Exercise 1

Jasper Steinberg

January 20, 2025

1 Database systems

1.1 a)

A database is a collection of related data organized logically to represent some aspect of the real world, such as housing addresses. It is designed for specific purposes with inherent meaning and intended users. A database management system (DBMS) is a software system that enables users to create, manage, and interact with databases. It provides tools for defining, constructing, and manipulating data while ensuring efficiency, security, and accessibility.

1.2 b)

Program-data independence means that the structure of the data is stored separately from the programs that access it. This allows changes to the database structure without requiring changes to application programs, making maintenance easier and systems more flexible. Multi-user support refers to the ability of a DBMS to allow multiple users to access and manipulate the database simultaneously while maintaining data integrity and consistency, which enhances collaboration and productivity. The self-describing nature of a database means it includes metadata that describes the structure, constraints, and meaning of the data. This makes the database easier to understand and manage, supports dynamic queries, and ensures the enforcement of constraints.

2 The ER model

2.1 a)

An entity is a specific object or thing in the real world with an independent existence, like a particular employee or a car. An entity class, or entity type, defines a group of entities that share the same attributes, such as the "EMPLOYEE" entity class, which includes all employee entities in the database.

A relation represents a specific instance of data, such as a set of relationships between entities. A relation class, in contrast, defines the type of relationships and the participating entities, such as the "WORKS_FOR" relation class describing how employees are associated with departments. Attributes that describe references, such as a manager of a department or an employee's supervisor, should be modeled as relationships in the ER model, rather than as simple attributes.

Every entity must have one or more key attributes to ensure that it is uniquely identifiable within its entity set. Key attributes, such as an employee ID or a department code, have distinct values for each entity and allow for efficient identification, differentiation, and access to entities in the database. Without key attributes, it would not be possible to uniquely distinguish one entity from another within the same entity class.

Statement no.	Your answer	Short explanation
1	True	TacoID is underlined.
2	True	All output lines from taco to sauces, vegetables, cheeses and meat have a (0,n) marker.
3	False	The TacoOrder relation is marked with (1,n) on the Order side.
4	True	(1,n) is uncapped.
5	False	Order has a pickup time, and each shop is unique.
6	True	Customer has (0,n) marker in relation to Order, so it can exist without an Order.
7	False	A vegetable can exist outside of a taco, weight is only an attribute of the relation.
8	True	Job title is attribute of WorksAt, and an employee work at (1,n) shops.
9	Maybe	No way of knowing.
10	False	Can we NAN as name is not the identifier.

Table 1: Task 2b

3 Weak entity types, functional data models and new demands

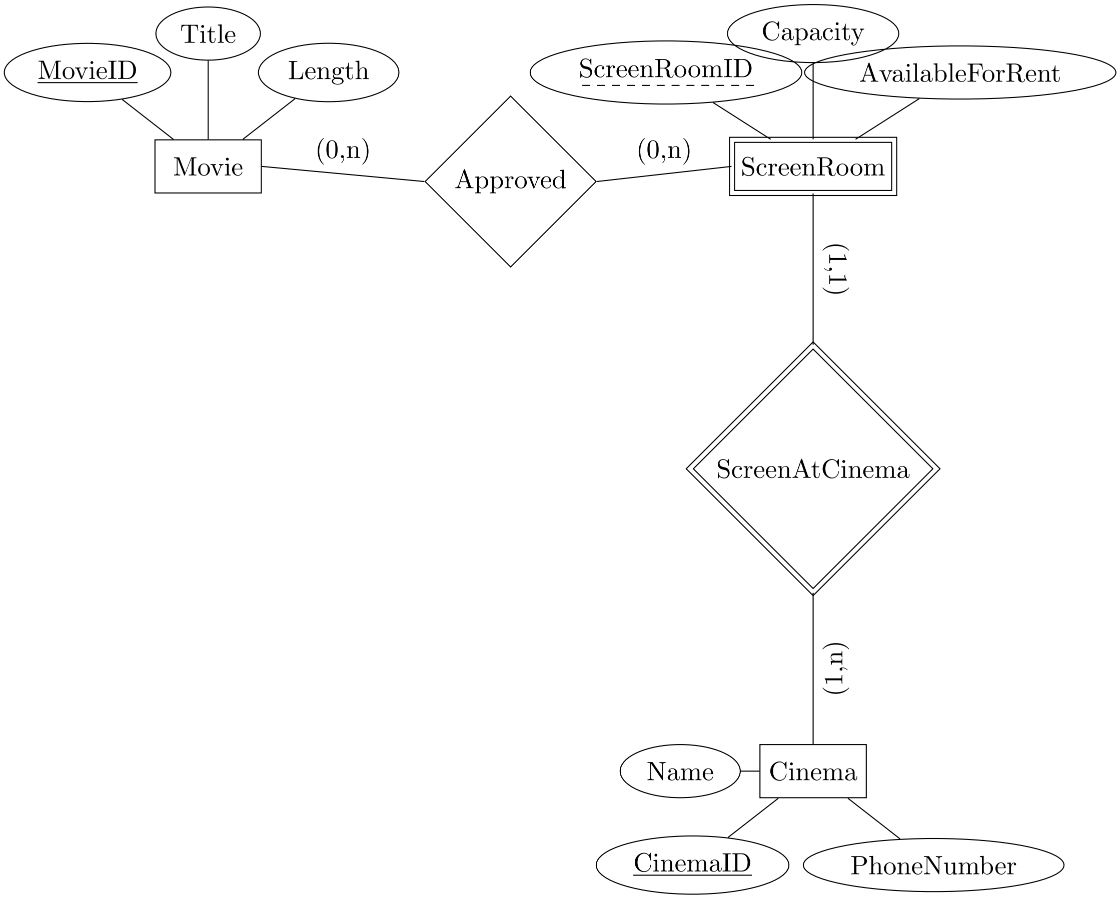
3.1 a)

Weak entity types are used when an entity dose not posses a natural unique identifier on its own, but can be uniquely identified with respects to another entity. The identifying entity type is the entity which ensures uniqueness, i.e. ”Cinema” in this case. The identifying Relationship is in this case the ”ScreenAtCinema” relation, which allows for differentiation of two rooms with the same name by specifying their respective cinemas. The partial key is ”ScreenRoomID”, which is the non-unique ID of the weak entity.

3.2 b)

A change to (0,1) is a problem, as a ScreenRoom now might not be associated to a Cinema, and we lose the general method of identification needed for a weak entity. In the case of (1,n) we can no longer model ScreenRoom as a weak identity type, as we have no way of ensuring uniqueness.

3.3 c)


Title
MovieID
Length
Movie
(0,n)
Approved
(0,n)
Capacity
ScreenRoomID
AvailableForRent
ScreenRoom
(1,1)
ScreenAtCinema
(1,n)
Name
Cinema
CinemaID
PhoneNumber

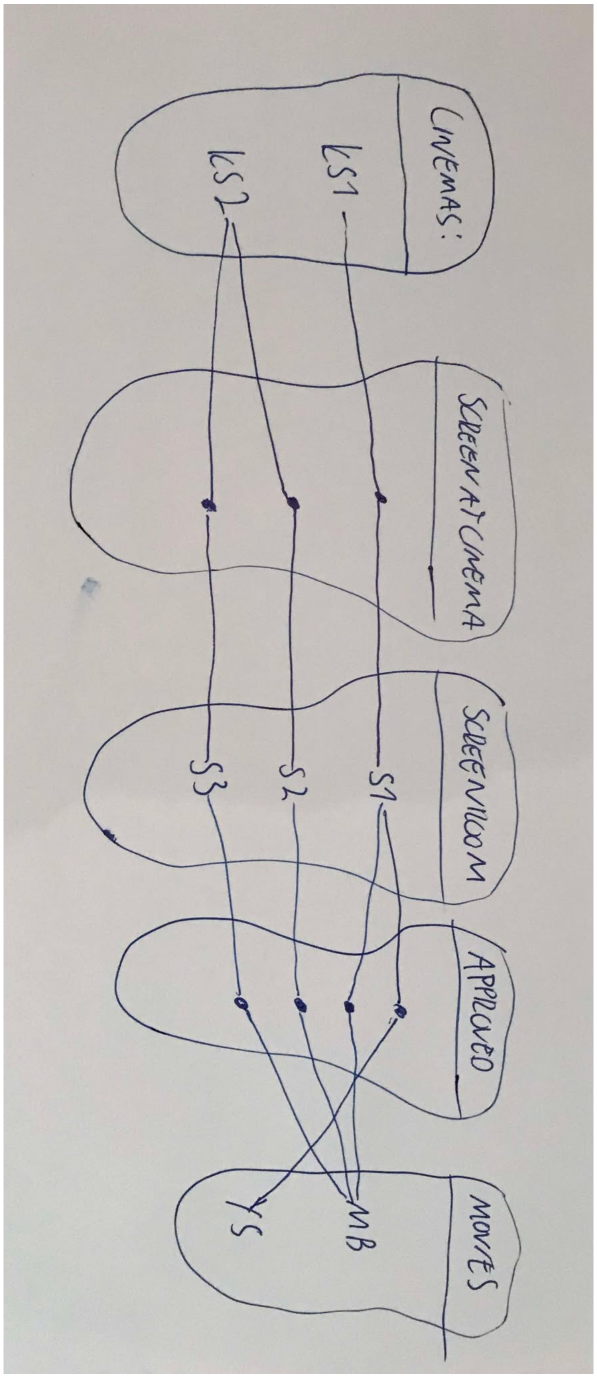
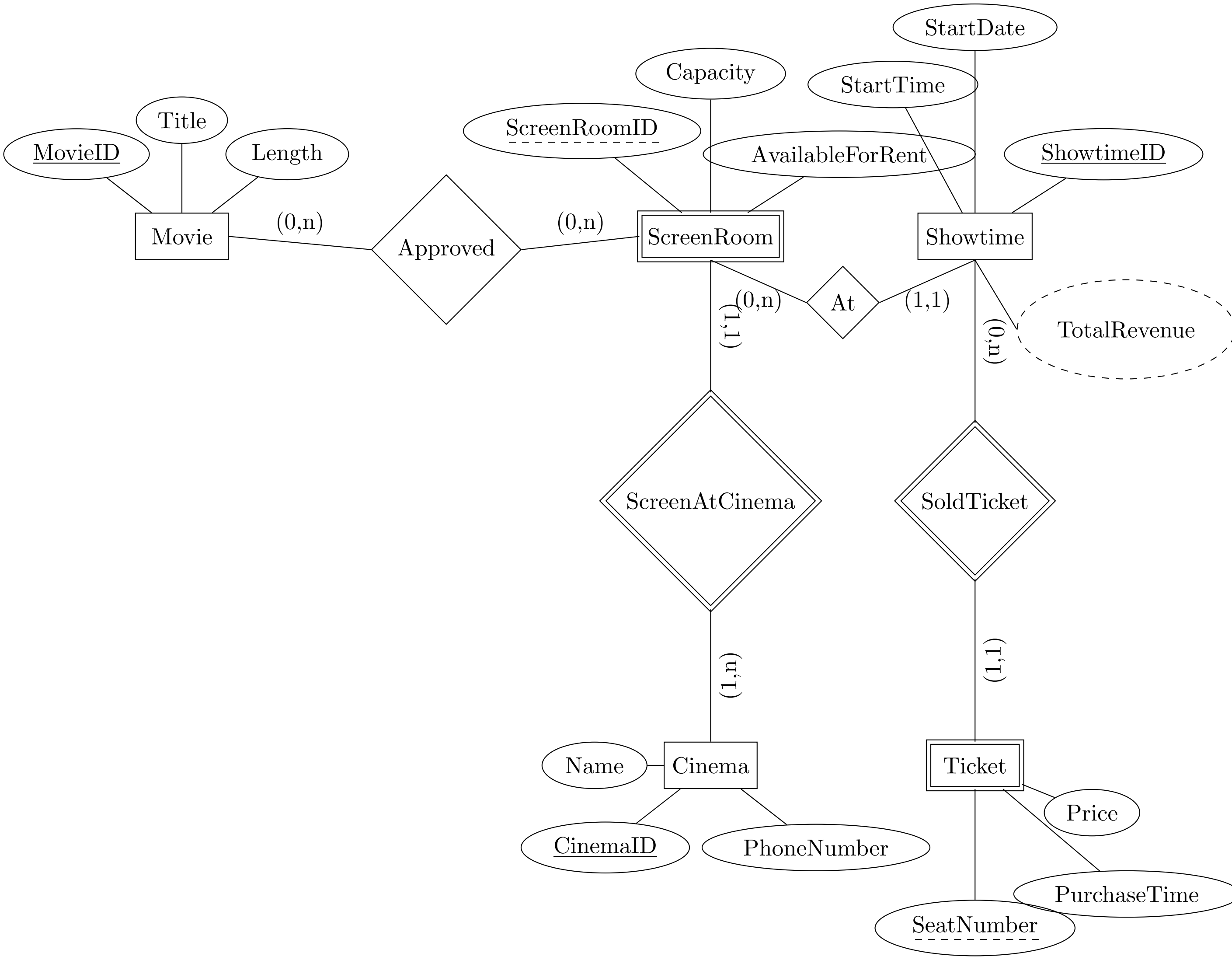
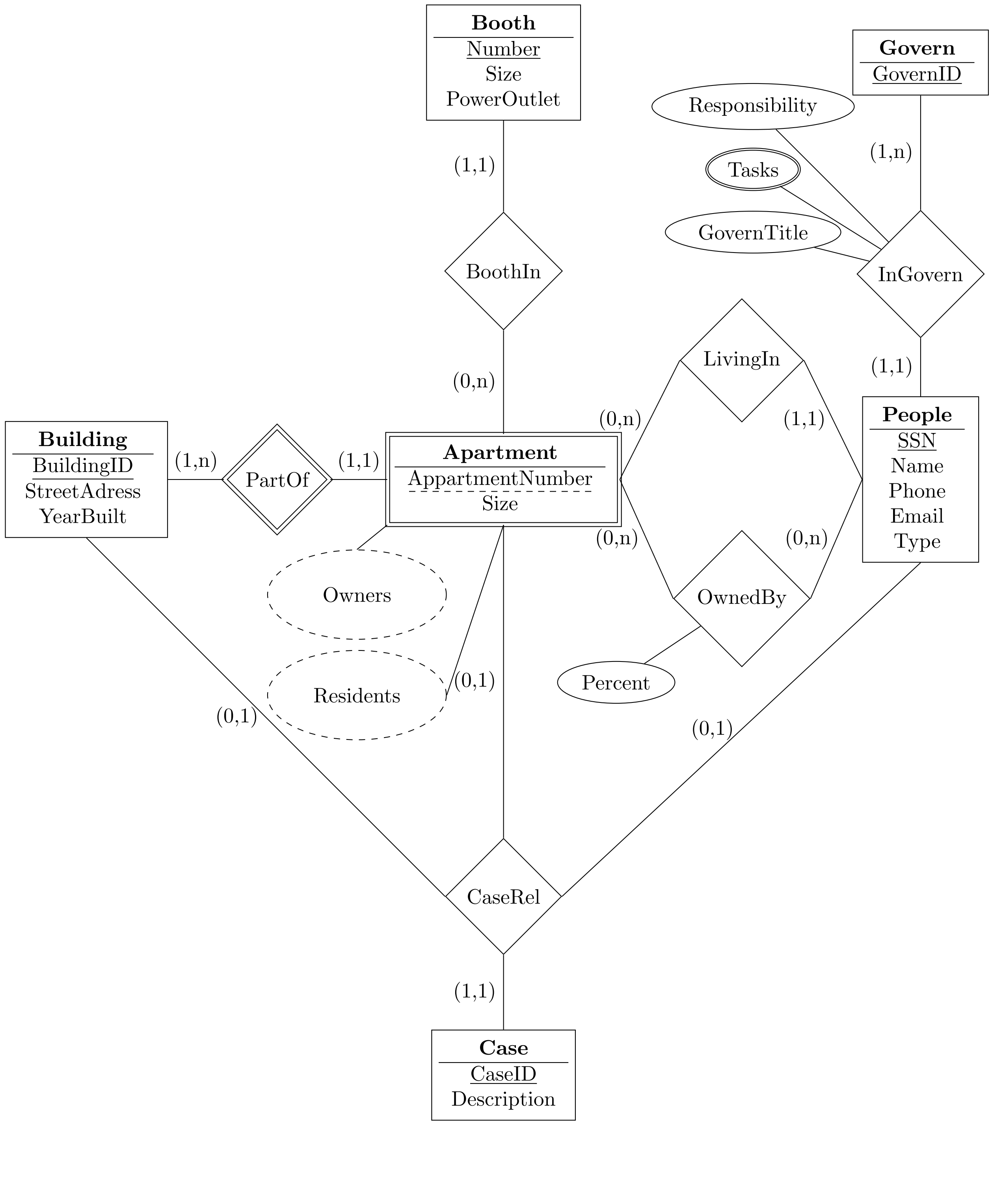


Figure 1: Task 3d)

3.4 e)


StartDate
Capacity
StartTime
Title
ScreenRoomID
MovieID
Length
AvailableForRent
ShowtimeID
Movie
(0,n)
Approved
(0,n)
ScreenRoom
Showtime
(0,n)
At
(1,1)
(1,1)
(0,n)
TotalRevenue
ScreenAtCinema
SoldTicket
(1,n)
(1,1)
Name
Cinema
Ticket
Price
CinemaID
PhoneNumber
PurchaseTime
SeatNumber

4 From a mini-world to an ER model


Booth
Number
Size
PowerOutlet
(1,1)
BoothIn
(0,n)
Govern
GovernID
Responsibility
Tasks
GovernTitle
(1,n)
InGovern
(1,1)
LivingIn
(0,n)
(1,1)
People
SSN
Name
Phone
Email
Type
Building
BuildingID
StreetAdress
YearBuilt
(1,n)
PartOf
(1,1)
Apartment
AppartmentNumber
Size
(0,n)
(0,n)
OwnedBy
Percent
Owners
Residents
(0,1)
(0,1)
(0,1)
CaseRel
(1,1)
Case
CaseID
Description