RELATIONAL MODEL & NORMAL FORMS

Team : Eureka!

Pavani Babburi : 2019101033
Alapan Sau : 2019101081
Nirmal Manoj C: 2019111011

Relational 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
4. Education(Profile) -> EDUCATION

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

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

N.B. The attribute “No of likes” from Page in ER Model is deprecated in PAGE relation as we want to implement it as a functional requirement.

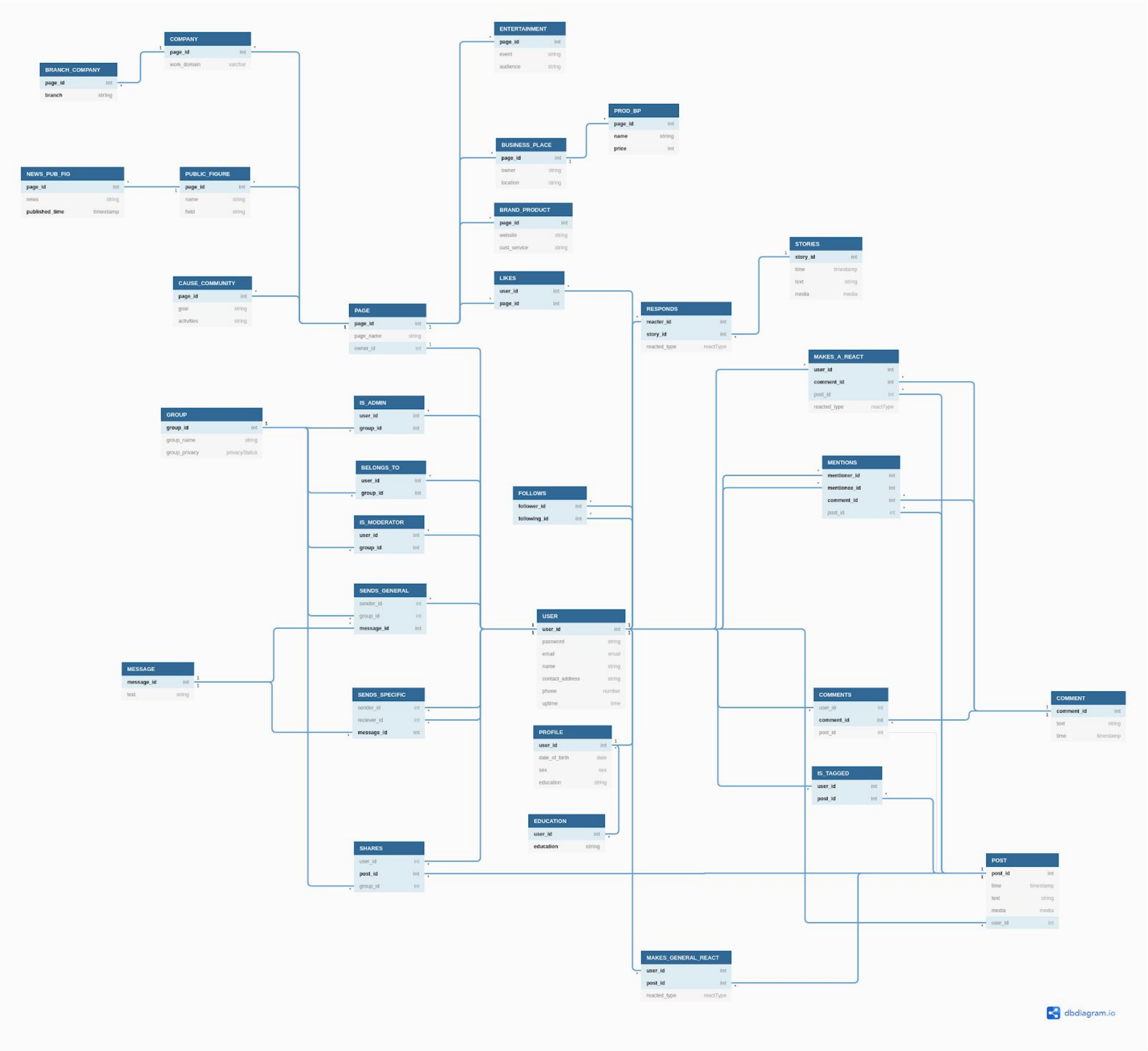

Fig : Relational Model (Mapped from The ER Diagram)

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"+ "mentionee_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".

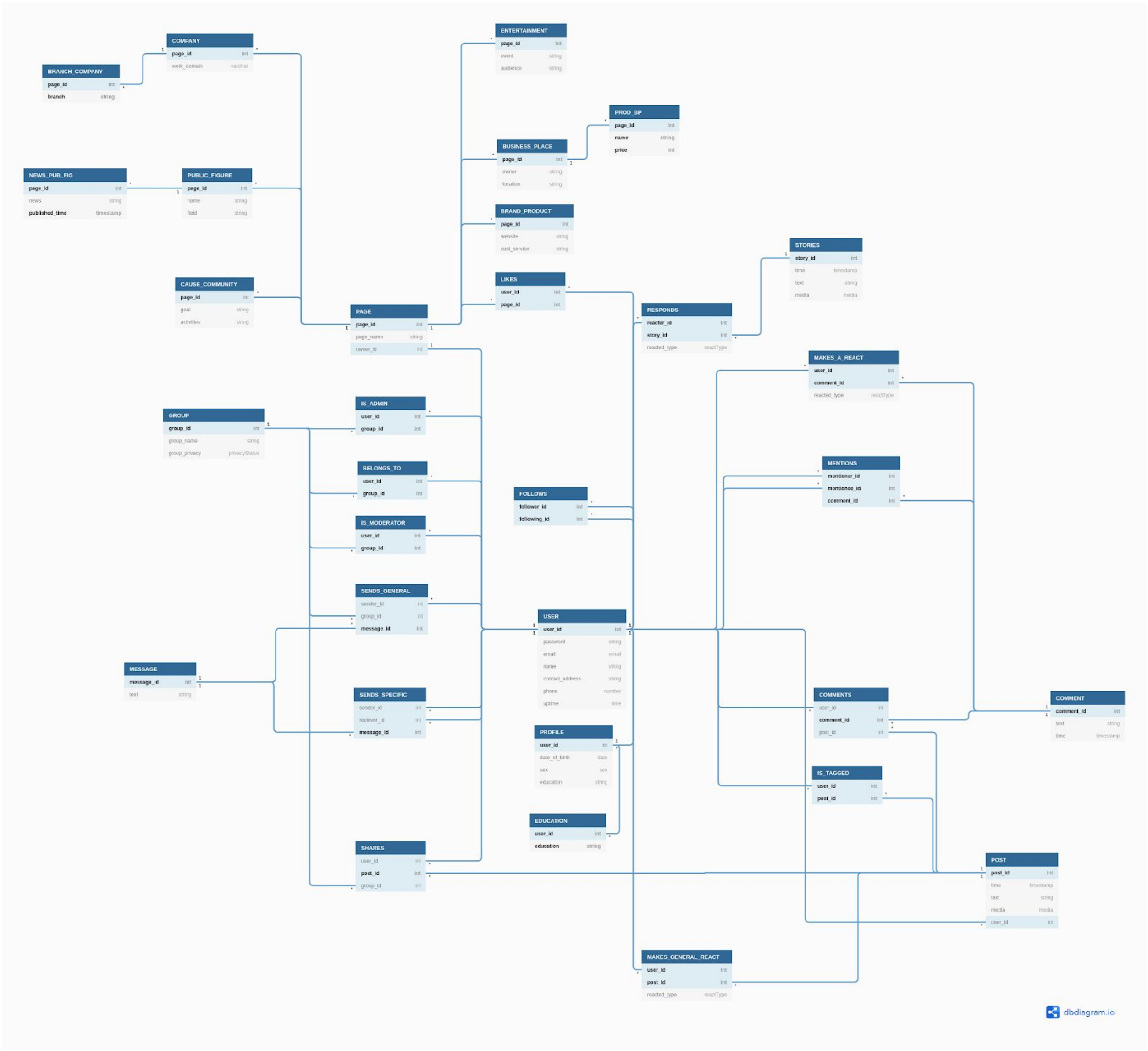

Fig : 2NF Model

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.