

DB Exam 1

2440006122 – Bryan Tamin

Entity Relationship Diagram [50%]

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

Users:

FK/PK	Column Name	Data Type
PK	ID	CHAR(10)
	FirstName	VARCHAR(255)
	LastName	VARCHAR(255)
	School	VARCHAR(255)
	Address	VARCHAR(255)
	Email	VARCHAR(255)
	PhoneNumber	VARCHAR(255)
	Location	VARCHAR(15)
	DateOfBirth	DATE
	Gender	CHAR(15)

Posts:

FK/PK	Column Name	Data Type
PK	ID	CHAR(10)
FK	UsersID	VARCHAR(255)
	Date	VARCHAR(255)
	Content	VARCHAR(255)

Friends:

FK/PK	Column Name	Data Type
PK	ID	CHAR(10)
FK	UsersID	VARCHAR(255)

Page:

FK/PK	Column Name	Data Type
PK	ID	CHAR(10)
	Name	VARCHAR(255)
	Content	VARCHAR(255)

PageLikes:

FK/PK	Column Name	Data Type
PK,FK	UsersID	CHAR(10)
PK,FK	PagesID	CHAR(10)

PostLikes:

FK/PK	Column Name	Data Type
PK,FK	UsersID	CHAR(10)
PK,FK	PostsID	CHAR(10)

Photos:

FK/PK	Column Name	Data Type
PK	ID	CHAR(10)
FK	PostsID	CHAR(10)
	ImageContent	VARBINARY(MAX)

Shares:

FK/PK	Column Name	Data Type
PK,FK	UsersID	CHAR(10)
PK,FK	PostsID	CHAR(10)

Comments:

FK/PK	Column Name	Data Type
PK	ID	CHAR(10)
FK	UsersID	CHAR(10)
FK	PostsID	CHAR(10)
	Date	DATE
	Content	VARCHAR(255)

CommentLikes:

FK/PK	Column Name	Data Type
PK,FK	CommentsID	CHAR(10)
PK,FK	UsersID	CHAR(10)

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

Master	Child	Relation
Users	Posts	One to Many
Users	Comments	One to Many
Users	PostLikes	One to Many
Users	CommentLikes	One to Many
Users	Shares	One to Many
Users	Friends	One to Many
Users	PageLikes	One to Many
Posts	Comments	One to Many
Posts	PostLikes	One to Many
Posts	CommentLikes	One to Many
Posts	Shares	One to Many
Pages	PageLikes	One to Many
Comments	CommentLikes	One to Many

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

Users:

Column Name	Constraint	Reason
ID	PK	Each Data must have PK
FirstName	NOT NULL	Crucial Data
Email	NOT NULL	Crucial Data
DateOfBirth	NOT NULL	Crucial Data
Gender	NOT NULL	Crucial Data
Gender	LIKE 'M' OR 'F' OR 'N'	M for male, F for female and N for None(Not specified)

Posts:

Column Name	Constraint	Reason
ID	PK	Each Data must have PK
UsersID	FK	Posts must be created by users
Date	NOT NULL	Auto Generated

Friends:

Column Name	Constraint	Reason
ID	PK	Each Data must have PK
UsersID	FK	Friends must be owned by users

Pages:

Column Name	Constraint	Reason
ID	PK	Each Data must have PK
Name	NOT NULL	Pages must have name

PageLikes:

Column Name	Constraint	Reason
UsersID	PK,FK	Composite keys for bridging many to many, one user can like many pages
PageID	PK,FK	Composite keys for bridging many to many, one page can be liked by many users

PostsLikes:

Column Name	Constraint	Reason
UsersID	PK,FK	Composite keys for bridging many to many, one user can like many posts
PostsID	PK,FK	Composite keys for bridging many to many, one post can be liked by many users

Photos:

Column Name	Constraint	Reason
ID	PK	Each Data must have PK
PostsID	FK	Photos is owned by Posts
ImageContent	NOT NULL	Cannot create Photos without Imagecontent

Shares:

Column Name	Constraint	Reason
UsersID	PK,FK	Composite keys for bridging many to many
PostsID	PK,FK	Composite keys for bridging many to many

Comments:

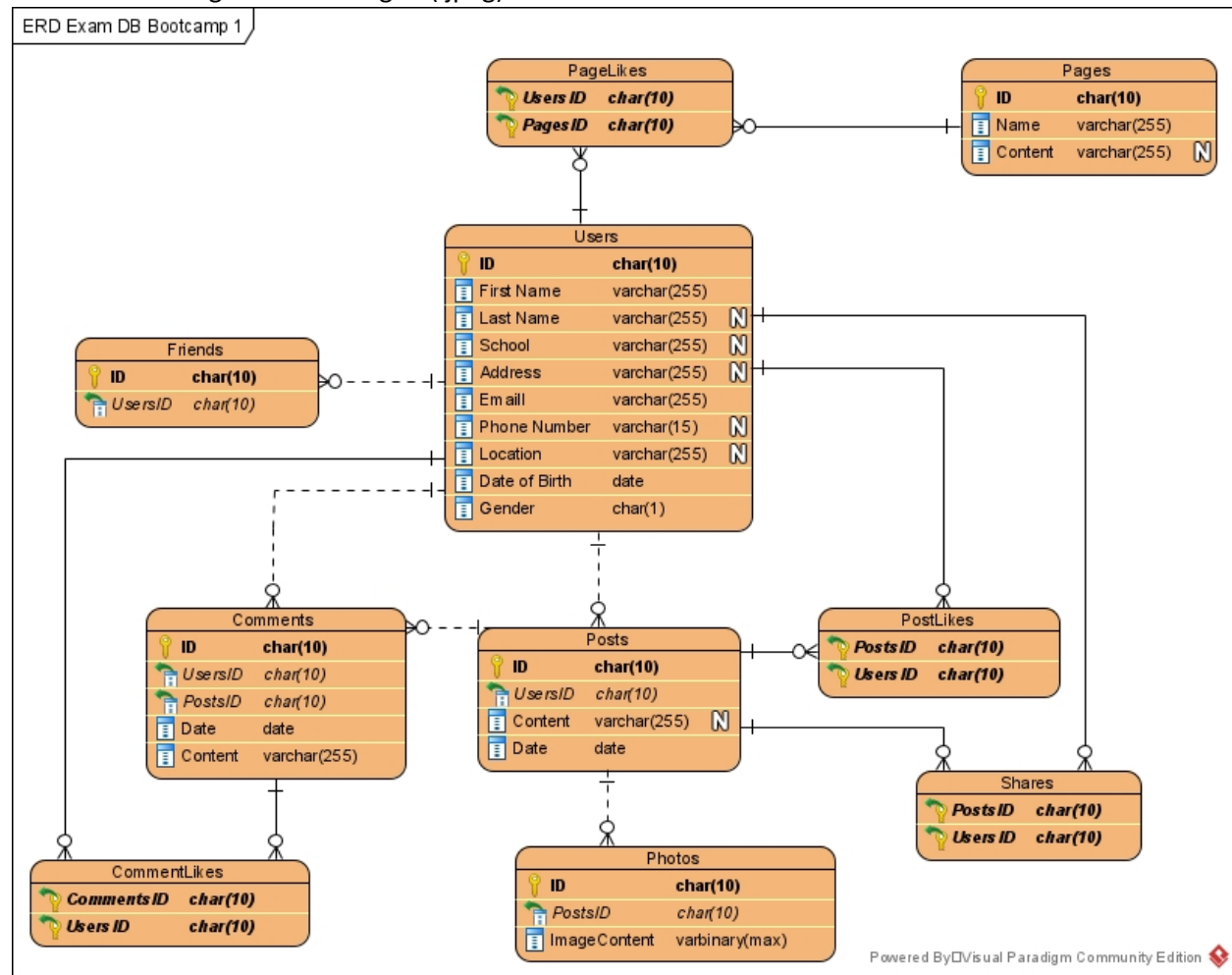
Column Name	Constraint	Reason
ID	PK	Each Data must have PK
UsersID	FK	Comments is created by Users
PostsID	FK	Comments is on the posts
Date	DATE	Auto Generated by system
Content	NOT NULL	Comment cannot be NULL / empty

CommentLikes:

Column Name	Constraint	Reason
CommentsID	PK,FK	Composite keys for bridging many to many, one comment can be liked by many users
UsersID	PK,FK	Composite keys for bridging many to many, one user can like many comments

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 is the maintenance of data accuracy and consistency, also refers to completeness, consistency, and accuracy of data, in other words it refers to the trustworthiness of the data.

Why it is important? Imagine using inaccurate and untrustworthy data for important project, it will impact the project significantly, it can even destroy the entire project.

In SQL server Data integrity is used to maintain consistency and accuracy of the data, we can maintain it in SQL by using Entity integrity, Referential Integrity and Domain integrity.

in Entity Integrity we can use Primary key, Unique key, and not null, declaring a column as primary key will make it have unique and not null.

in Referential Integrity we must make sure to check the validity of our foreign key, the data type of foreign key in the master and child table must be the same.

in Domain integrity we use check and Default for checking our data.

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

Each table must have a primary key, and primary key have the properties of must be unique and NOT NULL, Primary Key used to represent the data/row so we can search or call it. Foreign Key is used to connect two objects, by using other object primary key as their foreign key, it will create a relation between the two objects. Composite key is a combination of two or more columns to create a Primary key.

Example:

Primary key: UserID → because user id is unique to represent each person

foreign key → Staff object have UserID in it because staff is a user too.

composite key: in the transaction details there is itemID and usersID it is created to define the unique transaction each user did, if only stand alone ItemID or usersID will not create a unique transaction because user can buy many item and item can be bought by many users.

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

BEGIN TRAN is used to create a checkpoint in SQL, so if we want to test out something while we are not sure if it will destroy our data, we use BEGIN TRAN.

COMMIT is used after BEGIN TRAN have been used, it is used for save the changes after the checkpoint that we have made to the data.

ROLLBACK is used after BEGIN TRAN have been used, it is used to remove the changes after the checkpoint that we have made to the data, so the data will revert to when it is on checkpoint.

Example:

BEGIN TRAN & ROLLBACK

if we want to alter our data, and we forgot to write WHERE so it changes all of our data, if we use BEGIN TRAN first then we can ROLLBACK so our data will revert to the last checkpoint.

BEGIN TRAN

UPDATE Users
set Gender = 'M'

ROLLBACK

BEGIN TRAN & COMMIT

if we are not sure of the changes that will be made, we first have to use BEGIN TRAN and after we sure of the changes we can use COMMIT to save the changes.

```
BEGIN TRAN
```

```
UPDATE Users
```

```
set Gender = 'M'
```

```
WHERE Name LIKE 'Dono'
```

```
ROLLBACK
```

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