Steps to get to 3NF

1. Create initial relational design

2. Identify the FDs

3. Construct decomposed schema for which:

   - Natural joins do not add spurious tuples
   - All relations are in at least 3NF
   - All FDs are retained or can be reconstructed

Step #1 (Create initial relational design)

Employee (EmployeeID (**PK**), deptID(**FK**), fName, LName, address, salary, jobTitle)

Step #2 (Identify the FDs)

{EmployeeID} $\longrightarrow$ {deptID, fName, LName, address, salary, jobTitle}
{EmployeeID} $\longrightarrow$ {deptID}
{EmployeeID} $\longrightarrow$ {fName}
{EmployeeID} $\longrightarrow$ {LName}
{EmployeeID} $\longrightarrow$ {address}
{EmployeeID} $\longrightarrow$ {salary}
{EmployeeID} $\longrightarrow$ {jobTitle}

I'm going to ignore:
{EmployeeID, deptID} $\longrightarrow$ {fName, LName, address, salary, jobTitle}
and all it's decompositions, because it's just a bunch of partial dependencies.

I think these are the only FDs, because what if George Foreman and his sons all lived in the same house, worked in the same department and made the same amount of money? Any other FDs would be invalid.

Step #3 (1NF)

None of the attributes in Employee are set-valued $\therefore$ Employee is in 1NF.

Step #3 (2NF)

Employee is in 2NF, because every non-prime attribute of the relation is FFD upon a CK, in this case a PK (EmployeeID).

Step #3 (3NF)

All of the FDs in the relation satisfy condition (a) of the definition of 3NF, $\therefore$ Employee is in 3NF.