

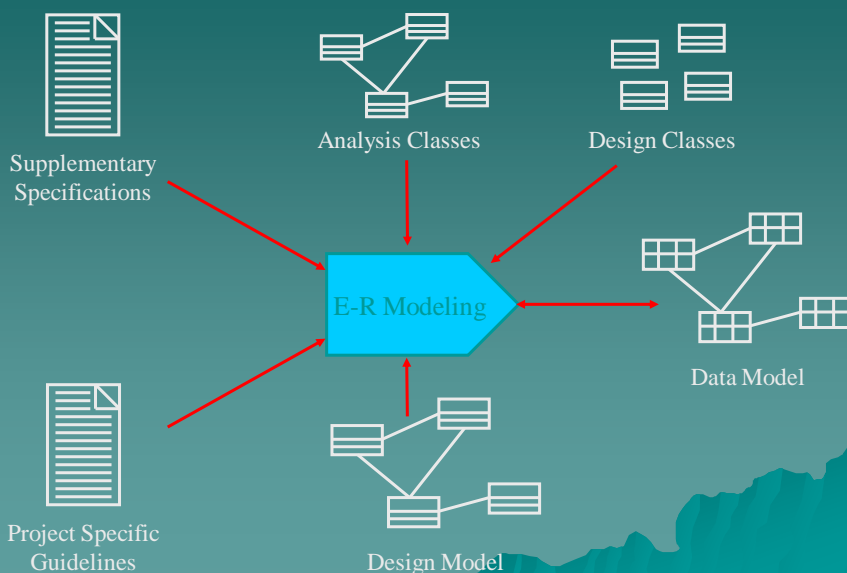
Vietnam and Japan Joint ICT HRD Program

ITSS Software Development Chapter 7. E-R modeling for persistent data

Nguyen Thi Thu Trang
trangntt-fit@mail.hut.edu.vn

1

E-R Modeling Overview



Content

- ⇒ 1. Object model and Rational Data Model
- 2. Map persistent design classes to Entities
- 3. Map class relationships to Relations

3

3

1.1. Relational Databases and OO

- ◆ RDBMS and Object Orientation are not entirely compatible
 - RDBMS
 - ◆ Focus is on data
 - ◆ Better suited for ad-hoc relationships and reporting application
 - ◆ Expose data (column values)
 - Object Oriented system
 - ◆ Focus is on behavior
 - ◆ Better suited to handle state-specific behavior where data is secondary
 - ◆ Hide data (encapsulation)

1.2. The Object Model

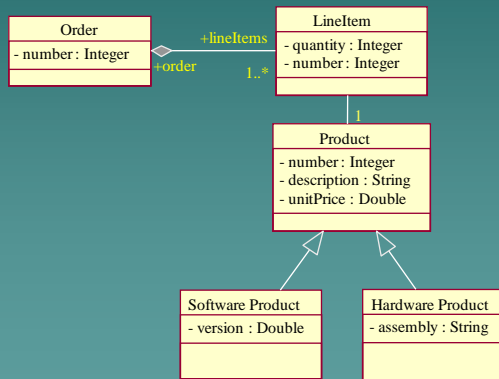
- ◆ The Object Model is composed of

- Classes (attributes)

- Relationships

- ◆ Associations

- ◆ Generalization



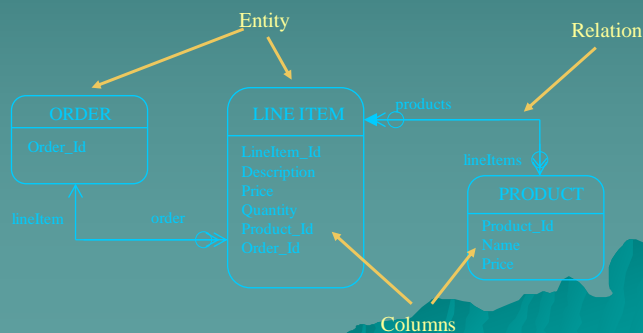
1.3. The Relational Data Model

- ◆ Relational data model is composed of

- Entities - Table

- Relations - Relationship

- Also called **E-R model**



1.3.1. Entities/Tables

- ◆ Entities is mapped to table when design physical database
- ◆ Including
 - Columns: Attributes
 - Rows: Concrete values of attributes

<u>courseID</u>	description	startDate	endDate	location
2008.11.001	This course...	12 Nov 2008	30 Nov 2008	D3-405
2008.11.002	This course...	22 Nov 2008	10 Dec 2008	T-403

7

1.3.2. Relations/Relationships

- ◆ Relations between entities or relationship between tables
- ◆ Multiplicity/Cardinality
 - One-to-one (1:1)
 - One-to-many (1:m)
 - Many-to-one (m:1)
 - Many-to-many (m:n)

(Normally, many-to-many relation is divided to one-to-many and many-to-one relations)

8

Dependency relationships

- ◆ The child entity can exist only when the parent entity exists
- ◆ The child entity has a foreign key referencing to the primary key of the parent entity
- ◆ This foreign key is included in the primary key of the child
- ◆ Solid line



9

Independency relationships

- ◆ The child entity can exist even if the parent entity does not exist
- ◆ The child entity has a foreign key referencing to the primary key of the parent entity
- ◆ This foreign key is not included in the primary key of the child
- ◆ Dash line



10

Content

1. Object model and Rational Data Model
- ➡ 2. Map persistent design classes to Entities
3. Map class relationships to Relations

11

11

2. Mapping Persistent Design Classes to Entities

- ◆ In a relational database
 - Every row is regarded as an object
 - A column in a table is equivalent to a persistent attribute of a class

SubjectInfo
- subjectID : String
- subjectName : String
- numberOfCredit : int



Attributes from object type	subjectID	subjectName	numberOfCredit
Object Instance	IT0001	CS Introduction	4

Content

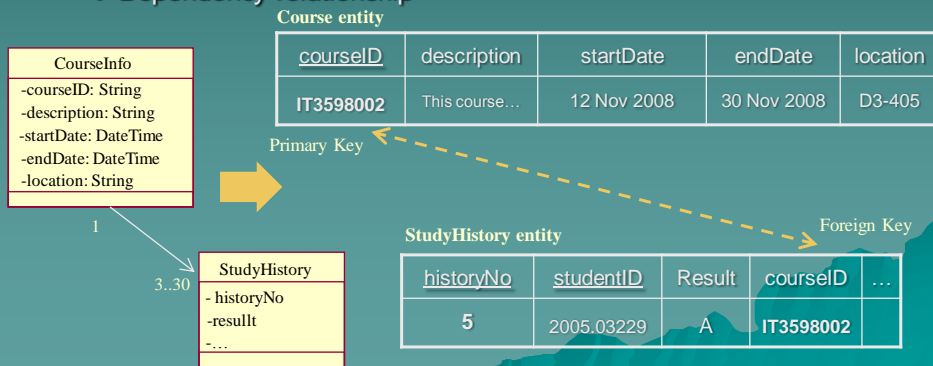
1. Object model and Rational Data Model
2. Map persistent design classes to Entities
- ➔ 3. Map class relationships to Relations

13

13

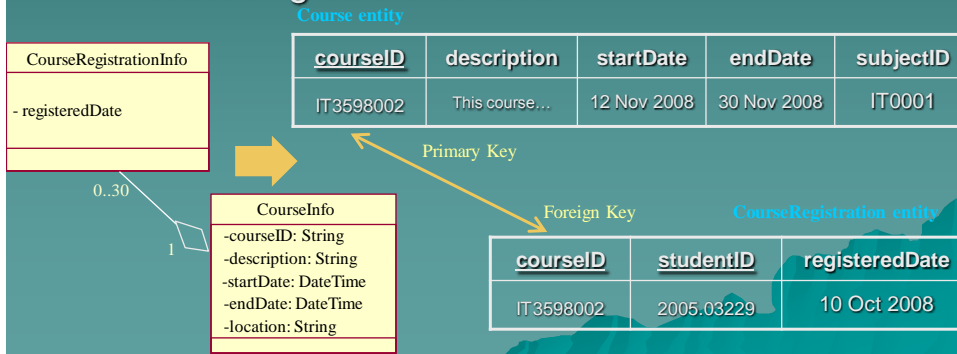
3.1. Mapping Associations Between Persistent Objects

- ◆ Associations between two persistent objects are realized as foreign keys to the associated objects.
 - A foreign key is a column in one table that contains the primary key value of associated object
- Dependency relationship



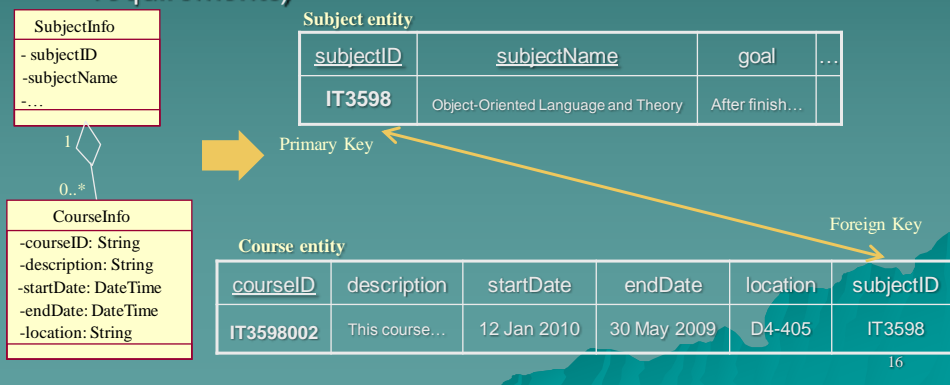
3.2. Mapping Aggregation to the Data Model

- ◆ Aggregation is also modeled using foreign key relationships
 - The use of composition implements a cascading delete constraint

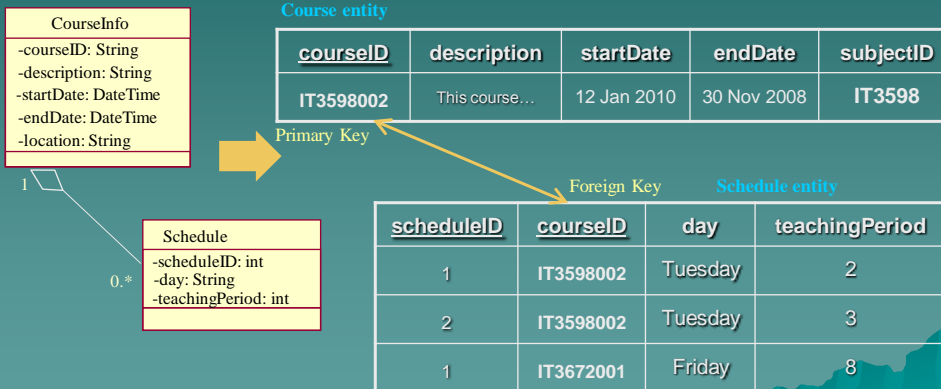


3.2. Mapping Aggregation to the Data Model (2)

- ◆ In some case, we can map to independency relationship to simplify the primary key.
- ◆ Example: CourseID is the primary key (according the requirements)



More example in Course Registration CS

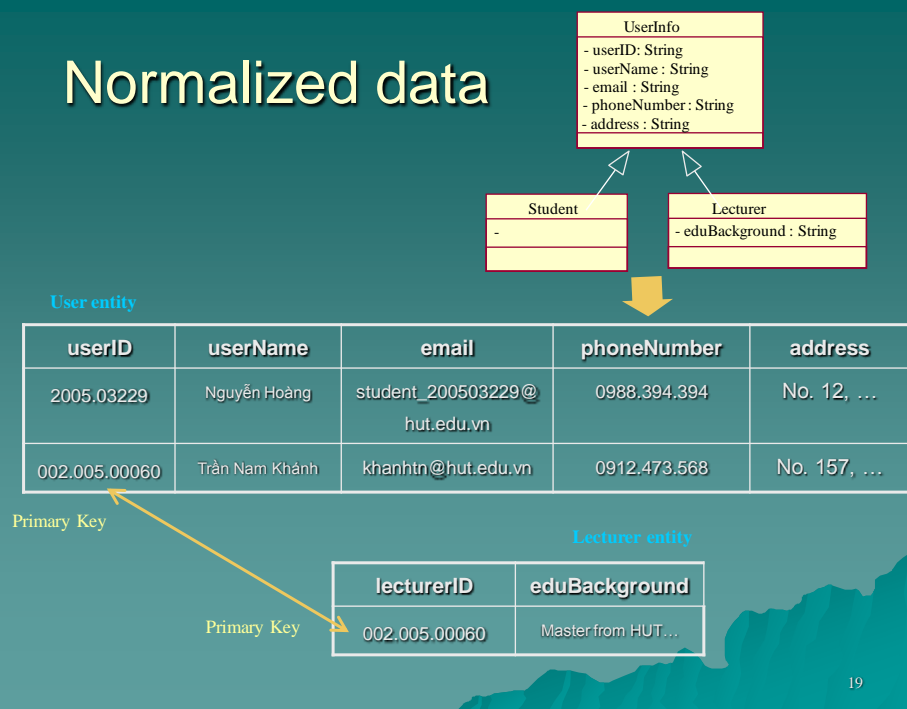


17

3.3. Modeling Inheritance in the Data Model

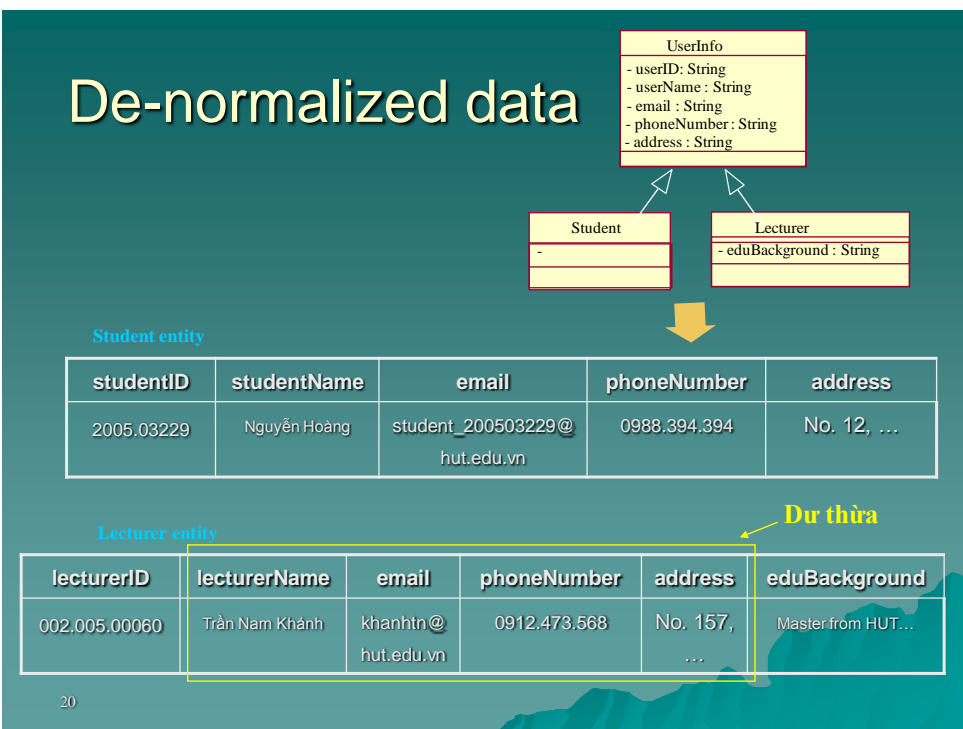
- ◆ A Data Model does not support modeling inheritance in a direct way
- ◆ Two options:
 - Use separate tables (normalized data)
 - Duplicate all inherited associations and attributes (de-normalized data)

Normalized data



19

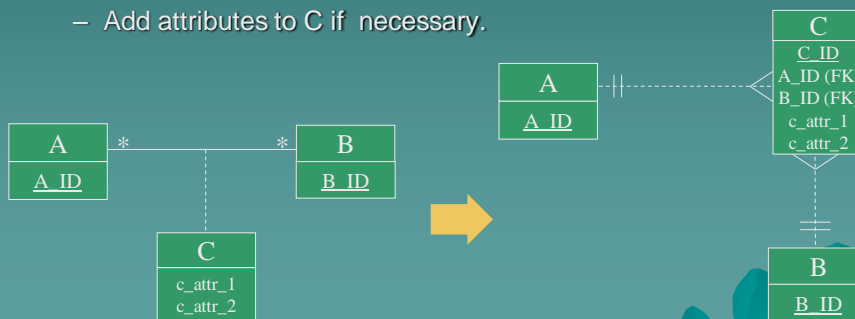
De-normalized data



20

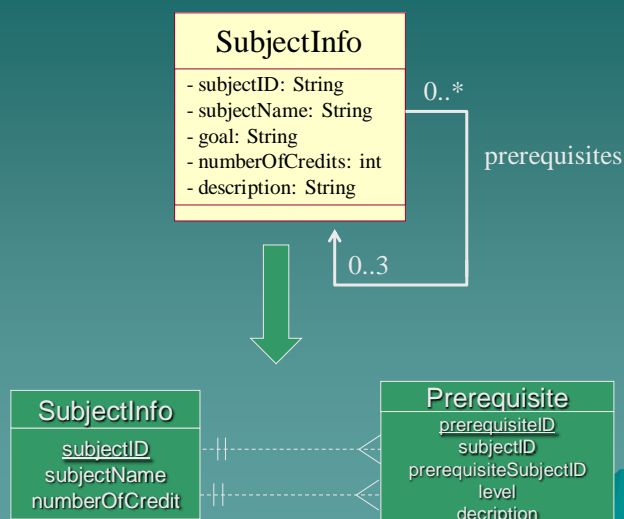
3.4. Mapping many-to-many cardinality

- ◆ Use an intermediate entity
- ◆ Example: The Cardinality of A and B is many-to-many
 - Add an intermediate entity called “C”
 - Place 2 foreign keys for C, referencing to 2 primary keys of A and B
 - Add attributes to C if necessary.



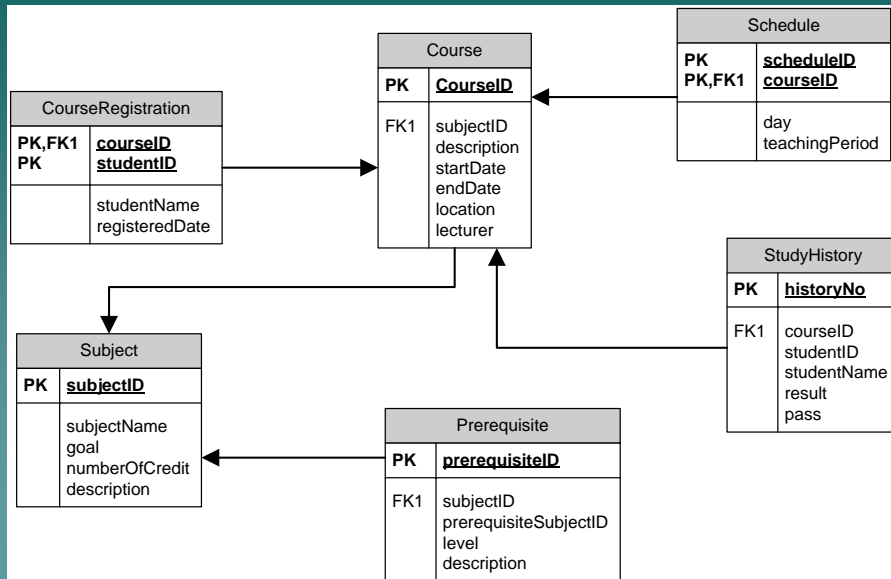
21

Example in Course Registration CS



22

E-R diagram



Checkpoints: E-R modeling

- ◆ Have all persistent classes been mapped to database structures?
- ◆ Have all relationships between classes mapped to relationships between entities?
- ◆ Have all class' attributes and states mapped to columns?



Question?

