Justin McLain
Assignment 2
CSCI 466

1. Company (EmpID, EmpName, EmpAddr, (ProjID, ProjName, MgrID, MgrName, HoursWorked))
    a. No this is in not in first normal form (1NF) to fix this we must assign a primary key to allow no repeating groups. To do this make EmpID and ProjID the primary key each of the values within the cells of this relation are atomic with EmpID and ProjID being the primary key with these the attributes are not repeating groups. So, it will look like (EmpID, EmpName, EmpAddr, ProjID, ProjName, MgrID, MgrName, HoursWorked)
    b. No, it is not in second normal for (2NF) this will have update anomalies because there is redundancy also along with an insertion anomaly where we can't add anything until the HoursWorked is added so if a new employee worked then we couldn't add them until they submitted their work hours. Also, it has deletion anomalies because if an Employee were fired then the there would be missing data for the ProjID. The functional dependency violations are ProjID -> ProjName, MgrID, MgrName and EmpID -> EmpName, EmpAddr. To fix this we must decompose, create a few new relations I will call the first EmpProj and **EmpProj** -> (EmpID, ProjID, HoursWorked). Next, I will make one called Project and **Project** -> (ProjID, ProjName) and lastly I will make one called Employee, **Employee** -> (EmpID, MgrID ,MgrName).
    c. No it is not in 3NF because there is transitive issues, meaning there is redundancies. To be in 3NF you must have no non-prime attributes determine any other non-prime attributes. Because MgrID -> MgrName this is what violates the 3NF. So again, we decompose. I will make a new relation called **Manage** and **Manage** -> (MgrID, MgrName). And I will also change **Employee** -> (EmpID, MgrID)

2. **StockExchange**(Company, Symbol, HQ, Date, ClosePrice)
    a. Yes, it is in first normal form because there are no repeating groups (all the values are atomic) and the Primary Key is (Symbol , Date).
    b. No, it is not in 2NF because Date is a partial key dependency it says that to determine company HQ and ClosePrice Date must be involved so to fix this we will decompose. I will make a relation called **CompInfo** -> (Symbol, Company, HQ) and then I will make relation with the name **Deal** -> (Date, ClosePrice). With this there is no non-prime attributes functionally determined by a subset of the primary key. And the non-prime attributes are fully

dependent on the Symbol, Date primary key with no partial key dependencies. And it is in 1NF from before.

c. Yes, it is in 3NF because there are no non-prime attributes determining other non-prime attributes (no transitive dependencies) and it is in 2NF.

3. **Property**(id, county, lotNum, lotArea, price, taxRate, (datePaid, amount))
   a. No, it is not in 1NF because not all the values are atomic (datePaid, amount)) to fix this make ID, datePaid the primary keys so it will look like this **Property**(id, county, lotNum, lotArea, price, taxRate, datePaid, amount)
   b. No, it is not in second normal form because there is insertion anomalies, deletion anomalies, and there is partial key dependencies to fix this we will decompose. I will make a relation called **lotInfo** -> (ID, county, LotNum, LotArea) and I will make a relation called. **costDeal** -> (datePaid, taxRate, amount, price). Now there is no partial key dependencies as the primary key determines all non-prime attributes and it is in 1NF already.
   c. No, It is not in 3NF because there is transitive dependencies. The issue is with lotArea and county where lotArea a non-prime determines price another non-prime and county a non-prime determines taxRate also a non-prime. Again, to fix this decomposition is necessary. As well as it being in 2NF. There is already, **lotInfo** -> (ID, county, LotNum, LotArea) and **costDeal** which has been changed to **costDeal** -> (datePaid, amount). For this decomposition make a relation called **locCost** -> (county, taxRate) that solves the issue of county and tax rate then make another relation called **sizeCost** -> (lotArea, price).
4. **Pharmacy**(patient_id, patient_name, address, (Rx_num, trademark_name, generic_name, (filldate, num_refills_left), num_refills))
   a. No this is not in 1NF because there is multiple non atomic values. To fix this make (patient_id , Rx_num, filldate) the primary key. Now it will look like this, **Pharmacy**(patient_id, patient_name, address, Rx_num, trademark_name, generic_name, filldate, num_refills_left, num_refills).
   b. No it is not in second form because there is partial key dependencies with Rx_num and filldate. To fix this it must first be in 1NF and then decomposition is necessary. Create a relation called **patient** ->(patient_id, patient_name, address) create another relation named **drug** -> (Rx_num, trademark_name, generic_name). And a last relation called **prescription** -> (filldate, num_refills, num_refills_left). With this there are no longer any partial key dependencies.

c. Yes it is in 3NF because it is in 2NF and there is no non-prime attributes functionally determining another non-prime attribute (no transitive dependencies).