Find all candidate keys. The candiates keys are { ProdID}, The set of key attributes are: { ProdID }  
for each non-trivial FD, check whether the LHS is a proper subset of some candidate key.

checking FD: ProdID --> ProdName,EnterTime,StoreAmount,InvID

The set of key attributes are: { ProdID } for each FD, check whether the LHS is superkey.  
checking functional dependency ProdID --> ProdName,EnterTime,StoreAmount,InvID

There is no violate 2nd NF and there is no transitivity so it does not violate 3rd NF. Therefore decompose is not necessary.

Find all candidate keys The candiates keys are { CustID}, The set of key attributes are: { CustID }  
for each non-trivial FD, check whether the LHS is a proper subset of some candidate key.

checking FD: CustID --> CustName,CustPhone,CustMail

The set of key attributes are: { CustID } for each FD, check whether the LHS is superkey.

checking functional dependency CustID --> CustName,CustPhone,CustMail

There is no violate 2nd NF and there is no transitivity so it does not violate 3rd NF. Therefore decompose is not necessary.

Find all candidate keys. The candiates keys are { EmpID,ManID}, The set of key attributes are: { EmpID,ManID } for each non-trivial FD, check whether the LHS is a proper subset of some candidate key. checking FD: EmpID,ManID --> EmpNameEmpExp,EmpSalEmpPhone,Empgender

The set of key attributes are: { EmpID,ManID } for each FD, check whether the LHS is superkey.  
checking functional dependency EmpID,ManID --> EmpNameEmpExp,EmpSalEmpPhone,Empgender

There is no violate 2nd NF and there is no transitivity so it does not violate 3rd NF. Therefore decompose is not necessary.

Find all candidate keys. The candiates keys are { InvID}, The set of key attributes are: { InvID }  
for each non-trivial FD, check whether the LHS is a proper subset of some candidate key checking FD: InvID --> InvName,InvSize,CapInf,InvAdress,StoreAmount

The candiates keys are { InvID}, The set of key attributes are: { InvID } for each FD, check whether the LHS is superkey

checking functional dependency InvID --> InvName,InvSize,CapInf,InvAdress,StoreAmount

There is no violate 2nd NF and there is no transitivity so it does not violate 3rd NF. Therefore decompose is not necessary.

Find all candidate keys. The candiates keys are { SaleID,CustID}, The set of key attributes are: { SaleID,CustID } for each non-trivial FD, check whether the LHS is a proper subset of some candidate key

checking FD: SaleID,CustID --> ProdID,EmpID,SalAmount

The candiates keys are { SaleID,CustID}, The set of key attributes are: { SaleID,CustID }  
for each FD, check whether the LHS is superkey

checking functional dependency SaleID,CustID --> ProdID,EmpID,SalAmount

There is no violate 2nd NF and there is no transitivity so it does not violate 3rd NF. Therefore decompose is not necessary.

Find all candidate keys. The candiates keys are { EmpID,dependent\_name}, The set of key attributes are: { EmpID,dependent\_name } for each non-trivial FD, check whether the LHS is a proper subset of some candidate key.  
checking FD: EmpID,dependent\_name --> Relationship

The candiates keys are { EmpID,dependent\_name}, The set of key attributes are: {EmpID,dependent\_name } for each FD, check whether the LHS is superkey or the RHS are all key attributes  
checking functional dependency EmpID,dependent\_name --> Relationship

There is no violate 2nd NF and there is no transitivity so it does not violate 3rd NF. Therefore decompose is not necessary.

Find all candidate keys. The candiates keys are { StoreAmount}, The set of key attributes are: { StoreAmount } for each non-trivial FD, check whether the LHS is a proper subset of some candidate key   
checking FD: StoreAmount --> InvID

The candiates keys are { StoreAmount}, The set of key attributes are: { StoreAmount }  
for each FD, check whether the LHS is superkey checking functional dependency StoreAmount --> InvID

There is no violate 2nd NF and there is no transitivity so it does not violate 3rd NF. Therefore decompose is not necessary.