NORMALIZATION : DATABASES

Q. What is normalization?

A. Normalization refers to the process of structuring a relational database in accordance with a series of "normal forms" in order to reduce data redundancy and improve data integrity.

- All normal forms are progressive, ie. a database in the 3rd Normal Form is also in the 1st and 2nd NF.

1NF : Each attribute only has "atomic" data
↳ ie. the simplest, and most "single" form of the data being represented

+ Each table has a unique identifier called a "primary key"

for example : "20 tyres at $90/tyre"

broken down into atomic data

↓

Product	Quantity	Price/unit
Tyre	20	$90

ID	FirstName	LastName	Date	Order
1	John	Smith	01/01/20	35 candybars at $1.25 /ea.
2	Jake	Doe	02/03/19	40 shovels at $50.00/ea.
3	Brenda	Brown	05/04/18	10 TVs at $950 /ea.

↳ This is not in 1NF

↳ Group of attributes

This database, converted into 1NF, would look like:

ID	FirstName	LastName	Date	Qty.	Price/unit	Product
1	John	Smith	01/01/20	35	1.25	Candybar
2	Jake	Doe	02/03/19	40	50.00	shovel
3	Brenda	Brown	05/04/18	10	950.00	TV

2NF : Every non-prime attribute depends only and only (exclusively) on the primary key
+ There are no partial dependencies

Now let's consider why the database we just normalized to 1NF is NOT in 2NF.

ID	FirstName	LastName	Date	Qty.	Price/unit	Product
1	John	Smith	01/01/20	35	1.25	Candybar
2	Jake	Doe	02/03/19	40	50.00	shovel
3	Brenda	Brown	05/04/18	10	950.00	TV

Note: As a rule of thumb, whenever it seems like a single table is showing too much information

We can't clearly tell whether this table is meant to contain data for "customers" or for on too many things (ie. in this case, "orders" our table is showing both customer details and order details), it's a good indication that it's not in 2NF and may need to be split up into multiple tables.

Remember: A table primarily shows data relevant to <u>one</u> entity

Normalized to 2NF, our database would now look like:
↙

CUSTOMER

customerID	FirstName	LastName
1	John	Smith
2	Jake	Doe
3	Brenda	Brown

ORDER

OrderID	CustomerID	Date	Qty.	Price/unit	Product
1	1	01/01/20	35	1.25	Candybar
2	2	02/03/19	40	50.00	shovel
3	3	05/04/18	10	950.00	TV

3NF : Attributes are determined ONLY by the primary key
aka. All attributes in a table vary as the primary key varies

Now let's consider why our 2NF database violates 3NF

CUSTOMER

customerID	FirstName	LastName
1	John	Smith
2	Jake	Doe
3	Brenda	Brown

ORDER

OrderID	CustomerID	Date	Qty.	Price/unit	Product
1	1	01/01/20	35	1.25	Candybar
2	2	02/03/19	40	50.00	shovel
3	3	05/04/18	10	950.00	TV

↑

This attribute does not necessarily vary as the primary key varies, because the price per unit is dependent on what product you're buying, not when or who is buying it, or how many are being bought

↓

Hence this table needs to be further broken down

Now, normalizing the above database into 3NF gives us the following:

CUSTOMER

customerID	FirstName	LastName
1	John	Smith
2	Jake	Doe
3	Brenda	Brown

ORDER

OrderID	CustomerID	Date	Qty.	ProductID
1	1	01/01/20	35	11
2	2	02/03/19	40	12
3	3	05/04/18	10	13

PRODUCT

ProductID	Price/unit	Product
11	1.25	Candybar
12	50.00	shovel
13	950.00	TV

Finally, the database has been normalized