**PART 2 – STATING/WRITING THE “MESSY” COMPANY BUSINESS RULES, BASED ON THE ANALYSIS YOU HAVE CONDUCTED IN THE PART 1**

1. ProductID must be unique for every product and require having six digits.
2. Phone number must have 11 digits, no more than or less than it.
3. Email addresses are required to follow the format of @domain\_name.
4. Customers can be eligible for only five promo codes based on the promo code’s criteria.
5. Customers can get 10% discounts on products using PROMO10 and DISCOUNT10, not for other promo codes.
6. Product tags must be three or less than three.
7. Customers can get 5% discounts on products using the FIVEOFF and STAFFDISC promo.
8. Customers can get 15% discounts on products using the SPRING promo.
9. STAFFDISC, SPRING15, DISCOUNT10 must come under the DiscountAppliedPC category which means the discount of 5,15,10 percentage respectively should be applied on the TotalSalePrice and updated on NewTotalSalePrice.
10. PROMO10, FIVEOFF must come underDiscountAppliedabs category which means the diduction of 10 and 5 respectively on TotalSalePrice and updated on NewTotalSalePrice.
11. After applying the promo code, it must require that the NewTotalSales price is less than compared to TotalSalePrice.
12. The description must have only words for the product.
13. SaleDate contains the date and time of order of customers, and it must be 12 digits.
14. Attribute customerAddress5 only has the postal code of the customer’s address which is not greater than 7 letters and not less than 6 letters.
15. Customer name attributes are built by combining customer first name and customer’s last name, and other things are not accepted.
16. A repeat customer has only two values, No and Yes; when one particular customer is repeated, then it must show Yes otherwise, it is needed to guide No.
17. Product Id must range between 4-6 digits.
18. Customer Address1 should have an Apartment or street Number and its Name.
19. Except @ not any special characters should be used in email.
20. Products whose TotalSalePrice is NULL should not be eligible for any sort of DISCOUTN or PROMOCDE.
21. Customer’s whose CustomerEmail Field is NULL is not eligible for DISCOUNT PromoCode.
22. Customer address one must be combined with street number and street address, nothing else.
23. Customer addresses three contain the only area of their address, and it must be in one word
24. Postal code is seperated by space from total character’s median

**PART: 3 DESIGNING, IMPLEMENTING AND POPULATING THE NORMALIZED TO THE 3RD NORMAL FORM “MESSY”**

The table **Messy** is not in 1NF How do we know that? Let’s discuss the rules of 1NF, 2NF, 3NF.

1. 1NF: Given below are the established rules to achieve the 1NF.

* Each cell to be single valued (Atomic Values).
* Entries in a column are same type.
* Rows uniquely identified- Add unique ID or Add more columns to make unique ID.
* Table stores data be in rows and columns.
* In Messy table there were repeated values in **ProductTags** column. It was showing three tags in one column. To achieve the first rule of 1NF, we added two more column to table.
* There was no Primary Key in table. According to the 3rd rule of 1NF there should be one or more unique ID to identify the rows uniquely. So, we altered the Messy table and make the OrderNo, ProductID as Primary key and we added one more column CustomerID as primary key for customer data.

**Note:** Check below diagram for detail understanding**.**

A picture containing table

Description automatically generatedTable

Description automatically generated

**1NF**

1. 2NF: Given below are the established rules to achieve the 1NF

* The table must be in 1NF.
* All attributes (Non key columns) depend on primary key.
* In simple word, we must check each column of table is dependent on any primary key or not. If column is dependent on any of one primary key, then keep that column in table as it is but what if not? Then create another table and move that column into that table. Messy table is already in 1NF. So, Let’s check this condition on Messy table.
* In Messy table the below columns are not dependent on any primary key. Thus, we createdseparate table name as **Product details** and added these below columns to this new table with **ProductID** as **primary key**.

1. ProductID
2. ProductDescription
3. ProductTag1
4. ProductTag1
5. ProductTag3
6. LineItemPrice

* Similarly, below columns are not dependent on any primary key. So, we created a new table name as **Customer Details** and added these below columns to this new table with **CustomerID** as **primary key**.

1. CustomerID
2. CustomerName
3. CustomerAddress1
4. CustomerAddress2
5. CustomerAddress3
6. CustomerAddress4
7. CustomerAddress5
8. CustomerEmail
9. CustomerPhoneNo

* However, other remaining columns are dependent on **OrderNo** primary key, so we kept that column as it is in **New\_Messy\_1NF** table. There are two foreign key columns also **ProductID** and **CustomerID** in that table**.**

Now we can say that tables are in 2NF

**Note:** Check the diagram on next page for detail understanding of 2NF.

Diagram, table

Description automatically generated

**2NF**

1. 3NF: Given below are the established rules to achieve the 1NF.

* All the columns in table can be determined by the primary key in the table and no other column (non-Transitive dependency).
* Table must be in 2NF
* First check the **Product Details** table, is every column dependent on the primary key or not with non-Transitive dependency?

Observe **ProductTag1, ProductTag2, ProductTag3** column is not uniquely dependent on **ProductID** primary key. Thus, again we must create new table **Tag\_Details** and move this above three columns into **Tag\_Details** table.

Add one more column in same (**Tag\_Details**) table **TagID** as primary key to all the three columns **ProductTag1, ProductTag2, ProductTag3** and keep **Product\_details** table with three columns **ProductID, ProductDescription, and LineItemPrice** only**.**

* Create new table **Product\_tag\_join** as intersection table for **Product Details** and **Tag\_Details** table with two foreign key columns in that **ProductID** and **TagID**.
* Now rename the **new\_Messy\_1NF** table by **Order\_Details**.
* Check every column of **Order\_Details** are uniquely dependent on primary key **OrderNo** or not.

Here the **Promocode** column is dependent on **OrderNo** but other two columns **DiscountAppliedAbs, DiscountAppliedPc** are transitively dependent on **Promocode.**

Thus, we must create a new table **Promo\_code** in which move columns **Promocode DiscountAppliedAbs, DiscountAppliedPc.**

Next add new column **PromoNo** as primary key in **Promo\_code** table and as foreign key in **Order\_Details** because **Promo\_code** has some Null values also, so, it can’t be a primary key and foreign key too.

* Create a new table **Order\_prod\_join** as intersection table for **Order\_details** and **Product\_details**. Add **ProductID** and **OrderNo** as foreign key in that table.

Now we can say that tables are in 3NF

**Note:** Check the diagram on next page for detail understanding of 3NF.

**Diagram

Description automatically generated**

**3NF**

On next page

Paste Table creation code and insertion code.

Cross check the table name and attributes once

Read the document once and compare the logic written by me(correct it if its wrong)

Add some business rule if you want

If all good then add above things to this doc and submit the assignment.

Thank you

Paresh