HOW TO TRANSFORM AN ENTITY RELATIONSHIP MODEL INTO A RELATIONAL MODEL?

Two very important models used in database design are the Entity-Relationship (E- R) Model and the Relational Model. Since I have already read two articles about relational model and entity-relationship model, I really wonder about the correlations and transformation between them. In the E-R model, data is represented using entities, and relationships are defined between these entities. However, with the relational model, the entities and their relationships follow strict guidelines. Usually, an E-R model is first developed, and then it is transformed into a relational model.

The relational model is similar to the E-R model but based on a branch of mathematics called relational algebra, and as such, there are strict definitions and rules regarding the elements of this model. Here are some of these definitions and rules:

- A relation is defined as a two dimensional table that contains rows and columns. Rows are instances of the entity and columns are attributes of the entity.
- No two rows of the table must be identical — each row must be unique (the contents of the row that is), and the order of the rows is not important.
- Each column must have a unique name, and the order of the columns is not important. They must also contain single values — multiple values and repeating groups are not allowed.
- A key is a group of one or more attributes that uniquely identifies a row — each table must have a key. Since each row in the table is unique, the key is what is used to distinguish one row from another. It can be comprised of a single attribute or it can be composite, comprising of more than one attribute.
- Every column (attribute) in a table must depend solely on the primary key. It should not depend on any other column in the table, or on part of the primary key if it is a composite key.

Exercise No.1

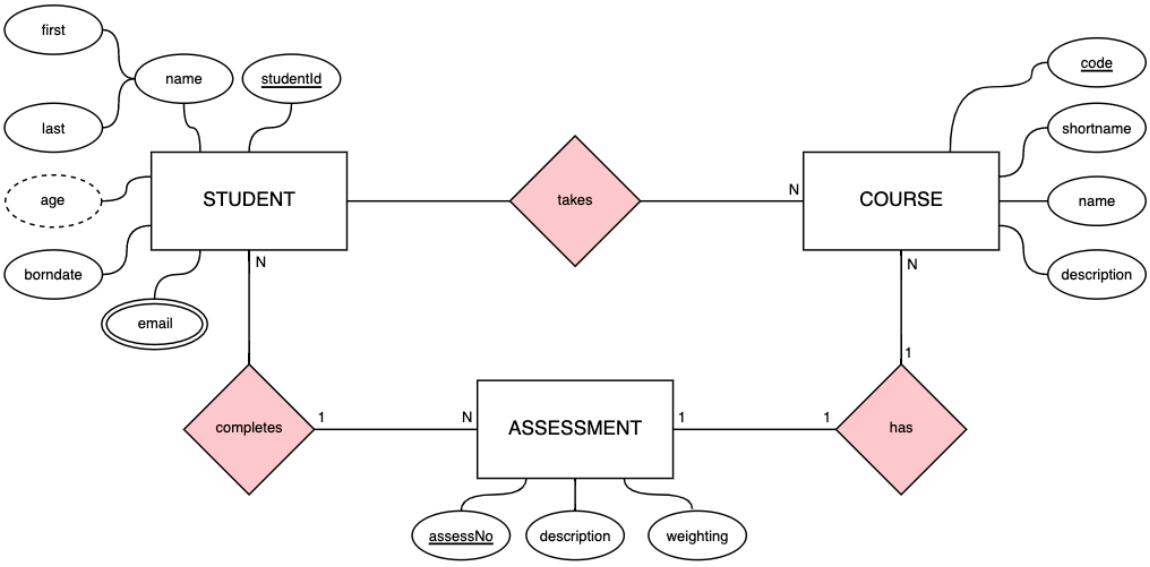
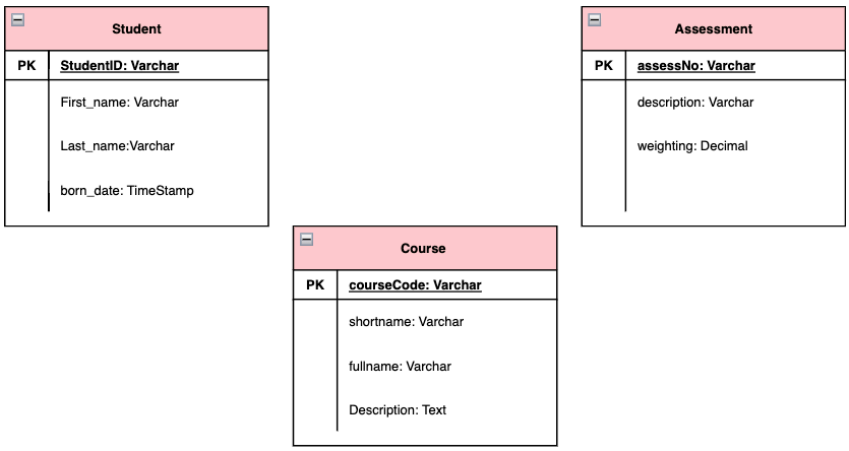


Fig. 1. A student takes 0 or N courses, each course has multiple assessment, and all students complete 1 or N of these assessments

An E-R diagram is used to represent the E-R model. It contains all known entities for the given scenario, their attributes, identifiers, and the relationships that exist among the entities. Consider the ER diagram in Fig 1

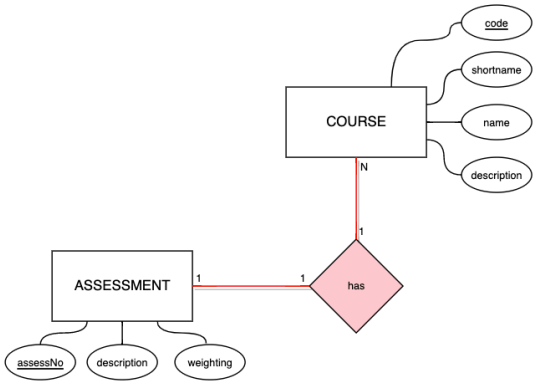
Step 1: Clearly identify the primary key and attributes for each entity defined in the E-R model, and ensure that it is in accordance with the rules of the relational model as previously discussed.



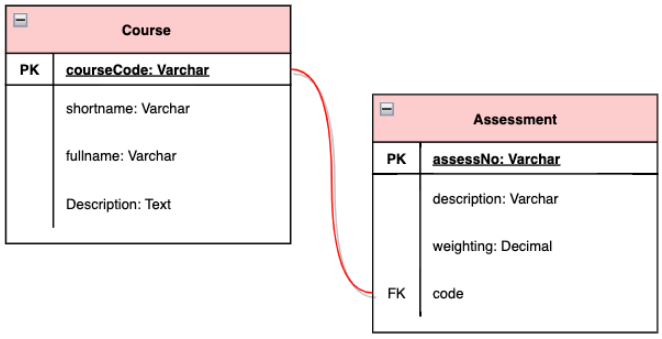
Each entity, with its clearly identified primary key (indicated by PK), and attributes satisfying the previously discussed rules, becomes a table in the relational model.

Step 2: Group together tables (formerly entities) and their relationships that have a cardinality of one relationship with 0:1 or 1:1 for their opt:card. (Optionality says what can and must happen in a relationship, and cardinality indicates the number of entity occurrences in a relationship)That is, absorb relationships where the cardinality is one into the corresponding tables.

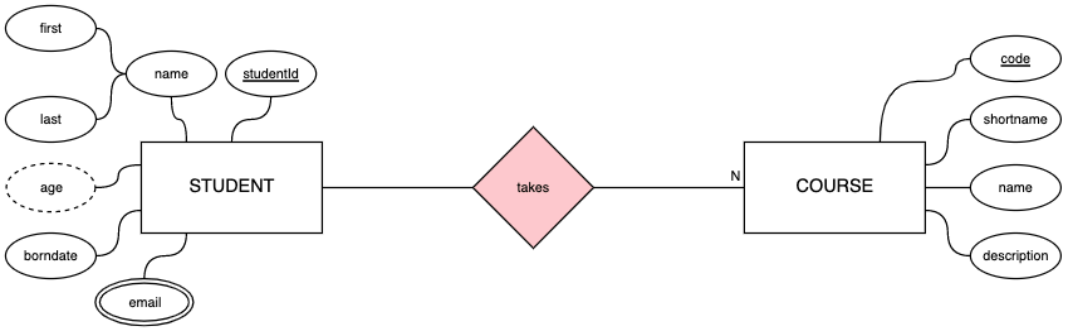
- Maintain the initial structure for the absorbing table – do not change its primary key or any of its attributes.



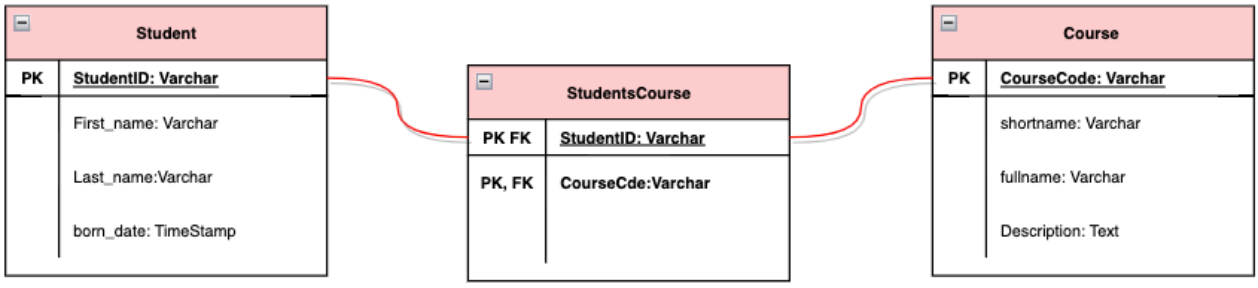
- The primary key of the other table in the relationship becomes a foreign key in the absorbing table. This is indicated by an FK in the absorbing table.
- If the cardinality between the absorbed relationship and the other table in the relationship was N (0:N or 1:N), then the modified table becomes the many part of a one-to-many relationship in the new relational model.
- Otherwise, if the cardinality between the absorbed relationship and the other table if the relationship was 1 (0:1 or 1:1), then the new relationship becomes a one-to-one relationship in the new relational model (not shown).



Step 3: The remaining relationships whose cardinalities are N (1:N or 0:N) on both sides become new tables in the new relational model.



- The primary keys from the two tables involved in the relationship become a composite primary key in the new table, and the new table usually has a name that is a combined form of the two original table names.
- The newly created table becomes the many part of the relationship between both tables, and thus creates a many-to-many relationship between the two pre-existing tables. In some instances, the newly created may have its own attributes, but this is rare.



Step 4: Consider other elements from ER as weak entities, composite attributes

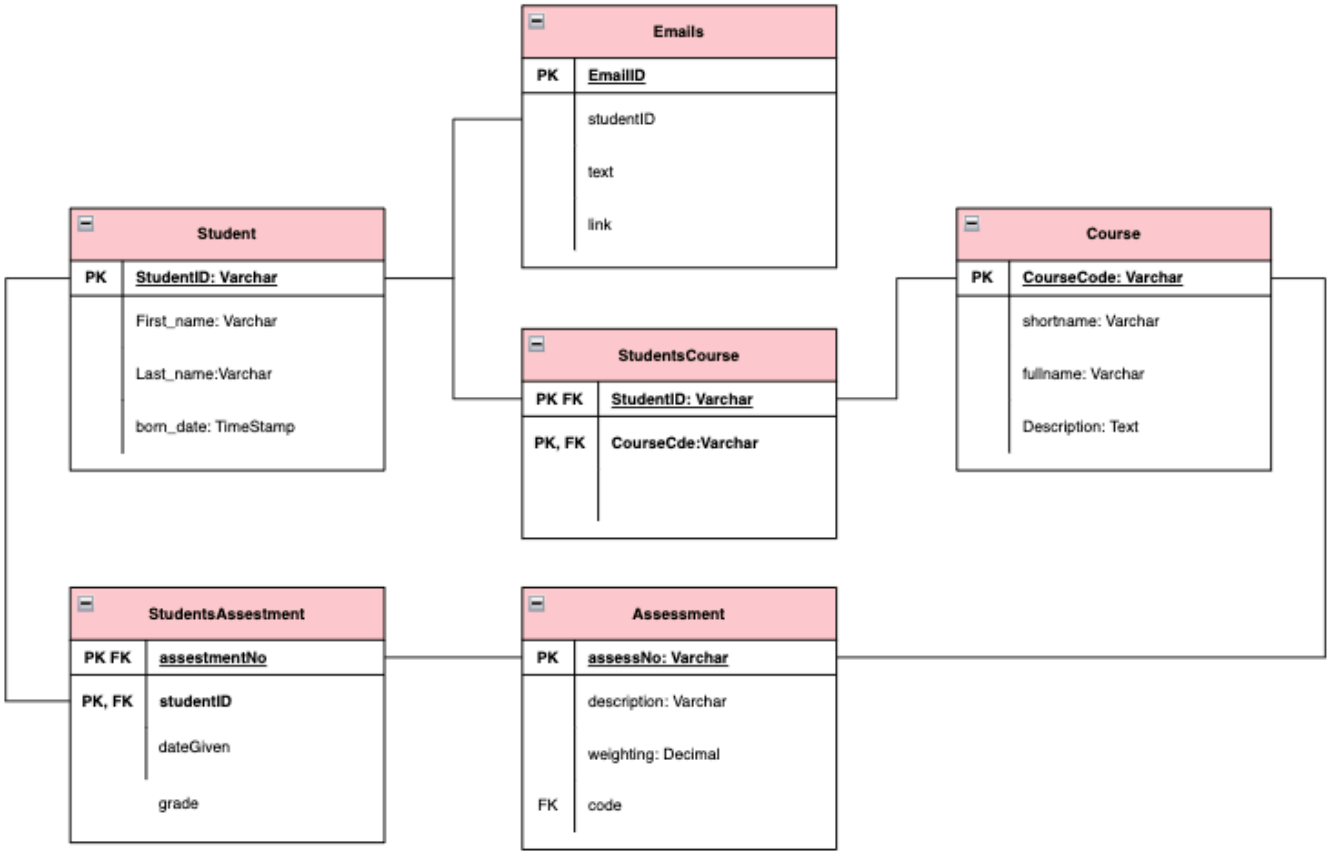
- Weak entity sets and identifying relations are translated into a single table and consider when an owner or strong entity is deleted all owned entities should also be deleted.
- Hierarchies, generalizations or specializations have two approaches, create a relation which have each specialization id and throw foreing key identify row. Another is set one or several subclasses for the attribute that indicates the specific class
- Participation constraints are modeled with a DDL language as SQL. We can capture participation constraints involving one entity are considered cases on UPDATE, DELETE or TRUNCATE

The resulting database relational schema is given below.

- Student(sudentID, first_name, last_name, born_date)
- Emails(emailId, studentID*, text, link)
- Course(courseCode, short_name, full_name, description)
- Assessment(assestmentCode, des)
- StudentAssestment(sudentID*, assessmentCode*, date_given, grade)
- StudentCourse(sudentID*, courseCode*,)

(_): primary keys, (*): foreign keys.

The resulting diagram is given below.



The ER diagram represents the conceptual level of database design. A relational schema is at the logical level of database design. When translating your diagram into a schema, an entity type turns into a table. The name of the entity should become the name of the table.

While an E-R diagram displays the logical nature of the data that exists in the user's domain, the relational model shows how this data will be represented in a Relational Database Management System (RDBMS). This model is important because the most common and popular database management systems in use today are relational, and relational model diagrams can be directly implemented on any RDBMS.

Normally one should minimize the number of tables in the database to reduce query processing time. However, we are focusing on the stable translation method which actually increases the number of tables in the database. It does not increase the performance of the system; however it creates a framework that is more easily maintained. System development is always a balancing act. The benefit of the stable translation is that it:

- Does not allow null values if possible
- Provide a semantically clear design
- Provide a design that accommodates potential changes of the schema

The reality of database design is that In general, decreasing the number of tables means increased efficiency of query processing. However, it also means more null values are allowed in columns during data insertion.

The last technique is called stable translation although there are other such as mapped and mapped with total/partial translation.

Do not forget

- Every entity becomes a table
    - All the attributes of the entity become the attributes of the table

- Every relationship becomes a table
    - Add Relation ID to the relationship table
    - Add any non-key attribute of the relationship to the relationship table
    - A weak relationship is always combined with the weak entity

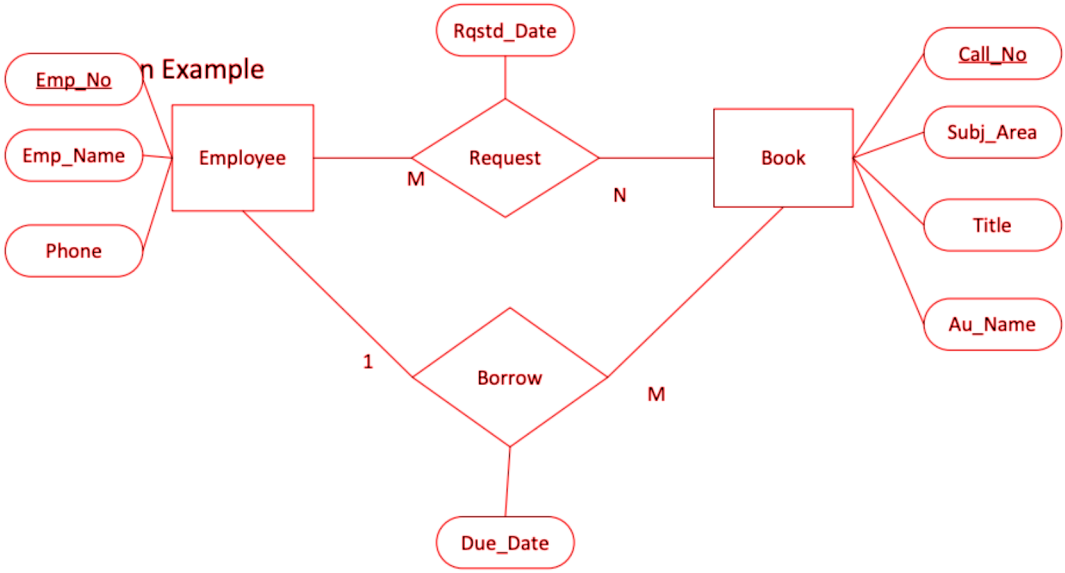
- Stable translation is called "stable" because a change in the cardinality or participation constraints does not change the table structure

- Advantage
  - Provides stable schema against constraint changes

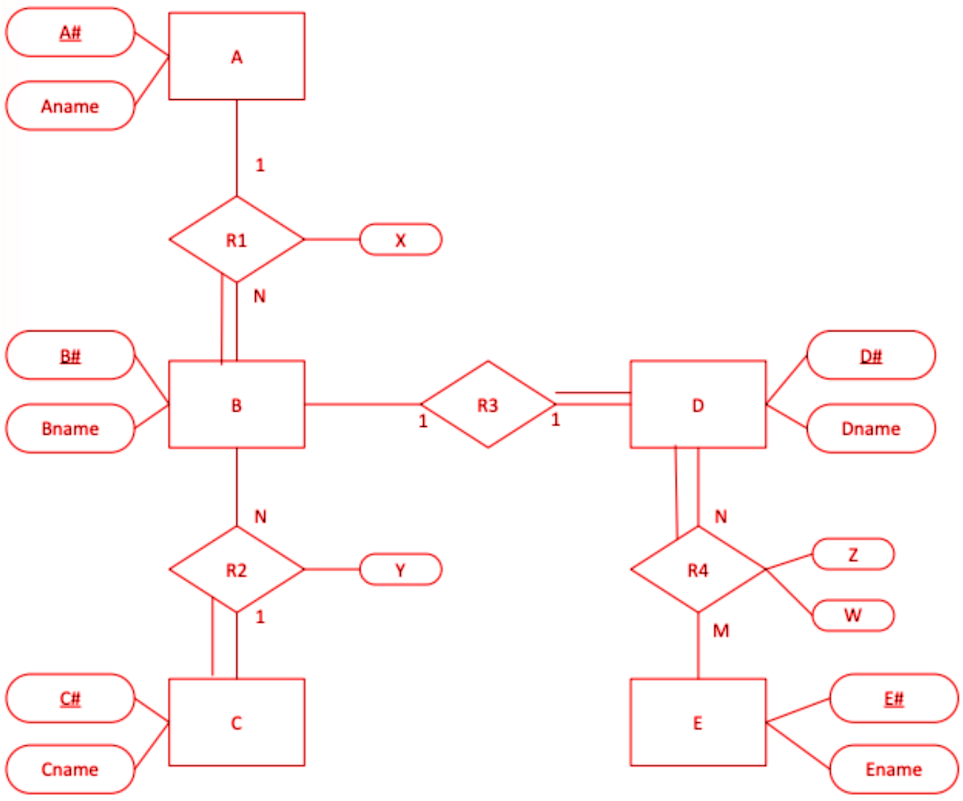
- Disadvantage
  - Generates a larger number of tables than other methods More time to process query statements

Extra exercises

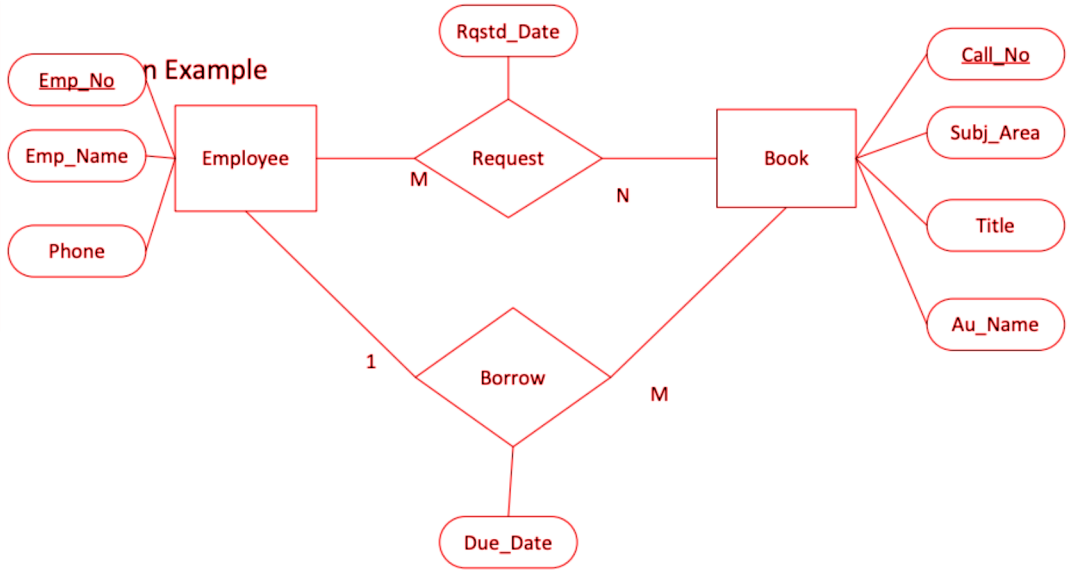
a. Translate the following ER diagrams into Relational schema and draw using UML



b. Translate the following ER diagrams into Relational schema and draw using UML



a) Translate the following ER diagrams into Relational schema and draw using UML



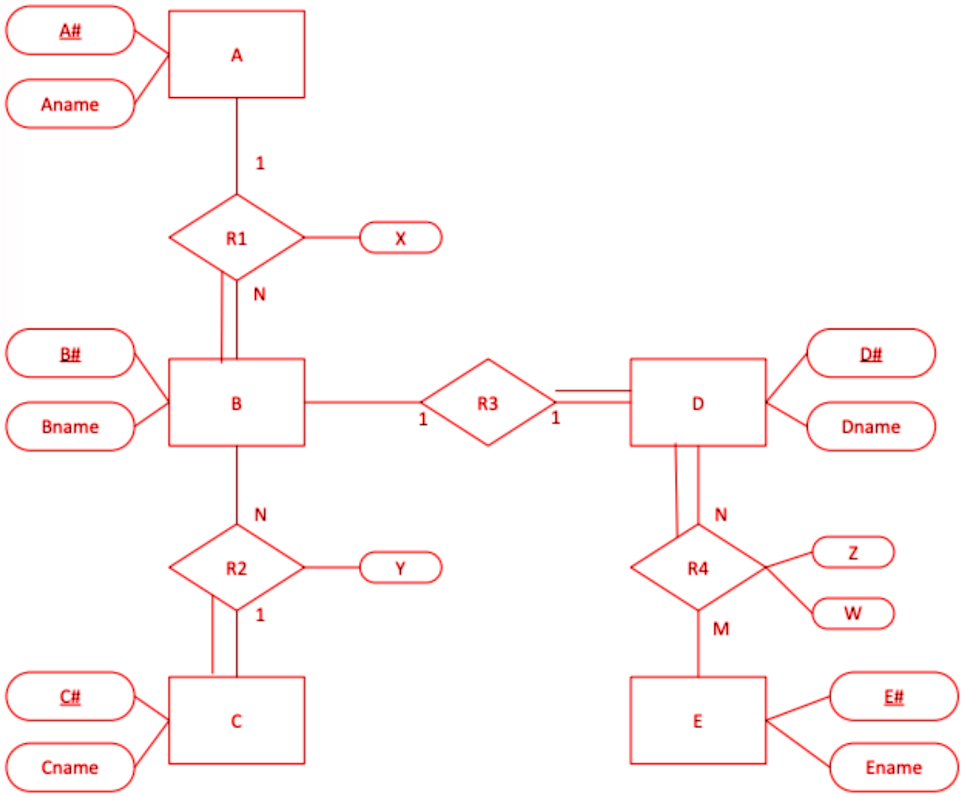
STABLE TRANSLATION METHOD

- **Employee(Emp_No, Emp_Name, Phone)**
- **Request(Emp_No, Call_No, Rqstd_Date)**
- **Borrow(Call_No, Emp_No, Due_Date)**
- **Book(Call_No, Subj_Area, Title, Au_Name)**

MAPPED TRANSLATION METHOD

- **Employee(Emp_No, Emp_Name, Phone)**
- **Request(Emp_No, Call_No, Rqstd_Date)**
- **Book(Call_No, Subj_Area, Title, Au_Name, Emp_No, Due_Date)**

b) Translate the following ER diagrams into Relational schema and draw using UML



STABLE TRANSLATION METHOD

- **A(A#, Aname)**
- **B(B#, Bname)**
- **C(C#, Cname)**
- **D(D#, Dname)**
- **E(E#, Ename)**
- **R1(B#, A#, X)**
- **R2(B#, C#, Y)**
- **R3(D#, B#)**
- **R4(D#, E#, Z, W)**