CSE 3330-002

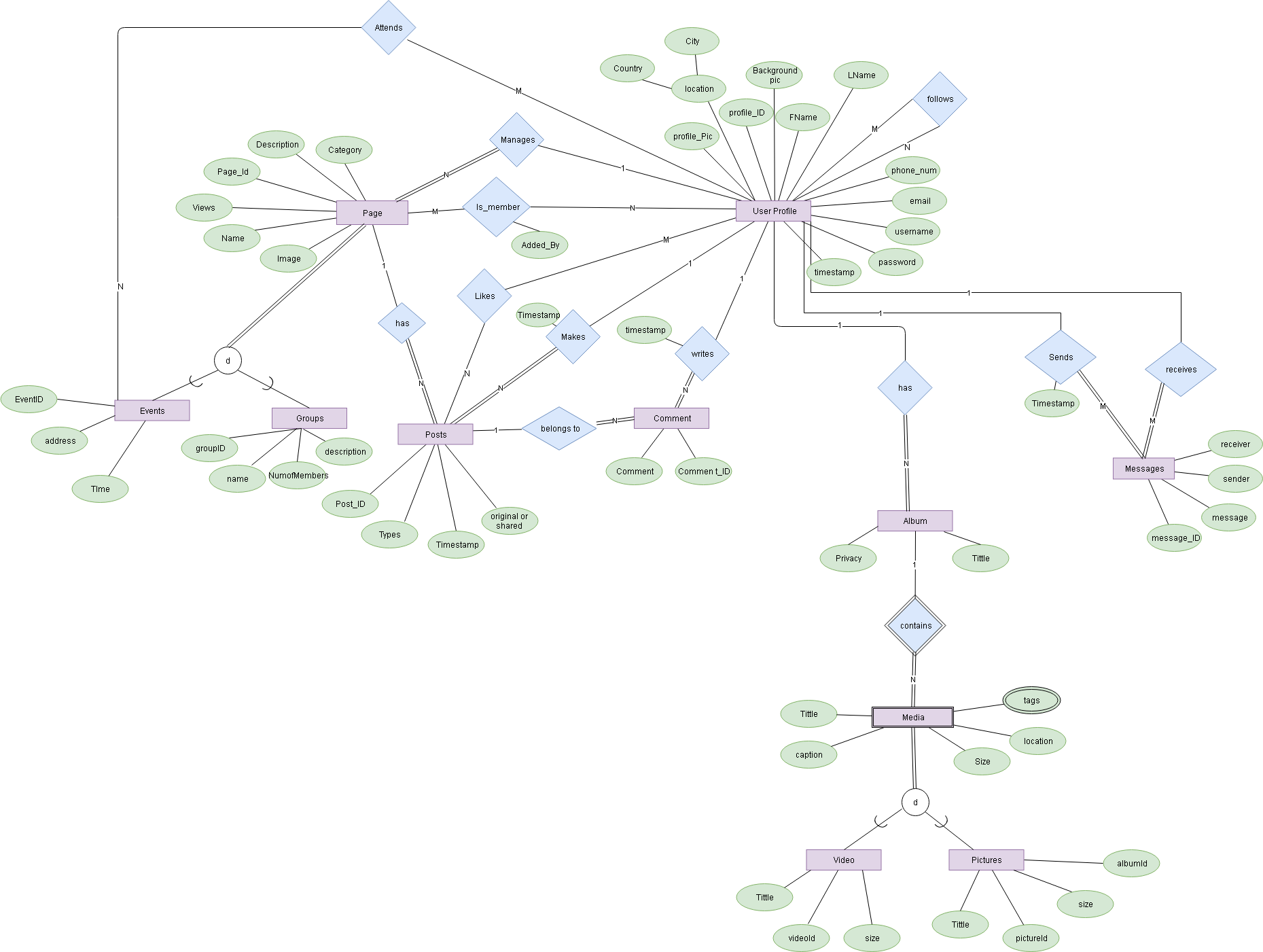
Project

Parts 1 & 2

Team 3: Anna Cox, Hedges Omordia, Diane Calderon

Part 1

EER schema diagram



Design Choices

* A page is split into two types of entities, a group, and an event. We decided to have this distinction because a profile can have additional relationships with an event than with a group and there are also additional attributes that an event can have.
* There are two relationships a profile can have with an event, being a member, which means that they were invited to the event, and attends, meaning that they are going to the event
* A profile can have two types of relationship with a message, either the sender or the receiver, but for each message, a profile cannot be both the sender and receiver
* There has to be at least one profile to one page.

Missing or incomplete requirements

* It is assumed that all profiles are active within a page, so there is nothing explicitly stating that the profile is active.

Assumptions

* Every page will have at least one profile, but can have many.
* Each message will only have one receiver, and one sender. There are no group messages
* Each post and comment has only one author.
* Comments are allowed, but you cannot comment a comment.
* An event must have at least one profile but can be shared with other profiles.
* Profiles contain albums of pictures
* A page can be managed by only one profile
* There are two types of media, videos and pictures
* Profiles can follow other profiles
* A Page can be specialized into groups and events
* Profiles can attend events
* Profiles must be added to a page by creator
* A post can be shared, meaning that a profile, who is not the original creator, can post a copy of the post
* We allow as many users to join our website. Each user is identified by Profile\_id
* Profiles can write(comment) on other pages
* Both creator and member profile can upload media. Each media file can be differentiated by attributes of its set. Creator has privilege to delete, and update files. Media entity set’s primary key is media\_id

Part 2

Specifying the schema

CREATE TABLE GROUPS (

group\_id int PRIMARY KEY,

name char(30) NOT NULL,

description char(100),

category char(30) NOT NULL,

views int,

image image?,

);

CREATE TABLE EVENTS (

event\_id int PRIMARY KEY,

name char(30) NOT NULL,

description char(100),

category char(30) NOT NULL,

views int,

image image?,

Location varchar(50),

event\_date date,

);

CREATE TABLE PROFILES (

profile\_id int PRIMARY KEY,

Fname char(30) NOT NULL,

Lname char(30),

phone\_num int,

email char(50) NOT NULL,

username char(30) NOT NULL,

password char(30) NOT NULL,

timestamp DATETIME,

country varchar(30),

city varchar(30),

profile\_photo int,

cover\_photo,

FOREIGN KEY profile\_photo REFERENCES MEDIA(media\_id),

FOREIGN KEY cover\_photo REFERENCES MEDIA(media\_id)

);

CREATE TABLE COMMENTS (

comment\_id int PRIMARY KEY,

comment char(150),

Timestamp DATETIME,

profile\_id int NOT NULL,

post \_id int NOT NULL,

FOREIGN KEY profile\_id REFERENCES Profile(profile\_id),

FOREIGN KEY post\_id REFERENCES Post(Post\_id),

);

CREATE TABLE MEMBERS (

group\_id int NOT NULL,

profile\_id int NOT NULL,

added\_by int,

admin boolean NOT NULL,

PRIMARY KEY (page\_id, profile\_id),

FOREIGN KEY group\_id REFERENCES GROUPS(group\_id) ON DELETE CASCADE,

FOREIGN KEY profile\_id REFERENCES PROFILES(profile\_id) ON DELETE CASCADE,

FOREIGN KEY added\_by REFERENCES PROFILES(profile\_id)

);

CREATE TABLE POSTS (

post\_id int PRIMARY KEY,

timestamp DATETIME NOT NULL,

original BOOLEAN,

profile\_id NOT NULL,

type NOT NULL,

FOREIGN KEY profile\_id REFERENCES PROFILES(profile\_id),

);

CREATE TABLE MESSAGES (

message\_id int PRIMARY KEY,

timestamp DATETIME NOT NULL,

sender int NOT NULL,

Receiver int NOT NULL,

Message VARCHAR(256),

FOREIGN KEY sender REFERENCES PROFILES(profile\_id),

FOREIGN KEY receiver REFERENCES PROFILES(profile\_id)

);

CREATE TABLE INVITED (

event\_id int NOT NULL,

profile\_id int NOT NULL,

Invited\_by int,

admin boolean NOT NULL,

PRIMARY KEY (event\_id, profile\_id),

FOREIGN KEY event\_id REFERENCES EVENTS(event\_id),

FOREIGN KEY profile\_id REFERENCES PROFILES(profile\_id),

FOREIGN KEY invited\_by REFERENCES PROFILES(profile\_id)

);

CREATE TABLE LIKES(

post\_id int NOT NULL,

profile\_id int NOT NULL,

PRIMARY KEY (post\_id, profile\_id),

FOREIGN KEY post\_id REFERENCES POSTS(post\_id),

FOREIGN KEY profile\_id REFERENCES PROFILES(profile\_id),

);

CREATE TABLE EVENT\_POSTS (

event\_id int NOT NULL,

post\_id int NOT NULL,

PRIMARY KEY (event\_id, post\_id),

FOREIGN KEY event\_id REFERENCES EVENTS(event\_id) ON DELETE CASCADE,

FOREIGN KEY post\_id REFERENCES POSTS(post\_id),

);

CREATE TABLE GROUP\_POSTS (

group\_id int NOT NULL,

post\_id int NOT NULL,

PRIMARY KEY (event\_id, post\_id),

FOREIGN KEY event\_id REFERENCES GROUPS(group\_id) ON DELETE CASCADE,

FOREIGN KEY post\_id REFERENCES POSTS(post\_id) ,

);

CREATE TABLE ALBUMS (

title varchar(40) NOT NULL,

profile\_id int NOT NULL,

album\_id int PRIMARY KEY,

privacy varchar(15),

FOREIGN KEY profile\_id REFERENCES PROFILES(profile\_id) ON DELETE CASCADE,

);

CREATE TABLE MEDIA (

media\_id int PRIMARY KEY,

album\_id int NOT NULL,

type varchar(20),

caption varchar(256),

size int NOT NULL,

location varchar(30),

FOREIGN KEY album\_id REFERENCES ALBUMS(album\_id)

);

CREATE TABLE MEDIA\_TAGS (

media\_id int NOT NULL,

tag varchar(30) NOT NULL,

PRIMARY KEY (media\_id, tag),

FOREIGN KEY media\_id REFERENCES MEDIA(media\_id) ON DELETE CASCADE

);

CREATE TABLE EVENT\_ATTENDEES (

event\_id int NOT NULL,

attendee int NOT NULL,

PRIMARY KEY (event\_id, attendee),

FOREIGN KEY event\_id REFERENCES EVENTS(event\_id) ON DELETE CASCADE,

FOREIGN KEY attendee REFERENCES PROFILES(profile\_id) ON DELETE CASCADE

);

CREATE TABLE FOLLOWERS (

follower int NOT NULL,

following int NOT NULL,

PRIMARY KEY (follower, following),

FOREIGN KEY follower REFERENCES PROFILES(profile\_id) ON DELETE CASCADE,

FOREIGN KEY following REFERENCES PROFILES(profile\_id) ON DELETE CASCADE

);

Mapping Choices

* For mapping the specialization of PAGE into EVENTS and GROUPS, it was chosen that each subclass has its own entity, because profiles have an additional relationship with EVENTS, which has its own entity, and references EVENTS, but not groups. It was not necessary to have PAGE with its own entity.
* The sends and receives relationship between MESSAGE and PROFILE can be represented by only one relation, MESSAGE, indicating both the sender and receiver

Schemas

