

DATABASE

NORMALIZATION

What is Normalization ?

- **NORMALIZATION** is a database design technique that reduces data redundancy.
- Normalization rules divides larger tables into smaller tables and links them using relationships.
- The purpose of Normalization in SQL is to eliminate redundant (repetitive) data and ensure data is stored logically.

Database Normal Forms

- **Here is a list of Normal Forms**
 - **1NF** (First Normal Form)
 - **2NF** (Second Normal Form)
 - **3NF** (Third Normal Form)
 - **BCNF** (Boyce-Codd Normal Form)
 - **4NF** (Fourth Normal Form)
 - **5NF** (Fifth Normal Form)
 - **6NF** (Sixth Normal Form)

Database Normalization With Examples

- Assume, a video library maintains a database of movies rented out. Without any normalization, all information is stored in one table as shown below.

FULL NAMES	PHYSICAL ADDRESS	MOVIES RENTED	SALUTATION
Janet Jones	First Street Plot No 4	Pirates of the Caribbean, Clash of the Titans	Ms.
Robert Phil	3 rd Street 34	Forgetting Sarah Marshal, Daddy's Little Girls	Mr.
Robert Phil	5 th Avenue	Clash of the Titans	Mr.

Database Normalization (1NF)

- Each table cell should contain a single value.
- Each record needs to be unique.

FULL NAMES	PHYSICAL ADDRESS	MOVIES RENTED	SALUTATION
Janet Jones	First Street Plot No 4	Pirates of the Caribbean	Ms.
Janet Jones	First Street Plot No 4	Clash of the Titans	Ms.
Robert Phil	3 rd Street 34	Forgetting Sarah Marshal	Mr.
Robert Phil	3 rd Street 34	Daddy's Little Girls	Mr.
Robert Phil	5 th Avenue	Clash of the Titans	Mr.

What is a key ?

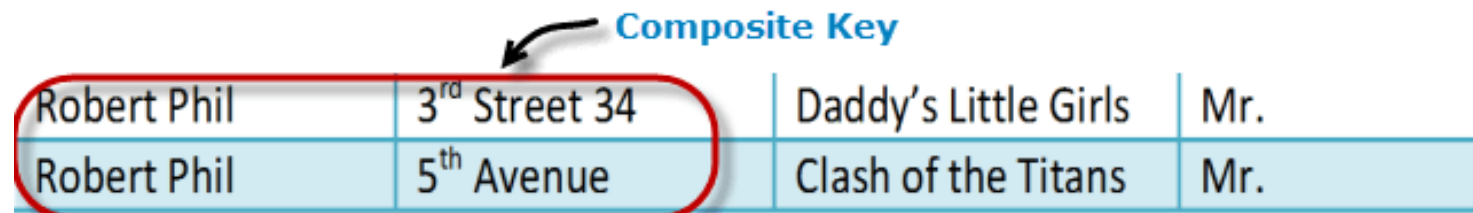
- A KEY is a value used to identify a record in a table uniquely. A KEY could be a single column or combination of multiple columns.
- Types of keys:
 1. Primary key
 2. Composite key
 3. Foreign key

What is a Primary key ?

- A primary is a single column value used to identify a database record uniquely.
- It has following attributes
 1. A primary key cannot be NULL
 2. A primary key value must be unique
 3. The primary key values should rarely be changed
 4. The primary key must be given a value when a new record is inserted.

What is a Composite Key?

- A composite key is a primary key composed of multiple columns used to identify a record uniquely
- In our database, we have two people with the same name Robert Phil, but they live in different places.
- Hence, we require both Full Name and Address to identify a record uniquely. That is a composite key.



Robert Phil	3 rd Street 34	Daddy's Little Girls	Mr.
Robert Phil	5 th Avenue	Clash of the Titans	Mr.

Names are common. Hence you need name as well Address to uniquely identify a record.

Database Normalization (2NF)

- We have to divide our 1NF table into two tables viz. Table 1 and Table 2. Table 1 contains member information. Table 2 contains information on movies rented.

Table 1: MemberInfo

PK	MEMBERSHIP ID	FULL NAMES	PHYSICAL ADDRESS	SALUTATION
	1	Janet Jones	First Street Plot No 4	Ms.
	2	Robert Phil	3 rd Street 34	Mr.
	3	Robert Phil	5 th Avenue	Mr.

Table 2: RentedMovies

FK	MEMBERSHIP ID	MOVIES RENTED
	1	Pirates of the Caribbean
	1	Clash of the Titans
	2	Forgetting Sarah Marshal
	2	Daddy's Little Girls
	3	Clash of the Titans

What is a Foreign Key ?

- Foreign Key references the primary key of another Table! It helps connect your Tables
 1. A foreign key can have a different name from its primary key
 2. It ensures rows in one table have corresponding rows in another
 3. Unlike the Primary key, they do not have to be unique. Most often they aren't

JOIN Tables

- A JOIN clause is used to combine rows from two or more tables, based on a related column between them.
- **SELECT** MI.FullNames, RM.MoviesRented
FROM MemberInfo as MI
JOIN RentedMovies as RM **ON** MI.MembershipID=RM. MembershipID;

FULL NAMES	MOVIES RENTED
Janet Jones	Pirates of the Caribbean
Janet Jones	Clash of the Titans
Robert Phil	Forgetting Sarah Marshal
Robert Phil	Daddy's Little Girls
Robert Phil	Clash of the Titans

Another JOIN example

- `SELECT MI.FullNames, RM.MoviesRented
FROM MemberInfo as MI
JOIN RentedMovies as RM ON MI.MembershipID=RM. MembershipID
WHERE MI.FullNames ='Janet Jones';`

FULL NAMES	MOVIES RENTED
Janet Jones	Pirates of the Caribbean
Janet Jones	Clash of the Titans

DB Normalization for Book App

books table

id	title	author	isbn	image_url	description
1	Book1	A	Isbn-32542	Img-url-1	desc1
2	Book2	B	Isbn-14885	Img-url-2	desc2
3	Book3	C	Isbn-32699	Img-url-3	desc3
4	Book4	D	Isbn-14142	Img-url-4	desc4

How to implement the following with the current table design ?

- **Insert a new author ??**

- If I want to add a new author, that means I have to add a new book (new row!!)

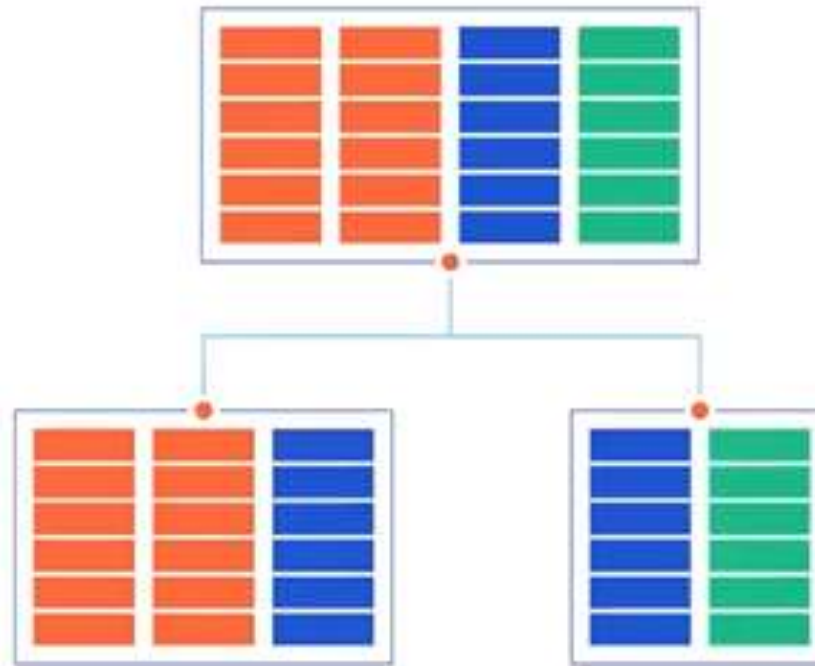
- **Add a new property to the author ??**

- If I want to add the author-nationality then I have to go over them row by row to change them all (waste of time and not efficient)

- **Update an author name ??**

- If I want to update an author name then I have to go over them row by row to change them all (waste of time and not efficient)

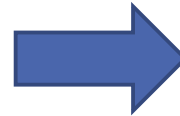
Our Database is de-normalized so it needs improvement (Normalization)



Book-app DB Normalization

books

id	title	author	isbn	image_url	description
1	Book 1	Alice	Isbn-32542	Img-url-1	desc1
2	Book 2	Bob	Isbn-14885	Img-url-2	desc2
3	Book 3	Cathy	Isbn-32699	Img-url-3	desc3
4	Book 4	Cathy	Isbn-14142	Img-url-4	desc4



FK

books

id	title	author_id	isbn	image_url	description
1	Book 1	1	Isbn-32542	Img-url-1	desc1
2	Book 2	2	Isbn-14885	Img-url-2	desc2
3	Book 3	3	Isbn-32699	Img-url-3	desc3
4	Book 4	3	Isbn-14142	Img-url-4	desc4

authors

PK

id	name
1	Alice
2	Bob
3	Cathy

Resources

- <https://www.guru99.com/database-normalization.html>
- https://www.w3schools.com/sql/sql_join.asp