Megan Gong HW 3

1.

(a)
Assumptions:

- OrderID = (CustomerNumber, OrderDate, OrderTime) #Order identity.
- (CustomerNumber, OrderDate, Time) → Employee # One form serves one employee handling that order
- Customer → Orders, Order → Parts #A single order from the Warehouse can contain multiple parts and a customer can place many orders.
- PartNumber → PartName, Type, CageCode # Part attributes
- (CustomerNumber, OrderDate, Time, PartNumber) → QuantityOrdered, OrderPrice # Order-time pricing on the line

(b) & (c)

Unnormalized relation:

Parts(CustomerNumber, CustomerName, CustomerType, OrderDate, OrderTime, Employee, (PartNumber, PartName,Type, CageCode, QuantityOrdered, OrderPrice))

1NF:

Parts(**CustomerNumber**, CustomerName, CustomerType, **OrderDate**, **OrderTime**, **Employee**, **PartNumber**, PartName,Type, CageCode, QuantityOrdered, OrderPrice)

#PK: CustomerNumber, OrderDate, OrderTime,Employee,PartNumber

2NF:

Customer(**CustomerNumber**, CustomerName, CustomerType)

#PK: CustomerNumber.

Part(**PartNumber**, PartName, Type, CageCode)

#PK: PartNumber.

Order(**CustomerNumber, OrderDate, OrderTime, Employee**)

#PK: (CustomerNumber, OrderDate, OrderTime,).

OrderLine(**CustomerNumber, OrderDate, OrderTime, PartNumber**, QuantityOrdered, OrderPrice)

#PK: (CustomerNumber, OrderDate, OrderTime, PartNumber); FK1: (CustomerNumber, OrderDate, OrderTime) → Order; FK2: PartNumber → Part

3NF (we can not have transitive dependencies):

No transitive dependencies remain (no valid Type → CageCode; cage is per Part).

So, all four tables are in 3NF.

(d)

Final named 3NF relations:

Customer(**CustomerNumber**,CustomerName,CustomerType)

Order(**CustomerNumber, OrderDate, OrderTime, Employee**)

Part(**PartNumber**, PartName, Type, CageCode)

OrderLine(***CustomerNumber, OrderDate, OrderTime, Employee, PartNumber***, QuantityOrdered, OrderPrice)

2.

(a)

Assumptions:

- staffNo → therapistName #staffNo uniquely identifies a therapist.
- patNo → patName, appointment #patNo uniquely identifies a patient and their (single) appointment.
- (staffNo, patNo) → branchNo #the branch where the visit occurs is determined by the pair (staffNo, patNo).

(b) & (c)

Unnormalized relation:

Therapy(staffNo, therapistName, (patNo, patName, appointmenDate, appointmentTime, branchNo))

Reapting group:

(patNo, patName, appointmentDate, appointmentTime, branchNo)

1NF:

AppointmentLine(**staffNo**, therapistName, **appointmentDate**, **appointmentTime**, **patNo**, patName, branchNo)

#PK: staffNo, appointmentDate, appointmentTime, patNo.

2NF(remove partial dependencies on the composite key):

Therapist(**staffNo**, therapistName)

Patient(**patNo**, patName)

TherapistDailyBranch(**staffNo**, **appointmentDate**, BranchNo)

Appointment(**staffNo**, **appointmentDate**, **appointmentTime**, **patNo**)

3NF(remove transitive dependencies):

Therapist(**staffNo**, therapistName)

Patient(**patNo**, patName)

TherapistDailyBranch(***staffNo***, **appointmentDate**, branchNo)

#PK: (staffNo, appointmentDate); FK: staffNo → Therapist.staffNo.

Appointment(***staffNo***, ***appointmentDate***, **appointmentTime**, ***patNo***)

#PK: (staffNo, appointmentDate, appointmentTime, patNo); FK1: (staffNo, appointmentDate) → TherapistDailyBranch(staffNo, appointmentDate); FK2: patNo → Patient.

(d)

Final named 3NF relations:

Therapist(**staffNo**, therapistName)

Patient(**patNo**, patName)

TherapistDailyBranch(***staffNo***, **appointmentDate***,* branchNo)

Appointment***(staffNo, appointmentDate***, **appointmentTime*,*** ***patNo***)

3.

(a)

Assumptions:

- contractNo → eventNo #the event number depend on the contract No.
- eventNo → eventLoc #the event location depends on the event number.
- eNo,contractNo → hours #the hours worked by an employee on an event depends on the employee number and contract No.

So, Primary key of the given Table → eNo, contractNo

(b)
The Given table is in 1NF as there is not multivalued attribute.
For Normalization to 3NF , we first Normalize to 2NF
To get 2NF, we need to remove partial dependency.

2NF:
Employee(**eNo**, eName)
EventContract (**contractNo**, eventNo, eventLoc)
HoursWorked(**eNo**, **contractNo**, hours)

Since the composite primary key is eNo and contractNo, the partial dependencies are eNo to eName and contractNo to eventNo.

(c)
To get 3NF, we need to remove transitive dependency.
We have transitive dependency of eventNo to eventLoc present in Table 2.

3NF:
Employee (**eNo**, eName)
EventContract (**contractNo**, eventNo) #eventNo reference as FK to event location.
HoursWorked (***eNo***, ***contractNo***, hours) #eNo references to employee and contractNo references event contract.
EventLocation (**eventNo**,eventLoc)

Those 4 Tables are in 3NF because they do not contain any partial or transitive dependency.

(d)

The names of the 4 Tables in 3NF are: Employee, EventContract, HoursWorked, EventLocation, which name the tables according to the data it stores.