

1a) This relation is not 1NF. The attributes in parenthesis are not atomic. An employee can be assigned to multiple Projects and have multiple Managers. The best way to fix this would be to have ProjectID as the Primary Key, since one Project can have only one team.

1b) Since this is not 1NF, this is also not 2NF. The reason for this is because all attributes must be dependent of the primary key. If we had the ProjectID as the Primary Key, then that would mean Employee and Manager names would be dependent on the ProjectID, and that is not the case. We would need multiple Primary Keys, such as EmployeeID, ProjectID, and ManagerID.

1c) As in 1b, this is also not 3NF because this is not 2NF. Project Names are repetitive, and will need to be changed in each column for each Employee working in that Project. Therefore, ProjectID can be a Foreign Key, acting as a Primary Key for Project Name.

2a) This is in 1NF

2b) This is in 2NF

2c) This is not 3NF. The Company and HQ is redundant information that can be added to a foreign key, that being Symbol.

3a) This is 1NF

3b) This is 2NF

3c) This is 3NF

4a) This is not 1NF. The attributes in the parenthesis are not atomic. To fix it, add another Primary Key such as RX Num.

4b) This is not 2NF because it is not 1NF. Trademark name is not dependent on rx refills or refills left. We can create a separate key for this as well.

4c) Lastly, this is not 3NF because it is not 2NF. We can create a foreign key in RX Num to hold Trademark name and generic name in case these ever change.