Batch: B1 Roll No.: 1711072
Experiment / assignment / tutorial No. 2

Grade: AA / AB / BB / BC / CC / CD /DD

Title: Mapping ER and EER Model to Relational Model

Objective: To apply mapping techniques to map ER diagram and EER to its equivalent relational model

Expected Outcome of Experiment:

CO 2: Convert entity-relationship diagrams into relational tables, populate a relational database and formulate SQL queries on the data Use SQL for creation and query the database.

Books/ Journals/ Websites referred:

G. K. Gupta :"*Database Management Systems*", McGraw – Hill

1. Korth, Slberchatz, Sudarshan : "Database Systems Concept", 6th Edition , McGraw Hill
2. Elmasri and Navathe, "Fundamentals *of Database Systems*", 5thEdition, PEARSON Education.

.

Relational Model

Relational Model represents the database as a collection of relations. Relational model can be thought of as table of values, each row in the table represents collection of related data values. In the relational model, each row in the table represents the fact that corresponds real world entity or relationship. The table name and column name are used to interpret the meanings of the values in each row.

In formal relational model terminology, a row is called tuple, a column header is called an attribute, and table is called a relation. The data type describing the types of values that can appear in each column is represented by a domain of possible values. Thus Relation is set of tuples.

Procedure for doing the Relation Model (ER to Relational Mapping)

1. **Mapping of Regular Entity**

   - For each regular (strong) entity type in the ER schema, create a relation R that includes all the simple attributes of E.
   - Choose one of the key attributes of E as the primary key for the relation

2. **Mapping of Weak Entity**

   - For each weak entity type W in the ER schema with owner entity type E, create a relation R and include all attributes of the weak entity as attributes of the new relation R.
   - Then, include the primary key of the owner entity as foreign key attributes of R
   - The primary key of R is the *combination of* the primary key(s) of the owner(s) and the partial key of the weak entity type W, if any.

3. Mapping of Binary 1:1 Relationship Types

- For each 1:1 relationship type identify the entities participating in the relationship. There are two possible approaches below:

- a) Foreign Key approach:

  Choose one of the relations and include a foreign key in one relation (S) which is the primary key of the other relation (T). It is better to choose an entity type with *total participation* in the relationship in the role of S.

- b) Merged relation option:

  An alternate mapping of a 1:1 relationship type is possible by merging the two entity types and the relationship into a single relation. This may be appropriate when *both participations are total.*

4. Mapping of Binary 1:N Relationship Types

- For each regular 1:N relationship type R, identify the relation S, which is the entity on the N-side of the relationship.
- Include as foreign key in S the primary key of the relation which is on the 1 side of the relationship
- Include any simple attributes of the 1:N relation type as attributes of S.

5. Mapping of Binary M:N Relationship Types

- For each M:N relationship type, *create a new relation* S to represent the relationship

- Include as foreign key attributes in S the primary keys of the entities on each side of the relationship; *the combination of the two primary keys will form the primary key* of S

- Also include any simple attributes of the M:N relationship type as attributes of S.

6. Mapping of Multivalued Attributes.

- For each multivalued attribute A, create a new relation. This relation will include an attribute corresponding to the multi-valued attribute, plus the primary key attribute of the relation that has the multi-valued attribute, K
- The primary key attribute of the relation is the foreign key representing the relationship between the entity and the multi-valued relation
- The primary key of R is the combination of A and K

7. Mapping of N-ary Relationship Types

- For each n-ary relationship type R, where n>2, create a new relation S to represent the relationship.
- Include as foreign key attributes in S the primary keys of the relations that represent the participating entities
- Also include any simple attributes of the n-ary relationship type as attributes of S

8. Options for Mapping Specialization or Generalization

- Convert each specialization with m subclasses $\{S_1, S_2,\ldots,S_m\}$ and generalized superclass C, where the attributes of C are $\{k,a_1,\ldots a_n\}$ and k is the (primary) key, into relational schemas using one of the four following options:

Option 8A: Multiple relations-Superclass and subclasses.

Option 8B: Multiple relations-Subclass relations only.

Option 8C: Single relation with one type attribute.

Option 8D: Single relation with multiple type attributes.

9. Mapping of Union Types (Categories).

- For mapping a category whose defining superclass have different keys, it is customary to specify a new key attribute, called a surrogate key, when creating a relation to correspond to the category.
- In the example below, create a relation OWNER to correspond to the OWNER category and include any attributes of the category in this relation. The primary key of the OWNER relation is the surrogate key, which we called OwnerId.

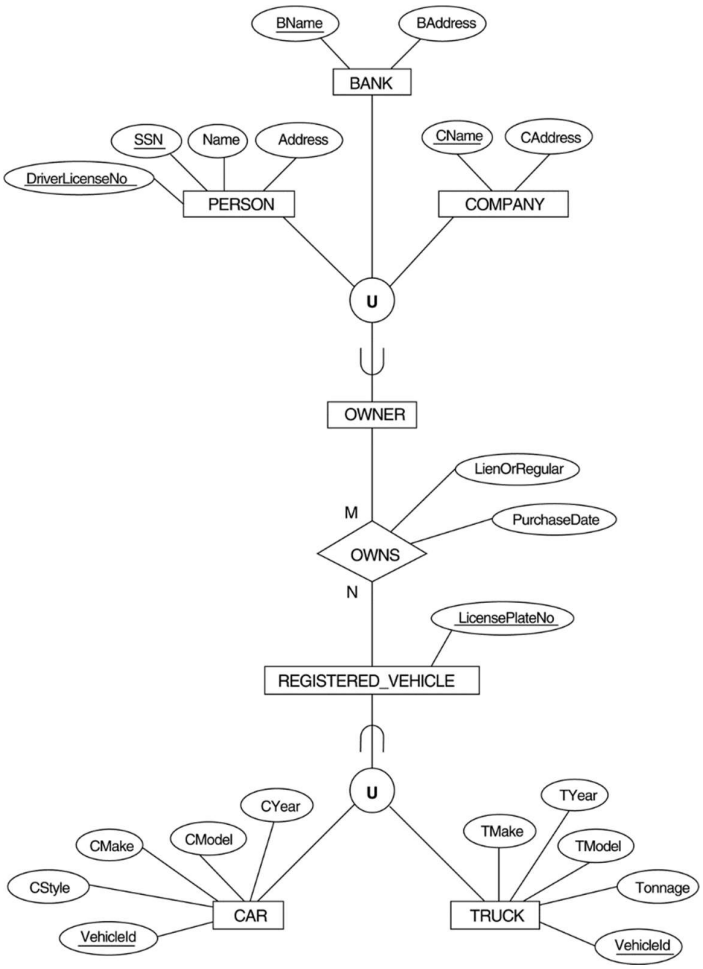


Figure 1: Two categories (union types): OWNER and REGISTERED_VEHICLE.

PERSON

SSN	DriverLicenseNo	Name	Address	

BANK

BName	BAddress	OwnerId

COMPANY

CName	CAddress	OwnerId

OWNER

OwnerId

REGISTERED_VEHICLE

VehicleId	LicensePlateNumber

CAR

VehicleId	CStyle	CMake	CModel	

TRUCK

VehicleId	TMake	TModel	Tonnage	TYear

OWNS

OwnerId	VehicleId	PurchaseDate	LienOrRegular

Figure 2: Mapping the EER categories (union types) in Figure 1 to relations.

Case Study considered for Database Design

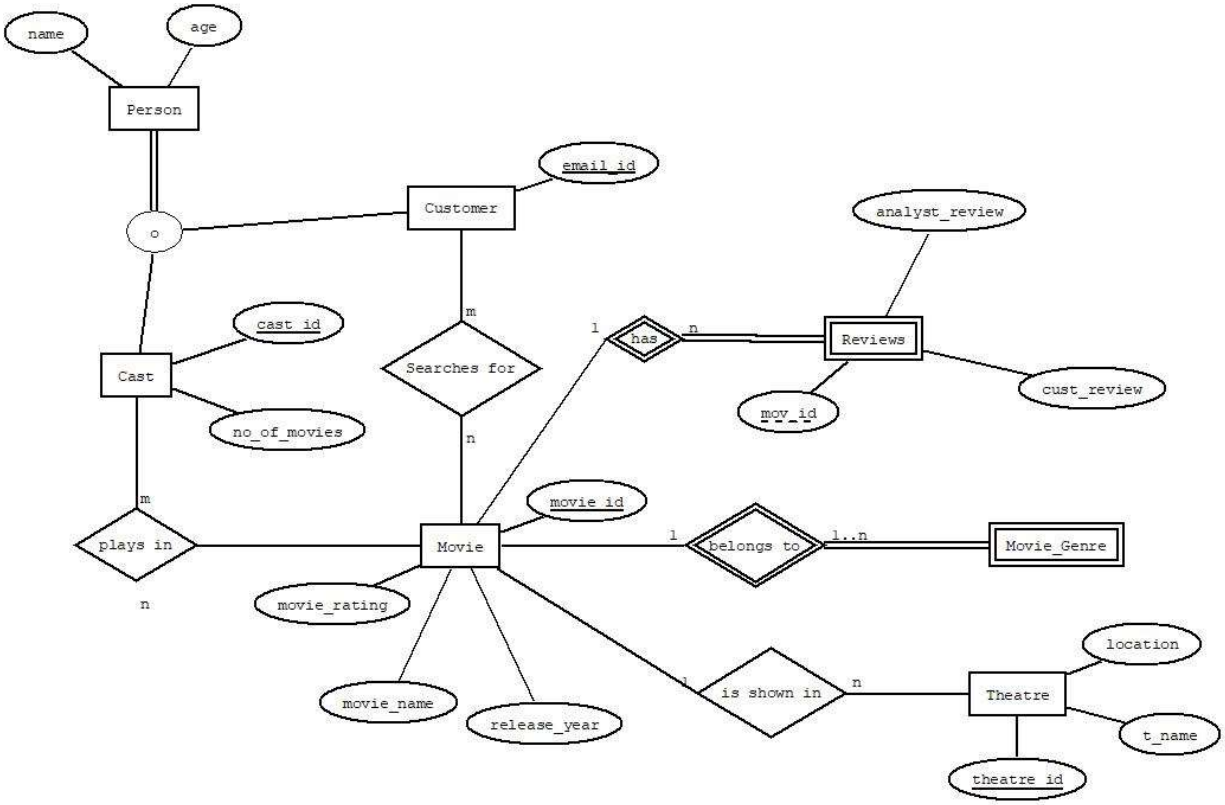
Online Movie Recommender System

Problem Statement:

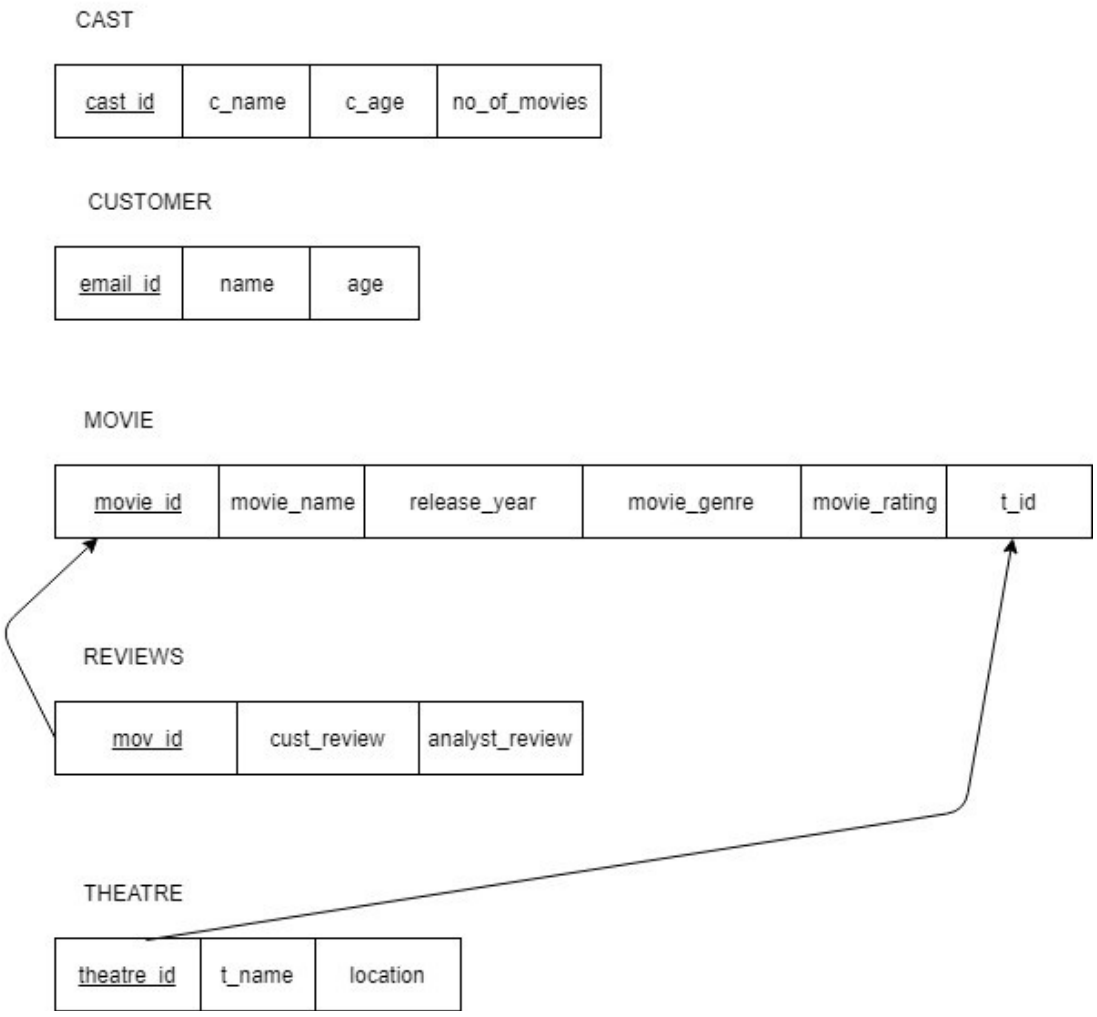
People seek information via words, recommendation letters, etc. The search engines retrieve this information using specific keywords mentioned by the user. Recommendation systems imitate this social process to enable quick filtering of the information on the web.
We aim to develop a database for this movie recommendation system which will contain the details of the following:
1. Movie, like the title, director, movie length;
2. Genre: the basis on which movies will be suggested
3. Customer: customer id, basic details
4. Theatre: name, address, timings
5. Cast: name, age, no. of movies done
6. Reviews: rating by customer, movie analyst

This database will serve as a basis for our movie recommendation system on the basis of user's genre preference, cast preference and review of the movies.

ER Diagram:



Relational Model for Project



Conclusion: Thus, by following the proper mapping techniques, we have successfully mapped our ER diagram to the Relational Model.