Entity Relationship Diagram

1. From the above diagram, list all of the objects including its attributes!
    a. Users:
        - User ID
        - First name
        - Last name
        - Gender
        - Location
        - Date of Birth
        - Address
        - School
        - Email
        - Phone Number

    b. Friends:
        - Friend ID
        - User ID

    c. Pages:
        - Page ID
        - Page Name
        - Page Content

    d. Page Likes:
        - User ID
        - Page ID

    e. Posts:
        - Post ID
        - User ID
        - Post Date
        - Post Content

    f. Post Likes:
        - Post ID
        - User ID

    g. Photos
        - Photo ID
        - Post ID
        - Image Content

h. Shares:
   - Post ID
   - User ID

i. Comments:
   - Comment ID
   - Post ID
   - User ID
   - Comment date
   - Comment content

j. Comment Likes:
   - Comment ID
   - User ID

2. Determine the relation between every object, specify the master and child table!
   a. Users (master) have:
      - many Friends (child)
      - many Page Likes (child) which is limited to 1 Page only
      - many Posts (child)

   b. Pages (master) have:
      - many Page Likes (child) which is limited to 1 User only

   c. Posts (master) have:
      - Post Likes (child)
      - Photos (child)
      - Shares (child)
      - Comments (child)

   d. Comments (master) have:
      - Comment Likes (child)

3. For each object, decide its constraint and specify the reason in detail!
   a. Users:
      - User ID (Primary Key (PK) with increment)
      - First name (not NULL, user has at least 1 name)
      - Last name (NULL-able, not every user has a last name)
      - Gender (not NULL, two choices – CHECK male OR female)
      - Location (not NULL, specify your country)
      - Date of Birth (not NULL with date data type, every human has a date of birth)
      - Address (not NULL, specify where do you live)
      - School (NULL-able, a user might not go to any schools at all)

- Email (not NULL, CHECK LIKE '%@%')
- Phone Number (not NULL, when you are posting you must have a phone number to be verified)

b. Friends:
   - Friend ID (PK)
   - User ID (FK from Users)

c. Pages:
   - Page ID (PK)
   - Page Name (not NULL)
   - Page Content (not NULL)

d. Page Likes:
   - User ID (FK from Users)
   - Page ID (FK from Pages)

e. Posts:
   - Post ID (PK)
   - User ID (FK from Users)
   - Post Date (not NULL with date data type)
   - Post Content (not NULL, write captions → if you make it NULL then do not post anything please)

f. Post Likes:
   - Post ID (FK from Posts)
   - User ID

g. Photos
   - Photo ID (PK)
   - Post ID (FK from Posts)
   - Image Content (NULL-able with blob data type → image, if you don't want to post images then make it NULL)

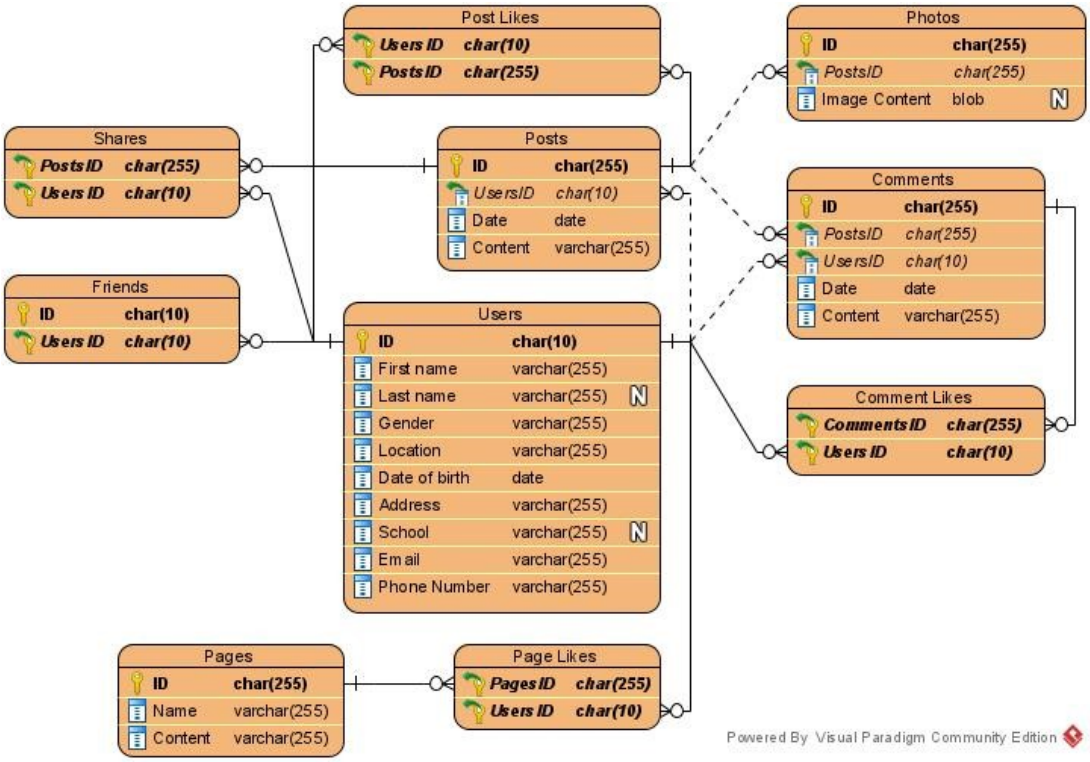
h. Shares:
   - Post ID (FK from Posts)
   - User ID

i. Comments:
   - Comment ID (PK)
   - Post ID (FK from Posts)
   - User ID
   - Comment date (not NULL with date data type)

- Comment content (not NULL, don't need to comment if it is NULL)

j. Comment Likes:
- Comment ID (FK from Comments)
- User ID

4. Gambar ERD:



Data Definition Language

1. Explain what is data integrity and how do we maintain it in SQL Server!
   Data integrity defines how consistent and accurate a data of the table is. Data integrity is maintained by a collection of processes, rules/constraints, and standards during the design phase.

2. Explain the difference and give example for: primary key, foreign key, and composite key!
   - Primary Key: Unique and Not NULL; a key that uniquely identifies an entity. Examples are Binusian NIM and KTP number.
   - Foreign Key: a key that creates a relationship between two entities. Main purpose is to allow the navigation between the two entities. Example, between Users and Posts. One user can have many posts, in which the posts are the user's, not anyone else. That is why in the entity Posts we need a foreign key that referenced to the Users entity.

- Composite Key: a key that combines 2 or more columns in a table to be a primary key. It is used to uniquely identify each row. Example is Post Likes, it needs both Users ID and Posts ID to identify.

3. Explain the following terms and give example: BEGIN TRAN, COMMIT, and ROLLBACK!
   - BEGIN TRAN
     Marks the starting point of a local transaction (acts as a checkpoint).

   - COMMIT
     Ends the transaction; can't use ROLLBACK again.

   - ROLLBACK
     Undo the last transaction made; if you UPDATE something, then you can ROLLBACK to the data before you do the UPDATE (other words, undo.)

   - Example:
     BEGIN TRAN
     UPDATE Heroes
     Set HeroNames = NULL
     ROLLBACK
     COMMIT