Task 1

Relational Schema

```
Patient(ssn, name, dob)
Primary Key ssn


Doctor(ssn, name, startdate)
Primary Key ssn


Specialist(ssn, specialization)
Primary Key ssn
Foregin Key ssn
    references Doctor(ssn)


generalPhysician(ssn, certification_date)
Primary Key ssn
Foregin Key ssn
    references Doctor(ssn)


Drug(trademark, formula)
Primary Key trademark

Prescirbes(doctor_ssn, patient_ssn, drug_trademark, pdate, quantity)
Primary Key (doctor_ssn, patient_ssn, drug_trademark, pdate)
Foregin Key doctor_ssn
    references Doctor(ssn)
Foregin Key patient_ssn
    references Patient(ssn)
Foregin Key drug_trademark
    references Drug(trademark)


Pharmacy(phone, name, address, street, city)
Primary Key phone


PharmaCompany(name, phone [1..3])
Primary Key name


Sells(pharmacy_phone, drug_trademark, company_name)
Primary Key (pharmacy_phone, drug_trademark, company_name)
Foregin Key company_name
    references PharmaCompany(name)
Foregin Key pharmacy_phone
    referances Pharmacy(phone)
Foregin Key drug_trademark
    references Drug(trademark)


SoldBy(company_name, drug_trademark)
Primary Key (company_name, drug_trademark)
Foreign Key company_name
    references PharmaCompany(name)
Foregin Key drug_trademark
    references Drug(trademark)
```

```
Contracts(company_name, pharmacy_phone, s_date, e_date, text)
Primary Key (company_name, pharmacy_phone, s_date)
Foregin Key comapny_name
    references PharmaCompany(name)
Foregin Key pharmacy_phone
    references Pharmacy(phone)


Employee(ssn, name)
Primary Key ssn


Works(employee_ssn, pharmacy_phone)
Primary Key (employee_snn)
Foregin Key employee_ssn
    references Employee(ssn)
Foregin Key pharmacy_phone
    references Pharmacy(phone)


Supervisor(supervisor_ssn, s_date)
Primary Key supervisor_ssn
Foregin Key supervisor_ssn
    references Employee(ssn)


Sees(patient_ssn, doctor_ssn)
Primary Key (patient_ssn, doctor_ssn)
Foregin Key patient_ssn
    references Patient(ssn)
Foregin Key doctor_ssn
    references Doctor(ssn)
```

Database Code

```
CREATE DATABASE data_base;
USE data_base;

CREATE TABLE PharmaCompany (
    name VARCHAR(255) PRIMARY KEY NOT NULL,
    phone VARCHAR(10) NOT NULL
);

CREATE TABLE Pharmacy (
    phone VARCHAR(10) PRIMARY KEY NOT NULL,
    name VARCHAR(255) NOT NULL,
    address VARCHAR(255) NOT NULL,
    street VARCHAR(255) NOT NULL,
    city VARCHAR(255) NOT NULL
);

CREATE TABLE Drug (
    trademark VARCHAR(255) PRIMARY KEY NOT NULL,
    formula TEXT NOT NULL
);

CREATE TABLE Sells (
    company_name VARCHAR(255) NOT NULL,
    pharmacy_phone VARCHAR(10) NOT NULL,
    drug_trademark VARCHAR(255) NOT NULL,
    PRIMARY KEY (company_name, pharmacy_phone, drug_trademark),
    FOREIGN KEY (company_name) REFERENCES PharmaCompany(name) ON DELETE CASCADE,
    FOREIGN KEY (pharmacy_phone) REFERENCES Pharmacy(phone) ON DELETE CASCADE,
    FOREIGN KEY (drug_trademark) REFERENCES Drug(trademark) ON DELETE CASCADE
);

CREATE TABLE SoldBy (
    company_name VARCHAR(255) NOT NULL,
    drug_trademark VARCHAR(255) NOT NULL,
    PRIMARY KEY (drug_trademark, company_name),
    FOREIGN KEY (company_name) REFERENCES PharmaCompany(name) ON DELETE CASCADE,
    FOREIGN KEY (drug_trademark) REFERENCES Drug(trademark) ON DELETE CASCADE
);

CREATE TABLE Contracts (
    company_name VARCHAR(255) NOT NULL,
    pharmacy_phone VARCHAR(255) NOT NULL,
    s_date DATE,
    e_date DATE,
    text TEXT,
    PRIMARY KEY (company_name, pharmacy_phone, s_date),
    FOREIGN KEY (company_name) REFERENCES PharmaCompany(name) ON DELETE CASCADE,
    FOREIGN KEY (pharmacy_phone) REFERENCES Pharmacy(phone) ON DELETE CASCADE
);
INSERT INTO pharmacy (phone, name, address, street, city)
VALUES
    ('54327612', 'City Drug', '36 South Cherry', 'Starkville', 'MS 39759'),
    ('87435217', 'Pill Pack', '29 E. Pine Lane', 'Stuart', 'FL 34997'),
    ('98463251', 'Better Life', '8004 Eagle St.', 'Sarasota', 'FL 34231'),
    ('45362819', 'Pharma Best', '15 Williams Drive', 'Elgin', 'IL 60120'),
    ('87340213', 'Be Well', '790 Clay Road', 'Ooltewah', 'TN 37363'),
    ('35446281', 'Absolute Care', '39 Spruce Drive', 'Charlottesville', 'VA 22901');

INSERT INTO drug (trademark, formula)
VALUES
    ('Ultran', 'tramadol'),
    ('Advil', 'ibuprofen'),
    ('Aleve', 'naproxen'),
    ('Bayer Aspirin', 'aspirin'),
    ('Zipsor', 'diclofenac'),
    ('Irenka', 'duloxetine'),
    ('Myoflex', 'Trolamine salicylate');

INSERT INTO pharmacompany (name, phone)
VALUES
    ('Janson & Janson', '23749912'),
    ('Pfizer', '45732810'),
    ('Bayer', '88374291'),
    ('Roche', '66372910'),
    ('Abbott', '66392014'),
    ('Allergan', '47639201'),
    ('CSL', '84192200'),
    ('Vertex Pharmaceuticals', '91228345');

INSERT INTO contracts (company_name, pharmacy_phone)
VALUES
    ('Janson & Janson', '54327612'),
    ('Janson & Janson', '54327612'),
    ('Pfizer', '54327612'),
    ('Bayer', '87340213'),
    ('Roche', '35446281'),
    ('CSL', '98463251'),
    ('Abbott', '87340213'),
    ('Vertex Pharmaceuticals', '87340213'),
    ('Allergan', '98463251'),
    ('Allergan', '35446281');

INSERT INTO soldby (company_name, drug_trademark)
VALUES
    ('anson & Janson', 'Ultram'),
    ('Janson & Janson', 'Aleve'),
    ('Janson & Janson', 'Zipsor'),
    ('Janson & Janson', 'Myoflex'),
    ('Pfizer', 'Ultram'),
    ('Pfizer', 'Zipsor'),
    ('Bayer', 'Bayer Aspirin'),
    ('Roche', 'Irenka'),
    ('CSL', 'Ultram'),
    ('Abbott', 'Aleve'),
    ('Vertex Pharmaceuticals', 'Irenka'),
    ('Allergan', 'Advil');

INSERT INTO sells (company_name, pharmacy_phone, drug_trademark)
VALUES
    ('54327612', 'Ultram', 'Pfizer'),
    ('54327612', 'Aleve', 'Abbott'),
    ('87340213', 'Aleve', 'Abbott'),
    ('35446281', 'Advil', 'Allergan'),
    ('98463251', 'Advil', 'Allergan'),
    ('35446281', 'Irenka', 'Vertex Pharmaceuticals');
```

Task 2

limo_id	journey_date	start_time	limo_reg	class	driver_id	price	driver_name
L1	20.02.21	10.00	DN3526	8	1	400	D1
L1	20.02.21	13.00	DN3526	8	1	400	D1
L1	21.02.21	10.00	DN3526	8	1	400	D1
L2	20.02.21	10.00	CY2534	12	2	600	D2
L2	22.02.21	14.00	CY2534	12	2	600	D2
L2	23.02.21	11.00	CY2534	12	2	600	D2

1. What should be the primary key of the table?

For the primary key we will need something that uniquely identifies a row. Limo_id uniquely identifies the limo, but this is not enough as there can be multiple drives done by a limo per day. Start_time uniquely identifies when the drive took place, but again this is not enough as the drives can take place over multiple days. So finally we need journey_date aswell so that we can uniquely identify a drive:

```
PRIMARY KEY (limo_id, journey_date, start_time)
```

2. List the functional dependencies related to the table.

The following depends on **limo_id**

limo_registration | limo_capacity | class | price (NOK) |

The following depens on **driver_id**

driver_name |

The following depens on **class**

price (NOK) |

3. In which normal form is this relation? Explain your answer.

1NF The table is 1NF because:

- There are no repeating groups.
- Each field contains atomic values.

2NF The table is not 2NF because:

- Partial dependencies exist.

3NF The table is not 3NF because:

- There are transantive dependencies such as class -> price, which depend on the primary key

4. Convert the table to 3NF.

To make the table into 3NF we need to remove **transitive dependencies** We can do this by splitting the table into multiple parts:

Limo Table:

| **limo_id** | **limo_registration** | **limo_capacity** | **class** |

```
PRIMARY KEY limo_id
```

Driver Table:

| **driver_id** | **driver_name** |

```
PRIMARY KEY driver_id
```

Class Table:

| **class** | **price** |

```
PRIMARY KEY class
```

And finally a Trip table that connects all the tables (main table)

Trip Table:

| **limo_id** | **journey_date** | **start_time** | **driver_id** |

```
PRIMARY KEY (limo_id, journey_date, start_time)<br>

FOREIGN KEY (limo_id) REFERENCES Limo_table(limo_id)<br>

FOREIGN KEY (driver_id) REFERENCES Driver_table(driver_id)
```

5. Are the tables you created in task 4 in BCNF too? Convert the tables to BCNF if not.

Is the table BCNF no. For a table to be **BCNF**, it must meet **3NF**, and for every functional dependency (X -> Y), X must be a superkey

Our table is not a BCNF becuase not all the function dependencies have a superkey e.g,

driver_id -> driver_name, where driver_id is not a superkey

Converting to BCNF:

Limo Table:

| **limo_registration** | **limo_capacity** | **class** |

```
PRIMARY KEY limo_registration
```

Driver Table:

| **driver_id** | **driver_name** |

```
PRIMARY KEY driver_id
```

Class Table:

| **class** | **price** |

```
PRIMARY KEY class
```

Trip Table:

| limo_id | journey_date | start_time | limo_registration | driver_id |

```
PRIMARY KEY (limo_id, journet_date, start_time)<br>
```

```
FOREIGN KEY (limo_registration) REFERENCES Limo_table(limo_registration) ON DELETE CASCADE<br>
```

```
FOREIGN KEY (driver_id) REFERENCES Driver_table(driver_id) ON DELETE CASCADE
```