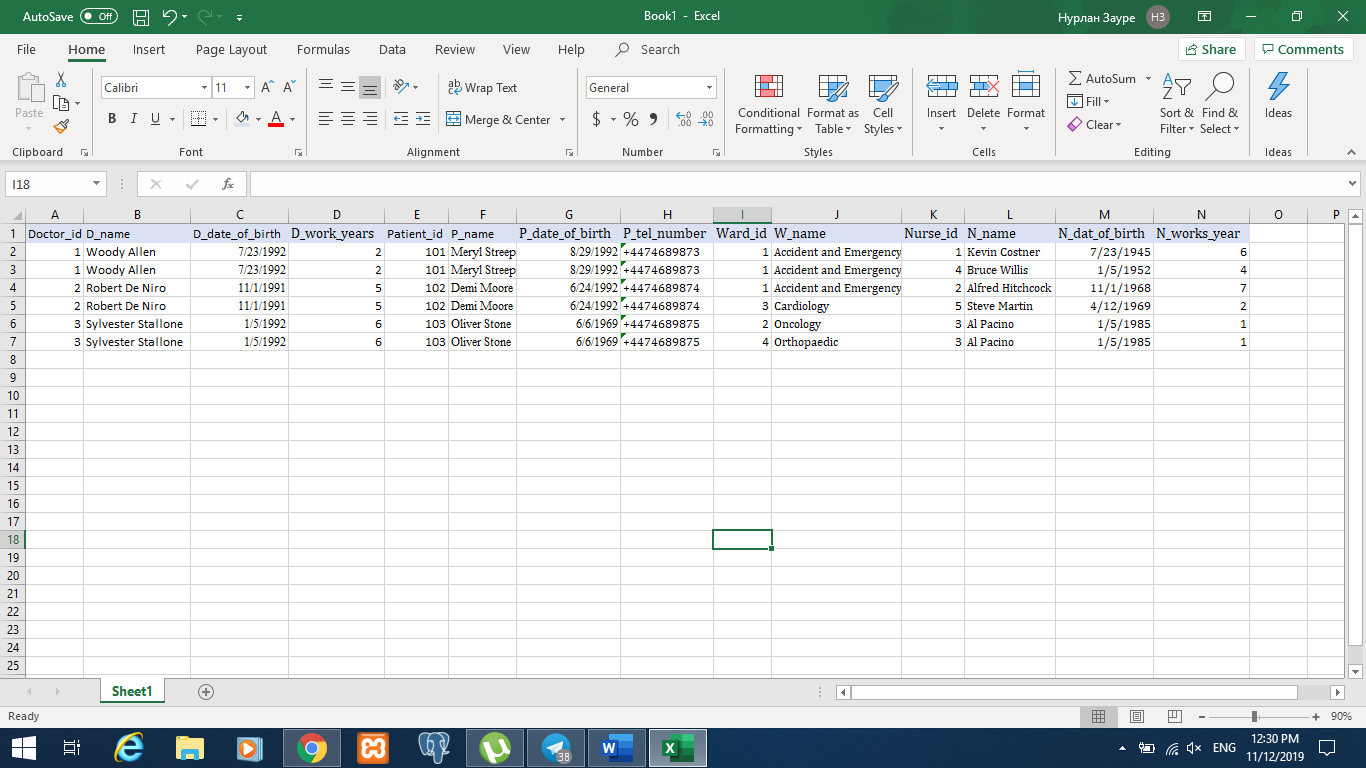
**Part 1**

**1nf**



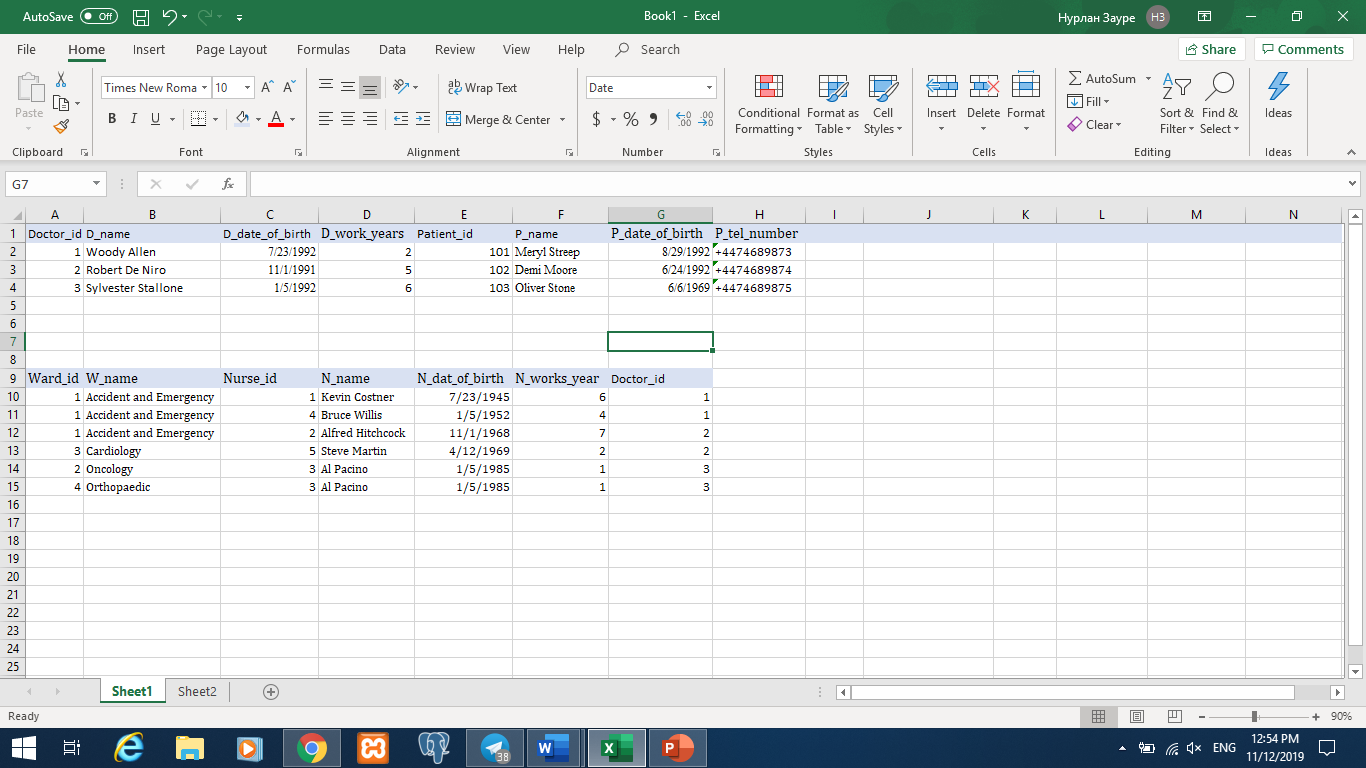
I have made all the data atomic, therefore each row doesn’t consist of more than one value.

Update: we have to change the values for 2 rows in order to change for example, experience in work of a doctor (d\_work\_years).

Insert: we can not add a doctor without his or her birthday

Delete: we have to remove all information in 2 rows in order to delete doctor Robert De Niro.

**2nf**



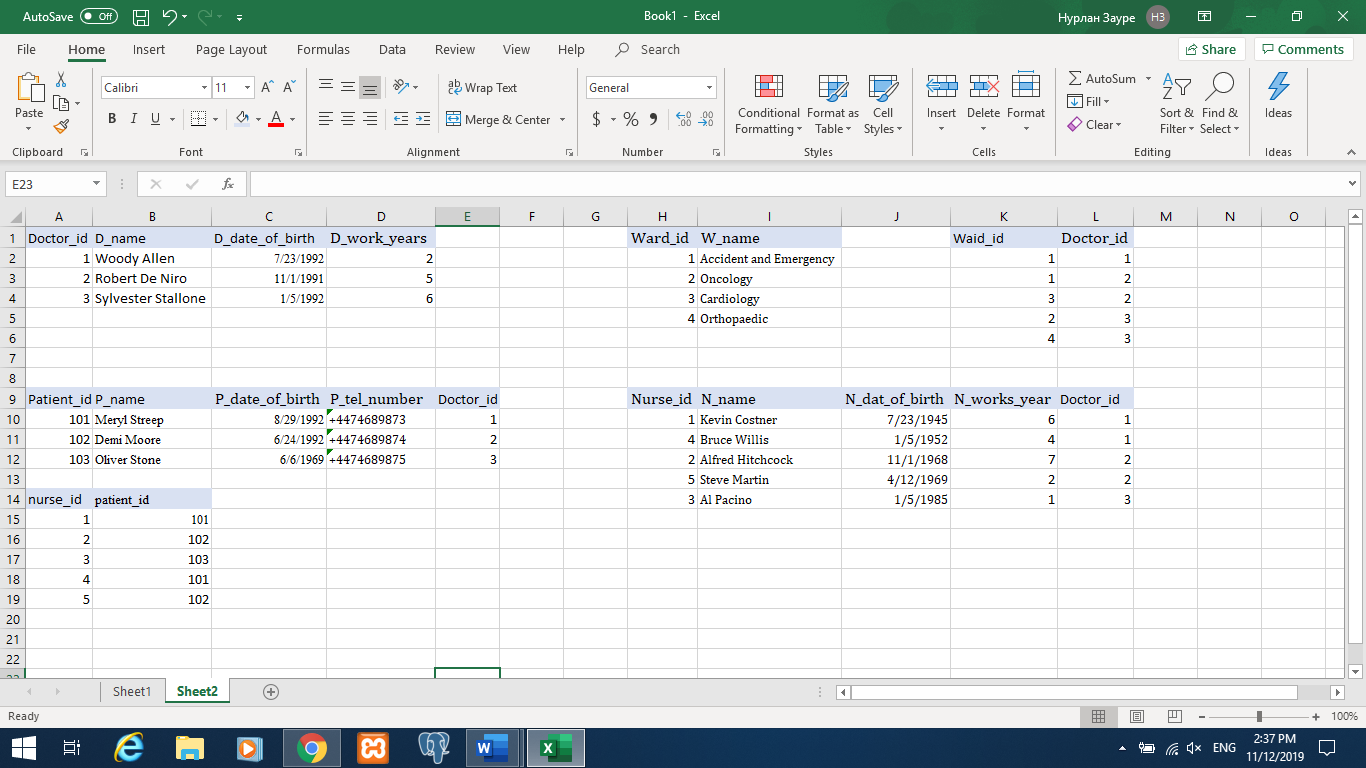
In 2 NF I removed partial dependency.

Update: we need to update 2 rows in ward table in order to change doctor’s information or nurse’s information.

Insert: we can not add a new doctor, for example, if he is practicing one week before starting to work because he doesn’t have any patients.

Delete: we have to remove nurse’s information as well, if we need to remove ward (for example, cardiology, so information about nurse with id=5 needs to be removed.)

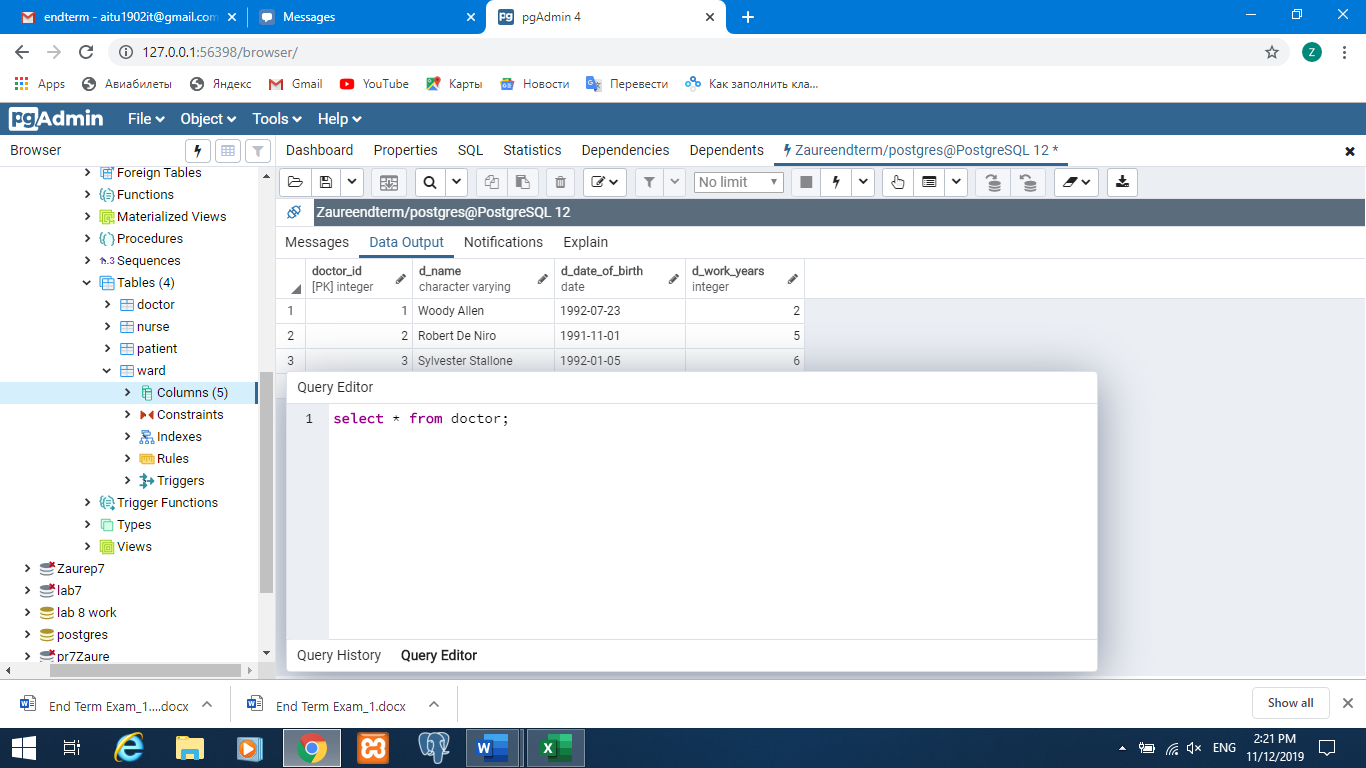
**3nf**

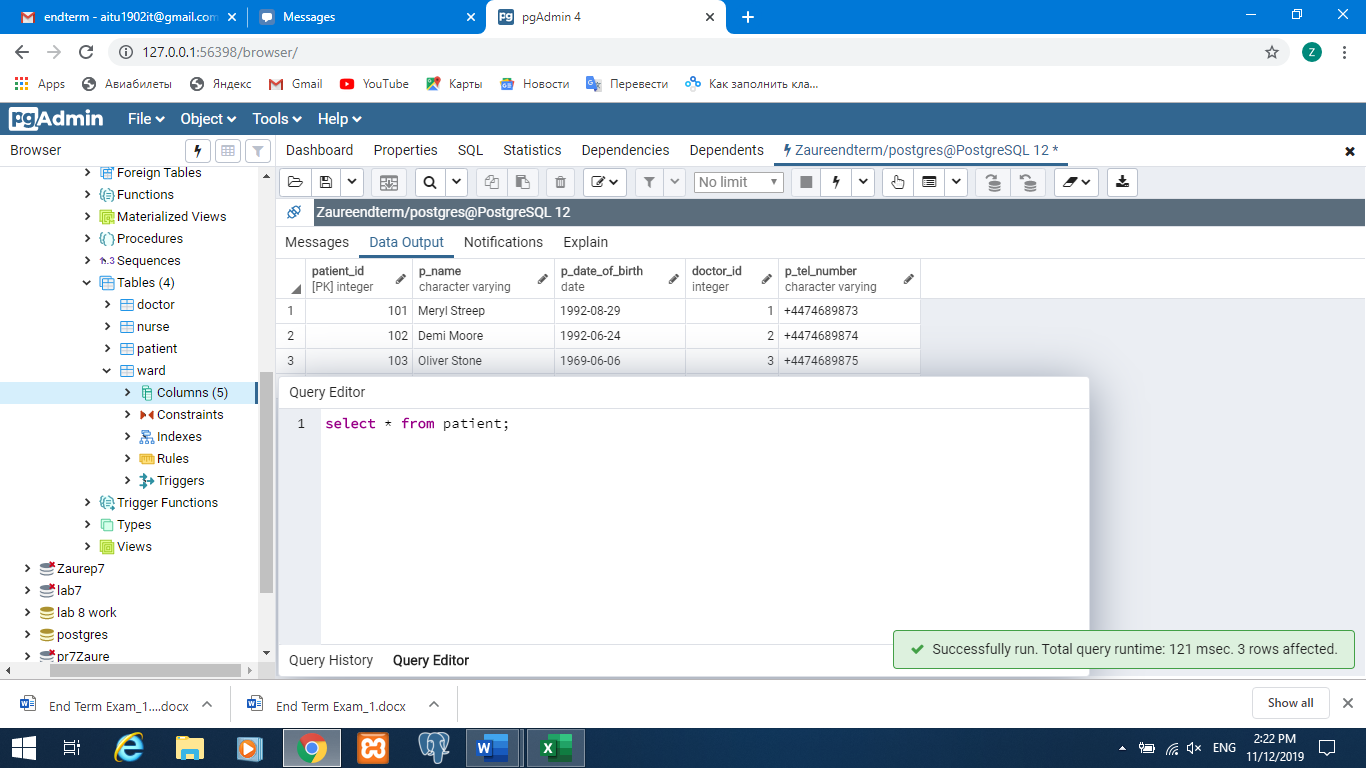


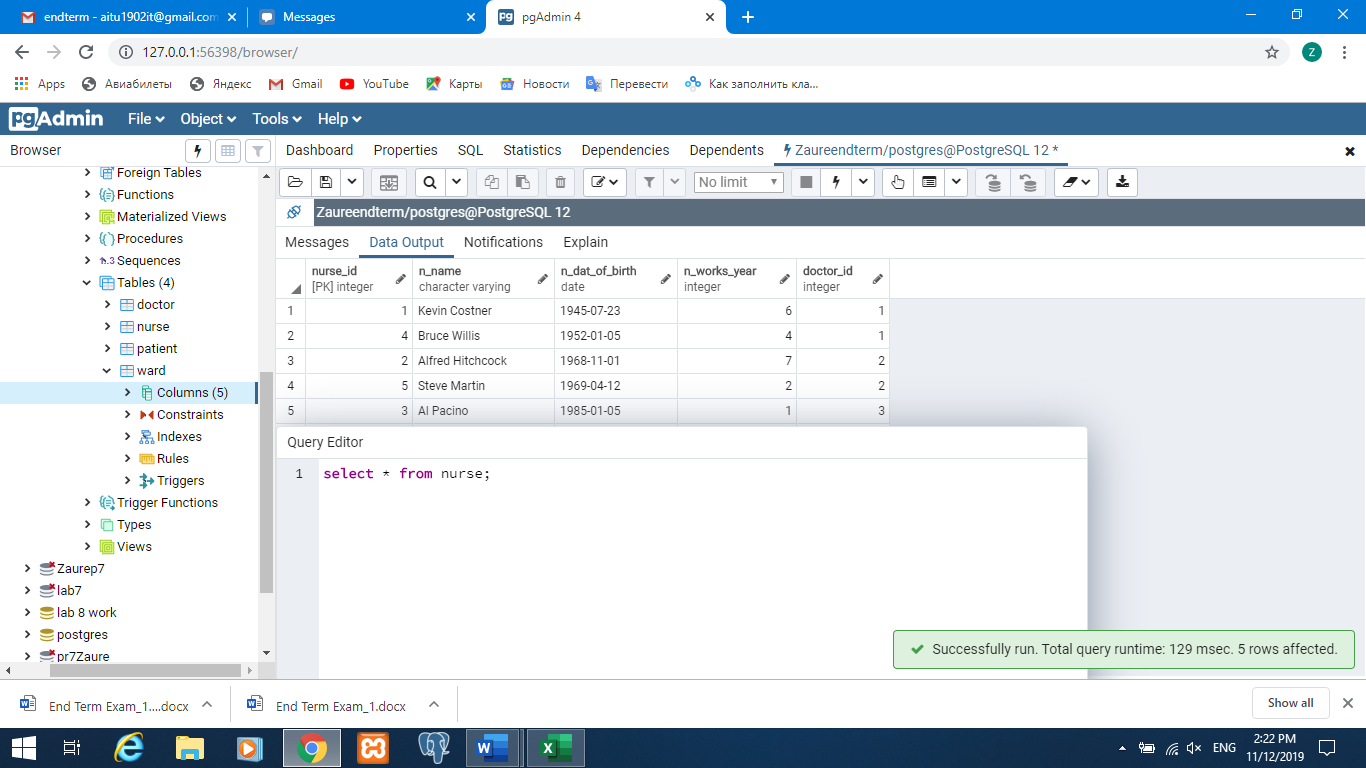
In 3 NF I removed transitive dependency. Therefore, all the data is normalized and there is no data redundancy.

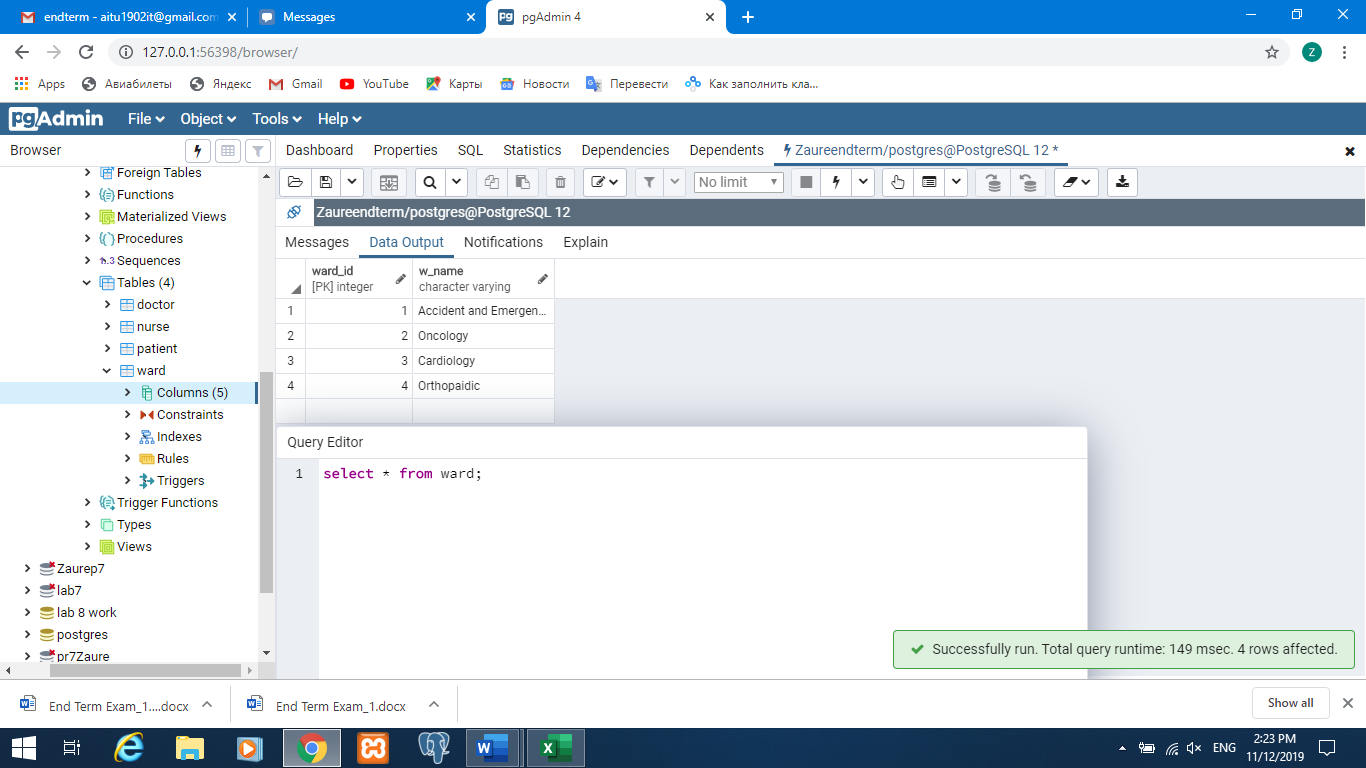
**Part 2**

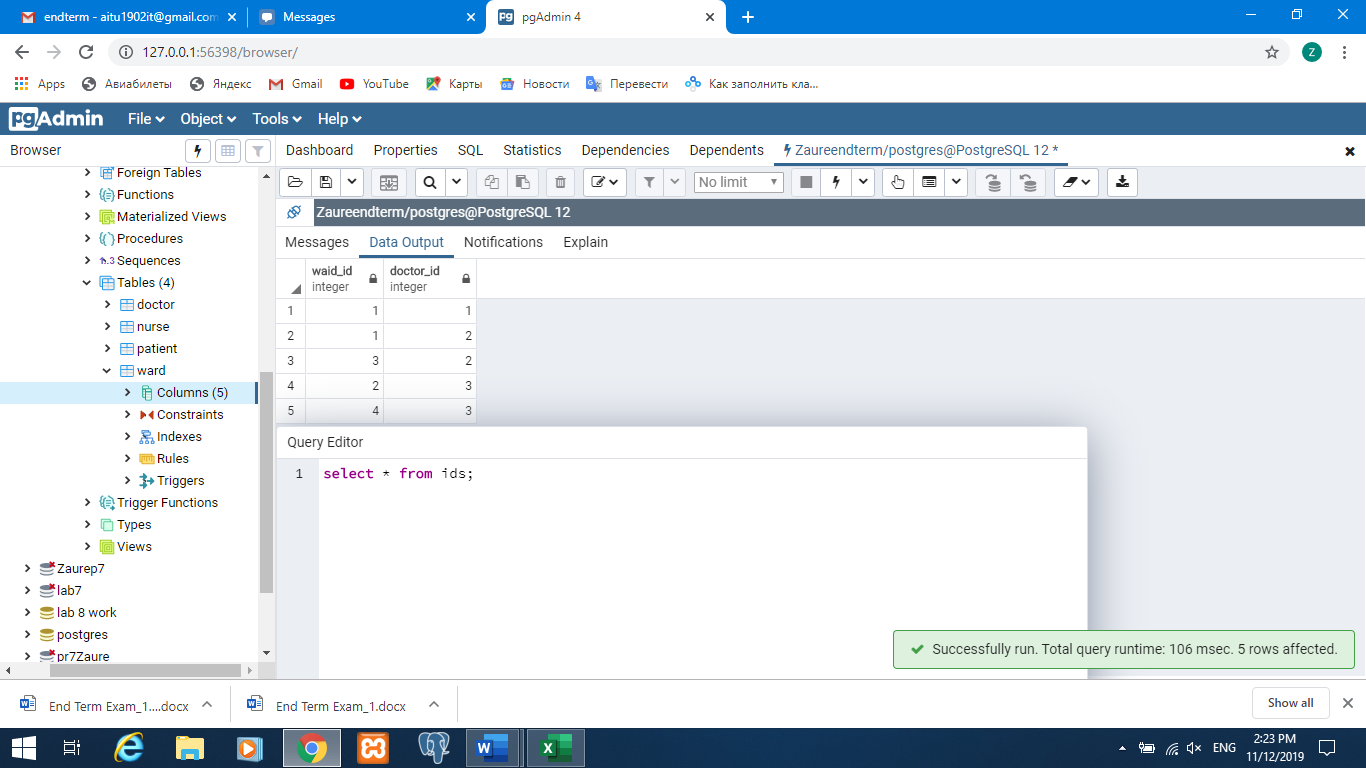
**1)**

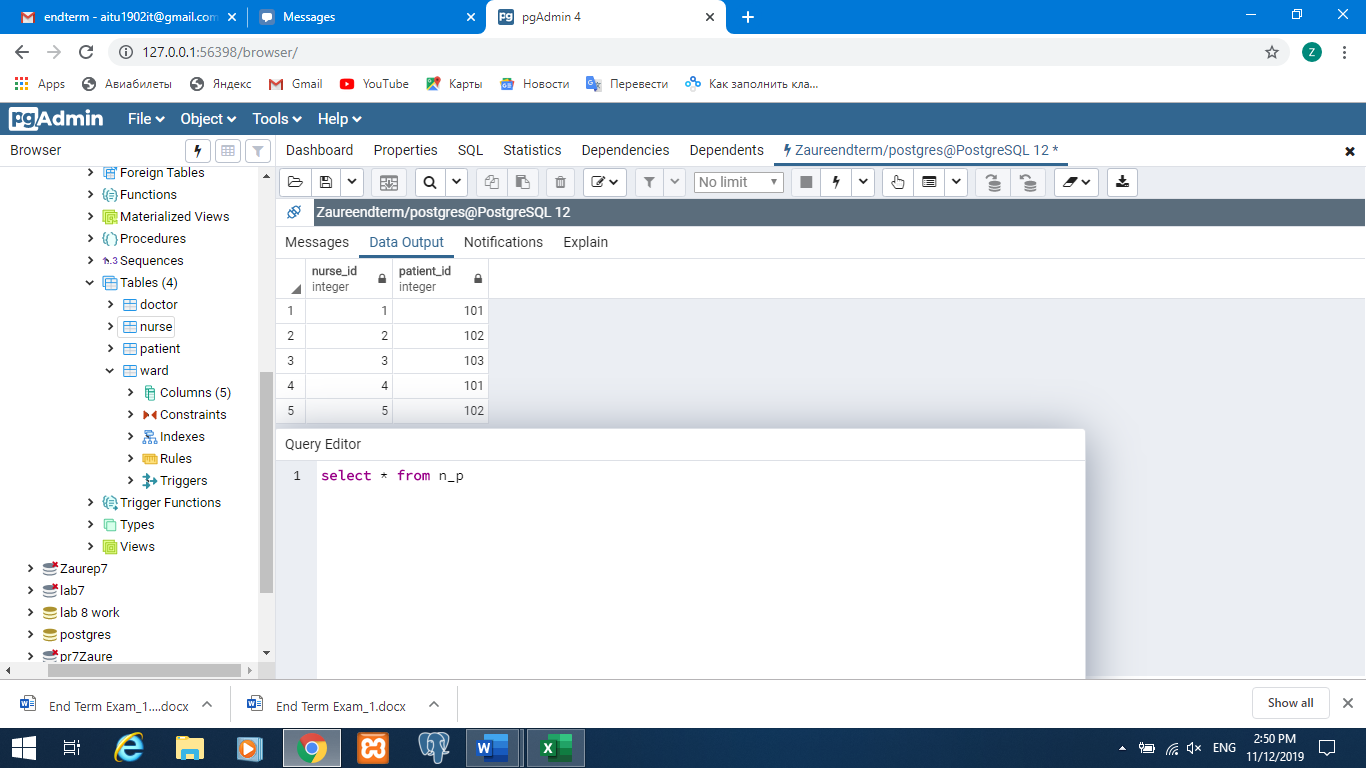




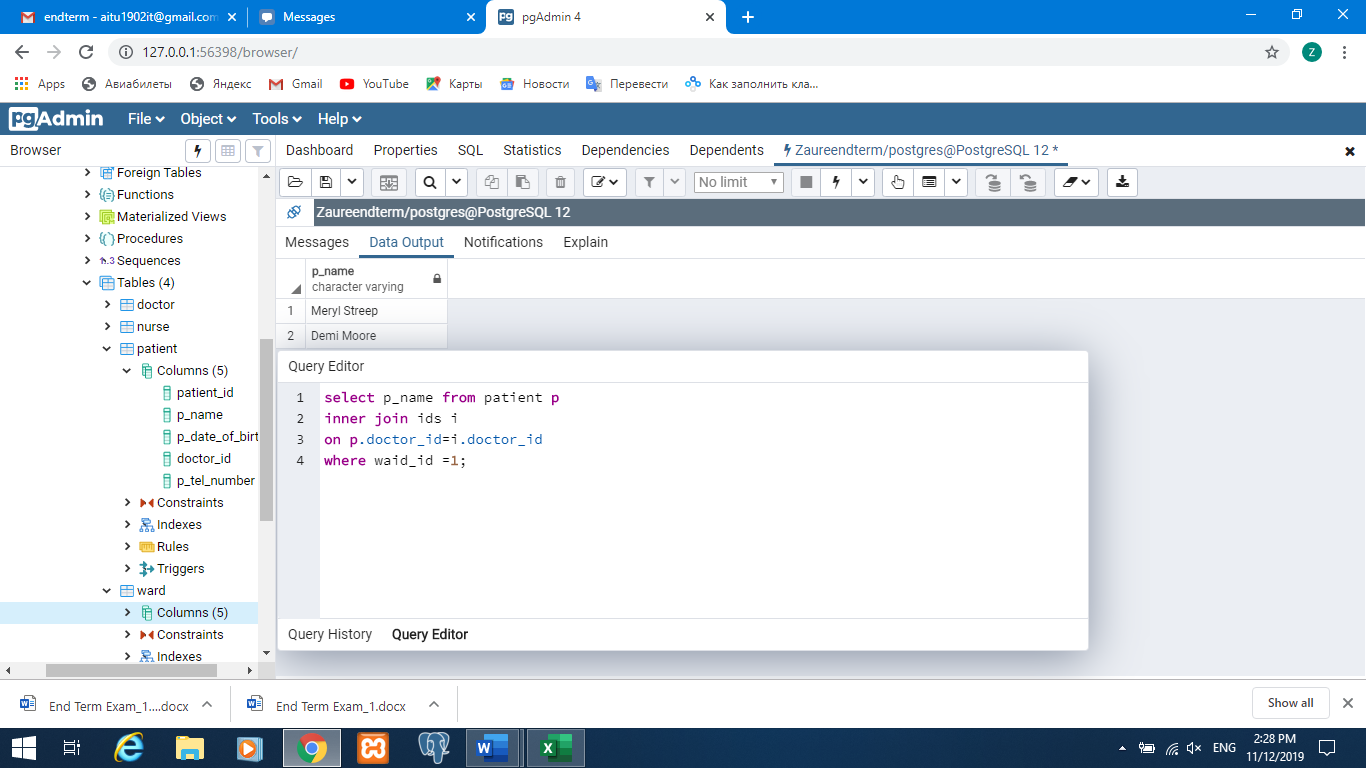




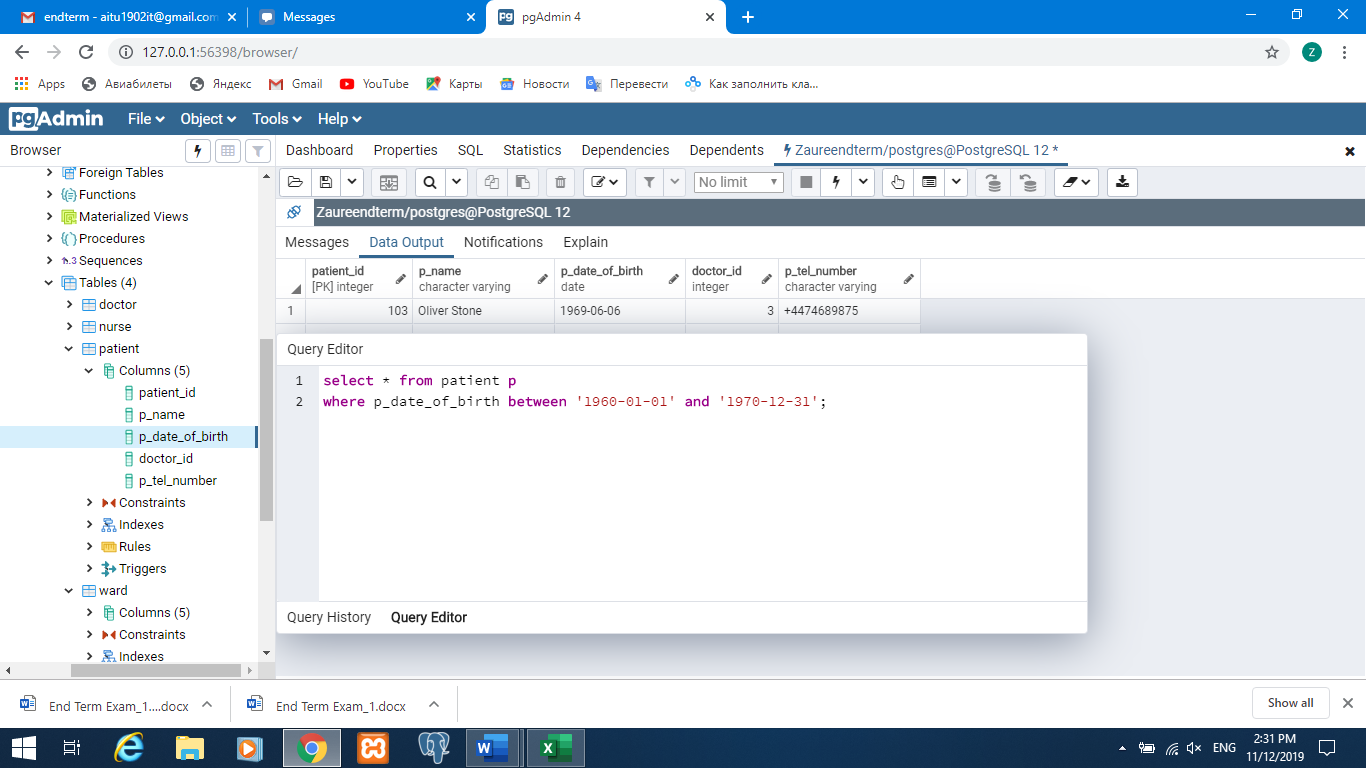




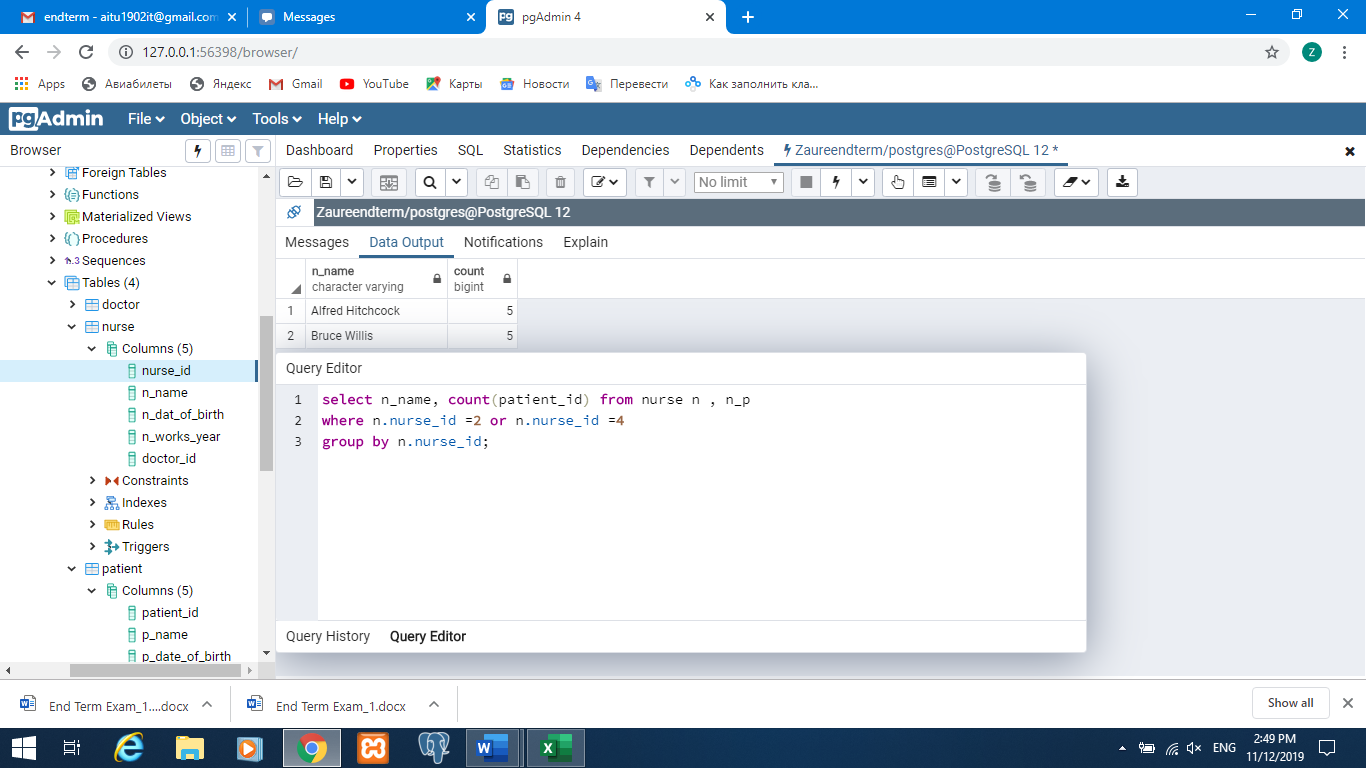
2)



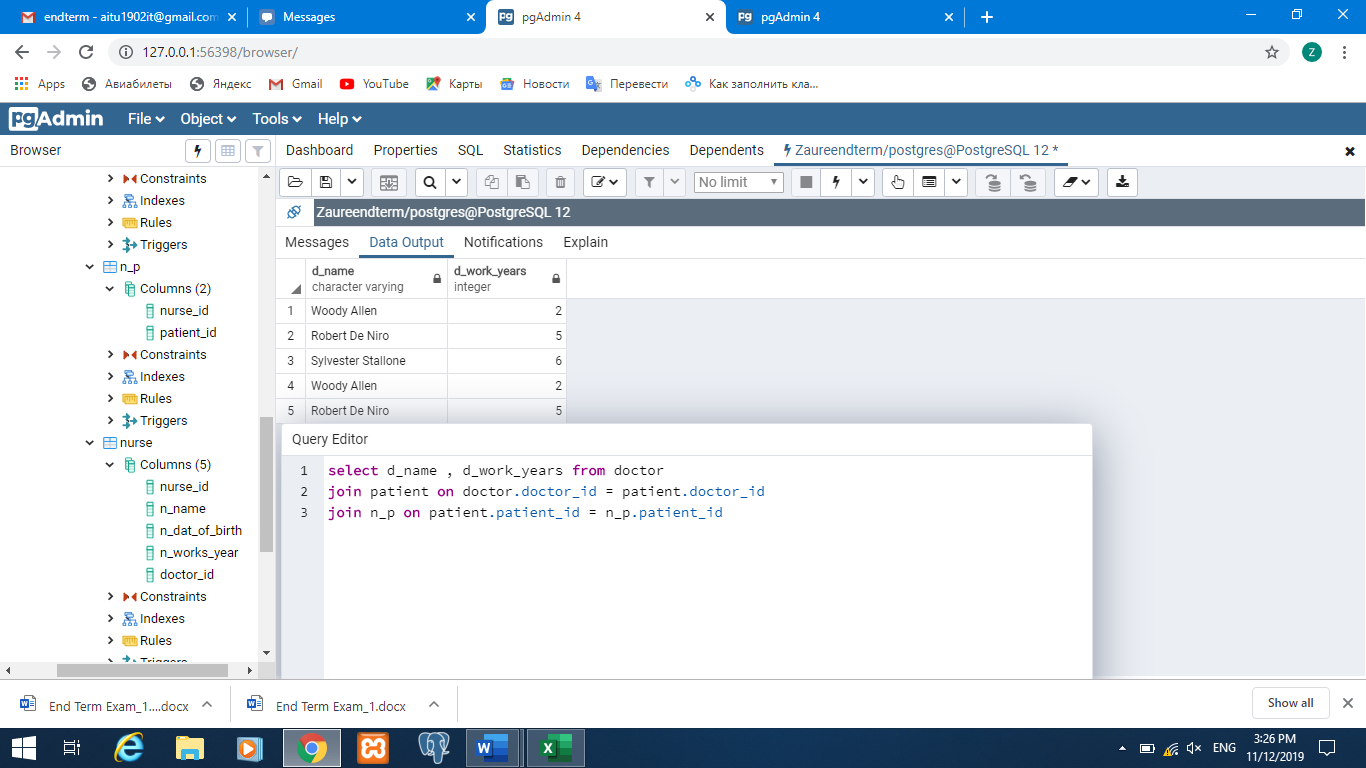
3)



4)



5)



6)

ALTER TABLE doctor

ADD COLUMN Salary integer;

ALTER TABLE nurse

ADD COLUMN Salary integer;

ALTER TABLE nurse

ADD Salary integer

CONSTRAINT sal\_sal DEFAULT 35000

ALTER TABLE doctor

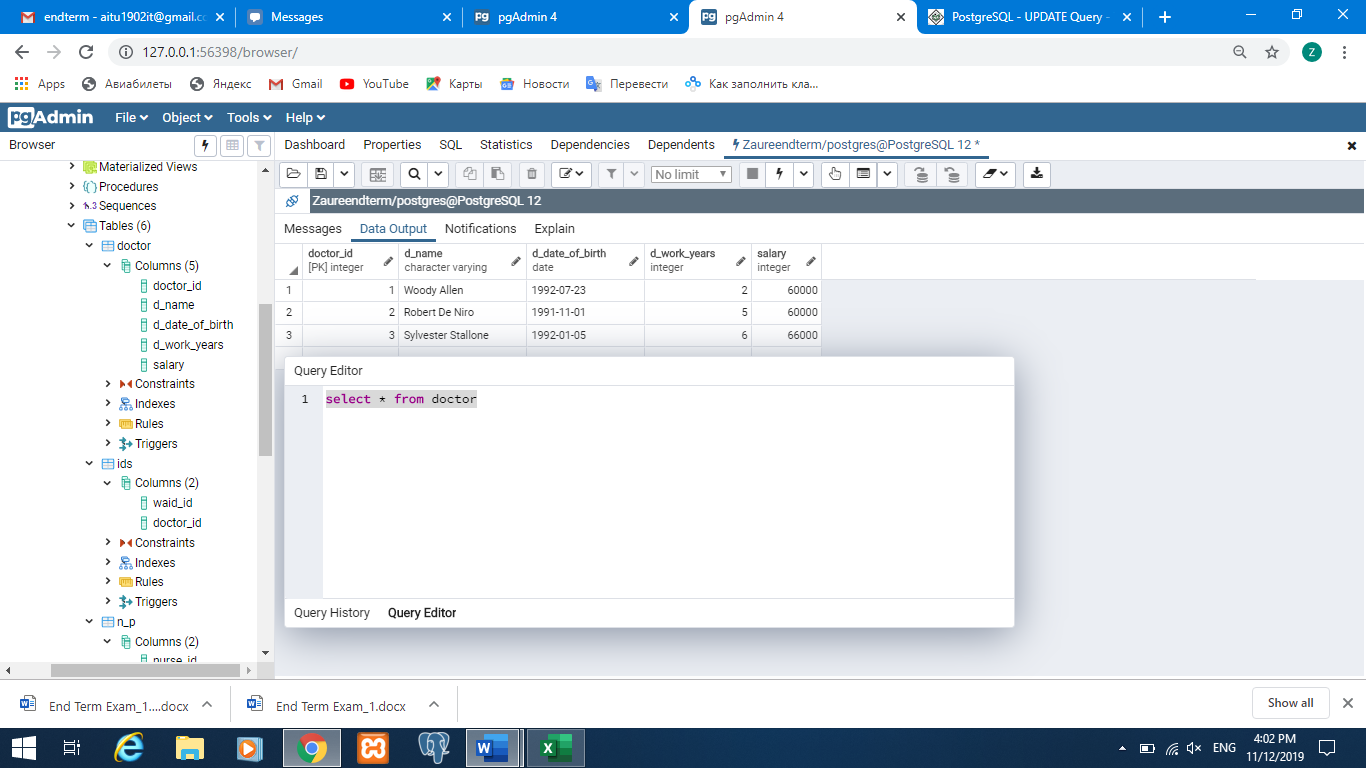
ADD Salary integer

CONSTRAINT sal\_sal DEFAULT 60000

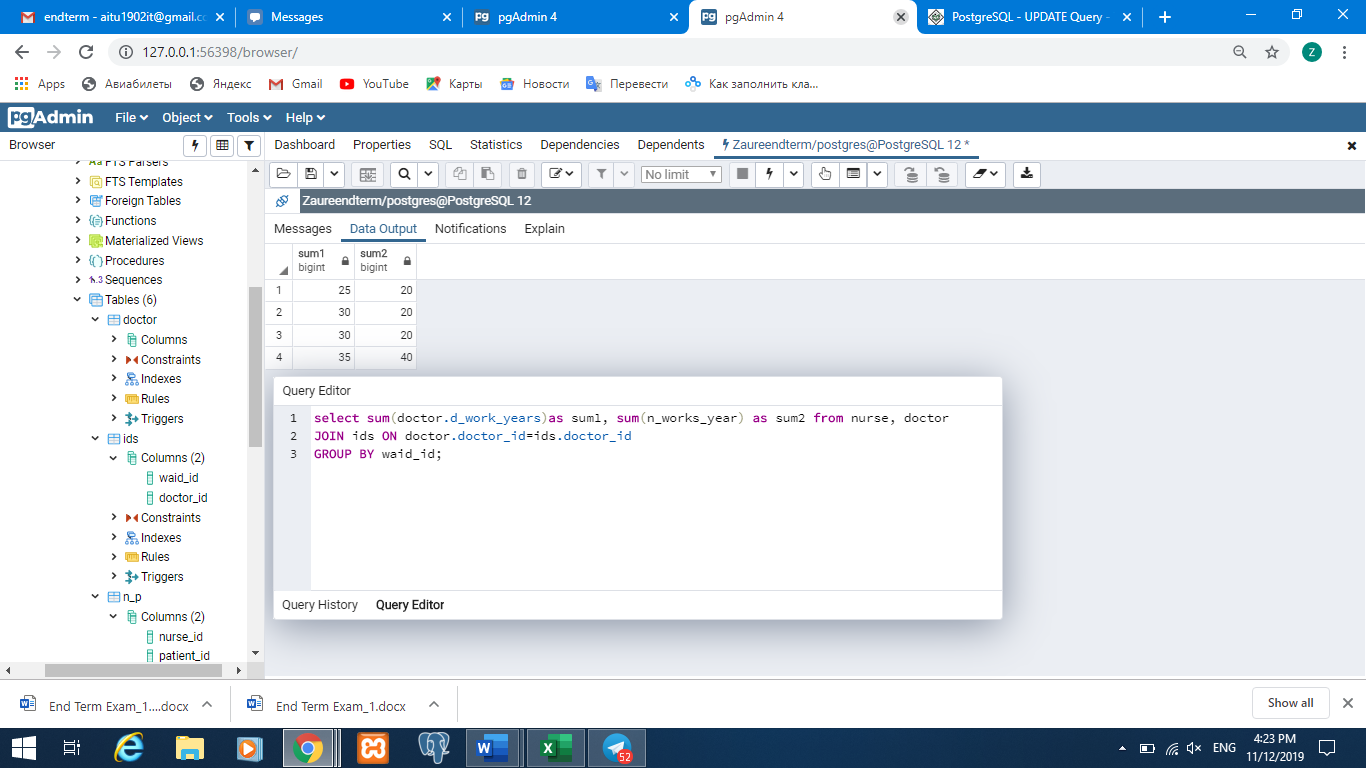
UPDATE doctor SET

salary = salary \* 1.1

WHERE d\_work\_years > 5



7)



8)

