Normalization

.NET

Database normalization is the process of structuring a relational database in order to reduce data redundancy and improve data integrity.

HTTPS://EN.WIKIPEDIA.ORG/WIKI/DATABASE_NORMALIZATION

Normalization Assignment (Pt. 1)

- Create an unnormalized table.
- List the information of your family members.
- There must be at least 5 attributes to each tuple (row) and at least 5 entities.

Databases – Normalization

https://www.tutorialspoint.com/dbms/database_normalization.htm
https://www.c-sharpcorner.com/UploadFile/0146e3/database-normalization/

Normalization is a method to prevent ***anomalies*** and keep the database in a consistent state. ***Fields*** and ***tables*** of a relational DB are organized to minimize redundancy and dependency.

Normalization involves dividing large ***tables*** into smaller (and less redundant) ***tables*** and defining relationships among their ***atomic*** data.

There are many normal forms but *1NF*, *2NF*, and *3NF* are primarily used.

Databases – Normalization

https://www.tutorialspoint.com/dbms/database_normalization.htm
https://www.c-sharpcorner.com/UploadFile/0146e3/database-normalization/

This table is not normalized. All the information is stored in one table.

Title	Name	City	Movies Watched
Mr.	Moore	Crowley	Avengers Endgame, Thor
Miss.	Garza	Monterrey	Ant-Man, Captain Marvel
Mr.	Moore	Dallas	Spider-Man: Homecoming, Doctor Strange, Iron Man 2

Databases – Normalization

https://www.tutorialspoint.com/dbms/database_normalization.htm
https://www.c-sharpcorner.com/UploadFile/0146e3/database-normalization/

1st Normal Form (1NF)

- Have a single ***Primary Key***.
- Each table cell should contain a single value.
- Each entity needs to be unique.
- The table contains atomic values only.

Title	Name	City	Movies Watched
Mr.	Moore	Crowley	Avengers Endgame
Mr.	Moore	Crowley	Thor
Miss.	Garza	Monterrey	Ant-Man
Miss.	Garza	Monterrey	Captain Marvel
Mr.	Moore	Dallas	Spider-Man: Homecoming
Mr.	Moore	Dallas	Doctor Strange
Mr.	Moore	Dallas	Iron Man 2

Databases – Normalization

https://www.tutorialspoint.com/dbms/database_normalization.htm
https://www.c-sharpcorner.com/UploadFile/0146e3/database-normalization/

2nd Normal Form (2NF) –

- First, be in *1NF*.
- Remove subsets of data that apply to multiple rows of a ***table*** and place them in separate ***tables*** with ***PK → FK*** relationships among the new tables.
- If the table is in *1NF* and every non-key attribute is dependent on the ***Primary Key***. then *2NF* is achieved.

Id(PK)	Title	Name	City 1
1	Mr.	Moore	Crowley
2	Miss.	Garza	Monterrey
3	Mr.	Moore	Dallas

Actions Taken:
The *1NF* table is divided into two tables. Table 1 contains only person information. ***Id*** is created as the ***Primary Key (PK)*** for Table 1. Table 2 contains the information for each movie. Table 2's new ***PK*** column is ***MovieId***.

MovieId(PK)	Id(FK)	Movie 2
22	1	Avengers Endgame
4	1	Thor
12	2	Ant-Man
21	2	Captain Marvel
16	3	Spider-Man: Homecoming
14	3	Doctor Strange

Databases – Normalization

https://www.tutorialspoint.com/dbms/database_normalization.htm
https://www.c-sharpcorner.com/UploadFile/0146e3/database-normalization/

To achieve *3NF*, there must be no dependencies between fields in a single row.

"Given a value for column B, do we then have only one possible value for column C?"

If yes, B and C should be put into a new table, with one of them becoming the ***Primary Key (PK)***. A reference to the new table should be left in the original table and marked as a ***Foreign Key***.

1

Id(PK)	Title	Name	City
1	Mr.	Moore	Crowley
2	Miss.	Garza	Monterrey
3	Mr.	Moore	Dallas

A ***Transitive Functional Dependency*** occurs when the change of one ***Candidate Key*** column might cause any other ***Candidate Key*** column to change. In table 1, changing the non-key column '*Name*' may change '*Title*'.

2

MovieId(PK)	Id(FK)	Movie
22	1	Avengers Endgame
4	1	Thor
12	2	Ant-Man
21	2	Captain Marvel
16	3	Spider-Man: Homecoming
14	3	Doctor Strange
3	3	Iron Man 2

Databases – Normalization

https://www.tutorialspoint.com/dbms/database_normalization.htm
https://www.c-sharpcorner.com/UploadFile/0146e3/database-normalization/

Table 1	
TitleId (PK)	Title
1	Mr.
2	Miss.
3	Mrs.
4	Dr.

Table 4	
CityId (PK)	City
76036	Crowley
75201	Dallas
32070	Monterrey
76701	Waco

A ***Transitive Functional Dependency*** occurs when the change of one ***Candidate Key*** column might cause any other ***Candidate Key*** column to change. In table 1, changing the Candidate Key column '*Name*' may change '*Title*'.

Table 2	
MovieId (PK)	Movie
22	Avengers Endgame
4	Thor
12	Ant-Man
21	Captain Marvel
16	Spider-Man: Homecoming
14	Doctor Strange
3	Iron Man 2

Junction Table	
MovieId (FK)	Id (FK)
22	1
4	1
12	2
21	2
16	3
14	3
3	3

Table 3			
Id (PK)	TitleId (FK)	Name	CityId (FK)
1	1	Moore	76036
2	2	Garza	75201
3	1	Moore	32070

Actions Taken:
Table 1 is divided. Two new tables are created to store Title and City. Table 2 is divided to isolate Movie data and a Junction table is created to show the Many-To-Many relationship between movie and person. The database is now in *3NF*.

Assignment (Pt. 2)

- Convert your Pt. 1 table to a 3NF table.