

COURSE: COMP 5531 Files and Databases

ASSIGNMENT: Assignment 1

LECTURE SECTION: NN

PROFESSOR: Dr. bipin c. DESAI

LAB INSTRUCTOR : yogesh YADAV

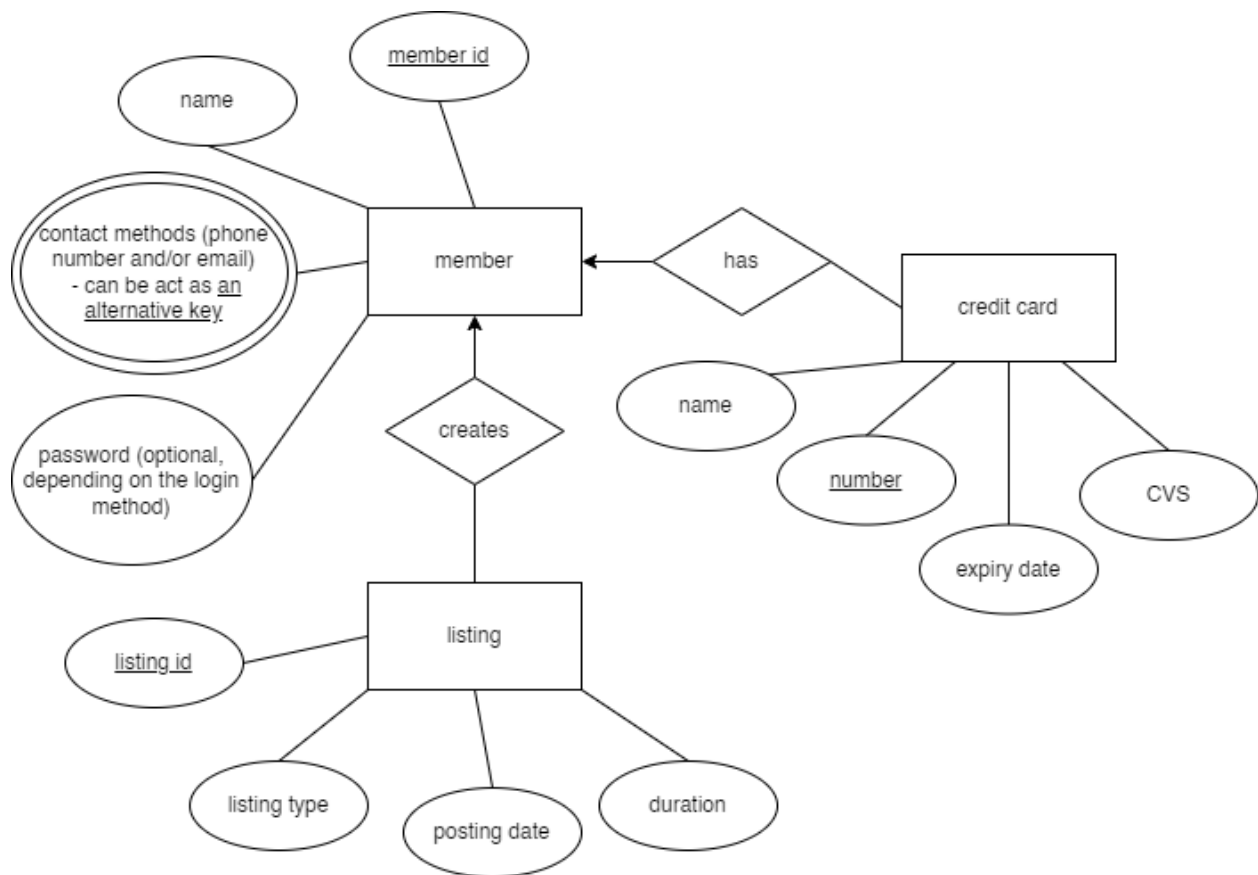
DUE: February 11, 2022

GROUP INFORMATION

Group Name: COMP 5531_group_3

Group Leader: fung sim, LEUNG, 40195538, f_eung@encs.concordia.ca
victor, NWATU, 40206992, v_nwatu@encs.concordia.ca
akshat, TYAGI, 40195161, ak_tyag@encs.concordia.ca
rim, EL OSTA, 40083858, r_elosta@encs.concordia.ca

Q1 first part:



We added the member id attribute to the member entity because the name attribute is not unique. Similarly, we added the listing id attribute to the listing entity because the other attributes mentioned in the question are not unique.

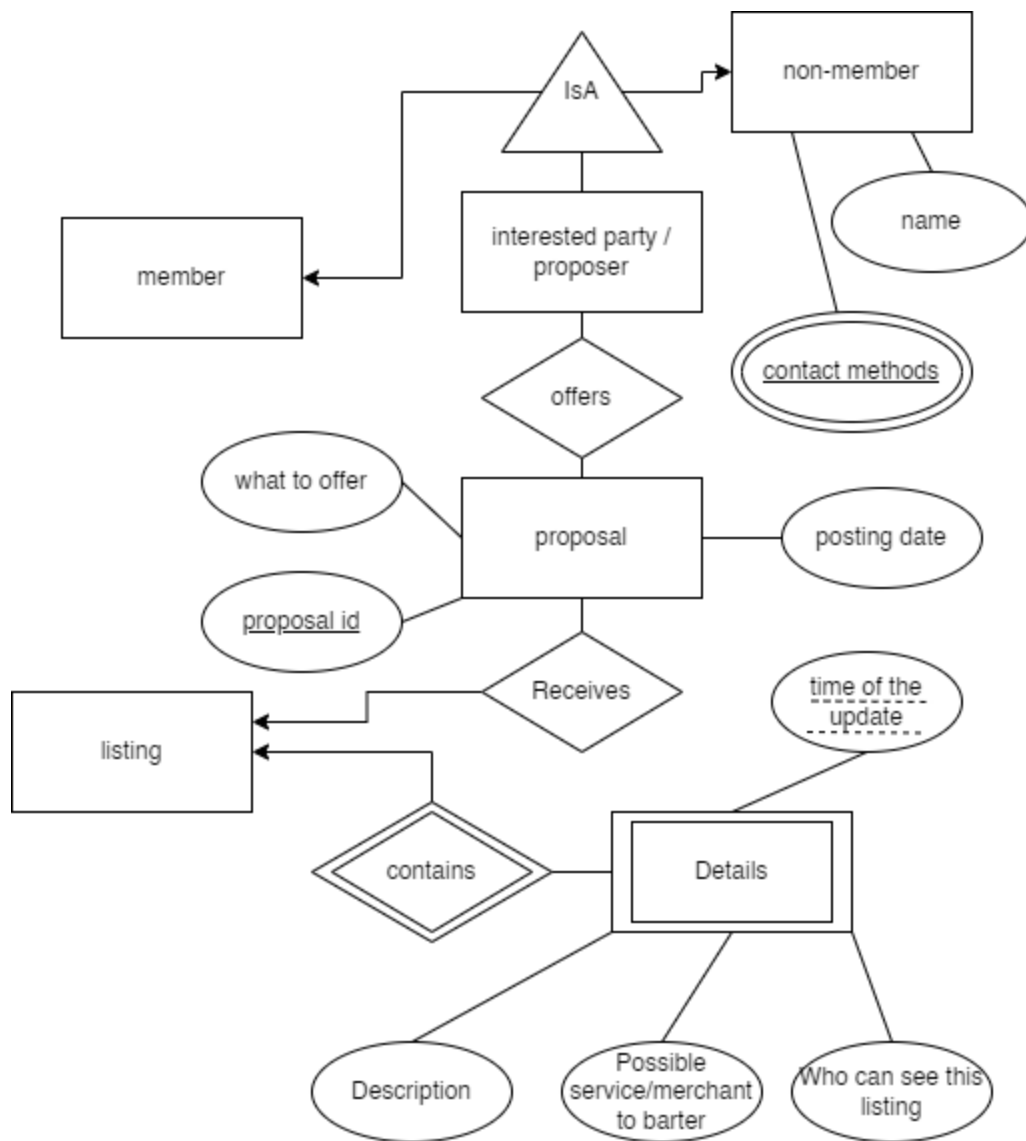
For the credit card entity, We have chosen the number attribute as its primary key since usually a database just stores the latest version of the same credit card. However, if the system records all the historical versions of the card, the primary key will be a combination of the number, expiry date and CVS attributes.

Although the question does not mention the contact method, we added it to the ER diagram as it makes sense the system needs to notify the member about their listings, or use it as login credentials. We designed this attribute as a multivalued attribute as a member may provide both their phone number and email address. A password attribute may not be needed if the system sends emails or texts to verify the login. The contact method can be an alternative key if each member has to leave a unique contact method (e.g. a mobile phone number or a personal email address).

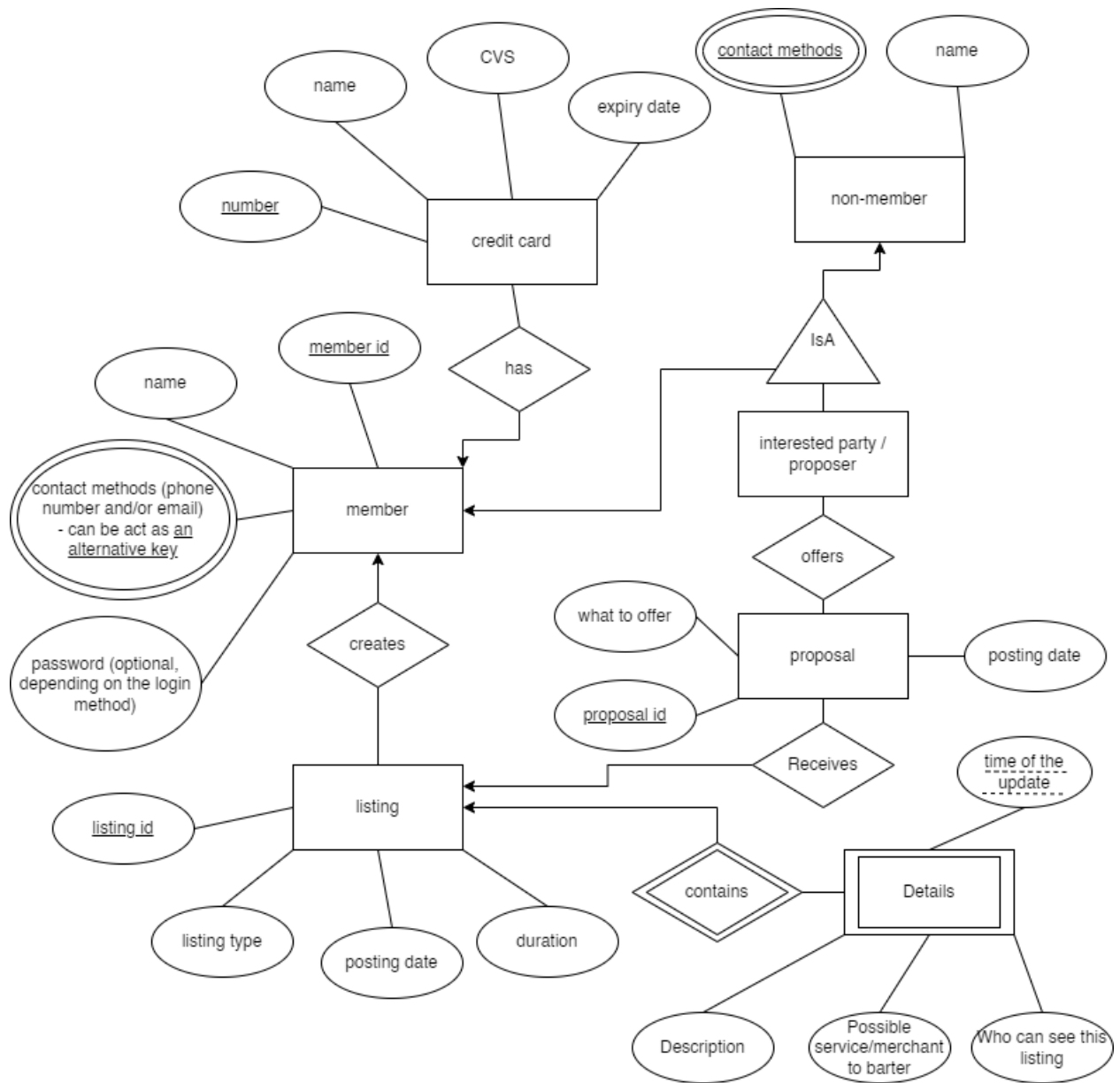
Question 1 second part:

We created the entity set Description as a weak entity as it does not have a unique key on its own and has a total participation constraint. We added the attribute time of the update as a discriminator. The database can, therefore, record the editing history of the description, and identify the exact version of a description by using a posting id from the listing entity set and a time of the update (forming a primary key). As a listing can have more than one version of a description, it is a one-to-many relationship.

A proposal entity was created and the attribute Proposal ID is its primary key. We used an IsA relationship to specify whether an interested party is a member or a non-member. We assign two attributes to the non-member entity set as it is reasonable that a member who offers a service or article for barter needs to contact the proposer. The contact method will be a primary key. A person can offer more than one proposal, so the “interested party offers proposal” relationship is one-to-many. A listing can receive any number of proposals, but each proposal is for a specific listing. Hence, the “listing receives proposal” relationship is one-to-many.



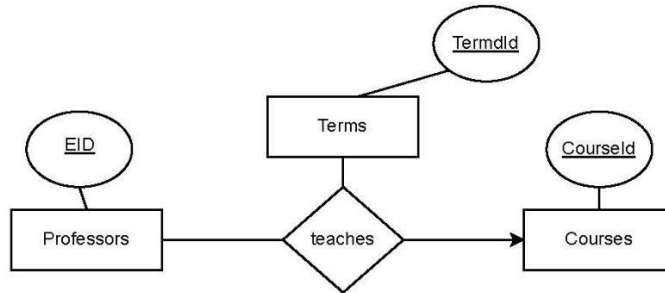
The complete ER diagram:



Question 2:

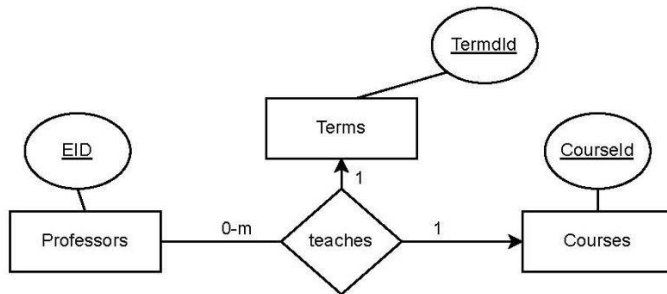
Professors can teach the same course in several terms, and each offering must be recorded. A Terms Entity has been added with TermId as the primary key to record each offering of the course

2A



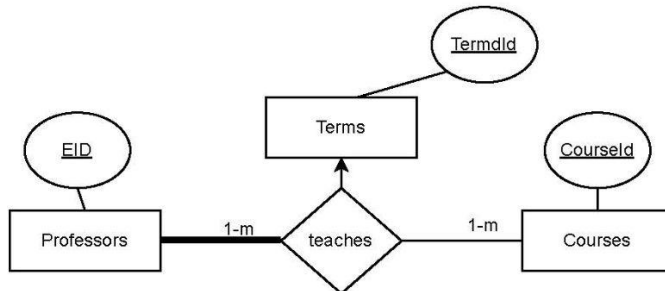
Professors can teach the same course in several term, and only the most recent such offering needs to be recorded. Since only the most recent offering can be recorded, there is a many to one relationship between professor and Terms (this would be the assumption for the rest of the entity diagrams).

2B



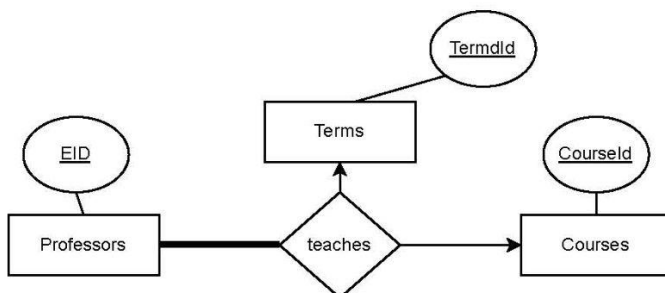
There is total participation between the professor and courses as every professor must teach at least one course

2C



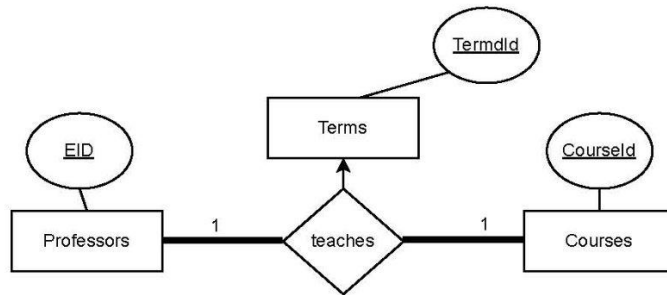
Every professor teaches exactly one course (no more, no less). There is total participation between the professor and courses as a must teach exactly one course

2D

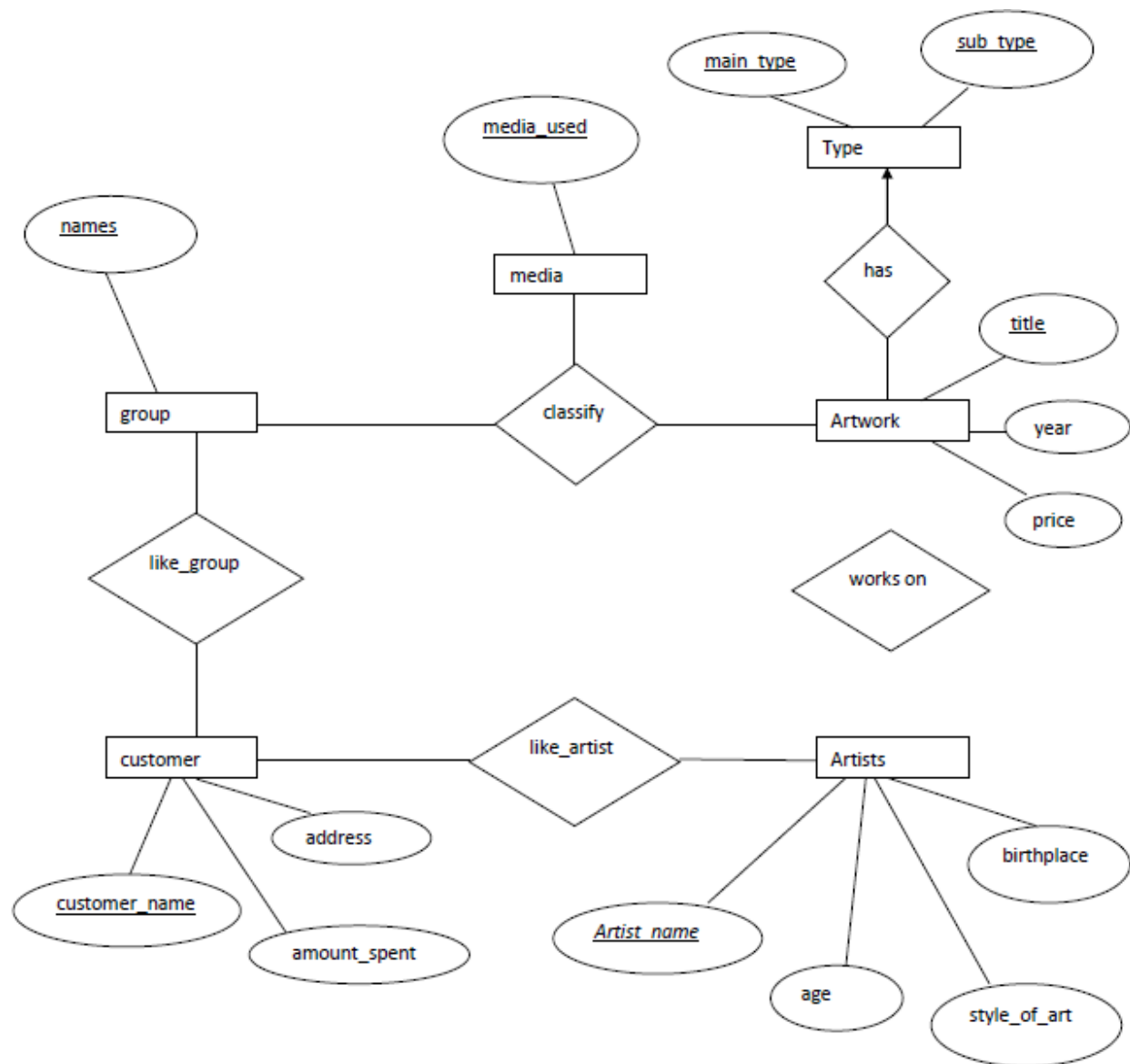


Every professor teaches exactly one course (no more, no less), and every course must be taught by some professor. Hence there is total participation between entities

2E



Question 3:



Explanation: Here in the schema of the gallery, below are the relations which can exist along with their attributes.

a) One artist can work on multiple artworks; which is defined by the one-to-many relationship named "works on". The Artists can have 4 attributes: "Artist_name", "age", "Style_of_art", "birthplace", where "Artist_name" is the primary key. And the Artwork has 3 attributes: "title", "year", "price" with "title" as the primary key.

b) There can be more than one Artworks for a particular Type. So there is a relation of one-to-many between the Artworks and Type. Also the type can have the attribute of sub_type. This relationship is named "has".

Assumption: we assume that an artwork can have only one type. The question explicitly talks about the group entity set but not much on description of type. We have assumed that there is not always a sub_type present for a type.

So we make the combination of Main_type and sub_type as the primary key in Type Entity.

c) Any single Artwork can belong to multiple groups. So the many-to-many relation "classify" exists between them. Also the same relationship exists between the media used to create an artwork. Whereas the Group has one attribute "names" which act as a primary key and Media has an attribute "media_used" as primary key respectively.

d) The other entity is the Customer with attributes "customer_name", "address", "amount_spent", where "customer_name" is the primary key. This entity has 2 relationships. One with the groups like which group artwork the customer purchases. Also the artists which the customer likes to purchase.

These relationships are named as "like_group" and "like_artist" respectively.