Normalization
ER diagram
Cardinality

Trainer: Mohammad Shabbir Taibani

AGENDA

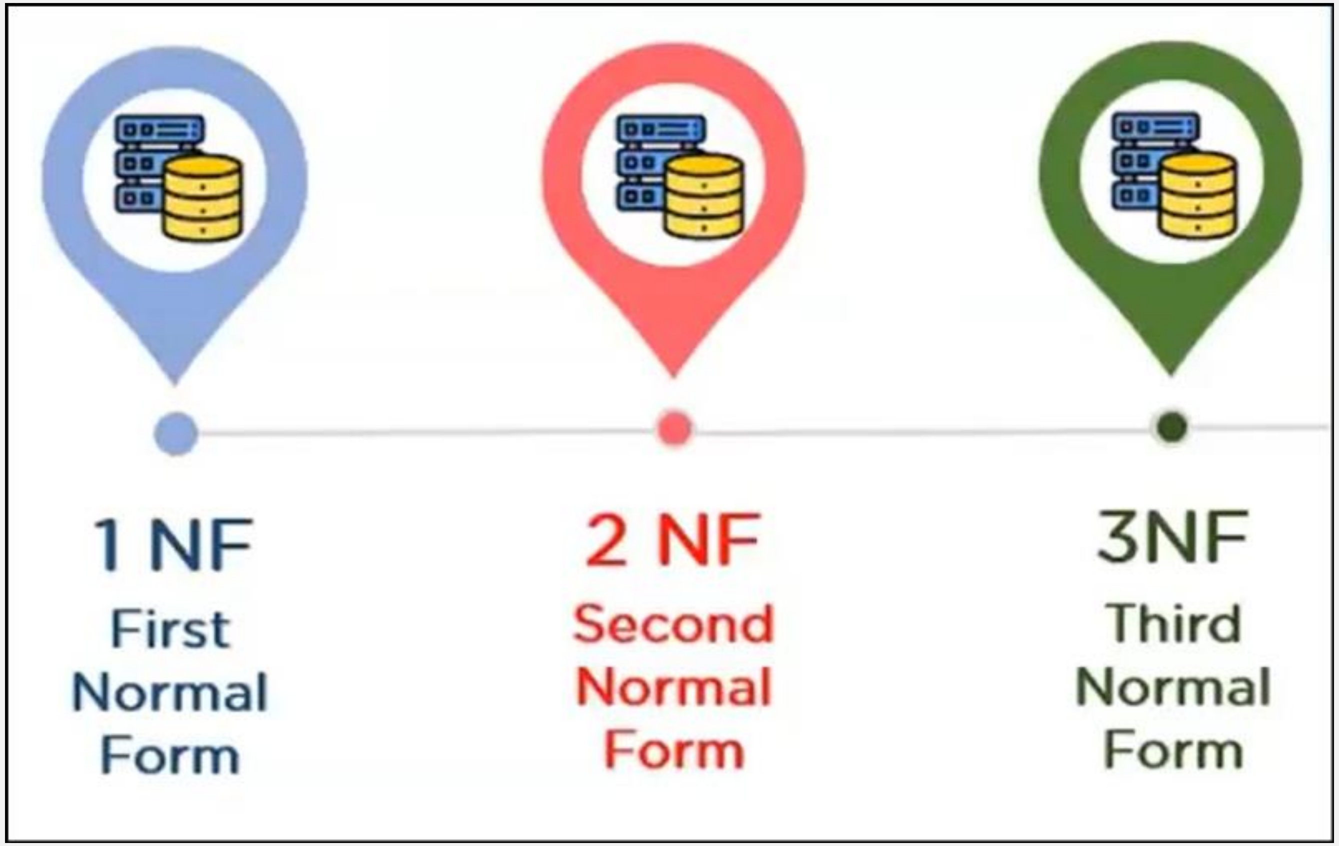
- Normalization 1NF,2NF,3NF
- Entity Relationship Diagrams
- Cardinality

WHAT IS NORMALIZATION

Normalization is a database design technique which organizes tables in a manner that reduces redundancy and dependency of data.

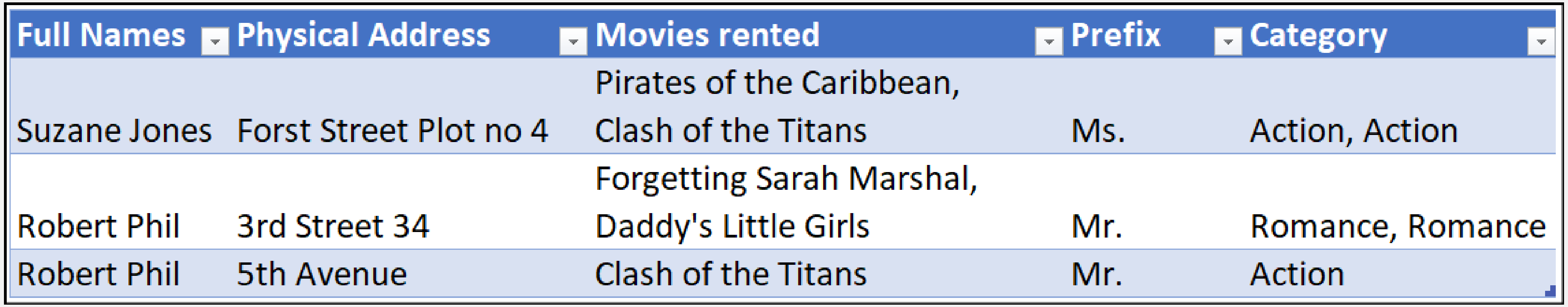
It divides larger tables to smaller tables and links them using relationships.

TYPES OF NORMAL FORM



EXAMPLE , 1NF

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



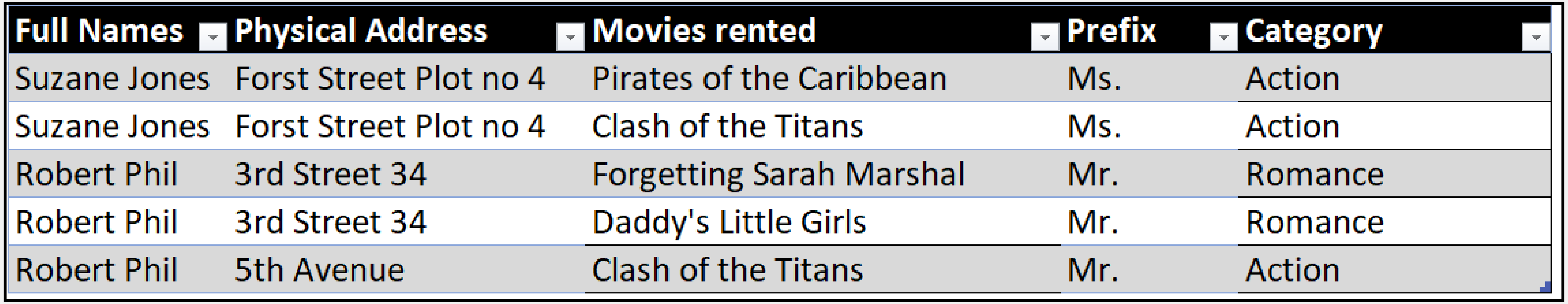
Full Names	Physical Address	Movies rented	Prefix	Category
Suzane Jones	Forst Street Plot no 4	Pirates of the Caribbean, Clash of the Titans	Ms.	Action, Action
Robert Phil	3rd Street 34	Forgetting Sarah Marshal, Daddy's Little Girls	Mr.	Romance, Romance
Robert Phil	5th Avenue	Clash of the Titans	Mr.	Action

Here you see Movies Rented column has multiple values.

EXAMPLE , 1NF

1NF (First Normal Form) Rule

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



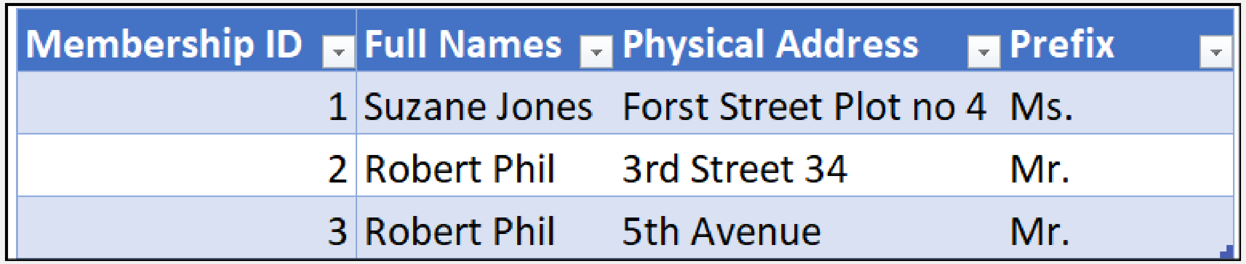
Full Names	Physical Address	Movies rented	Prefix	Category
Suzane Jones	Forst Street Plot no 4	Pirates of the Caribbean	Ms.	Action
Suzane Jones	Forst Street Plot no 4	Clash of the Titans	Ms.	Action
Robert Phil	3rd Street 34	Forgetting Sarah Marshal	Mr.	Romance
Robert Phil	3rd Street 34	Daddy's Little Girls	Mr.	Romance
Robert Phil	5th Avenue	Clash of the Titans	Mr.	Action

EXAMPLE , 2NF

Rule 1- Be in 1NF
Rule 2- Single Column Primary Key & No Partial Dependency

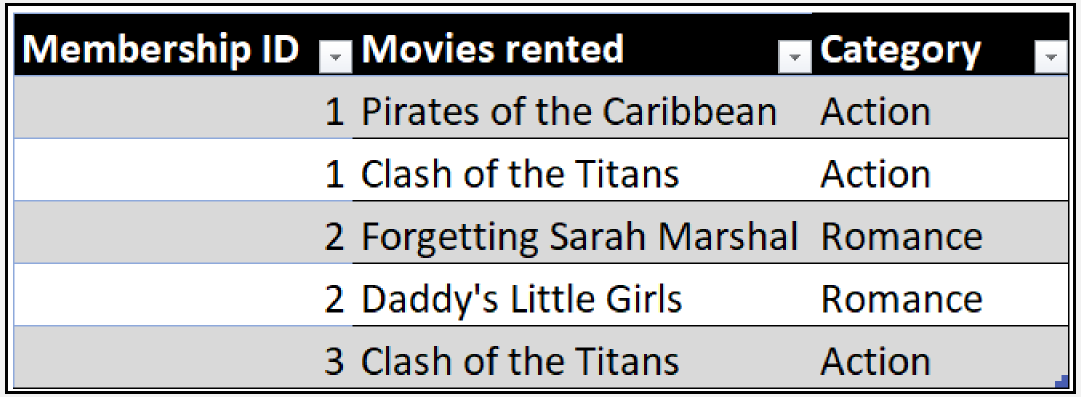
It is clear that we can't move forward to make our simple database in 2 Normalization form unless we partition the table above.

PRIMARY KEY



Membership ID	Full Names	Physical Address	Prefix
1	Suzane Jones	Forst Street Plot no 4	Ms.
2	Robert Phil	3rd Street 34	Mr.
3	Robert Phil	5th Avenue	Mr.

FOREIGN KEY

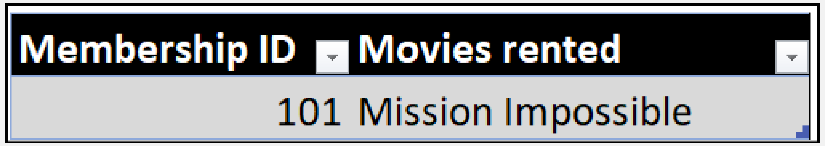


Membership ID	Movies rented	Category
1	Pirates of the Caribbean	Action
1	Clash of the Titans	Action
2	Forgetting Sarah Marshal	Romance
2	Daddy's Little Girls	Romance
3	Clash of the Titans	Action

We have introduced a new column called Membership_id which is the primary key for table 1. Records can be uniquely identified in Table 1 using membership id

EXAMPLE , 2NF……

Why do you need a foreign key?
Suppose some one try to insert as mentioned below



Membership ID	Movies rented
101	Mission Impossible

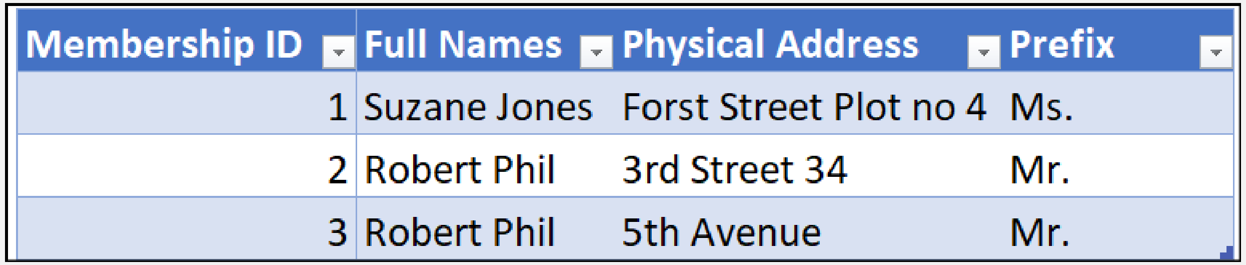
Here Database will throw error as Member id 101 is not present in table 1

You will only be able to insert values into your foreign key that exist in the unique key in the parent table. This helps in referential integrity.

Now, if somebody tries to insert a value in the membership id field that does not exist in the parent table, an error will be shown!

FUNCTIONAL DEPENDENCY

PRIMARY KEY



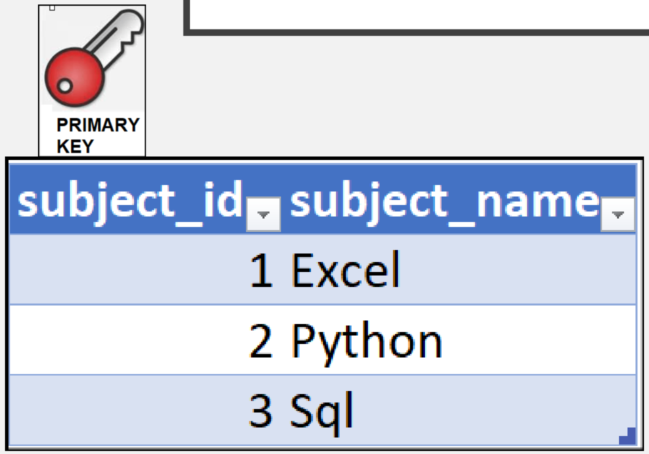
Membership ID	Full Names	Physical Address	Prefix
1	Suzane Jones	Forst Street Plot no 4	Ms.
2	Robert Phil	3rd Street 34	Mr.
3	Robert Phil	5th Avenue	Mr.

Membership_ID ---> Full Names
Membership_ID ---> Physical Address
Membership_ID ---> Prefix

Every column is dependent on Membership id , hence it is known as Dependency or mainly Functional dependency

PARTIAL DEPENDENCIES

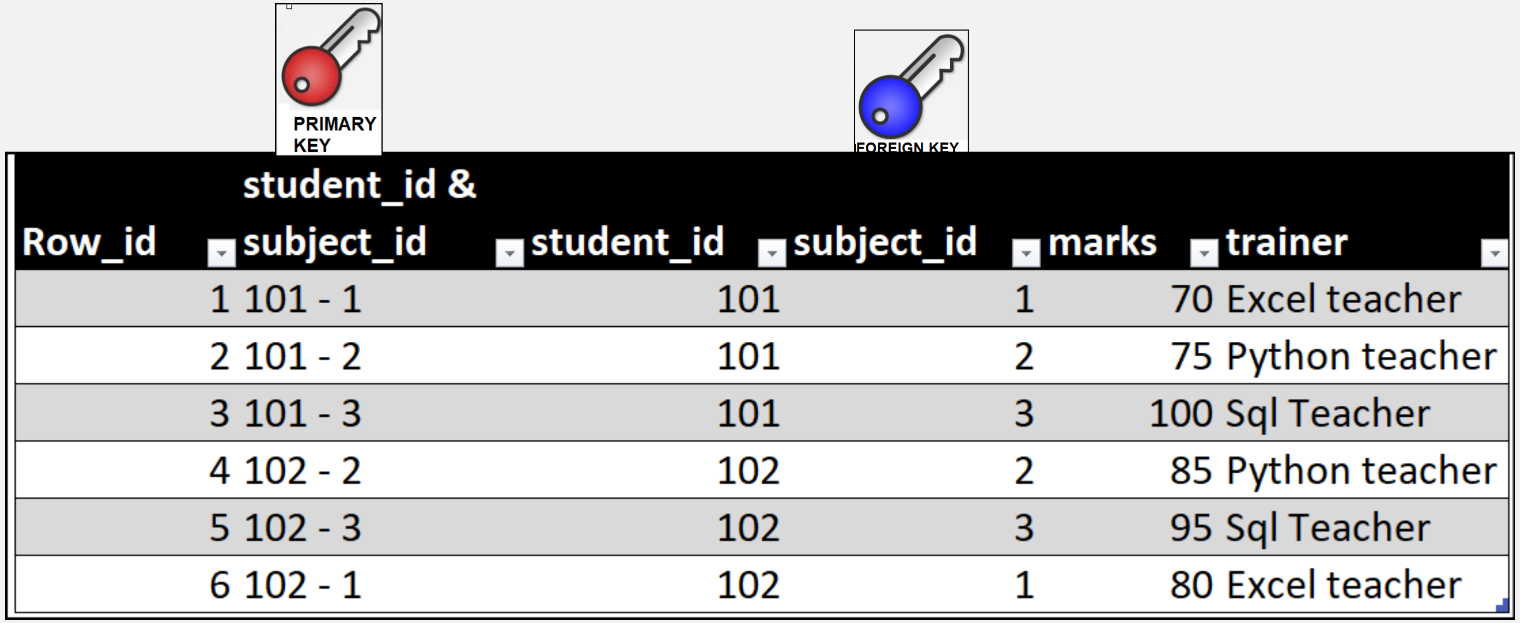
Subject


PRIMARY KEY

subject_id	subject_name
1	Excel
2	Python
3	Sql

- Student_id + Subject_id -→ marks
- Student_id + Subject_id -→ trainer
- Subject_id -→ trainer

Marks


PRIMARY KEY
FOREIGN KEY

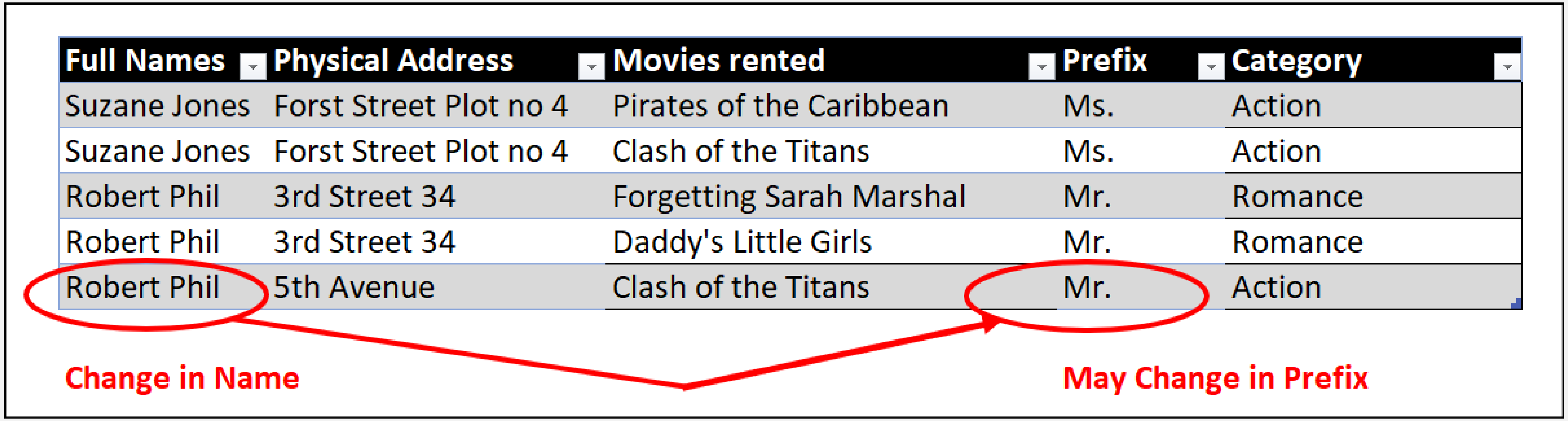
Row_id	student_id & subject_id	student_id	subject_id	marks	trainer
1	101 - 1	101	1	70	Excel teacher
2	101 - 2	101	2	75	Python teacher
3	101 - 3	101	3	100	Sql Teacher
4	102 - 2	102	2	85	Python teacher
5	102 - 3	102	3	95	Sql Teacher
6	102 - 1	102	1	80	Excel teacher

this is partial dependency where an attribute in a table depends on only part of primary key and not on whole key

TRANSITIVE FUNCTIONAL DEPENDENCIES

A transitive functional dependency is when changing a non-key column, might cause any of the other non-key columns to change

Consider the table 1. Changing the non-key column Full Name may change Prefix



Full Names	Physical Address	Movies rented	Prefix	Category
Suzane Jones	Forst Street Plot no 4	Pirates of the Caribbean	Ms.	Action
Suzane Jones	Forst Street Plot no 4	Clash of the Titans	Ms.	Action
Robert Phil	3rd Street 34	Forgetting Sarah Marshal	Mr.	Romance
Robert Phil	3rd Street 34	Daddy's Little Girls	Mr.	Romance
Robert Phil	5th Avenue	Clash of the Titans	Mr.	Action

Change in Name

May Change in Prefix

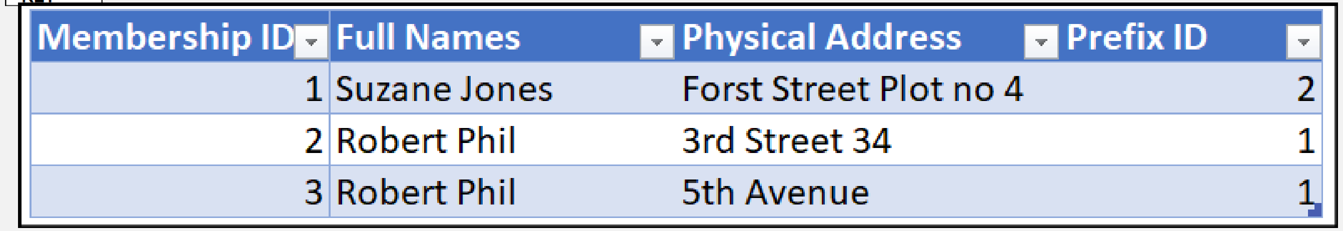
EXAMPLE , 3NF

Rule 1- Be in 1NF
Rule 2- Has no transitive functional dependencies

To move our 2NF table into 3NF, we again need to again divide our table.

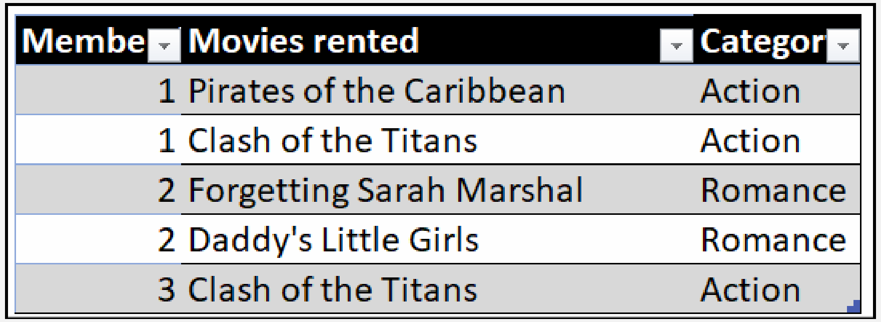
PRIMARY KEY

FOREIGN KEY



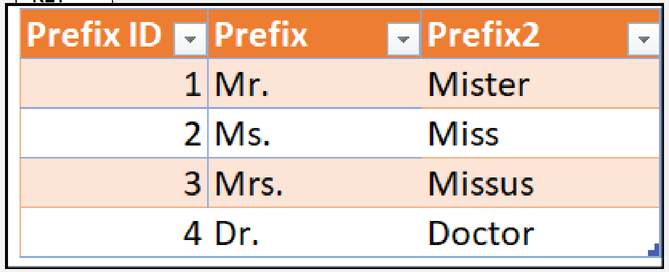
Membership ID	Full Names	Physical Address	Prefix ID
1	Suzane Jones	Forst Street Plot no 4	2
2	Robert Phil	3rd Street 34	1
3	Robert Phil	5th Avenue	1

FOREIGN KEY



Membe	Movies rented	Categor
1	Pirates of the Caribbean	Action
1	Clash of the Titans	Action
2	Forgetting Sarah Marshal	Romance
2	Daddy's Little Girls	Romance
3	Clash of the Titans	Action

PRIMARY KEY



Prefix ID	Prefix	Prefix2
1	Mr.	Mister
2	Ms.	Miss
3	Mrs.	Missus
4	Dr.	Doctor

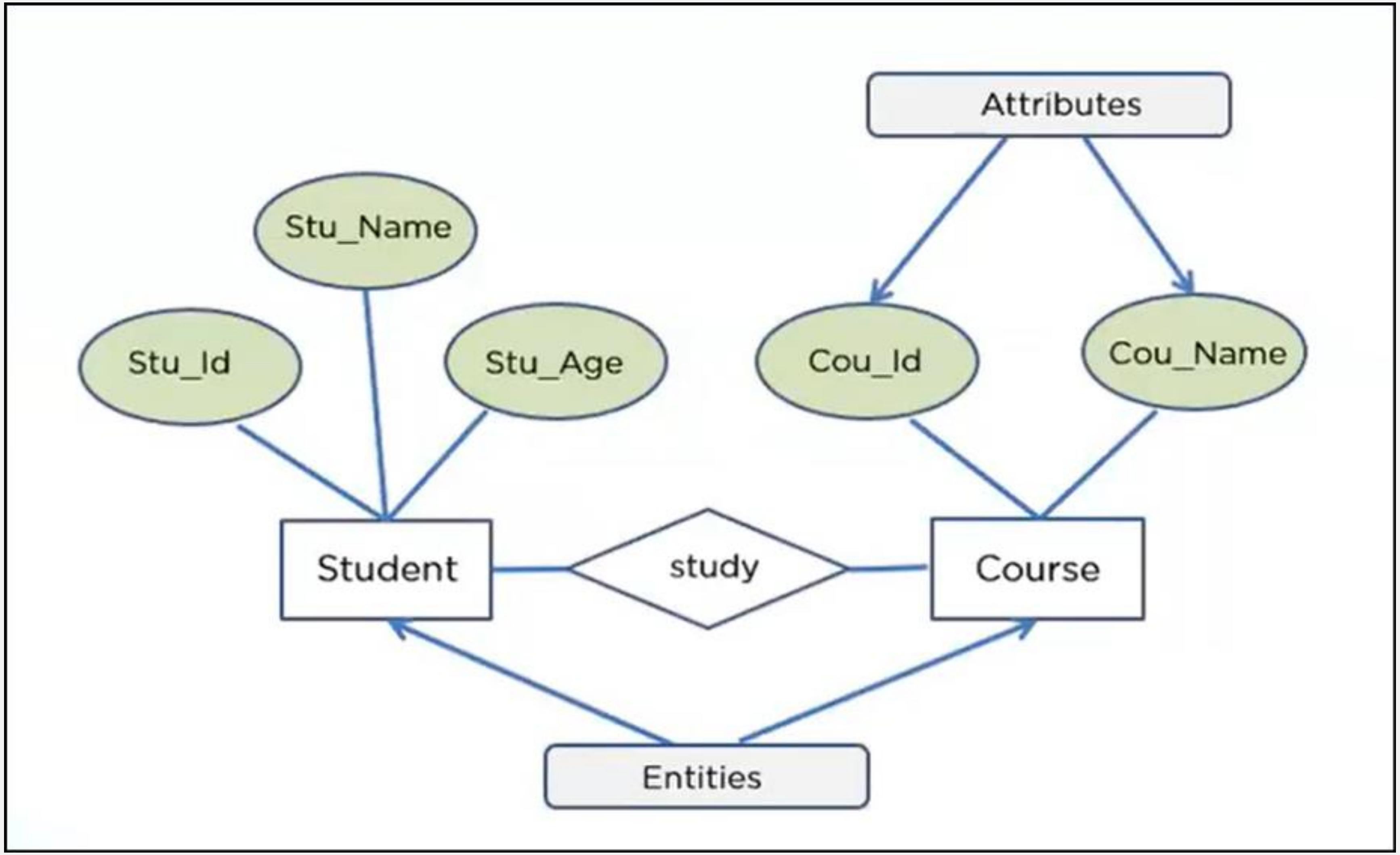
We have again divided our tables and created a new table which stores Prefix
There are no transitive functional dependencies, and hence our table is in 3NF

CONCLUSION

Now our little example is at a level that cannot further be decomposed to attain higher forms of normalization. In fact, it is already in higher normalization forms. Separate efforts for moving into next levels of normalizing data are normally needed in complex databases.

SHORT BREAK

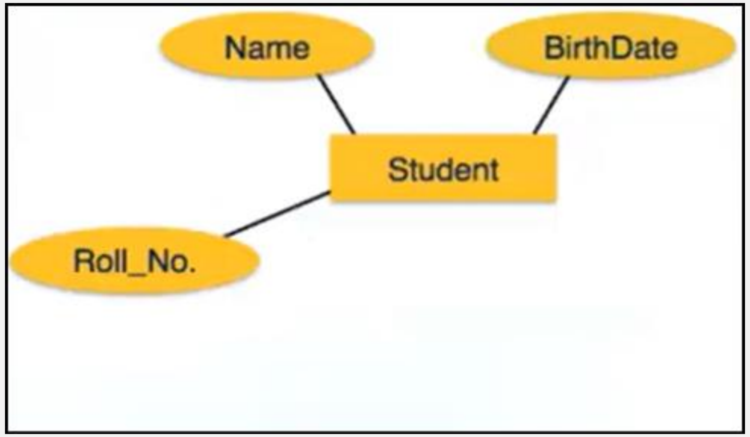
ENTITY RELATIONSHIP DIAGRAMS

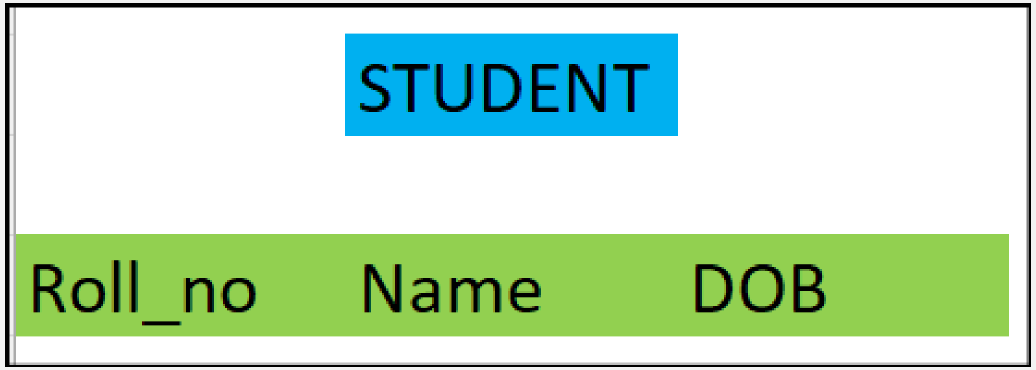


- □ Entity: Table
- ○ Attribute: Column
- ◇: Relationship

4 TYPES OF ATTRIBUTE

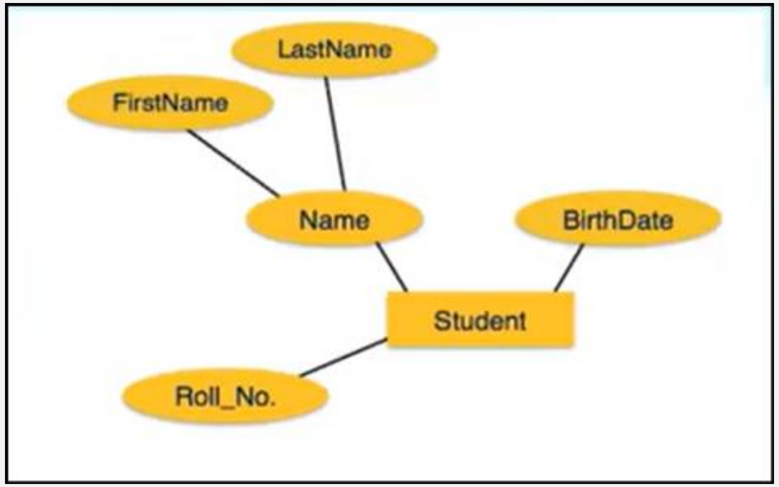
Simple



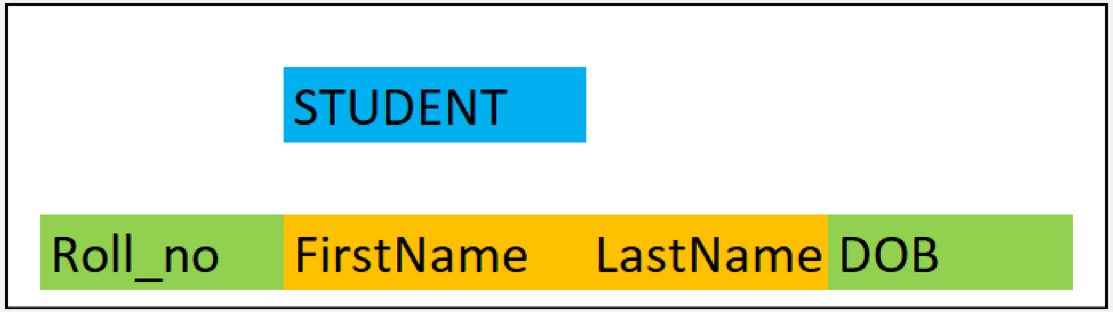

STUDENT

Roll_no	Name	DOB

Composite



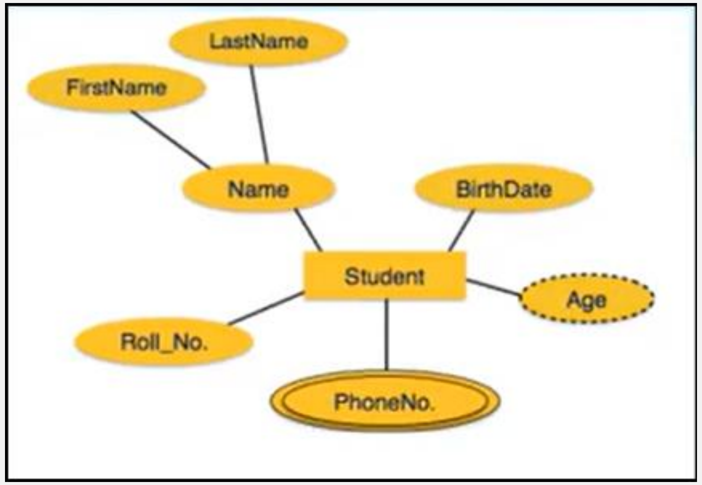
Combination of First & Last Name


STUDENT

Roll_no	FirstName	LastName	DOB

4 TYPES OF ATTRIBUTE

Multi-values

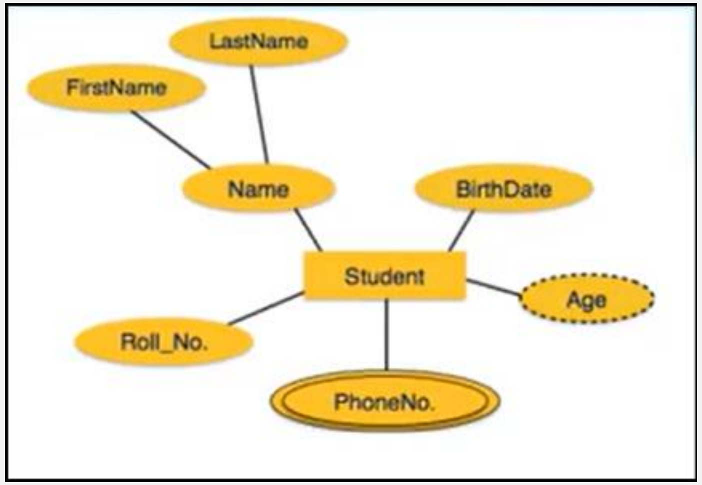


2 Mobile number

STUDENT

Roll_no	FirstName	LastName	DOB	Mobile1	Mobile2

Derived



Age is calculated based on DOB

STUDENT

Roll_no	FirstName	LastName	DOB	Age
				Formula

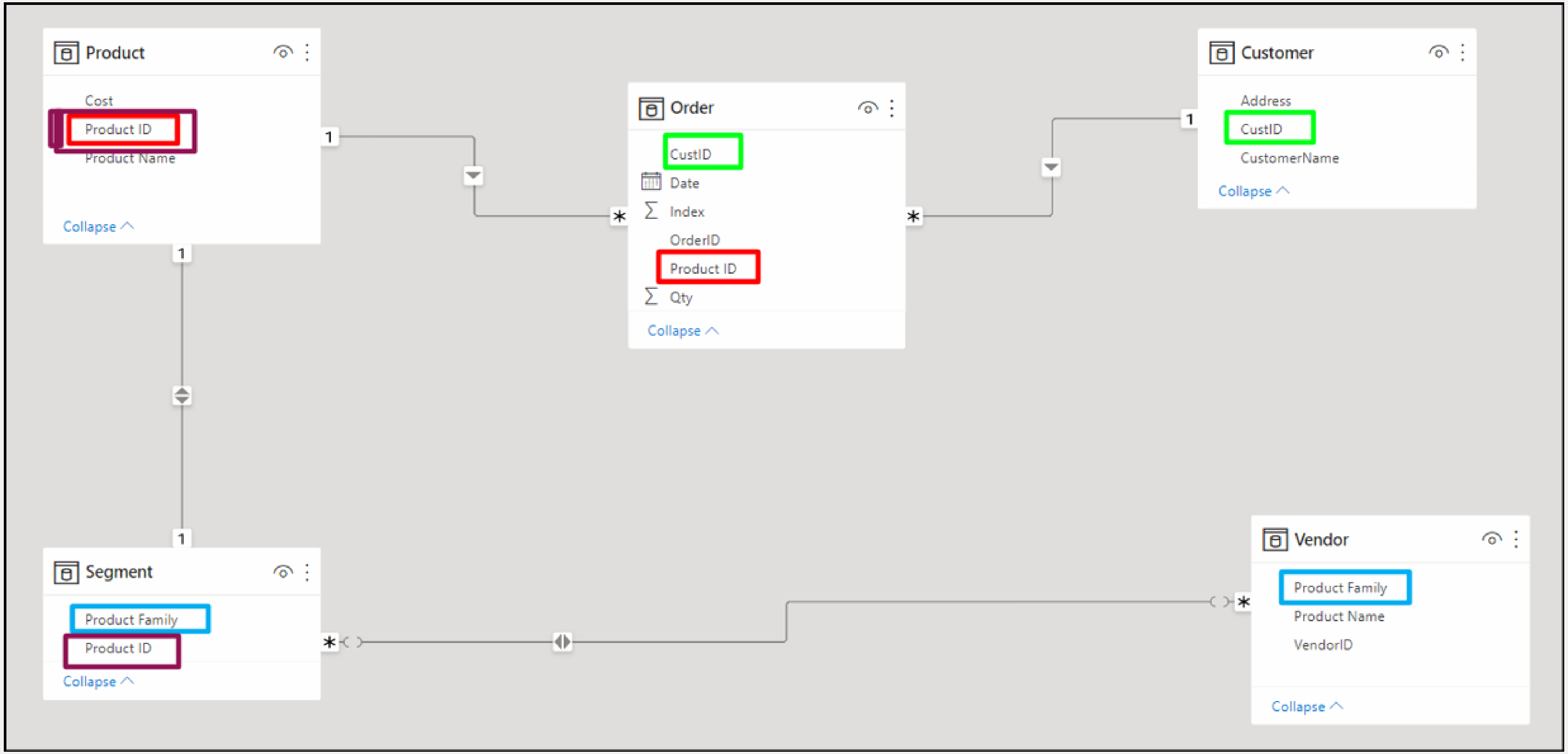
RELATIONSHIPS & CARDINALITY

Cardinality represents the number of times entity of an entity set participates in a relation set

Types of Cardinality between 2 tables

- One to Many (1 -*)
- Many to One (* -1)
- One to One (1- 1)
- Many to Many (* - *)

ER MODEL



Product
Cost
Product ID
Product Name
Collapse
Order
CustID
Date
Index
OrderID
Product ID
Qty
Collapse
Customer
Address
CustID
CustomerName
Collapse
Segment
Product Family
Product ID
Collapse
Vendor
Product Family
Product Name
VendorID
Collapse
1
1
1
1
*
*
*

QUESTION
N
ANSWER

THANK YOU