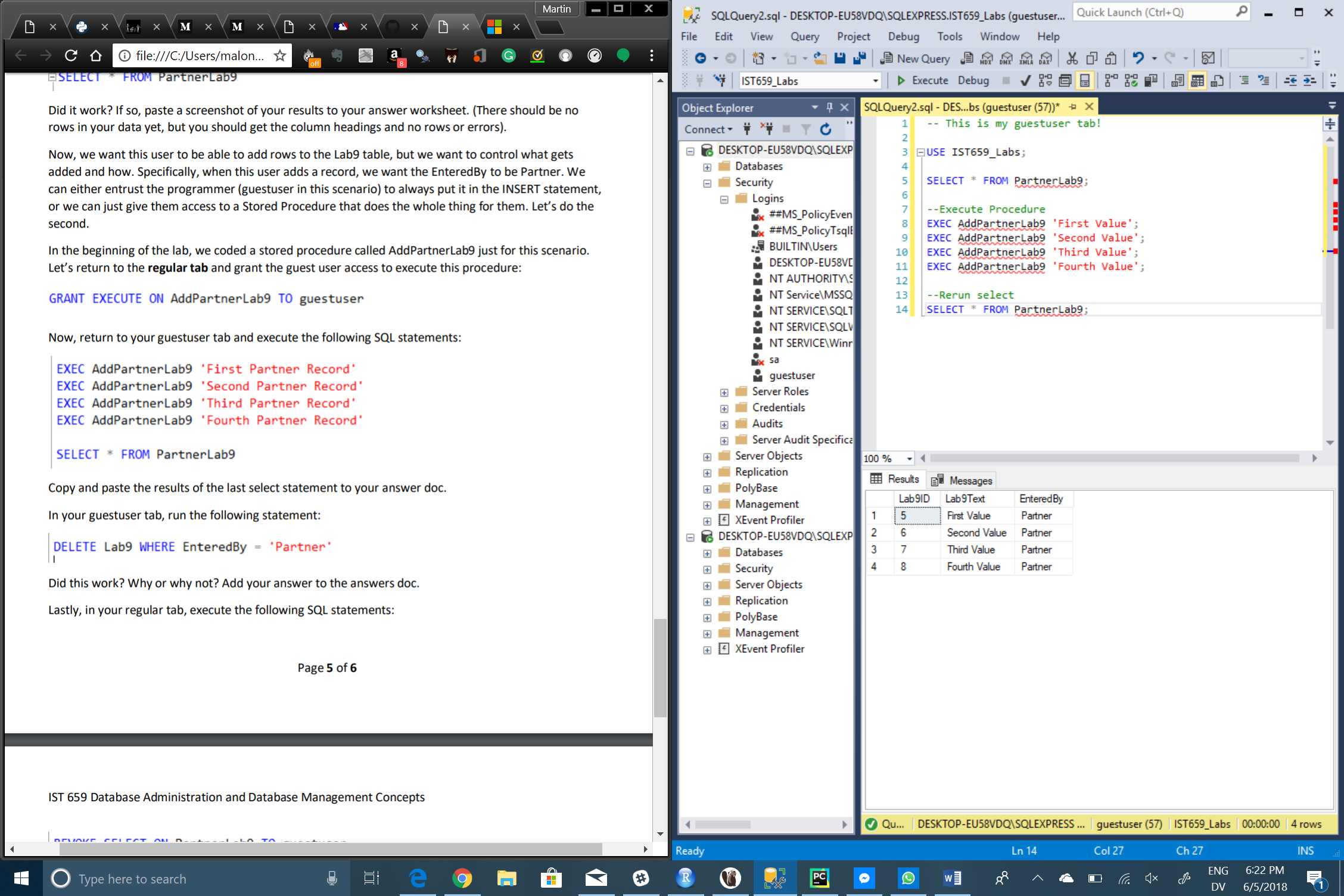
Step 1 – Create Some Data



Step 2 – Connect as a Guest and do some things





--Delete statement

DELETE Lab9 WHERE EnteredBy = 'Partner';

Delete statement doesn’t work because we haven’t granted DELETE access to the guestuser.

3 – Additional questions

1. What does the CREATE USER statement do? Hint: It’s not creating a user….

The CREATE USER statement creates a database user that is able to log onto the SQL Server.

1. Which types of objects are securable with GRANT and DENY statements? (Research required. )

Objects that can be GRANTed or DENied permissions are: Databases, Objects, Schemas, Searches, Servers, Procedures, Tables, Views, and Systems.

1. Describe a scenario by which you would not want a user reading the table directly but rather using a view instead.

A scenario in which I would grant user a view table instead of a reading table is for reports that need to be generated constantly and that require no further data manipulation whatsoever.

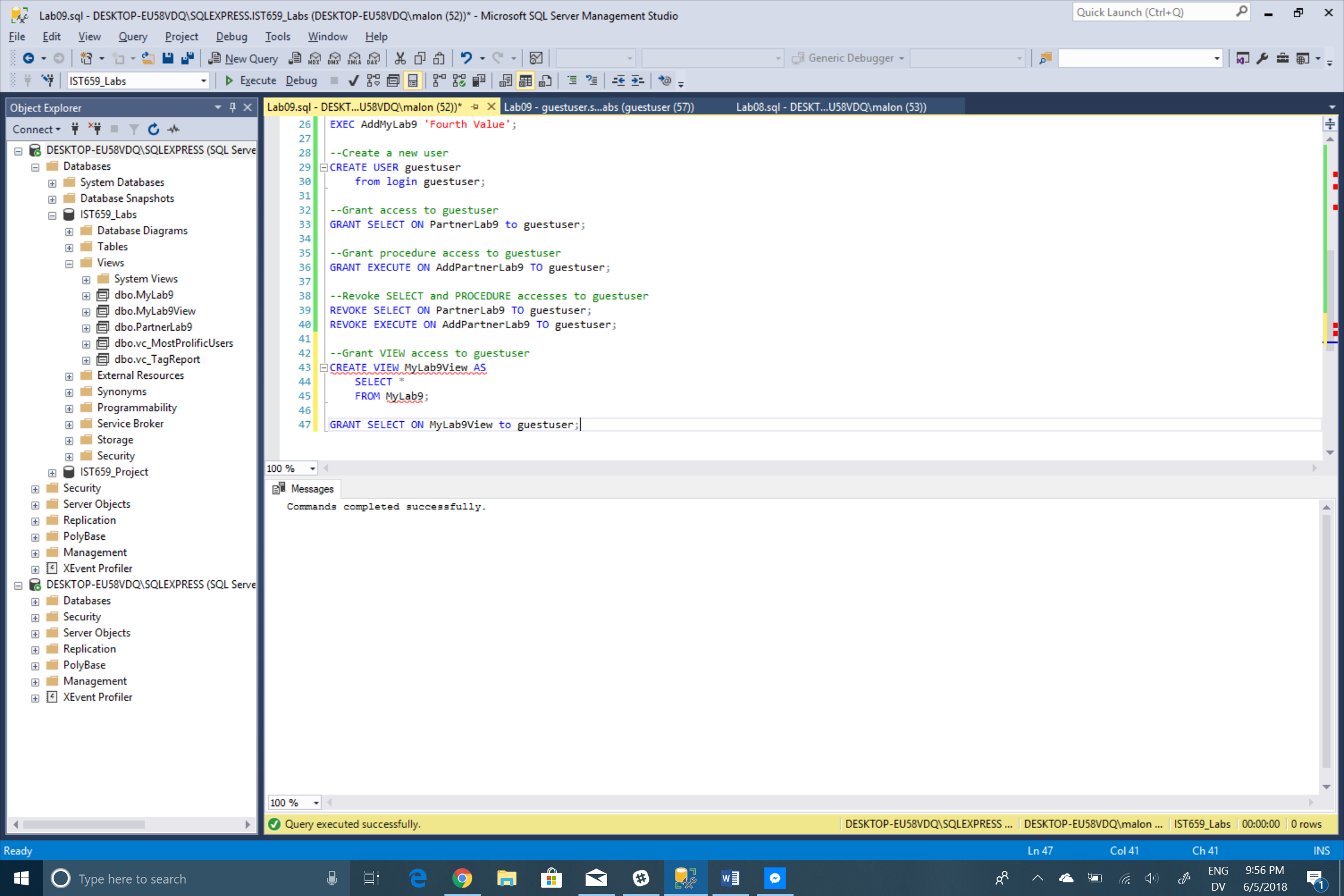
1. Describe a scenario by which you would not want a user updating data in the table directly but rather using a stored procedure instead?

Such a scenario would be a sales report that updates on a daily basis after all sales have been tallied and finalized. If the user were to manually update it when there is already a stored procedure, it could lead to double counting certain sales items.

1. Write SQL to deny the guest user permissions to delete from a table called Product.

DENY SELECT ON Product TO guestuser;

1. Write SQL to allow the guest user the ability to run one of your SQL Views. Place your sql here and the message returned when executed.



SQL Code

--This is my regular tab!

CREATE TABLE Lab9 (

Lab9ID int identity primary key,

Lab9Text char(30) not null,

EnteredBy char(30) not null

);

CREATE VIEW PartnerLab9 AS

SELECT \* FROM Lab9 WHERE EnteredBy = 'Partner';

CREATE PROCEDURE AddPartnerLab9 (@Lab9Text char(30)) AS

BEGIN

INSERT INTO Lab9 (Lab9Text, EnteredBy) VALUES (@Lab9Text, 'Partner')

END;

CREATE VIEW MyLab9 AS

SELECT \* FROM Lab9 WHERE EnteredBy = 'Me';

CREATE PROCEDURE AddMyLab9 (@Lab9Text char(30)) AS

BEGIN

INSERT INTO Lab9 (Lab9Text, EnteredBy) VALUES (@Lab9Text, 'Me')

END;

EXEC AddMyLab9 'First Value';

EXEC AddMyLab9 'Second Value';

EXEC AddMyLab9 'Third Value';

EXEC AddMyLab9 'Fourth Value';

--Create a new user

CREATE USER guestuser

from login guestuser;

--Grant access to guestuser

GRANT SELECT ON PartnerLab9 to guestuser;

--Grant procedure access to guestuser

GRANT EXECUTE ON AddPartnerLab9 TO guestuser;

--Revoke SELECT and PROCEDURE accesses to guestuser

REVOKE SELECT ON PartnerLab9 TO guestuser;

REVOKE EXECUTE ON AddPartnerLab9 TO guestuser;

--Grant VIEW access to guestuser

CREATE VIEW MyLab9View AS

SELECT \*

FROM MyLab9;

GRANT SELECT ON MyLab9View to guestuser;

-- This is my guestuser tab!

USE IST659\_Labs;

SELECT \* FROM PartnerLab9;

--Execute Procedure

EXEC AddPartnerLab9 'First Value';

EXEC AddPartnerLab9 'Second Value';

EXEC AddPartnerLab9 'Third Value';

EXEC AddPartnerLab9 'Fourth Value';

--Rerun select

SELECT \* FROM PartnerLab9;

--Delete statement

DELETE Lab9 WHERE EnteredBy = 'Partner';