Normalization

- Makes your DB easier to understand
- Removes redundancy
- Protects from 3 anomolies:
  - Insertion
    - impossible to insert a certain datatype into your database
  - Update
    - When updating a value of a column leads to inconsistencies in your DB
  - Deletion
    - Deletion of data leads to unintended additional data loss

* Normalization fixes these problems by implementing different levels of database normalization, 1NF, 2NF, 3NF

1NF - Minimum Safety guarantee

Product

ID	color	Price
1	red, green	15.50
2	yellow	20
3	green	25
4	yellow, blue	10
5	red	30

Price

ID product	Price
1	15.50
2	20
3	25
4	10
5	30

Color

ID product	Color
1	red
1	green
2	yellow
3	green
4	yellow
4	blue
5	red

Things that violate 1NF:

- Using row order to convey info.
- Mixing data types within a column
- having a table without a PK
- Storing a repeating group of data on a single row

2NF

- A tables non-key column relates to the PK.
- Each non-key column must be dependent on the pk.

Purchase_Detail

Customer ID	StoreID	Location
1	1	Los Angeles
1	3	San Francisco
2	1	Los Angeles
3	2	New York
4	3	San Francisco

PK — non-key column

Purchase_Table

Customer ID	Store ID
1	1
1	3
2	1
3	2
4	3

Store_Table

Store ID	Location
1	Los Angeles
2	New York
3	San Fran

Should satisfy the following:

- is in 1NF
- No partial dependencies
  - * All the non-key attributes are fully dependent on PK

3NF

- Forbids dependency on a non-key attribute
- Where every non-key attribute should depend on the key, the whole key, and nothing but the key

Book Detail_Table

BookID	GenreID	GenreType	Price
1	1	Gardening	10
2	2	Sports	12
3	1	Gardening	11
4	3	Travel	9
5	2	Sports	20

PK — * GenreType is reliant on GenreID

Book

BookID	GenreID	Price
1	1	10
2	2	12
3	1	11
4	3	9
5	2	20

Genre

GenreID	Type
1	Gardening
2	Sports
3	Travel

Should Satisfy the following:

- is in 1NF
- is in 2NF
- No transitive Dependencies