

Entity Relationship Diagram [50%]

1. From the above diagram, list all of the objects including its attributes! (.pdf)

Object	Attributes
Users	User ID
	First Name
	Last Name
	School
	Address
	Email
	Phone Number
	Location
	Date of Birth
	Gender
Friends	Friend ID
	User ID
Posts	Post ID
	Post Date
	Post Content
	User ID
Page Likes	User ID
	Page ID
Pages	Page ID
	Page Name
	Page Content
Post Likes	Post ID
	User ID
Photos	Photo ID
	Image Content
	Post ID
Shares	Post ID
	User ID
Comments	Comment ID
	Comment Date
	Comment Content
	User ID
	Post ID
Comment Likes	Comment ID
	User ID

2. Determine the relation between every object, specify the master and child table! (.pdf)

Relation

Master	Relation	Child
Users	1 to many	Posts
		Friends
		Page Likes
Posts	1 to many	Post Likes
		Photos
		Shares
		Comments
Pages	1 to many	Page Likes
Comments	1 to many	Comment Likes

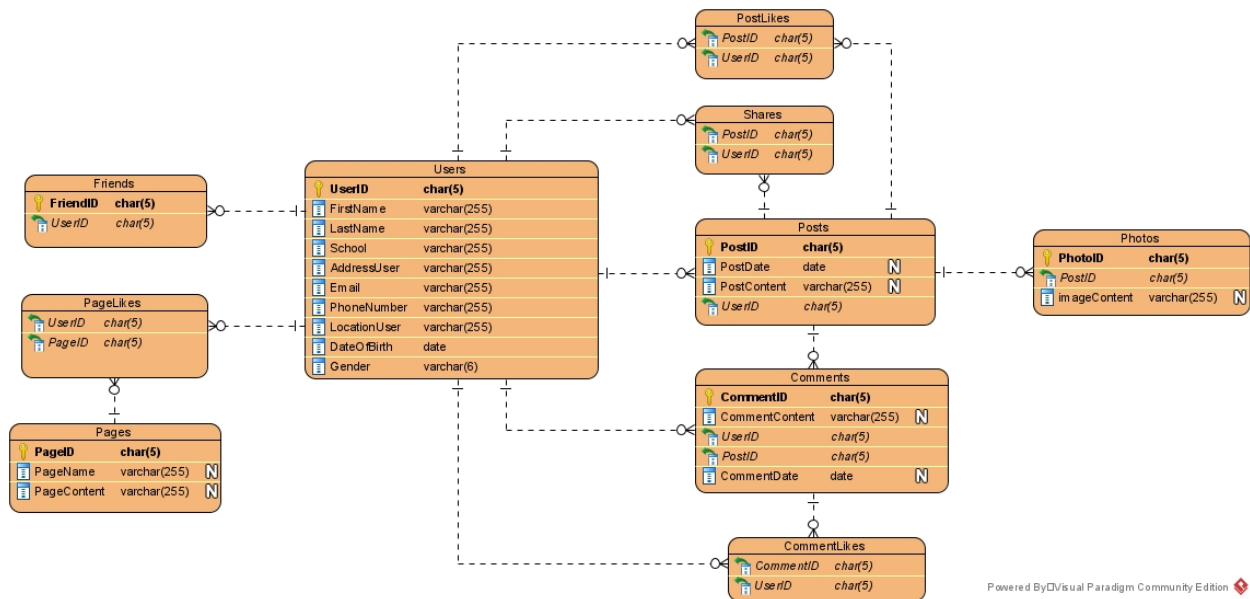
3. For each object, decide its constraint and specify the reason in detail! (.pdf)

- Users
 1. User ID
Constraint: PRIMARY KEY, karena ID harus unique dan tidak boleh NULL
 2. First Name, Last Name, School, Address, Email, PhoneNumber, Location, Date of Birth, Gender
Constraint: NOT NULL, karena semua attribute tersebut diperlukan (wajib ada) dalam pembuatan User baru
- Friends
 1. Friend ID
Constraint: PRIMARY KEY, karena ID harus unique dan tidak boleh NULL
 2. User ID
Constraint: FOREIGN KEY, untuk membentuk relationship dari object Friends (Child) dengan Users (Master)
- Page Likes
 1. User ID
Constraint: FOREIGN KEY, untuk membentuk relationship dari object Page Likes (Child) dengan Users (Master)
 2. Page ID
Constraint: FOREIGN KEY, untuk membentuk relationship dari object Page Likes (Child) dengan Page (Master)
- Pages
 1. Page ID: PRIMARY KEY, karena ID harus unique dan tidak boleh NULL
- Post Likes
 1. User ID
Constraint: FOREIGN KEY, untuk membentuk relationship dari object Post Likes (Child) dengan Users (Master)
 2. Post ID

Constraint: FOREIGN KEY, untuk membentuk relationship dari object Post Likes (Child) dengan Posts (Master)

- Shares
 1. User ID
Constraint: FOREIGN KEY, untuk membentuk relationship dari object Post Likes (Child) dengan Users (Master)
 2. Post ID
Constraint: FOREIGN KEY, untuk membentuk relationship dari object Shares (Child) dengan Posts (Master)
- Posts
 1. Post ID
Constraint: PRIMARY KEY, karena ID harus unique dan tidak boleh NULL
 2. User ID
Constraint: FOREIGN KEY, untuk membentuk relationship dari object Posts (Child) dengan Users (Master)
- Comments
 1. Comment ID
Constraint: PRIMARY KEY, karena ID harus unique dan tidak boleh NULL
 2. User ID
Constraint: FOREIGN KEY, untuk membentuk relationship dari object Comments (Child) dengan Users (Master)
 3. Post ID
Constraint: FOREIGN KEY, untuk membentuk relationship dari object Comments (Child) dengan Posts (Master)
- Comment Likes
 1. User ID
Constraint: FOREIGN KEY, untuk membentuk relationship dari object Comment Likes (Child) dengan Users (Master)
 2. Comment ID
Constraint: FOREIGN KEY, untuk membentuk relationship dari object Comment Likes (Child) dengan Comments (Master)
- Photos
 1. Photo ID
Constraint: PRIMARY KEY, karena ID harus unique dan tidak boleh NULL
 2. Post ID
Constraint: FOREIGN KEY, untuk membentuk relationship dari object Photos (Child) dengan Posts (Master)

4. Draw the above diagram in "ERD format" which includes the data types, primary and foreign key, and relation between objects. Please choose appropriate tools, we recommend using Visual Paradigm.
(.jpeg)



Data Definition Language [50%]

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

Data Integrity adalah konsistensi dan akurasi dari sebuah data selama data itu ada.

Data Integrity terbagi menjadi:

- Entity Integrity

Entity Integrity bergantung pada Primary Key yang mengidentifikasi bagian-bagian dari sebuah data, untuk memastikan agar data yang ada tidak ada yang sama (unique) dan tidak ada Field dalam table yang NULL. Cara menjaga Entity Integrity adalah dengan menggunakan Constraint Primary Key dan Not NULL

- Referential Integrity

Referential Integrity memastikan relationship antar Table. Cara menjaga Referential Integrity adalah dengan menggunakan Constraint Foreign Key

- Domain Integrity

Domain Integrity memastikan semua data dalam database mengikuti aturan yang telah ditetapkan (values, range, dan format). Cara menjaga Domain Integrity adalah dengan menggunakan Constraint Check dan Default











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

- Primary Key

Adalah sebuah column unique dan bukan NULL yang mengidentifikasi seluruh baris yang ada dalam sebuah table. 1 Table hanya dapat memiliki 1 Primary Key.

Contoh:

1. UserID







Users	
 UserID	char(5)
 FirstName	varchar(255)
 LastName	varchar(255)
 School	varchar(255)
 AddressUser	varchar(255)
 Email	varchar(255)
 PhoneNumber	varchar(255)
 LocationUser	varchar(255)
 DateOfBirth	date
 Gender	varchar(6)

- Foreign Key

Adalah column yang berfungsi untuk membentuk relationship antara 2 table. 1 Table dapat memiliki lebih dari 1 Foreign Key (tidak unique).

Contoh:

1. UserID



Posts	
 PostID	char(5)
 PostDate	date 
 PostContent	varchar(255) 
 UserID	char(5)

- Composite Key

Adalah sebuah Primary Key yang terbentuk dari kombinasi dari 2 atau lebih column yang saat keduanya atau semuanya digunakan menjamin keuniquen.

Contoh:

1. OrderID dan ProductID sama-sama merupakan Primary Key dari Orders

Orders	
 OrderID	char(5)
 ProductID	char(5)

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

- BEGIN TRAN: untuk memulai proses transaction dan mengunci table

Contoh:

```
BEGIN TRAN
UPDATE Users
SET UserName = Joko
```

- COMMIT: untuk menyimpan perubahan dilakukan secara permanen dan membuka table

Contoh:

```
BEGIN TRAN
UPDATE Users
SET UserName = Joko
```

COMMIT

Maka perubahan yang dilakukan pada table Users yaitu pada column UserName menjadi permanen

- ROLLBACK: untuk membatalkan perubahan yang dilakukan ke sebelum terjadi perubahan dan membuka table

Contoh:

```
BEGIN TRAN
UPDATE Users
SET UserName = Joko
```

ROLLBACK

Maka perubahan yang dilakukan pada table Users yaitu pada column UserName kembali pada sebelum terjadi perubahan

4. Create all of the tables above according to your answer in the previous section! (.sql)