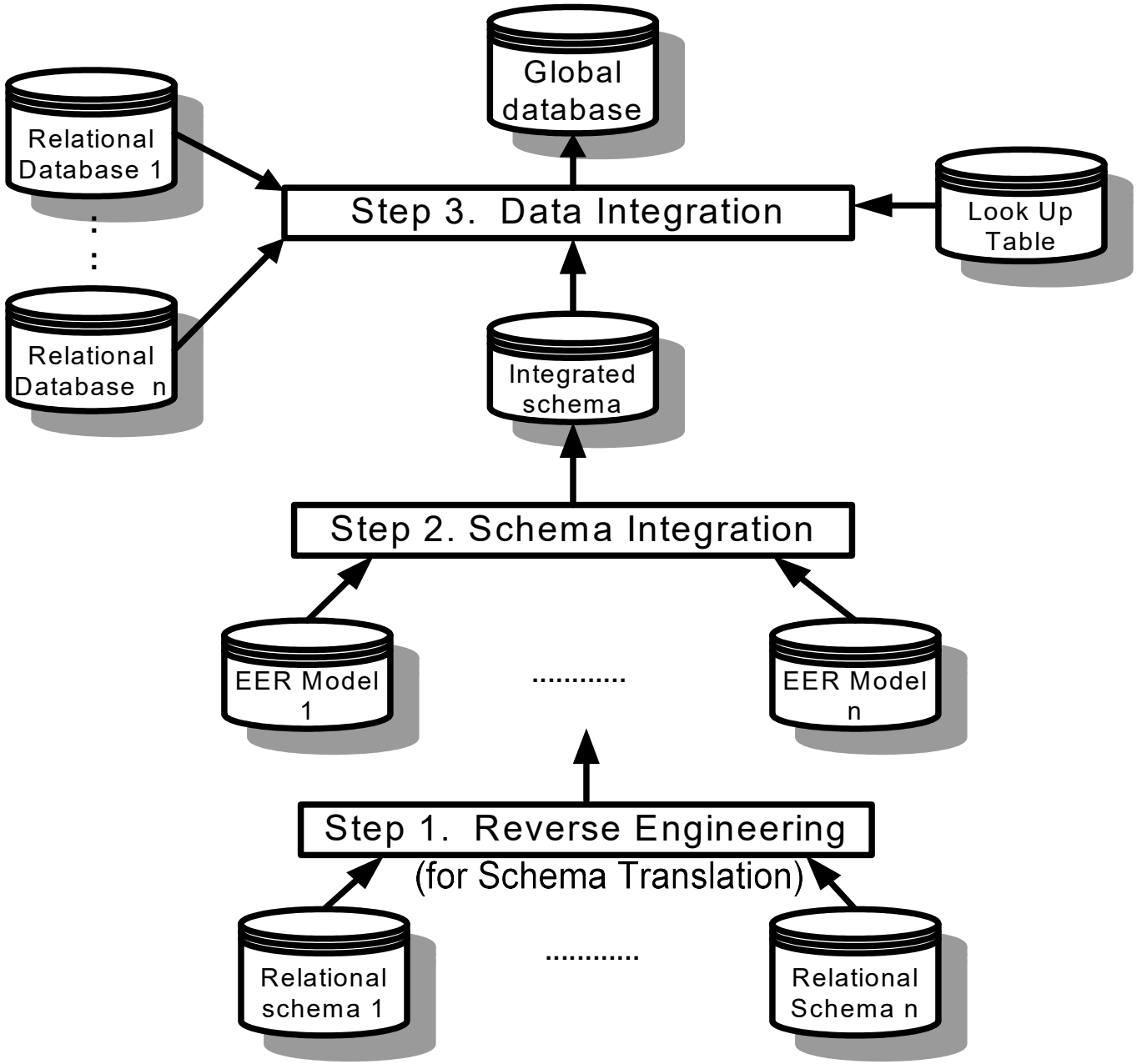
Architecture of multiple databases integration



Data Integration Concepts

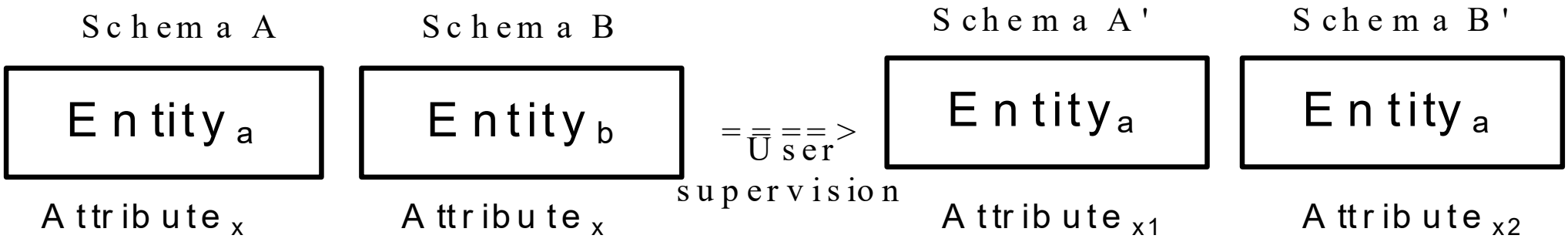
- Data semantics define the relationships between data for user's data requirements.
- Data semantics are presented in database conceptual schema such as EER model and DTD Graph.
- Only relevant data can be integrated for an application.
- Data relevance depends on user's data requirements for an application.
- Data consistency are the standard of data domain value and format. Inconsistent data must be transformed before data integration.
- User supervision means users input for user's data requirements, and which are for database design and schema integration

Schema integration of EER models

```
Begin For each existing database do
  Begin If  its conceptual schema does not exist
          then reconstruct its EER model by reverse engineering;
         For each pair of existing databases' EER models schema A and schema B do
        begin resolve semantic conflicts between EER model A and EER model B;/step1/
              Merge classes between EER model A and EER model B;            /*step2*/
              Merge relationships between EER model A and EER model B;      /*step3*/
         end
    end
```

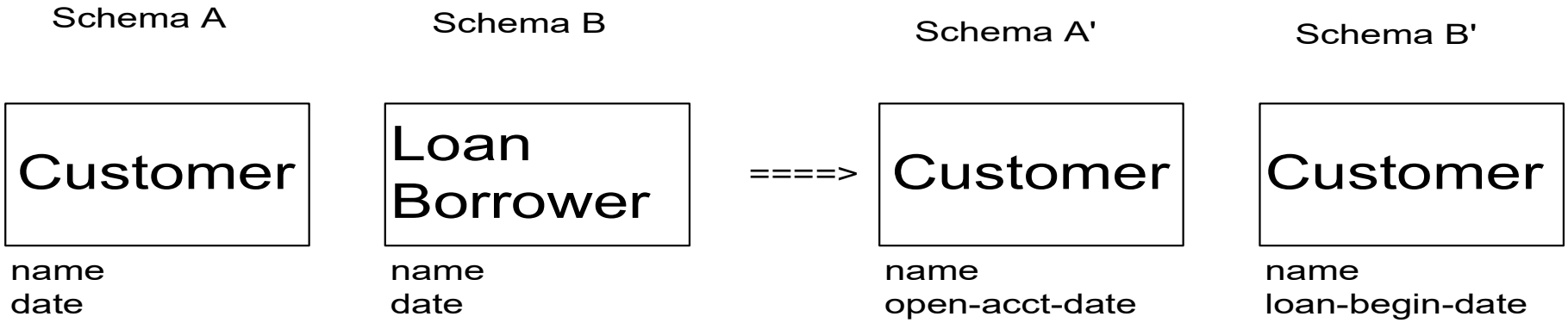
Step 1 Resolve conflicts among EER model

TH1: Giải quyết xung đột về đồng âm và đồng nghĩa



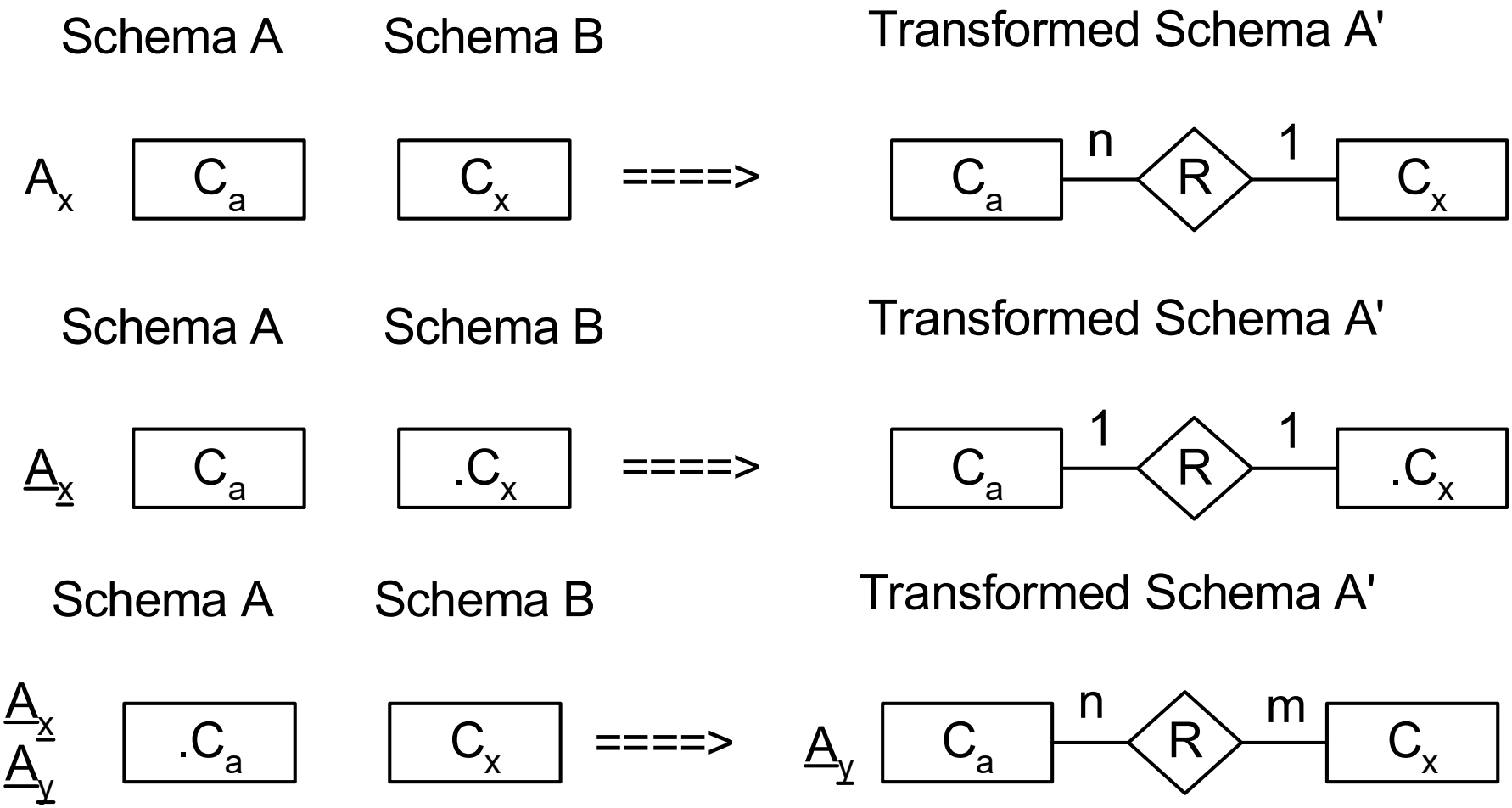
(note: $Class_a$ and $Class_b$ are synonyms, $Attribute_x$ are homonyms)

Example

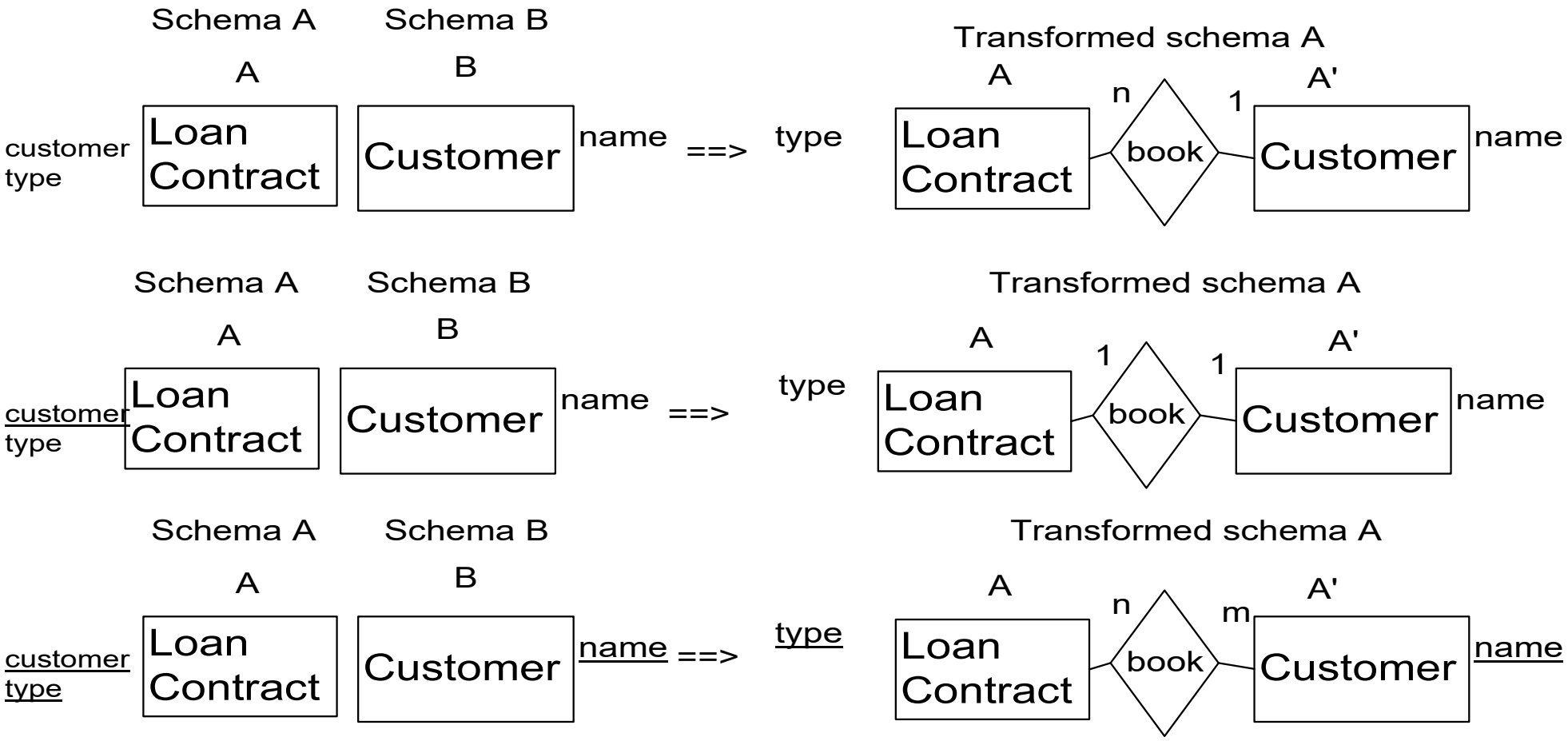


- Loan Borrower và Customer là 1 ví dụ về đồng nghĩa
- name, date trong Loan Borrower và Customer là 1 ví dụ về đồng âm; nhưng chỉ có date cần đổi tên vì khác ý nghĩa.

TH2: Giải quyết xung đột kiểu dữ liệu

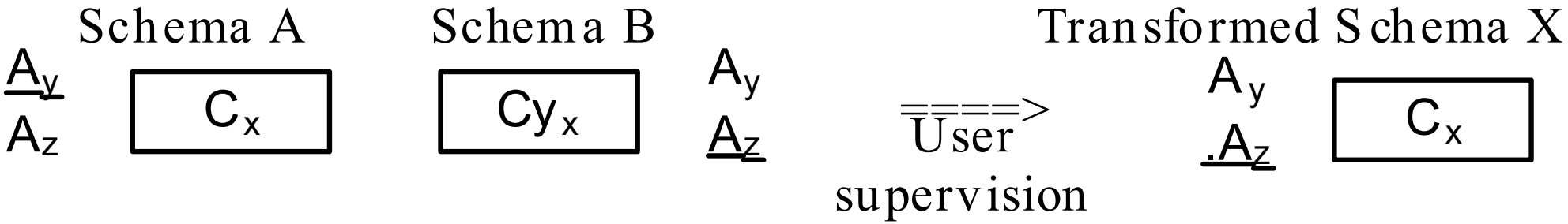


Example


Schema A
Schema B
A
B
customer type
Loan Contract
Customer
name
==>
type
Transformed schema A
A
n
1
A'
Loan Contract
book
Customer
name
Schema A
Schema B
A
B
customer type
Loan Contract
Customer
name
==>
type
Transformed schema A
A
1
1
A'
Loan Contract
book
Customer
name
Schema A
Schema B
A
B
customer type
Loan Contract
Customer
name
==>
type
Transformed schema A
A
n
m
A'
Loan Contract
book
Customer
name

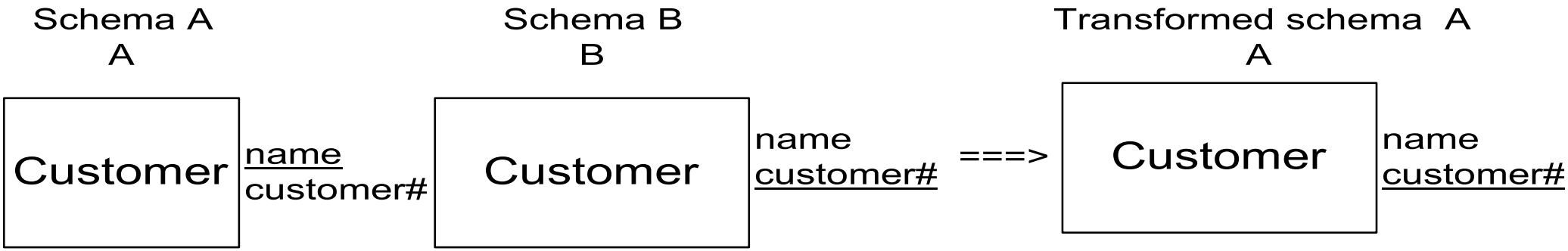
- Lược đồ A có hai thuộc tính là customer và type, lược đồ B có thuộc tính name trong đó name và customer có chung kiểu dữ liệu. Khi tích hợp hai lược đồ này ta sinh thêm một quan hệ giữa hai lược đồ A và B bởi chúng có thuộc tính chung kiểu dữ liệu và bản chất giống nhau.
- Ta xét ba trường hợp sau:
  - Nếu thuộc tính customer không phải là khóa thì khi tích hợp A và B thêm một quan hệ **book** giữa hai lược đồ này: B thành A'; A' và A có quan hệ 1-nhiều.
  - Nếu thuộc tính customer là khóa chính thì khi tích hợp A và B thêm một quan hệ **book** giữa hai lược đồ này: B thành A'; A' và A có quan hệ 1-1
  - Nếu thuộc tính customer là thuộc tính tạo khóa chính thì khi tích hợp A và B thêm một quan hệ **book** giữa hai lược đồ này: B thành A' ; A' và A có quan hệ nhiều-nhiều và khóa của A là **type** và khóa của A' là **name**

TH3: Giải quyết xung đột khóa

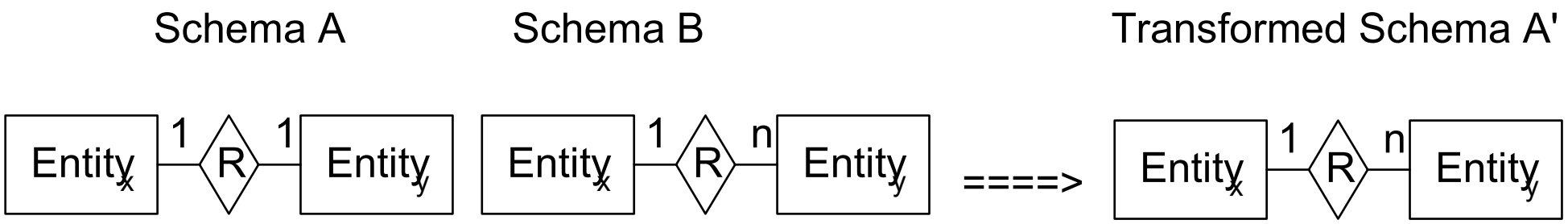


Bản chất của hai tập thực thể A và B là như nhau nhưng khóa khác nhau nên khi tích hợp chúng lại thành một tập thực thể X, chúng ta cần giải quyết xung đột về khóa bằng cách chọn một trong hai khóa của A và B làm khóa cho X

Example


Schema A
A
Customer
name
customer#
Schema B
B
Customer
name
customer#
===>
Transformed schema A
A
Customer
name
customer#

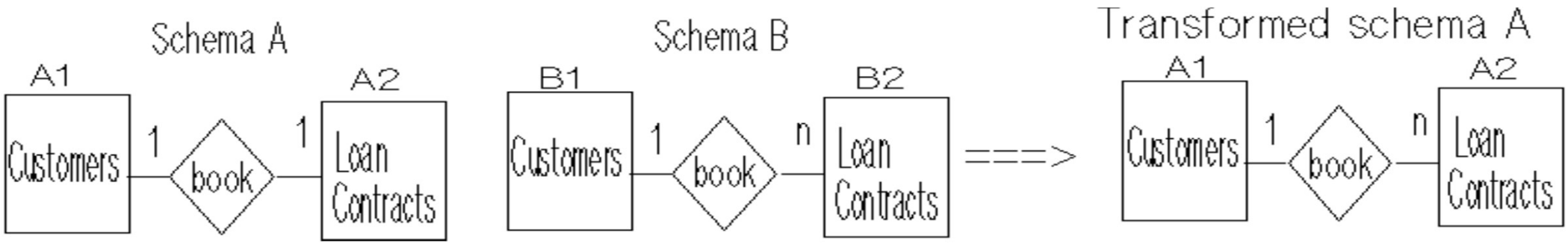
TH4: Giải quyết xung đột về lực lượng



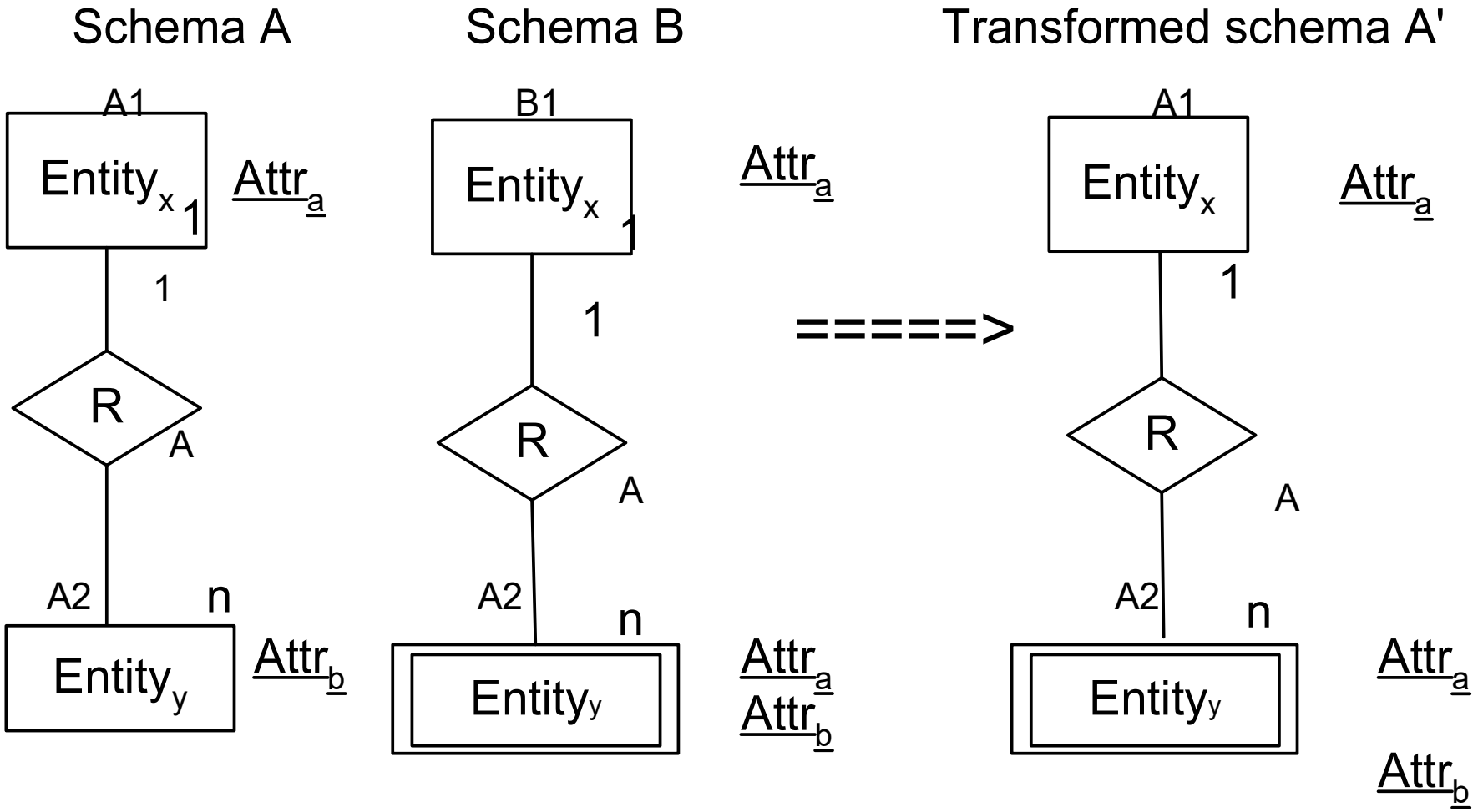
Hai lược đồ A và B giống hệt nhau ngoại trừ lực lượng của quan hệ giữa hai tập thực thể, một lược đồ có quan hệ 1-1, còn quan hệ kia có quan hệ 1-nhiều.

Khi tích hợp hai lược đồ này chúng ta chuyển thành một lược đồ có hai tập thực thể đó và một quan hệ một-nhiều giữa hai tập thực thể

Example


Schema A
A1
A2
Customers
1
book
1
Loan Contracts
Schema B
B1
B2
Customers
1
book
n
Loan Contracts
===>
Transformed schema A
A1
A2
Customers
1
book
n
Loan Contracts

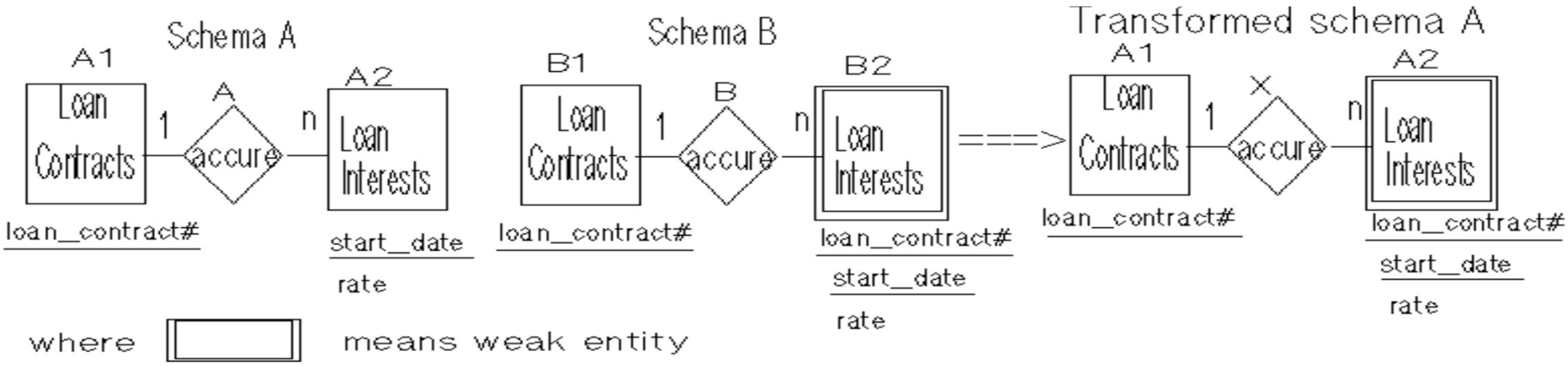
TH5: Giải quyết xung đột của thực thể yếu



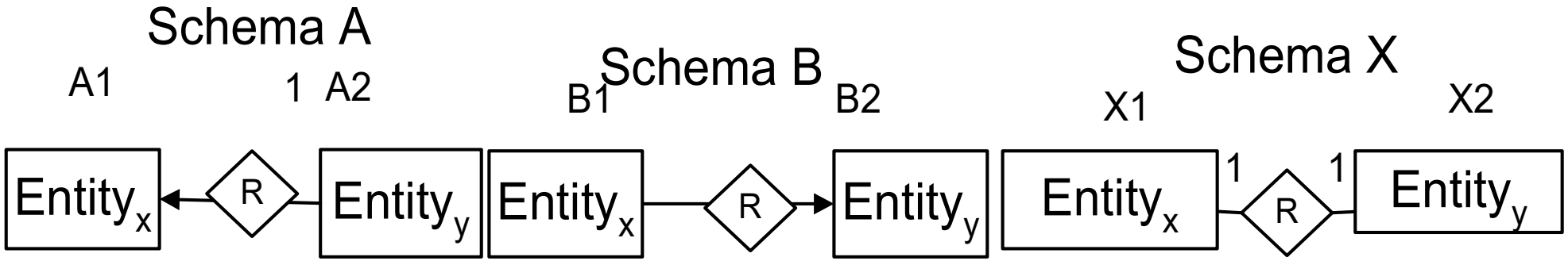
- Hai lược đồ A và B giống nhau, chỉ khác là tập thực thể Y là thực thể yếu trong lược đồ B, nhưng trong lược đồ A là tập thực thể bình thường

- Khi tích hợp hai lược đồ này chúng ta có một lược đồ giống hệt lược đồ B, có nghĩa là tập thực thể Y được chuyển sang thành tập thực thể yếu sau khi tích hợp

Example

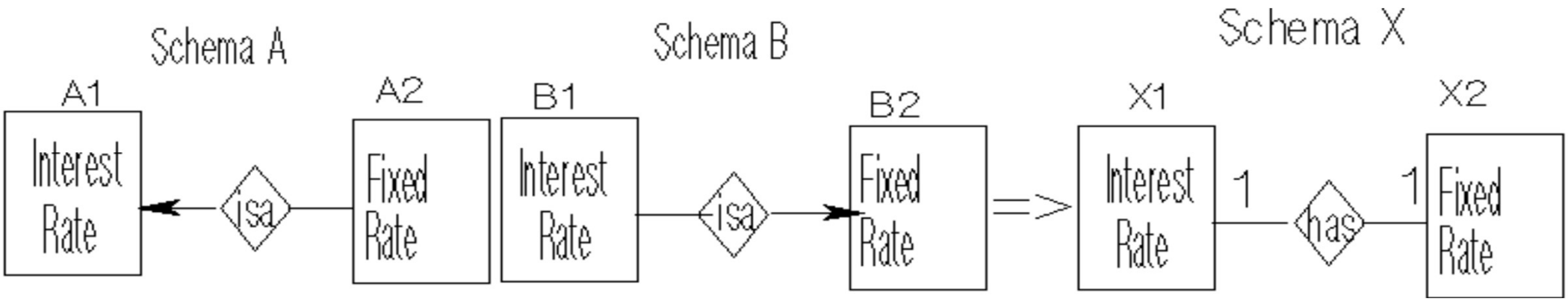

Schema A
A1
Loan Contracts
1
A
accure
n
A2
Loan Interests
loan_contract#
start_date
rate
Schema B
B1
Loan Contracts
1
B
accure
n
B2
Loan Interests
loan_contract#
start_date
rate
===>
Transformed schema A
A1
Loan Contracts
1
X
accure
n
A2
Loan Interests
loan_contract#
loan_contract#
start_date
rate
where
means weak entity

TH6: Giải quyết xung đột trên thực thể kiểu con



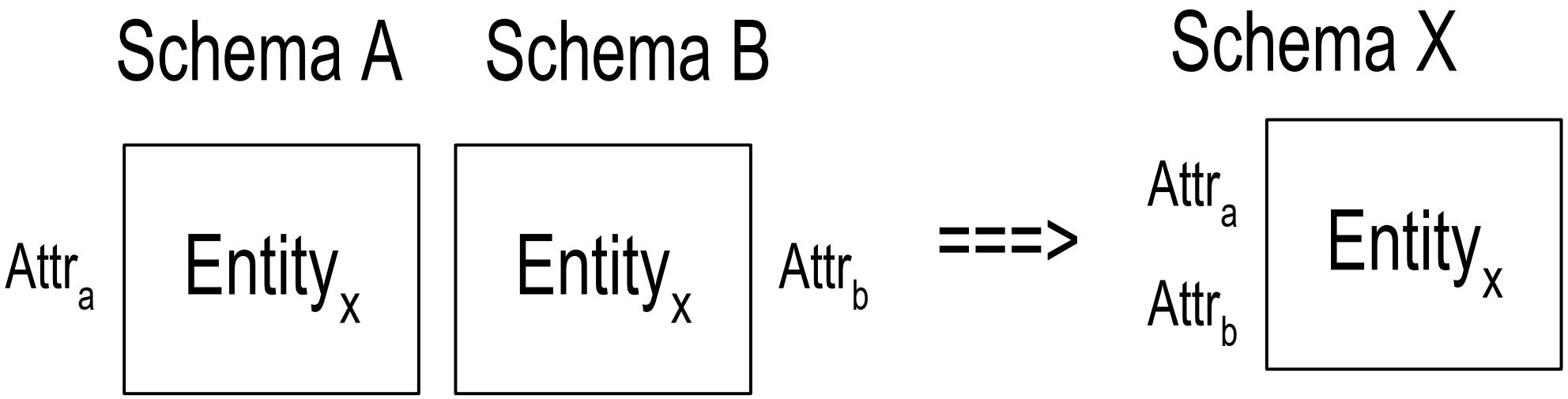
Cho hai tập thực thể X và Y có ràng buộc như trên hình vẽ: X là con của Y, Y là con của X, khi tích hợp lại ta có X và Y có quan hệ 1-1

Example

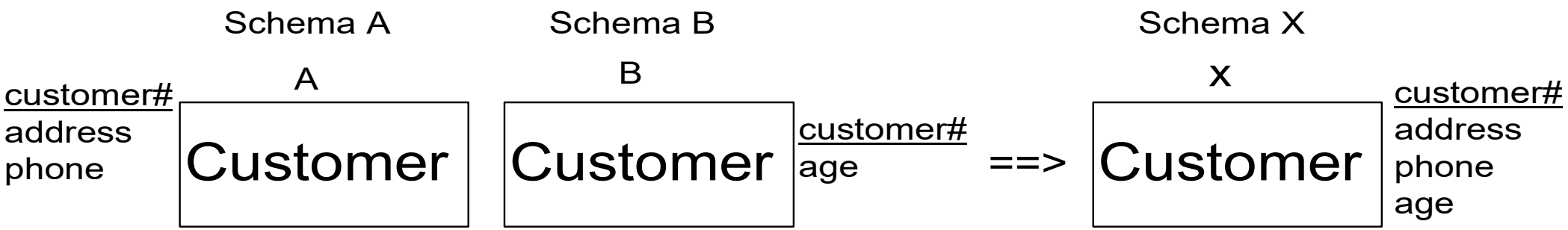

Schema A
Schema B
Schema X
A1
A2
B1
B2
X1
X2
Interest Rate
isa
Fixed Rate
Interest Rate
isa
Fixed Rate
=>
Interest Rate
1
has
1
Fixed Rate

Step 2 Merge entities

TH1: Trộn các tập thực thể bằng phép hợp

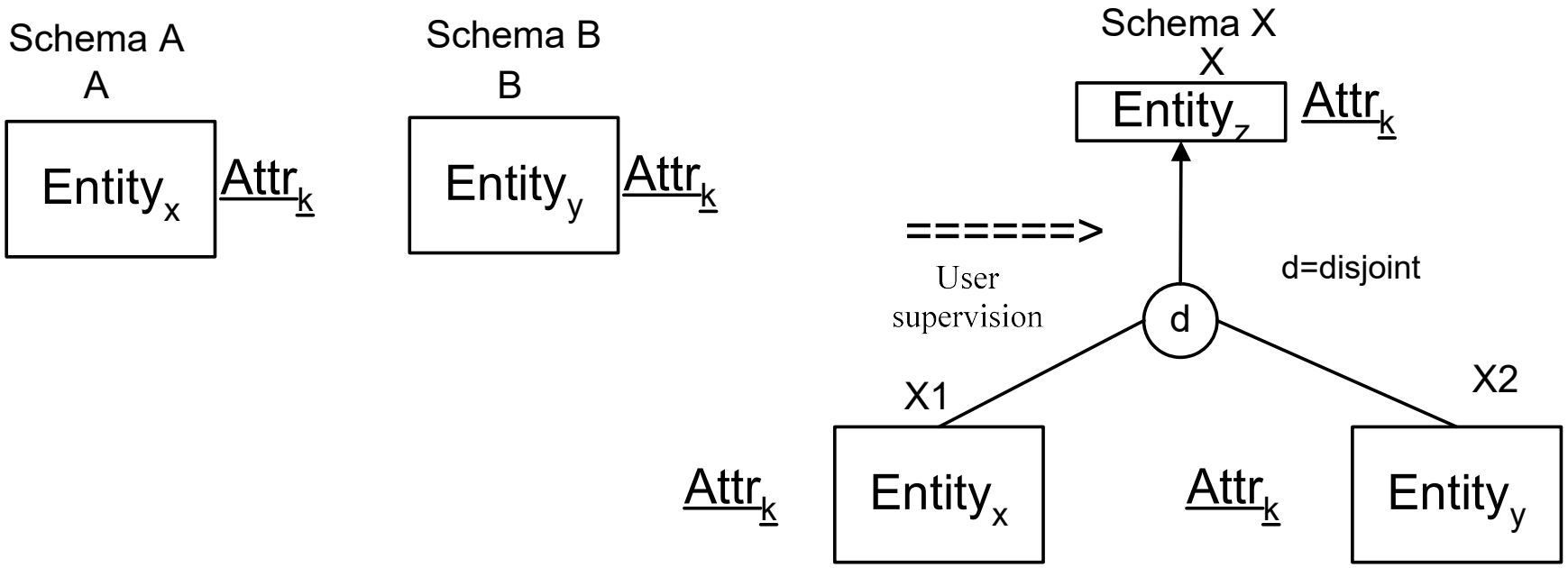


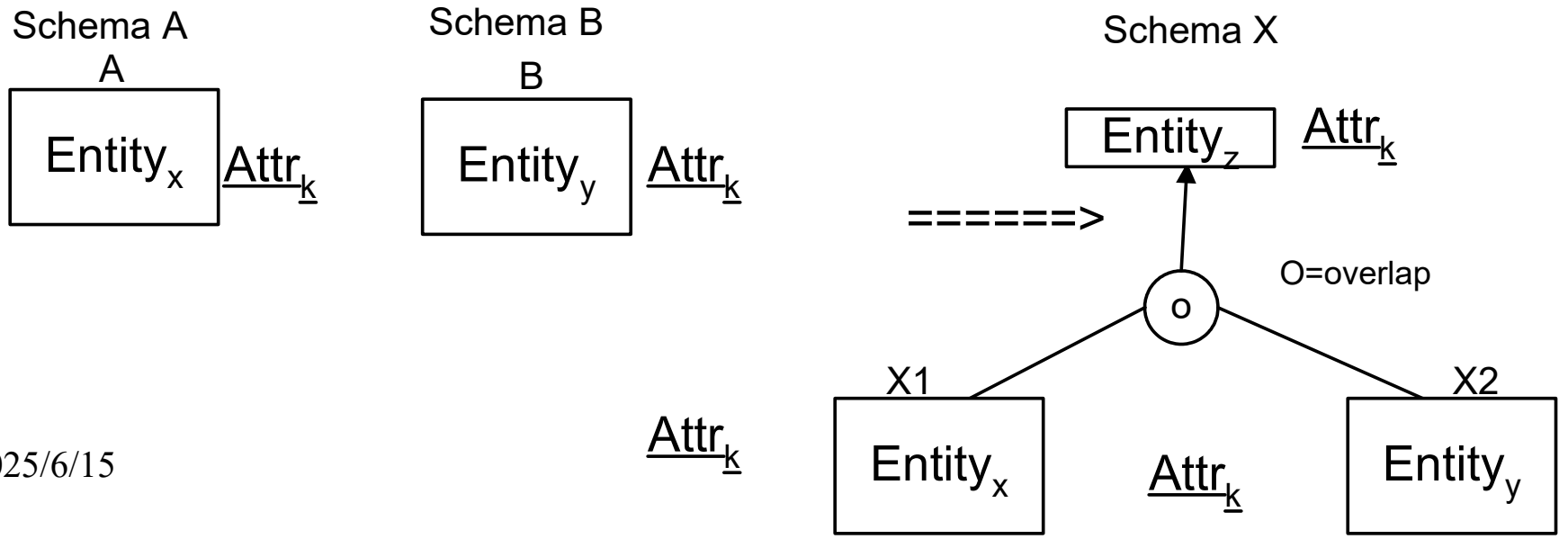
Example



Chú ý rằng như contraint NOT NULL có thể phải bỏ tùy dữ liệu 2 schema

TH2: Trộn EER bằng cách khái quát hóa





Example

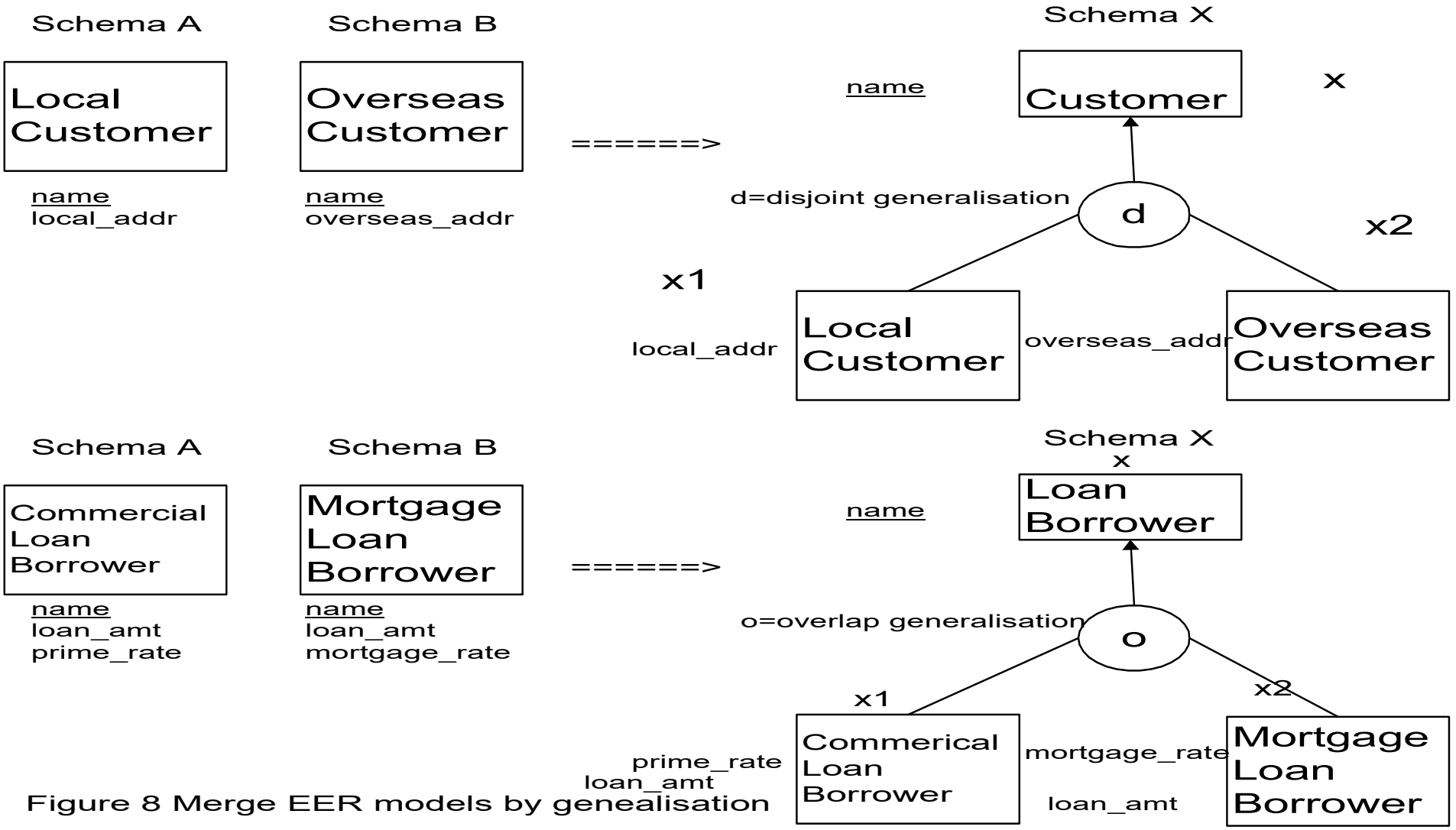
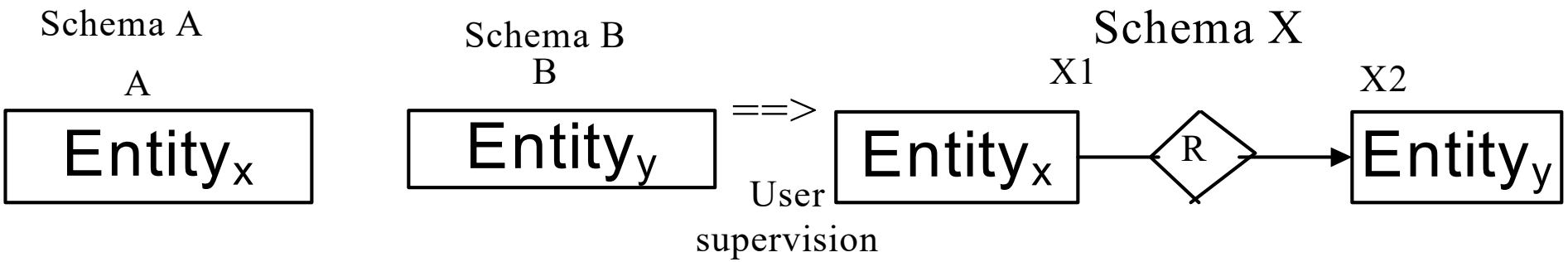


Figure 8 Merge EER models by genealisation

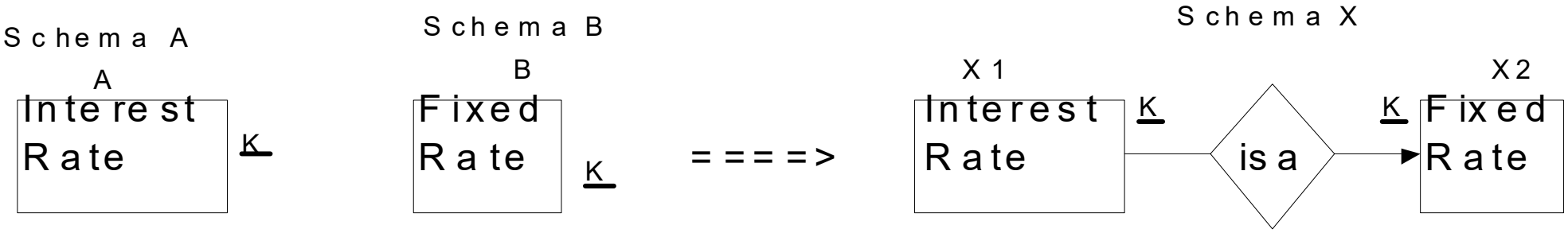
TH3: Trộn EER bằng quan hệ loại con



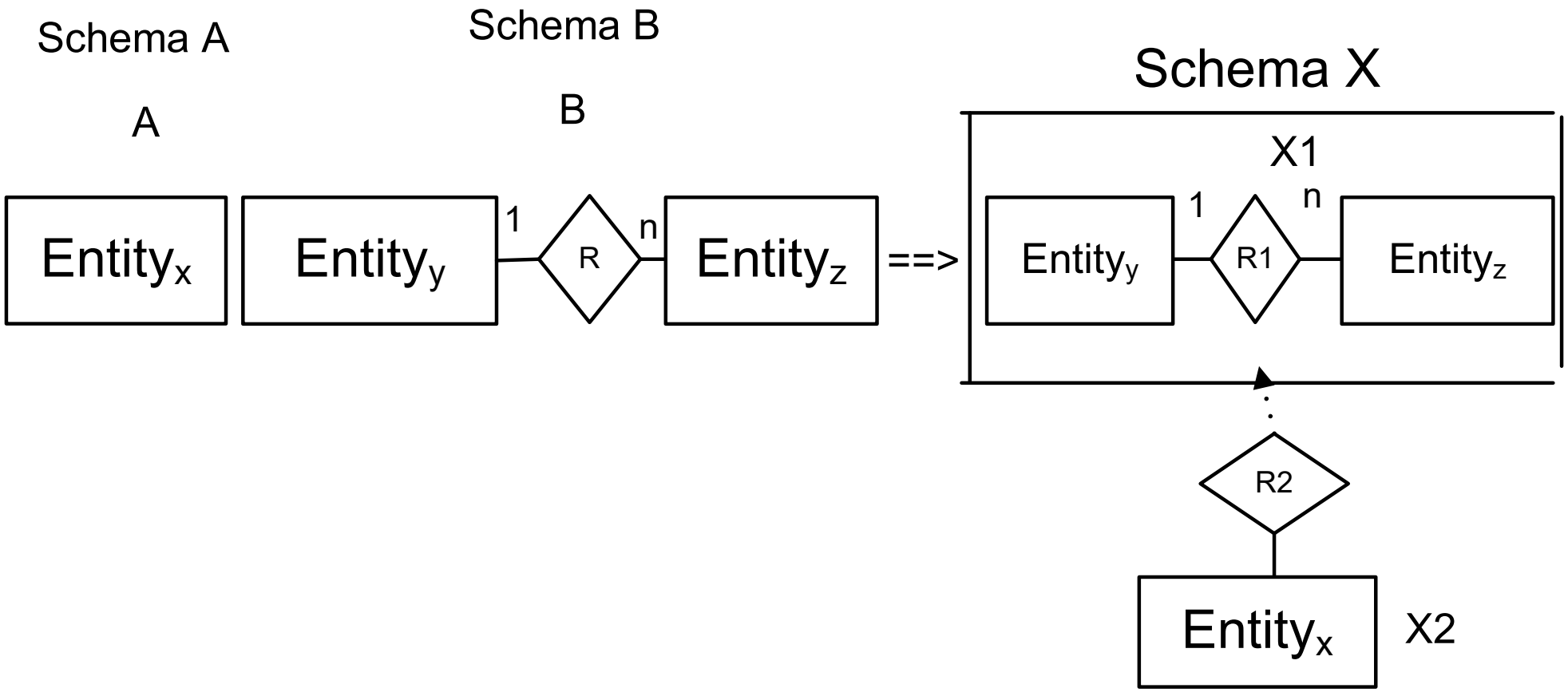
Hai lược đồ A và B có hai tập thực thể X và Y và chung nhau khóa K.

Nếu hai tập thực thể này có quan hệ cha con thì sau khi tích hợp ta thu được một lược đồ với hai tập thực thể và một quan hệ cha con (subtype) giữa chúng

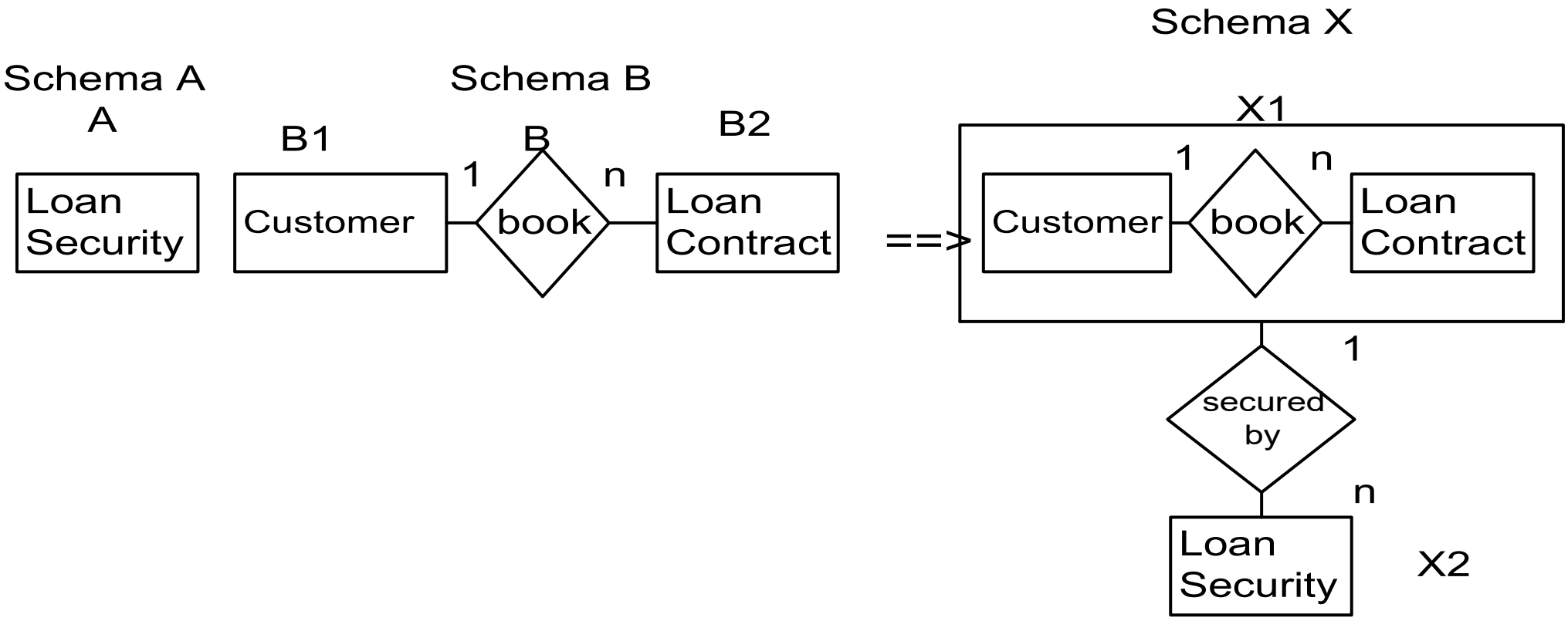
Example


Schema A
A
Interest Rate
K
Schema B
B
Fixed Rate
K
====>
Schema X
X1
Interest Rate
K
is a
K
X2
Fixed Rate

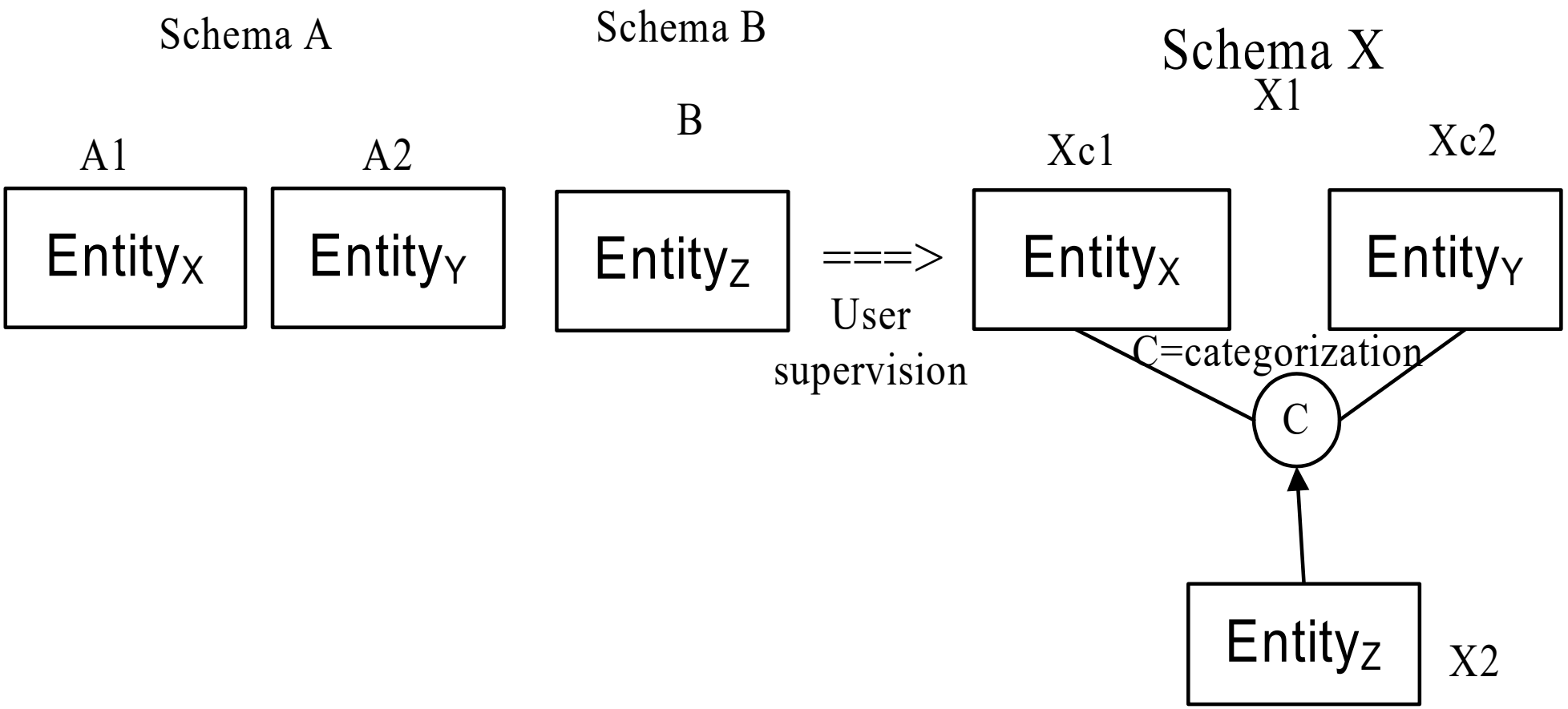
TH4: Trộn EER bằng cách dùng thực thể tích hợp (Aggregation)



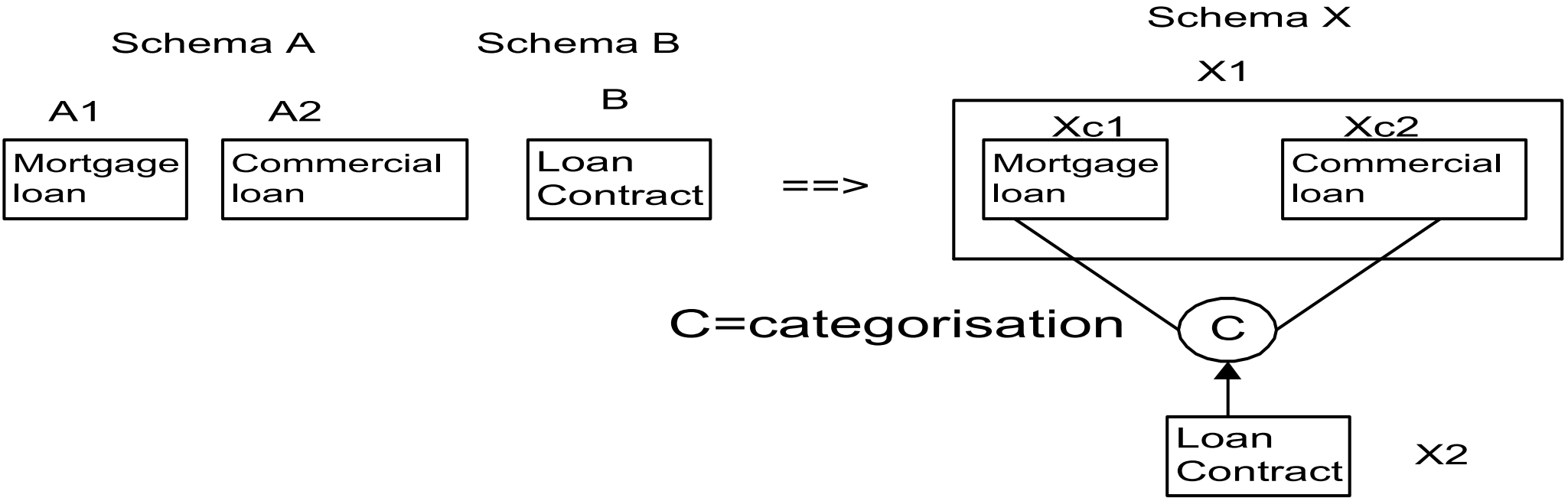
Example


Schema A
A
Loan Security
Schema B
B1
Customer
B
1
book
n
B2
Loan Contract
==>
Schema X
X1
1
Customer
book
n
Loan Contract
1
secured by
n
Loan Security
X2

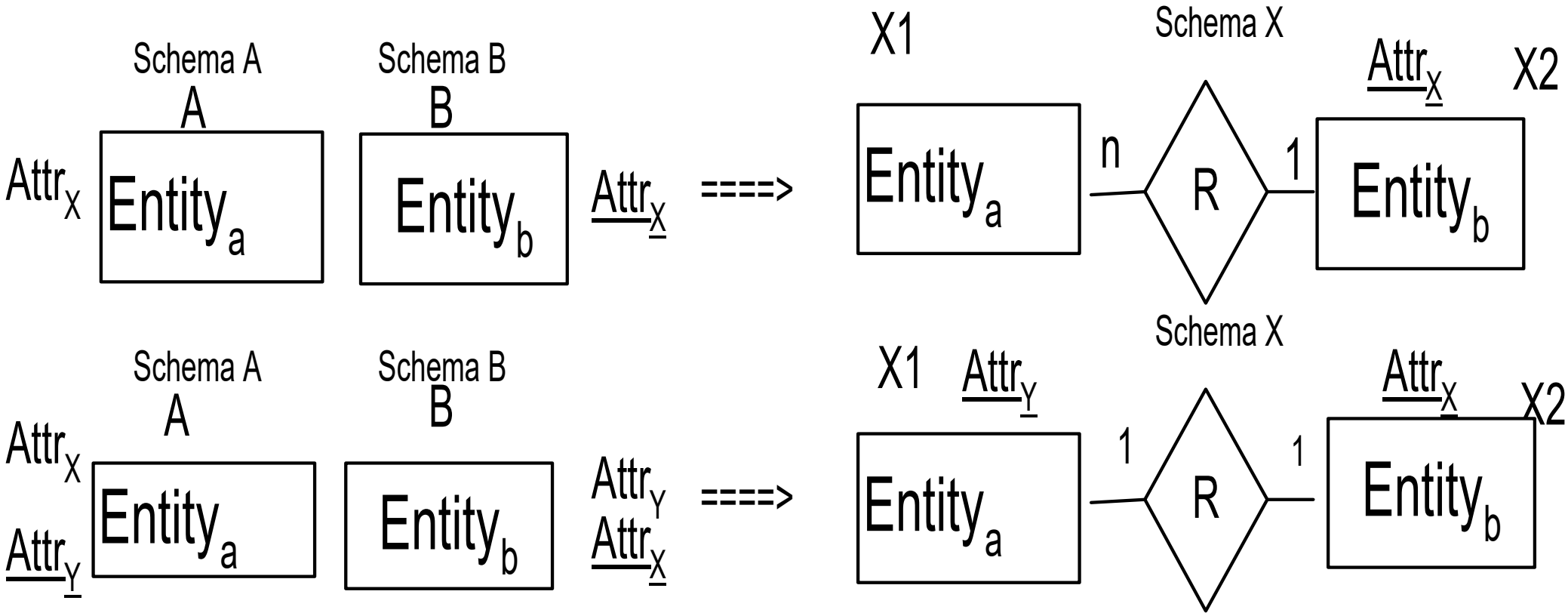
TH5: Trộn EER bằng cách dùng phân loại



Example


Schema A
Schema B
Schema X
A1
A2
B
X1
Xc1
Xc2
Mortgage loan
Commercial loan
Loan Contract
==>
Mortgage loan
Commercial loan
C=categorisation
C
Loan Contract
X2

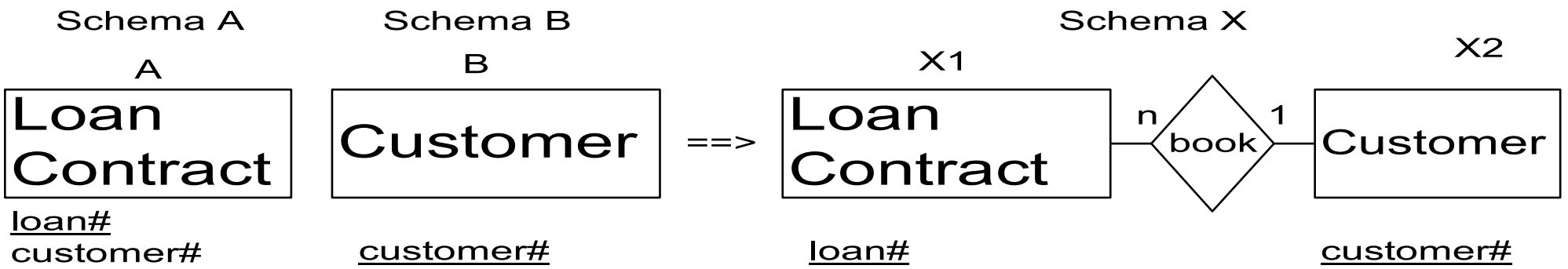
TH6: Trộn EER bằng cách dùng quan hệ hai ngôi

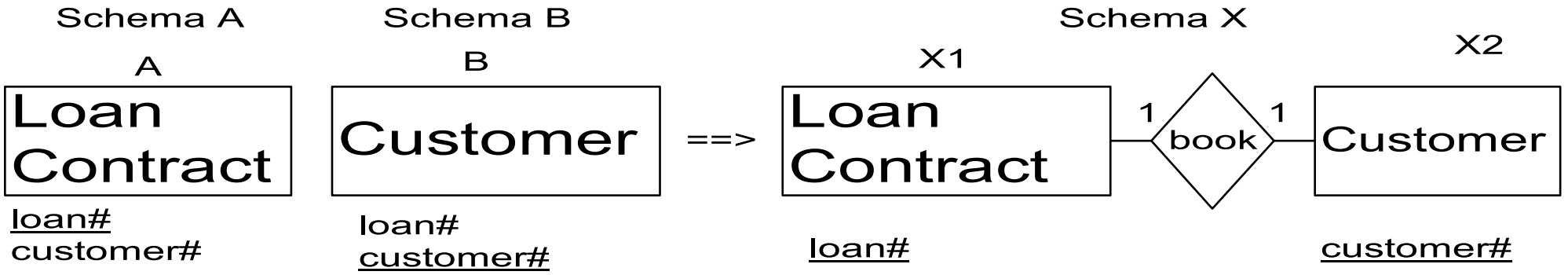


- Trường hợp 1: lược đồ A chứa tập thực thể A với thuộc tính X không phải khóa chính, lược đồ B chứa tập thực thể B cũng có thuộc tính X nhưng là khóa chính thì khi trộn hai lược đồ đó thu được lược đồ X với hai thực thể A và B và một quan hệ một-nhiều R giữa chúng

- Trường hợp 2: lược đồ A chứa tập thực thể A với thuộc tính X không phải khóa chính và thuộc tính Y là khóa chính, lược đồ B chứa tập thực thể B cũng có thuộc tính X nhưng là khóa chính và Y không là khóa chính thì khi trộn hai lược đồ đó thu được lược đồ X với hai thực thể A và B và một quan hệ một-một R giữa chúng

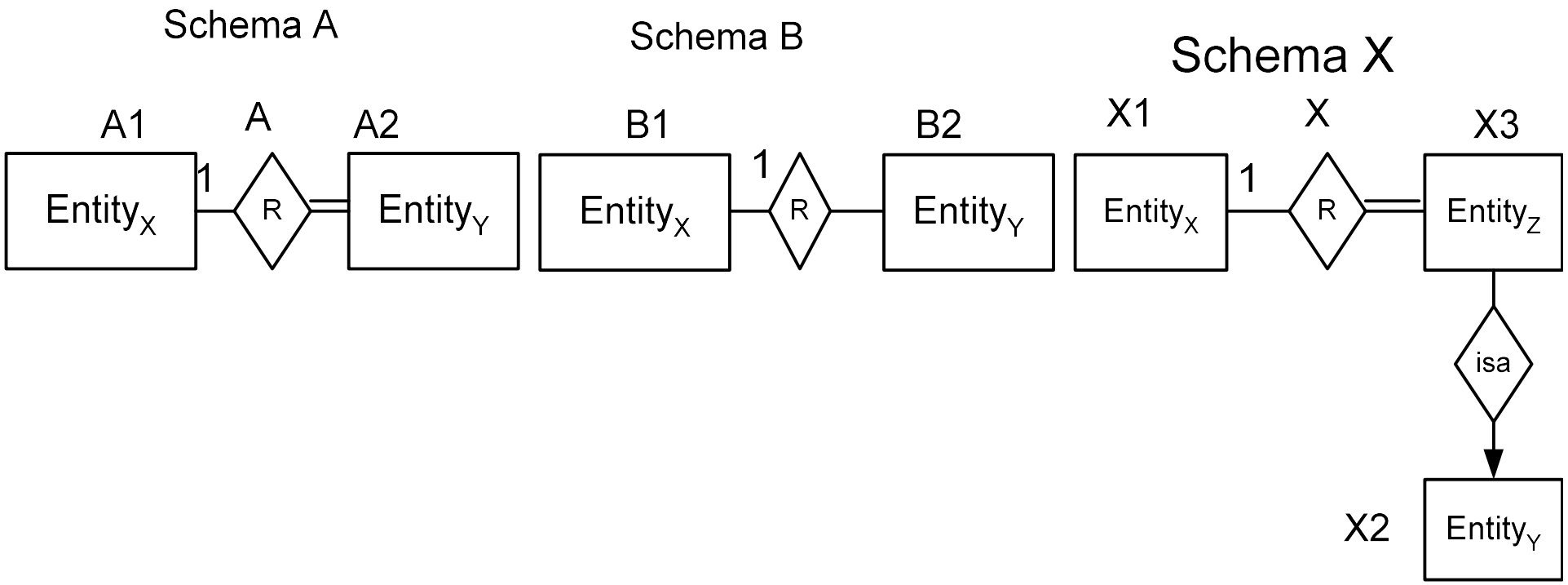
Example


Schema A
A
Loan Contract
loan#
customer#
Schema B
B
Customer
customer#
==>
Schema X
X1
Loan Contract
loan#
n
book
1
X2
Customer
customer#


Schema A
A
Loan Contract
loan#
customer#
Schema B
B
Customer
loan#
customer#
==>
Schema X
X1
Loan Contract
loan#
1
book
1
X2
Customer
customer#

Step 3 Merge relationships

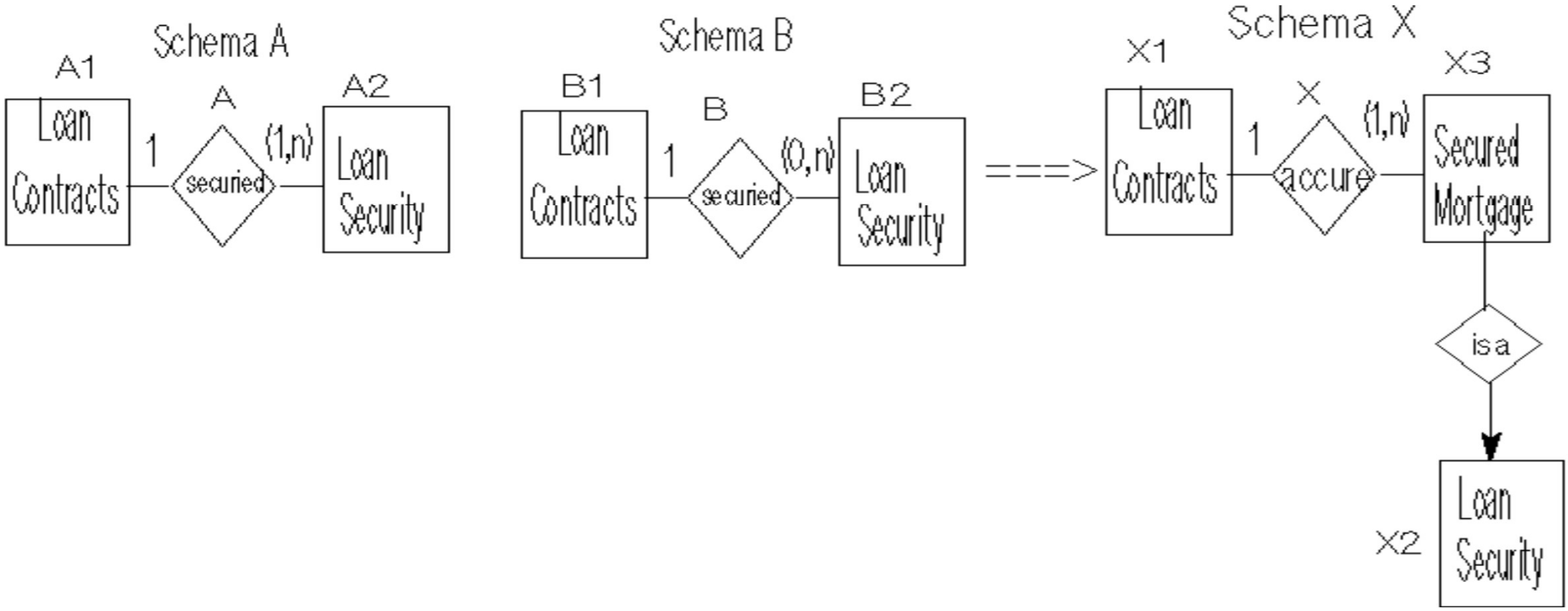
TH1: Trộn các quan hệ bằng quan hệ cha con



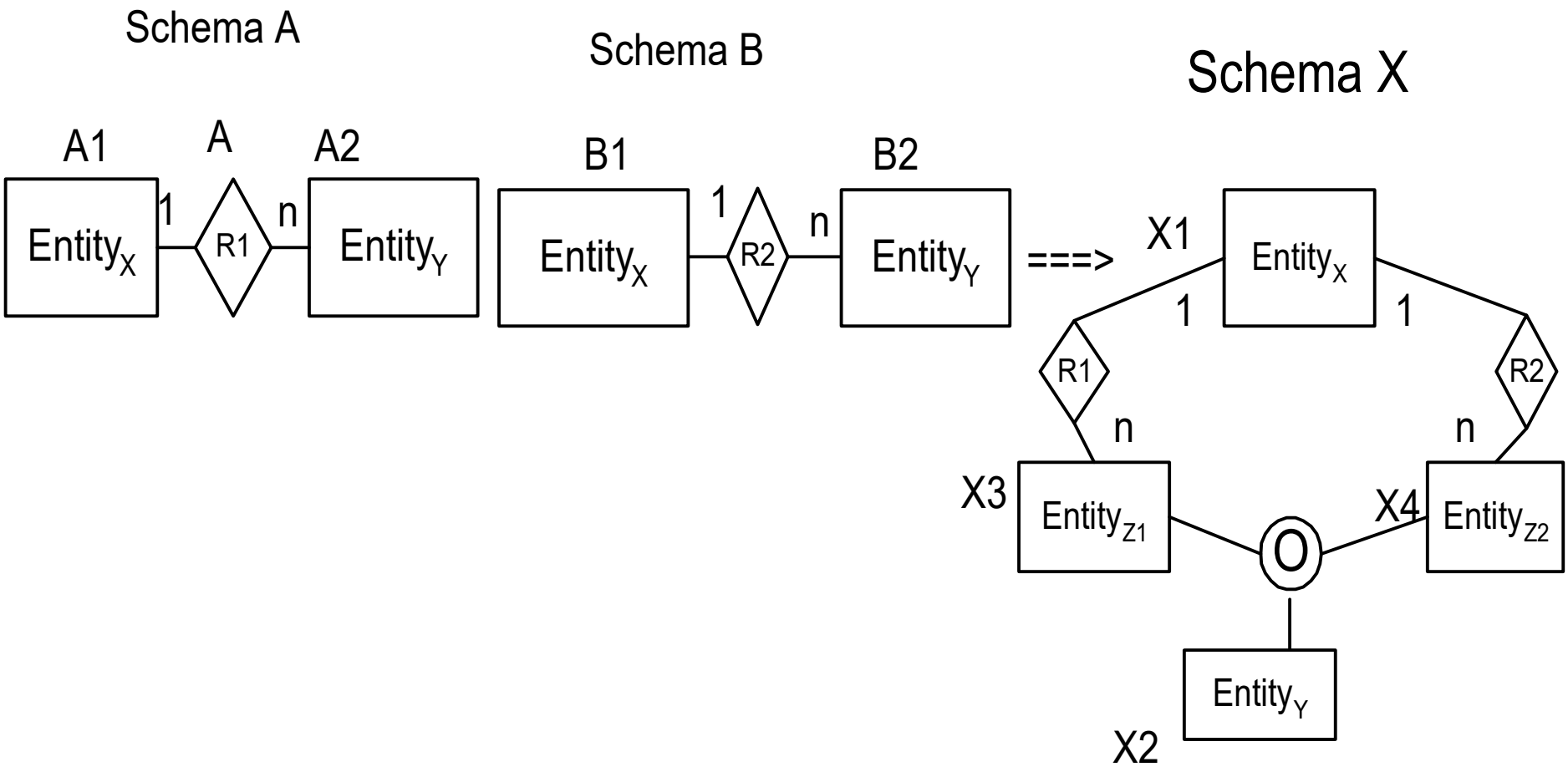
- Lược đồ A và B phần lớn là giống nhau chỉ khác tập thực thể Y tham gia toàn phần vào quan hệ R trong lược đồ A, còn tham gia một phần vào quan hệ R trong lược đồ B.

- Khi trộn hai lược đồ, ta đưa thêm vào một tập thực thể Z có quan hệ R toàn phần với X và quan hệ là-một với Y (như hình vẽ dưới đây). Nhờ quan hệ R và quan hệ là-một, X quan hệ R với Y nhưng là một phần vì không phải tất cả các thực thể trong Y đều nằm trong Z, mà Z thì quan hệ toàn phần với X

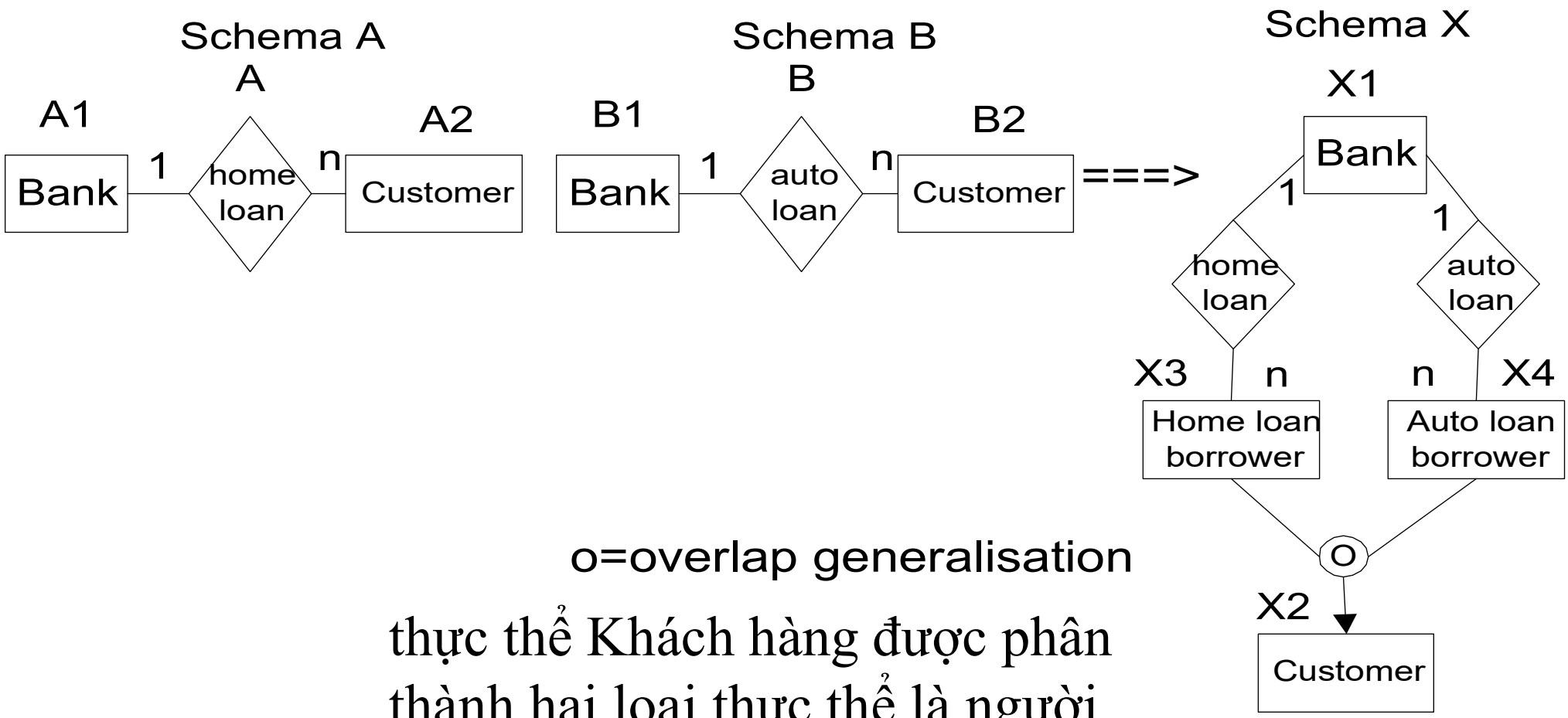
Example


Schema A
A1
Loan Contracts
1
A
securied
(1,n)
A2
Loan Security
Schema B
B1
Loan Contracts
1
B
securied
(0,n)
B2
Loan Security
===>
Schema X
X1
Loan Contracts
1
X
accure
(1,n)
X3
Secured Mortgage
is a
X2
Loan Security

TH2: Trộn các quan hệ bằng khái quát hóa trùng lặp



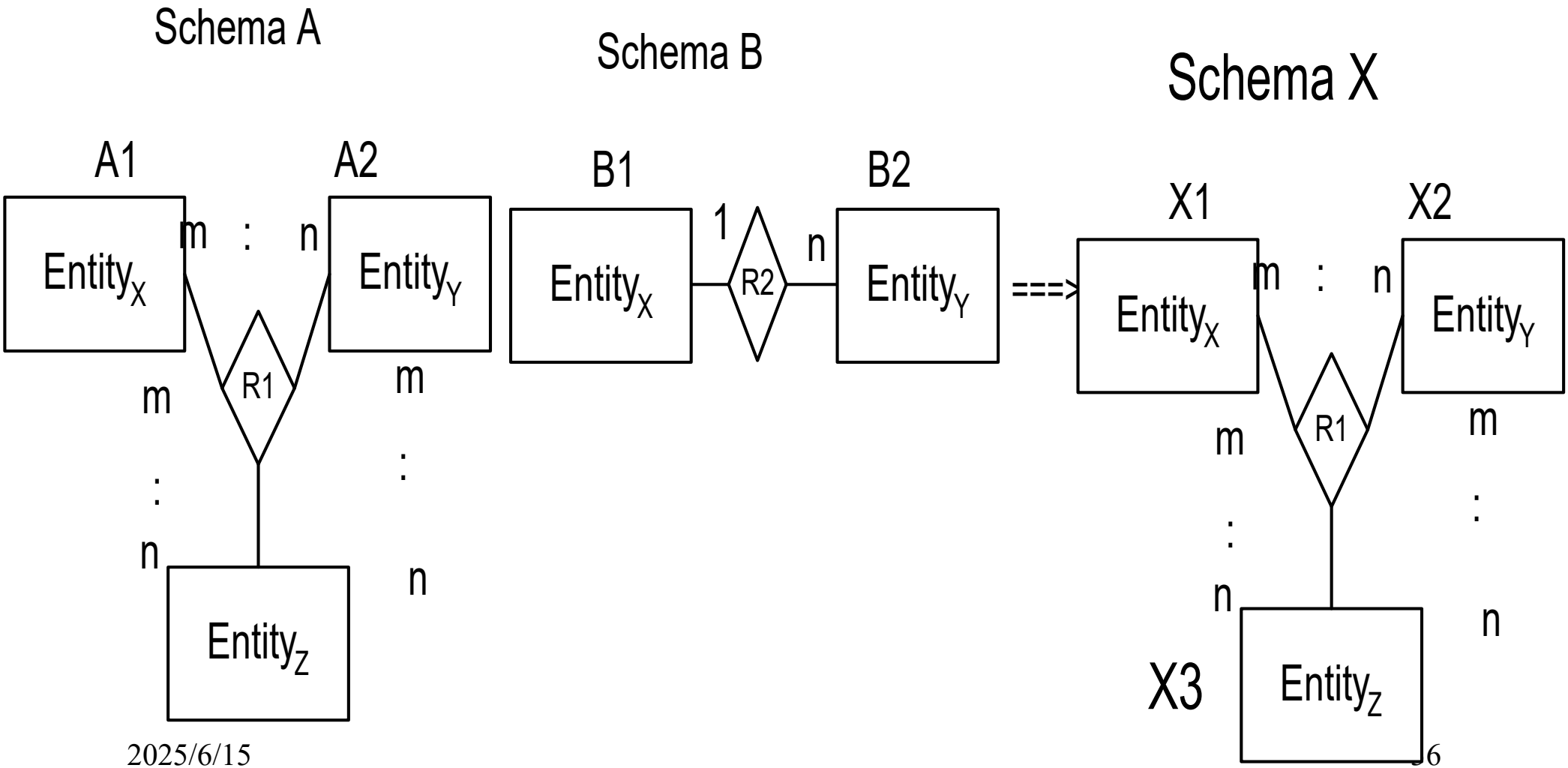
Example



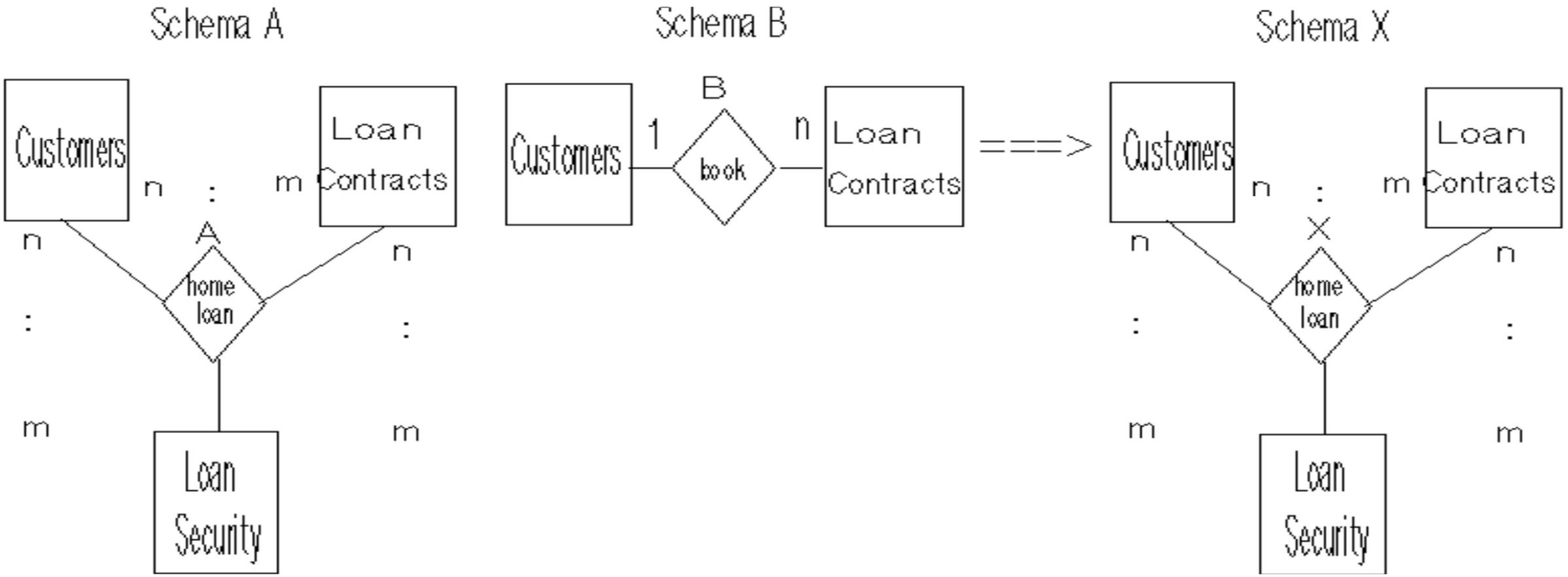
o=overlap generalisation

thực thể Khách hàng được phân thành hai loại thực thể là người vay tiền mua nhà và tập người vay tiền mua ô tô

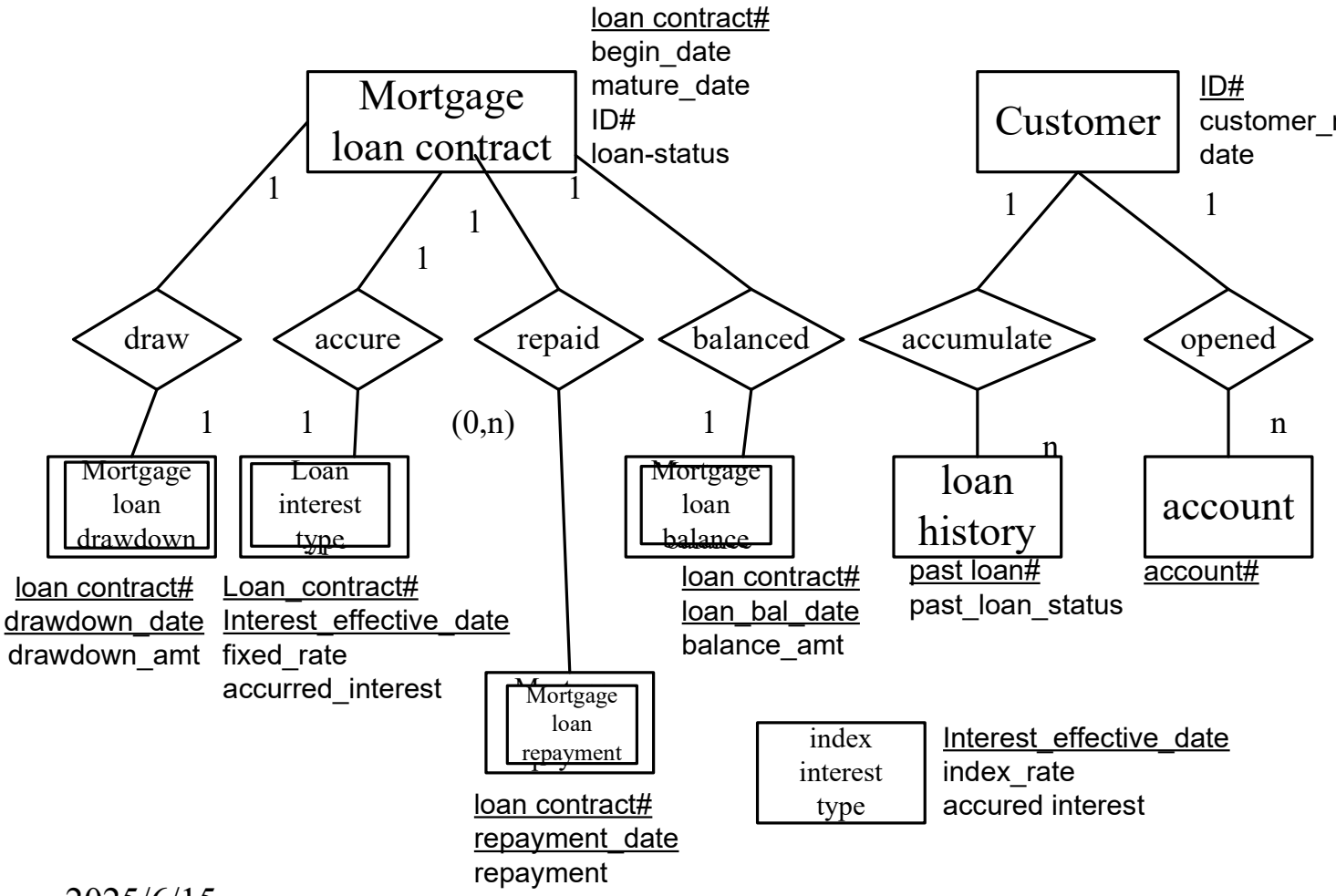
TH3: Sát nhập các quan hệ ít ngôi thành một quan hệ có ngôi cao hơn



Example


Schema A
Customers
n
:
m
Loan Contracts
A
home loan
n
:
m
n
:
m
Loan Security
Schema B
Customers
1
B
book
n
Loan Contracts
===>
Schema X
Customers
n
:
m
Loan Contracts
X
home loan
n
:
m
n
:
m
Loan Security

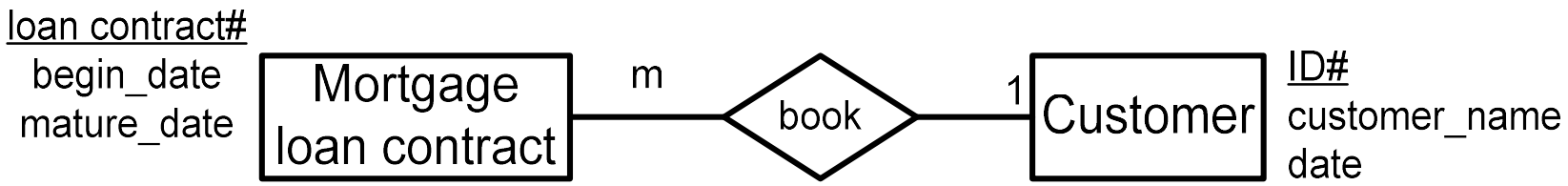
Case Study: Một ngân hàng gồm có các cơ sở dữ liệu với các lược đồ tương ứng sau: một lược đồ khách hàng, một cho các hợp đồng vay nợ mua nhà, và lược đồ thứ ba cho tỉ lệ lãi suất. Chúng ta cần tích hợp chúng thành một hệ thống vay nợ ngân hàng quốc tế.



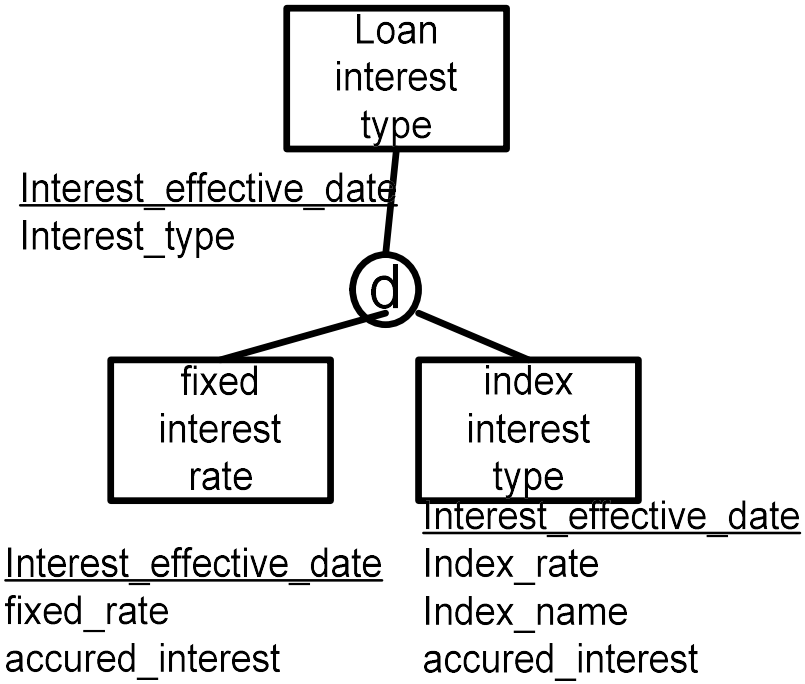
* Loan drawdown: Tiền mỗi lần trả
* interest type: Lãi suất (loại lãi)
* loan repayment: Số lần phải trả
* loan balance: Số tiền còn lại phải trả

Step 2 (trang 26 + 18) Merge entities by Implied binary relationship and Generalization

Since data ID# appears in entity Mortgage Loan Contract and entity Customer, they are in one-to-many cardinality with foreign key on the "many" side.



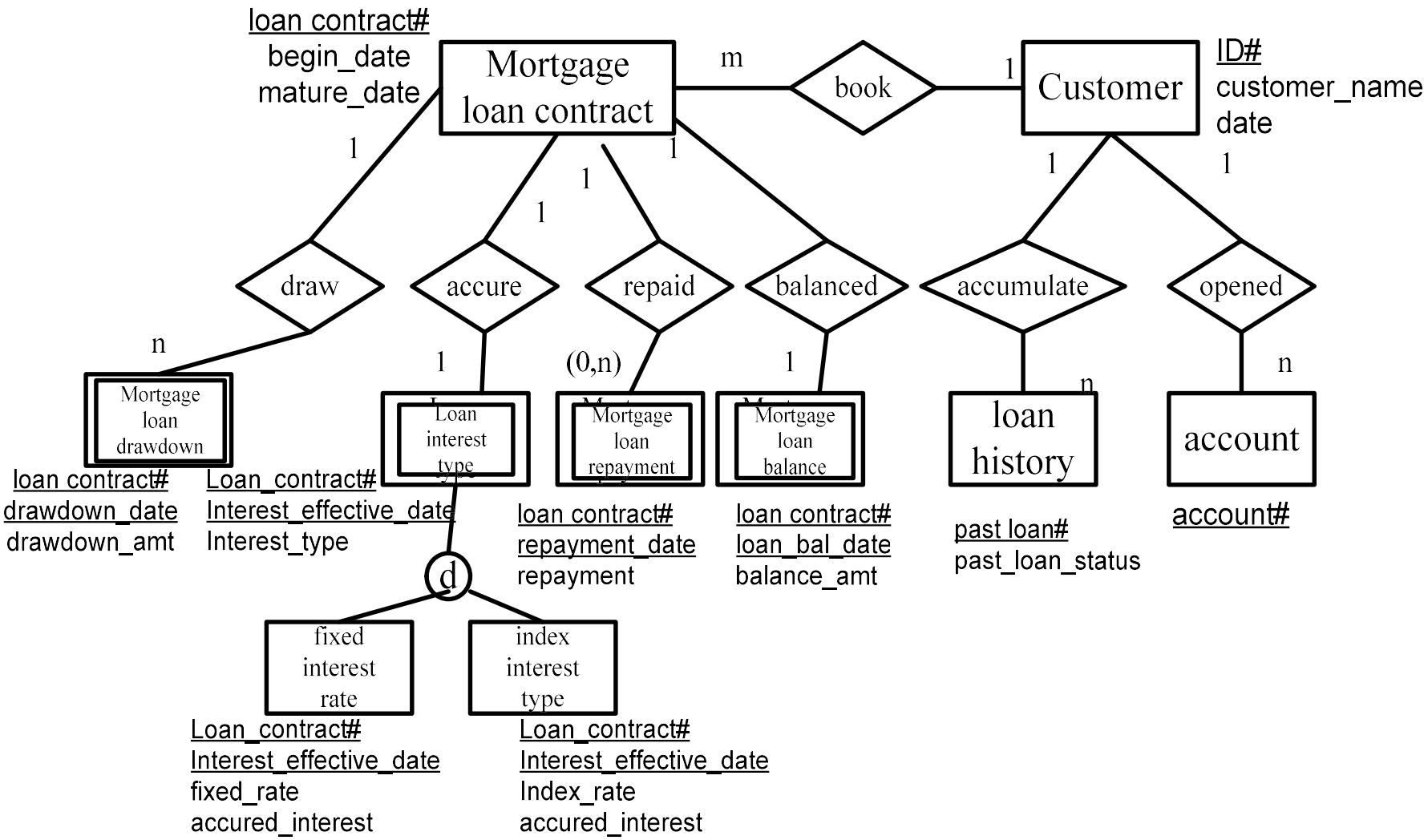
Since Fixed and Index Interest Rate both have the same key, Interest_effective_date, they can be generalized disjointed as either fixed or index rate.



Giải thích quan hệ tạo disjoint giữa các interest rate

- Loan interest type là chỉ loại lãi suất cố định (chứa fixed_rate) chỉ chứa trong hợp đồng vay nợ mua nhà.
- Index interest type là 1 cơ sở dữ liệu chỉ chứa đúng 1 bảng mang tên này, chú ý rằng cơ sở dữ liệu này đã được LƯỢC BỎ rất nhiều bảng và chỉ để lại 1 bảng bao hàm ý nghĩa của tất cả các bảng còn lại, mang ý nghĩa là TẤT CẢ các loại tỉ lệ lãi suất (lưu trữ chung tất cả các loại lãi suất mà ngân hàng này có với tỉ lệ khác nhau, lãi suất cố định (fixed_rate) chỉ là 1 loại tỉ lệ lãi suất có chứa trong đây) (có thể hiểu là lãi suất cho hợp đồng mua nhà khác với lãi suất cho hợp đồng mua xe, index_rate của chúng khác nhau).
- Khi kết nối 2 entity index interest type và loan interest type, để giữ cho quan hệ giữa mortage loan contract và loan interest type vẫn giữ quan hệ mang lãi suất cố định, ta có 1 entity cha mới vẫn để tên là Loan interest type mang nghĩa là tất cả các loại lãi suất mà ngân hàng này quản lý có chung 1 thuộc tính vẫn giữ nguyên ý nghĩa là interest_effective_date.
- Khi đó sinh ra 2 entity con là generalized từ loại lãi suất chung trên bao gồm lãi suất cố định (dành cho mortage loan contract) và các loại lãi suất CÒN LẠI mà ngân hàng quản lý (index interest type).
- Đó là lí do mà Loan interest type có thêm thuộc tính interest_type để phân biệt 2 loại entity này.
- Ngoài ra, index interest type có thêm thuộc tính index_name để phân biệt các loại lãi suất CÒN LẠI đó.

Thus, we can derive cardinality from the implied relationship between these entities, and integrate the two schemas into one EER model.



Reading assignment

"Information Systems Reengineering and Integration" by Joseph Fong, published by Springer Verlag, 2006, pp. 282-310.

Review question 3

Can two Relational Schemas be integrated into one Relational Schema? Explain your answer.

Which steps need user supervision for integrating two Extended Entity Relationship Models into one Extended Entity Relationship Model? Explain your answer.

Tutorial question 3

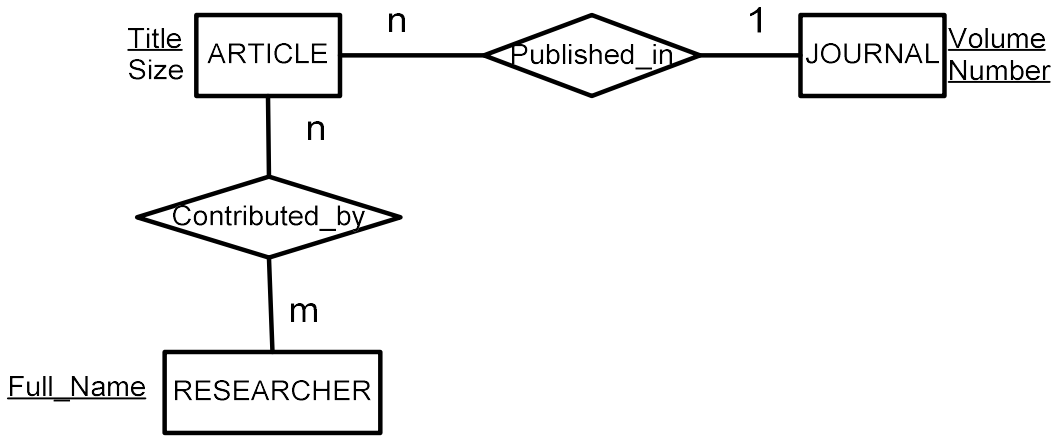
Provide an integrated schema for the following two views which are merged to create a bibliographic database. During identification of correspondences between the two views, the users discover the followings:

RESEARCHER and AUTHOR are synonyms (đồng nghĩa),
CONTRIBUTED_BY and WRITTEN_IN are synonyms,
ARTICLES belongs to a SUBJECT (thuộc về một).
ARTICLES and BOOK can be generalized (khái quát hóa) as PUBLICATION.

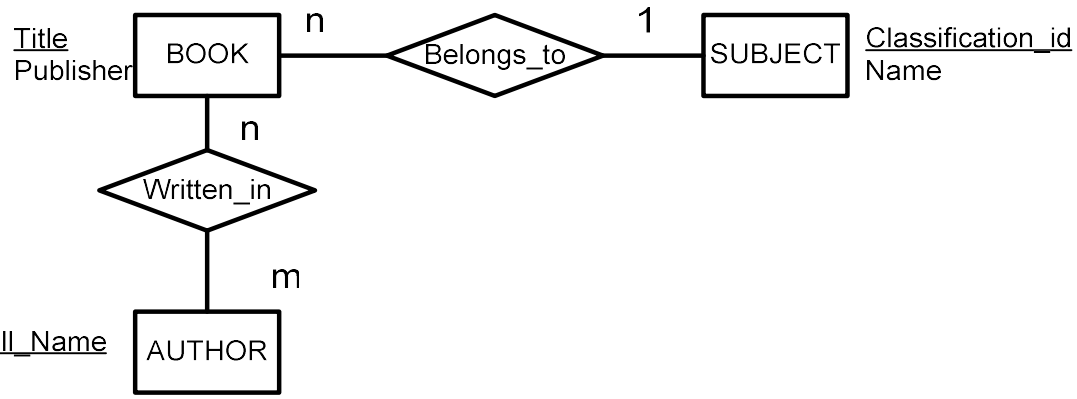
Hints: Given two subclass entities have same relationship(s). The two subclasses entities can be generalized into a

superclass entity and the subclass relationship(s) can also be generalized into a superclass relationship..

View 1



View 2



- Bước 1 (Resolve conflicts):

TH1: RESEARCHER và AUTHOR bản chất giống nhau nhưng tên gọi khác nhau

+ 2 attribute có thuộc tính cùng tên là Full_Name vẫn mang chung ý nghĩa

- Bước 2 (Merge entities):

TH1: Trộn 2 entity AUTHOR cùng tên thành 1.

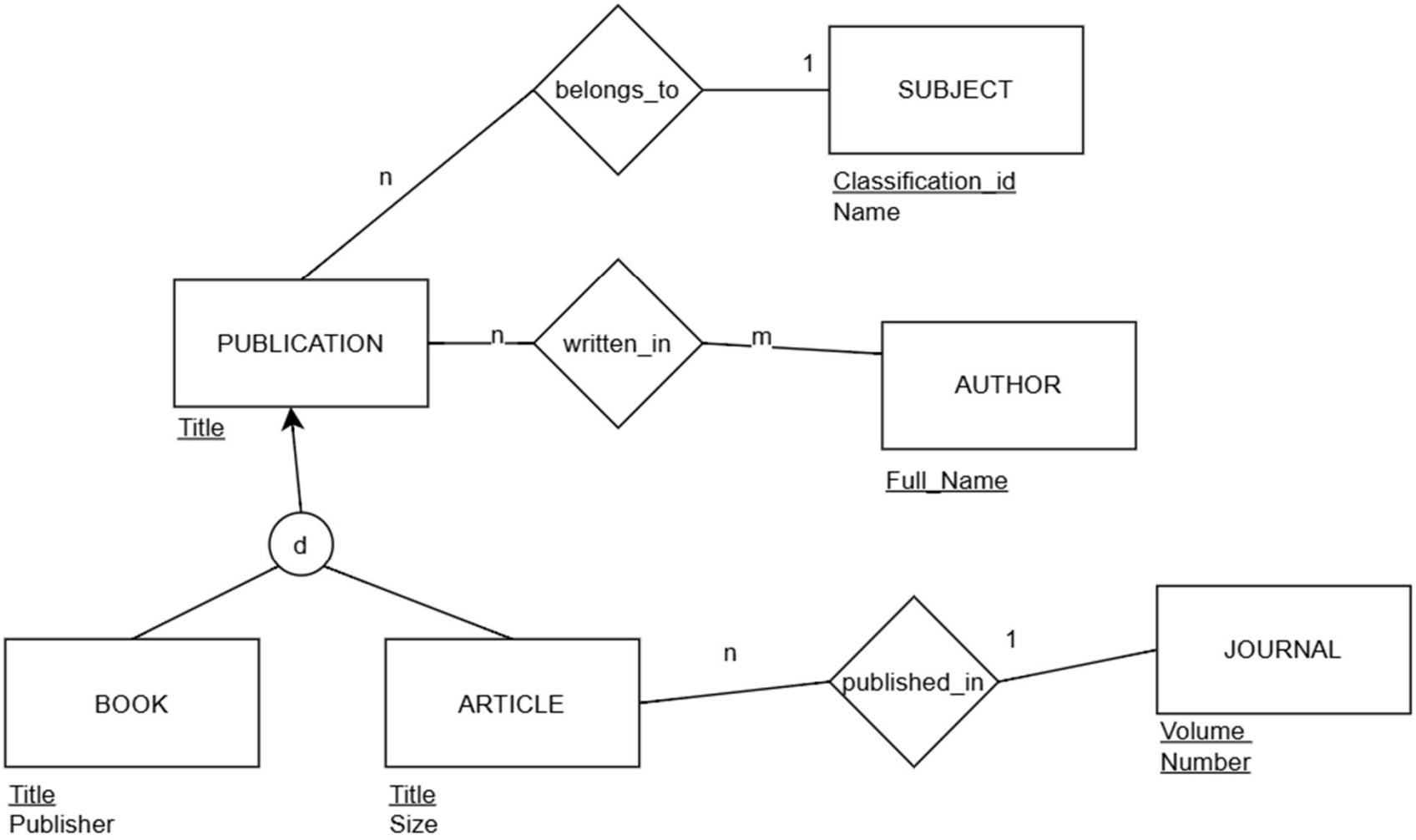
TH2: Trộn ARTICLE và BOOK (cùng khóa Title) bằng cách tổng quát hóa thành PUBLICATION

Sách và bài báo không giao nhau nên để disjoint

- Bước 3 (Merge relationships):

- contributed_by và written_in đồng nghĩa => hợp thành 1 quan hệ giữa AUTHOR với PUBLICATION

- Cả ARTICLE và BOOK đều belongs to a SUBJECT => Chuyển quan hệ belongs_to cho PUBLICATION


belongs_to
1
SUBJECT
Classification_id
Name
n
PUBLICATION
Title
n
written_in
m
AUTHOR
Full_Name
d
BOOK
Title
Publisher
ARTICLE
Title
Size
n
published_in
1
JOURNAL
Volume
Number