**Lucrarea de laborator N4**

**Declanșatoare în SQL**

1. ***Creați bază de date Persoane formată din următoarele tabele (puteți folosi baza de date de la lucrarea de laborator precedentă):***

Persoane(idPersoana, Numele, Vârsta)

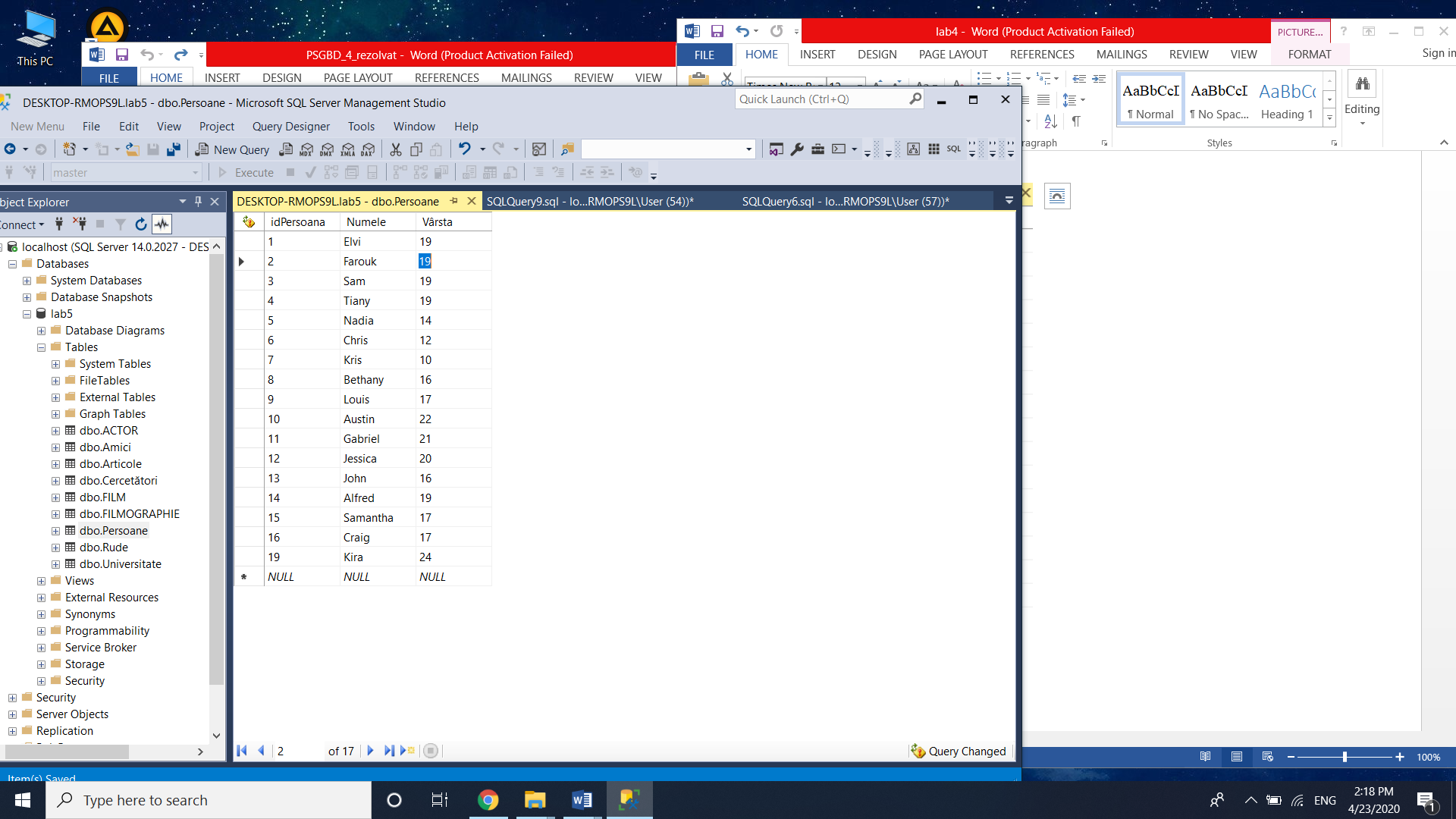
Rude(idPersoana1, idPersoana2)

Amici(idPersoana1, idPersoana2)

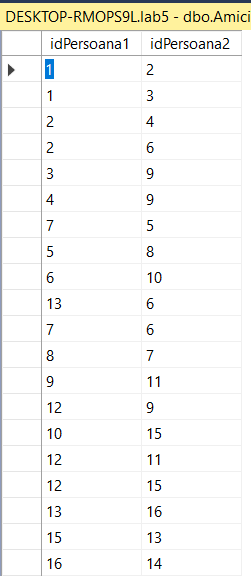
1. ***Completați tabelele cu date folosind scriptul SQL creat în lucrarea nr. 2.***

Trebuie să obțineți următoarea structură.

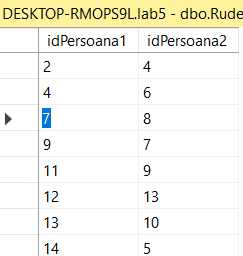
***Tabelul Persoane:***



***Tabelul Amici:***



***Tabelul Rude:***



1. ***Sarcini de realizat:***
2. Să se creeze un declanșator care după adăugarea unei persoane în tabelul Persoane, va adăuga o legătură de prietenie dintre persoana dată și Elvia.

CREATE TRIGGER new\_amicii\_Elvirei

AFTER INSERT ON [dbo].[Persoane] FOR EACH ROW

BEGIN

INSERT INTO [dbo].[Amici](idPersoana1, idPersoana2)

VALUES (NEW.idPersoana, (SELECT [dbo].[Persoane].idPersoana

FROM [dbo].[Persoane]

WHERE [dbo].[Persoane].numele = 'Elvira'));

END

Să se verifice lucrul declanșatorului prin adăugarea unei persoane în tabelul Persoane. Care este efectul declanșatorului?



1. Să se creeze un declanșator care înainte de adăugare a unei persoane în tabelul Persoane, verifică dacă această persoană există deja în tabel. Dacă există deja o persoană cu același nume (ca și cel inserat) să se afișeze un mesaj de eroare „Atenție! Așa persoană există!”. Obs. Se va folosi următoarea structură pentru afișarea mesajului: signal sqlstate '45000' set message\_text = 'MESAJ'

CREATE TRIGGER existPers

BEFORE INSERT ON [dbo].[Persoane]

FOR EACH ROW

BEGIN

IF EXISTS (SELECT \*

FROM [dbo].[Persoane]

WHERE [dbo].[Persoane].Numele = NEW.Numele)

THEN

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Atenție! Așa persoană există';

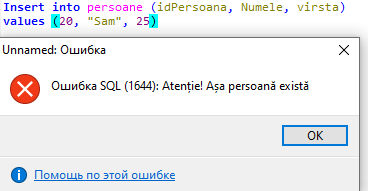
END IF;

END

Să se verifice lucrul declanșatorului prin adăugarea unei persoane în tabelul Persoane cu același nume. Care este efectul declanșatorului?

**Insert** **into** persoane (idPersoana, Numele, vârsta)

**values** (18, "Kris", 22)



cu un nume nou diferit:

**Insert** **into** persoane (idPersoana, Numele, vârsta)

**values** (19, "Kira", 24)

C:\Users\User\AppData\Local\Microsoft\Windows\INetCache\Content.Word\DESKTOP-RMOPS9L.lab5 - dbo.Persoane - Microsoft SQL Server Management Studio 4_23_2020 1_51_03 PM (2).png

1. Să se creeze un declanșator care după inserarea a două persoane în tabelul rude creează o legătură de prietenie dintre aceste două persoane. Dacă inserăm perechea (x, y) în tabelul Rude, în tabelul Amici se va insera aceeași pereche de persoane (x, y). Să se verifice lucrul declanșatorului.

CREATE TRIGGER amici\_rude

BEFORE INSERT ON [dbo].[Rude] FOR EACH ROW

BEGIN

IF EXISTS (SELECT \*

FROM [dbo].[Rude]

WHERE ([dbo].[Rude].idPersoana1 = NEW.idPersoana1 AND [dbo].[Rude].idPersoana2 = NEW.idPersoana2)

OR ([dbo].[Rude].idPersoana1 = NEW.idPersoana2 AND Rude.idPersoana2 = NEW.idPersoana1))

THEN

SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Atentie! Legatura exista';

ELSE

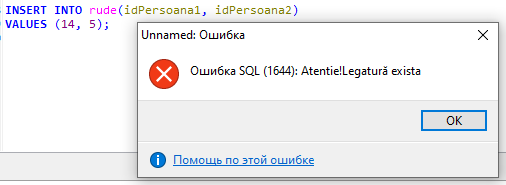
INSERT INTO [dbo].[Amici](idPersoana1, idPersoana2)

VALUES (NEW.idPersoana1, NEW.idPersoana2);

END IF;

END

Dacă perechea introdusă există, se va returna un mesaj de eroare:



1. Să se creeze un declanșator care va elimina legătura de prietenie între două persoane, dacă legătura de rudenie între ele este eliminată. Dacă perechea (x, y) nu mai este în tabelul Rude, atunci perechea dată este eliminată automat din tabelul Amici. Demonstrați lucrul declanșatorului.

CREATE TRIGGER stergere

AFTER DELETE ON [dbo].[Rude] FOR EACH ROW

BEGIN

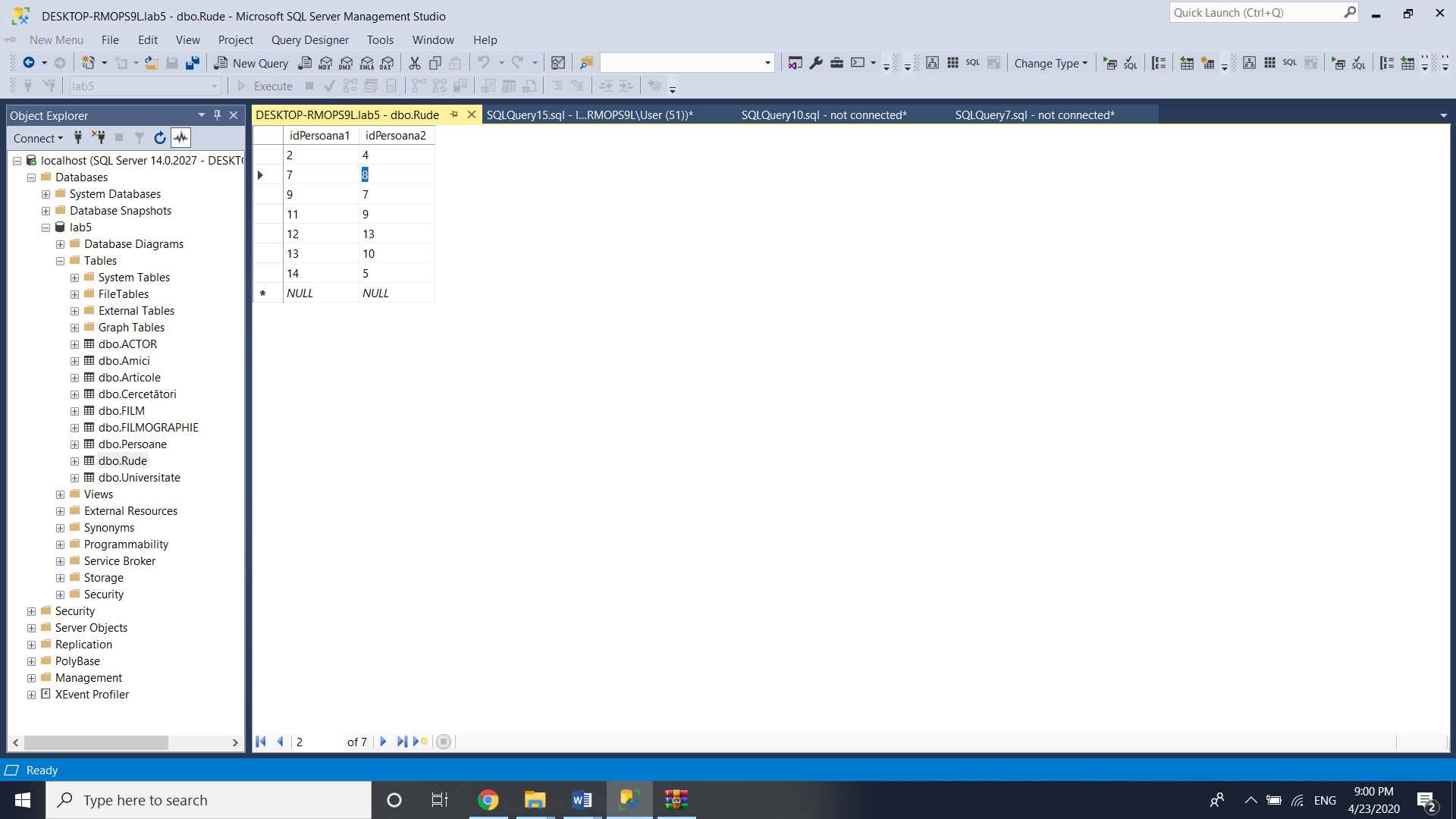
DELETE FROM [dbo].[Amici]

WHERE [dbo].[Amici].idPersoana1 = OLD.idPersoana1 AND [dbo].[Amici].idPersoana2 OLD.idPersoana2;

END

delete from rude

where idPersoana1 = 4 and idPersoana2 = 6



1. Să se creeze un declanșator care ar elimina toate referințele la o persoană din tabelele Rude și Amici, când această persoană este eliminată din tabelul Persoane. Demonstrați lucrul declanșatorului.

CREATE TRIGGER sterge\_rudele

AFTER DELETE ON [dbo].[Persoane] FOR EACH ROW

BEGIN

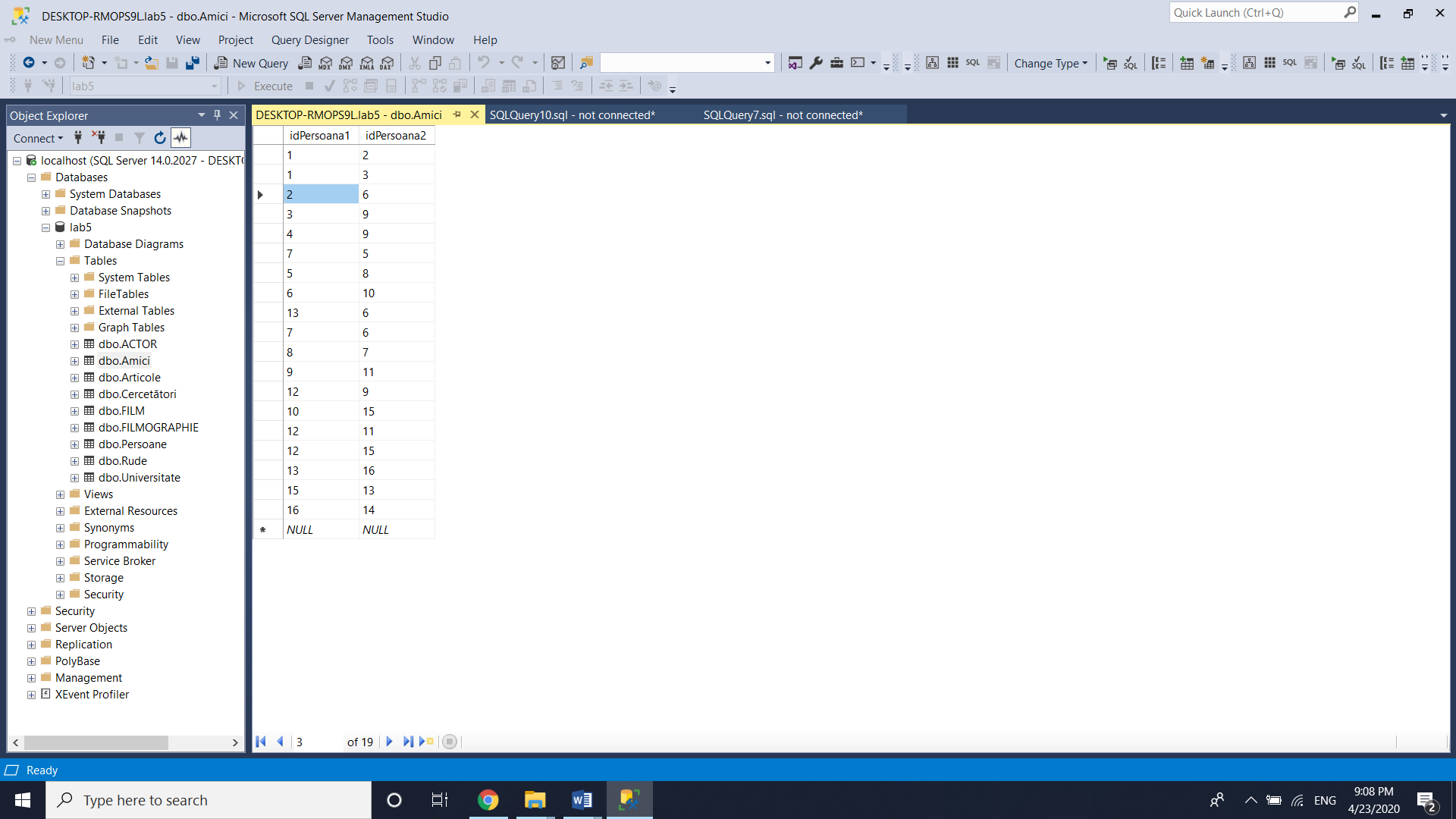
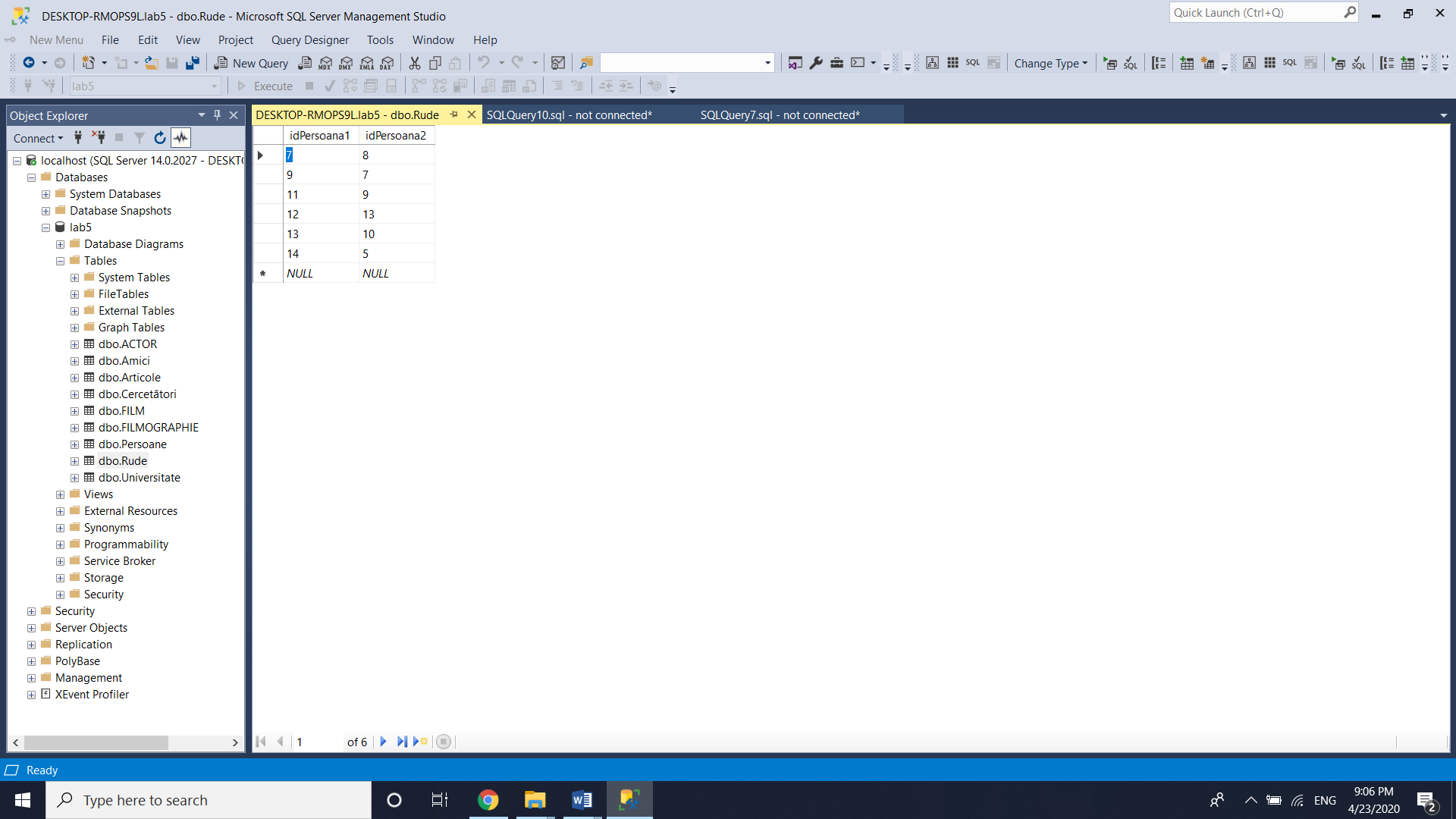
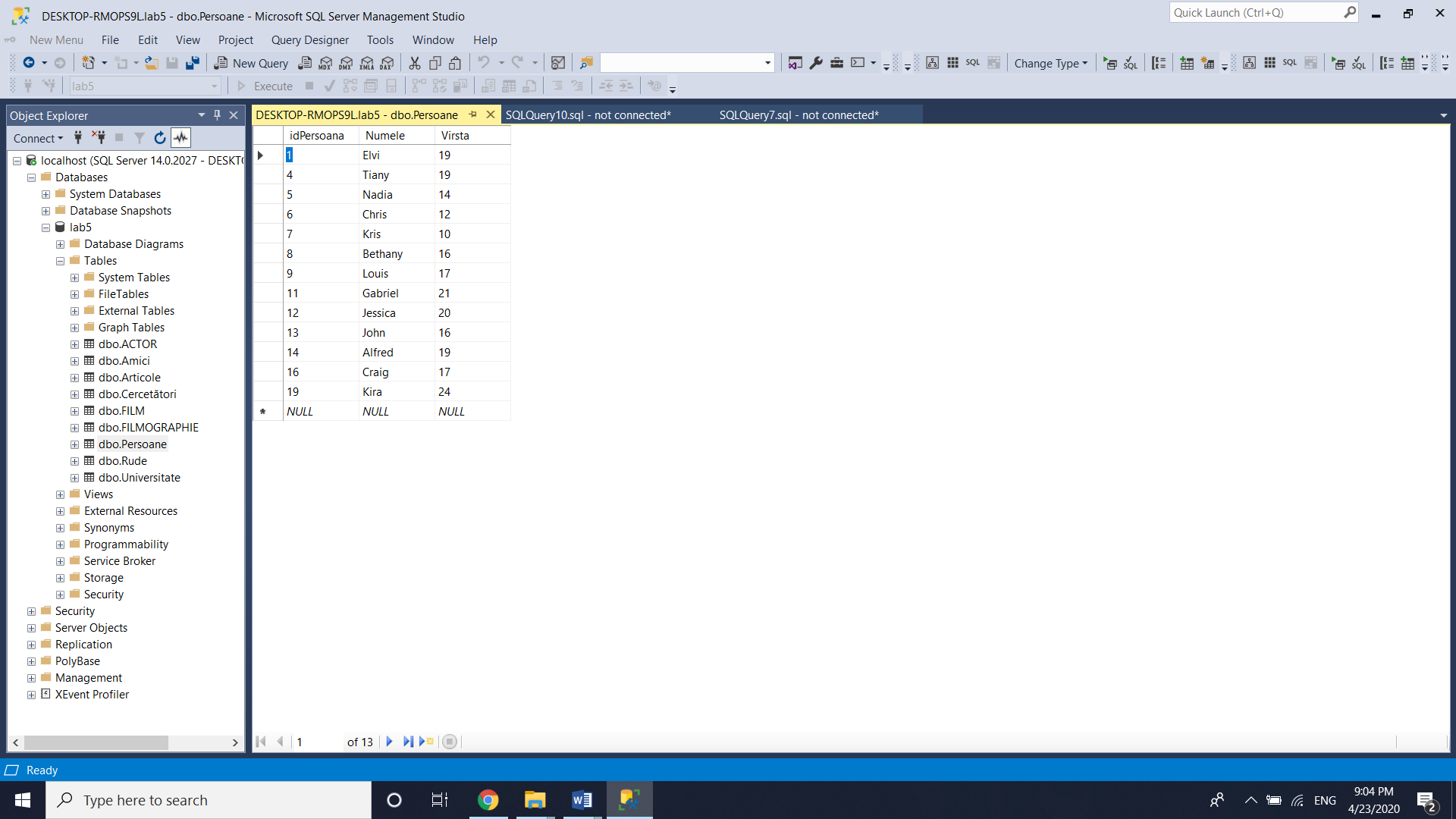
DELETE FROM [dbo].[Amici]

WHERE OLD.idPersoana = [dbo].[Amici].idPersoana1 OR OLD.idPersoana = [dbo].[Amici].idPersoana2;

DELETE FROM [dbo].[Rude]

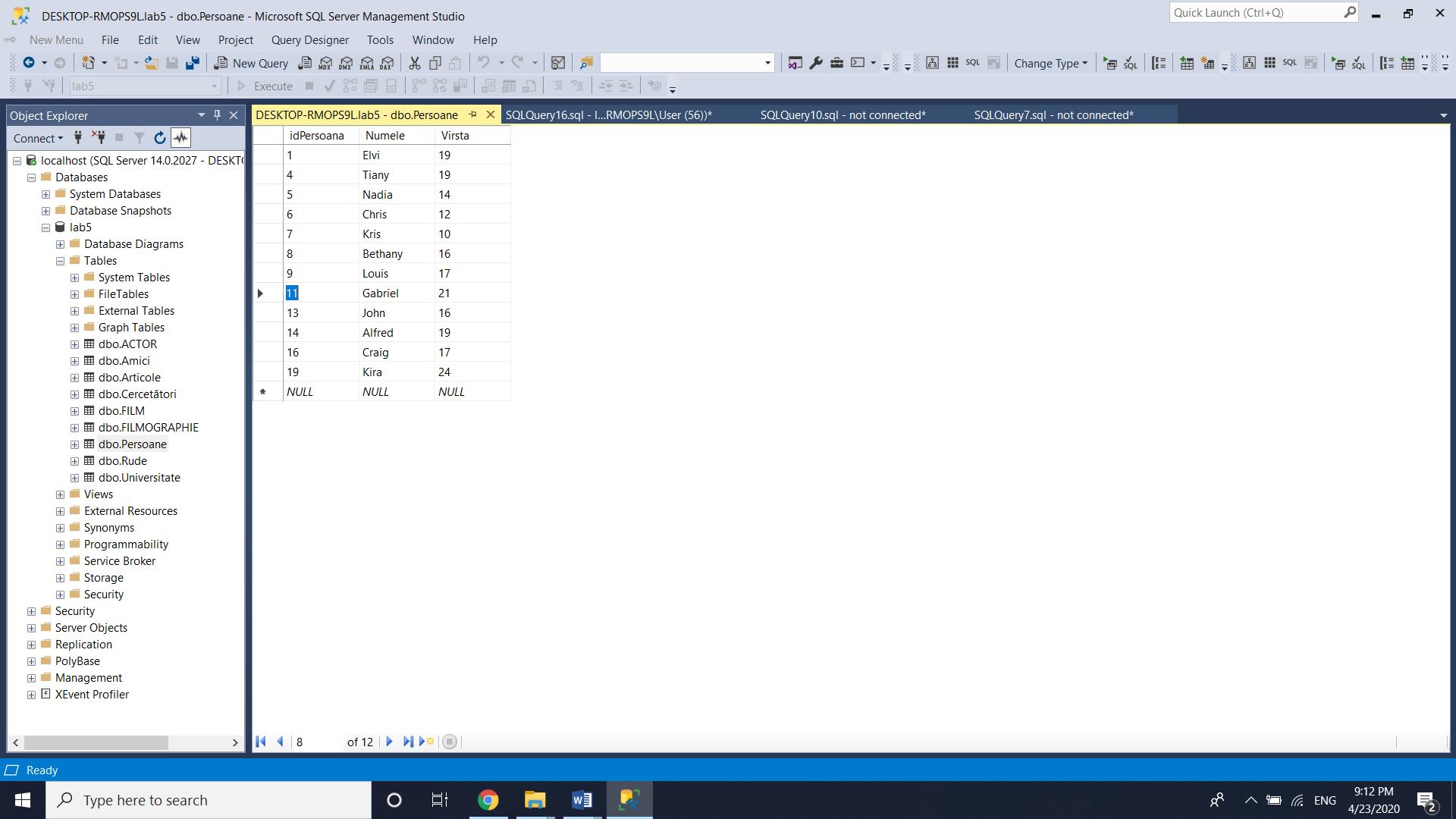
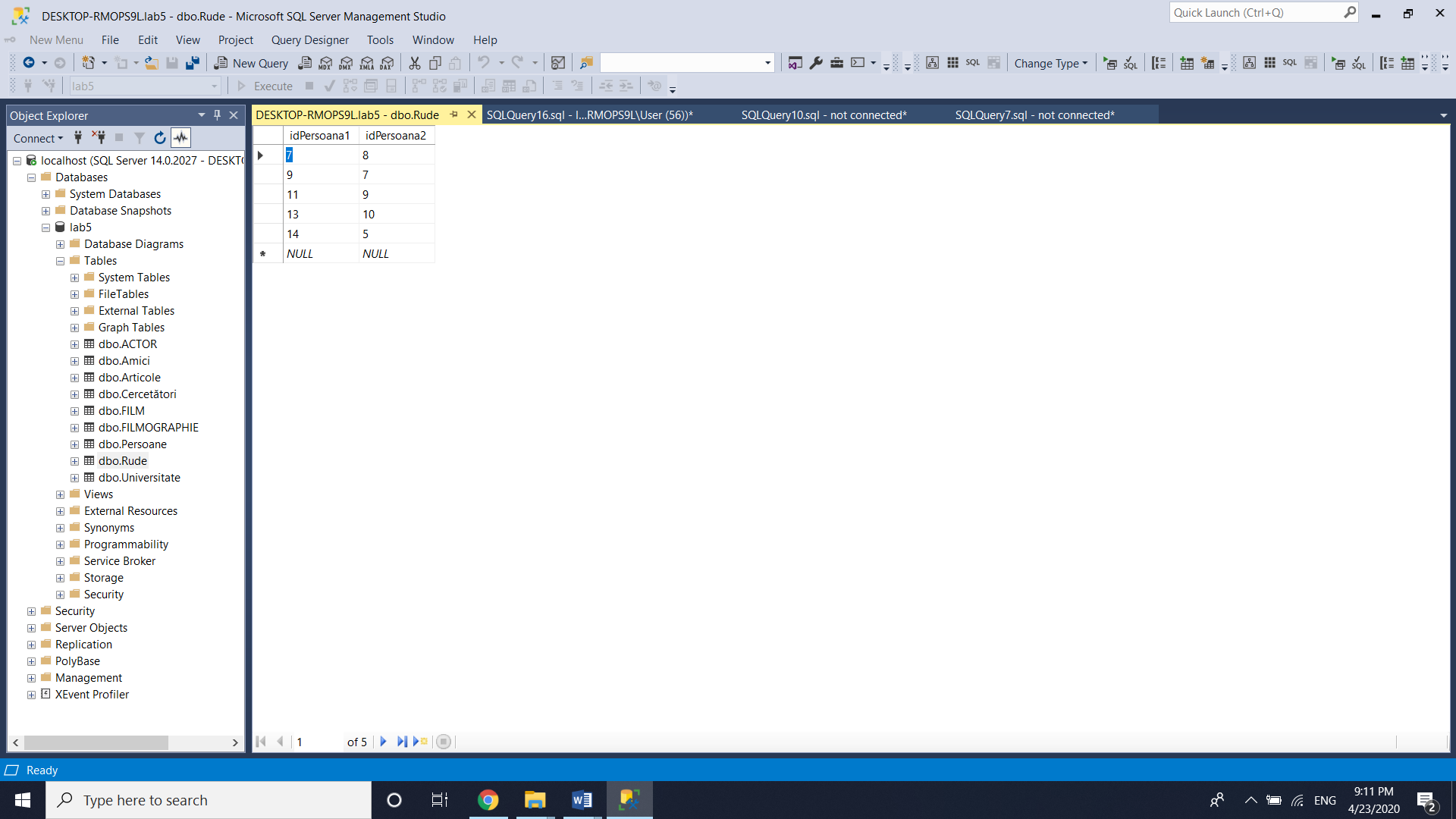
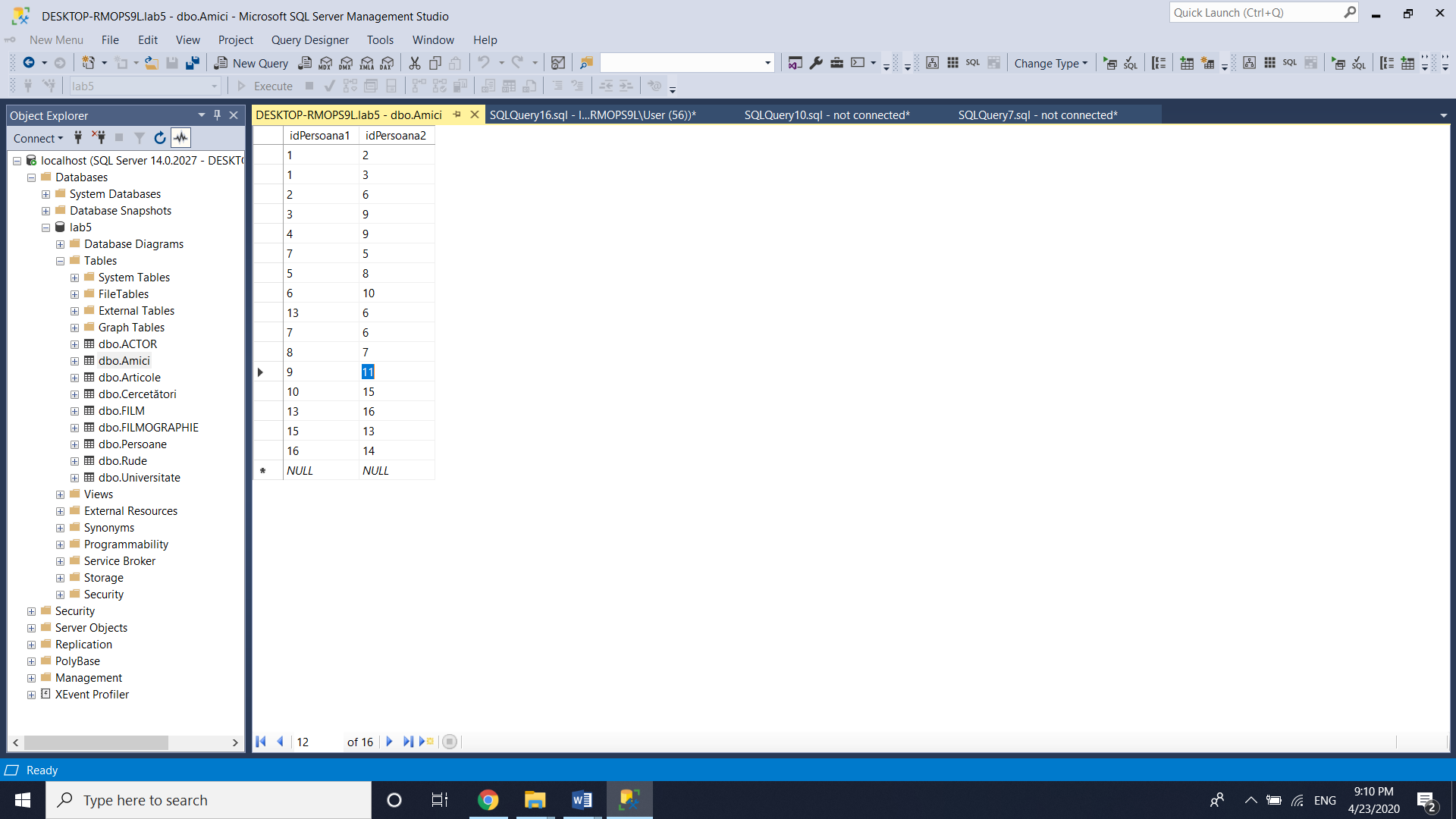
WHERE OLD.idPersoana = [dbo].[Rude].idPersoana1 OR OLD.idPersoana = [dbo].[Rude].idPersoana2;

END



Delete from Persoane

where idPersoana = 12



1. Pentru această sarcină eliminați cheile străine din tabelul Amici. Să se creeze un declanșator care va lucra ca cheile străine eliminate. Să se demonstreze situațiile în care acționează declanșatorul dat.

CREATE TRIGGER elimen\_cheile\_straine

BEFORE INSERT ON [dbo].[Amici] FOR EACH ROW

BEGIN

IF NOT EXISTS (SELECT \*

FROM [dbo].[Persoane]

WHERE [dbo].[Persoane].idPersoana = NEW.idPersoana1 OR [dbo].[Persoane].idPersoana = NEW.idPersoana2)

THEN

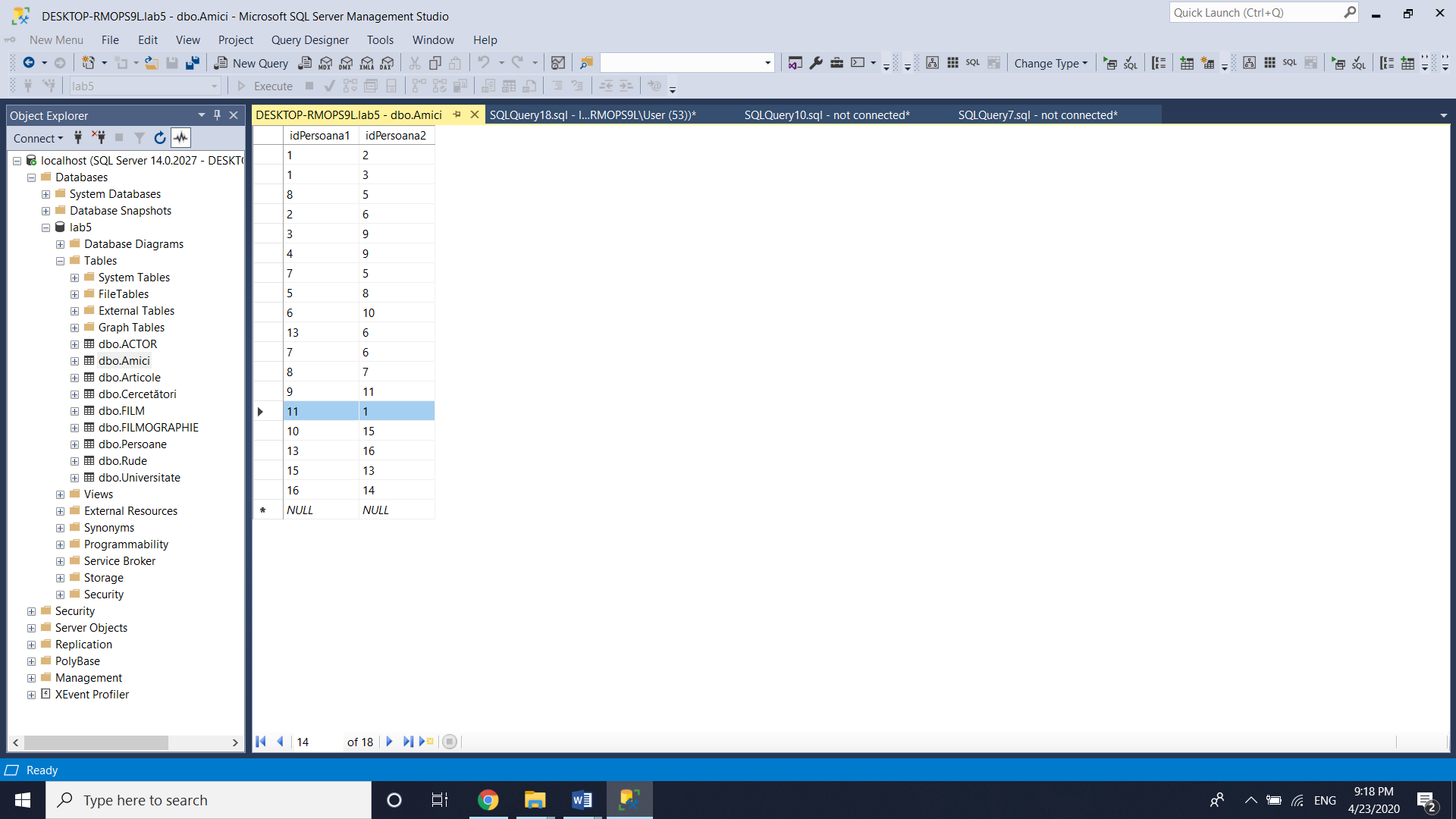
SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = 'Atenție eroare';

END IF;

END Dacă în tabelul Amici se va adăuga o pereche de persoane (x, y), declanșatorul se va asigura că persoanele cu codul x și y există în tabelul Persoane:

Insert into Amici(idPersoana1, idPersoana2)

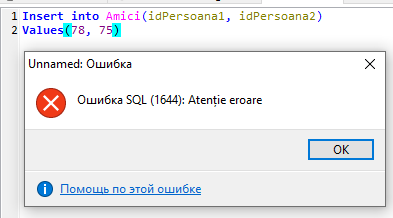
Values(11, 1)



În caz de depistare a încercării de a introduce persoane care nu există în tabelul Persoane se va afișa un mesaj de eroare:

Insert into Amici(idPersoana1, idPersoana2)

Values(21, 111)



1. Să se creeze un declanșator care va implementa modificarea în cascadă a datelor, astfel la modificarea datelor din tabelul Persoane (idPersoana) aceste modificări vor fi propagate în tabelul Amici. Demonstrați lucrul declanșatorului.

CREATE TRIGGER modific\_cascada\_date

AFTER UPDATE ON [dbo].[Persoane] FOR EACH ROW

BEGIN

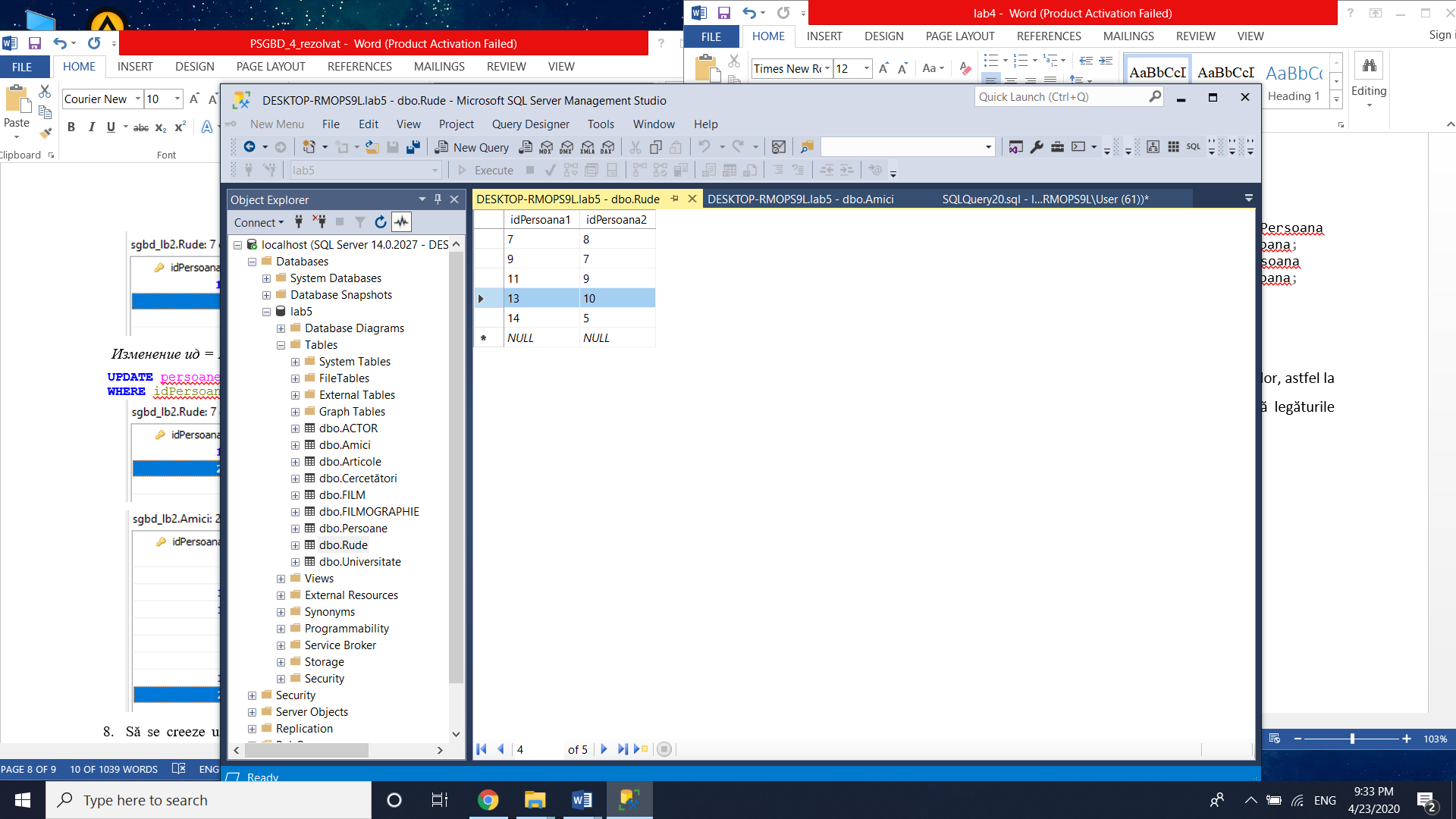
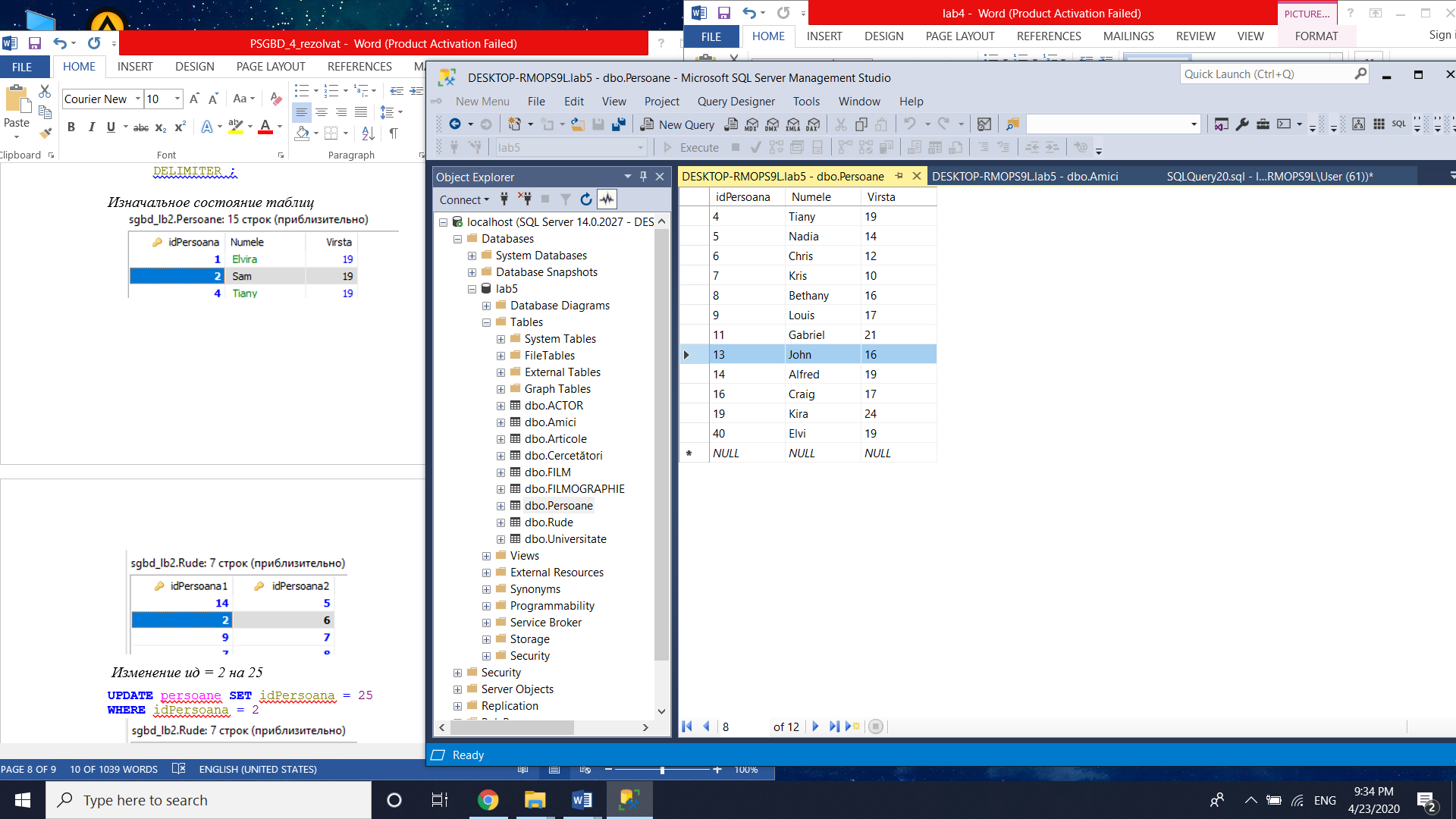
UPDATE [dbo].[Amici] SET [dbo].[Amici].idPersoana1 = NEW.idPersoana

WHERE [dbo].[Amici].idPersoana1 = OLD.idPersoana;

UPDATE [dbo].[Amici] SET [dbo].[Amici].idPersoana2 = NEW.idPersoana

WHERE [dbo].[Amici].idPersoana2 = OLD.idPersoana;

END

UPDATE persoane SET idPersoana = 77

WHERE idPersoana = 13

