COMP 3111
SOFTWARE ENGINEERING

LECTURE 7
SYSTEM REQUIREMENTS CAPTURE
DOMAIN MODELING EXERCISE

EXERCISE: MOVIE SHOP DOMAIN MODEL

The following are the requirements for a web-based system to computerize the management of the sale and rental of movies for a movie shop.

- The system must be able to handle both physical and digital movies.
- It must be able to record which movies are sold and rented and by whom.
- For sold movies, the quantity sold should be recorded; for physical movie rental, which copy is rented and when it is due back should be recorded.
- The system should keep track of overdue rentals of physical movies and send email notices to customers who have movies overdue.
- There will be a customer membership option for an annual fee, which will entitle a member to discounts (10%) on the sale and rental of movies.
- Members should be able to make reservations for physical movie rentals either in person at the shop, by telephone or via the Web.
- A member can reserve at most five physical movies at any one time, but there is no limit on how many physical movies a member or nonmember can rent at any one time.
- As an added feature, the shop would like to allow customers (either members or nonmembers) to input, via the Web, mini-reviews (up to 100 words) and a rating (from 1, lowest, to 10, highest) of movies they have purchased or rented.

EXERCISE: MOVIE SHOP DOMAIN MODEL (CONT'D)

- These reviews should be anonymous if the customer so wishes (i.e., customers can specify whether they want their name to be made known when other customers browse the reviews).
- A sales clerk should be able to enter and update the following information about all customers (members or nonmembers): name, address, phone number, age, sex, and email address.
- Members are assigned a membership number by the shop when they become members and a password, which allows them to change their personal information and to buy and rent digital movies via the Web.
- The shop manager should be able to generate various reports on the sale and rental of movies.
- A sales clerk should be able to sell and rent physical movies and process the return of rented physical movies.
- When selling or renting physical movies, a sales clerk must be able to look up customer information and determine whether the customer is a member.
- A sales clerk must be able to enter basic information about a movie (i.e., movie id, title, leading actor(s), director, producer, genre, synopsis, release year, running time, selling price, and rental price).

EXERCISE: MOVIE SHOP DOMAIN MODEL (CONT'D)

From the movie sales and rental shop requirements statement:

a) identify all the classes, attributes, association classes, associations, aggregations/compositions, generalizations and multiplicity constraints that are relevant to include in the domain model for the new system. *(Only those that are explicitly given in or implied by the requirements statement should be included.)*

b) Construct a class diagram showing how the classes identified in (a) are related by associations, aggregations/ compositions and generalizations. Show the *most likely multiplicities for all associations*, making reasonable assumptions where necessary. If a multiplicity cannot be inferred from the requirements statement or common real-world domain knowledge, then indicate this with a "?". *Do not show the attributes of the classes in the class diagram.*

EXERCISE:
MOVIE SHOP DOMAIN MODEL ANALYSIS

We first analyze the requirements statement to determine the requirements for the domain model and then present the domain model.

- The system must be able to handle both physical and digital movies.

  classes: Movie

  attributes: Movie: isPhysical, isDigital

- It must be able to record which movies are sold and rented and by whom.

  classes & associations: Customer *Purchases* Movie
  Customer *Rents* Movie

- For sold movies, the quantity sold should be recorded; for physical movie rental, which copy is rented and when it is due back should be recorded.

  classes & associations: Movie *Has* RentalCopy
  Customer *RentsPhysical* RentalCopy

  attributes: Purchases: quantity
  RentalCopy: copyNumber, returnDate

EXERCISE:
MOVIE SHOP DOMAIN MODEL ANALYSIS

- The system should keep track of overdue rentals of physical movies and send email notices to customers who have movies overdue.

  attributes: Customer: email

- There will be a customer membership option for an annual fee, which will entitle a member to discounts (10%) on the sale and rental of movies.

  generalization: Member *is a kind of* Customer
  → Customer *Generalizes* Member

- Members should be able to make reservations for physical movie rentals either in person at the shop, by telephone or via the Web.

  classes & associations: Member *Reserves* RentalCopy

- A member can reserve at most five physical movies at any one time, but there is no limit on how many physical movies a member or nonmember can rent at any one time.

  constraint: max-card(Member, *Reserves*) = 5
  max-card(Customer, *RentsPhysical*) = *

EXERCISE:
MOVIE SHOP DOMAIN MODEL ANALYSIS

- As an added feature, the shop would like to allow customers (either members or nonmembers) to input, via the Web, mini-reviews (up to 100 words) and a rating (from 1, lowest, to 10, highest) of movies they have purchased or rented.

  classes & associations: Customer *Provides* Review *IsFor* Movie

  → Customer *Provides* Review
  Review *IsFor* Movie

  attributes: Review: reviewText, rating

- These reviews should be anonymous if the customer so wishes (i.e., customers can specify whether they want their name to be made known when other customers browse the reviews).

  attributes: Review: anonymous

- A sales clerk should be able to enter and update the following information about all customers (members or nonmembers): name, address, phone number, age, sex, and email address.

  attributes: Customer: name, address, phoneNumber, age, sex, email

EXERCISE:
MOVIE SHOP DOMAIN MODEL ANALYSIS

- Members are assigned a membership number by the shop when they become members and a password, which allows them to change their personal information and to buy and rent digital movies via the Web.

  attributes: Member: memberNumber, password

- The shop manager should be able to generate various reports on the sale and rental of movies.

  functional requirement: no new domain model requirements

- A sales clerk should be able to sell and rent physical movies and process the return of rented physical movies.

  functional requirement: no new domain model requirements

- When selling or renting physical movies, a sales clerk must be able to look up customer information and determine whether the customer is a member.

  functional requirement: no new domain model requirements

EXERCISE:
MOVIE SHOP DOMAIN MODEL ANALYSIS

- A sales clerk must be able to enter basic information about a movie (i.e., movie id, title, leading actor(s), director, producer, genre, synopsis, release year, running time, selling price, and rental price).

  attributes: Movie: movieId, title, leadingActor[0..*], director, producer, genre, synopsis, releaseYear, runningTime, sellingPrice, rentalPrice

EXERCISE:
MOVIE SHOP DOMAIN MODEL ANALYSIS

Classes and Associations

Customer *Purchases* Movie

Customer *Rents* Movie

Movie *Has* RentalCopy

Customer *RentsPhysical* RentalCopy

Customer *Provides* Review

Review *IsFor* Movie

Member *Reserves* RentalCopy

Association Classes

Purchases: quantity

Generalizations

Customer *Generalizes* Member

Constraints

max-card(Member, *Reserves*) = 5

max-card(Customer, *RentsPhysical*) = *

Attributes

Customer: name, address, phoneNumber, age, sex, email

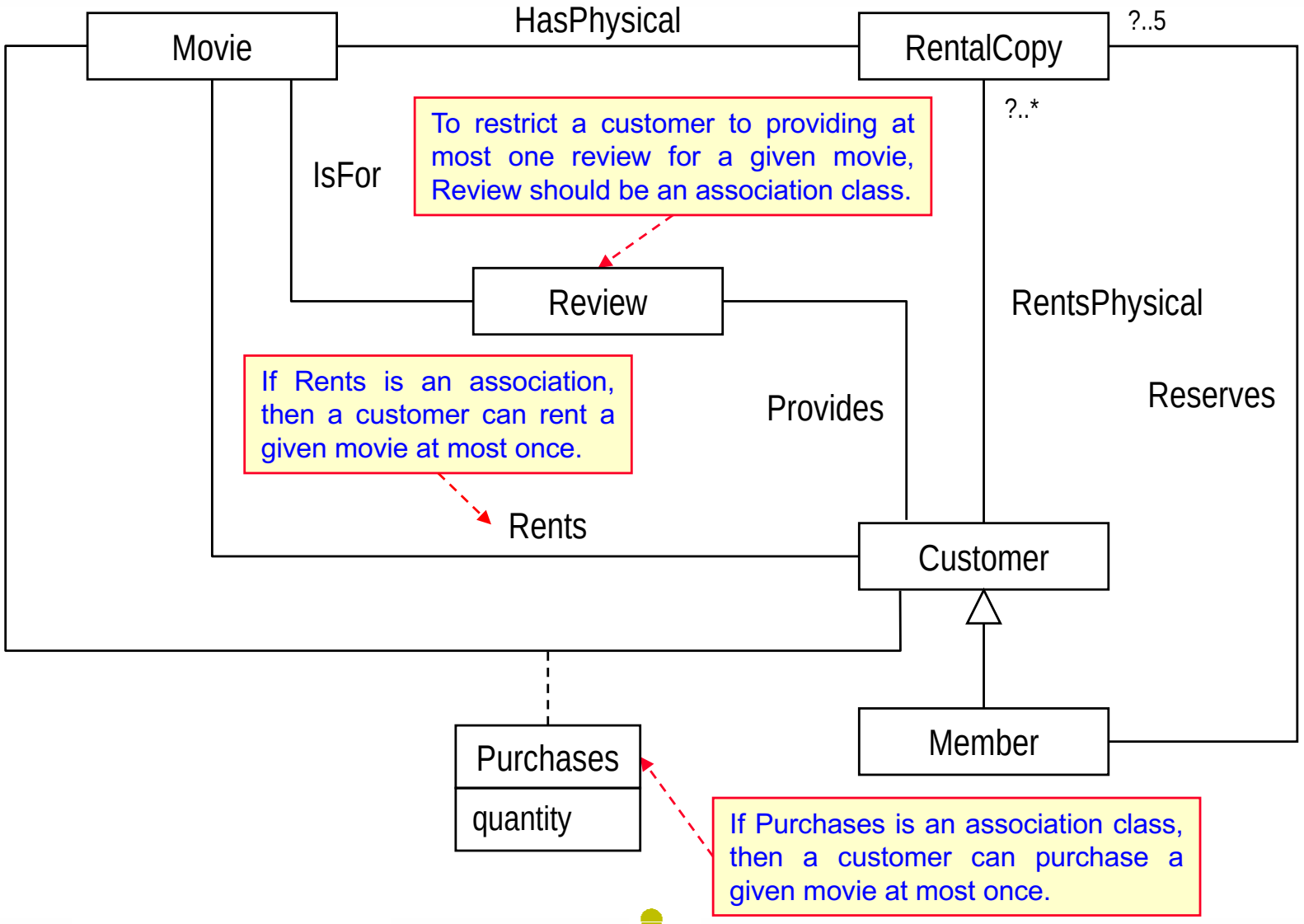
Member: memberNumber, password

Movie: movieId, title, leadingActor[0..*], director, producer, genre, synopsis, releaseYear, runningTime, sellingPrice, rentalPrice, isPhysical, isDigital

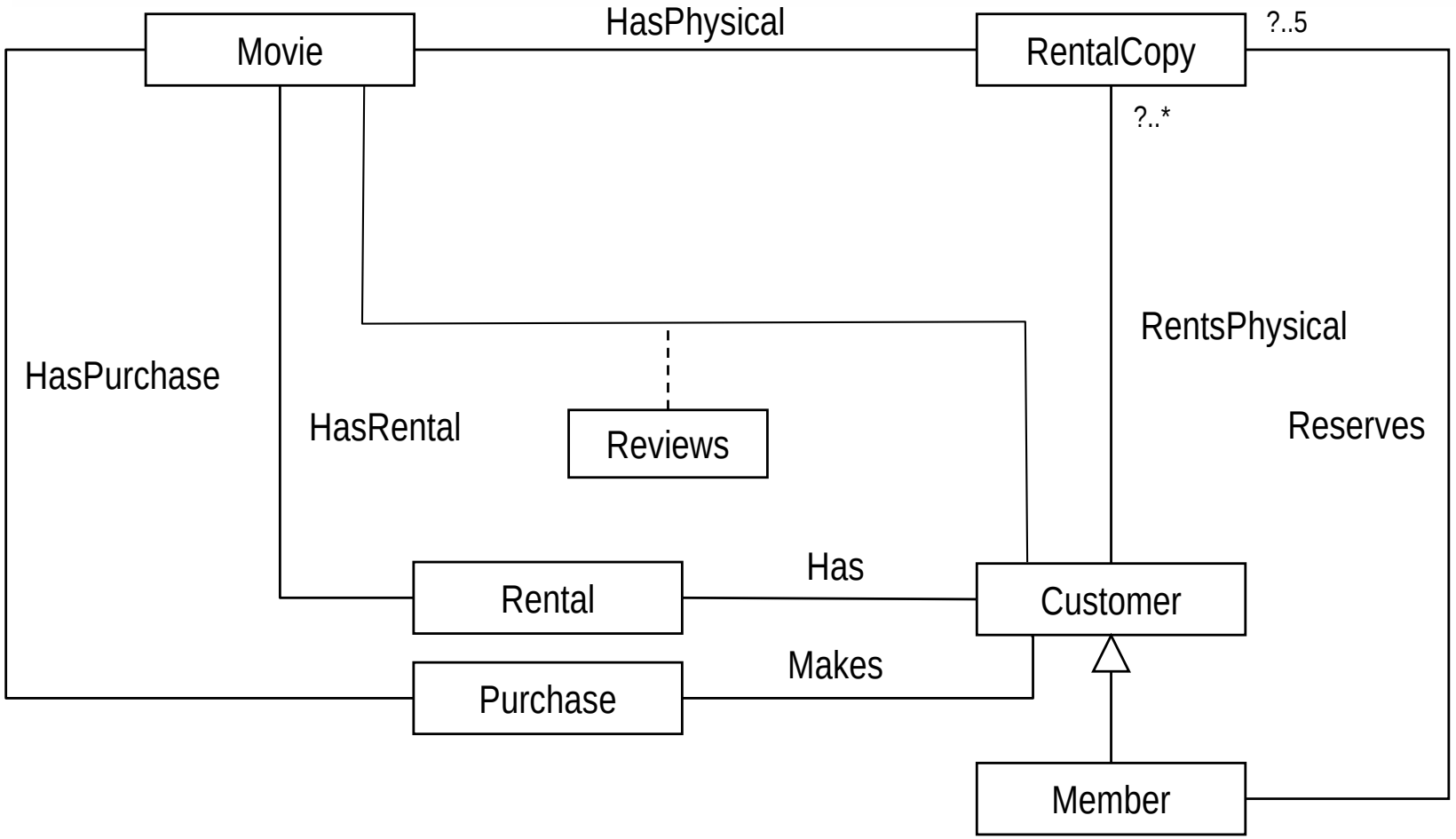
RentalCopy: copyNumber, returnDate

Review: reviewText, rating, anonymous

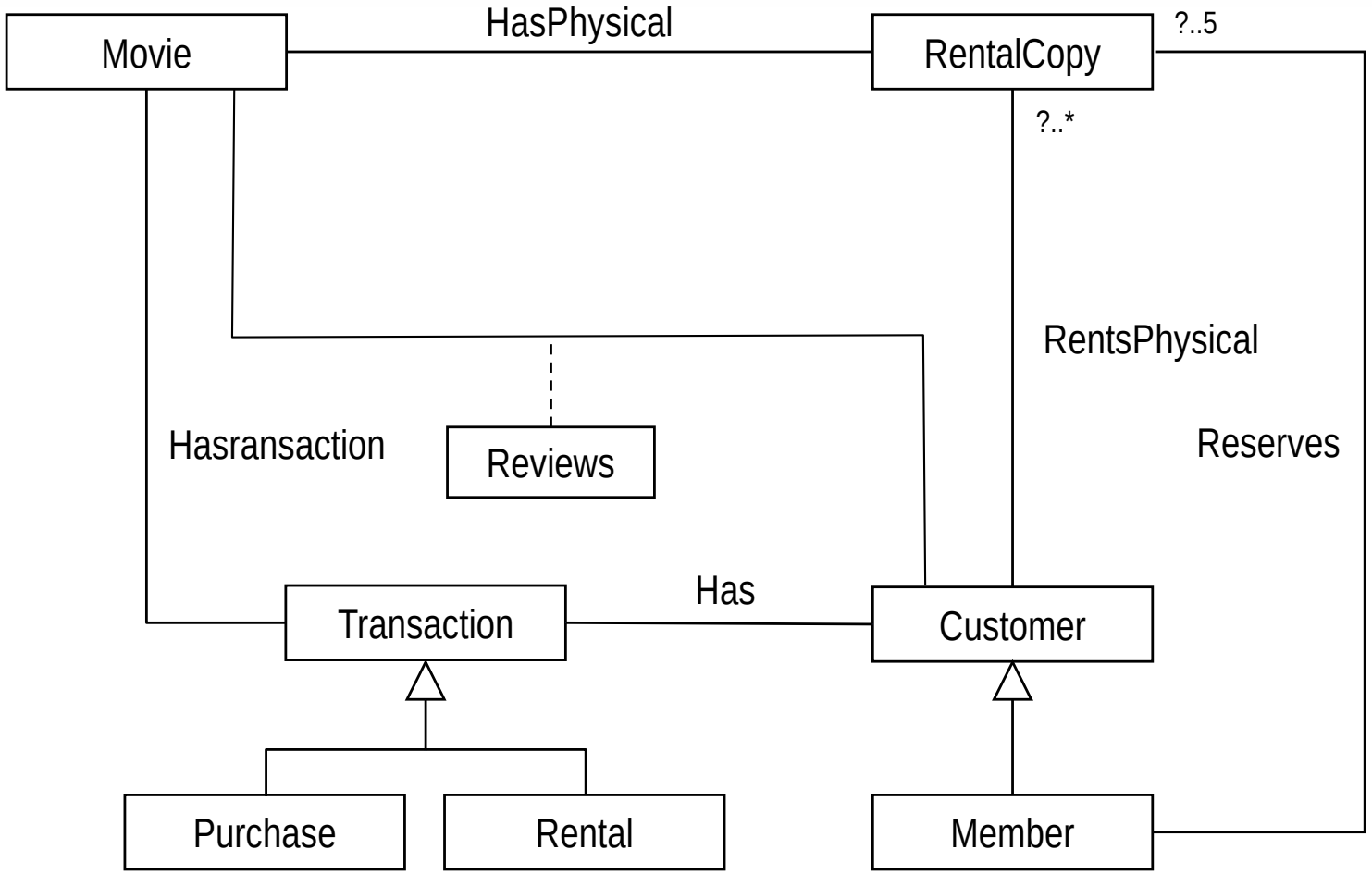
EXERCISE:
MOVIE SHOP DOMAIN MODEL SOLUTION



EXERCISE:
MOVIE SHOP DOMAIN MODEL SOLUTION



EXERCISE:
MOVIE SHOP DOMAIN MODEL SOLUTION



EXERCISE:
MOVIE SHOP DOMAIN MODEL ANALYSIS

Constraints from Real World Knowledge

A member may not have reserved any physical rental copy.

min-card(Member, *Reserves*) = 0

A physical rental copy may not currently be reserved by any member, but it can be reserved by at most one member at a time.

min-card(RentalCopy, *Reserves*) = 0 max-card(RentalCopy, *Reserves*) = 1

A customer may not currently have rented any physical rental copies.

min-card(Customer, *RentsPhysical*) = 0

A physical rental copy may not currently be rented by any customer, but it can be rented by at most one customer at a time.

min-card(RentalCopy, *RentsPhysical*) = 0 max-card(RentalCopy, *RentsPhysical*) = 1

A movie can have no or it can have many physical rental copies.

min-card(Movie, *HasPhysical*) = 0 max-card(Movie, *HasPhysical*) = *

A physical rental copy is a copy of exactly one movie.

min-card(RentalCopy, *HasPhysical*) = 1 max-card(RentalCopy, *HasPhysical*) = 1

EXERCISE:
MOVIE SHOP DOMAIN MODEL ANALYSIS

A movie can have no or many reviews.

min-card(Movie, *Reviews*) = 0 max-card(Movie, *Reviews*) = *

A customer can provide no or many reviews.

min-card(Customer, *Reviews*) = 0 max-card(Customer, *Reviews*) = *

A customer can have no or many rentals.

min-card(Customer, *Has*) = 0 max-card(Customer, *Has*) = *

Each rental is for exactly one customer.

min-card(Rental, *Has*) = 1 max-card(Rental, *Has*) = 1

A movie can have no or many rentals.

min-card(Movie, *HasRental*) = 0 max-card(Movie, *HasRental*) = *

A rental is for exactly one movie.

min-card(Rental, *HasRental*) = 1 max-card(Rental, *HasRental*) = 1

A customer can make no or many purchases.

min-card(Customer, *Makes*) = 0 max-card(Customer, *Makes*) = *

EXERCISE:
MOVIE SHOP DOMAIN MODEL ANALYSIS

A purchase is made by exactly one customer.

min-card(Purchase, *Makes*) = 1 max-card(Purchase, *Makes*) = 1

A movie can have no or many purchases.

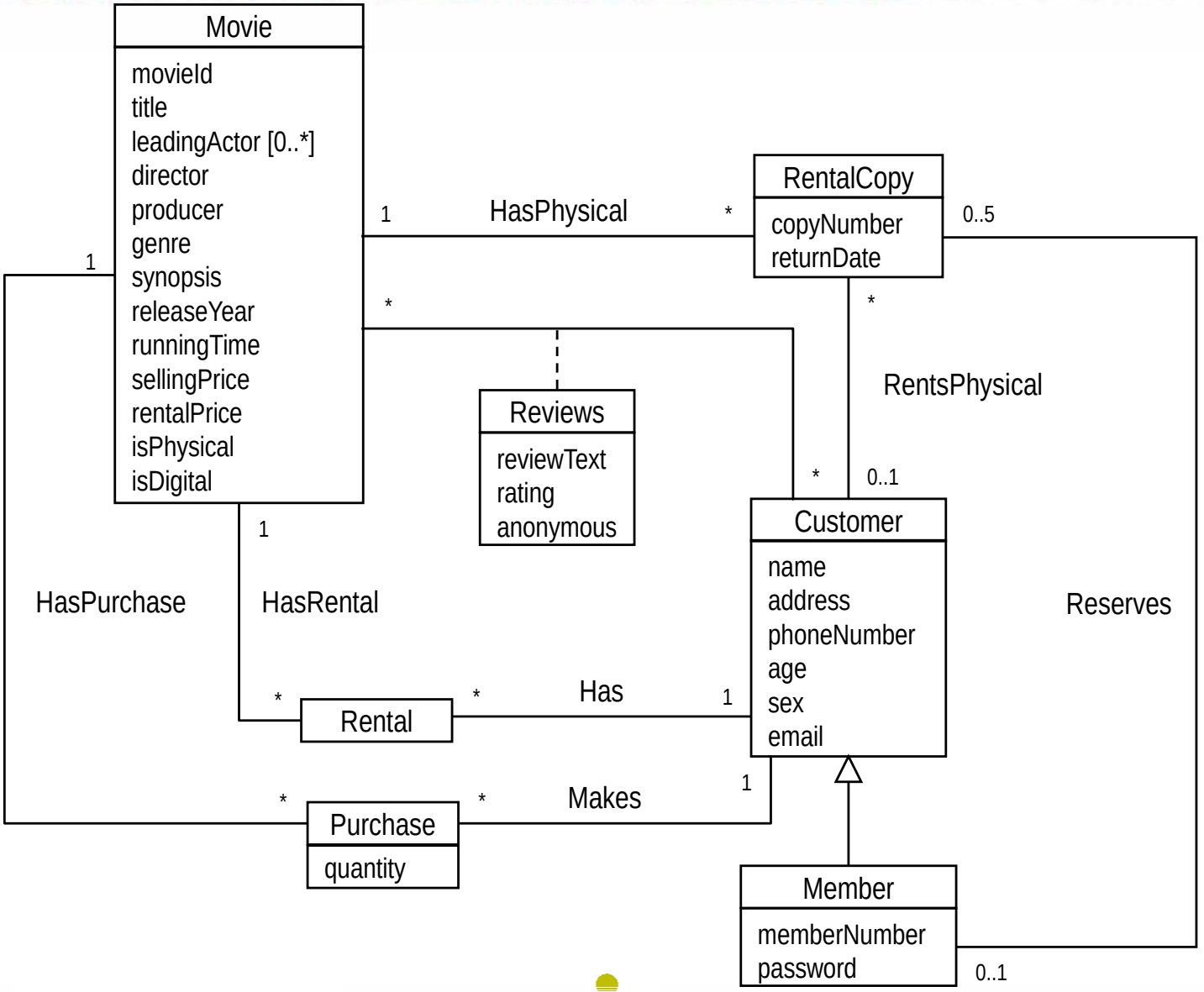
min-card(Movie, *HasPurchase*) = 0 max-card(Movie, *HasPurchase*) = *

Each purchase is for exactly one movie.

min-card(Purchase, *HasPurchase*) = 1 max-card(Purchase, *HasPurchase*) = 1

Remark: An instance of the class Movie *does not* represent an instance of a physical copy of the movie, but a description of the movie. Hence, the same instance of Movie (i.e., the description) can be related to many rental copies, rentals and purchases).

EXERCISE:
MOVIE SHOP DOMAIN MODEL SOLUTION



EXERCISE:
MOVIE SHOP DOMAIN MODEL COMMON ERRORS

- Using “ids” to relate classes
- Representing the client/organization (e.g., Shop)
- Incorrect use/overuse of generalization
  (e.g., Person, Nonmember, PhysicalMovie, DigitalMovie)
- Incorrect use/overuse of aggregation/composition
  (e.g., Movie<>——RentalCopy, Movie<>——Review,
  Customer<>——Member)
- Incorrect use of association class
- Incorrect constraints (e.g., XOR)

EXERCISE:
MOVIE SHOP DOMAIN MODEL COMMON ERRORS

- Representing operations
  (e.g., generates, browses, enters, looks up, etc.)
- Storing reports
- Representing implementation aspects
  (e.g., System, Web, telephone, reports)
- Over specifying the model
  (e.g., sales clerk, manager)

Key question: What information about things/ procedures needs to be persistent (i.e., in files or a database)?