1.

Objects:

- Users
- Friends
- Pages
- Page Likes
- Posts
- Post Likes
- Photos
- Shares
- Comments
- Comment Likes

Attributes:

- Users:
    - ID
    - First Name
    - Last Name
    - School
    - Address
    - Email
    - Phone Number
    - Location
    - Date of Birth
    - Gender
- Friends:
    - **RelationID**
    - User ID
    - Friend ID
- Pages
    - ID
    - Page Name
    - Page Content
- Page Likes
    - **UserID**
    - **PageID**
- Posts
    - ID
    - Post Date
    - Post Content
    - User ID
- Post Likes
    - **Post ID**
    - **User ID**
- Photos
    - ID

        - Post ID
        - Image Content
    - Shares
        - **Post ID**
        - **User ID**
    - Comments
        - **ID**
        - Comment Date
        - Comment Content
        - User ID
        - Post ID
    - Comment Likes
        - **Comment ID**
        - **User ID**

2. Relation:

Users one to many:

- Posts
- Friends
- Page Likes
- Shares
- Comment Likes
- Post Likes
- Comments

Pages one to many:

- Page Likes

Posts one to many:

- Photos
- Post Likes
- Shares
- Comments
- **Many to one**: Users

Shares many to one:

- Users
- Posts

Post Likes many to one:

- Users
- Posts

Comment one to many:

- Comment Like
- **Many to One:** Users

Photos many to one:

- Posts

Comment Likes many to one:

- Comments
- Users

Shares many to one:

- Users
- Posts

Friends many to one:

- Users

Master and child relations:

MASTER	CHILD
User	Friend
User	Shares
User	Post Likes
User	Posts
User	Comments
User	Comment Likes
User	Page Likes
Pages	Page Likes
Posts	Photos
Posts	Comments
Posts	Shares
Posts	Post Likes

3. Constraint Each Objects

Users:

ID = NOT NULLABLE PRIMARY KEY (tidak dapat dikosongkan dan merupakan pembeda didalam table Users)

First name = NOT NULLABLE (semua orang mempunyai nama depan jadi tidak dapat null)

Last name = NULLABLE (saya punya beberapa teman yang hanya mempunyai 1 kata nama, tanpa nama belakang)

Gender = NOT NULLABLE (jenis kelamin dapat diindikasikan dengan char ‘L’ atau ‘P’)

Email = NOT NULLABLE CHECK ‘%@%.%’ (not nullable karena jaman skrg akun social media membutuhkan verifikasi email, butuh format email dimana untuk validasi)

School = NULLABLE (nullable karena tidak wajib)

Address = NULLABLE (nullable karena tidak wajib)

Location = NULLABLE (nullable karena tidak wajib)

Phone number= NOT NULLABLE (terkadang verifikasi atau reset akun membutuhkan data phone number)

Date of Birth = NOT NULLABLE (informasi dasar yang dimiliki tiap orang)

Friends

RelationID = PRIMARY KEY (Sebagai atribut tunggal dalam relasi antar user dan friend)

UsersID = FOREIGN KEY references Users (penghubung dengan table users)

FriendID = NOT NULLABLE

Posts

ID = PRIMARY KEY (sebagai atribut tunggal dlm table post dengan integer 10 digit)

Post Content = NOT NULLABLE (isi content dibutuhkan)

Post Date = NOT NULLABLE (auto mekanisme untuk mencatat waktu post)

Photos

ID = PRIMARY KEY (sebagai atribut tunggal dlm table post dengan integer 10 digit)

Image Content = NOT NULLABLE (berupa link, yang dimana terdapat image untuk content jd tidak dpt null)

PostsID = FOREIGN KEY references Posts (Sebagai penghubung dengan table Posts)

Comments

ID = PRIMARY KEY (sebagai atribut tunggal dlm table post dengan integer 10 digit)

Comment Date = NOT NULLABLE (auto mekanisme untuk mencatat waktu post, jadi tidak mungkin comment tanpa date)

Comment Content = NOT NULLABLE (dibutuhkan isi untuk content dari comment, meskipun hanya 1 karakter)

PostsID = FOREIGN KEY references Posts (Sebagai penghubung dengan table Posts)

UsersID = FOREIGN KEY references Users (penghubung dengan table users)

COMPOSITE KEY TABLE (Unique kalua terdapat 2 column tersebut, jika dipisah tidak unique lagi) {

Post Likes

UsersID = PRIMARY KEY FOREIGN KEY references Users

PostsID = PRIMARY KEY FOREIGN KEY references Posts

Shares

UsersID = PRIMARY KEY FOREIGN KEY references Users

PostsID = PRIMARY KEY FOREIGN KEY references Posts

Page Likes

UsersID = PRIMARY KEY FOREIGN KEY references Users

PagesID = PRIMARY KEY FOREIGN KEY references Pages

Post Likes

UsersID = PRIMARY KEY FOREIGN KEY references Users

CommentsID = PRIMARY KEY FOREIGN KEY references Comments

}

DATA DEFINITION LANGUANGE

1. Explain what is data integrity and how do we maintain it in SQL Server!

   Data integrity mengacu pada keakuratan dan konsistensi dari suatu data.

   Ada beberapa cara kita untuk melakukan *maintain* terhadap data integrity

- Entity Integrity
  Dengan mendefine unique di dalam table menggunakan primary key
  PRIMARY KEY
- Referential Integrity
  Dengan melakukan koneksi / match agar foreign key terhubung dengan primary key
  Hal ini dapat dilakukan di dalam sql dengan perintah:
  ON UPDATE [CASCADE | SET NULL | SET DEFAULT]
  ON DELETE [CASCADE | SET NULL | SET DEFAULT]
- Domain Integrity
  Dengan melakukan validitas untuk data yang akan dimasukkan kedalam table, hal ini dapat kita lakukan dengan memasukkan constraint atau define data size dengan kemungkinan data yang akan dimasukkan

2. Explain the difference and give example for: primary key, foreign key, and composite key!

Perbedaannya adalah primary key merupakan suatu atribut unique dari suatu table, sedangkan foreign merupakan atribut yang merupakan references dari atribut lain dan composite merupakan atribut yang di references dari table lain dan merupakan suatu unique/identifier yang biasanya berjumlah 2 atau lebih.

CONTOH:

PRIMARY KEY: Mahasiswa yang di identifikasi menggunakan NIM sbg PRIMARY KEY

FOREIGN KEY: Terdapat data table di TOKO A membutuhkan data NIM mahasiswa sebagai syarat transaksi, maka dari itu hal tersebut di references ke table Mahasiswa

COMPOSITE KEY: Detail dari pembelian buku tersebut akan dimasukkan kedalam table dengan mengambil NIM Mahasiswa dan ID Buku, maka daftar gabungan NIM dan ID akan menjadi unique bila dipasangkan

3. Explain the following terms and give example: BEGIN TRAN, COMMIT, and ROLLBACK!

BEGIN TRAN merupakan statement sebagai penanda awal sebelum terjadinya perubahan, COMMIT merupakan statement untuk mengunci suatu perubahan yang menandakan bahwa perubahan success dan tidak dapat di rollback kembali, sedangkan ROLLBACK merupakan statement yang digunakan untuk Kembali ke penanda awal atau *savepoint* dari suatu transaksi.

CONTOH:

Di ibaratkan sebuah Game, Game tersebut dapat melakukan jual beli antar user. Lalu perusahaan tersebut melakukan update-update didalam game yang menyebabkan BUG yang membuat para Users dapat melakukan duplicate didalam game. Sebelum melakukan update-update tersebut pihak developer akan melakukan BEGIN TRAN. Lalu apabila sudah terverifikasi BUG ditemukan maka akan dilakukan ROLLBACK, dan akan kembali ke keadaan sebelum update. Maka semua transaksi antar user dan BUG duplicate akan ter *roll back* ke semula. Apabila sudah terindikasi bahwa update tidak terdapat BUG, maka akan dilakukan COMMIT sebagai *finishing* untuk update versi tersebut.