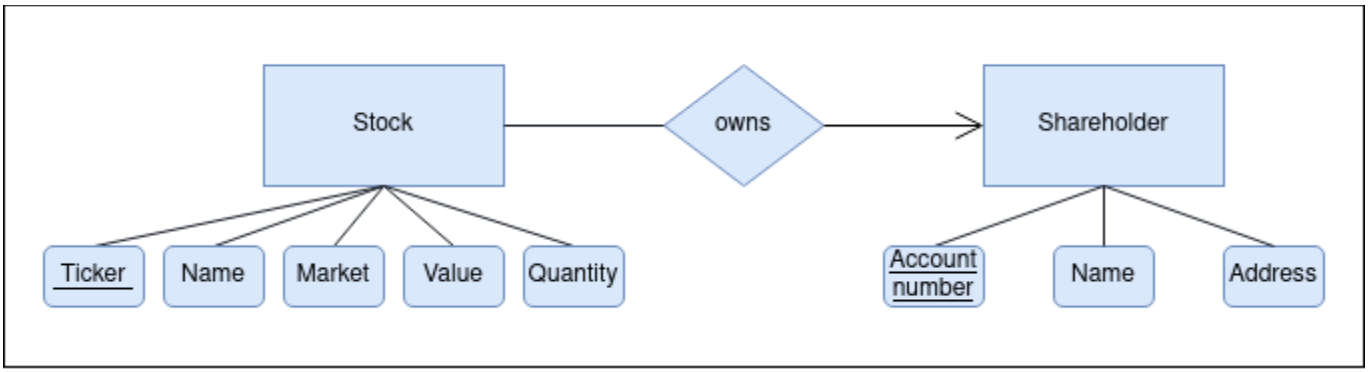
1.

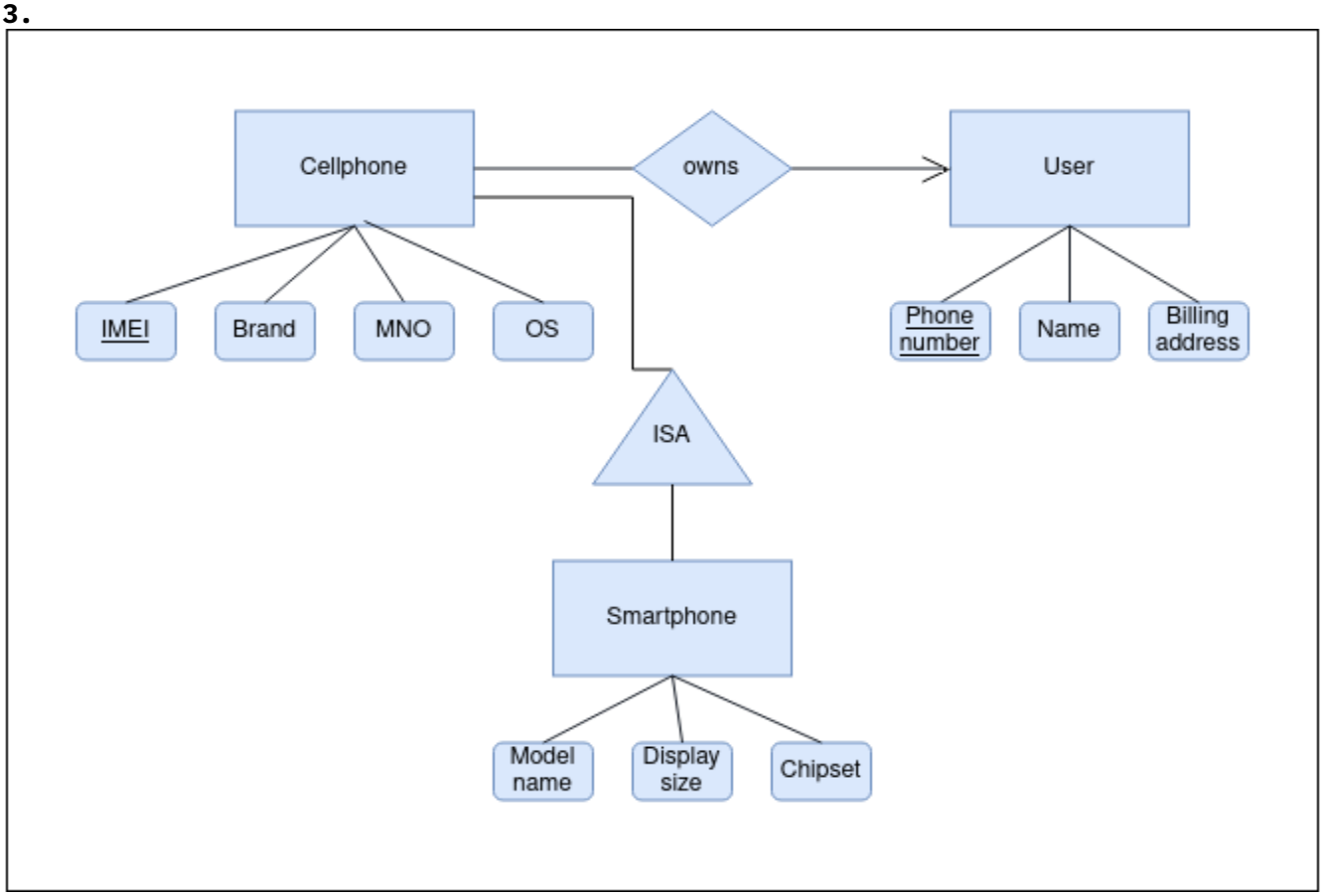

Stock
owns
Shareholder
Ticker
Name
Market
Value
Quantity
Account number
Name
Address

2.

```
CREATE TABLE Stocks (
      ticker VARCHAR(10) ALWAYS AS IDENTITY,
      name VARCHAR(255) NOT NULL,
      market VARCHAR(255) NOT NULL,
      value FLOAT NOT NULL,
      quantity INT NOT NULL,
      PRIMARY KEY(ticker)
);

CREATE TABLE Shareholders (
      account_number INT ALWAYS AS IDENTITY
      owned_ticker VARCHAR(10) ALWAYS AS IDENTITY,
      name VARCHAR(255),
      address VARCHAR(255),
      PRIMARY KEY(account_number)
      PRIMARY KEY(owned_ticker)
      CONSTRAINT fk_stocks
            FOREIGN KEY(owned_ticker)
                  REFERENCES Stocks(ticker)
);
```

3.


Cellphone
owns
User
IMEI
Brand
MNO
OS
Phone number
Name
Billing address
ISA
Smartphone
Model name
Display size
Chipset

4. Map using the nulls method

Table(Phone number, Name, IMEI, Brand, MNO, OS, Model name, Display size, Chipset)

```
CREATE TABLE Phones (
      phone_number INT ALWAYS AS IDENTITY,
      name VARCHAR(255),
      billing_address VARCHAR(255),
      imei INT NOT NULL,
      brand VARCHAR(255) NOT NULL,
      mno VARCHAR(255),
      os VARCHAR(255),
      model_name
      PRIMARY KEY(ticker)
);
```

5. Map using the ER method

User(Phone number, IMEI, Name, Billing address)
Cellphone(IMEI, Brand, MNO, OS)
Smartphone(IMEI, Model name, Display size, Chipset)

```
CREATE TABLE Users (
      phone_number INT ALWAYS AS IDENTITY,
      imei INT,
      name VARCHAR(255),
      billing_address VARCHAR(255),
      imei INT NOT NULL,
      brand VARCHAR(255) NOT NULL,
      mno VARCHAR(255),
      os VARCHAR(255),
      model_name
      PRIMARY KEY(phone_number),
      CONSTRAINT fk_cellphones
            FOREIGN KEY(imei)
                  REFERENCES Cellphones(imei)
);

CREATE TABLE Cellphones (
      imei INT ALWAYS AS IDENTITY,
      brand VARCHAR(255) NOT NULL,
      mno VARCHAR(255),
      os VARCHAR(255),
      PRIMARY KEY(imei)
      CONSTRAINT fk_smartphone
            FOREIGN KEY(imei)
                  REFERENCES Smartphones(imei)
);

CREATE TABLE Smartphone (
      model_name VARCHAR(255) ALWAYS AS IDENTITY,
      imei INT NOT NULL,
      display_size FLOAT,
      chipset VARCHAR(255),
      PRIMARY KEY(imei),
      CONSTRAINT fk_cellphone
            FOREIGN KEY(imei)
                  REFERENCES Cellphones(imei)
);
```

6.

Person(name, address, child_of_name, child_of_address, mother_of_name, mother_of_address, father_of_name, father_of_address, married_to_name, married_to_address)

```
CREATE TABLE Person (
      name VARCHAR(255) ALWAYS AS IDENTITY,
      address VARCHAR(255) ALWAYS AS IDENTITY,
      child_of_name VARCHAR(255),
      child_of_address VARCHAR(255),
      mother_of_name VARCHAR(255),
      mother_of_address VARCHAR(255),
      father_of_name VARCHAR(255),
      father_of_address VARCHAR(255),
      married_to_name VARCHAR(255),
      married_to_address VARCHAR(255),
      PRIMARY KEY(name),
      PRIMARY KEY(address),
);
```

7.

Person(name, address)
Child(name, address, child_of_name, child_of_address)
Mother(name, address, mother_of_name, mother_of_address, married_to_name, married_to_address)
Father(name, address, father_of_name, father_of_address, married_to_name, married_to_address)

```
CREATE TABLE Person (
        name VARCHAR(255) ALWAYS AS IDENTITY,
        address VARCHAR(255) ALWAYS AS IDENTITY,
        PRIMARY KEY(name),
        PRIMARY KEY(address),
);

CREATE TABLE Child (
        name VARCHAR(255) ALWAYS AS IDENTITY,
        address VARCHAR(255) ALWAYS AS IDENTITY,
        child_of_name VARCHAR(255),
        child_of_address VARCHAR(255),
        PRIMARY KEY(name),
        PRIMARY KEY(address),
        CONSTRAINT fk_person_name
                FOREIGN KEY(name)
                        REFERENCES Person(name)
        CONSTRAINT fk_person_address
                FOREIGN KEY(address)
                        REFERENCES Person(address)
);

CREATE TABLE Mother (
        name VARCHAR(255) ALWAYS AS IDENTITY,
        address VARCHAR(255) ALWAYS AS IDENTITY,
        mother_of VARCHAR(255),
        mother_of_address VARCHAR(255),
        married_to_name VARCHAR(255),
        married_to_address VARCHAR(255),
        PRIMARY KEY(name),
        PRIMARY KEY(address),
        CONSTRAINT fk_child_name
                FOREIGN KEY(name)
                        REFERENCES Child(name)
        CONSTRAINT fk_child_address
                FOREIGN KEY(address)
                        REFERENCES Child(address)
        CONSTRAINT fk_married_to_name
                FOREIGN KEY(married_to_name)
                        REFERENCES Father(name)
        CONSTRAINT fk_married_to_address
                FOREIGN KEY(married_to_address)
                        REFERENCES Father(address)
);

CREATE TABLE Father (
        name VARCHAR(255) ALWAYS AS IDENTITY,
        address VARCHAR(255) ALWAYS AS IDENTITY,
        father_of VARCHAR(255),
        father_of_address VARCHAR(255),
        married_to_name VARCHAR(255),
        married_to_address VARCHAR(255),
        PRIMARY KEY(name),
        PRIMARY KEY(address),
        CONSTRAINT fk_child_name
                FOREIGN KEY(name)
                        REFERENCES Child(name)
        CONSTRAINT fk_child_address
                FOREIGN KEY(address)
                        REFERENCES Child(address)
        CONSTRAINT fk_married_to_name
                FOREIGN KEY(married_to_name)
                        REFERENCES Mother(name)
        CONSTRAINT fk_married_to_address
                FOREIGN KEY(married_to_address)
                        REFERENCES Mother(address)
);
```