

Pharmacy System

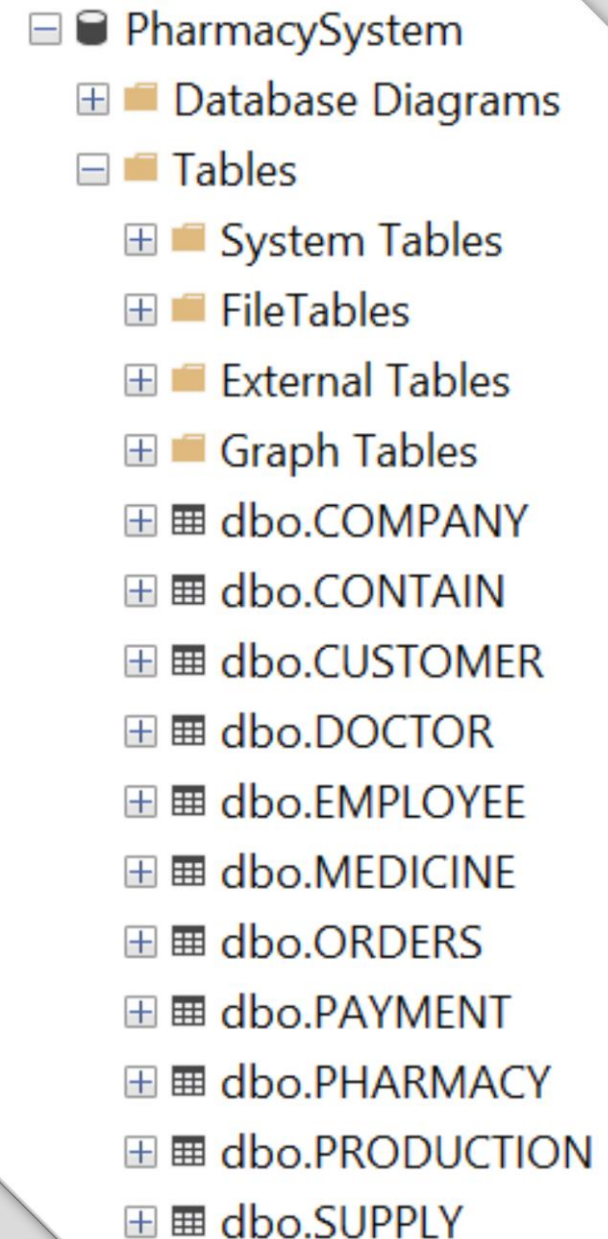
Esra Eryılmaz

171044046



Pharmacy System Details

- The main purpose of this project is to manage the database of a pharmacy efficiently.
- The pharmacist will use the pharmacy management system to minimize the time and resources by maintaining the details of the medicines and the customers(patients) systemically so that the data can be used in the possible quickest time.
- There are 11 tables



What have been done in the project?

- User requirements,
- E-R diagram,
- Normalizations,
- Functional dependencies,
- Table, relation, key details,
- Database and high level connection (terminal based interface),
- Queries (Select, insert, join, views...).
- Additional (inheritance, weak entities...)

The details are in the project report...

E-R DIAGRAM

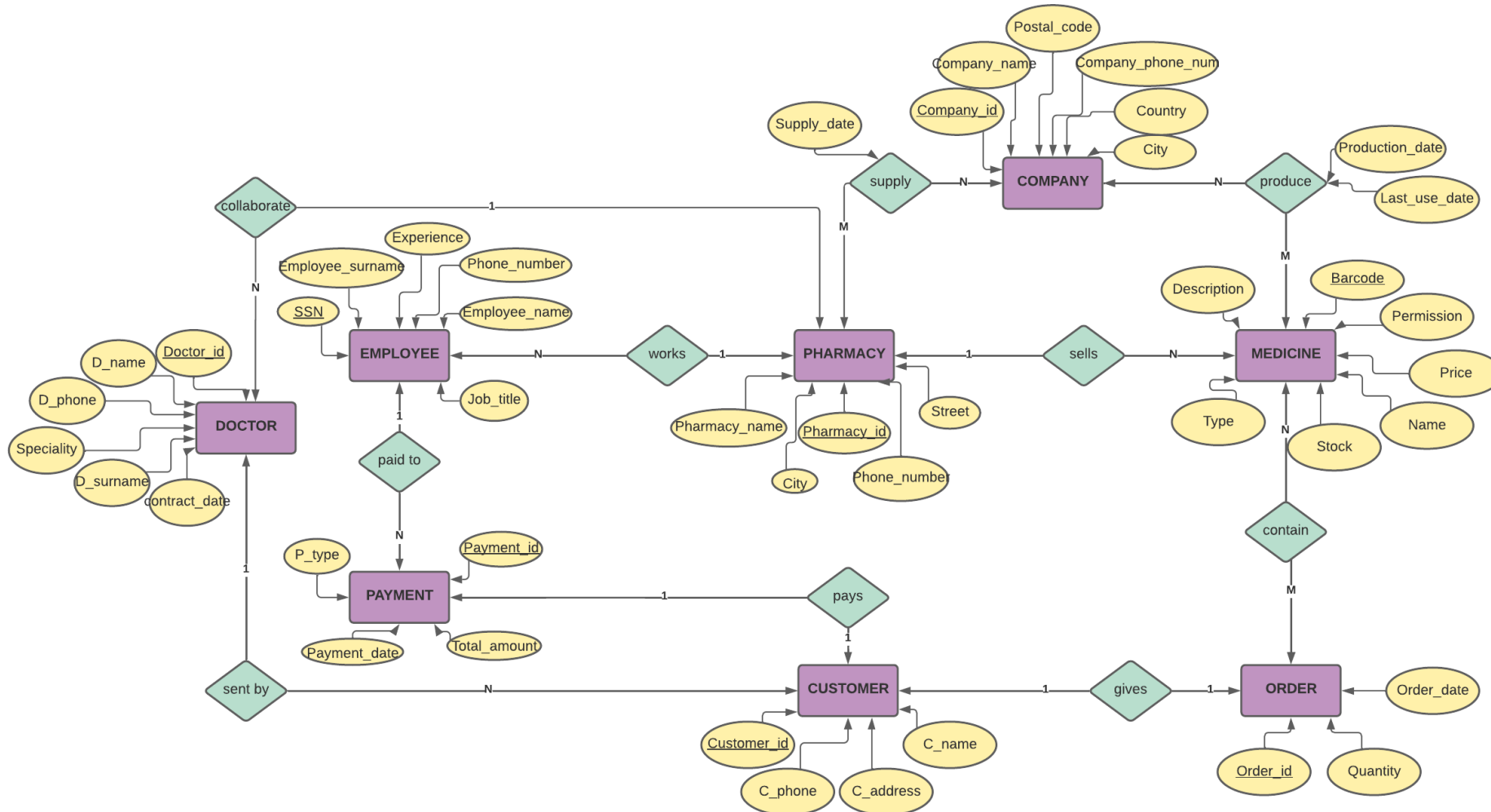
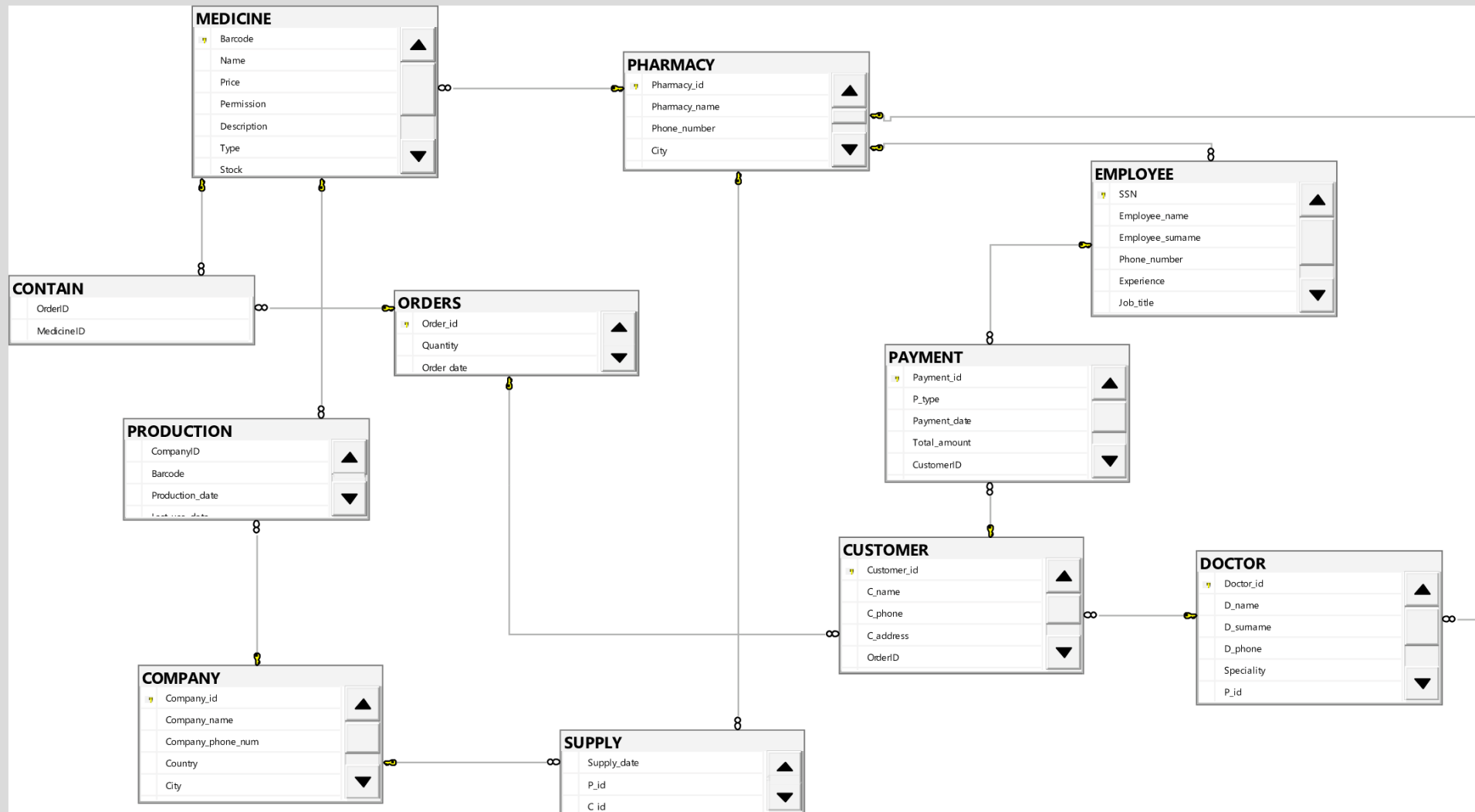
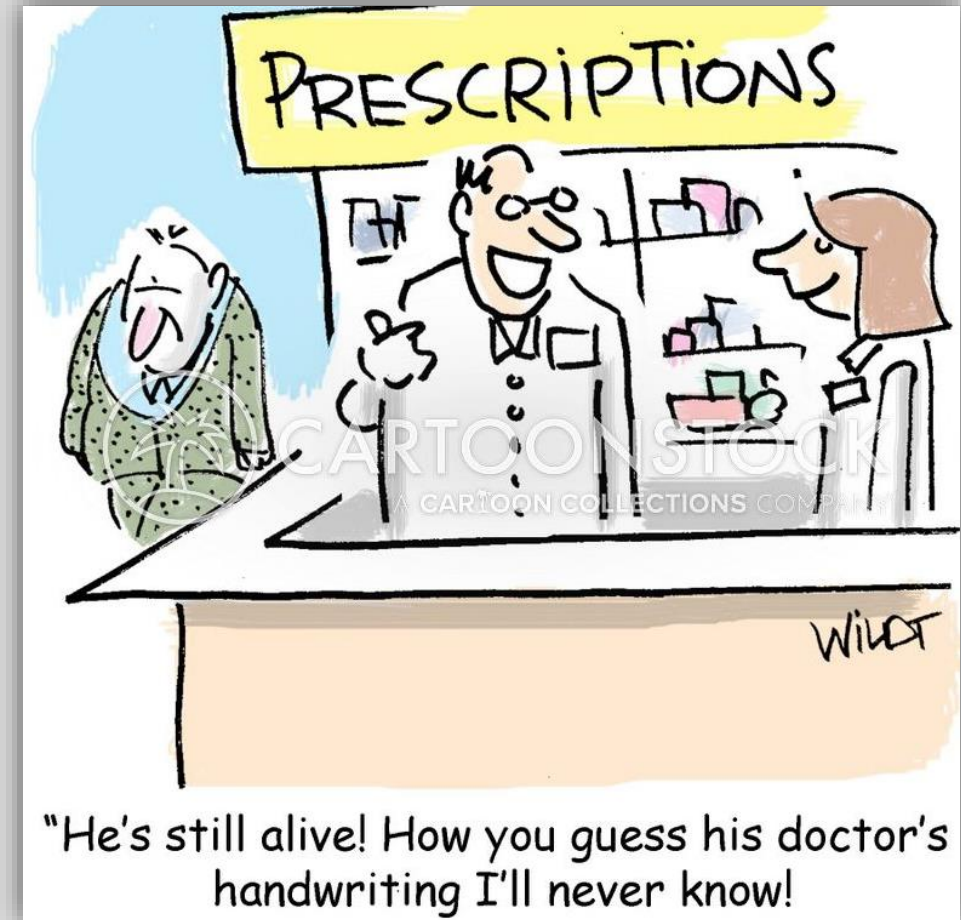


Table Diagram



User Requirements

- Pharmacy can sell medicine,
- Pharmacy can supply medicine from the company
- Employees can work in pharmacy,
- Employees can receive payment from the customer,
- Doctors can collaborate with pharmacy,
- Doctors can send customers to the pharmacy,
- Company can produce medicine,
- Order contains medicine,
- Customers can pay payments to the employee,
- Customers can order medicine from the pharmacy



Functional Dependencies

- **Company_id** → Company_name, Company_phone_num, Country, City, Postal_code
- **Pharmacy_id** → Pharmacy_name, Phone_number, City, Street
- **Supply_date** → P_id, C_id
- **Order_id** → Quantity, Order_date
- **Barcode** → Name, Price, Permission, Description, Type, Stock, P_id
- **SSN** → Employee_name, Employee_surname, Phone_number, Experience, Job_title, P_ID
- **OrderID** → MedicineID
- **CompanyID** → Barcode, Production_date, Last_use_date
- **Doctor_id** → D_name, D_surname, D_phone, Speciality, P_id, contract_date
- **Customer_id** → C_name, C_phone, C_address, OrderID, DoctorID
- **Payment_id** → P_type, Payment_date, Total_amount, CustomerID, EmployeeID

High Level Connection

Microsoft SQL Server - Python

```
# Connecting to a Microsoft SQL Server with pyodbc
import pyodbc

# define the server and the database
server = 'ESRA'
database = 'PharmacySystem'

# Define the connection string
conx = pyodbc.connect(
    'DRIVER={SQL Server}; \
    SERVER='+ server +'; \
    DATABASE='+ database +';\
    Trusted_Connection=yes;'
)

conx_string = "driver={SQL SERVER}; server=ESRA; database=PharmacySystem; trusted_connection=YES;"

query1 = "SELECT MEDICINE.Barcode, MEDICINE.Name, MEDICINE.Description FROM MEDICINE"

conx = pyodbc.connect(conx_string)
cursor = conx.cursor()
cursor.execute(query1)
data = cursor.fetchall()

for d in data:
    print(d)
```

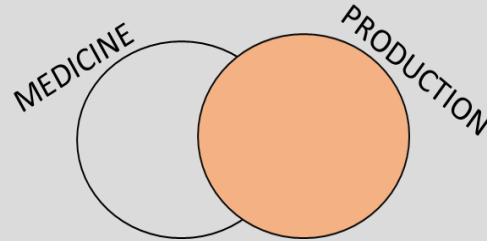

Creation - Insertion

	Barcode	Name	Price	Permission	Description	Type	Stock	P_id
1	100	Advil	6	1	ağrı kesici	tablet	35	1
2	101	Sebamed	40	1	cilt bakım	krem	15	1
3	102	Aerius	10	1	alerji hâpı	tablet	25	1
4	103	Nexium	20	1	mide ilacı	tablet	20	1
5	104	Norvasc	15	1	tansiyon ilacı	tablet	10	1
6	105	Perebron	9	1	öksürük şurubu	şurup	10	1
7	106	Metpamid	12	1	bulantı şurubu	şurup	10	1
8	107	Naprosyn	47	1	romatizma ilacı	tablet	20	1
9	108	Aspirin	1	1	ağrı kesici	tablet	35	1
10	109	Corasbin	17	1	kan sulandırıcı	tablet	20	1
11	110	Voltaren Emulgel	48	1	romatizma ilacı	jel	10	1
12	111	Ocuvite Complete	22	1	göz vitamini	tablet	10	1
13	112	Otrivin	16	1	burun spreyi	sprey	20	1
14	113	Benexol b12	10	1	vitamin ilacı	tablet	35	1
15	114	Asiviral Krem	10	1	uçuk kremi	krem	20	1
16	115	Terramycin	13	1	yara merhemi	krem	10	1
17	116	Siprogut	9	1	kulak damlası	da...	10	1
18	117	Glifor	19	1	şeker ilacı	tablet	10	1
19	118	Ritalin	30	0	uyuşturucu	tablet	10	1
20	119	Medikinet Retart	40	0	uyuşturucu	tablet	20	1
21	120	Prozac	30	0	sakinleştirici	tablet	10	1

```
CREATE TABLE MEDICINE ( Barcode INT NOT NULL,  
Name Varchar(30) NOT NULL,  
Price FLOAT NOT NULL,  
Permission BIT NOT NULL,  
Description VARCHAR(255) NOT NULL,  
Type VARCHAR(15) NOT NULL,  
Stock INT NOT NULL,  
P_id INT NOT NULL,  
Primary KEY (Barcode),  
Foreign KEY (P_id) REFERENCES PHARMACY(Pharmacy_id) );
```

```
INSERT into MEDICINE VALUES(100, 'Advil',6.00,1,'ağrı kesici','tablet',35,1 );  
INSERT into MEDICINE VALUES(101, 'Sebamed',40.0,1,'cilt bakım','krem',15,1 );  
INSERT into MEDICINE VALUES(102, 'Aerius',10.00,1,'alerji hâpı','tablet',25,1 );  
INSERT into MEDICINE VALUES(103, 'Nexium',20.00,1,'mide ilacı','tablet',20,1 );  
INSERT into MEDICINE VALUES(104, 'Norvasc',15.00,1,'tansiyon ilacı','tablet',10,1 );  
INSERT into MEDICINE VALUES(105, 'Perebron',9.00,1,'öksürük şurubu','şurup',10,1 );  
INSERT into MEDICINE VALUES(106, 'Metpamid',12.00,1,'bulantı şurubu','şurup',10,1 );  
INSERT into MEDICINE VALUES(107, 'Naprosyn',47.00,1,'romatizma ilacı','tablet',20,1 );  
INSERT into MEDICINE VALUES(108, 'Aspirin',1.00,1,'ağrı kesici','tablet',35,1 );  
INSERT into MEDICINE VALUES(109, 'Corasbin',17.00,1,'kan sulandırıcı','tablet',20,1 );  
INSERT into MEDICINE VALUES(110, 'Voltaren Emulgel',48.00,1,'romatizma ilacı','jel',10,1 );  
INSERT into MEDICINE VALUES(111, 'Ocuvite Complete',22.00,1,'göz vitamini','tablet',10,1 );  
INSERT into MEDICINE VALUES(112, 'Otrivin',16.00,1,'burun spreyi','sprey',20,1 );  
INSERT into MEDICINE VALUES(113, 'Benexol b12',10.00,1,'vitamin ilacı','tablet',35,1 );  
INSERT into MEDICINE VALUES(114, 'Asiviral Krem',10.00,1,'uçuk kremi','krem',20,1 );  
INSERT into MEDICINE VALUES(115, 'Terramycin',13.00,1,'yara merhemi','krem',10,1 );  
INSERT into MEDICINE VALUES(116, 'Siprogut',9.00,1,'kulak damlası','damla',10,1 );  
INSERT into MEDICINE VALUES(117, 'Glifor',19.00,1,'şeker ilacı','tablet',10,1 );  
INSERT into MEDICINE VALUES(118, 'Ritalin',30.00,0,'uyuşturucu','tablet',10,1 );  
INSERT into MEDICINE VALUES(119, 'Medikinet Retart',40.00,0,'uyuşturucu','tablet',20,1 );  
INSERT into MEDICINE VALUES(120, 'Prozac',30.00,0,'sakinleştirici','tablet',10,1 );
```

Join



- Outer Right Join

Show medicines which will expire before 2025- 05-05

	Name
1	Sebamed
2	Norvasc
3	Ocuvite Complete
4	Otrivin
5	Benexol b12
6	Medikinet Retart
7	Prozac

```
----- Outer Right Join (Show medicines which will expire before 2025-05-05)
SELECT MEDICINE.Name
FROM MEDICINE
RIGHT JOIN PRODUCTION ON MEDICINE.Barcode = PRODUCTION.Barcode
Where PRODUCTION.Last_use_date < '2025-05-05'
```

View

- View 1

View of customers' informations with their doctors

	D_name	D_surname	D_phone	Speciality	C_name	C_address	C_phone
1	Mehmet	Öz	777777	Genel Cerrahi	Yaprak Şahin	Kadıköy	555-666
2	Manolya	Kaya	222222	Psikiyatri	Safiye Derenoğlu	Beyoğlu	222-333
3	Ayşe	Demir	666666	Göz Hastalıkları	Sevgi Akın	Tuzla	111-222
4	Manolya	Kaya	222222	Psikiyatri	Yusuf Yıldız	Üsküdar	999-999
5	Yaprak	Şahin	111000	Ortopedi	Fatma Kara	Şişli	333-555
6	Hatice	Koç	888888	Endokrinoloji	Mustafa Çetin	Beykoz	444-111
7	Meryem	Çelik	444444	Kadın Hastalıkları	Emine Kılıç	Ataşehir	999-000
8	Ali	Arslan	555555	Dahiliye	Zeynep Yıldırım	Bakırköy	777-555
9	Manolya	Kaya	222222	Psikiyatri	Gülben Derenoğlu	Ümraniye	888-222
10	Burak	Saz	111111	KBB	Onur Yılmaz	Üsküdar	111-999

```
----- VIEW 1 (view of customers' informations with their doctors) -----
```

```
CREATE VIEW view1 AS
```

```
SELECT DOCTOR.D_name, DOCTOR.D_surname, DOCTOR.D_phone, DOCTOR.Speciality, CUSTOMER.C_name, CUSTOMER.C_address, CUSTOMER.C_phone
```

```
FROM DOCTOR, CUSTOMER
```

```
WHERE DOCTOR.Doctor_id = CUSTOMER.DoctorID;
```

Additional Things

- Inheritance → EMPLOYEE is a subclass of PHARMACY
- Weak entities → CONTAIN, SUPPLY, and PRODUCTION tables are weak entities.
They depend on some other tables, without other tables they cannot exist.

Thank you for
listening to my
presentation i
hope u enjoyed!