Question 1

R(A, B, C, D, E, F, G, H)
Functional Dependencies:

- A → D, E
- C → G
- A, C → H, F

Part 1: is this relation in $_1NF$?
Yes, This relation is in $_1NF$. There are no repeating groups here.

Part 2: is this relation in $_2NF$?
Since we have yet to establish a primary key, we can't determine if our relation is in $_2NF$. Our only real candidate key here is {A, C, B}. So we make that our primary key.

Now that we have our primary key, we can clearly see that we our relation is not in $_2NF$ since two subsets of our primary key (A and C) are determinates of non-prime attributes.

$A \rightarrow D,\ E$ (violates $_2NF$).
$C \rightarrow G$ (violates $_2NF$).

By performing decomposition, we can move to $_2NF$. We need to perform decomposition on all functional dependencies that violate $_2NF$. There are three cases that violates $_2NF$, so after decomposition we will be left with four relations instead of one.

Schema in $_2NF$:

- → R_1(A, B, C,)
- → R_2(A, D, E)
- → R_3(C, G)
- → R_4(A, C,H,F)

Part 3: is this relation in $_3NF$?
Yes, we are already in $_3NF$

Question 2

Property(id, county, lotNum, lotArea, price, taxRate, (dataPaid, amount))

Functional Dependencies:

- id → county, lotNum, lotArea, price, taxRate
- lotArea → price
- county → taxRate
- id, dataPaid → amount

`Part 1:` is this relation in ${}_1NF$?
No, there are repeating groups here, so we are not in ${}_1NF$. To fix this, we need to select a primary key that makes every value atomic.

Making the primary key {`id, dataPaid`} brings us to ${}_1NF$. Our new schema and table now look like this.
Property(id, county, lotNum, lotArea, price, taxRate, dataPaid, amount).

id	county	lotNum	lotArea	price	taxRate	datePaid	amount.
001	Will	59	G5	$3500	1%	02-05-2012	$1200
001	Will	59	G5	$3500	1%	02-20-2012	$2300

`Part 2:` is this relation in ${}_2NF$?
No, this relation is not in ${}_2NF$. The current state of our relation does not have full dependency. Looking at our functional dependencies, we see that a subset of our primary key is the determinent of some non-prime attributes. This violates the rules of ${}_2NF$.

id → county, lotNum, lotArea, price, taxRate. (violates ${}_2NF$).

To fix this, we should use decomposition, which splits our relation into two tables:
Property(id, county, lotNum, lotArea, price, taxRate)

id	county	lotNum	lotArea	price	taxRate

Payments(id, datePaid, amount)

id	datePaid	amount

Part 3: is this relation in $_3NF$?
No, We are not in $_3NF$ because there are some transitive dependencies in our list of functional dependencies. county functionally determining taxRate, and lotArea functionally determining price are both transitive dependencies, which violates $_3NF$.

lotArea → price (violates $_3NF$)
county → taxRate (violates $_3NF$)

To fix this, we need to perform decomposition on all cases that violate $_3NF$. That leaves us with:

Property(id, lotNum).
Payments (id, datePaid, amount)
LandInfo(lotArea, price)
TaxInfo(county, taxRate)

Question 3

Pharmacy(patient_id, patient_name, address, (Rx_num, trademark_name, generic_name, (filldate, num_refills_left), num_refills))

Functional Dependencies:

- patient_id → patient_name, address
- patient_id, Rx_num → trademark_name, generic_name
- Rx_num → num_refills
- Rx_num, filldate → num_refills_left

Part 1: is this relation in $_1NF$?
No, This relation is not in $_1NF$. There are many repeating groups here, so not all values are atomic. We can move to $_1NF$ by selecting a primary key that makes all values atomic.

The only real candidate key here is {patient_id, Rx_num, filldate}. Since it is the only candidate, we will make it the primary key.

Our new schema is:

Pharmacy(patient_id, patient_name, address, Rx_num, trademark_name, generic_name, filldate, num_refills_left, num_refills)

Part 2: is this relation in $_2NF$?
This relation is not in $_2NF$ since we have subsets of the primary key that are able to functionally determine non-prime attributes.

patient_id → patient_name, address (violates $_2NF$)
patient_id, Rx_num → trademark_name, generic_name (violates $_2NF$)
Rx_num, → num_refills (violates $_2NF$)
Rx_num, filldate → num_refills_left (violates $_2NF$)

After decomposition, we have:

Schema in $_2NF$:

- → ***Pharmacy***(patient_id, Rx_num, fill_date)
- → ***Patient***(patient_id, patient_name, address)
- → ***Prescriptions***(patient_id, Rx_num, trademark_name, generic_name)
- → ***TotalRefills***(Rx_num, num_refills)
- → ***RemainingRefills***(Rx_num, filldate, num_refills_left)

Part 3: is this relation in $_3NF$?
Yes, we are already in $_3NF$.

Question 4

Company(EmpID, EmpName, EmpAddr, (ProjID, ProjName, MgrID, MgrName, HoursWorked))

Functional Dependencies:

- ➤ EmpID → EmpName, EmpAddr
- ➤ ProjID → ProjName, MgrID, MgrName
- ➤ EmpID, ProjID → HoursWorked
- ➤ MgrID → MgrName

Part 1: is this relation in $_1NF$?
This relation is not in $_1NF$ since there are repeating groups present. We need a primary key that is able to functionally determine all other attributes. The primary key we should go with in this case is: {EmpID, ProjID}.

This primary key gives us the schema:
Company(EmpID, EmpName, EmpAddr, ProjID, ProjName, MgrID, MgrName, HoursWorked)

Part 2: is this relation in ${}_2NF$?
This relation is not in ${}_2NF$ since some subsets of the primary key can functionally determine non-prime attributes.

EmpID → EmpName, EmpAddr (violates ${}_2NF$)
ProjID → ProjName, MgrID, MgrName (violates ${}_2NF$)

We can fix this by performing decomposition on every case that violates ${}_2NF$. This leaves us with the following schema:

- ***Company***(EmpID, ProjID,HoursWorked)
- ***Employee***(`EmpID`, EmpName, EmpAddr)
- ***Project***(`ProjID`, ProjName, MgrID, MgrName)

Part 3: is this relation in ${}_3NF$?
No, This relation is not in ${}_3NF$ since we have a non-prime attribue (`MgrID`), that can functionally determine another non-prime attribute (MgrName). To move into ${}_3NF$, we need to perform more decomposition. We need to remove `MgrID` and `MgrName` from the **Project** relation, and move it into a new relation. That leaves us with:

- ***Company***(EmpID, ProjID,HoursWorked)
- ***Employees***(`EmpID`, EmpName, EmpAddr)
- ***Projects***(`ProjID`, ProjName)
- ***Managers***(MgrID, MgrName)

Question 5

StockExchange(Company, Symbol, HQ, Date, ClosePrice)

Functional Dependencies:

- Symbol, Date → Company, HQ, ClosePrice
- Symbol → Company, HQ
- Symbol → HQ

Part 1: is this relation in ${}_1NF$?
There are no repeating groups present, so this relation is in ${}_1NF$.

However, to move on any further, we need a primary key so we should establish one. Our only candidate key here is {Symbol, Date}. We will make this our primary key. Our new schema is:

StockExchange(Company, Symbol, HQ, Date, closePrice)

Part 2: is this relation in ${}_2NF$?
This relation is not in ${}_2NF$ since a subset of the primary key is a determinent of non-key attributes.

Symbol $\rightarrow$ Company, HQ (violates ${}_2NF$)
Symbol $\rightarrow$ HQ (violates ${}_2NF$)

After performing the decomposition steps for every violation of ${}_2NF$ (two in this case), we are left with three relations:

StockExchange(Symbol, Date, closePrice)
StockCompany(Symbol, Company, HQ)
StockLocation(Symbol, HQ)

Part 3: is this relation in ${}_3NF$?
Yes, this relation is in ${}_3NF$