

1NF Conformant:

To transform the data into the first normal form (1NF), we need to ensure that each attribute contains atomic (indivisible) values, and there are no repeating groups within a row.

Tables in 1NF:

Components Table:

Component_ID (Primary Key)
Component
Manufacturer_ID (Foreign Key)
Manufacturer
Quality_Control Table:

Test_Number (Primary Key)
Component_ID (Foreign Key)
Quality_Control_Type
Result
Manufacturing_Cost Table:

Manufacturer_ID (Primary Key)
Manufacturing_cost
Quantity Table:

Manufacturer_ID (Primary Key)
Quantity
Assembling_Plant Table:

Assembling_Plant_ID (Primary Key)
Assembly_Cost
Product
Assembling_Plant

Each table now contains atomic values, and there are no repeating groups within each row.
Relationships between tables are established through foreign key-primary key relationships.

```
-- Create the table for Quality and Manufacturing data
CREATE TABLE quality_manufacturing (
    Product VARCHAR(255),
    Component_Id INT,
    Component VARCHAR(255),
    Quality_Control_Type VARCHAR(255),
    Result VARCHAR(255),
    Reason VARCHAR(255),
    Manufacturer_ID VARCHAR(255),
    Manufacturer VARCHAR(255),
    Manufacturing_Cost DECIMAL(10, 2)
);
```

```
-- Insert data into the Quality and Manufacturing table
INSERT INTO quality_manufacturing (Product, Component_Id, Component, Quality_Control_Type, Result, Reason, Manufacturer_ID, Manufacturer, Manufacturing_Cost)
SELECT Product, Component_Id, Component, Quality_Control_Type, Result, Reason, Manufacturer_ID, Manufacturer, Manufacturing_Cost
FROM dataraw;

-- Retrieve and view data from the Quality and Manufacturing table
SELECT * FROM quality_manufacturing;
```

2NF Conformant:

In order to reach the second normal form (2NF), we need to ensure that each non-prime attribute (attribute not part of the primary key) is fully functionally dependent on the entire primary key.

To achieve 2NF, we need to ensure that all attributes are fully dependent on the entire primary key. In the 1NF tables, some attributes in Quality_Control, Manufacturing_Cost, and Quantity tables are dependent only on part of the primary key (Manufacturer_ID).

Transformed Tables in 2NF:

Components Table (2NF):

Component_ID (Primary Key)

Component

Manufacturers Table (2NF):

Manufacturer_ID (Primary Key)

Manufacturer

Quality_Control Table (2NF):

Test_Number (Primary Key)

Component_ID (Foreign Key)

Quality_Control_Type

Result

Manufacturing_Details Table (2NF):

Manufacturer_ID (Primary Key)

Manufacturing_cost

Quantity_Details Table (2NF):

Manufacturer_ID (Primary Key)

Quantity

Assembling_Plant Table (2NF):

Assembling_Plant_ID (Primary Key)

Assembly_Cost

Product

Assembling_Plant

Here, each attribute in the tables (except for foreign keys) is fully functionally dependent on the primary key of its respective table, meeting the requirements of the second normal form.

3NF Conformant:

To achieve the third normal form (3NF) from the 2NF tables, we need to ensure that there are no transitive dependencies—non-prime attributes are not dependent on other non-prime attributes within the same table.

To convert these tables to 3NF, we need to identify and remove any transitive dependencies.

Transformed 3NF Tables:

Components Table (3NF):

Component_ID (Primary Key)

Component

Manufacturers Table (3NF):

Manufacturer_ID (Primary Key)

Manufacturer

Quality_Control Table (3NF):

Test_Number (Primary Key)

Component_ID (Foreign Key)

Quality_Control_Type

Result

Manufacturing_Cost Table (3NF):

Manufacturer_ID (Primary Key)

Manufacturing_cost

Quantity Table (3NF):

Manufacturer_ID (Primary Key)

Quantity

Assembling_Plant Table (3NF):

Assembling_Plant_ID (Primary Key)

Assembly_Cost

Product

Assembling_Plant

```
-- Create Quality Control table
CREATE TABLE IF NOT EXISTS quality_control (
    component_id INT,
    quality_control_type VARCHAR(255),
    result VARCHAR(255),
    reason VARCHAR(255)
);

-- Insert data into Quality Control table
INSERT IGNORE INTO quality_control (component_id, quality_control_type, result, reason)
SELECT Component_Id, Quality_Control_Type, Result, Reason
FROM quality_manufacturing;

-- Create Manufacturer table
CREATE TABLE IF NOT EXISTS manufacturer (
    manufacturer_id VARCHAR(255),
    manufacturer VARCHAR(255)
);

-- Insert data into Manufacturer table
INSERT IGNORE INTO manufacturer (manufacturer_id, manufacturer)
SELECT Manufacturer_ID, Manufacturer
FROM quality_manufacturing;

-- Create Components table
CREATE TABLE IF NOT EXISTS components (
    component_id INT,
    product VARCHAR(255),
    component VARCHAR(255)
);

-- Insert data into Components table
INSERT IGNORE INTO components (component_id, product, component)
SELECT Component_Id, Product, Component
FROM quality_manufacturing;

-- Create Manufacturing Cost table
CREATE TABLE IF NOT EXISTS manufacturing_cost (
    manufacturer_id VARCHAR(255),
    manufacturing_cost DECIMAL(10, 2)
);

-- Insert data into Manufacturing Cost table
INSERT IGNORE INTO manufacturing_cost (manufacturer_id, manufacturing_cost)
SELECT Manufacturer_ID, Manufacturing_Cost
FROM quality_manufacturing;
```

```
-- Create Assembling Cost table
CREATE TABLE IF NOT EXISTS assembling_cost (
    assembling_plant_id VARCHAR(255),
    assembly_cost DECIMAL(10, 2)
);

-- Insert data into Assembling Cost table
INSERT IGNORE INTO assembling_cost (assembling_plant_id, assembly_cost)
SELECT Assembling_Plant_ID, Assembly_Cost
FROM dataraw;

-- Create Assembly Plant table
CREATE TABLE IF NOT EXISTS assembly_plant (
    assembling_product VARCHAR(255),
    assembling_plant_id VARCHAR(255),
    assembly_plant VARCHAR(255)
);

-- Insert data into Assembly Plant table
INSERT IGNORE INTO assembly_plant (assembling_product, assembling_plant_id, assembly_plant)
SELECT Assembling_Product, Assembling_Plant_ID, Assembly_Plant
FROM dataraw;
```