Assignment 3

3.17. Composite and multivalued attributes can be nested to any number of levels. Suppose we want to design an attribute for a STUDENT entity type to keep track of previous college education. Such an attribute will have one entry for each college previously attended, and each such entry will be composed of college name, start and end dates, degree entries (degrees awarded at that college, if any), and transcript entries (courses completed at that college, if any). Each degree entry contains the degree name and the month and year the degree was awarded, and each transcript entry contains a course name, semester, year, and grade. Design an attribute to hold this information. Use the conventions in Figure 3.5.

{Previous_College(College_name, Start_date, End_date, {Degrees_Awarded(Degree_name, Degree_Month Degree_Year)}, {Courses_completed(Course_name, Semester, Year, Grade)}}

3.21. Design an ER schema for keeping track of information about votes taken in the U.S. House of Representatives during the current two-year congressional session. The database needs to keep track of each U.S. STATE's Name (e.g., 'Texas', 'New York', 'California') and include the Region of the state (whose domain is {'Northeast', 'Midwest', 'Southeast', 'Southwest', 'West'}). Each CONGRESS_PERSON in the House of Representatives is described by his or her Name, plus the District represented, the Start_date when the congressperson was first elected, and the political Party to which he or she belongs (whose domain is {'Republican', 'Democrat', 'Independent', 'Other'}). The database keeps track of each BILL (i.e., proposed law), including the Bill_name, the Date_of_vote on the bill, whether the bill Passed_or_failed (whose domain is {'Yes', 'No'}), and the Sponsor (the congressperson(s) who sponsored—that is, proposed—the bill). The database also keeps track of how each congressperson voted on each bill (domain of Vote attribute is {'Yes', 'No', 'Abstain', 'Absent'}). Draw an ER schema diagram for this application. State clearly any assumptions you make.

ENTITY TYPES:

- **STATE**
  - Represents a unique state in the United States. Its name ***SName*** is the unique attribute since every state's name is unique. There is one other attribute, ***Region***, which represents one of

the five regions in the USA; *"Northeast", "Midwest", "Southeast", "Southwest", or "West"*. It can have one of these regions as its value.

- **CONGRESS_PERSON**
  - This entity type represents the state representatives in the House of Representatives. The unique attribute if this entity type is ***RName*** which represents the representative's name. Other attributes of this entity type are ***District***, ***Start_date***, and ***Party***. ***District*** refers to the district which the congress person represents. ***Start_date*** reflects the date when they were elected, and ***Party*** tells us what political party they belong to. The political party options include *Republican, Democrat, Independent or Other*.

- **BILL**
  - This entity type represents a proposed legislation and it has 3 attributes. Its unique attribute is its name, ***Bill_name***. Its other two attributes are ***Date_of_vote*** and ***Pass_or_fail.*** They represent the date that the bill was voted on and whether or not the bill passed, respectively. The attribute ***Pass_or_fail*** is of Boolean type represent as **"***Yes"* if it passed and *"No"* if it failed.

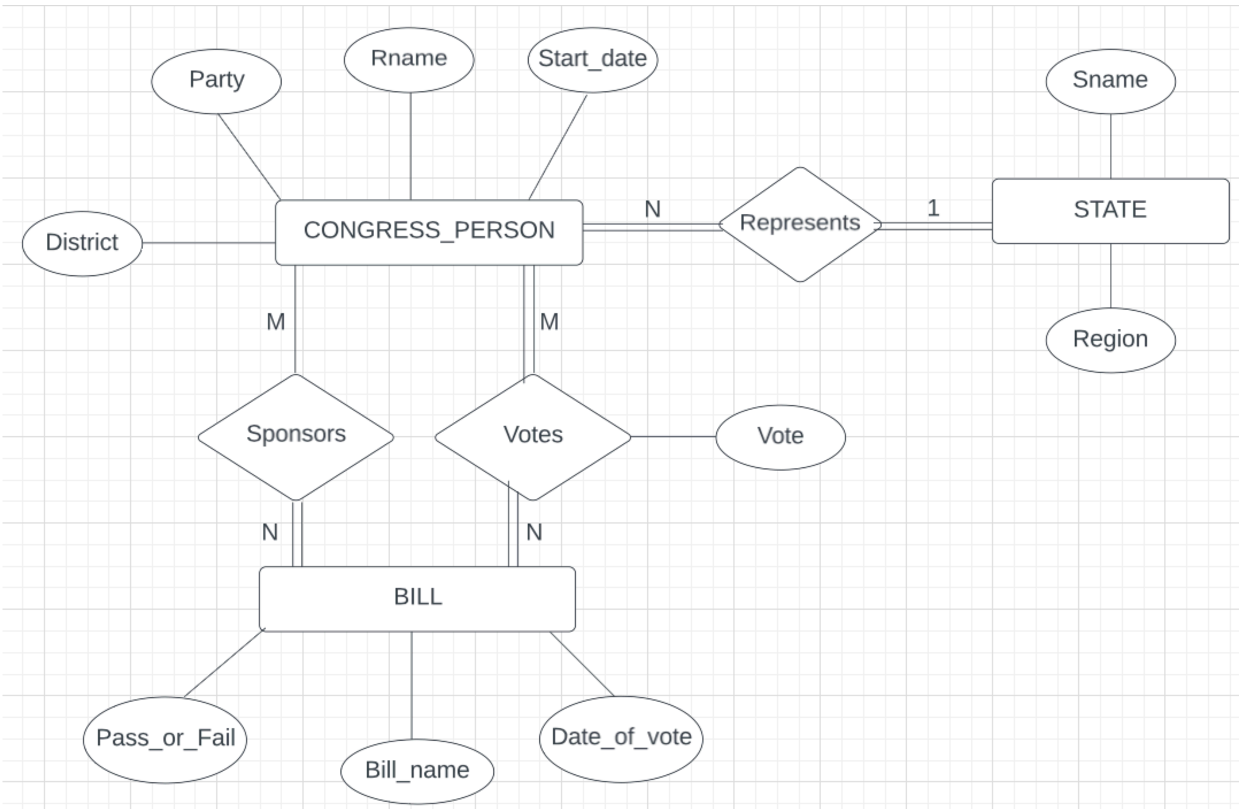
RELATIONSHIP TYPES

- **SPONSORS**
  - Relates the entity types ***CONGRESS_PERSON*** and ***BILL***.
  - Cardinality of this relationship type is as follows:
    - A lone congressman can sponsor many bills and a collection of sponsors can sponsor a single bill
    - Therefore, the cardinality of this relationship type is ***many to many***.
  - Bill's participation is total in this relationship since is must be sponsored by a state representative.

- **VOTES**
  - Also relates to the entity types ***CONGRESS_PERSON*** and ***BILL***.
  - This relationship stores how the congress representative voted on a particular bill which is stored in its attribute ***Vote*** which has the domain *{'Yes', 'No', 'Abstain', 'Absent'}*.
  - Since a state representative can vote for many different bills and single bill can get votes from many different congressman, this relationship type's cardinality is ***many to many***.
  - Every congressman must vote and every bill must be voted on, therefore, both the bill's and the congressmen's participation are total.

- **REPRESENTS**
  - This relationship type relates ***CONGRESS_PERSON*** and **STATE** entity types.

- Since it is possible that each state can be represented by multiple representatives but a representative can only represent one state, the cardinality of this relationship type is ***many to one***.
- Participation is total for both the congressman and the state since each state must be represented and each congressman must represent a state.



3.23. Consider the ER diagram shown in Figure 3.22 for part of a BANK database. Each bank can have multiple branches, and each branch can have multiple accounts and loans.

a. List the strong (nonweak) entity types in the ER diagram.

b. Is there a weak entity type? If so, give its name, partial key, and identifying relationship.

c. What constraints do the partial key and the identifying relationship of the weak entity type specify in this diagram?

d. List the names of all relationship types, and specify the (min, max) constraint on each participation of an entity type in a relationship type. Justify your choices.

e. List concisely the user requirements that led to this ER schema design.

f. Suppose that every customer must have at least one account but is restricted to at most two loans at a time, and that a bank branch cannot have more than 1,000 loans. How does this show up on the (min, max) constraints?

a. **BANK, BANK_BRANCH, ACCOUNT, LOAN, CUSTOMER.**

b. **Weak entity type: BANK_BRANCH.**
Partial key of BANK_BRANCH: Branch_no.
Identifying relationship: BRANCHES.

c.
Partial key constraints: Total participation constraint.
Identifying relationship constraints: 1:N cardinality ratio constraint.

d.
BRANCHES:
1 BANK to N BANK_BRANCH
A bank can have many different local branches.

ACCTS:
1 BANK_BRANCH to N ACCOUNT
A bank branch can have many different accounts.

LOANS:
1 BANK_BRANCH to N LOAN
A bank branch can have many different loans.

A_C:
M ACCOUNT TO N CUSTOMER
Many different customers can have many different accounts.

L_C:
M LOAN TO N CUSTOMER
Many different customers can have many different loans.

e.
A bank is organized into branches, which have their address and branch number recorded.

The bank branches store the accounts and loans of customers, recording things like : account numbers, account balance, account type, loan numbers, loan amounts, and load types.

Customer information is also stored, recording their SSN, name, phone number, and address.

f.
This would change the A_C relationship type, with a (1,M) constraint on ACCOUNT.

Also, the L_C relationship type would have a (0,2) constraint on LOAN.

The LOANS relationship type would have a (0,2000) constraint on LOAN.

3.28. Consider the ER schema for the MOVIES database in Figure 3.25.

Assume that MOVIES is a populated database. ACTOR is used as a generic term and includes actresses. Given the constraints shown in the ER schema, respond to the following statements with *True, False*, or *Maybe*. Assign a response of *Maybe* to statements that, although not explicitly shown to be *True*, cannot be proven *False* based on the schema as shown. Justify each answer.

a. There are no actors in this database that have been in no movies.
b. There are some actors who have acted in more than ten movies.
c. Some actors have done a lead role in multiple movies.
d. A movie can have only a maximum of two lead actors.
e. Every director has been an actor in some movie.
f. No producer has ever been an actor.
g. A producer cannot be an actor in some other movie.
h. There are movies with more than a dozen actors.
i. Some producers have been a director as well.
j. Most movies have one director and one producer.
k. Some movies have one director but several producers.
l. There are some actors who have done a lead role, directed a movie, and produced a movie.
m. No movie has a director who also acted in that movie.

a. **True. There is a total participation constraint on ACTOR for the PERFORMS_IN relationship.**

b. **Maybe. The M:N cardinality ratio constraint on PERFORMS_IN allows for this possibility.**

c. **Maybe. Any two actors can have leading roles in a movie.**

d. **True. The LEAD_ROLE relationship has a "2:N" constraint on it.**

e. Maybe. There is no total participation constraint on ALSO_A_DIRECTOR.

f. False. Actors can be producers, as indicated by the ACTOR_PRODUCER relationship.

g. False. A producer can be an actor, and an actor can perform in many different movies.

h. Maybe. There is no maximum number constraint for MOVIE on the PERFORMS_IN relationship.

i. Maybe. Such a relationship between DIRECTOR and PRODUCER, while possible, is not part of the database schema.

j. Maybe. A movie can have many different producers.

k. Maybe. A movie can have one or many different producers.

l. Maybe. The database schema allows for such occurrences.

m. False. A director can participate in both the ALSO_A_DIRECTOR and DIRECTS relationships.