Assignment 3-Normalization

Exercise 1: Parts Order

a. Assumptions:
   - Multiple parts can be ordered in a single transaction
   - Every employee has a unique name
   - CustomerNumber, Date, and Time together identify an order since there is no orderID provided and we can't add any additional fields.

b. Process of Normalization:

   Initial form:
   CustomerNumber
   CustomerName
   CustomerType
   Date
   Time
   Employee
   PartNumber 1
   PartNumber 2
   PartNumber 3
   PartName …
   PartType …
   CageCode
   QuantityOrdered
   UnitPrice

   The initial form is not in any Normal Form (unnormalized) because there are multiple parts ordered in a single transaction. In order to move it to 1NF, we need to get rid of the repeating groups by creating a new row for each part ordered. So the table would look like this:

   1NF

   customerNumber (PK)
   customerName
   customerType
   date (PK)
   time (PK)
   employee
   partNumber (PK)
   partName
   partType

cageCode
quantityOrdered
unitPrice

This table is not in 2NF because some of the attributes have dependencies on part of the PKs. For example, CustomerName and CustomerType depend only on CustomerNumber. PartName, PartType and cageCode depend only on PartNumber. And the attribute employee depends on CustomerNumber, Date, Time (not PartNumber).
In order to move the table to 2NF, we need to remove partial dependencies by splitting into separate tables. So the tables would look like the following:

2NF

Order_detail
customerNumber (PK)
date (PK)
time (PK)
partNumber (PK)
quantityOrdered
unitPrice

Order
customerNumber (PK)
date (PK)
time (PK)
employee

Customer
customerNumber (PK)
customerName
customerType

Part
partNumber (PK)
partName
partType
cageCode

I would say these tables are already in a pretty good place with no transitive dependencies, except Employee in Order is a name and could have additional attributes (e.g., phone), but since only the name is provided and it's unique, it's separated into its own table to avoid redundancy.

3NF

Order_detail
customerNumber
date
time
partNumber
quantityOrdered
unitPrice

Order
customerNumber
date
time
employee

Customer
customerNumber
customerName
customerType

Part
partNumber
partName
partType
cageCode

Employee
Employee

Exercise 2: Panacea Mental Health Corporation

This table is in 1NF because there are no repeating groups and StaffNo and AppointmentDateTime together uniquely identify an appointment.

a. Assumptions
    - A therapist can't have two appointments at the same exact time
    - Patients can have multiple appointments per day with different therapists.
b. Normalization Process

    The table is not in 2NF because there are partial dependencies. TherapistName depends only on StaffNo, and PatName depends only on PatNo, not on the entire composite keys. In order to move the table to 2NF, we should split them into different tables.

2NF

Appointment
staffNo (PK)
appointmentDateTime (PK)
patNo
branchNo

Therapist
staffNo (PK)
therapistName

Patient
patNo (PK)
patName

No further changes are required because there are no transitive dependencies and the tables are already in 3NF.

3NF

Appointment
staffNo
appointmentDateTime
patNo
branchNo

Therapist
staffNo
therapistName

Patient
patNo
patName

Exercise 3: Event Management

a. Assumptions:
    - Each contract applies to one event (implying eventNo is tied to contractNo).
    - Employees work on multiple contracts.
    - Employees can work on one event together.

- eventNo uniquely identifies each event, and does not depend on contractNo
- eNo and contractNo together identify a specific work assignment.

b. Normalization Process

This table is already in 1NF because there are no repeating groups and eNo and contractNo are the composite keys.The table is not in 2NF because eName depends only on eNo, violating full dependency on the composite PK (eNo, contractNo). Also, eventLoc depends on eventNo, which depends on contractNo (not directly on the full PK, but we can check this in 3NF).

In order to move this table to 2NF, we need to remove partial dependencies by separating them into different tables. The 2NF version would look like:

2NF

Contract
contractNo (PK)
eNo (PK)
Hours
eventNo
eventLoc

Employee
eNo (PK)
eName

The Contract table is not in 3NF because there is transitive dependency. eventLoc depends on eventNo, so we have to make a separate table for events.

3NF

Employee_contract
contractNo
eNo
Hours
eventNo

Employee
eNo
eName

Event
eventNo
eventLoc