

CSIT882 Data Management Systems

Entity-relationship model

Subject coordinator: Chen Chen

School of Computing and Information Technology - University of Wollongong

Disclaimer: subject materials are sourced from previous offerings of CSIT882

Entity-Relationship Data Modelling

In ER, the world is viewed as **a collection of inter-related “entities”**.

ER has **three** major modelling constructs:

- entity: objects (“things”) in your world that you are interested

Person, Restaurants, Books, University Courses,...

- attribute: data item describing a property of interest

Person (name, phone number, DOB, ...)

- relationship: association between entities (objects)

Person **dines-at** Restaurant

Entity-Relationship (ER) Diagrams

ER diagrams are a graphical tool for data modelling.

An ER diagram consists of:

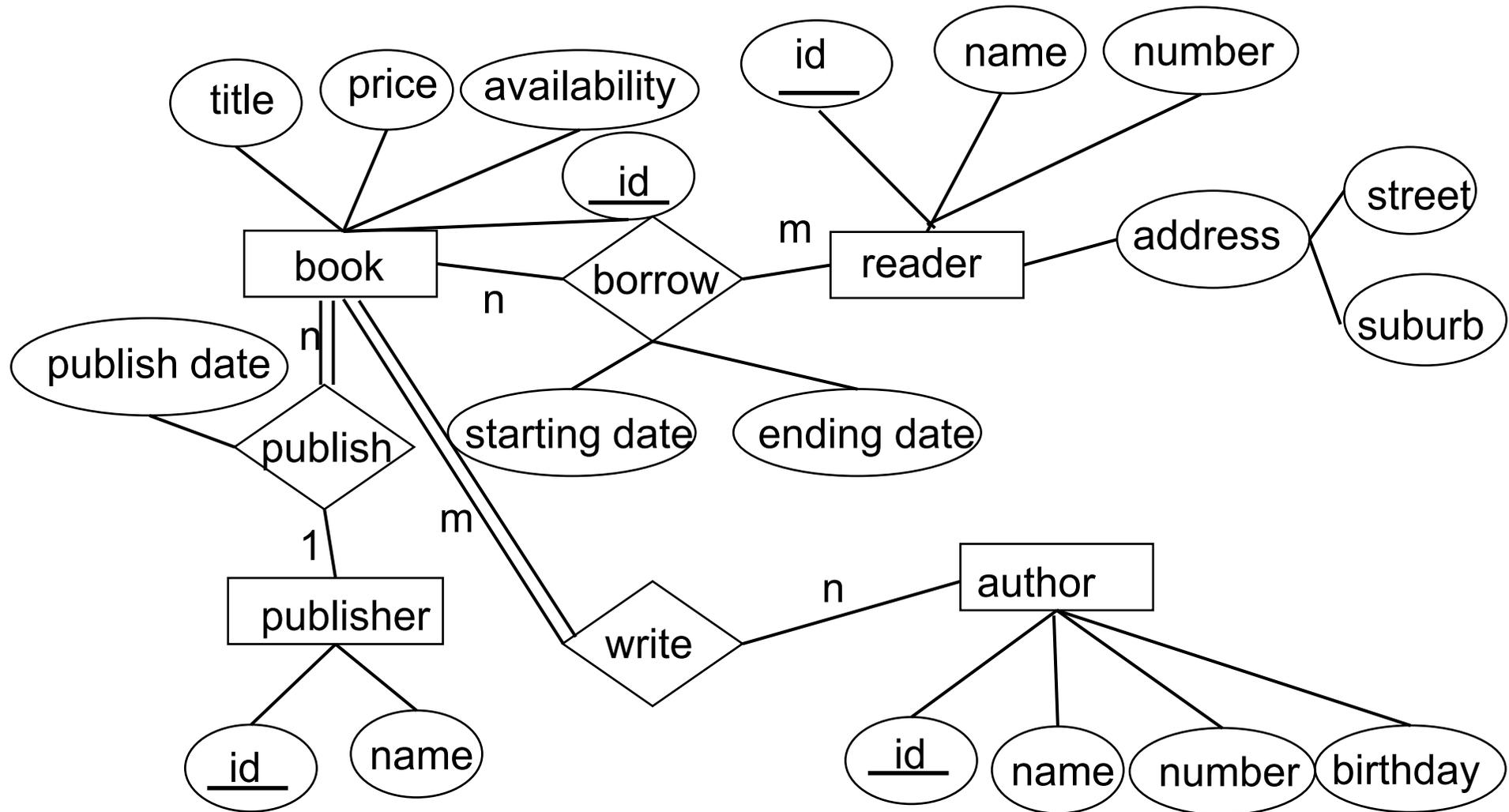
- a collection of entity set definitions
- a collection of relationship set definitions
- attributes associated with entity and relationship sets
- connections between entity and relationship sets

Terminology: when discussing "entity sets", we frequently say just "entity"

The ER model is not a standard, so many notation variations exist.

Entity-Relationships

Example ER Diagram: entities, attributes, relationships/connections



Entity

An **entity type** defines a collection of entities that have the same attributes.

- An entity type describes the **schema** for a *set of entities* that share the same structure.
- The collection of entities of a particular entity type is grouped into an **entity set**, which is also called the **extension** of the entity type.
- An entity type is represented in ER diagrams as a rectangular box enclosing the entity type name.

Entity Type Example

Entity Type Name:

EMPLOYEE

COMPANY

Name, Age, Salary

Name, Headquarters, President

Entity Set:
(Extension)

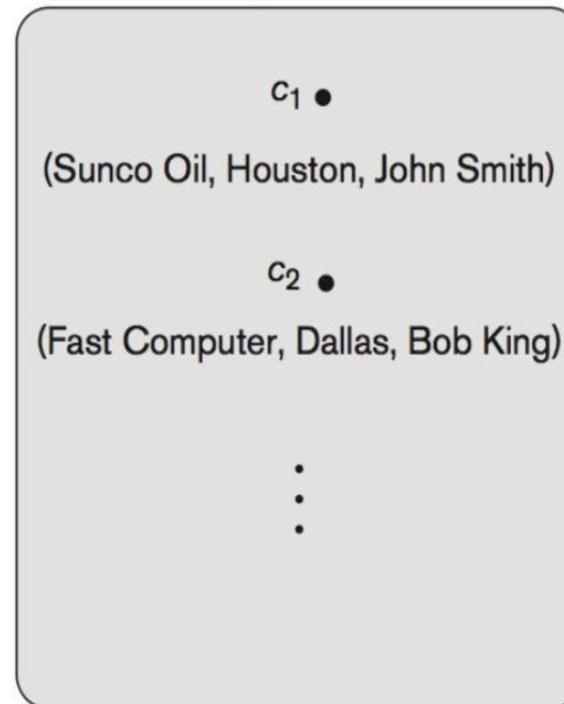
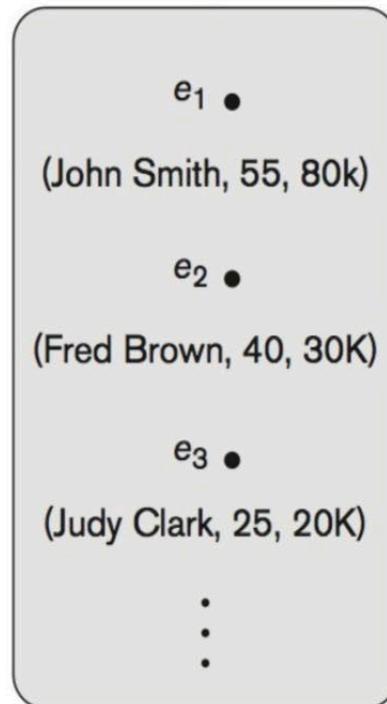
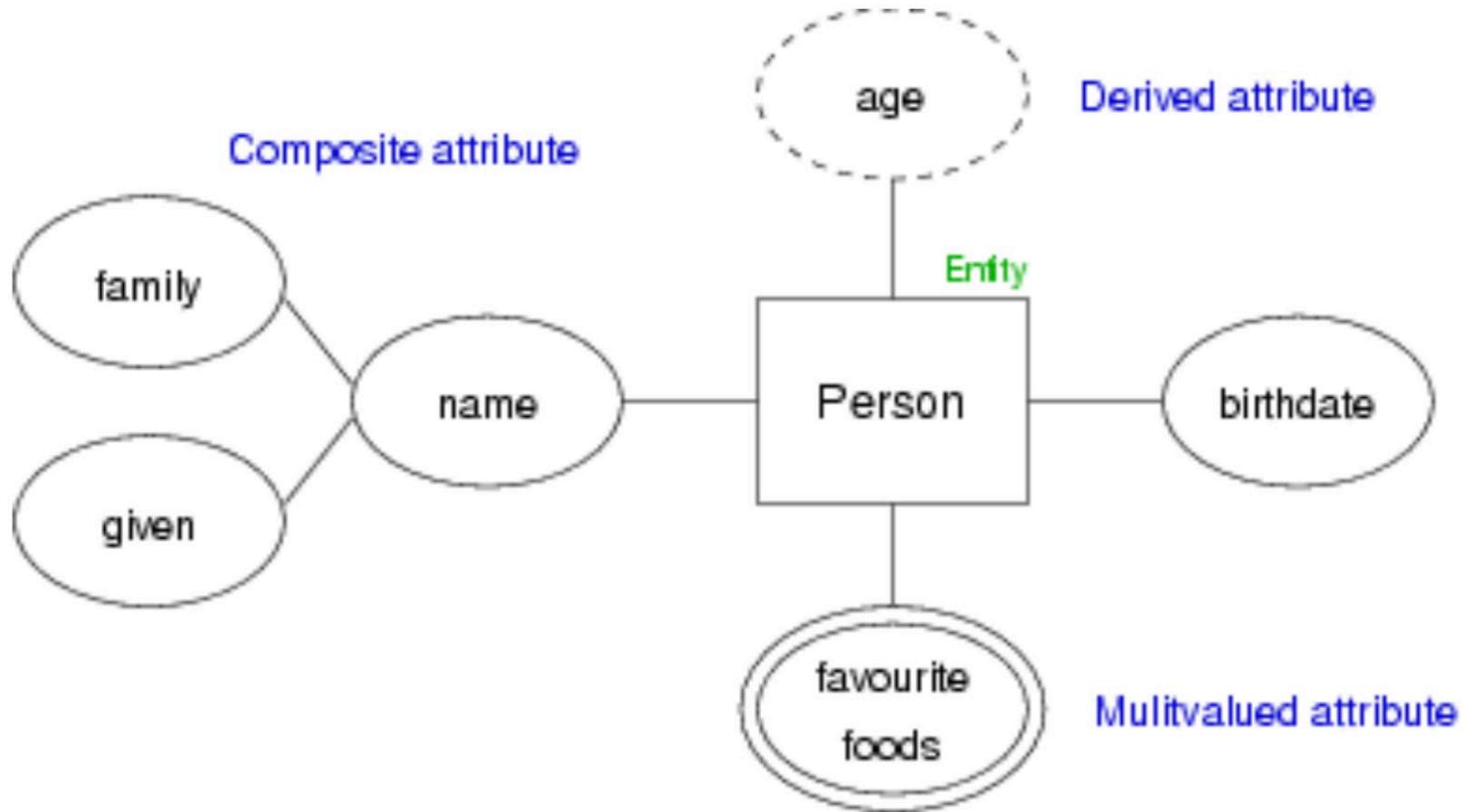


Figure 3.6

Two entity types, EMPLOYEE and COMPANY, and some member entities of each.

Attribute

Example of attribute notations



Attributes

Simple Attributes (or **Atomic** Attributes) are attributes that are not divisible.

- Each simple attribute of an entity type is associated with a value set (or domain of values), which specifies the set of values that may be assigned to that attribute for each individual entity.
- e.g., Entity = Student, Attributes = Student number, name...

Composite attributes can be divided into smaller subparts, which represent more basic attributes with independent meanings.

- Some semantics cannot be captured using atomic attributes

Question

Question: is *Address* a simple attribute or a composite attribute?

- *Address = 'SCIT building (No.3) Northfields Ave Wollongong, NSW 2522 Australia'*

How should we model *Address*?

Question

Question: is *Address* a simple attribute or a composite attribute?

- *Address* = 'SCIT building (No.3) Northfields Ave Wollongong, NSW 2522 Australia'

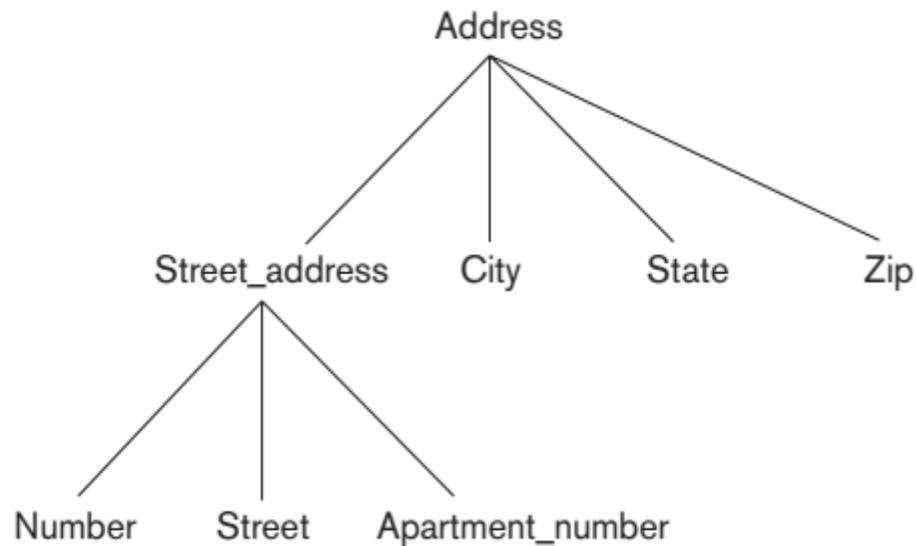


Figure 7.4

A hierarchy of composite attributes.

Question

Composite attributes are useful for situations when

- The end-user sometimes refers to the composite attribute as a unit,
- But at other times refers specifically to its components.

Question: Can't I just let my composite attributes, be simple attributes instead?

Question

Composite attributes are useful for situations when

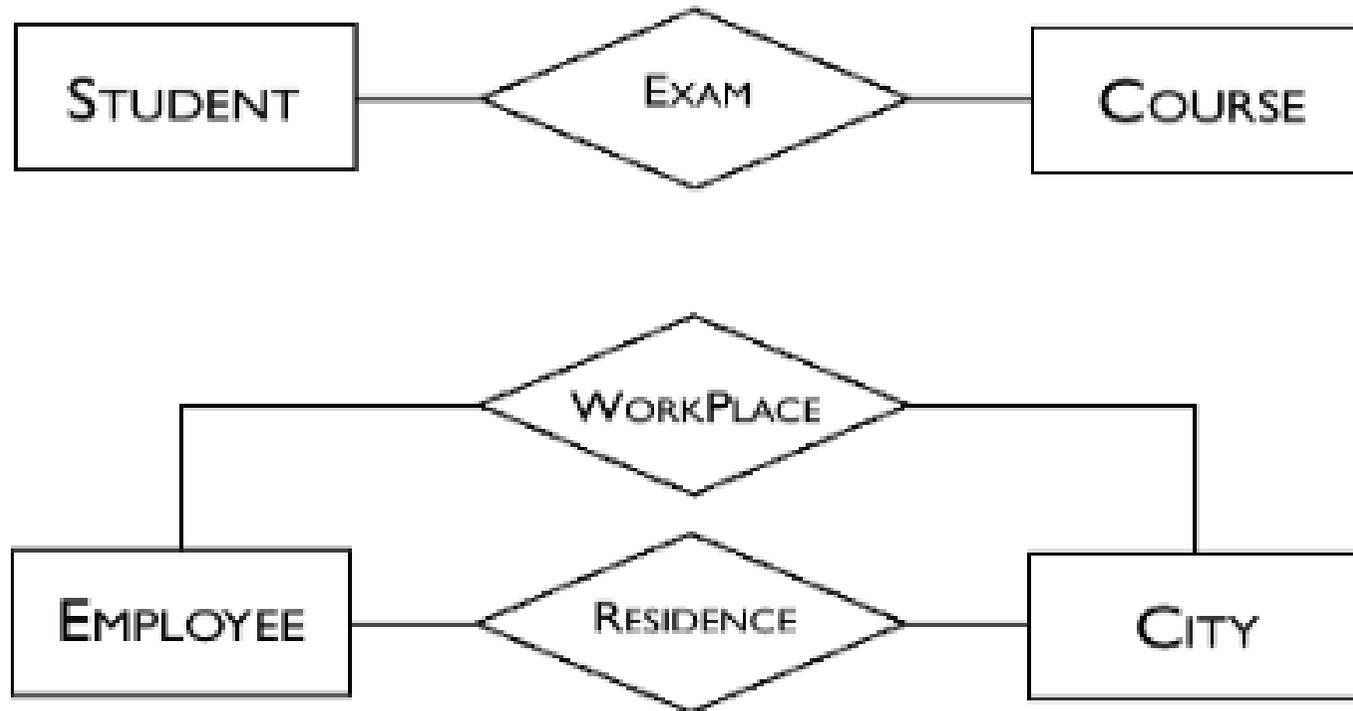
- The end-user sometimes refers to the composite attribute as a unit,
- But at other times refers specifically to its components.

Question: Can't I just let my composite attributes, be simple attributes instead?

Answer: If the composite attribute is referenced only as a whole, there is no need to subdivide it into component attributes.

Relationship

Another Big Component of ER: They represent logical links between two or more entities.



Visualising an ER Data Model

Notations:



Entity Type



Exercise: A Library Database

- A book is uniquely identified by its book-id. For each book, we also record its title, price, and availability.
- A reader is uniquely identified by his/her reader-id and we also record his/her name, phone number and address. The address is composed of street and suburb.
- A publisher is uniquely identified by its publisher-id. For each publisher, the name is also recorded.
- An author is uniquely identified by his/her author-id. For each author, the name, phone number and birth date are also recorded.
- A reader can borrow zero or more books and a book can be borrowed by zero or more readers. Thus, we need to record the starting date and ending date for the borrowing relationship.
- A publisher can publish zero or more books and a book is published by exactly one publisher. We also need to record the date of publication.
- An author can write zero or more books and a book is written by one or more authors

Exercise: A Library Database

- A **book** is uniquely identified by its *book-id*. For each book, we also record its *title*, *price*, and *availability*.
- A **reader** is uniquely identified by his/her *reader-id* and we also record his/her *name*, *phone-number* and *address*. The address is composed of *street* and *suburb*.
- A **publisher** is uniquely identified by its *publisher-id*. For each publisher, the *name* is also recorded.
- An **author** is uniquely identified by his/her *author-id*. For each author, the *name*, *phone-number* and *birth-date* are also recorded.
- A **reader** can borrow zero or more books and a book can be borrowed by zero or more readers. Thus, we need to record the *starting-date* and *ending-date* for the borrowing relationship.
- A **publisher** can publish zero or more books and a book is published by exactly one publisher. We also need to record the *date-of-publication*.
- An **author** can write zero or more books and a book is written by one or more authors.

Exercise

Entity focused

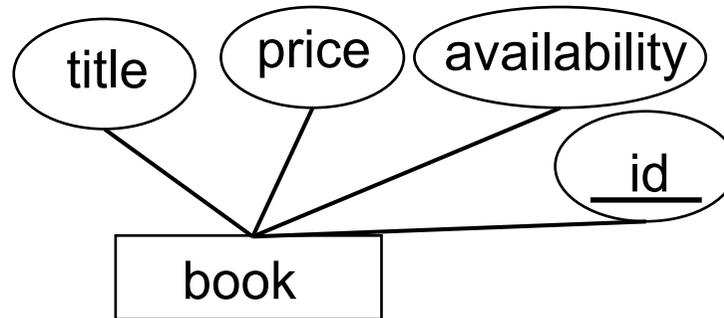
- A **book** is uniquely identified by its *book-id*. For each book, we also record its *title*, *price*, and *availability*.
- A **reader** is uniquely identified by his/her *reader-id* and we also record his/her *name*, *phone-number* and *address*. The address is composed of *street* and *suburb*.
- A **publisher** is uniquely identified by its *publisher-id*. For each publisher, the *name* is also recorded.
- An **author** is uniquely identified by his/her *author-id*. For each author, the *name*, *phone-number* and *birth-date* are also recorded.

Relation focused

- A **reader** can borrow zero or more books and a book can be borrowed by zero or more readers. Thus, we need to record the *starting-date* and *ending-date* for the borrowing relationship.
- A **publisher** can publish zero or more books and a book is published by exactly one publisher. We also need to record the *date-of-publication*.
- An **author** can write zero or more books and a book is written by one or more authors.

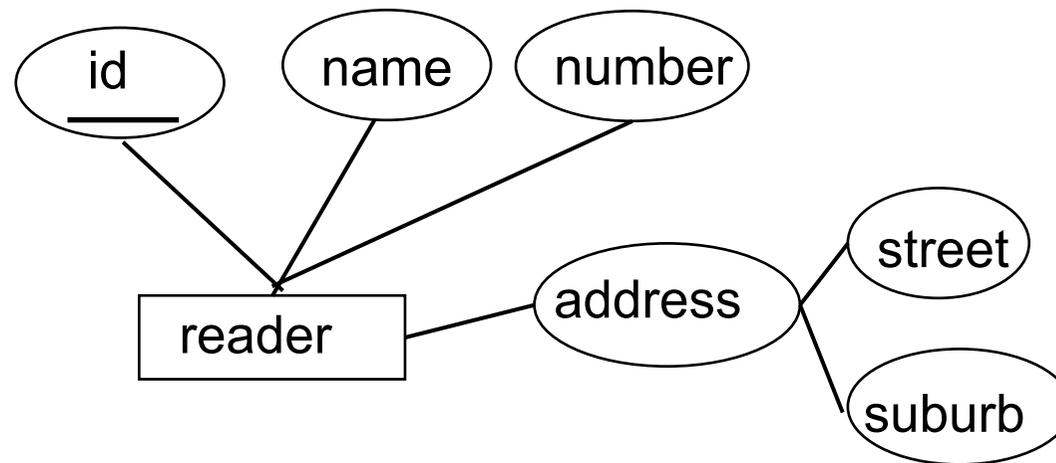
Example (1/7)

A book is uniquely identified by its book id. For each book, we also record its title, price, and availability.



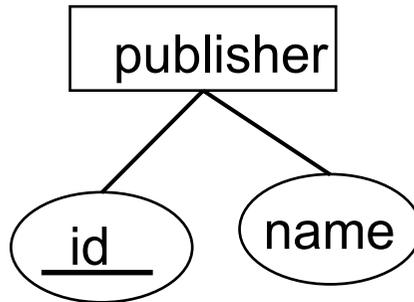
Example (2/7)

A reader is uniquely identified by his/her reader id and we also record his/her name, phone number, dob and address. The address is composed of street and suburb.



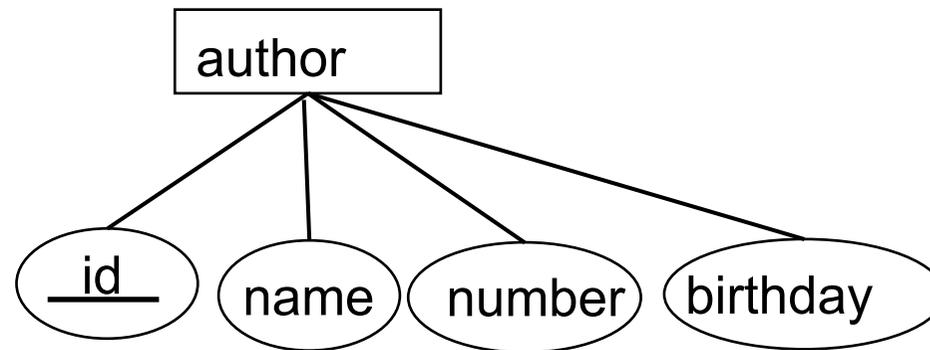
Example (3/7)

A publisher is uniquely identified by its publisher id. For each publisher, the name is also recorded.



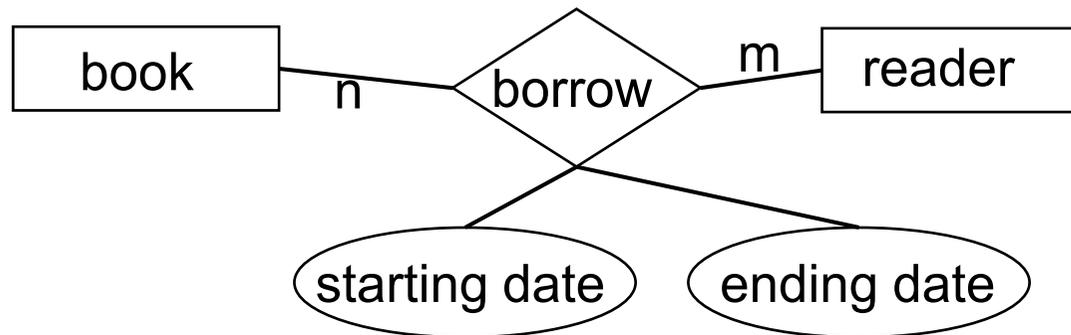
Example (4/7)

An author is uniquely identified by his/her author id. For each author, the name, phone number and birth date are also recorded.



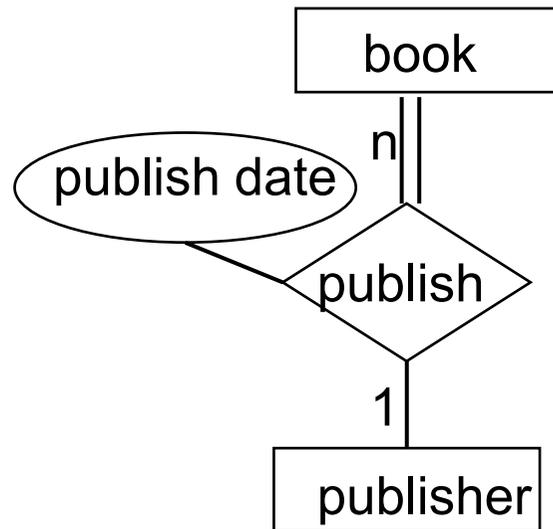
Example (5/7)

A reader can borrow zero or more books and a book can be borrowed by zero or more readers. Thus, we need to record the starting date and ending date for the borrowing relationship.



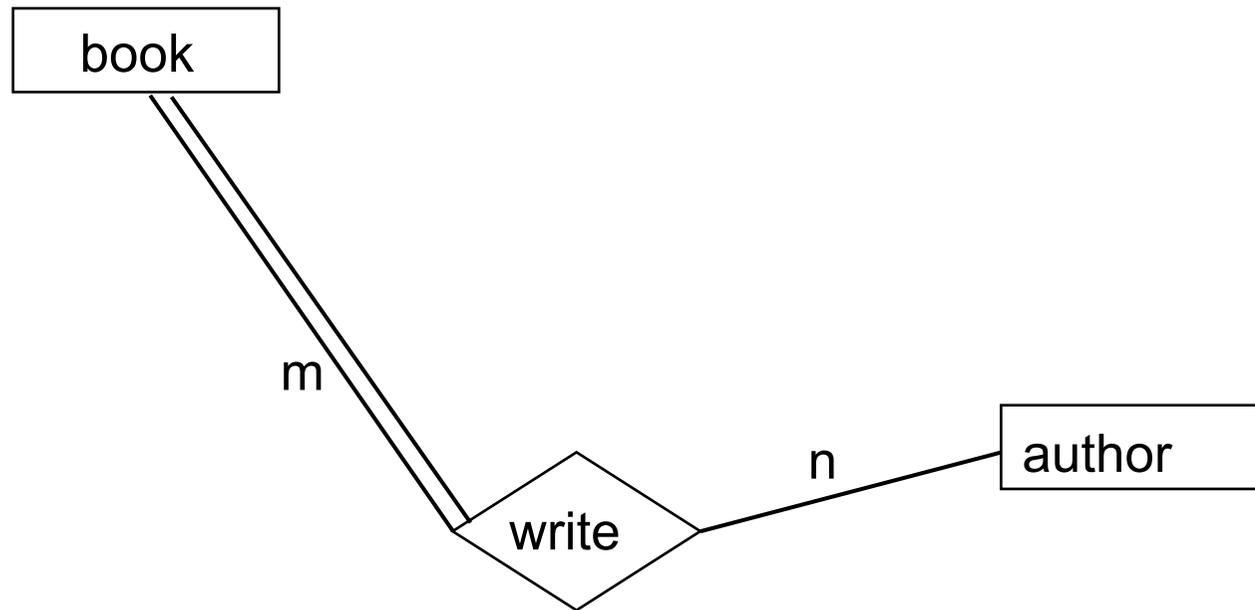
Example (6/7)

A publisher can publish zero or more books and a book is published by exactly one publisher. We also need to record the date of publication.

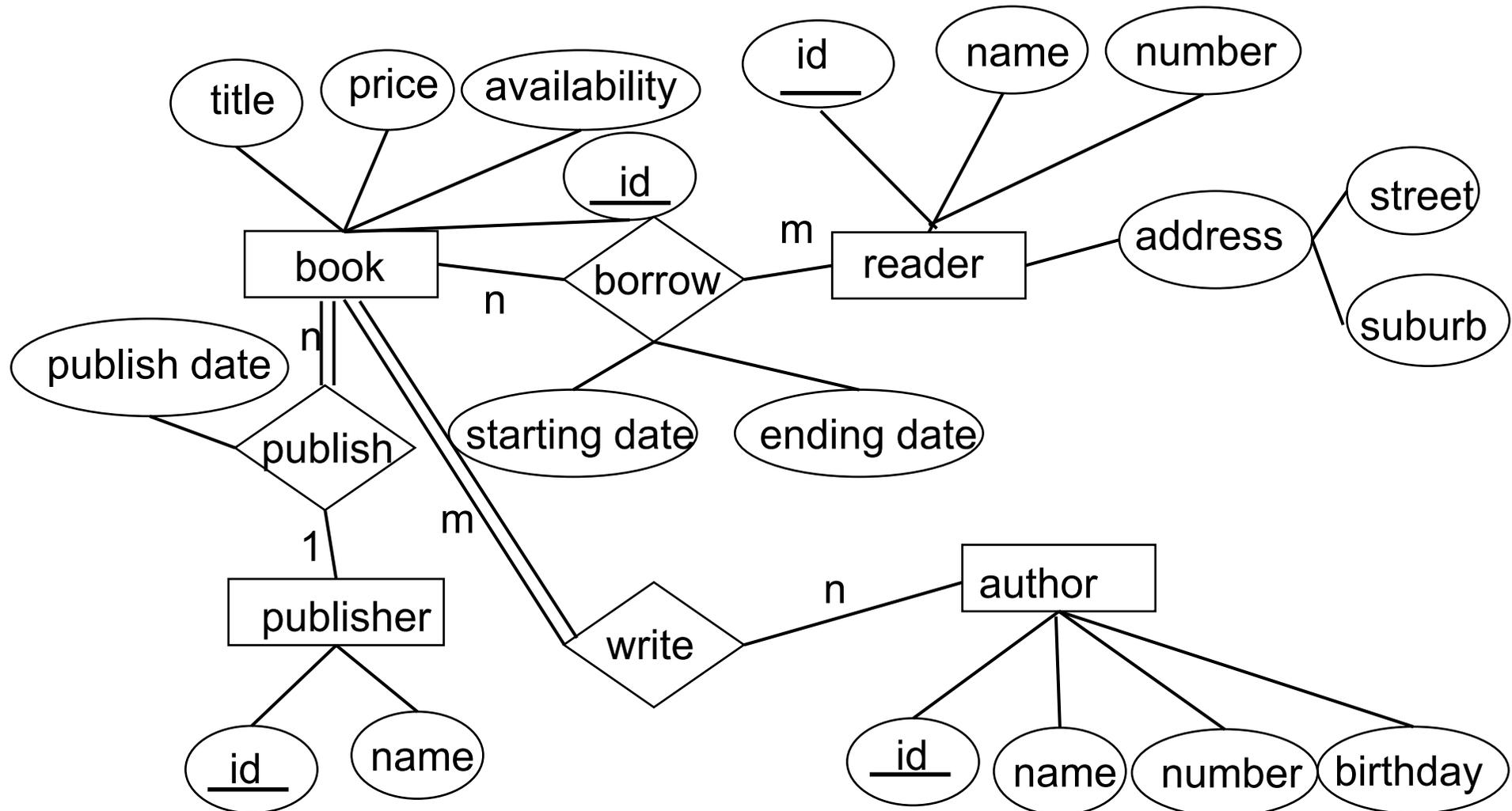


Example (7/7)

An author can write zero or more books and a book is written by one or more authors.



Full sample diagram



Note that in your assignment, you are asked to use UML simplified class diagrams to create a conceptual schema, rather than using the ER diagram 😊