



ERD

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

Objects :

- Users :
 - UserID
 - FirstName
 - LastName
 - School
 - Address
 - Email
 - PhoneNumber
 - Location
 - DateofBirth
 - Gender
- Posts
 - UserID

- PostID
 - PostContent
 - PostDate
- PostLikes
 - PostID
 - UserID
- Photos
 - PostID
 - PhotoID
 - ImageContent
- Shares
 - PostID
 - UserID
- Comments
 - PostID
 - UserID
 - CommentID
 - CommentDate
 - CommentContent
- CommentLikes
 - CommentID
 - UserID
- Friends
 - UserID
 - FriendID
- Pages
 - PageID
 - PageName
 - PageContent

- PageLikes
 - UserID
 - PageID

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

- Posts(Child) – Users(Master) Many To One
- Users(Child) – Friends(Master) One To Many
- Users(Child) – PageLikes(Master) One To Many
- PageLikes(Child) – Pages(Master) Many To One
- Posts(Child) – PostLikes(Master) One To Many
- Posts(Child) – Photos(Master) One To Many
- Posts(Child) – Shares(Master) One To Many
- Posts(Child) – Comments(Master) One To Many
- Comments(Child) – CommentLikes(Master) One To Many

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

- Users
 - UserID Primary Key karena setiap user mempunyai ID yang berbeda dan tidak boleh duplikat
 - UserID Foreign Key karena me-reference Object lain seperti Posts, Friends, PageLikes
- Posts
 - UserID dan PostID sebagai Primary Key karena setiap user mempunyai ID yang berbeda, setiap user mempunyai post tersendiri

- UserID dan PostID juga digunakan sebagai Foreign Key karena me-reference object lain seperti PostLikes, Photos, Shares yang sesuai dengan PostID serta UserID nya.
- Friends
 - UserID digunakan sebagai Primary Key karena setiap User mempunyai teman yang berbeda beda
- PageLikes
 - UserID digunakan sebagai Primary Key karena setiap User mempunyai PageLike dan Page yang berbeda-beda
 - PageID digunakan sebagai Foreign Key yang me-reference Pages
- Pages
 - PageID digunakan sebagai Primary Key karena setiap Page unik dan berbeda
- PostLikes
 - PostID dan UserID digunakan sebagai Primary Key karena setiap like harus sesuai dengan postnya dan juga sesuai dengan usernya
- Photos
 - PostID digunakan sebagai Primary Key karena setiap Photos harus melihat terlebih dahulu dia diletakkan di post mana
- Shares
 - PostID dan UserID digunakan sebagai Primary Key karena jumlah share harus sesuai dengan postnya dan juga sesuai dengan usernya
- Comments
 - PostID dan UserID digunakan sebagai Primary Key karena Comment dari post harus sesuai dengan postnya dan juga sesuai dengan usernya

- UserID dan CommentID juga digunakan sebagai Foreign Key untuk me-reference CommentLikes. Karena CommentLikes harus sesuai dengan UserID dan juga CommentID.
- CommentLikes
 - CommentID dan UserID digunakan sebagai Primary Key karena jumlah comment dari post harus sesuai dengan postnya dan juga sesuai dengan usernya

DDL

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

Data integrity adalah Jaminan bahwa data yang kita masukkan ke dalam server konsisten dan juga saat ada perubahan data, tidak menghasilkan inkonsistensi / kerusakan

Dalam SQL Server terdapat Integrity Constraint yang mengatur dan menjaga integritas data. Terdapat beberapa cara untuk menjaga integritas data dalam SQL Server

- Entity Integrity Constrain

Setiap entity harus ada pengidentifikasi antara 1 entity dengan yang lainnya (setiap entity unik) dengan adanya Primary Key sehingga tidak ada informasi yang ter-duplikat ataupun membingungkan.
- Domain Integrity Constrain

Setiap data yang masuk harus sesuai dengan apa yang diinginkan dengan menggunakan CHECK dan DEFAULT

Contoh :

- Tanggal lahir harus di CHECK sehingga mengikuti aturan seperti 'dd/mm/yyyy'
- Ketika nilai mahasiswa tidak diinput, bisa di DEFAULT kan 0 untuk mencegah error

- Referential Integrity Constrain

Referential Integrity berfokus pada apa yang terjadi ketika suatu data diubah pada satu table yang terikat / di-reference table lain.

Terdapat 2 hal yang kita harus perhatikan ketika terjadi update pada suatu table Master (yg di-reference) terhadap table Child (yg me-reference):

- ON UPDATE
- ON DELETE

Terdapat 3 pilihan ketika ON UPDATE ataupun ON DELETE yaitu :

- CASCADE
Artinya ketika row di table master ter-delete / ter-update, table child mengikuti value (jika ter-update) ataupun delete row (jika ter-delete)
- SET NULL.
Ketika row di table master ter-delete / ter-update, maka value di table child menjadi NULL
- SET DEFAULT.

Ketika row di table master ter-delete / ter-update, maka value di table child menjadi DEFAULT value dari column tersebut yang telah di set pada waktu pembuatan table.

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

Primary Key adalah key yang membedakan antara 1 entity dengan entity lain dalam satu table.

Contoh :

NIM (PRIMARY KEY)	Nama Siswa
2401955132	Aradhana Indra Daniswara
2401955434	Budi
2401955555	Toni
2401427782	Dodi

Foreign Key digunakan untuk menghubungkan antara 1 tabel dengan table yang lain. Foreign Key akan me-reference Primary Key dari table lain

Contoh :

NIM (PRIMARY KEY & FOREIGN KEY)	Nama Siswa
2401955132	Aradhana Indra Daniswara
2401955434	Budi
2401955555	Toni
2401427782	Dodi

NIM (PRIMARY KEY)	Jurusan
2401955132	Computer Science
2401955434	Food Technology
2401955555	Information System
2401427782	Cyber Security

NIM di table atas menggunakan NIM sebagai Foreign Key yang me-reference table yang bawah yang menggunakan NIM sebagai Primary Key nya.

Composite Key berisi 2 atau lebih Primary Key yang digunakan untuk membedakan setiap entity secara unik ketika satu Primary Key saja tidak cukup.

Contoh

NIM	ClassID
2401955132	LL01
2401955132	LM01
2401955434	LX01
2401955434	LL01
2401955555	LL01
2401427782	LS01

Dalam hal ini, kita tidak bisa menggunakan NIM saja sebagai Primary Key karena 1 NIM bisa terdapat dalam 2 Class dan 1 Class bisa terdapat 2 NIM yang berbeda. Maka dari itu NIM + ClassID digunakan sebagai Composite Key.

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

BEGIN TRAN bisa diibaratkan sebagai Checkpoint seperti di Game. Ketika BEGIN TRAN dijalankan, segala efek / command yang dilakukan terhadap database tersebut beserta table table nya akan disimpan sementara. ROLLBACK akan meng-UNDO command / efek apapun yang dilakukan setelah BEGIN TRAN. Sedangkan Commit akan meng-SAVE command / efek apapun yang dilakukan setelah BEGIN TRAN.