INFO20003 Database Systems

Week 3

Q1) ER Review – fundamental concepts

- Entity, weak entity
- Attribute
- Business rules to relationships

Entity, Weak Entity

- Entity: a real-world object or concept distinguishable from other objects
  - Can be **concrete**, such as “person” or “book”
  - Can be **abstract**, such as “currency” or “sporting match”

- Weak Entity: an entity that **cannot be uniquely identified** without referring to another entity
  - Only has **partial key/weak key → require other entities in order to exist in database**
  - E.g. a company insurance policy that insures an employee and any dependents
  - Has a mandatory one, identifying relationship with owner entity

Q1) ER Review – fundamental concepts

- Entity, weak entity
- Attribute
- Business rules to relationships

Attributes

- Attribute: The characteristics describing an entity/relationship

- E.g., for an “employee” entity, every employee will have their **name**, **date of birth** and **address** stored.

- Each attribute has a domain – a set of permitted value

Q1) ER Review – fundamental concepts

- Entity, weak entity
- Attribute
- Business rules to relationships

Business Rules to Relationships

- Business rules are used to define:
  - Entities
  - Attributes
  - Relationships
  - Constraints

Relationships

- A relationship can be between **two or more entities**.

- Key constraints
- Participating constraints

Key Constraint

A restriction on the **maximum** number of associations that 1 entity can have in a particular relationship set

- One-to-one

An entity in A can have at most one association with an entity in B and vice versa

- One-to-many

An entity in A can have association with many entities in B.

However, an entity in B is only associated with at most one entity in A

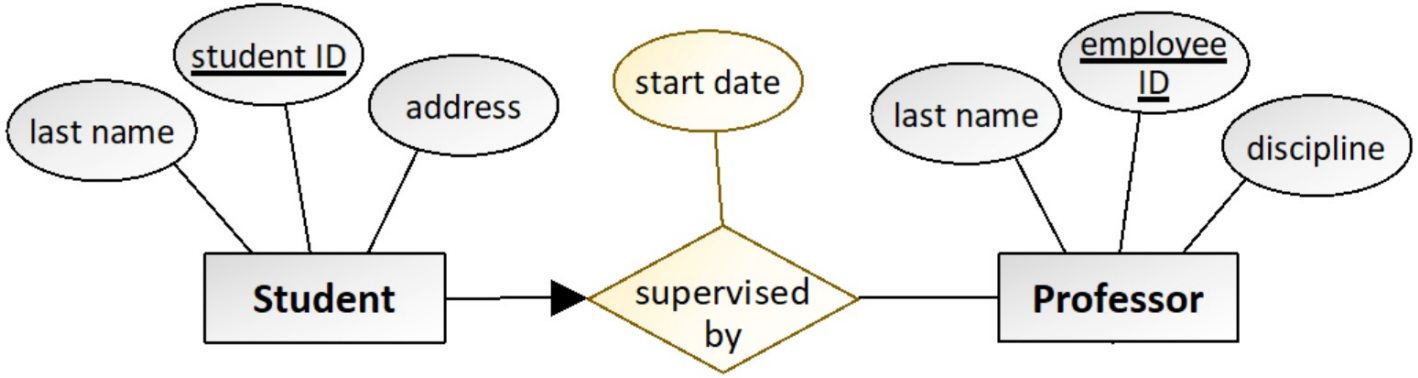
- Many-to-many

An entity in A can have association with many entities in B and vice versa

Key constraint - example

- "A student is allowed to be supervised by at most one professor, but the professor on the other hand can supervise more than one student"

- Key constraint?
  - One-to-many
  - One: student
  - Many: professor



Relationships

- A relationship can be between **two or more entities**.

- Key constraints
- Participating constraints

Participation constraint

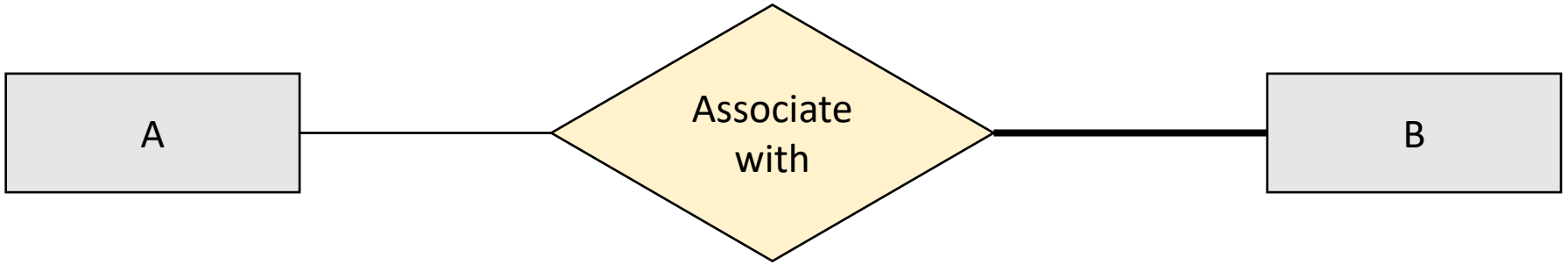
A restriction on the **minimum** number of associations that 1 entity should have in a particular relationship set

- Total Participation

Every entity in the entity set MUST take part in the relationship

- Partial Participation

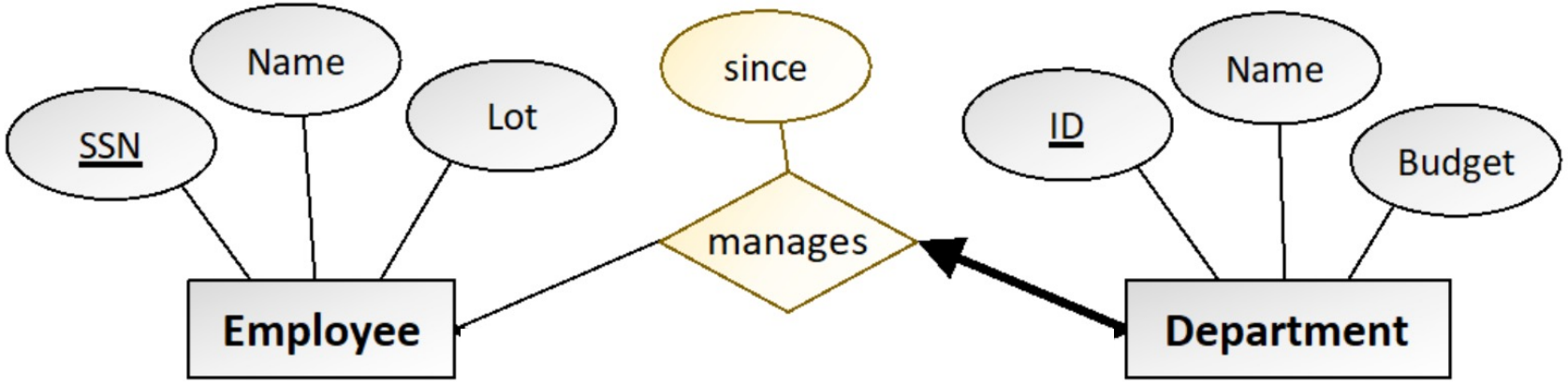
Not compulsory for every entity to take part in the relationship



A: partial participation
B: total participation (bold line)

Participation constraint - example

- “All departments must have a manager, however not every employee manages a department”



Q2)

2. Consider the following case study:

A cinema chain operates a number of cinemas. Each cinema has several screens, numbered starting from 1. The chain keeps track of the size (in feet) and seating capacity of every screen, as well as whether the screen offers the Gold Class experience.

The cinema chain owns hundreds of movie projectors – both film projectors (16 mm and 35 mm) and digital projectors (2D and 3D). The chain stores key information about each projector, namely its serial number, model number, resolution and hours of use. Each movie screen has space for a single projector; technicians must be able to identify which screen each projector is currently projecting onto.

A wide range of movies are shown at these cinemas. The system should keep track of the last time a movie was shown on a particular screen. The marketing department needs to know the movie's title and year of release, along with the movie's rating (G, PG, M, MA15+ or R18+).

Each cinema has a numeric ID, name and address. For cinemas that are not owned outright, the business also keeps track of yearly rent. The system needs to be able to generate weekly activity reports for the chain's chief operating officer.

Follow the steps to create a conceptual model in Chen's notation:

a. Revise last week's identified **entities**.
b. Form **relationships** between entities.
c. Apply **constraints** (key constraints and participation constraints) to the relationships.
d. Add **attributes** which describe the entities and relationships.
e. Finalise your conceptual model by marking **weak entities**, **identifying relationships** and **key attributes**.

Q2a) Entities

- Cinema
- Screen
- Projector
- Movie

"Cinema chain" is **not** an entity in this scenario. You do not normally include the actual business or company whose business processes you are modelling. This is because there is only one instance of this company, and there is no data to store about it in any case.

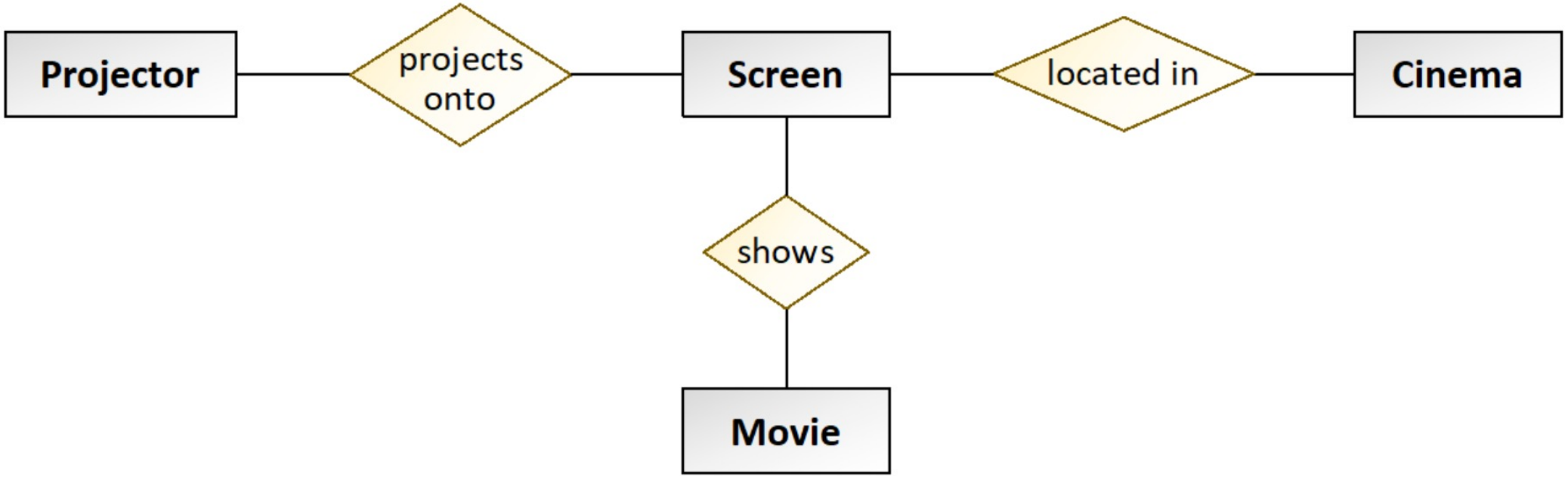
Q2b) Relationships

- “Each cinema has several screens”
  - screen is **located in** cinema (or cinema **contains** screen)

- “Technicians must be able to identify which movie screen each projector is currently projecting onto”
  - projector **projects onto** movie screen (or movie screen is **projected onto by** projector)

- “The system should keep track of the last time a movie was shown on a particular screen”
  - screen **shows** movie (or movie is **shown on** screen)

Q2b) Relationships

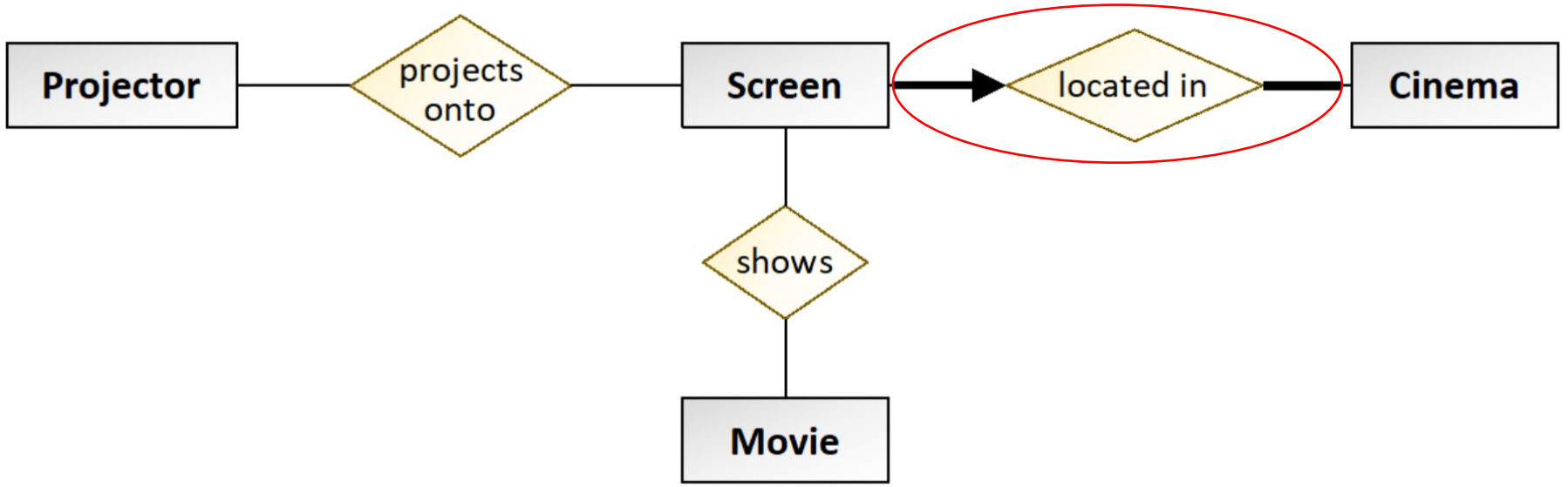


Avoid using vague words like “has” to label your relationships.

Q2c) Constraints

"Located in" relationship

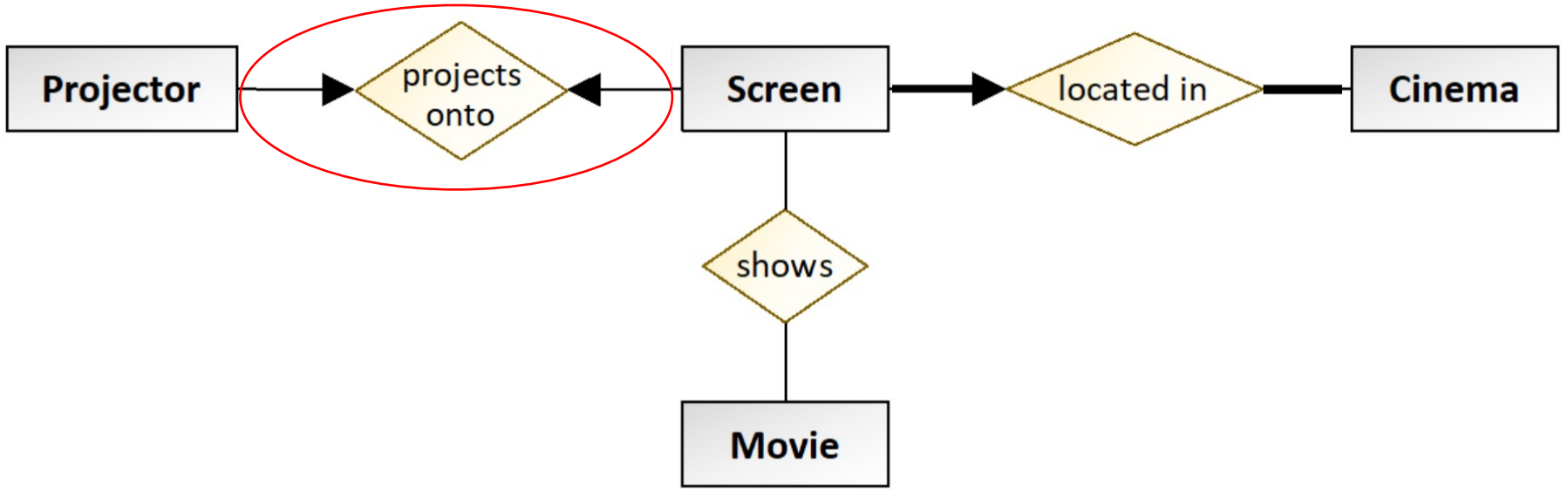
- A screen must be located in exactly one cinema.
  - screen is "mandatory one", drawn as a bold arrow
- A cinema must contain at least one screen.
  - cinema is "mandatory many", drawn as a bold line



Q2c) Constraints

"Projects onto" relationship

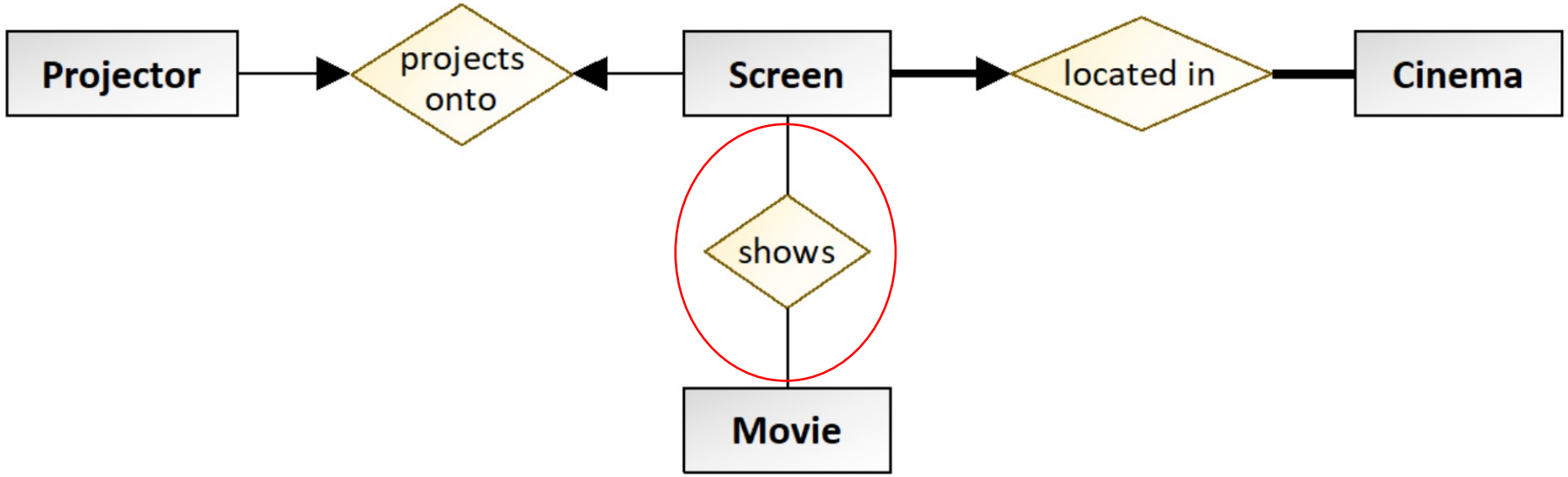
- A projector may project onto exactly one screen.
  - projector is "optional one", drawn as a thin arrow
- A screen may be projected onto by exactly one projector.
  - screen is "optional one", drawn as a thin arrow



Q2c) Constraints

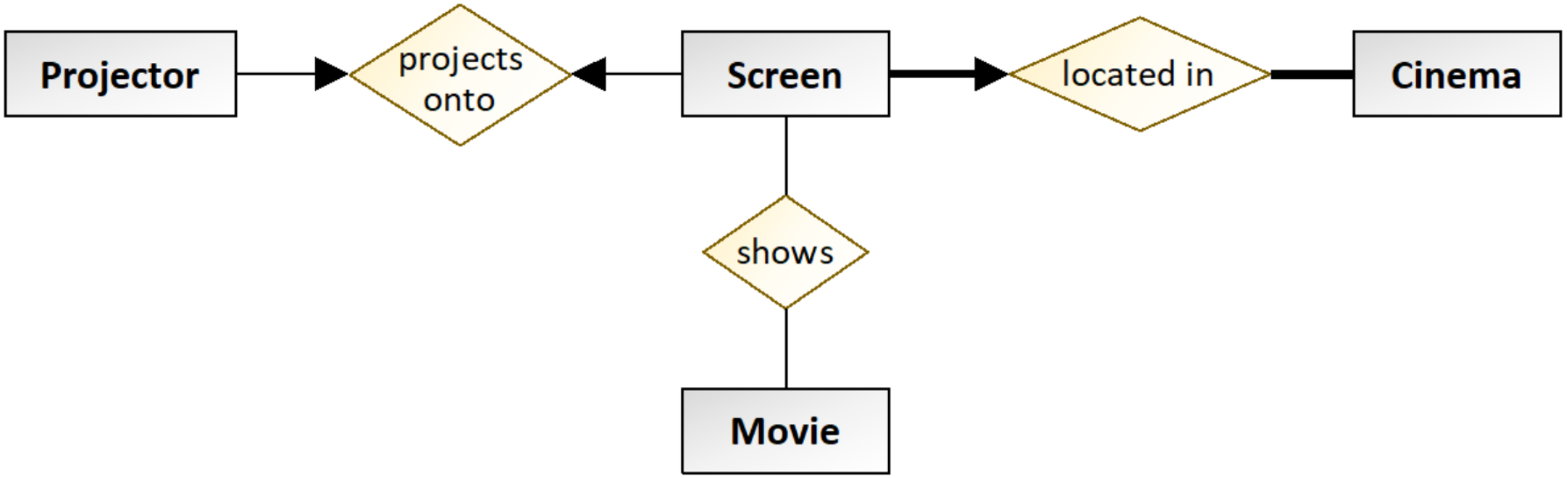
"Shows" relationship

- A screen may show many movies (over time).
  - screen is "optional many", drawn as a thin line
- A movie may be shown on many screens (over time).
  - movie is "optional many", drawn as a thin line



Q2c) Constraints

Not all constraints are spelt out in the requirements analysis. You need to use common sense to fill in the gaps.



Q2d) Attributes

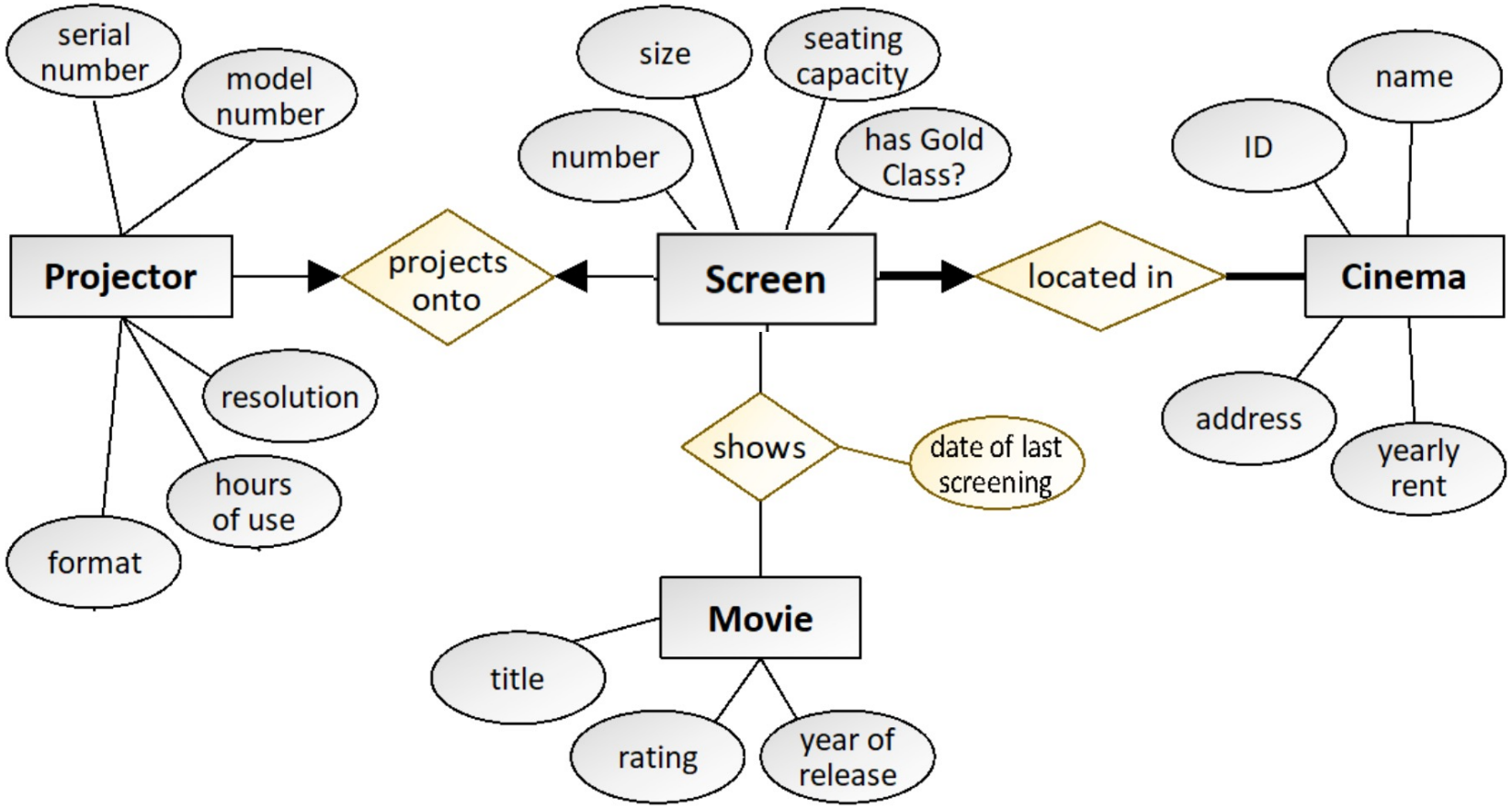
Attributes for entity:

- Cinema (ID, name, address, yearly rent)
- Screen (number, size, seating capacity, has Gold Class?)
- Projector (format [16 mm film/35 mm film/2D digital/3D digital], serial number, model number, resolution, hours of use)
- Movie (title, year of release, rating)

Attribute for relationship set:

- "The system should keep track of the last time a movie was shown on a particular screen"
- → "Shows" relationship (date of last screening)

Q2d) Attributes



Q2e) Weak entities & Identifying Relationships

- Weak entities must have “mandatory one” relationship with another entity
  - Bold arrow coming out of it

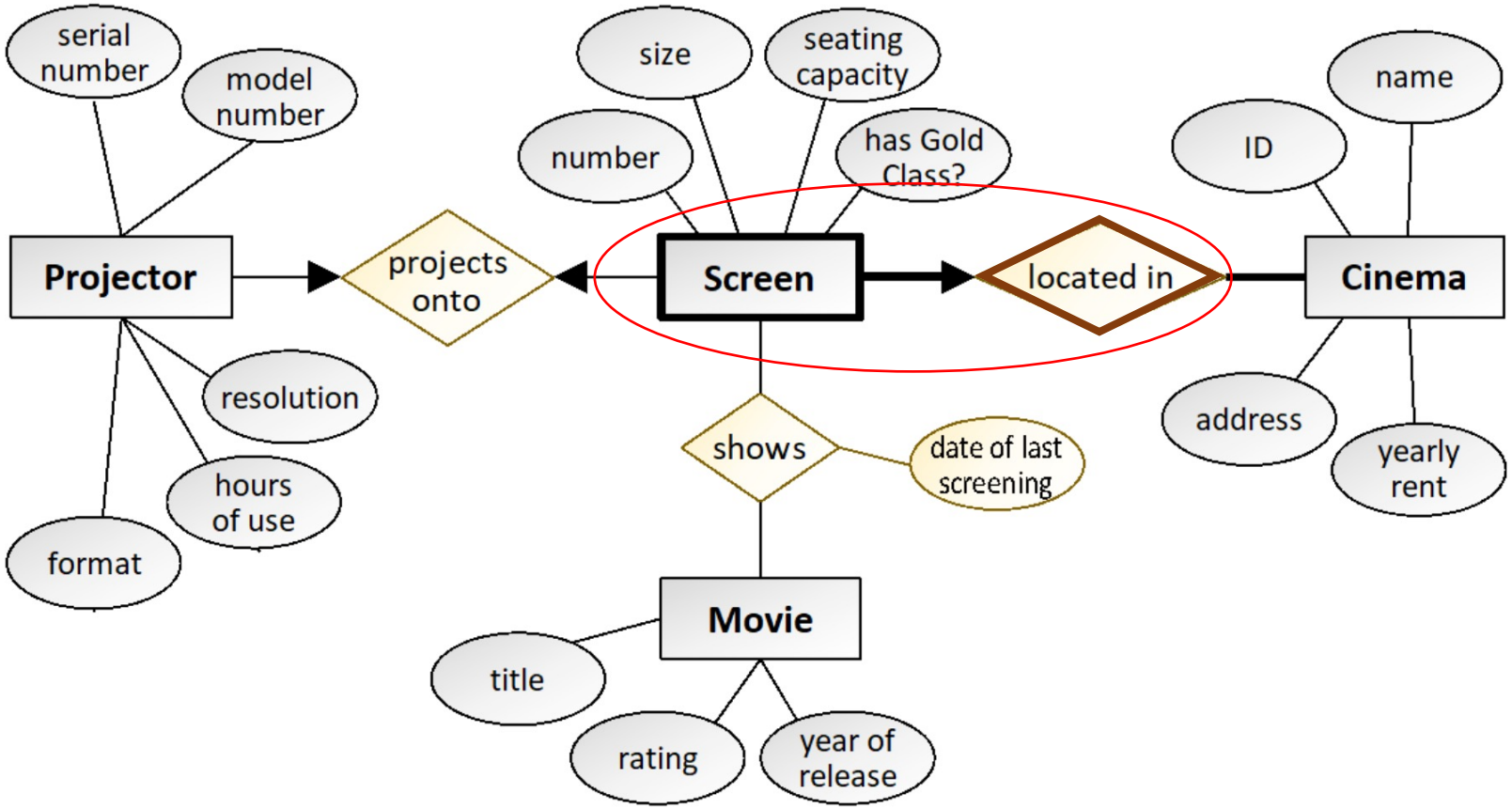
- Weak entity must have at least 1 identifying relationship
  - Bold diamond

Q2e) Weak entities & Identifying Relationships

- A screen only exists as long as the cinema it is located in continues to exist.
- Also, it can only be identified in terms of the cinema it is located in
  - E.g. “Melbourne central cinema, screen 1”

- Screen is a weak entity, and “located in” is its identifying relationship

Q2e) Weak entities & Identifying Relationship



Q2e) Key attributes

- Cinema (ID, name, address, yearly rent)
- Projector (format [16 mm film/35 mm film/2D digital/3D digital], serial number, model number, resolution, hours of use)

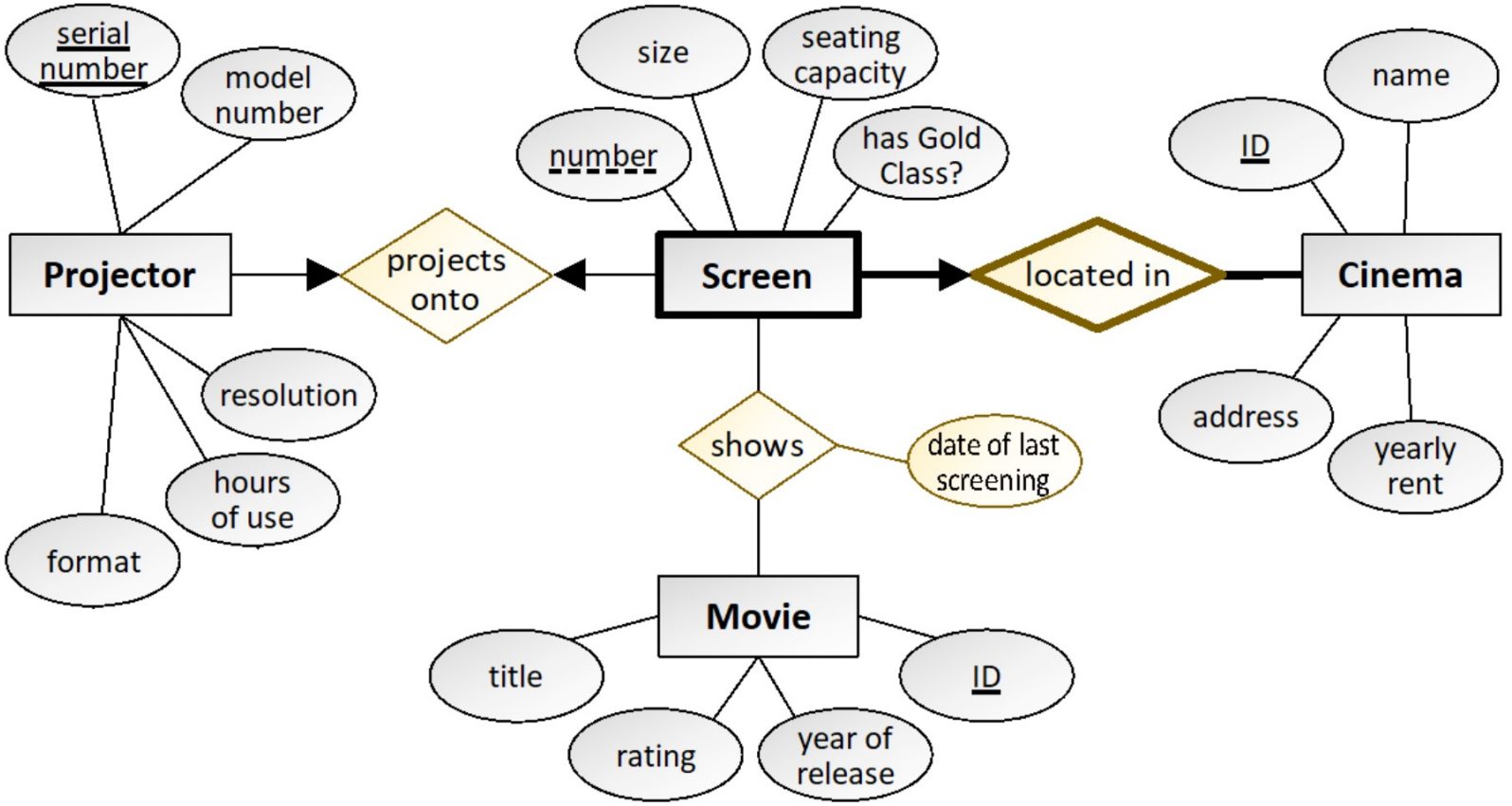
> A surrogate key is a unique, meaningless number used to identify each instance of an entity. Bank account numbers and fast food order numbers are common examples of surrogate keys.

- Movie (ID, title, year of release, rating)

> Partial keys are only relevant for weak entities. Every weak entity must have a partial key.

- Screen (number, size, seating capacity, has Gold Class?)

Q2)


serial number
model number
Projector
resolution
hours of use
format
projects onto
size
seating capacity
number
has Gold Class?
Screen
shows
date of last screening
Movie
title
rating
year of release
ID
located in
name
ID
Cinema
address
yearly rent

Q3a)

What needs to be changed to convert a conceptual design to a logical design? Develop a logical design for the above case study.

- **Resolve multi-valued attributes**
  - Flatten if there is a small finite number of values
  - Otherwise split into another table
- **Resolve composite attributes**
  - Redrawing the component parts as separate attributes
- **Resolve Relationships**

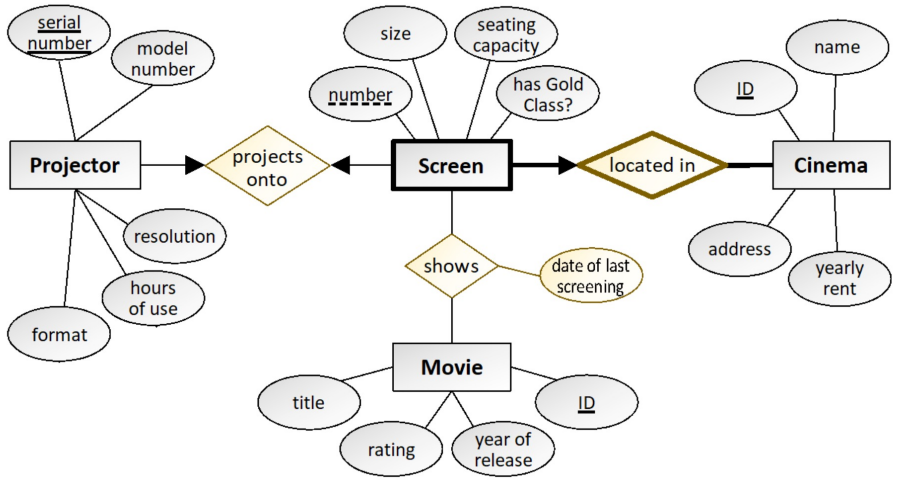
- Name changed to **CamelCase**

Q3a) Resolving Relationships

- **Resolve one-to-one relationships**
  - Add FK to either table
  - Preference to the total participation end to reduce NULL value

- **Resolve one-to-many relationships**
  - Add FK to the ‘one’ side

- **Resolve many-to-many relationships**
  - Create a new entity called associative entity

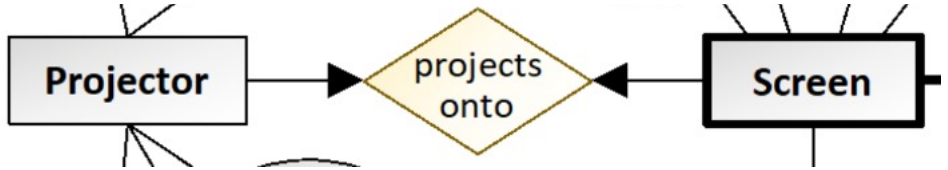


Q3a) Resolving Relationships

- **Resolve one-to-one relationships**
  - Add FK to either table
  - Preference to the total participation end to reduce NULL value

Q3a) Resolving relationships

- One-to-one relationship:



Screen (ScreenNumber, Size, SeatingCapacity, HasGoldClass, ProjectorSerialNumber)
FK

Or you can add FK to the Projector table
(but Screen PK is {ScreenNumber, CinemaID}, see next slide)

Q3a) Resolving Relationships

- **Resolve one-to-one relationships**
  - Add FK to either table
  - Preference to the total participation end to reduce NULL value

- **Resolve one-to-many relationships**
  - Add FK to the ‘one’ side

Q3a) Resolving relationships

- One-to-many relationship:

To resolve the Cinema-Screen relationship, we add a foreign key on the Screen table. Because this is an **identifying relationship**, this foreign key, "cinema ID", will also be a primary key (known as a "primary foreign key"):

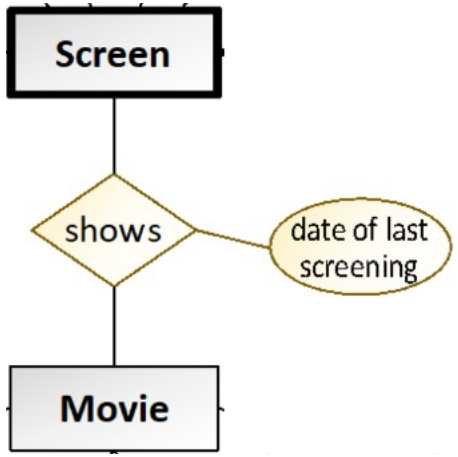
FK FK
Screen (CinemaID, ScreenNumber, Size, SeatingCapacity, HasGoldClass, ProjectorSerialNumber)

Q3a) Resolving Relationships

- **Resolve one-to-one relationships**
  - Add FK to either table
  - Preference to the total participation end to reduce NULL value
- **Resolve one-to-many relationships**
  - Add FK to the ‘one’ side
- **Resolve many-to-many relationships**
  - Create a new entity called associative entity

Q3a) Resolving relationships

- Many-to-many relationship:



FK FK FK
MovieScreening (CinemaID, ScreenNumber, MovieID, DateOfLastScreening)

Q3a)

Cinema (CinemaID, Name, Address, YearlyRent)

Screen (CinemaID (FK), ScreenNumber, Size, SeatingCapacity, HasGoldClass, ProjectorSerialNumber (FK))

MovieScreening (CinemaID (FK), ScreenNumber (FK), MovieID (FK), DateOfLastScreening)

Movie (MovieID, Title, YearOfRelease, Rating)

Projector (SerialNumber, Format, ModelNumber, Resolution, HoursOfUse)

Q3b)

- What will you change in the logical model to generate a physical model?

- Add Data types
- Add NULL/NOT NULL constraint (optional/mandatory)

Q3b) Data types

- **Foreign key columns** must have the same data type as the primary key column they refer to

- If it is **not clear** which data type to use, pick any data type that seems **reasonable**

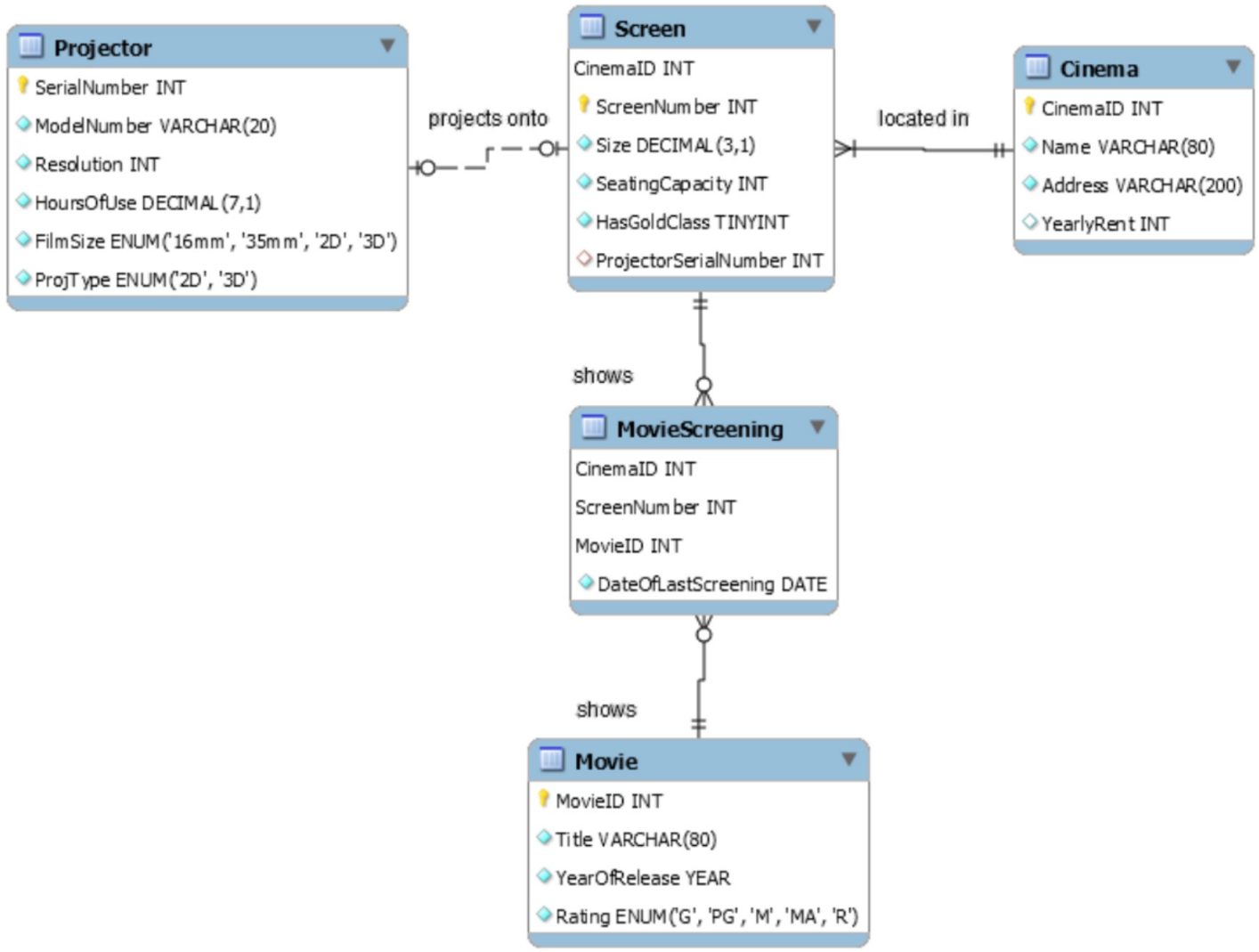
Q3b) NULL/NOT NULL

- **Primary key columns** must always be NOT NULL.

- For **foreign key columns**, look at the participation constraint on ER diagram
  - Mandatory → NOT NULL

- For **other columns**, think carefully about whether the data in that column is required or optional
  - E.g. “For cinemas that are not owned outright, the business also keeps track of yearly rent.” → NULL

Q3b)


Projector
SerialNumber INT
ModelNumber VARCHAR(20)
Resolution INT
HoursOfUse DECIMAL(7,1)
FilmSize ENUM('16mm', '35mm', '2D', '3D')
ProjType ENUM('2D', '3D')
projects onto
Screen
CinemaID INT
ScreenNumber INT
Size DECIMAL(3,1)
SeatingCapacity INT
HasGoldClass TINYINT
ProjectorSerialNumber INT
located in
Cinema
CinemaID INT
Name VARCHAR(80)
Address VARCHAR(200)
YearlyRent INT
shows
MovieScreening
CinemaID INT
ScreenNumber INT
MovieID INT
DateOfLastScreening DATE
shows
Movie
MovieID INT
Title VARCHAR(80)
YearOfRelease YEAR
Rating ENUM('G', 'PG', 'M', 'MA', 'R')