

## Relation Model

The Entities that were mapped to Relations in the new model are -

1. User -> USER
2. Post -> POST
3. Comment -> COMMENT
4. Group -> GROUP
5. Message -> MESSAGE
6. Page -> PAGE

The Weak Entities are mapped to Relations using the identifying relationships as foreign key as -

1. Profile -> PROFILE (PK = "user\_id" from USER )
2. React -> MAKES\_A\_REACT (PK = "comment\_id" from COMMENT + "user\_id" from USER )  
-> MAKES\_A\_GENERAL\_REACT (PK = "user\_id" from USER + "post\_id" from POST )  
-> RESPONDS (PK = "user\_id" from USER + "story\_id" from STORY)

The sub-classes of Page were mapped to Relations in the new model as -

1. Brand/Product -> BRAND\_PRODUCTS
2. Local Business -> BUSINESS\_PLACE
3. Company/Organisation/Institution -> COMPANY
4. Artists/Band/Public Figure -> PUBLIC\_FIGURE
5. Cause/Community -> CAUSE\_COMMUNITY
6. Entertainment -> ENTERTAINMENT

The relations created to map the multi-valued attributes of other entities are as follows-

1. News(Public Figure) -> NEWS\_PUB\_FIG
2. Products(Business Place) -> PROD\_BP
3. Branches(Company) -> BRANCH\_COMPANY

The Binary Relationships of structural constraints 1:N or N:1 are mapped by using foreign keys in respective entity relations.

1. User CREATES Post -> user\_id from USER as a foreign key in POST
2. User IS\_OWNER Page -> owner\_id from USER as a foreign key in PAGE

The Binary Relationships of structural constraints 1:N or N:1 are mapped by creating new relations.

1. User FOLLOWS User -> FOLLOWS
2. User LIKES Page -> PAGE
3. User BELONGS\_TO Group -> BELONGS\_TO
4. User IS\_ADMIN Group -> IS\_ADMIN
5. User IS\_MODERATOR Group -> IS\_MODERATOR
6. User IS\_TAGGED Post -> IS\_TAGGED

The N-ary Relationships are mapped by creating new relations.

1. User COMMENTS a Comment on a Post -> COMMENTS

2. User MENTIONS a User in a Comment in a Post -> MENTIONS
3. User SENDS\_SPECIFIC Message to a User -> SENDS\_SPECIFIC
4. User SENDS\_GENERAL Message to a Group -> SENDS\_GENERAL
5. User SHARES a Post in a Group -> SHARES

## 1NF Model

The Relation Model is devoid of multivalued/ composite attributes. So the relation model passes the 1NF test. Hence, no modifications required.

## 2NF Model

In the current model, two relations failed the 2NF test,

1. MENTIONS: The “post\_id” attribute is functionally dependent on “comment\_id”. However, the primary key of the relation is “mentioner\_id”+ “mentioneed\_id”+ “comment\_id”. Thus, “post\_id” is partially dependent. To resolve the case, we just dropped the “post\_id” attribute from the relation. And we don’t make another relation since the existing COMMENTS relation already projects the functional dependency of “post\_id” on “comment\_id”.
2. MAKES\_A\_REACT: The “post\_id” attribute is functionally dependent on “comment\_id”. However, the primary key of the relation is “user\_id” + “comment\_id”. Thus, “post\_id” is partially dependent. To resolve the case, we just dropped the “post\_id” attribute from the relation. And we don’t make another relation since the existing COMMENTS relation already projects the functional dependency of “post\_id” on “comment\_id”.

With the above modifications, the model passes the 2NF test.

## 3NF Model

The current Model as modified for 2NF test, is devoid of any transitive functional dependencies. So the relation model passes the 3NF test. Hence, no modifications required.