Jacob Kurbis (Z1945650)
CSCI 466 Section 2
Normalization Assignment 3

1. Stock Exchange(Company, Symbol, HQ, Date, ClosePrice)
Functional Dependencies:
‣Symbol, Date ⟶ Company, HQ, ClosePrice
‣Symbol ⟶ Company, HQ
‣Symbol ⟶ HQ

a). This is in 1NF

b). This is not in 2NF. The second and third dependencies suggest that Company and HQ can be determined by part of the primary key Symbol, which are partial dependencies. This can be fixed by decomposing into two separate relations and their functional dependencies.

Stock Price(<u>Symbol</u>, <u>Date</u>, ClosePrice)
Symbol,Date → ClosePrice

Company Info(<u>Symbol</u>, Company, HQ)
Symbol → Company, HQ

c). This is in 3NF

2. Company(EmpID, EmpName, EmpAddr,(ProjID, ProjName, MgrID, MgrName, HoursWorked))
Functional Dependencies:
‣EmpID ⟶ EmpName, EmpAddr
‣ProjID ⟶ ProjName, MgrID, MgrName
‣EmpID, ProjID ⟶ HoursWorked
‣MgrID ⟶ MgrName

a). This is not in 1NF. The Company relation violates 1NF due to repeating values in the nested tuple(ProjID, ProjName, MgrID, MgrName, HoursWorked). To fix this, the relation can be decomposed into distinct "Employee" and "Project" relations, identified by EmpID and ProjID, respectively.

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

EmpID, ProjID ⟶ HoursWorked
MgrID ⟶ MgrName

b). Based on the 1NF of the database, it is in 2NF.

c). This is not in 3NF. MgrID is a non-prime attribute and has a transitive dependency because MgrID determines MgrName based on the last dependency. To fix this, the second dependency can be decomposed to remove the transitive dependency.

Employee(EmpID, EmpName, EmpAddr)
EmpID ⟶ EmpName, EmpAddr

Project(ProjectID, ProjName, MgrID)
ProjID ⟶ ProjName, MgrID

Manager(MgrID, MgrName)
MgrID ⟶ MgrName

Work(EmpID, ProjID, HoursWorked)
EmpID, ProjID ⟶ HoursWorked

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

a). This is not 1NF. The attributes in the parenthesis are not atomic. To fix it, add another Primary Key such as Rx_num and filldate.

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

b). This is not 2NF. Trademark name is not dependent on num_refills or num_refills_left. We can fix this by creating a new composite key for this as well.

Patient Info(patient_id, patient_name, address)
patient_id ⟶ patient_name, address

Prescription Info(Rx_num, trademark_name, generic_name, num_refills)
Rx_num ⟶ trademark_name, generic_name, num_refills

Refill Info(Rx_num, filldate, num_refills_left)
Rx_num, filldate ⟶ num_refills_left

Patient Prescription(patient_id, Rx_num)

c). This is in 3NF

4. R(A,B,C,D,E,F,G,H)
Functional Dependencies:
▸A ⟶ D, E
▸C ⟶ G
▸A,C ⟶ H, F

a). This is in 1NF(Assuming that B is also part of the Primary Key)

b). This is not in 2NF, because D and E are partially dependent on A which is part of the candidate key, and C ⟶ G violates 2NF for the same reason. It can potentially be fixed by decomposition like so.

R1(A,D,E)
A ⟶ D, E

R2(C,G)
C ⟶ G

R3(A,B,C,F,H)
A,C ⟶ H, F

c). This is in 3NF.

5). Property (id, county, lotNum, lotArea, price, taxRate, (datePaid,amount))
Functional Dependencies:
▸id ⟶ county, lotNum, lotArea, price, taxRate
▸lotArea ⟶ price
▸county ⟶ taxRate

▸id, datePaid ⟶ amount

a). This is not in 1NF. The nested relation (datePaid,amount) implies non-atomic values which violate the requirement of 1NF, so a new primary key of id and datepaid could help fix this.

Property (id, county, lotNum, lotArea, price, taxRate, datePaid, amount)
id ⟶ county, lotNum, lotArea, price
lotArea ⟶ price
county ⟶ taxRate
id, datePaid ⟶ amount

b). This is not in 2NF. Price is dependent on lotArea and lotArea is not part of the candidate key. As well as taxRate being dependent on county and county not being part of the candidate key. We can fix this by removing these partial dependencies.

Property Details(id, lotNum lotArea, county, datePaid, amount)
id ⟶ county, lotNum, lotArea
id, datePaid ⟶ amount

Area Price(lotArea, price)
lotArea ⟶ price

CountyTax(county, taxRate)
county ⟶ taxRate

c). This is in 3NF