**Normalization and 3NF – Assignment 7**

**Part 1: Verifying that the tables are in 3NF**

To verify if the tables provided are in Third Normal Form (3NF), we need to go through the normalization process step by step. Here's a breakdown of how to approach this for each table:

**First Normal Form (1NF)**

1NF requires that the table has only atomic (indivisible) values and that there are no repeating groups.

For all the given tables (**Customer**, **Item**, **Store**, **Billing**, **Employee**), they appear to be in 1NF because:

* All attributes have atomic values.
* There are no repeating groups or arrays.

**Second Normal Form (2NF)**

A table is in 2NF if it is in 1NF and all non-key attributes are fully functionally dependent on the primary key.

Since all the tables provided have a single column as the primary key, and none of the other columns are dependent on a part of a composite key (there are no composite keys at all), they inherently satisfy the condition for 2NF.

**Third Normal Form (3NF)**

A table is in 3NF if it is in 2NF and all the attributes are functionally dependent only on the primary key and there are no transitive dependencies (non-key attributes do not depend on other non-key attributes).

Customer Table

* **Candidate Key**: **Customer\_ID**
* Customer Table: This table seems to be in 3NF. Each attribute depends only on the Customer\_ID, which is the primary key.
* **Functional Dependencies**:
  + **Customer\_ID** → **F\_Name**, **L\_Name**, **Address**, **Email**, **C\_Points**, **C\_Gifts**, **C\_History**
* No transitive dependencies are apparent from the given structure.

Item Table

* **Candidate Key**: **Item\_ID**
* This table also appears to be in 3NF. Each attribute depends only on the Item\_ID, the primary key.
* **Functional Dependencies**:
  + **Item\_ID** → **Item\_Name**, **Item\_Price**, **Item\_Quantity**
* No transitive dependencies are apparent from the given structure.

Store Table

* **Candidate Key**: **Branch\_ID**
* Again, in 3NF because every non-key attribute (Branch\_Name, Branch\_Location, Branch\_Stock) directly depends on the primary key Branch\_ID.
* **Functional Dependencies**:
  + **Branch\_ID** → **Branch\_Name**, **Branch\_Location**, **Branch\_Stock**
* No transitive dependencies are apparent from the given structure.

Billing Table

* **Candidate Key**: **Transaction\_ID**
* All non-key attributes depend only on Transaction\_ID, which is the primary key. It seems to be in 3NF.
* **Functional Dependencies**:
  + **Transaction\_ID** → **Taxes**, **Total\_Amount**, **Discounts**, **Payment\_Method**, **Status**, **Returns**, **Transaction\_Date**
* No transitive dependencies are apparent from the given structure.

Employee Table

* **Candidate Key**: **E\_ID**
* All non-key attributes are dependent on E\_ID, the primary key, so it's in 3NF.
* **Functional Dependencies**:
  + **E\_ID** → **F\_Name**, **L\_Name**, **Address**, **Email**, **E\_Hours**, **P\_Number**, **E\_Wage**, **Banking\_Info**
* No transitive dependencies are apparent from the given structure.

All the tables in our store database are in 3NF. If discrepancies are found or additional functional dependencies are provided, the decomposition process would involve:

1. Identifying the partial/transitive dependencies.

2. Creating new tables to eliminate these dependencies.

3. Reassigning the dependent attributes to the new tables.

4. Establishing referential integrity with foreign keys.