**THEORY EXAMINATION**

1. ERD
2. Berikut list table dan atributnya:
3. Users
4. User ID
5. First Name
6. Last Name
7. School
8. Address
9. Email
10. Phone Number
11. Location
12. Date of Birth
13. Gender
14. Friends
15. Friend ID
16. User ID
17. Pages
18. Page ID
19. Page Name
20. Page Content
21. Page Likes
22. Page ID
23. User ID
24. Posts
25. Post ID
26. Post Content
27. Post Date
28. User ID
29. Post Likes
30. Post ID
31. User ID
32. Photos
33. Photo ID
34. Image Content
35. Post ID
36. Shares
37. Post ID
38. User ID
39. Comments
40. Comment ID
41. Comment Date
42. Comment Content
43. Post ID
44. User ID
45. Comment Likes
46. Comment ID
47. User ID
48. Format “master-child”.
49. Users-Friends
50. Users-Page Likes
51. Users-Posts
52. Pages-Page Likes
53. Posts-Post Likes
54. Posts-Photos
55. Posts-Shares
56. Posts-Comments
57. Comments-Comment Likes
58. List table
59. Users
60. User ID (PK, char(5), not null)

User ID menjadi PK karena harus unik dan sebagai pengenal dari table Users. Char panjang 5 karena User ID berbentuk ‘USXXX’. Not null karena PK tidak boleh kosong.

1. First Name (varchar, not null)

Berbentuk varchar karena nama depan dari seseorang tidak bisa ditebak panjangnya (tidak pasti). Not null karena setiap orang pasti memiliki nama, setidaknya nama depan.

1. Last Name (varchar, nullable)

Berbentuk varchar karena nama belakang dari seseorang tidak bisa ditebak panjangnya (tidak pasti). Nullable karena bisa saja namanya hanya terdiri dari satu kata.

1. School (varchar, not null)

Berbentuk varchar karena nama sekolah seseorang tidak bisa ditebak panjangnya (tidak pasti). Not null karena jika kita lihat, diagram ini merupakan study network diagram yang berhubungan dengan sekolah si user.

1. Address (varchar, nullable)

Berbentuk varchar karena alamat seseorang tidak bisa ditebak panjangnya (tidak pasti). Nullable karena untuk menghargai privasi masing-masing user sehingga tidak harus diisi.

1. Email (varchar, not null, check like ‘%@%’)

Berbentuk varchar karena email seseorang tidak bisa ditebak panjangnya (tidak pasti). Not null karena email dibutuhkan sebagai *contact person* si user. Kemudian terdapat check untuk memastikan bahwa yang diinput oleh user merupakan sebuah email.

1. Phone Number (varchar, not null, check like ‘+62%’)

Berbentuk varchar karena nomor telepon seseorang tidak bisa ditebak panjangnya (tidak pasti). Not null karena nomor telepon diperlukan juga untuk mengontak si user selain dari email. Kemudian terdapat check untuk memastikan bahwa yang diinput oleh user merupakan sebuah nomor telepon dari Indonesia.

1. Location (varchar, nullable)

Berbentuk varchar karena lokasi seseorang tidak bisa ditebak panjangnya (tidak pasti). Nullable karena untuk menghargai privasi si user. Jadi, tidak diharuskan untuk mengisi lokasi.

1. Date of Birth (date, not null, year <= 2003)

Berbentuk date karena berisi sebuah tanggal lahir. Year <= 2003 karena harus 18 tahun ke atas. Not null karena untuk memastikan umurnya di atas 18 tahun.

1. Gender (varchar, not null, check gender like ‘Male’ or ‘Female’)

Berbentuk varchar karena panjang karakter jenis kelamin berbeda. Not null karena dibutuhkan untuk mengetahui jenis kelaminnya. Kemudian terdapat check untuk memastikan input dari user merupakan ‘Male’ atau ‘Female’.

1. Friends
2. Friend ID (PK, char(5), not null)

Friend ID menjadi PK karena dia harus unique dan dapat menjadi elemen pengenal dari table Friends. Char(5) karena bentuk ID nya ‘FRXXX’. Not null karena sebuah Primary Key tidak boleh NULL.

1. User ID (FK dari Users)

User ID menjadi FK karena atribut ini diambil dari table Users.

1. Pages
2. Page ID (PK, char(5), not null, check Page ID like ‘PG%’)

Page ID menjadi PK karena dia harus unique dan dapat menjadi elemen pengenal dari table Pages. Char(5) karena bentuk ID nya ‘PGXXX’. Not null karena sebuah Primary Key tidak boleh NULL.

1. Page Name (varchar, not null, unique)

Berbentuk varchar karena nama dari sebuah halaman tidak bisa ditebak panjangnya (tidak pasti). Not null karena sebuah halaman harus mempunyai nama sebagai tanda pengenal. Unique karena sebuah page tidak boleh sama namanya.

1. Page Content (varchar, not null, unique)

Berbentuk varchar karena konten di dalam sebuah halaman tidak bisa ditebak panjangnya (tidak pasti). Not null karena sebuah halaman tidak mungkin kosong (NULL). Unique karena sebuah konten di dalam sebuah halaman pasti berbeda.

1. Page Likes
2. Page ID (PK, FK dari Pages)

Page ID menjadi FK karena atribut ini diambil dari table Pages dan atribut ini dibutuhkan karena setiap page, tak terkecuali Page Likes pasti memiliki Page ID.

1. User ID (FK dari Users)

User ID menjadi FK karena atribut ini diambil dari table Users dan atribut ini dibutuhkan karena setiap page likes pasti dimiliki oleh seorang user.

1. Posts
2. Post ID (PK, char(5), not null)

Post ID menjadi PK karena dia harus unique dan dapat menjadi elemen pengenal dari table Posts. Char(5) karena bentuk ID nya ‘PSXXX’. Not null karena sebuah Primary Key tidak boleh NULL.

1. Post Content (varchar, not null, unique)

Berbentuk varchar karena nama isi konten dari sebuah post tidak bisa ditebak panjangnya (tidak pasti). Not null karena sebuah konten tidak mungkin kosong. Unique karena konten dari sebuah post tidak boleh sama, kalo sama plagiarism.

1. Post Date (date, not null)

Berbentuk date untuk menunjukkan tanggal post tersebut diunggah. Not null karena sebuah post pasti memiliki tanggal unggahannya kapan.

1. User ID (FK dari Users)

FK karena User ID diambil dari table Users.

1. Post Likes
2. Post Likes ID (PK, char(5), not null)

Post Likes ID ada sebagai pengenal dan PK untuk table Post Likes karena Post ID dan User ID bisa ter-duplicate jika satu user melakukan like sebanyak lebih dari satu kali pada post yang berbeda ataupun sebaliknya. Char (5) karena berbentuk ‘PLXXX’.

1. Post ID (FK dari Posts)

FK karena Post ID diambil dari table Users.

1. User ID (FK dari Users)

FK karena User ID diambil dari table Users.

1. Photos
2. Photo ID (PK, char(5), not null, check Photo ID like ‘PH%’)

Photo ID menjadi PK karena dia harus unique dan dapat menjadi elemen pengenal dari table Photos. Char(5) karena bentuk ID nya ‘PHXXX’. Not null karena sebuah Primary Key tidak boleh NULL.

1. Image Content (BLOB, nullable)

Image Content berbentuk BLOB karena menyimpan data foto. Nullable karena dalam setiap posts tidak diharuskan untuk mencantumkan foto di dalamnya.

1. Post ID (FK dari Posts)

FK karena diambil dari table Posts. Karena photos ini ada di dalam sebuah posts, maka Post ID dibutuhkan sebagai sekaligus penghubung antara table Photos dan Posts.

1. Shares
2. Share ID (PK, char(5), not null)

Share ID ada sebagai pengenal dan PK untuk table shares karena Post ID dan User ID bisa ter-duplicate jika satu user melakukan share sebanyak lebih dari satu kali, baik pada post yang sama atau berbeda. Char (5) karena berbentuk ‘SHXXX’.

1. Post ID (FK dari Posts)

FK karena diambil dari table Posts. Karena shares ini berhubungan dengan sebuah posts, maka Post ID dibutuhkan sekaligus sebagai penghubung antara table Shares dan Posts.

1. User ID (FK dari User)

FK karena diambil dari table User. Karena Shares ini pasti dilakukan oleh seorang user, maka User ID dibutuhkan sekaligus sebagai penghubung antara table Photos dan Posts.

1. Comments
2. Comment ID (PK, char(5), not null)

Photo ID menjadi PK karena dia harus unique dan dapat menjadi elemen pengenal dari table Photos. Char(5) karena bentuk ID nya ‘CMXXX’. Not null karena sebuah Primary Key tidak boleh NULL.

1. Comment Date (date, not null)

Berbentuk date karena menampung tanggal. Not null karena tanggal dari sebuah komen dibutuhkan untuk keperluan data/history.

1. Comment Content (varchar, not null)

Berbentuk varchar karena nama isi konten dari sebuah comment tidak bisa ditebak panjangnya (tidak pasti). Not null karena dalam sebuah komen tidak boleh kosong isinya.

1. User ID (FK dari Users)

FK karena diambil dari table User. Karena Comment ini pasti dilakukan oleh seorang user, maka User ID dibutuhkan sekaligus sebagai penghubung antara table Photos dan Posts.

1. Comment Likes
2. Comment Like ID (PK, char(5), not null)

Comment Like ID ada sebagai pengenal dan PK untuk table Comment Likes karena Comment ID dan User ID bisa ter-duplicate jika satu user melakukan like ke berbagai comment dan sebaliknya. Char (5) karena berbentuk ‘CLXXX’.

1. Comment ID (FK dari Comments)

Comment ID dibutuhkan karena sebuah like membutuhkan penanda yang menunjuk pada sebuah comment.

1. User ID (FK dari Users)

User ID dibutuhkan karena sebuah like pasti muncul dari sebuah User.