



---

# DATENBANK

---

Doç. Dr. Selçuk KIRAN

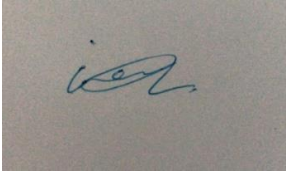


İrem Duranlı - 138417038

Caner Ünsal - 138417852

Özgüradz Çelik - 138417056

September 2020



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 Verkäuferbewertung;
DROP TABLE Rückerstattung_Lieferung;
DROP TABLE Rückerstattung;
DROP TABLE Auftrag_Detail;
DROP TABLE Zahlung;
DROP TABLE Auftrag;
DROP TABLE Produkt;
DROP TABLE Verkäufer;
DROP TABLE Käufer;

```

## CREATE

```

CREATE TABLE Käufer(
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 Verkäufer(
Vmitglied_id int NOT NULL,
Vorname nvarchar(20),
Nachname nvarchar(20),
Email nvarchar(50),
Passwort nvarchar(15),
Telefonnummer bigint,
Betriebsname 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_produk_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 Käufer(Kmitglied_id)
)

```

```

CREATE TABLE Zahlung(
Zahlung_id bigint NOT NULL,
Auftrag_id bigint NOT NULL,
Geschenkkarte bit,
Rabattcode nvarchar(10),
Attributenamen int,
Kreditkartennummer 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
Käufer(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
Verkäufer(Vmitglied_id),
CONSTRAINT fk_rück_id FOREIGN KEY (Rück_id) REFERENCES Rückerstattung(Rück_id)
)

```

```

CREATE TABLE Verkäuferbewertung(
Verkäuferbewertung_id bigint NOT NULL,
Mitglied_id int NOT NULL,
Qualität decimal(2,1),
Rücklaufequote decimal(2,1),
Zeitsicherung decimal(2,1),
CONSTRAINT pk_verkäuferbewertung_id PRIMARY KEY (Verkäuferbewertung_id),

```

```
CONSTRAINT fk_verkauferebewertungmitglied_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  
Käufer(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)

	Produktnummer	Produkt
1	3456789455	Oyun Hamuru
2	3456789456	Kabartma Tozu
3	3456789457	Pirinç
4	3456789458	Yaş Pasta

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

(Zeigt den Durchschnittspreis der aufgegebenen Bestellungen an)

	Durchschnitt
1	216,00

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

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

	Produkt	Gesamtstock	Preis
1	Kabartma Tozu	20	2,00
2	Boya Fırçası	50	8,00
3	Pirinç	15	14,00
4	Tshirt	30	25,00
5	Oyun Hamuru	12	25,00
6	Yaş Pasta	8	70,00
7	Pantolon	50	75,00
8	Ayakkabı	50	80,00

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

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

	Vorname	Nachname	Telefonnummer
1	Caner	Ünsal	5594556556

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)

	KMitglied_id	Gesamtkosten
1	1	160,00
2	2	210,00
3	3	150,00
4	4	240,00

6) `SELECT Vorname,Nachname FROM Käufer ORDER BY Vorname ASC;`

(Zeigt die Kundennamen an und sortiert alphabetisch nach Namen)

	Vorname	Nachname
1	Caner	Ünsal
2	Engin	Akbaba
3	Feyza	Mihman
4	İrem	Duranlı
5	Oğuz	Kanmaz
6	Özgür	Çelik

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)

	KMitglied_id	Auftrag
1	1	1
2	2	1
3	3	1
4	4	1
5	5	1

8) `SELECT COUNT (KMitglied_id) AS Kundenanzahl FROM Käufer;`

(Zeigt die Gesamtzahl der Kunden)

	Kundenanzahl
1	6

9) `SELECT SUM(Menge) as Gesamtmenge_von_Auftraegen FROM Auftrag_Detail WHERE Menge between '0' AND '100';`

(Zeigt die angegebene Gesamtbestellung an)

	Gesamtmenge_von_Auftraegen
1	17

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)

	Kurier
1	Mng
2	Yurtiçi
3	Aras
4	Sürat
5	Mng

11) `SELECT Strasse FROM Lieferung UNION SELECT Rech_Strasse FROM Zahlung ORDER BY Strasse;`

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

	Strasse
1	Mavi caddesi
2	San caddesi
3	Siyah caddesi
4	Turuncu caddesi
5	Yeşil caddesi

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)

	Produkt_id
1	3456789452
2	3456789453
3	3456789454
4	3456789455
5	3456789456
6	3456789457
7	3456789458
8	3456789459
9	3456789465

13) `SELECT COUNT(Kmitglied_id), Email FROM Käufer GROUP BY Email HAVING COUNT(Kmitglied_id) < 5;`

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

	(No column name)	Email
1	1	caner@msn.com
2	1	engin@msn.com
3	1	feyza@msn.com
4	1	irem@msn.com
5	1	oguz@msn.com
6	1	özgür@msn.com



(Sortiert Produkte nach min max Preis)

	Preis	Maximum
1	2,00	Kabartma Tozu
2	8,00	Boya Fırçası
3	14,00	Pirinç
4	25,00	Tshirt
5	70,00	Yaş Pasta
6	75,00	Pantolon
7	80,00	Ayakkabı
8	150,00	Cüzdan

**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')
```

	Lieferung_id	Kurier	Strasse	PLZ	Stadt	Lief_datum
▶	1	Mng	Sarı caddesi	34350	İstanbul	2020-03-02 00:00:00
	2	Yurtiçi	Yeşil caddesi	43620	Kütahya	2020-02-17 00:00:00
	3	Aras	Mavi caddesi	81610	Düzce	2020-03-02 00:00:00
	4	Sûrat	Siyah caddesi	54924	Sakarya	2020-01-24 00:00:00
	5	Mng	Turuncu caddesi	21852	Denizli	2020-04-26 00:00:00
	6	NULL	NULL	22582	Edirne	NULL
	7	NULL	NULL	35456	İzmir	NULL
	8	NULL	NULL	58753	Sivas	NULL
	9	NULL	NULL	6159	Ankara	NULL
	10	NULL	NULL	59148	Tekirdağ	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL

## 20) INSERT INTO

```
Käufer(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)

[illegible]

## UPDATE

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

(Ändert den Nachnamen des Kunden)

	Kmitglied_id	Vorname	Nachname	Email	Passwort	Mitgliedschaft...	Telefonnummer
▶	1	Caner	Ünsal	caner@msn.com	11111111	True	5594556556
	2	İrem	Duranlı	irem@msn.com	22222222	True	5584564534
	3	Özgür	Çelik	özgür@msn.com	33333333	True	5434567889
	4	Feyza	Mihman	feyza@msn.com	44444444	True	5365478569
	5	Engin	Akbaba	engin@msn.com	55555555	True	5485543214
	6	Aleyna	Şümşet	aleyna@msn.c...	66666666	False	5456565686
	7	Oğuz	Kanmaz	oguz@msn.com	77777777	True	5426896532
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

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

(Aktualisiert die Zeitsicherungspunkte der zweiten Verkäuferbewertung)

	Verkäuferbewertung_id	VMitglied_id	Qualität	Rücklaufequote	Zeitsicherung
1	156741	12	3.2	2.0	4.5
2	167852	13	4.2	1.0	4.9

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

(Aktualisiert der Bewertung der Überprüfung\_id 7545)

	Überprüfung_id	Produkt_id	KMitglied_id	Bewertung
▶	7542	3456789452	5	1
	7543	3456789453	4	5
	7545	3456789454	4	3
	7546	3456789455	3	5
	7547	3456789456	2	2
*	NULL	NULL	NULL	NULL

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

(Aktualisiert den Preis der fünfte Bestellung)

	Auftrag_id	Gesamtkosten	KMitglied_id
1	1	160,00	1
2	2	210,00	2
3	3	150,00	3
4	4	240,00	4
5	5	350,00	5

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

(Aktualisiert die produkt\_id der fünfzehnten Bestellung)

	VMitglied_id	Lieferung_id	Produkt_id
1	11	1	3456789459
2	12	5	3456789452
3	13	4	3456789455
4	14	2	3456789457
5	15	3	3456789454

## DELETE

26) DELETE FROM Käufer WHERE Mitgliedschaftsstatus=0

(Löscht Sie diejenigen, deren Mitgliedschaftsstatus inaktiv ist)

	Kmitglied_id	Vorname	Nachname	Email	Passwort	Mitgliedschaft...	Telefonnummer
▶	1	Caner	Ünsal	caner@msn.com	11111111	True	5594556556
	2	İrem	Duranlı	irem@msn.com	22222222	True	5584564534
	3	Özgür	Çelik	özgür@msn.com	33333333	True	5434567889
	4	Feyza	Mihman	feyza@msn.com	44444444	True	5365478569
	5	Engin	Akbaba	engin@msn.com	55555555	True	5485543214
	7	Oğuz	Kanmaz	oguz@msn.com	77777777	True	5426896532
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

27) DELETE FROM Lieferung WHERE Strasse IS NULL;

(Löscht die Bestellung ohne Adressangaben)

	Lieferung_id	Kurier	Strasse	PLZ	Stadt	Lief_datum
1	1	Mng	San caddesi	34350	İstanbul	2020-03-02 00:00:00.000
2	2	Yurtiçi	Yeşil caddesi	43620	Kütahya	2020-02-17 00:00:00.000
3	3	Aras	Mavi caddesi	81610	Düzce	2020-03-02 00:00:00.000
4	4	Sürat	Siyah caddesi	54924	Sakarya	2020-01-24 00:00:00.000
5	5	Mng	Turuncu caddesi	21852	Denizli	2020-04-26 00:00:00.000

## ALTER TABLE / DROP

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

(Fügt der Tabelle eine Standspalte hinzu)

	VMitglied_id	Rück_id	Stand
1	11	2640	aktiv
2	15	2641	erledigt

29) ALTER TABLE Käufer

DROP COLUMN Mitgliedschaftstatus;

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

	Kmitglied_id	Vorname	Nachname	Email	Passwort	Telefonnummer
1	1	Caner	Ünsal	caner@msn.com	11111111	5594556556
2	2	İrem	Duranlı	irem@msn.com	22222222	5584564534
3	3	Özgür	Çelik	özgür@msn.com	33333333	5434567889
4	4	Feyza	Mihman	feyza@msn.com	44444444	5365478569
5	5	Engin	Akbaba	engin@msn.com	55555555	5485543214
6	7	Oğuz	Kanmaz	oguz@msn.com	77777777	5426896532

30) ALTER TABLE Käufer

ADD CONSTRAINT Telefonnummer UNIQUE(Telefonnummer)

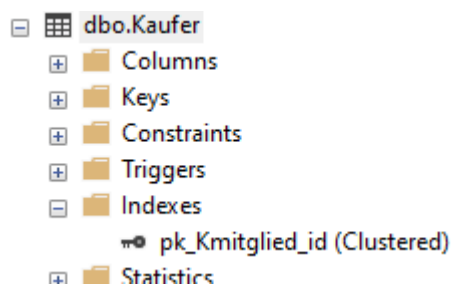
(Fügt der Käufer-Tabelle die Telefonnummer als eindeutigen Wert hinzu)



31) ALTER TABLE Käufer

DROP CONSTRAINT Telefonnummer

(Löscht die eindeutige Telefonnummer in der Käufer-Tabelle)



32) ALTER TABLE Zahlung

DROP COLUMN Attributenamen

Löscht die Attributnamenspalte in der zahlung-Tabelle

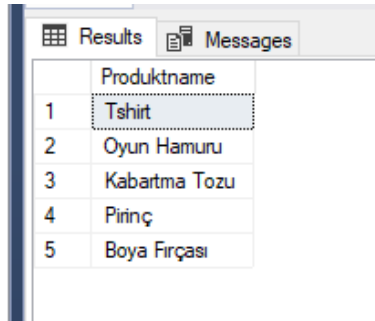
	Zahlung_id	Auftrag_id	Geschenkkarte	Rabattcode	Kreditkartennummer	Besitzervorname	Besitzernachname	Kartendatum	Zahldatum	Rech_strasse	Rech_PLZ	Rech_stadt	CVC
1	65489721	1	1	C432EA	5555444433331111	İrem	Duranlı	2024-02-01 00:00:00.000	2020-03-01 00:00:00.000	San caddesi	34350	İstanbul	456
2	65489722	2	0	EQS451	5555444433332222	Caner	Ünsal	2025-04-02 00:00:00.000	2020-02-15 00:00:00.000	Yeşil caddesi	43620	Kütahya	872
3	65489723	3	1	SLB985	5555444433333333	Feyza	Akçagöz	2020-04-03 00:00:00.000	2020-02-29 00:00:00.000	Mavi caddesi	81610	Düzce	266
4	65489724	4	0	665SDG	5555444433334444	Engin	Akbaba	2020-05-04 00:00:00.000	2020-01-21 00:00:00.000	Siyah caddesi	54924	Sakarya	379
5	65489725	5	1	A89C52	5555444433335555	Özgür	Çelik	2025-04-28 00:00:00.000	2020-04-24 00:00:00.000	Turuncu caddesi	21852	Denizli	893

## VIEW

33) CREATE VIEW vMin50 AS

```
SELECT Pname AS Produktname FROM Produkt WHERE Preis<50
```

(Zeigt Produkte unter 50 Lira an)

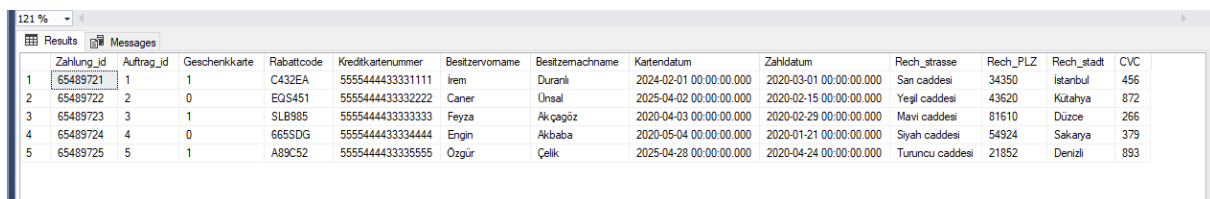


	Produktname
1	Tshirt
2	Oyun Hamuru
3	Kabartma Tozu
4	Pirinç
5	Boya Fırçası

34) CREATE VIEW vZahlungen AS

```
SELECT * FROM Zahlung;
```

(Zeigt zahlende Kunden und deren Informationen)

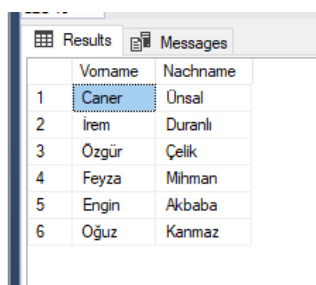


Zahlung_id	Auftrag_id	Geschenkkarte	Rabattcode	Kreditkartennummer	Bestizervorname	Bestizernachname	Kartendatum	Zahldatum	Rech_strasse	Rech_PLZ	Rech_stadt	CVC
1	65489721	1	C432EA	5555444433331111	Irem	Duranlı	2024-02-01 00:00:00.000	2020-03-01 00:00:00.000	San caddesi	34350	Istanbul	456
2	65489722	2	EQS451	5555444433332222	Caner	Ünsal	2025-04-02 00:00:00.000	2020-02-15 00:00:00.000	Yegil caddesi	43620	Kütahya	872
3	65489723	3	SLB985	5555444433333333	Feyza	Akçagöz	2020-04-03 00:00:00.000	2020-02-29 00:00:00.000	Mavi caddesi	81610	Düzce	266
4	65489724	4	665SDG	5555444433334444	Engin	Akbaba	2020-05-04 00:00:00.000	2020-01-21 00:00:00.000	Siyah caddesi	54924	Sakarya	379
5	65489725	5	A89C52	5555444433335555	Özgür	Çelik	2025-04-28 00:00:00.000	2020-04-24 00:00:00.000	Turuncu caddesi	21852	Denizli	893

35) CREATE VIEW KauferAbf AS

```
SELECT Vorname,Nachname FROM Kaufer
```

(Zeigt aktuelle Kunden an)

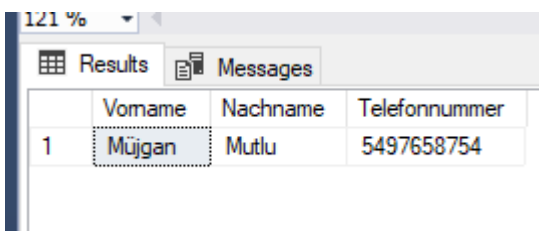


	Vorname	Nachname
1	Caner	Ünsal
2	İrem	Duranlı
3	Özgür	Çelik
4	Feyza	Mihman
5	Engin	Akbaba
6	Oğuz	Kanmaz

36) CREATE VIEW vMutluGıda AS

```
SELECT Vorname,Nachname,Telefonnummer FROM Verkäufer WHERE Betriebsname='Mutlu Gıda';
```

(Zeigt die Informationen des betreffenden Unternehmens)



	Vorname	Nachname	Telefonnummer
1	Müjgan	Mutlu	5497658754

37) CREATE VIEW vMngLief AS

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

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

Results		Messages
	Kurier	Stadt
1	Mng	Istanbul
2	Mng	Denizli

## INDEX

38) CREATE INDEX ix\_kaufer

ON Kaufer(Vorname,Nachname,Telefonnummer);

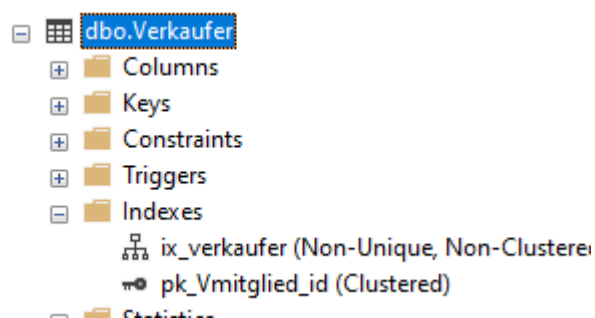
(Erstellt einen Indexwert mit dem Namen ix\_kaufer)



39) CREATE INDEX ix\_verkauf

ON Verkäufer(Vorname,Nachname,Telefonnummer,Betriebsname);

(Erstellt einen Indexwert mit dem Namen ix\_verkauf)



## 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)

	Name	Vorname	Lieferant
1	Duranlı	İrem	Mng
2	Ünsal	Caner	Yurtiçi
3	Akçagöz	Feyza	Aras
4	Akbaba	Engin	Sürat
5	Çelik	Özgür	Mng

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)

	Produktname	Bewertung
1	Pantolon	1
2	Ayakkabı	5
3	Tshirt	5
4	Oyun Hamuru	5
5	Kabartma Tozu	2

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

(Zeigt an, wer wie viele Artikel zurückgegeben hat)

	Vorname	Nachname	Rückerstattungsmenge
1	Engin	Akbaba	1
2	Feyza	Mihman	1
3	Feyza	Mihman	2
4	Özgür	Çelik	1

43) `SELECT` Vorname, Nachname, Kreditkartennummer `As` Kreditkarte\_INFO `FROM` Käufer `INNER`  
`JOIN` Zahlung `ON` Vorname = Besitzervorname;

(Zeigt Karteninformationen zahlender Kunden an)

	Vorname	Nachname	Kreditkarte_INFO
1	Caner	Ünsal	5555444433332222
2	İrem	Duranlı	5555444433331111
3	Özgür	Çelik	5555444433335555
4	Feyza	Mihman	5555444433333333
5	Engin	Akbaba	5555444433334444

44) `SELECT Vorname,Nachname,Gesamtkosten as Betrag FROM Auftrag LEFT JOIN Käufer ON Auftrag.KMitglied_id = Käufer.KMitglied_id;`

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

	Vorname	Nachname	Betrag
1	Caner	Ünsal	160,00
2	İrem	Duranlı	210,00
3	Özgür	Çelik	150,00
4	Feyza	Mihman	240,00
5	Engin	Akbaba	350,00

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

	Name	Vorname	Stadt	Geschenkkarte	Rabattcode
1	Duranlı	İrem	İstanbul	1	C432EA
2	Ünsal	Caner	Kütahya	0	EQS451
3	Akçagöz	Feyza	Düzce	1	SLB985
4	Akbaba	Engin	Sakarya	0	665SDG
5	Çelik	Özgür	Denizli	1	A89C52

## TRIGGER

46) `CREATE TRIGGER keinLiefern`

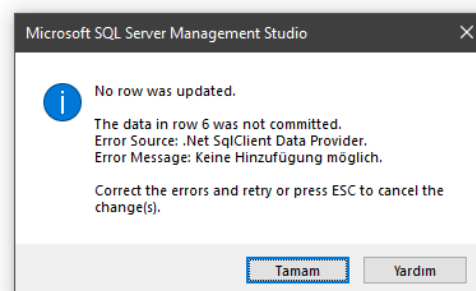
`ON Verkäufer`

`AFTER INSERT`

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

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

ID	Mehmet	Demirel	mehmet@msn....	еееееееееее	349/038/36	Demirel Naibur
17	jklkj	kjnkjn	kjnkjn	kjnkjn	5413543	şlkscdm
NULL	NULL	NULL	NULL	NULL	NULL	NULL





## PROCEDURE

```
47) CREATE PROCEDURE uspTelefon
    @Vorname nvarchar(20) = N'Caner',
    @Nachname nvarchar(20) = N'%'
```

AS

```
    SET NOCOUNT ON;
    SELECT Vorname, Nachname, Telefonnummer FROM Käufer
    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;
```

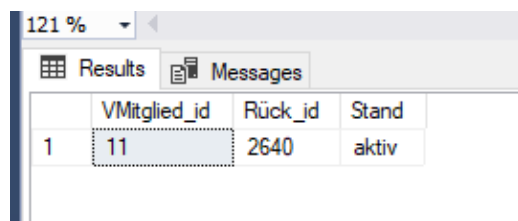
	Vorname	Nachname	Telefonnummer
1	Caner	atar	6456487564
2	Caner	Ünsal	5594556556

## TRANSACTION

```
48) BEGIN TRANSACTION;
```

```
DELETE FROM Rückerstattung_Lieferung WHERE Stand = 'erledigt'
COMMIT;
```

(Löscht die genehmigte Rückgabe aus der Tabelle)

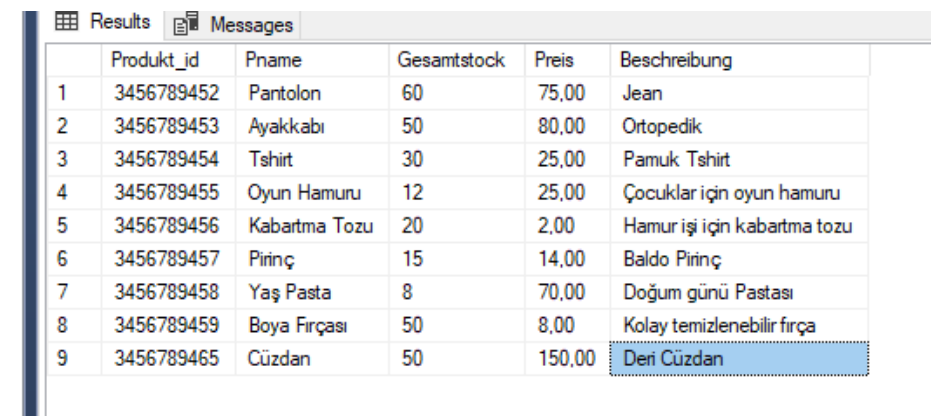


	VMitglied_id	Rück_id	Stand
1	11	2640	aktiv

```
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)



	Produkt_id	Pname	Gesamtstock	Preis	Beschreibung
1	3456789452	Pantolon	60	75,00	Jean
2	3456789453	Ayakkabı	50	80,00	Ortopedik
3	3456789454	Tshirt	30	25,00	Pamuk Tshirt
4	3456789455	Oyun Hamuru	12	25,00	Çocuklar için oyun hamuru
5	3456789456	Kabartma Tozu	20	2,00	Hamur işi için kabartma tozu
6	3456789457	Pirinç	15	14,00	Baldo Pirinç
7	3456789458	Yaş Pasta	8	70,00	Doğum günü Pastası
8	3456789459	Boya Fırçası	50	8,00	Kolay temizlenebilir fırça
9	3456789465	Cüzdan	50	150,00	Deri Cüzdan