**Question-01:**

**Solution:**

(a) Based on the provided form, we can identify the following functional dependencies:

1. Patient name → Full Name, Ward Number, Ward Name

2. Ward Number → Ward Name

3. Drug Number → Name, Description

4. Drug Number, Start Date → Finish Date

Assumptions:

- Each patient has a unique name.

- Each ward has a unique number.

- Each drug has a unique number.

- Start Date and Finish Date together determine the duration of drug administration.

(b) To normalize the attributes shown in Figure into a set of well-designed 3NF relations, we need to identify the functional dependencies and ensure that each relation satisfies the requirements of the normalization process. Here's an illustration of the normalization process:

1NF (First Normal Form):

Create a relation for each repeating group or multi-valued attribute:

Relation 1: Patient (Patient name, Full Name, Ward Number)

Relation 2: Ward (Ward Number, Ward Name)

2NF (Second Normal Form):

Ensure that each non-key attribute is fully functionally dependent on the entire primary key:

Relation 3: Drug (Drug Number, Name, Description)

3NF (Third Normal Form):

Ensure that there are no transitive dependencies where a non-key attribute depends on another non-key attribute:

Relation 4: Prescription (Drug Number, Start Date, Finish Date)

(c) Primary, alternate, and foreign keys in the 3NF relations:

Relation 1: Patient (Primary Key: Patient name)

Relation 2: Ward (Primary Key: Ward Number)

Relation 3: Drug (Primary Key: Drug Number)

Relation 4: Prescription (Primary Key: Drug Number, Start Date; Foreign Key: Drug Number references Drug)

Note: The form provided in the question seems to have duplicate entries for the drug "Morphine." It's advisable to ensure that each drug has a unique drug number to maintain data integrity in the database.

**Question-02:**

**Solution:**

(a) The table shown in Figure is susceptible to update anomalies, which include:

Insertion Anomaly: Inability to insert a new staff member's details if they haven't worked at any hotel yet. For example, if a new staff member is hired but hasn't been assigned to any hotel, there would be no place to store their information in the current table structure.

Deletion Anomaly: Removing a record of a staff member's work hours would also delete their personal information if they have no other records. For instance, if Smith J only worked once and that record is deleted, all information about Smith J would be lost.

Update Anomaly: Updating a staff member's information would require modifying multiple records if they have worked at multiple hotels. For example, if Smith J changes their address, it would need to be updated in every record where Smith J has worked.

(b) Based on the provided table, we can identify the following functional dependencies:

1. NIN (National Insurance Number) → eName (Employee Name)

2. contractNo → hNo (Hotel Number), HLoc (Hotel Location)

Assumptions:

- Each NIN corresponds to a unique staff member.

- Each contractNo corresponds to a unique contract.

- A contractNo is associated with a specific hotel represented by hNo and HLoc.

- An employee's name (eName) is dependent on their NIN.

- A hotel's location (HLoc) is dependent on its hNo.

(c) The process of normalizing the table shown in Figure to 3NF involves the following steps:

1NF (First Normal Form):

Break the table into smaller tables, each with a single theme:

- Staff (NIN, eName)

- Contract (contractNo, hNo)

- Hotel (hNo, HLoc)

- Hours (NIN, contractNo, hours)

2NF (Second Normal Form):

Ensure that each non-key attribute is fully functionally dependent on the entire primary key:

- Staff (NIN, eName)

- Contract (contractNo, NIN, hNo)

- Hotel (hNo, HLoc)

- Hours (NIN, contractNo, hours)

3NF (Third Normal Form):

Ensure that there are no transitive dependencies where a non-key attribute depends on another non-key attribute:

- Staff (NIN, eName)

- Contract (contractNo, NIN)

- Hotel (hNo, HLoc)

- ContractHotel (contractNo, hNo)

- Hours (NIN, contractNo, hours)

Primary, alternate, and foreign keys in the 3NF relations:

- Staff (Primary Key: NIN)

- Contract (Primary Key: contractNo; Foreign Key: NIN references Staff)

- Hotel (Primary Key: hNo)

- ContractHotel (Primary Key: contractNo, hNo; Foreign Keys: contractNo references Contract, hNo references Hotel)

- Hours (Primary Key: NIN, contractNo; Foreign Keys: NIN, contractNo references Contract)

Note: The assumption is made that a contract is uniquely identified by contractNo and is associated with a single staff member.

**Question-03:**

**Solution:**

(a) Examples of insertion, deletion, and update anomalies in the table are as follows:

Insertion Anomaly: If a staff member has not worked at any hotel yet, there is no place to insert their information into the table. For example, if a new staff member joins Instant Cover but hasn't been assigned to any hotel, their information cannot be inserted into the table.

Deletion Anomaly: If all records of a staff member's work are deleted, their personal information would also be lost. For instance, if both records for Smith J are deleted, there would be no information about Smith J in the table.

Update Anomaly: When updating a staff member's information, it needs to be modified in multiple records if they have worked at multiple hotels. For example, if Hocine D changes their name, it would need to be updated in both records where Hocine D has worked.

(b) Based on the provided table, we can identify the following functional dependencies:

1. NIN (National Insurance Number) → eName (Employee Name)

2. contractNo → hNo (Hotel Number), HLoc (Hotel Location)

Assumptions:

- Each NIN corresponds to a unique staff member.

- Each contractNo corresponds to a unique contract.

- An employee's name (eName) is dependent on their NIN.

- A contractNo is associated with a specific hotel represented by hNo and HLoc.

(c) The process of normalizing the table shown in Figure to 3NF involves the following steps:

1NF (First Normal Form):

Break the table into smaller tables, each with a single theme:

- Staff (NIN, eName)

- Contract (contractNo, NIN)

- Hotel (hNo, HLoc)

- Hours (NIN, contractNo, hours)

2NF (Second Normal Form):

Ensure that each non-key attribute is fully functionally dependent on the entire primary key:

- Staff (NIN, eName)

- Contract (contractNo, NIN)

- Hotel (hNo, HLoc)

- Hours (NIN, contractNo, hours)

3NF (Third Normal Form):

Ensure that there are no transitive dependencies where a non-key attribute depends on another non-key attribute:

- Staff (NIN, eName)

- Contract (contractNo, NIN)

- Hotel (hNo, HLoc)

- ContractHotel (contractNo, hNo)

- Hours (NIN, contractNo, hours)

Primary, alternate, and foreign keys in the 3NF relations:

- Staff (Primary Key: NIN)

- Contract (Primary Key: contractNo; Foreign Key: NIN references Staff)

- Hotel (Primary Key: hNo)

- ContractHotel (Primary Key: contractNo, hNo; Foreign Keys: contractNo references Contract, hNo references Hotel)

- Hours (Primary Key: NIN, contractNo; Foreign Keys: NIN, contractNo references Contract)

Note: The assumption is made that a contract is uniquely identified by contractNo and is associated with a single staff member.