Nama : Nicholas Setiono

Nama Discord : Nich728

Email : nicholas032@binus.ac.id

ENTITY RELATIONSHIP DIAGRAM

1. Answer can be viewed in ERD
2. Master -> Child
   - Users -> Page Likes
   - Users -> Friends
   - Users -> Posts
   - Users -> Post Likes
   - Users -> Comments
   - Users -> Comment Likes
   - Users -> Shares
   - Pages -> Page Likes
   - Comment -> Comment Likes
   - Posts -> Comment
   - Posts -> Photos
   - Posts -> Shares
   - Posts -> Post Likes

3. I gave most of the objects a PRIMARY KEY, but some don't need them. I also gave all the child tables FOREIGN KEYs as they are referenced to the master table. I gave a CHECK constraint to the gender attribute in Users as I made them use CHAR(1) as a data type, meaning they can only have a char of length 1. It will check whether the input is 'B' or 'G', else it will not pass the check.
4. ERD can be viewed in Github

DATA DEFINITION LANGUAGE

1. Data Integrity is the overall accuracy, reliability, completeness, and consistency of data. The higher the integrity, the better. Therefore, it is best to commonly check your data integrity and ensure that it will stay good. Making a clear and good relationship between objects in SQL Server will help maintain the data integrity.

2. Primary Key means a key attribute which defines what an object is. It differentiates an object with another. Even if there are two of the same type of objects, we will be able to differentiate them using this key. For example, a student will have a student id to help differentiate them among other students.

   Foreign Key means a key attribute that is referenced from another object. For example, a post in social media will have a key we can call "UserId". This key will be referenced from the User object, meaning the "UserId" in Post will be the same as the "id" in Users.

   A Composite Key is a primary key that has two or more attributes. So, an object can have more than 1 key that differentiates them among others.

3. BEGIN TRAN : This marks the start or the beginning of a transaction. This will work as a checkpoint when making changes to your data.

   COMMIT : This will save all the changes made between BEGIN TRAN and COMMIT. You cannot use COMMIT without using BEGIN TRAN before. COMMIT will also end the transaction.

   ROLLBACK : This will remove all the changes made and revert the current data into the condition it had at the beginning of the transaction. In other words, it will revert the data into the state before BEGIN TRAN. This can also only be used after you use BEGIN TRAN. If you use ROLLBACK or COMMIT without using BEGIN TRAN before, it will cause an error.

4. You can view the commands I used in Github