

Datenbank

Doç. Dr. Selçuk KIRAN

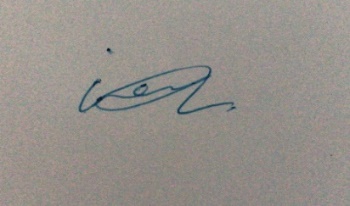


September 2020

İrem Duranlı - 138417038

Caner Ünsal - 138417852

Özgürazad Çelik - 138417056

**Şerefim ve inandığım tüm değerler adına yemin ederim ki bu sınavda hiçbir yerden kopya çekmedim ve sınav kurallarına karşı gelmedim.**

**Ich schwöre im Namen meiner Ehre und all der Werte, von denen ich geglaubt habe, dass ich bei dieser Prüfung gar nichts von irgendwo kopiert und eingefügt habe.**

DROP TABLE PDistribution;

DROP TABLE Lieferung;

DROP TABLE Überprüfung;

DROP TABLE Verkauferbewertung;

DROP TABLE Rückerstattung\_Lieferung;

DROP TABLE Rückerstattung;

DROP TABLE Auftrag\_Detail;

DROP TABLE Zahlung;

DROP TABLE Auftrag;

DROP TABLE Produkt;

DROP TABLE Verkaufer;

DROP TABLE Kaufer;

**CREATE**

CREATE TABLE Kaufer(

Kmitglied\_id int NOT NULL,

Vorname nvarchar(20),

Nachname nvarchar(20),

Email nvarchar(50),

Passwort nvarchar(15),

Mitgliedschaftstatus bit,

Telefonnummer bigint,

CONSTRAINT pk\_Kmitglied\_id PRIMARY KEY (Kmitglied\_id)

)

CREATE TABLE Verkaufer(

Vmitglied\_id int NOT NULL,

Vorname nvarchar(20),

Nachname nvarchar(20),

Email nvarchar(50),

Passwort nvarchar(15),

Telefonnummer bigint,

Betriebname nvarchar(50),

CONSTRAINT pk\_Vmitglied\_id PRIMARY KEY (Vmitglied\_id)

)

CREATE TABLE Produkt(

Produkt\_id bigint NOT NULL,

Pname nvarchar(50),

Gesamtstock int,

Preis money,

Beschreibung nvarchar(200),

CONSTRAINT pk\_produkt\_id PRIMARY KEY (Produkt\_id)

)

CREATE TABLE Lieferung(

Lieferung\_id bigint NOT NULL,

Kurier nvarchar(15),

Strasse nvarchar(50),

PLZ int,

Stadt varchar(15),

Lief\_datum datetime,

CONSTRAINT pk\_lieferung\_id PRIMARY KEY (Lieferung\_id)

)

CREATE TABLE Auftrag(

Auftrag\_id bigint NOT NULL,

Gesamtkosten money,

Mitglied\_id int NOT NULL,

CONSTRAINT pk\_auftrag\_id PRIMARY KEY (Auftrag\_id),

CONSTRAINT fk\_mitglied\_id FOREIGN KEY (Mitglied\_id) REFERENCES Kaufer(Kmitglied\_id)

)

CREATE TABLE Zahlung(

Zahlung\_id bigint NOT NULL,

Auftrag\_id bigint NOT NULL,

Geschenkkarte bit,

Rabattcode nvarchar(10),

Attributename int,

Kreditkartenummer char(16),

Besitzervorname nvarchar(20),

Besitzernachname nvarchar(20),

Kartendatum datetime,

Zahldatum datetime,

Rech\_strasse nvarchar(50),

Rech\_PLZ int,

Rech\_stadt varchar(15),

CVC smallint,

CONSTRAINT pk\_zahlung\_id PRIMARY KEY (Zahlung\_id),

CONSTRAINT fk\_auftrag\_id FOREIGN KEY (Auftrag\_id) REFERENCES Auftrag(Auftrag\_id)

)

CREATE TABLE Auftrag\_Detail(

Produkt\_id bigint NOT NULL,

Auftrag\_id bigint NOT NULL,

Menge tinyint,

Stückpreis money,

CONSTRAINT pk\_auftragdetailprodukt\_id PRIMARY KEY (Produkt\_id, Auftrag\_id),

CONSTRAINT fk\_produkt\_id FOREIGN KEY (Produkt\_id) REFERENCES Produkt(Produkt\_id),

CONSTRAINT fk\_Auftragdetail\_id FOREIGN KEY (Auftrag\_id) REFERENCES Auftrag(Auftrag\_id)

)

CREATE TABLE Rückerstattung(

Rück\_id int NOT NULL,

Mitglied\_id int NOT NULL,

Auftrag\_id bigint NOT NULL,

Rück\_menge tinyint,

Rückpreis money,

CONSTRAINT pk\_rück\_id PRIMARY KEY (Rück\_id),

CONSTRAINT fk\_rückmitglied\_id FOREIGN KEY (Mitglied\_id) REFERENCES Kaufer(Kmitglied\_id),

CONSTRAINT fk\_rückauftrag\_id FOREIGN KEY (Auftrag\_id) REFERENCES Auftrag(Auftrag\_id)

)

CREATE TABLE Rückerstattung\_Lieferung(

Mitglied\_id int NOT NULL,

Rück\_id int NOT NULL,

CONSTRAINT pk\_mitglied\_id PRIMARY KEY (Mitglied\_id, Rück\_id),

CONSTRAINT fk\_rücklieferungmitglied\_id FOREIGN KEY (Mitglied\_id) REFERENCES Verkaufer(Vmitglied\_id),

CONSTRAINT fk\_rück\_id FOREIGN KEY (Rück\_id) REFERENCES Rückerstattung(Rück\_id)

)

CREATE TABLE Verkauferbewertung(

Verkauferbewertung\_id bigint NOT NULL,

Mitglied\_id int NOT NULL,

Qualität decimal(2,1),

Rücklaufequote decimal(2,1),

Zeitsicherung decimal(2,1),

CONSTRAINT pk\_verkauferbewertung\_id PRIMARY KEY (Verkauferbewertung\_id),

CONSTRAINT fk\_verkauferbewertungmitglied\_id FOREIGN KEY (Mitglied\_id) REFERENCES Verkaufer(Vmitglied\_id)

)

CREATE TABLE Überprüfung(

Überprüfung\_id bigint NOT NULL,

Produkt\_id bigint NOT NULL,

Mitglied\_id int NOT NULL,

Bewertung tinyint,

CONSTRAINT pk\_überprüfung\_id PRIMARY KEY (Überprüfung\_id, Produkt\_id),

CONSTRAINT fk\_überprüfungprodukt\_id FOREIGN KEY (Produkt\_id) REFERENCES Produkt(Produkt\_id),

CONSTRAINT fk\_überprüfungmitglied\_id FOREIGN KEY (Mitglied\_id) REFERENCES Kaufer(Kmitglied\_id)

)

CREATE TABLE PDistribution(

Mitglied\_id int NOT NULL,

Lieferung\_id bigint NOT NULL,

Produkt\_id bigint NOT NULL,

CONSTRAINT pk\_distributionmitglied\_id PRIMARY KEY (Mitglied\_id, Lieferung\_id, Produkt\_id),

CONSTRAINT fk\_distributionmitglied\_id FOREIGN KEY (Mitglied\_id) REFERENCES Verkaufer(Vmitglied\_id),

CONSTRAINT fk\_distributionlieferung\_id FOREIGN KEY (Lieferung\_id) REFERENCES Lieferung(Lieferung\_id),

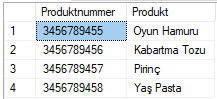
CONSTRAINT fk\_distributionprodukt\_id FOREIGN KEY (Produkt\_id) REFERENCES Produkt(Produkt\_id),

)

**SELECT**

1) SELECT Produkt\_id as Produktnummer, Pname as Produkt FROM Produkt WHERE Gesamtstock < 30;

(Zeigt Produkte mit weniger als 30 auf Lager)



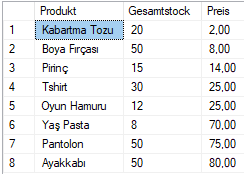
2) SELECT AVG (Gesamtkosten) AS Durchschnitt FROM Auftrag;

(Zeigt den Durchschnittspreis der aufgegebenen Bestellungen an)



3) SELECT PName as Produkt,Gesamtstock,Preis FROM Produkt ORDER BY Preis;

(Zeigt die Lagermenge der Produkte an und sortiert sie nach ihren Preisen)

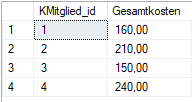


4) SELECT Vorname,Nachname,Telefonnummer FROM Kaufer WHERE Vorname LIKE 'C%';

(Zeigt die angeforderten Informationen des Kunden an, dessen Name mit C beginnt)

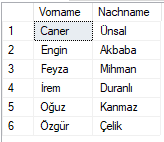


5) SELECT KMitglied\_id,Gesamtkosten FROM Auftrag WHERE Gesamtkosten between '100' AND '250';

(Zeigt diejenigen, mit ihren Kunden-IDs, die zwischen 100 und 250 Lira bestellen)

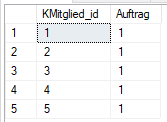
6) SELECT Vorname,Nachname FROM Kaufer ORDER BY Vorname ASC;

(Zeigt die Kundennamen an und sortiert alphabetisch nach Namen)



7) SELECT KMitglied\_id, COUNT(Auftrag\_id) AS Auftrag FROM Auftrag GROUP BY KMitglied\_id;

(Zählt und gruppiert die Anzahl der Bestellungen nach Kunden-ID)



8) SELECT COUNT (KMitglied\_id) AS Kundenanzahl FROM Kaufer;

(Zeigt die Gesamtzahl der Kunden)



9) SELECT SUM(Menge) as Gesamtmenge\_von\_Auftraegen FROM Auftrag\_Detail WHERE Menge between '0' AND '100';

(Zeigt die angegebene Gesamtbestellung an)



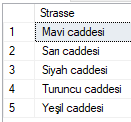
10) SELECT Kurier FROM Lieferung WHERE EXISTS (SELECT Strasse FROM Zahlung WHERE Zahlung.Rech\_strasse= Lieferung.Strasse);

(Zeigt an, welche Versandhaus auf Straßen fährt, die in Lieferung- und Zahlung-Tabellen gleich sind)



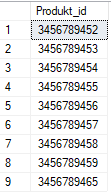
11) SELECT Strasse FROM Lieferung UNION SELECT Rech\_Strasse FROM Zahlung ORDER BY Strasse;

(Vergleicht Lieferungsinformationen mit Straßennamen in Zahlungsinformationen und sortiert sie alphabetisch)



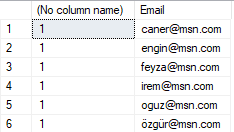
12) SELECT Produkt\_id FROM Produkt UNION SELECT Produkt\_id FROM Auftrag\_Detail ORDER BY Produkt\_id;

(Vergleicht Produkt IDs in Zahlung und Produkt und sortiert sie alphabetisch)



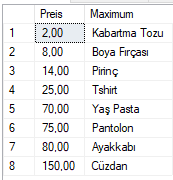
13) SELECT COUNT(Kmitglied\_id), Email FROM Kaufer GROUP BY Email HAVING COUNT(Kmitglied\_id) < 5;

(Zeigt an, wie oft eine E-Mail komprimiert wurde)



14) SELECT Preis, MAX(Pname) Maximum FROM Produkt GROUP BY Preis;

(Sortiert Produkte nach min max Preis)



**INSERT**

15) INSERT INTO Lieferung(Lieferung\_id,PLZ,Stadt)

VALUES(6,22582, 'Edirne')

16) INSERT INTO Lieferung(Lieferung\_id,PLZ,Stadt)

VALUES(7,35456,'İzmir')

17) INSERT INTO Lieferung(Lieferung\_id,PLZ,Stadt)

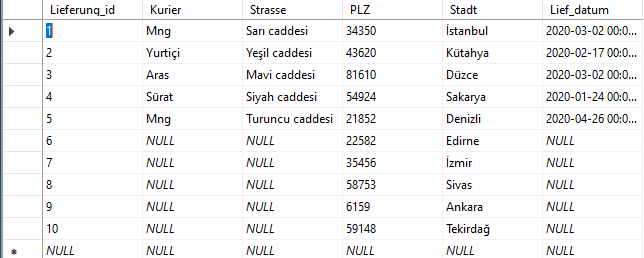
VALUES(8,58753,'Sivas')

18) INSERT INTO Lieferung(Lieferung\_id,PLZ,Stadt)

VALUES(9,06159,'Ankara')

19) INSERT INTO Lieferung(Lieferung\_id,PLZ,Stadt)

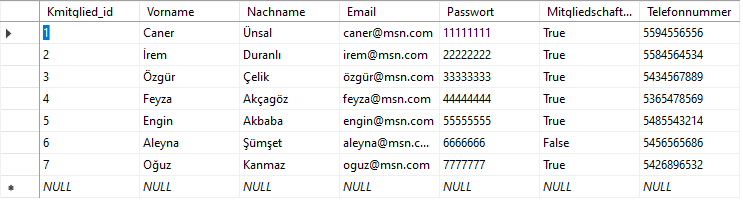
VALUES(10,59148,'Tekirdağ')



20) INSERT INTO Kaufer(Kmitglied\_id,Vorname,Nachname,Email,Passwort,Mitgliedschaftstatus,Telefonnummer)

VALUES (7,'Oğuz','Kanmaz','oguz@msn.com','7777777',1,5426896532)

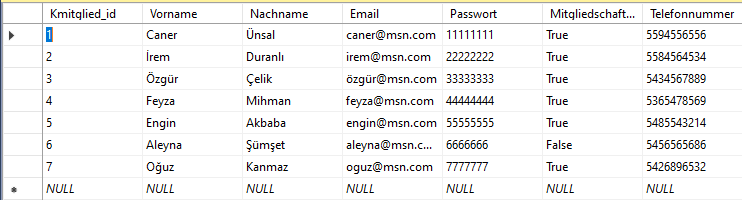
(Fügt dem System neue Kunden hinzu und zeigt deren Informationen an)



**UPDATE**

21) UPDATE Kaufer SET Nachname='Mihman' WHERE Kmitglied\_id = 4;

(Ändert den Nachnamen des Kunden)



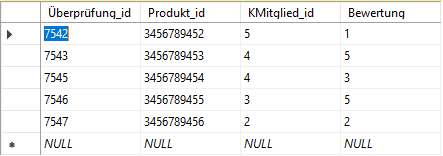
22) UPDATE Verkauferbewertung SET Zeitsicherung='4.9' WHERE Verkauferbewertung\_id= 167852;

(Aktualisiert die Zeitsicherungspunkte der zweiten Verkauferbewertung)



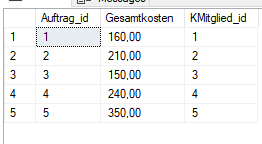
23) UPDATE Überprüfung SET Bewertung='3' WHERE Überprüfung\_id = 7545;

(Aktualisiert der Bewertung der Überprüfung\_id 7545)



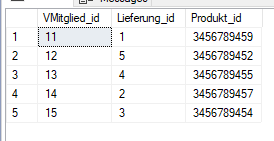
24) UPDATE Auftrag SET Gesamtkosten=350 WHERE Auftrag\_id = 5;

(Aktualisiert den Preis der fünfte Bestellung)



25) UPDATE PDistribution SET Produkt\_id=3456789454 WHERE VMitglied\_id=15;

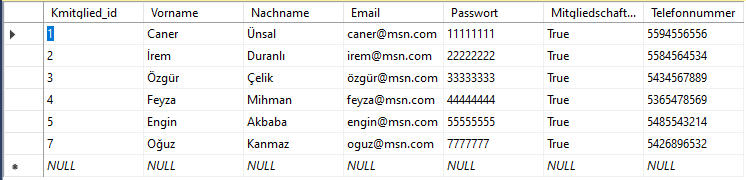
(Aktualisiert die produkt\_id der fünfzehnten Bestellung)



**DELETE**

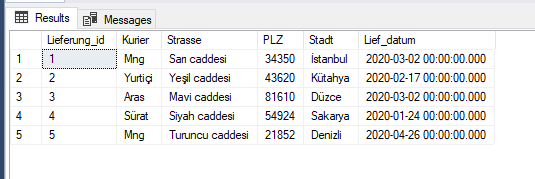
26) DELETE FROM Kaufer WHERE Mitgliedschaftstatus=0

(Löscht Sie diejenigen, deren Mitgliedschaftsstatus inaktiv ist)



27) DELETE FROM Lieferung WHERE Strasse IS NULL;

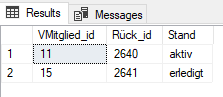
(Löscht die Bestellung ohne Adressangaben)



**ALTER TABLE / DROP**

28) ALTER TABLE Rückerstattung\_Lieferung ADD Stand char(11);

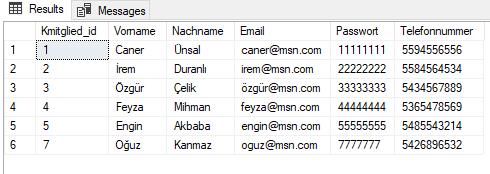
(Fügt der Tabelle eine Standspalte hinzu)



29) ALTER TABLE Kaufer

DROP COLUMN Mitgliedschaftstatus;

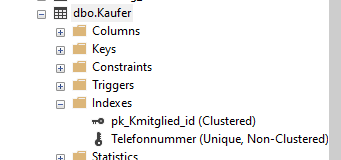
(Entfernt die Spalte mitgliedschaftstatus in der Caufer-Tabelle.)



30) ALTER TABLE Kaufer

ADD CONSTRAINT Telefonnummer UNIQUE(Telefonnummer)

(Fügt der Kaufer-Tabelle die Telefonnummer als eindeutigen Wert hinzu)



31) ALTER TABLE Kaufer

DROP CONSTRAINT Telefonnummer

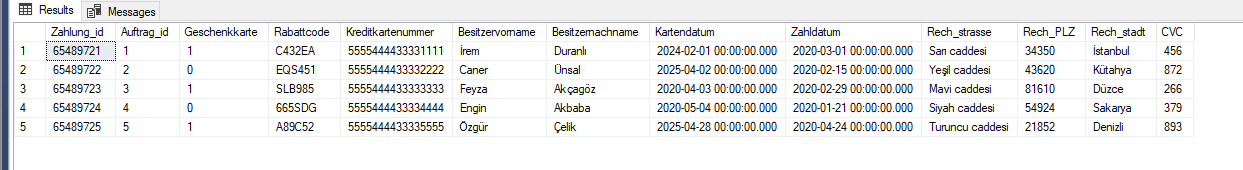
(Löscht die eindeutige Telefonnummer in der Kaufer-Tabelle)



32) ALTER TABLE Zahlung

DROP COLUMN Attributename

Löscht die Attributnamenspalte in der zahlung-Tabelle

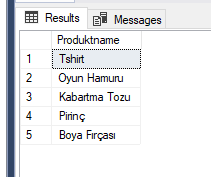


**VIEW**

33) CREATE VIEW vMin50 AS

SELECT Pname AS Produktname FROM Produkt WHERE Preis<50

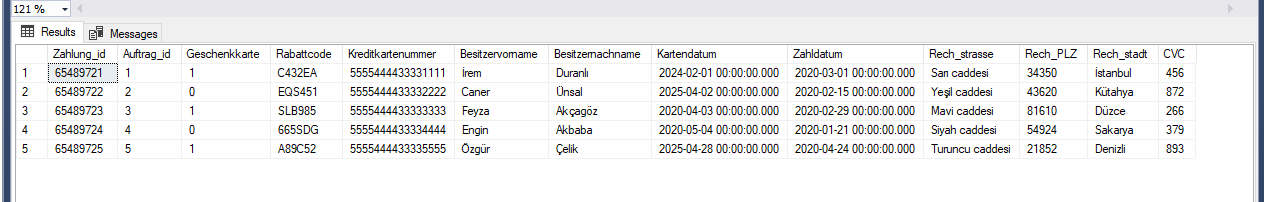
(Zeigt Produkte unter 50 Lira an)



34) CREATE VIEW vZahlungen AS

SELECT \* FROM Zahlung;

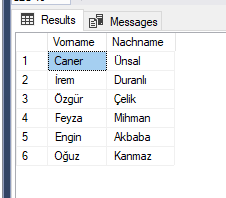
(Zeigt zahlende Kunden und deren Informationen)



35) CREATE VIEW KauferAbf AS

SELECT Vorname,Nachname FROM Kaufer

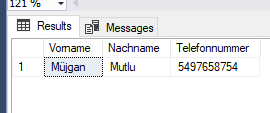
(Zeigt aktuelle Kunden an)



36) CREATE VIEW vMutluGıda AS

SELECT Vorname,Nachname,Telefonnummer FROM Verkaufer WHERE Betriebname='Mutlu Gıda';

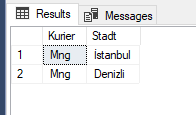
(Zeigt die Informationen des betreffenden Unternehmens)



37) CREATE VIEW vMngLief AS

SELECT Kurier,Stadt FROM Lieferung WHERE Kurier='Mng'

(Zeigt an, in welche Städte das betreffende Frachtunternehmen liefert)

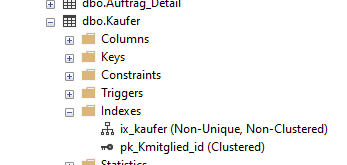


**INDEX**

38) CREATE INDEX ix\_kaufer

ON Kaufer(Vorname,Nachname,Telefonnummer);

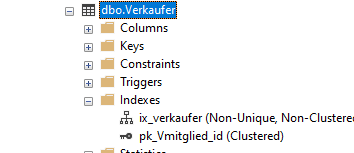
(Erstellt einen Indexwert mit dem Namen ix\_kaufer)



39) CREATE INDEX ix\_verkaufer

ON Verkaufer(Vorname,Nachname,Telefonnummer,Betriebname);

(Erstellt einen Indexwert mit dem Namen ix\_verkaufer)

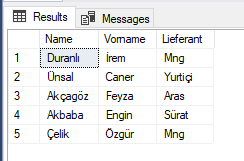


**JOIN**

40) SELECT Besitzernachname AS Name, Besitzervorname AS Vorname, Kurier AS Lieferant

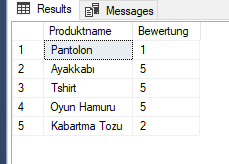
FROM Lieferung INNER JOIN Zahlung ON Lieferung.Strasse = Zahlung.Rech\_strasse;

(Zeigt an, wen mit welchem Kurier geliefert werden soll)



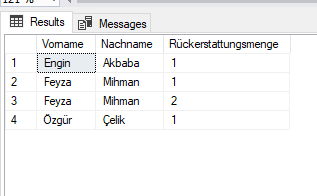
41) SELECT PName AS Produktname, Bewertung FROM Produkt INNER JOIN Überprüfung ON Produkt.Produkt\_id = Überprüfung.Produkt\_id;

(Zeigt an, welches Produkt wie viele Punkte hat)



42) SELECT Vorname,Nachname,Rück\_menge AS Rückerstattungsmenge FROM Kaufer INNER JOIN Rückerstattung ON Kaufer.Kmitglied\_id = Rückerstattung.KMitglied\_id;

(Zeigt an, wer wie viele Artikel zurückgegeben hat)



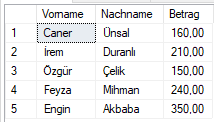
43) SELECT Vorname,Nachname,Kreditkartenummer As Kreditkarte\_INFO FROM Kaufer INNER JOIN Zahlung ON Vorname = Besitzervorname;

(Zeigt Karteninformationen zahlender Kunden an)



44) SELECT Vorname,Nachname,Gesamtkosten as Betrag FROM Auftrag LEFT JOIN Kaufer ON Auftrag.Kmitglied\_id = Kaufer.KMitglied\_id;

(Mitglied\_id vergleicht laut Auftrag und nennt den Nachnamen)



45) SELECT Besitzernachname AS Name, Besitzervorname AS Vorname, Stadt, Geschenkkarte, Rabattcode FROM Lieferung INNER JOIN Zahlung ON Lieferung.Stadt = Zahlung.Rech\_stadt



**TRIGGER**

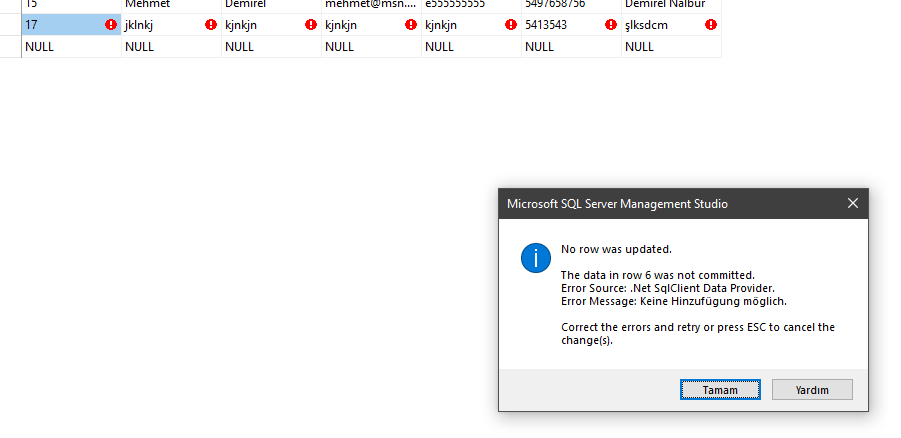
46) CREATE TRIGGER keinLiefern

ON Verkaufer

AFTER INSERT

AS RAISERROR ('Keine Hinzufügung möglich.',16,10);

(Verhindert, dass neue Daten in die Tabelle eingegeben werden)



**PROCEDURE**

47) CREATE PROCEDURE uspTelefon

@Vorname nvarchar(20) = N'Caner',

@Nachname nvarchar(20) = N'%'

AS

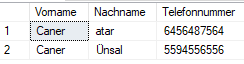
SET NOCOUNT ON;

SELECT Vorname, Nachname, Telefonnummer FROM Kaufer

WHERE Vorname LIKE @Vorname AND Nachname LIKE @Nachname;

(Die anderen mit vorname Caner vergleichen die mit irgendwelchen Informationen und geben die Ausgabe.)

GO EXEC uspTelefon;



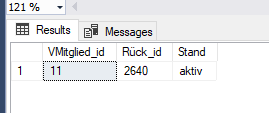
**TRANSACTION**

48) BEGIN TRANSACTION;

DELETE FROM Rückerstattung\_Lieferung WHERE Stand = 'erledigt'

COMMIT;

(Löscht die genehmigte Rückgabe aus der Tabelle)



49) BEGIN TRANSACTION;

INSERT INTO Produkt VALUES(3456789465,'Cüzdan',50,150,'Deri Cüzdan');

SAVE TRANSACTION SP1

INSERT INTO Produkt VALUES(3456789461,'Klavye',150,500,'Gamer Klavye');

SAVE TRANSACTION SP2

INSERT INTO Produkt VALUES(3456789462,'Laptop',93,5128,'Gamer Laptop');

ROLLBACK TRANSACTION SP1

COMMIT;

(Fügt der Tabelle 3 neue Elemente hinzu und löscht die letzten 2)

