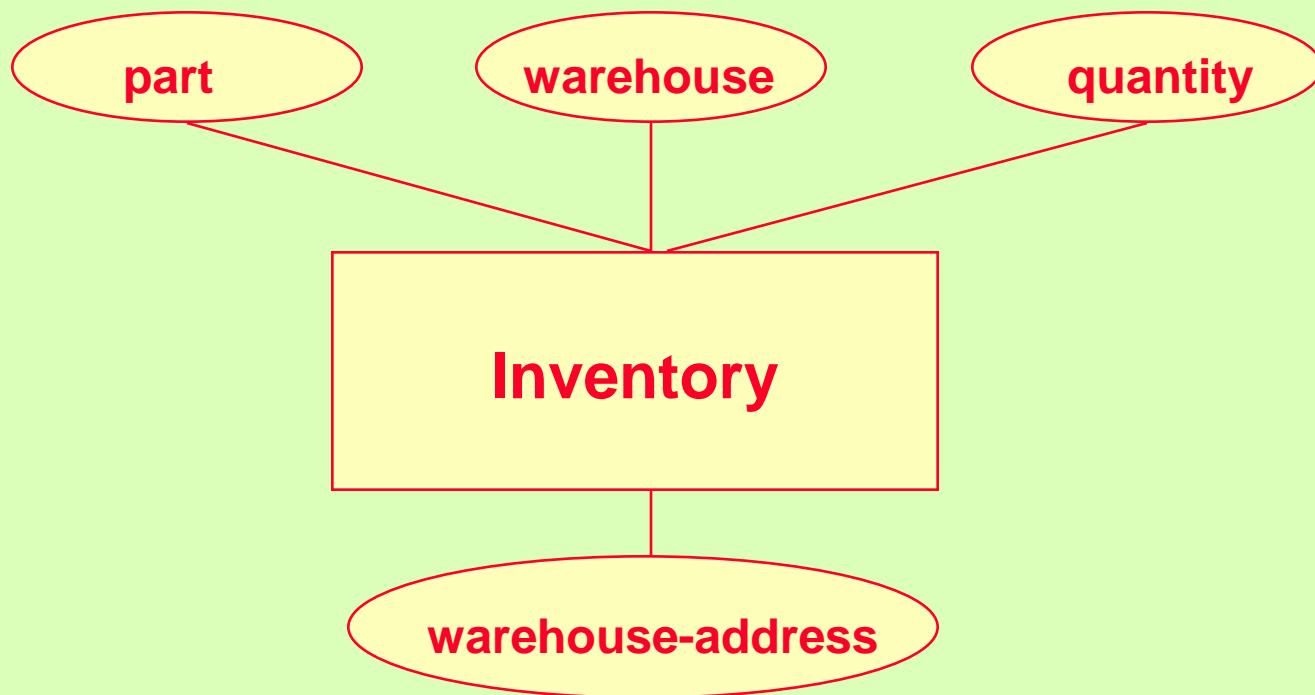


Chuẩn hóa CSDL

Các vấn đề trong các bảng quan hệ

Ví dụ 1:



Chuẩn hóa CSDL

Các vấn đề trong các bảng quan hệ

Inventory

part	warehouse	quantity	warehouse-address
bolt	King Bros.	20	Long St. 20
nut	King Bros.	40	Long St. 20
screw	King Bros.	10	Long St. 20
NULL	Lucid Ltd.	NULL	Compton Pl. 10

The diagram illustrates several anomalies in the Inventory table:

- Partial Update Anomaly:** The row for 'bolt' shows a partial update where only the quantity (20) is updated, while the part name (bolt) and warehouse (King Bros.) remain unchanged.
- Deletion Anomaly:** The row for 'nut' shows a deletion where the part name (nut) and warehouse (King Bros.) are deleted, leaving the quantity (40) as the only remaining data.
- Insertion Anomaly:** The row for 'screw' shows an insertion of a new part ('screw') into an existing warehouse ('King Bros.') without updating the quantity.
- Update Anomaly:** The row for 'NULL' shows an update where the warehouse is changed to 'Lucid Ltd.' and the quantity is set to NULL, while the part name remains NULL.
- Redundancy:** The address 'Long St. 20' is repeated three times in the 'warehouse-address' column, indicating redundancy.

Chuẩn hóa CSDL

Các vấn đề trong các bảng quan hệ

INVENTORY table

- Địa chỉ nhà kho lặp lại trong nhiều bộ chứa mặt hàng có tại nhà kho đó
- Nếu địa chỉ một nhà kho thay đổi, các bộ lưu các mặt hàng chứa trong nhà kho này đều phải thay đổi theo
- Nếu có thời điểm nào đó một nhà kho không có mặt hàng nào thì không có bộ để lưu địa chỉ nhà kho, hoặc cần phải có 1 bộ mà thuộc tính về mặt hàng và số lượng phải NULL

Chuẩn hóa CSDL

Các vấn đề trong các bảng quan hệ

Ví dụ 2:

Institution

Employee Location	Department
Peter 15.101	ITACS
Paul 15.101	ITACS
Kate 15.104	ITACS
Mike 10.001	Math
Jane 10.008	Math

Chuẩn hóa CSDL

Các vấn đề trong các bảng quan hệ

Institution table

- Vị trí của phòng ban lặp lại trong mọi bộ chứa nhân viên của phòng ban đó
- Nếu vị trí phòng ban thay đổi thì mọi bộ lưu nhân viên của phòng ban đó phải được cập nhật
- Nếu một phòng ban chưa có nhân viên thì không có bộ để lưu vị trí của phòng ban hoặc có bộ mà một số thuộc tính phải là NULL

Chuẩn hóa CSDL

Các vấn đề trong các bảng quan hệ

Ví dụ 3:

Insertion test

Employee

emp#	car	
child		
100	TRS200	John
100	TRS200	Kate
100	SWG100	John
100	SWG100	Kate
100	???	Peter

Chuẩn hóa CSDL

Các vấn đề trong các bảng quan hệ

∴ Employee table

- Việc đăng ký số xe phải được lập lại cho từng người con
- Tên của một người con cần được lập lại với mỗi số đăng ký xe (bùng nổ số bộ !)
- Do sự dư thừa dữ liệu, dữ liệu có thể trở nên mất tính nhất quán
- Nếu 1 người không có con (hoặc xe) thì trị thuộc tính con (hoặc xe) cần phải để NULL

Chuẩn hóa CSDL

Các vấn đề trong các bảng quan hệ

Ví dụ 4:

Kết 2 bảng thông qua thuộc tính **city**

Supplier

s#	sname	salary	city
10	Peter	100K	Paris
20	Joan	120K	Rome

Company

cname	city
Golden Bolts	Paris
Red Nuts	Rome
Green Tomatoes	Rome

Chuẩn hóa CSDL

Các vấn đề trong các bảng quan hệ

Supplier JOIN Company

s#	sname	salary	city	cname
10	Peter	100K	Paris	Golden Bolts
20	Joan	120K	Rome	Red Nuts
20	Joan	120K	Rome	Green Tomatoes

A red oval encloses the second and third rows of the table, which both have s# 20 and city Rome. Red arrows point from the cname column of the first row ('Red Nuts') to the cname column of the second row ('Green Tomatoes'). A question mark is located to the right of the oval.

Nhà cung cấp 'Peter' sống và làm việc cho một công ty ở 'Paris'.

Nhà cung cấp 'Joan' sống tại 'Rome' và làm việc cho công ty 'Green Tomatoes' nhưng không làm việc cho công ty 'Red Nuts'

Chuẩn hóa CSDL

Các vấn đề trong các bảng quan hệ

Bảng Supplier và Company

- Kết 2 bảng thông qua thuộc tính **city** có thể sinh ra các bộ biểu diễn thông tin sai

Supplier JOIN Company

s#	sname	salary	city	cname
10	Peter	100K	Paris	Golden Bolts
20	Joan	120K	Rome	Red Nuts
20	Joan	120K	Rome	Green Tomatoes

Chuẩn hóa CSDL

Sự bất thường của phép cập nhật (*Update anomalies*)

Sự bất thường của phép thêm vào (*Insertion anomaly*)

- Việc thêm 1 hàng kéo theo việc thêm các hàng khác (vẫn đề bùng nổ số hàng)

- Việc thêm 1 hàng tạo ra một số thuộc tính NULL

Employee			
emp#	car	child	
100	TRS200	John	
100	TRS200	Kate	
100	SWG100	John	
100	SWG100	Kate	
100	???	Peter	

Chuẩn hóa CSDL

Sự bất thường của phép cập nhật

Sự bất thường của phép loại bỏ (*Deletion anomaly*)

- Việc loại một hàng dẫn đến việc mất thông tin
- Việc xóa bỏ 1 thông tin dẫn đến việc loại nhiều hơn 1 hàng

Institution

Employee	Department	Location
Peter	ITACS	15.101
Paul	ITACS	15.101
Kate	ITACS	15.104
Mike	Math	10.001
Jane	Math	10.008

Chuẩn hóa CSDL

Sự bất thường của phép cập nhật

Sự bất thường của phép sửa đổi (*Modification anomaly*)

- Việc sửa đổi một hàng đòi hỏi việc sửa đổi ở cả một số hàng khác

Employee

emp#	car	
100	TRS200	John
100	TRS200	Kate
100	SWG100	John
100	SWG100	Kate

Chuẩn hóa CSDL

Nguyên tắc chỉ đạo trong thiết kế CSDL

- ∴ Thiết kế các bảng quan hệ sao cho việc giải thích ý nghĩa của nó được dễ dàng. Không kết hợp nhiều thuộc tính không liên quan trong cùng một bảng
- ∴ Thiết kế các bảng quan hệ sao cho tránh được những sự bất thường trong việc thêm, bớt hay sửa đổi các bộ
- ∴ Thiết kế các bảng quan hệ sao cho số thuộc tính có thể NULL là tối thiểu

Chuẩn hóa CSDL

Nguyên tắc chỉ đạo trong thiết kế CSDL

∴ Thiết kế các bảng quan hệ sao cho chúng có thể được kết bởi các điều kiện bằng nhau trên các thuộc tính khóa chính hoặc khóa ngoại, mà không sinh ra các bộ không có trong thực tế

Chuẩn hóa CSDL

Phụ thuộc hàm (*Functional dependency-FD*)

Cho lược đồ quan hệ $R = (A_1, \dots, A_n)$ và 2 tập X, Y khác rỗng là 2 tập con của tập thuộc tính của R .

FD $X \rightarrow Y$ (đọc là: "X xác định hàm Y") là đúng trong R nếu, với mọi thể hiện r của R, không có hai hàng có trùng các trị trên tập thuộc tính X mà có trị khác nhau trên một hoặc hơn các thuộc tính trong tập Y

Chuẩn hóa CSDL

Phụ thuộc hàm

Ví dụ:

**EMPLOYEE(e#,ename, dept_name,
dept_addr, class)**

**Với X = { dept_name }, Y= { dept_addr },
thì FD X -> Y là đúng trong EMPLOYEE**

**Tên phòng ban xác định hàm địa chỉ của
phòng ban,
hoặc,
1 phòng ban có duy nhất 1 địa chỉ**

Chuẩn hóa CSDL

Phụ thuộc hàm

Employee

e#	ename	dept_name	dept_addr	class
10	Peter	Math	G16	2nd
20	Joan	CS	G15	1st
30	Mike	CS	G15	1st
40	Kate	CS	G15	1st
50	Peter	Law	G20	2nd
60	Albert	Physics	G20	1st

e# -> ename

Phụ thuộc hàm
tầm thường,
không bao giờ bị
vi phạm

Chuẩn hóa CSDL

Phụ thuộc hàm

Employee

e#	ename	dept_name	dept_addr	class
10	Peter	Math	G16	2nd
20	Joan	CS	G15	1st
30	Mike	CS	G15	1st
40	Kate	CS	G15	1st
50	Peter	Law	G20	2nd
60	Albert	Physics	G20	1st

e# -> dept_name

Chuẩn hóa CSDL

Phụ thuộc hàm

Employee

e#	ename	dept_name	dept_addr	class
10	Peter	Math	G16	2nd
20	Joan	CS	G15	1st
30	Mike	CS	G15	1st
40	Kate	CS	G15	1st
50	Peter	Law	G20	2nd
60	Albert	Physics	G20	1st

e# -> dept_addr

Chuẩn hóa CSDL

Phụ thuộc hàm

Employee

e#	ename	dept_name	dept_addr	class
10	Peter	Math	G16	2nd
20	Joan	CS	G15	1st
30	Mike	CS	G15	1st
40	Kate	CS	G15	1st
50	Peter	Law	G20	2nd
60	Albert	Physics	G20	1st

e# -> class

Chuẩn hóa CSDL

Phụ thuộc hàm

Employee

e#	ename	dept_name	dept_addr	class
10	Peter	Math	G16	2nd
20	Joan	CS	G15	1st
30	Mike	CS	G15	1st
40	Kate	CS	G15	1st
50	Peter	Law	G20	2nd
60	Albert	Physics	G20	1st

$\text{dept_name} \rightarrow \text{class}$

Chuẩn hóa CSDL

Phụ thuộc hàm

Employee

e#	ename	dept_name	dept_addr	class
10	Peter	Math	G16	2nd
20	Joan	CS	G15	1st
30	Mike	CS	G15	1st
40	Kate	CS	G15	1st
50	Peter	Law	G20	2nd
60	Albert	Physics	G20	1st

e# → ename,
e# → dept_name,
e# → dept_addr,
e# → class,

minimal key → any other attribute

Chuẩn hóa CSDL

Phụ thuộc hàm

Employee

e#	ename	dept_name	dept_addr	class
10	Peter	Math	G16	2nd
50	Peter	Law	G20	2nd
20	Joan	CS	G15	1st
30	Mike	CS	G15	1st
40	Kate	CS	G15	1st
60	Albert	Physics	G20	1st

e# \rightarrow ename,
e# \rightarrow dept_name,
e# \rightarrow dept_addr,
e# \rightarrow class,
dept_name \rightarrow class

but ename \rightarrow e#
but dept_name \rightarrow e#

Chuẩn hóa CSDL

Phụ thuộc hàm

∴ Chú ý rằng:

- $e\# \rightarrow e\#$
- $\text{class} \rightarrow \text{class}$
- $e\#, \text{ename} \rightarrow \text{ename}$
- $\text{if } (e\# \rightarrow \text{ename}) \text{ then } (e\#, \text{dept_name} \rightarrow \text{ename})$
- $\text{if } (e\# \rightarrow \text{ename}) \text{ and } (e\# \rightarrow \text{dept_name}) \text{ and }$
 $(e\# \rightarrow \text{dept_addr}) \text{ then }$
 $e\# \rightarrow \text{ename, dept_name, dept_address}$
- $\text{if } (e\# \rightarrow \text{dept_name}) \text{ and } (\text{dept_name} \rightarrow \text{class}) \text{ then }$
 $e\# \rightarrow \text{class}$

Chuẩn hóa CSDL

Tiên đề Armstrong

\therefore If $X \supseteq Y$ then $X \rightarrow Y$ (reflexivity)

∴ If $X \rightarrow Y$ then $XZ \rightarrow YZ$ (augmentation)

\therefore If $X \rightarrow Y$ and $Y \rightarrow Z$ then $X \rightarrow Z$

(transitivity)

Tập tiên đề đầy đủ và tối thiểu

Chuẩn hóa CSDL

Các quy tắc bổ sung

∴ If $X \rightarrow Y$ and $X \rightarrow Z$ then $X \rightarrow YZ$
(union)

∴ If $X \rightarrow Y$ and $WY \rightarrow Z$ then $WX \rightarrow Z$
(pseudotransitivity)

∴ If $X \rightarrow Y$ and $Y \supseteq Z$ then $X \rightarrow Z$
(decomposition)

Chuẩn hóa CSDL

Dẫn xuất từ các phụ thuộc hàm

Ví dụ:

Given $F = \{A \rightarrow B, B \rightarrow C\}$, is it true that $A \rightarrow C$?

$A \rightarrow B \ \& \ B \rightarrow C$



transitivity axiom

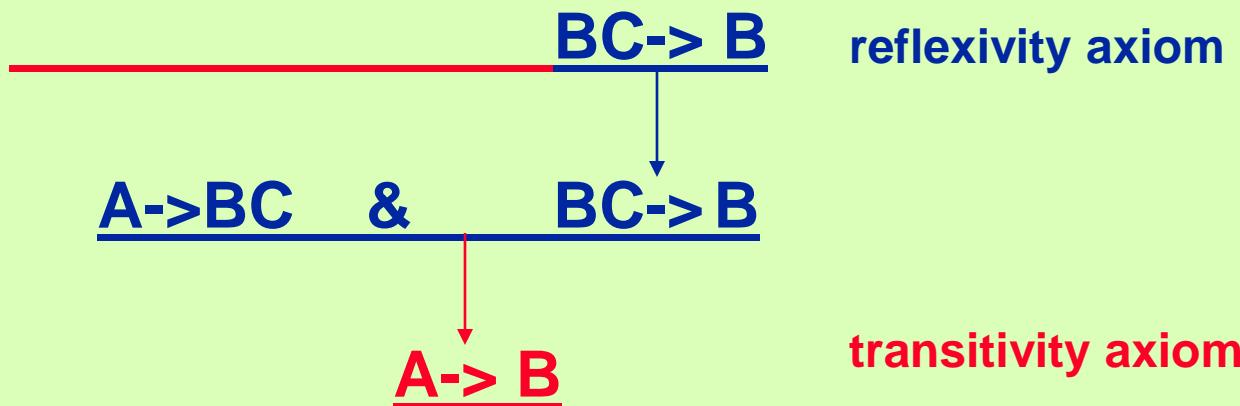
$A \rightarrow C$

Chuẩn hóa CSDL

Dẫn xuất từ các phụ thuộc hàm

Ví dụ:

Given $F = \{A \rightarrow BC\}$,
is it true that $A \rightarrow B$ and $A \rightarrow C$?



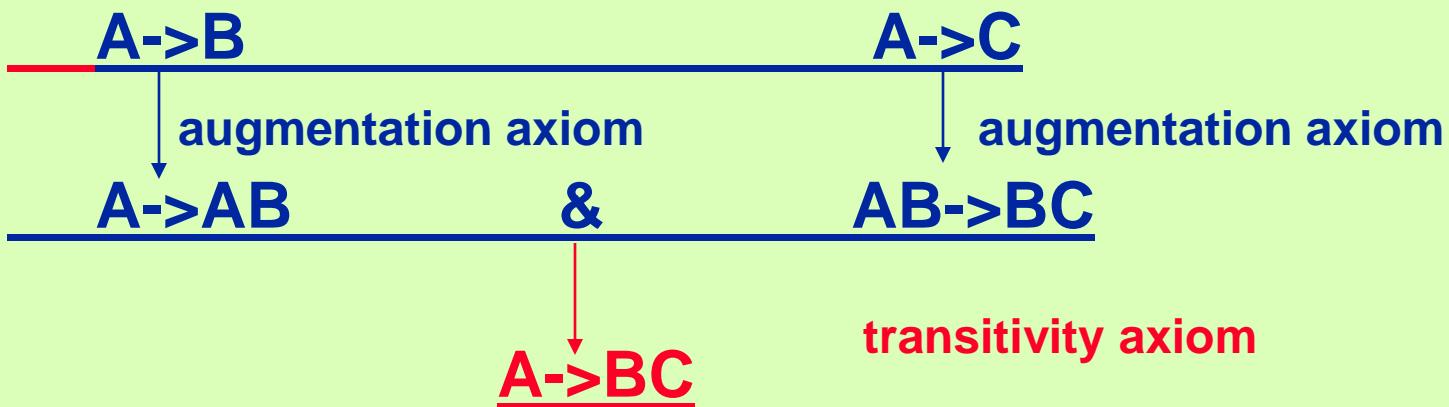
Tương tự, suy ra $A \rightarrow C$

Chuẩn hóa CSDL

Dẫn xuất từ các phụ thuộc hàm

Ví dụ:

Given $F = \{A \rightarrow B, A \rightarrow C\}$,
is it true that $A \rightarrow BC$?

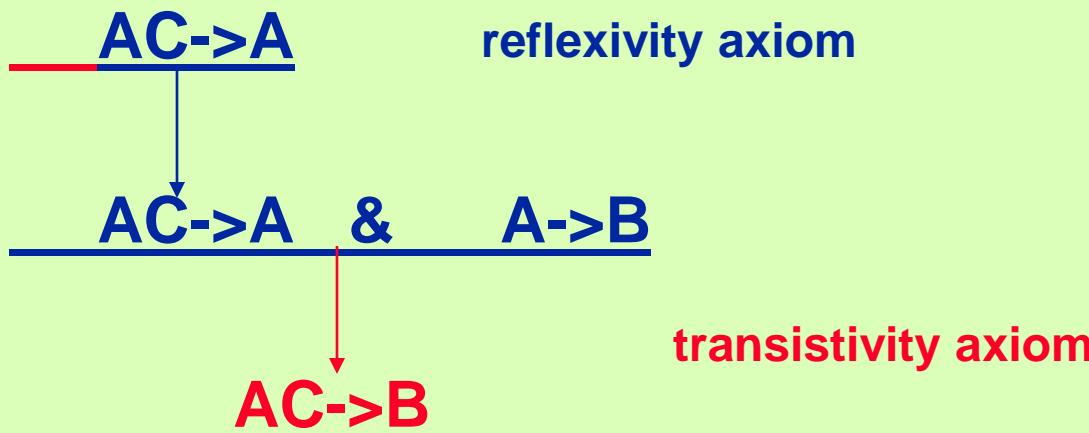


Chuẩn hóa CSDL

Dẫn xuất từ các phụ thuộc hàm

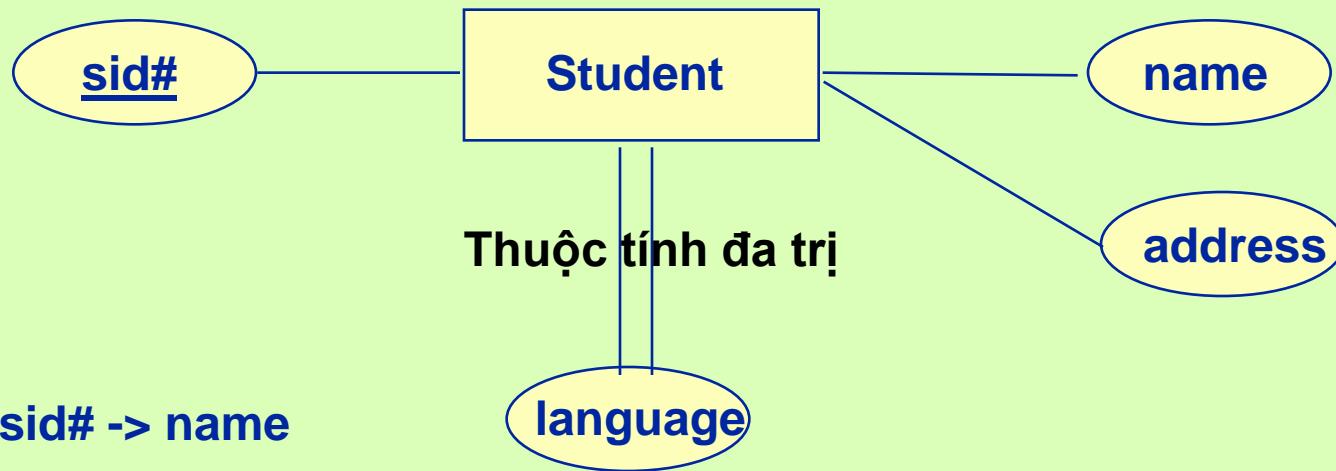
Ví dụ:

Given $F = \{A \rightarrow B\}$,
is it true that $AC \rightarrow B$?



Chuẩn hóa CSDL

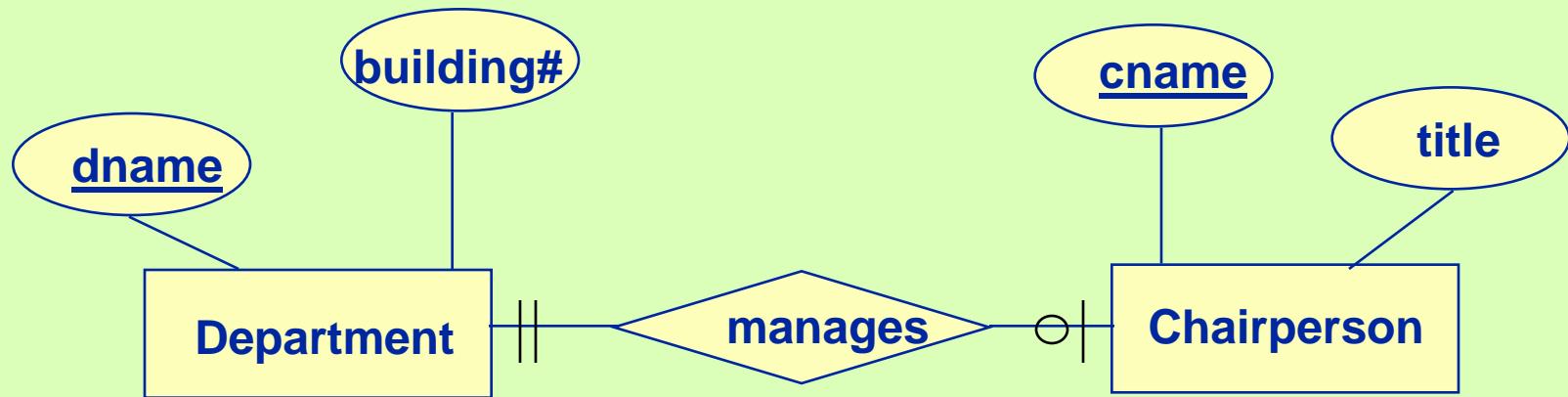
- FD vs. entity



$\text{name} ? \text{sid\#}$
 $\text{name} ? \text{address}$

Chuẩn hóa CSDL

- FD vs. relationship



$dname \rightarrow building\#$

$dname \rightarrow cname$

$cname \rightarrow title$

$cname \rightarrow dname$

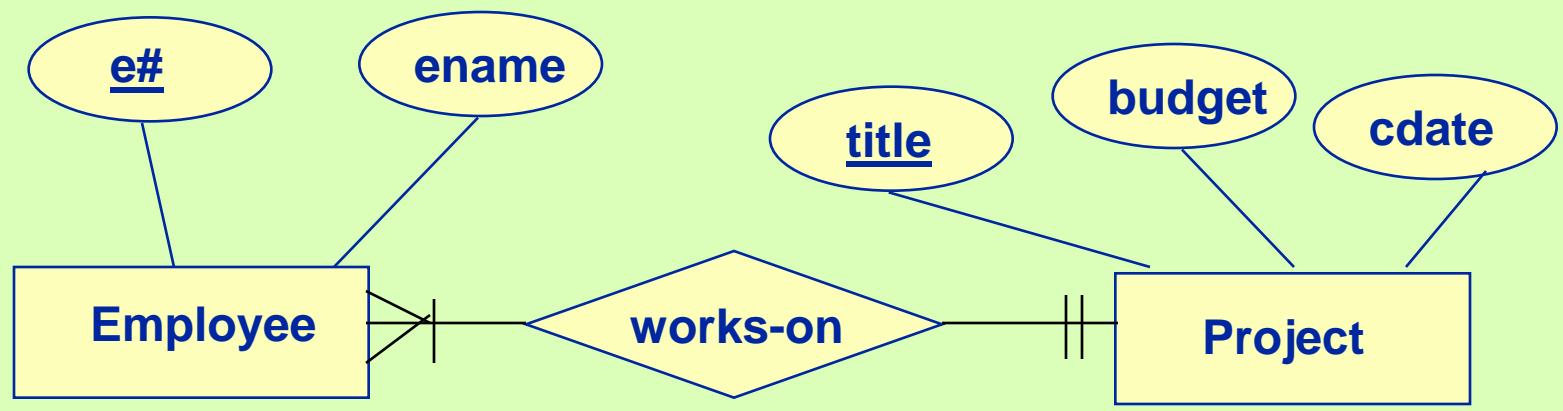
$dname \rightarrow title$

$cname \rightarrow building\#$

$dname, cname \rightarrow building\#, title$

Chuẩn hóa CSDL

- Functional dependency vs. relationship



$e\# \rightarrow \text{ename}$

$e\# \rightarrow \text{title}$

$\text{title} \rightarrow \text{budget}$

$\text{title} \rightarrow \text{cdate}$

$e\# \rightarrow \text{ename, title}$

$e\# \rightarrow \text{budget}$

$e\# \rightarrow \text{cdate}$

Chuẩn hóa CSDL

Dạng chuẩn 1 (*First Normal Form - 1NF*)

Lược đồ quan hệ đạt dạng chuẩn 1 nếu mọi hàng trong bảng quan hệ tương ứng có cùng số thuộc tính và chỉ chứa các trị nguyên tố, có nghĩa là không có sự lặp lại các thuộc tính và các nhóm trị

Ví dụ 1: lược đồ quan hệ sau không đạt dạng chuẩn 1

Family

person_name	children	cars
Peter	{ Mike Joan},	{PPP777}
Paul	{}	{ KGN004, MN3345}

Chuẩn hóa CSDL

Khóa (Keys)

∴ Siêu khóa (Superkey)

Siêu khóa là một tập con khác rỗng các thuộc tính của lược đồ quan hệ R sao cho với bất kỳ 2 bộ t_1, t_2 trong bảng quan hệ định nghĩa trên R thì $t_1[X] \neq t_2[X]$

Lưu ý rằng khi X là siêu khóa trong R thì X xác định hàm mọi thuộc tính của R: $X \rightarrow R$

∴ Khóa tối thiểu (Minimal key) Khóa dự tuyển là khóa tối thiểu

Khóa tối thiểu K là siêu khóa kèm thêm tính chất là nếu loại khỏi K bất kỳ thuộc tính nào cũng làm cho K không còn là siêu khóa

Chuẩn hóa CSDL

Khóa

∴ Khóa chính (Primary key)

Khóa chính là một khóa tối thiểu được người phân tích chọn để cài đặt

∴ Khóa dự tuyển (Candidate key)

Các khóa dự tuyển là các khóa tối thiểu khác mà không phải là khóa chính

Chuẩn hóa CSDL

Thuộc tính

∴ Thuộc tính nguyên tố (*Prime attribute*)

Thuộc tính nguyên tố là một thuộc tính của lược đồ quan hệ R mà là thành phần của ít nhất một khóa dự tuyển

∴ Thuộc tính không nguyên tố (*Nonprime attribute*)

Thuộc tính không nguyên tố là thuộc tính không phải là thuộc tính nguyên tố

Chuẩn hóa CSDL

**Phụ thuộc hàm đầy đủ (*full FD*) và
phụ thuộc hàm riêng phần (*partial FD*)**

∴ **Phụ thuộc hàm đầy đủ**

Phụ thuộc hàm đầy đủ là phụ thuộc hàm $X \rightarrow Y$ mà
nếu loại bất kỳ thuộc tính A nào trong X thì
 $(X-A) \text{ not} \rightarrow Y$

∴ **Phụ thuộc hàm riêng phần**

Phụ thuộc hàm riêng phần là phụ thuộc hàm mà
không phải là phụ thuộc hàm đầy đủ

Chuẩn hóa CSDL

Dạng chuẩn 2 (*Second Normal Form -2NF*)

Lược đồ quan hệ đạt dạng chuẩn 2 nếu mọi thuộc tính không nguyên tố A trong R phụ thuộc hàm đầy đủ vào khóa chính của R

Ví dụ 2:

Inventory			
part	warehouse	quantity	warehouse-address
bolt	King Bros.	20	Long St. 20
nut	King Bros.	40	Long St. 20
screw	King Bros.	10	Long St. 20
NULL	Lucid Ltd.	NULL	Compton Pl. 10

Auntnity phụ thuộc hàm toàn phần vào khóa, còn ware-add FD phụ thuộc riêng phần vào part, warehouse -> quantity

Part,warehouse là khóa

Việc cập nhật nếu chỉ kiểm tra khóa chính thôi mà không kiểm tra phụ thuộc hàm thì sẽ bị mất tính nhất quán dữ liệu

Chuẩn hóa CSDL

Dạng chuẩn 2

INVENTORY (part, warehouse, quantity, warehouse-address)

$F = \{ \text{part, warehouse} \rightarrow \text{quantity},$

$\text{warehouse} \rightarrow \text{warehouse-address} \}$

Khóa : ???

F^+ là tập bao đóng của F , là F và các dẫn xuất theo định luật Armstrong

Vì $\text{warehouse} \rightarrow \text{warehouse-address}$,

$\Rightarrow \text{part, warehouse} \rightarrow \text{warehouse-address}$

Khóa : part, warehouse

INVENTORY không đạt 2NF vì có warehouse-address (thuộc tính không nguyên tố) không phụ thuộc đầy đủ vào khóa Cách giải thích trong các bài tập về dạng chuẩn

Chuẩn hóa CSDL

Dạng chuẩn 2

Lược đồ **INVENTORY** nên được phân rã thành các lược đồ sau:

P (part, warehouse, quantity)

F = { part, warehouse -> quantity }

Khóa: part, warehouse

2NF

W (warehouse, warehouse-address)

F = { warehouse -> warehouse-address }

Khóa: warehouse

2NF

Chuẩn hóa CSDL

Phụ thuộc hàm bắc cầu (*Transitive FD*)

Phụ thuộc hàm $X \rightarrow Y$ đúng trong R là bắc cầu nếu tồn tại tập con khác rỗng Z của R , Z không phải là tập con của một khóa nào trong R , và các phụ thuộc hàm $X \rightarrow Z$ và $Z \rightarrow Y$ là đúng trong R

Đọc là: “ Y phụ thuộc bắc cầu vào X trong lược đồ R” nếu $X \rightarrow Y$ là đúng trong R và $X \rightarrow Y$ là phụ thuộc hàm bắc cầu

Chuẩn hóa CSDL

Dạng chuẩn 3 (*Third Normal Form - 3NF*)

Lược đồ quan hệ R đạt dạng chuẩn 3 nếu R đạt dạng chuẩn 2 và không có một thuộc tính không nguyên tố nào của R phụ thuộc bắc cầu vào khóa chính của R

Định nghĩa khác:

Lược đồ quan hệ R đạt 3NF nếu với mọi FD $X \rightarrow A$ đúng trong R thì:

(i) X là siêu khóa trong R,
hoặc

(ii) A là thuộc tính nguyên tố trong R

Chuẩn hóa CSDL

Dạng chuẩn 3

Ví dụ 3:

SUPPLIER (s#, sname, cname, city)

$F = \{ s\# \rightarrow sname,$

$s\# \rightarrow cname, cname \rightarrow city, s\# \rightarrow city \}$

Khóa : s#

2NF

Lược đồ SUPPLIER không đạt 3NF vì thuộc tính city phụ thuộc bắc cầu vào s#

Chuẩn hóa CSDL

Dạng chuẩn 3

Lược đồ SUPPLIER nên được phân rã thành các lược đồ sau:

S (s#, **sname**, **cname**)

F= { s# -> sname, s# -> cname }

Khóa : s#

3NF

C (**cname**, city)

F= { cname -> city }

Khóa : cname

3NF

Chuẩn hóa CSDL

Dạng chuẩn 3

Ví dụ 4:

LOCATION (city, street, zip-code)

$F = \{ \text{city}, \text{street} \rightarrow \text{zip-code},$

zip-code \rightarrow city

Khóa₁ : city, street

Khóa₂ : street, zip-code

Lược đồ LOCATION đạt 3NF vì thuộc tính

city ở vế phải của FD **zip-code \rightarrow city** là

thuộc tính nguyên tố

Chuẩn hóa CSDL

Dạng chuẩn 3

LOCATION (city, street, zip-code)

Location

city	street	zip-code
NY	55th	484
NY	56th	484
LA	55th	473
LA	56th	473
LA	57th	474

Vấn đề còn tồn tại:

Sự lặp lại của <LA ... 473> và <NY ... 484>, bị ràng buộc bởi FD:
zip-code -> city
là dư thừa trong bảng LOCATION

Chuẩn hóa CSDL

Dạng chuẩn Boyce-Codd *(Boyce-Codd Normal Form - BCNF)*

Lược đồ quan hệ R đạt BCNF nếu với mọi FD
X->A đúng trong R thì X là siêu khóa trong R

Chuẩn hóa CSDL

Dạng chuẩn Boyce-Codd

Ví dụ 5: SCL (student, course, lecturer)

$F = \{ \text{student, course} \rightarrow \text{lecturer},$
 $\text{lecturer} \rightarrow \text{course} \}$

Khóa₁ : student, course; **Khóa₂**: student, lecturer

Lược đồ quan hệ SCL không đạt BCNF vì thuộc tính lecturer không là siêu khóa

Lược đồ quan hệ SCL đạt 3NF vì thuộc tính **course** ở
về phải của FD **student, course → lecturer** là
thuộc tính nguyên tố

Chuẩn hóa CSDL

Dạng chuẩn Boyce-Codd

Lược đồ quan hệ

SCL (student, course, lecturer)

nên được phân rã thành các lược đồ sau:

S (student, course)

Khóa: student, course

BCNF

L (lecturer, course)

F= { lecturer -> course }

Khóa: lecturer

BCNF

Chuẩn hóa CSDL

Chuẩn hóa các lược đồ quan hệ

Cho lược đồ quan hệ R (A₁, ..., A_n)

1. Xác định các phụ thuộc hàm
2. Xác định các khóa tối thiểu
3. Xác định dạng chuẩn cao nhất mà R thỏa
4. Phân rã R thành các lược đồ sao cho mỗi lược đồ đạt BCNF

Chuẩn hóa CSDL

Chuẩn hóa các lược đồ quan hệ

Ví dụ 6:

S (s#, city, status, p#, qty)

$F = \{ s\# \rightarrow \text{city},$
 $s\# \rightarrow \text{status},$
 $\text{city} \rightarrow \text{status},$
 $s\#, p\# \rightarrow \text{qty},$
 $s\#, p\# \rightarrow \text{city},$
 $s\#, p\# \rightarrow \text{status} \}$

Khóa : s#, p#

1NF

not 2NF

Chuẩn hóa CSDL

Chuẩn hóa các lược đồ quan hệ

S (s#, city, status, p#, qty)

Phân rã S thành các lược đồ sau

1NF

SP (s#, p#, qty)

F = { $s\#, p\# \rightarrow qty$ }

BCNF

khóa: s#, p#

SUPPLIER (s#, city, status)

F = { $s\# \rightarrow city$, $city \rightarrow status$ }, $s\# \rightarrow status$ }

2NF

khóa: s#

Chuẩn hóa CSDL

Chuẩn hóa các lược đồ quan hệ

Tiếp tục phân rã

SUPPLIER (s#, city, status)

$F = \{ s\# \rightarrow city, s\# \rightarrow status, city \rightarrow status \}$

2NF

thành:

SUPPLIER' (s#, city)

$F = \{ s\# \rightarrow city \}$

BCNF

LOCATION (city, status)

$F = \{ city \rightarrow status \}$

BCNF

Chuẩn hóa CSDL

Chuẩn hóa các lược đồ quan hệ

Ví dụ 7: **SSC** (supplier, status, city)

$F = \{ \text{supplier} \rightarrow \text{status},$
 $\text{city} \rightarrow \text{status}, \text{supplier} \rightarrow \text{city} \}$

Khóa: supplier

2NF

Phụ thuộc bắc cầu

➤ *2 cách phân rã có thể:*

S (supplier, city)

$F = \{\text{supplier} \rightarrow \text{city}\}$

C (city, status)

$G = \{\text{city} \rightarrow \text{status}\}$

BCNF

S (supplier, status)

$F = \{\text{supplier} \rightarrow \text{status}\}$

T (supplier, city)

$G = \{\text{supplier} \rightarrow \text{city}\}$

BCNF

Chuẩn hóa CSDL

Phụ thuộc đa trị (*Multivalued dependency -MVD*)

- ∴ Cho X, Y là 2 tập con khác rỗng của lược đồ quan hệ R
- ∴ Phụ thuộc đa trị $X \rightarrow\!\!\!-\> Y \sqcup Z$ là đúng trong R nếu với mỗi 2 bộ v và w trong bảng quan hệ $r(R)$ mà $v[X] = w[X]$ thì tồn tại một bộ t trong $r(R)$ sao cho:

$$\left. \begin{array}{l} (i) \quad v[XY] = t[XY] \\ (ii) \quad w[XZ] = t[XZ] \end{array} \right\} \longrightarrow$$

v:	X	Y	_
w:	X	_	Z
t:	X	Y	Z

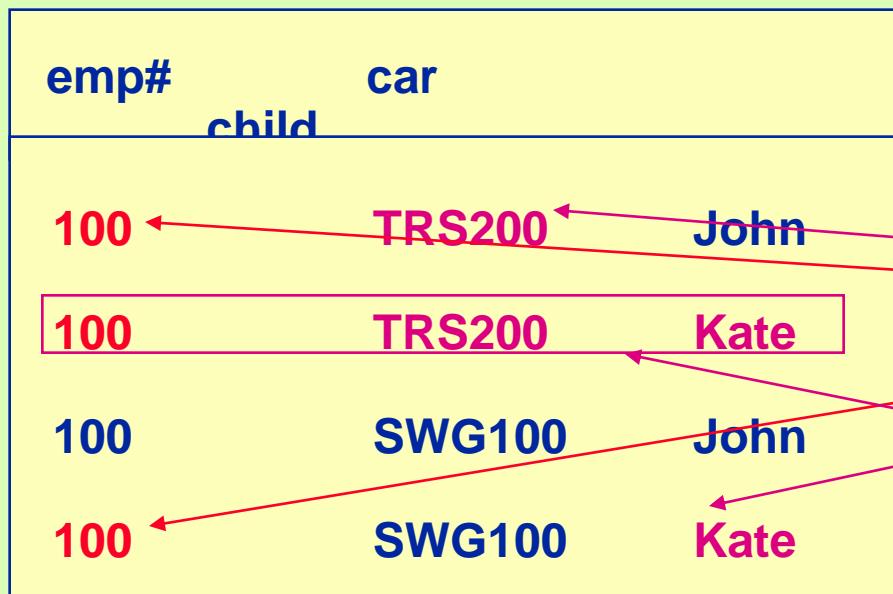
Khi đó, $\pi_{XY}(r) \text{ JOIN } \pi_{XZ}(r) = r$

Chuẩn hóa CSDL

Phụ thuộc đa trị

Ví dụ 8:

Employee



MVD:

$\text{emp\#} \rightarrow\!\!> \text{car} \mid \text{child}$

or

emp\# car

$\underline{\text{emp\#}}$ $\underline{\text{child}}$

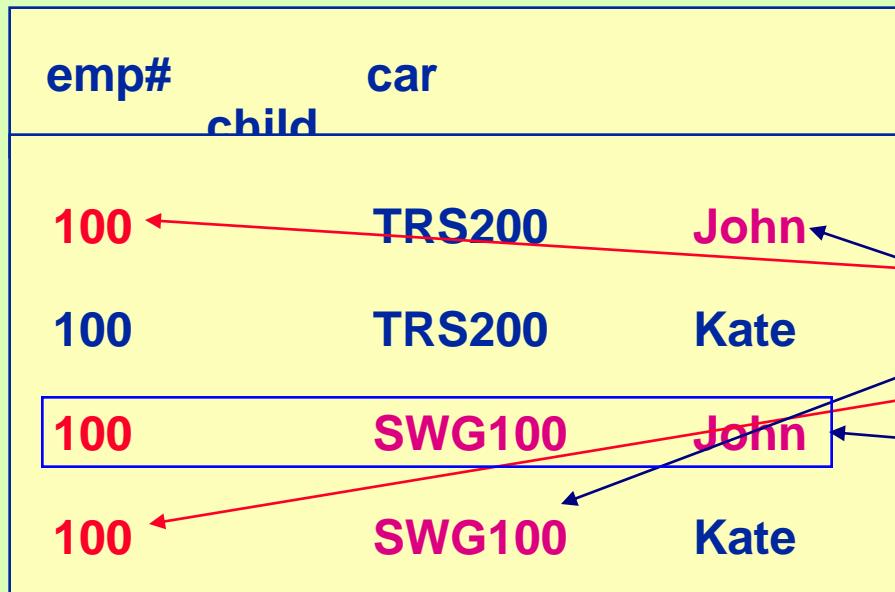
emp\# car child

Chuẩn hóa CSDL

Phụ thuộc đa trị

Ví dụ 8:

Employee



MVD:

$\text{emp\#} \rightarrow\!-\!\!-\! \text{car} \mid \text{child}$

or

$\text{emp\#} \text{ car}$

$\text{emp\#} \text{ child}$

emp\# car child

Chuẩn hóa CSDL

Dạng chuẩn 4 (*Fourth Normal Form -4NF*)

Lược đồ quan hệ R đạt dạng chuẩn 4 nếu với
mọi phụ thuộc đa trị không tầm thường

$X \rightarrow\!\!\!-\> Y \mid Z$

thì X là siêu khóa của R

Ví dụ 9:

EMPLOYEE (emp#, car, child)

MVD = { $\text{emp\#} \rightarrow\!\!\!-\> \text{car} \mid \text{child}$ }

BCNF

Không có FD nào trong R, MVD trên là không tầm thường,
 emp\# không là siêu khóa, EMPLOYEE không đạt 4NF

Chuẩn hóa CSDL

Dạng chuẩn 4

EMPLOYEE (emp#, car, child)

nên phân rã thành các lược đồ sau:

S1 (emp#, car)

BCNF

4NF

S2 (emp#, child)

4NF

Chuẩn hóa CSDL

- 3NF versus BCNF

Ví dụ 1:

Address (City, Street, ZipCode)

$F = \{ \text{ZipCode} \rightarrow \text{City},$
 $\text{City, Street} \rightarrow \text{ZipCode} \}$

Keys:

K1 : City, Street

: City,Street \rightarrow ZipCode

K2 : ZipCode, Street

: ZipCode \rightarrow City

ZipCode, Street \rightarrow City

Chuẩn hóa CSDL

- 3NF versus BCNF

Address (City, Street, ZipCode)

$$F = \{ \boxed{\text{ZipCode} \rightarrow \text{City}}, \\ \text{City, Street} \rightarrow \text{ZipCode} \}$$

K1 : City, Street

K2 : ZipCode, Street

ZipCode không phải
là khóa tối thiểu

not BCNF

Chuẩn hóa CSDL

- 3NF versus BCNF

Address (City, Street, ZipCode)

$$F = \{ \boxed{\text{ZipCode} \rightarrow \text{City}}, \text{City, Street} \rightarrow \text{ZipCode} \}$$

K1 : City, Street
K2 : ZipCode, Street

City là thành phần
của khóa tối thiểu

3NF

Chuẩn hóa CSDL

- 3NF versus BCNF

Address (City, Street, ZipCode)

$F = \{ \text{ZipCode} \rightarrow \text{City},$
 $\text{City, Street} \rightarrow \text{ZipCode} \}$

- Phân rã Address thành (City, Street) và (City, ZipCode) sẽ không giữ được phụ thuộc hàm City, Street \rightarrow ZipCode
- Không thể phân rã Address để đạt BCNF mà vẫn bảo toàn phụ thuộc hàm

Chuẩn hóa CSDL

- 3NF versus BCNF

Address (City, Street, ZipCode)

$F = \{ \text{ZipCode} \rightarrow \text{City},$
 $\text{City, Street} \rightarrow \text{ZipCode} \}$

Lời giải:

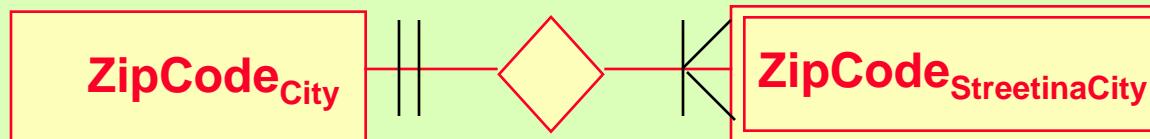
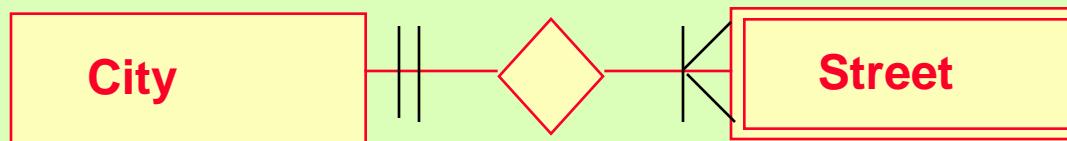
– ZipCode được tạo thành bởi:

ZipCode ($\text{Zip}_{\text{CodeCity}}$, $\text{Zip}_{\text{CodeStreet in a City}}$)

– Cấu trúc của ZipCode đồng dạng với cấu trúc của (City, Street)

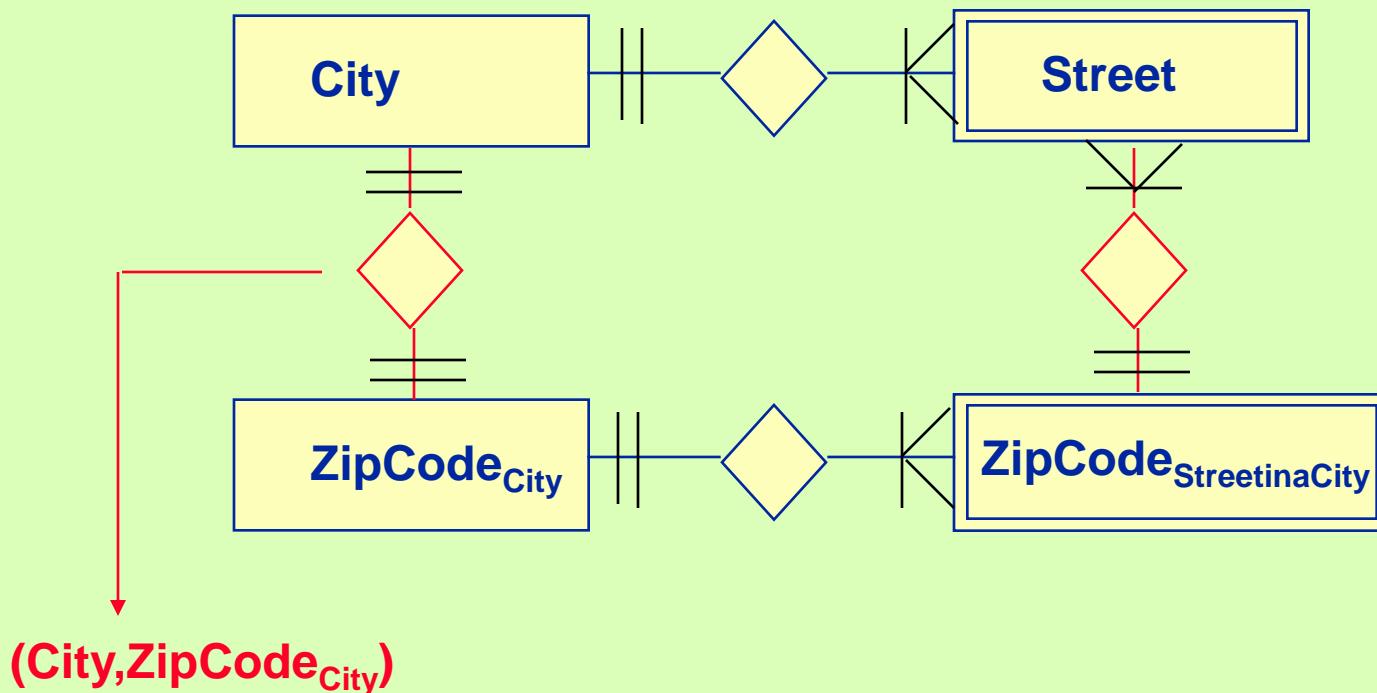
Chuẩn hóa CSDL

- 3NF versus BCNF



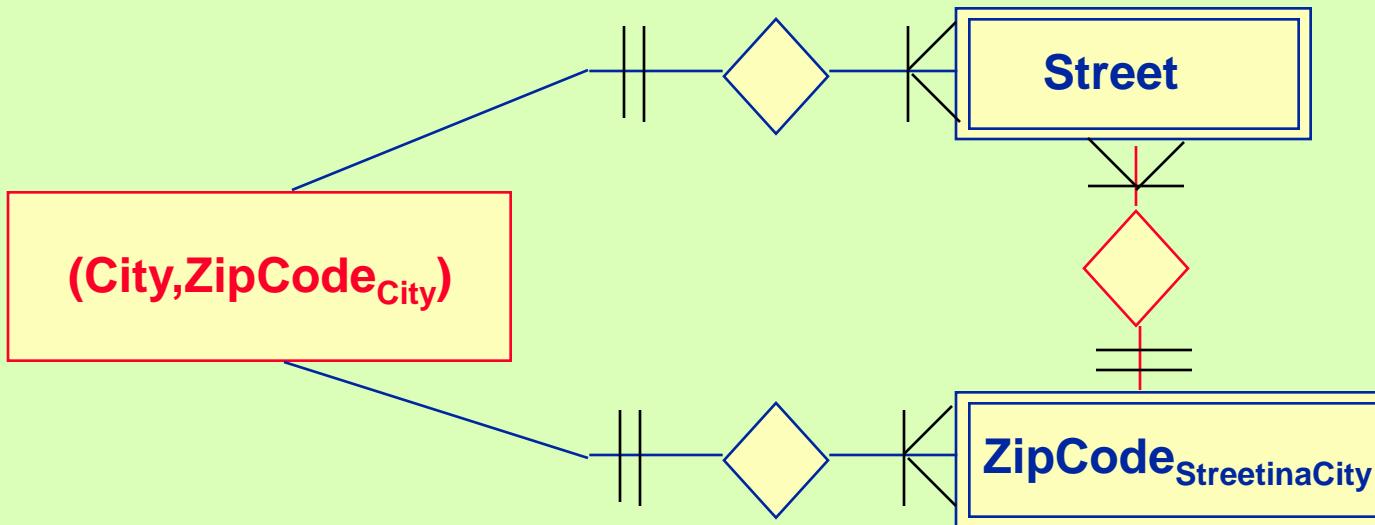
Chuẩn hóa CSDL

- 3NF versus BCNF



Chuẩn hóa CSDL

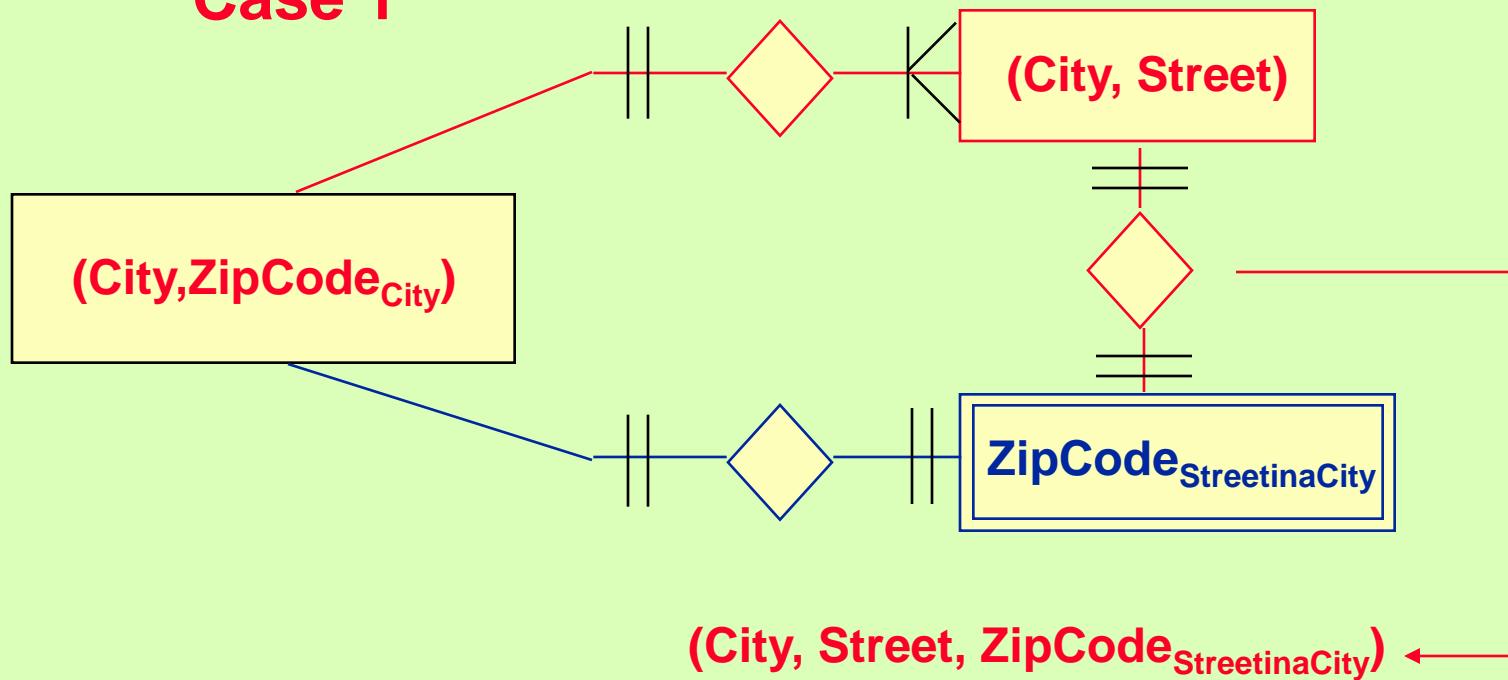
- 3NF versus BCNF



Chuẩn hóa CSDL

- 3NF versus BCNF

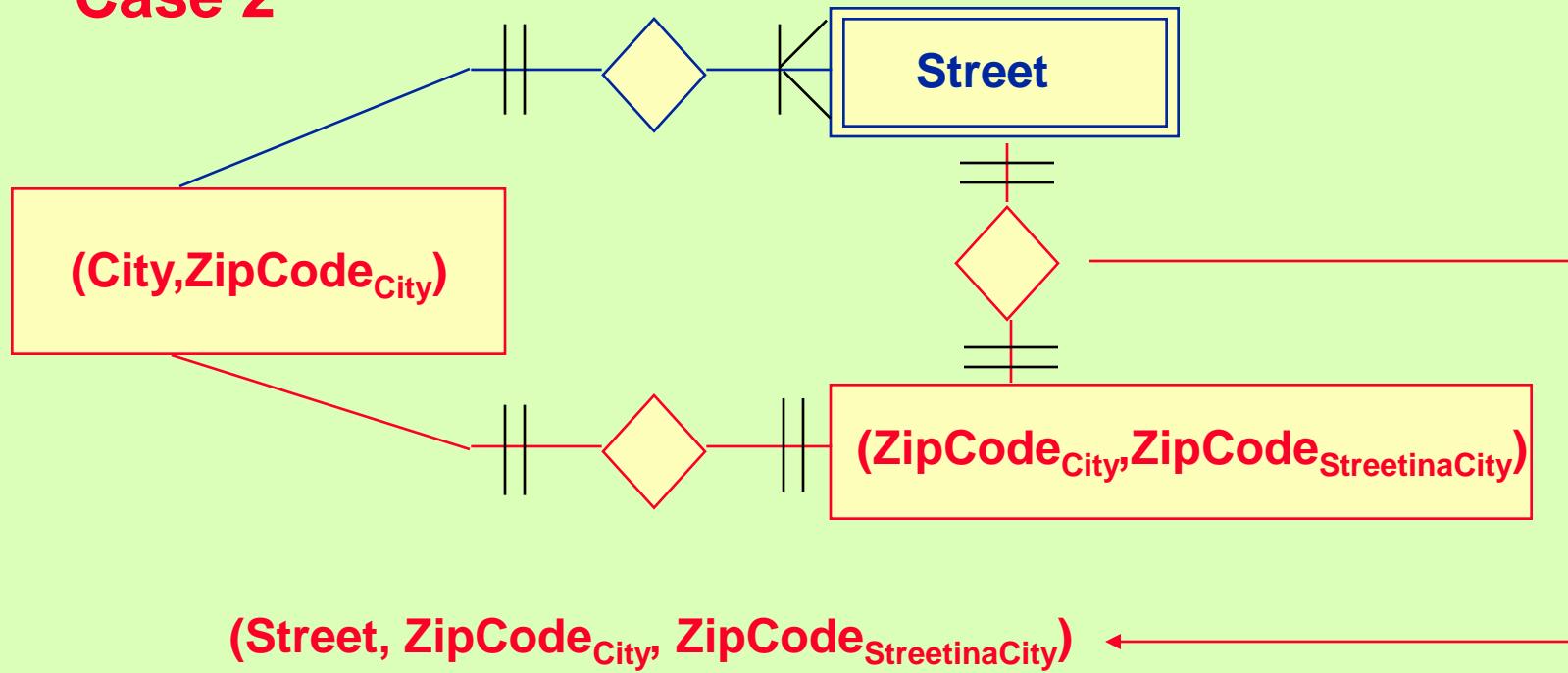
Case 1



Chuẩn hóa CSDL

- 3NF versus BCNF

Case 2



Chuẩn hóa CSDL

- 3NF versus BCNF

Case 1

Address (City, Street, ZipCode)



Address (City, Street, ZipCode_{City}, ZipCode_{StreetinACity})



A1 (City, ZipCode_{City})

A2 (City, Street, ZipCode_{StreetinACity})

City \rightarrow ZipCode_{City}

ZipCode_{City} \rightarrow City

City, Street \rightarrow ZipCode_{StreetinACity}

BCNF

Chuẩn hóa CSDL

- 3NF versus BCNF

Case 2

Address (City, Street, ZipCode)



Address (City, Street, ZipCode_{City}, ZipCode_{StreetinACity})



A1 (City, ZipCode_{City})

A2 (Street, ZipCode_{City}, ZipCode_{StreetinACity})

City \rightarrow ZipCode_{City}

ZipCode_{City} \rightarrow City

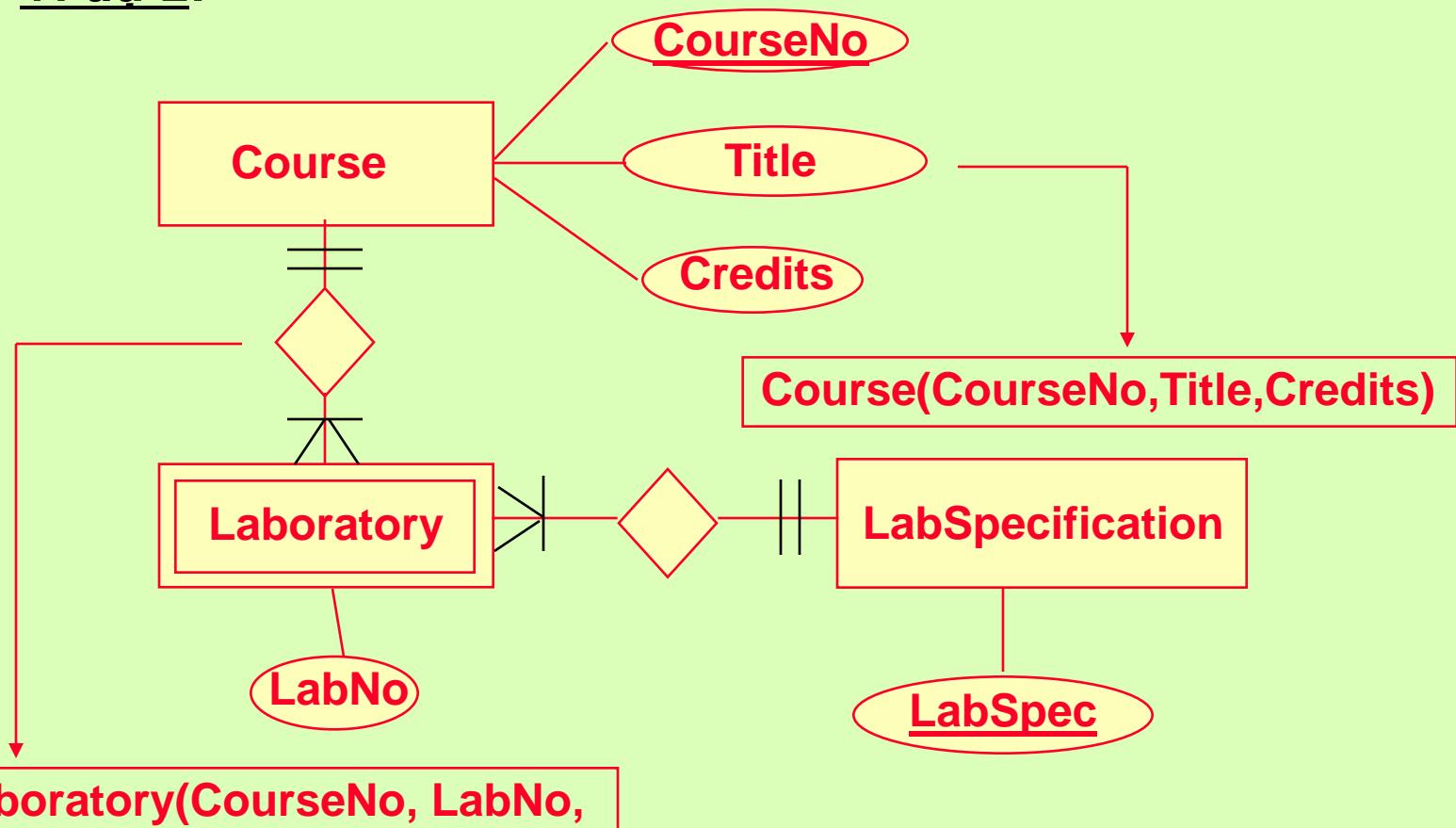
ZipCode_{City}, ZipCode_{StreetinACity} \rightarrow Street

BCNF

Chuẩn hóa CSDL

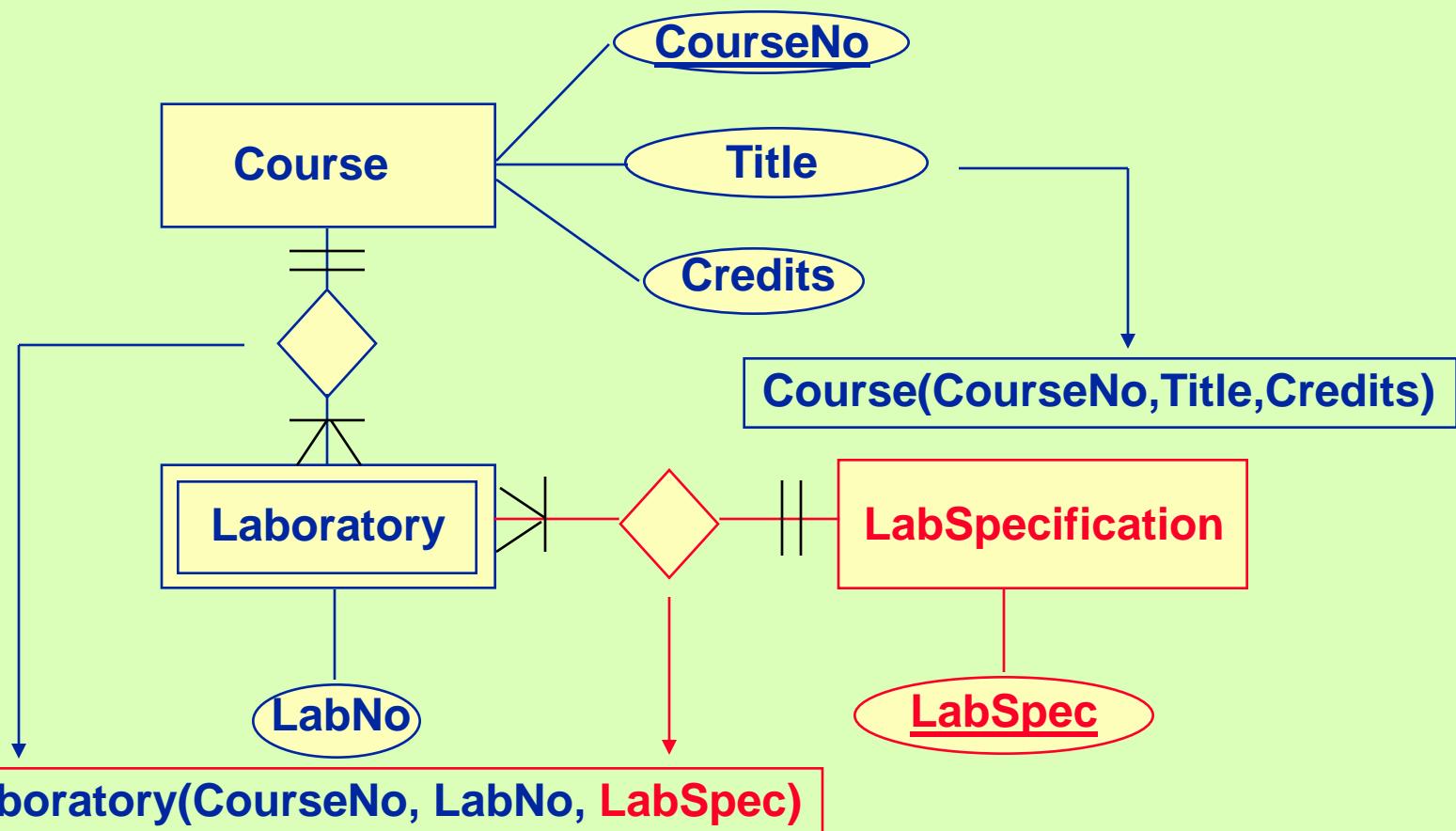
- 3NF versus BCNF

Ví dụ 2:



Chuẩn hóa CSDL

- 3NF versus BCNF



Chuẩn hóa CSDL

- 3NF versus BCNF

Laboratory (CourseNo, LabNo, LabSpec)

$F = \{LabSpec \rightarrow CourseNo,$
 $CourseNo, LabNo \rightarrow LabSpec \}$

LabSpec $\rightarrow\!/\!\rightarrow$ LabNo vì một phòng thí nghiệm chuyên biệt có thể được sử dụng cho nhiều thí nghiệm

Chuẩn hóa CSDL

- 3NF versus BCNF

Laboratory (CourseNo, LabNo, LabSpec)

3NF

$F = \{LabSpec \rightarrow CourseNo,$
 $CourseNo, LabNo \rightarrow LabSpec \}$

K1 : Course, LabNo :Course, LabNo \rightarrow LabSpec

K2 : LabSpec, LabNo :LabSpec \rightarrow Course

LabSpec,LabNo \rightarrow Course

not BCNF

Chuẩn hóa CSDL

- 3NF versus BCNF

Không thể phân rã Laboratory để đạt BCNF mà vẫn bảo toàn phụ thuộc hàm

Laboratory (CourseNo, LabNo, LabSpec)

$F = \{ \text{LabSpec} \rightarrow \text{CourseNo},$
 $\text{CourseNo}, \text{LabNo} \rightarrow \text{LabSpec} \}$

Phân rã thành:

L1 ($\text{CourseNo}, \text{LabSpec}$)

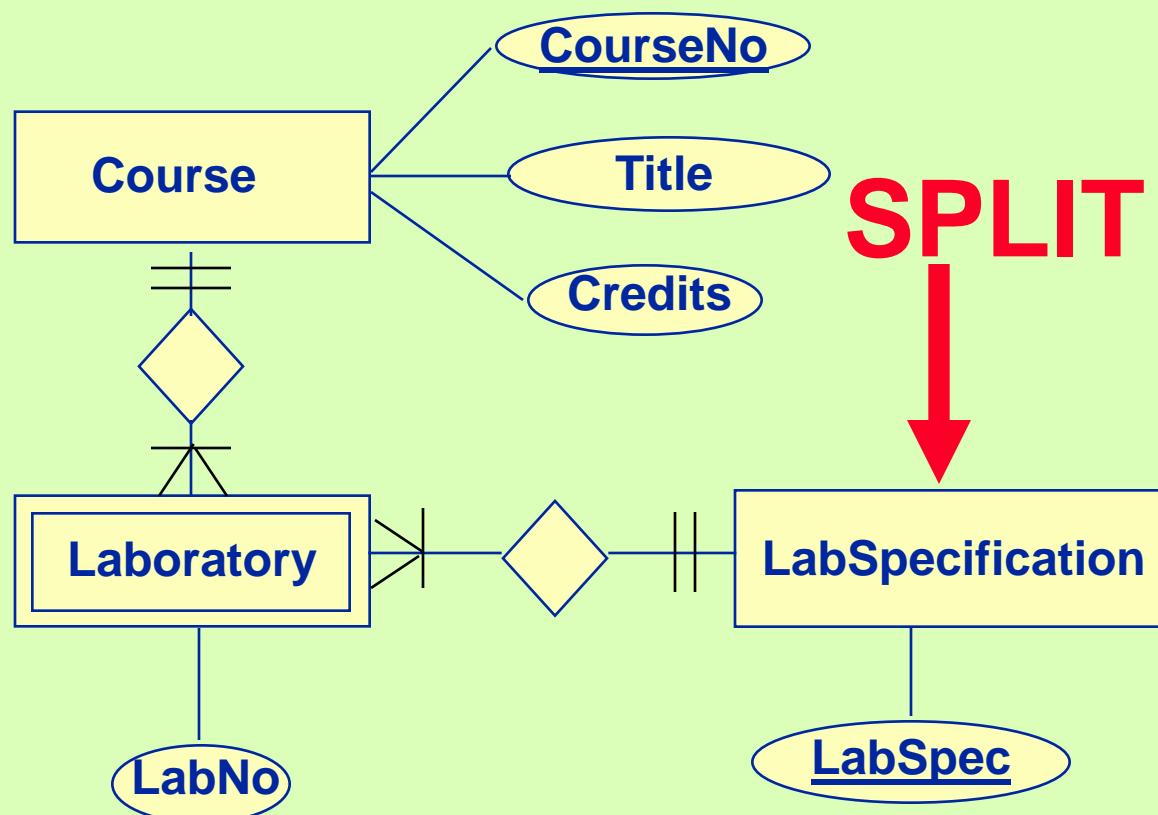
L2 ($\text{LabNo}, \text{LabSpec}$)

Làm mất phụ thuộc hàm:

Một môn học và một phòng thí nghiệm có một đặc tính riêng

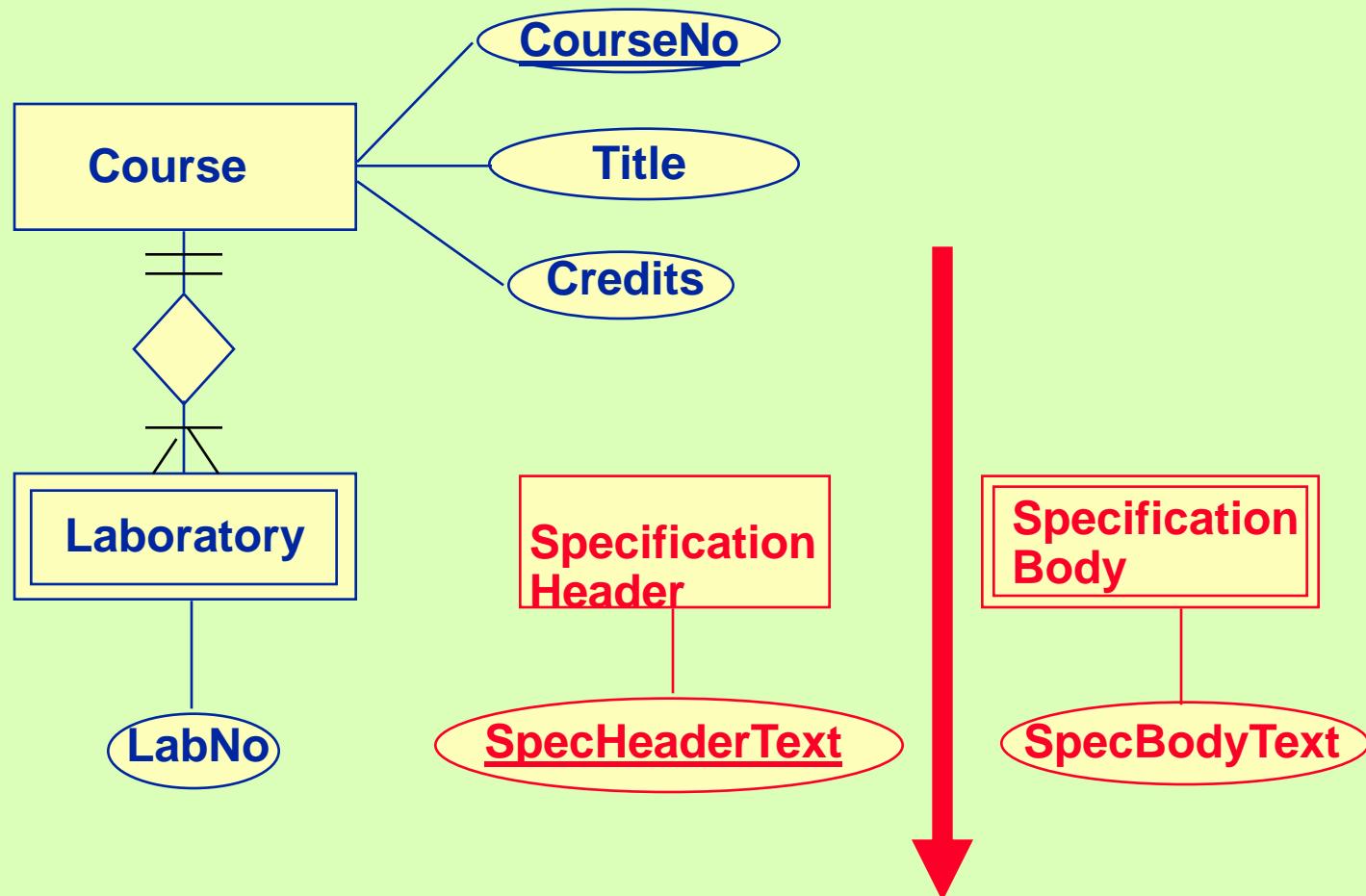
Chuẩn hóa CSDL

- 3NF versus BCNF



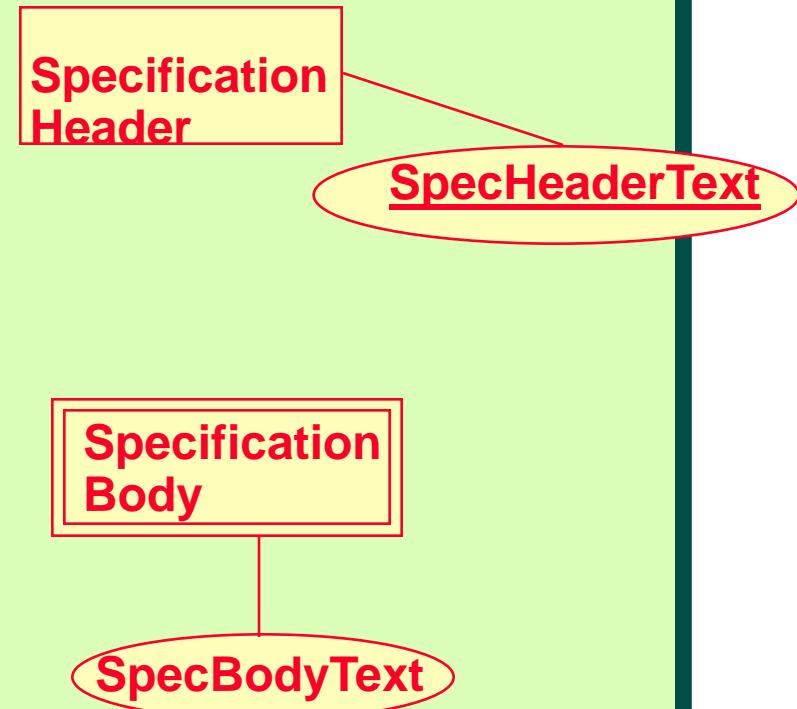
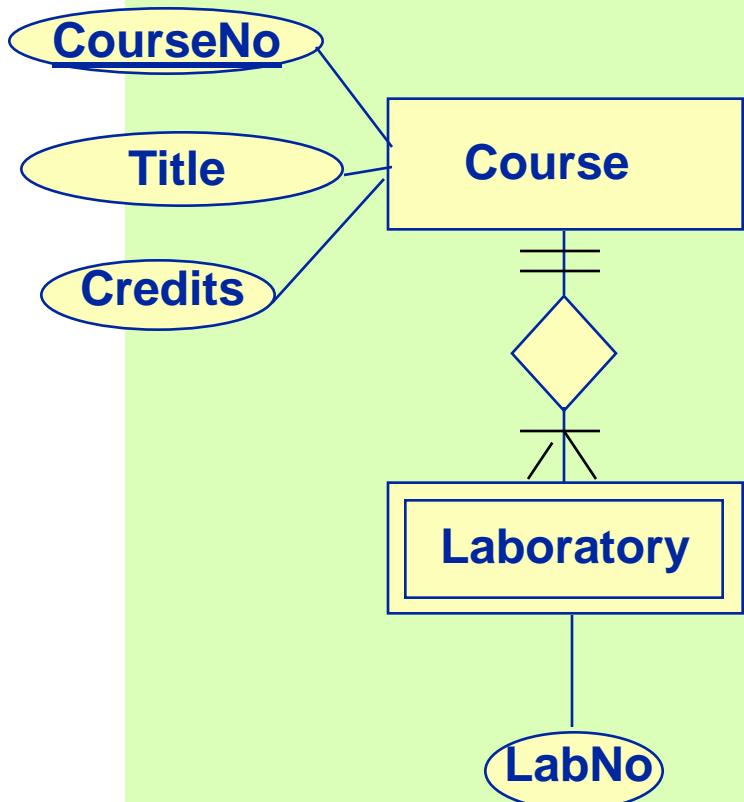
Chuẩn hóa CSDL

- 3NF versus BCNF



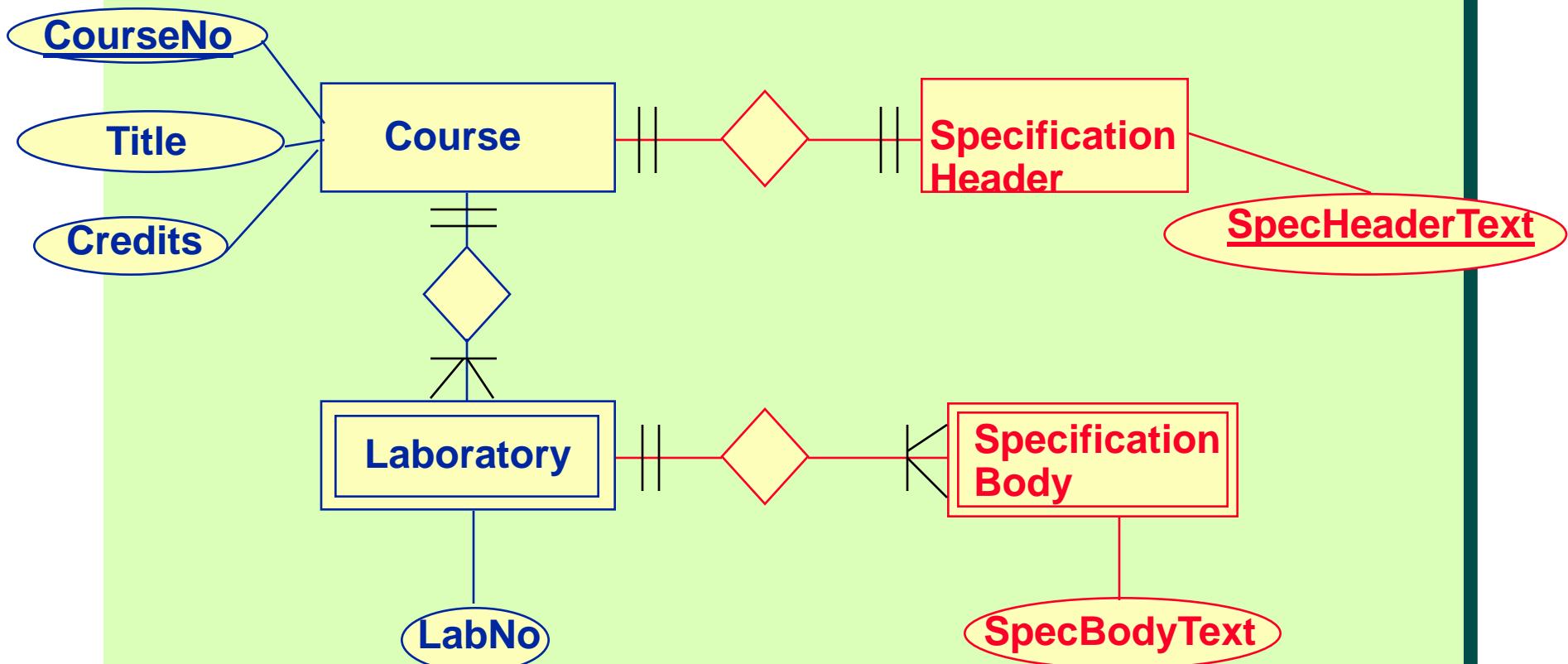
Chuẩn hóa CSDL

- 3NF versus BCNF



Chuẩn hóa CSDL

- 3NF versus BCNF



Chuẩn hóa CSDL

- 3NF versus BCNF

Case 1

Laboratory (CourseNo, LabNo, LabSpec)



Laboratory
(CourseNo, LabNo, SpecHeaderText, SpecBodyText)



L1 (CourseNo, SpecHeaderText)

L2 (CourseNo, LabNo, SpecBodyText)

CourseNo → SpecHeaderText

CourseNo, LabNo → SpecBodyText

Chuẩn hóa CSDL

- 3NF versus BCNF

Case 2

Laboratory (CourseNo, LabNo, LabSpec)



Laboratory
(CourseNo, LabNo, SpecHeaderText, SpecBodyText)



L1 (CourseNo, SpecHeaderText)

L2 (LabNo, SpecHeaderText, SpecBodyText)

CourseNo → SpecHeaderText

LabNo, SpecHeaderText → SpecBodyText

Khóa của lược đồ quan hệ

- **Định nghĩa khóa dựa trên tập phụ thuộc hàm**
Cho $R(U,F)$, U là tập thuộc tính, F là tập phụ thuộc hàm trong R

Với $X \subset U$, X là một khóa của R nếu:

- $X \rightarrow^{\infty} F^+$, và
- $\neg \exists Y \subset X, Y \neq X$, mà $Y \rightarrow^{\infty} F^+$

- R phải có ít nhất 1 khóa và có thể có nhiều khóa
- Chú ý:
Cho $R(U)$, ta luôn có $U \rightarrow U$. Nếu U thỏa điều kiện 2 ở trên thì U là khóa

Khóa của lược đồ quan hệ

- **Khóa chính (primary key):** Là một siêu khóa tối thiểu được người phân tích chọn để cài đặt
- **Khóa dự tuyển (candidate key):** Là các siêu khóa tối thiểu khác không được chọn làm khóa chính
- Khi cài đặt vào CSDL, khóa dự tuyển trở thành khóa duy nhất (unique key) và còn được gọi là khóa thứ cấp (secondary key)

Khóa của lược đồ quan hệ

- Khóa giúp cho việc nhận biết và tìm record một cách nhanh chóng trong các quan hệ
- Ràng buộc đối với khóa chính: NOT NULL và UNIQUE
- Ràng buộc đối với khóa dự tuyển: UNIQUE

Bao đóng của tập thuộc tính

- Bao đóng của tập thuộc tính X tương ứng tập phụ thuộc hàm F (ký hiệu là X_F^+) là tất cả các thuộc tính A sao cho $X \rightarrow A$ có thể được suy dẫn logic từ F

$$X \rightarrow A \in F^+ \Leftrightarrow A \subseteq X_F^+$$

- Thuật toán tìm X_F^+

Input: $R(U, F)$, $X \subseteq U$

Output: X_F^+

Phương pháp:

- $X_0 = X$;
- Lặp: $X_{i+1} = X_i \cup Z$ sao cho $\exists (Y \rightarrow Z) \in F$ và $Y \subseteq X_i$ cho đến khi $X_{i+1} = X_i$
- Ta có $X_F^+ = X_i$

Thuật toán tìm khoá

Cho $R(U,F)$, các ký hiệu trong thuật toán tìm khoá:

- $U_{right}(U_{Left})$: Các thuộc tính ở về phải của tất cả các PTH
- $\mathcal{N} = U - U_{right}$: Các thuộc tính cô lập (không xuất hiện trong bất kỳ FD nào) và các thuộc tính chỉ xuất hiện ở về trái của các FD
 $\Rightarrow \mathcal{N} \subseteq$ khoá
- $\mathcal{D} = U_{right} - U_{left}$: Các thuộc tính chỉ xuất hiện ở về phải của các FD
 $\Rightarrow \mathcal{D} \not\subseteq$ khoá
- $\mathcal{L} = U - (\mathcal{N} \cup \mathcal{D})$: là các thuộc tính có thể thuộc hoặc không thuộc khoá

Thuật toán tìm khoá (tt)

- Thuật toán tìm tất cả các khoá của một lược đồ quan hệ dựa trên tập PTH

Input: R(U, F)

Output: Các khoá của R

Phương pháp:

- $\mathcal{N} = U - U_{right}$. Nếu $\mathcal{N}_F^+ = U$, thì \mathcal{N} là khoá duy nhất của R.
Giải thuật dừng
 - Tìm $\mathcal{D} = U_{right} - U_{left}$, $\mathcal{L} = U - (\mathcal{N} \cup \mathcal{D})$:
 - Với mọi $\mathcal{L}_i \subseteq \mathcal{L}$: Thủ với từng \mathcal{L}_i , các \mathcal{L}_i có ít phần tử làm trước $X = \mathcal{N} \cup \mathcal{L}_i$, Nếu $X_F^+ = U$ thì X là một khoá của R
- Lưu ý: Nếu $X = \mathcal{N} \cup \mathcal{L}_i$ là khoá thì không thủ với các $\mathcal{L}_j \subseteq \mathcal{L}$ mà $\mathcal{L}_i \subseteq \mathcal{L}_j$

Phân rã các lược đồ quan hệ

Phân rã (decomposition) một lược đồ quan hệ $R(U)$ là thay R bằng tập các lược đồ quan hệ con R_1, R_2, \dots, R_k của R sao cho $U_1 \cup U_2 \cup \dots \cup U_k = U$, với U_i là tập thuộc tính của R_i , trong đó giao của các U_i không nhất thiết phải rỗng

- Khi muốn khôi phục lại r (thể hiện của R) ta thực hiện phép kết của các r_i (thể hiện của R_i tương ứng)
- Hỏi:
 - Dữ liệu lưu tại các r_i có phản ánh trung thực r hay không?
 - Khi cần có thể khôi phục lại được r từ các r_i hay không?

Phân rã bảo toàn nội dung

Cho $R(U,F)$. Nếu R được phân rã bởi phép phân rã ρ thành các R_i , sao cho với mọi thể hiện $r(R)$ thoả tập phụ thuộc hàm F , ta có được $r = s$ (với s có được bằng cách kết các r_i), lúc đó ρ được gọi là một phân rã bảo toàn nội dung.

- Khi phân rã bảo toàn nội dung, r ban đầu luôn được tái tạo một các trung thực bằng cách kết nối các r_i
- Kết trong trường hợp này gọi là kết không mất thông tin (*lossless join*)

Phân rã bảo toàn nội dung (tt)

Định lý:

Cho $R(U,F)$. Phân rã R thành R_1, R_2 là bảo toàn nội dung (btnd) tương ứng với F nếu và chỉ nếu

$$R_1 \cap R_2 \rightarrow R_1 - R_2 \text{ hay } R_1 \cap R_2 \rightarrow R_2 - R_1$$

Ví dụ: $R(ABC)$, $F = \{A \rightarrow B\}$

- Nếu R được phân rã thành $R_1(AB)$ và $R_2(AC)$ thì btnd
- Nếu R được phân rã thành $R_1(AB)$ và $R_2(BC)$ thì không btnd

=> Một phép phân rã có btnd hay không là còn tùy thuộc vào tập pth

Phân rã bảo toàn phụ thuộc hàm

- Chiếu của một tập phth F trên tập thuộc tính Z, ký hiệu $\P_Z(F)$, là tập các phụ thuộc hàm $X \rightarrow Y$ trong F^+ sao cho $XY \subseteq Z$
- Khi phân rã $R(U, F)$ thành các $R_i(U_i, F_i)$ thì:
 - $U_i = U$ và $F_i = \P_{U_i}(F)$
- Phép phân rã p là bảo toàn phth nếu:
 - F_i suy dẫn được tất cả các phth trong F

Phân rã bảo toàn phụ thuộc hàm (tt)

- Các pth trong F là các ràng buộc toàn vẹn trên R
- Khi phân rã btnd nhưng không bảo toàn pth thì dữ liệu có thể được phục hồi giống như ban đầu nhưng không đảm bảo ràng buộc toàn vẹn dữ liệu
- Những pth không được bảo toàn trong phép phân rã cần được kiểm tra, thông qua việc kết nối một số r_i , trong quá trình cập nhật dữ liệu
- Trường hợp lý tưởng là phân rã vừa btnd vừa bảo toàn pth
- Chỉ những phép phân rã bảo toàn nội dung mới được cài đặt

Chuẩn hoá dữ liệu và các dạng chuẩn

Định nghĩa:

- ∴ **Dạng chuẩn (normal form):** Là trạng thái của một quan hệ có được do áp dụng những quy tắc liên quan đến pth của quan hệ đó.
- ∴ **Chuẩn hoá dữ liệu (data normalization)** là quá trình phân rã lược đồ quan hệ chưa chuẩn hoá (có dạng chuẩn thấp) thành các lược đồ quan hệ nhỏ hơn nhưng ở dạng chuẩn cao hơn (có cấu trúc tốt hơn)

Các bước chuẩn hóa

Các bước chuẩn hóa dữ liệu

CSDL chưa chuẩn hóa
Tồn tại thuộc tính đa trị

Loại bỏ thuộc tính đa trị

CSDL ở dạng chuẩn 1
1NF

Loại bỏ PTH không đầy

đủ

2NF

Loại bỏ PTH bắt câu

3NF

Boyce Codd Normal Form
BCNF

*Loại bỏ những
PTH đa trị*

4NF

*Loại bỏ những dư
thừa phi chuẩn còn
lại*

5NF

Tổng kết về chuẩn hoá CSDL

- Mục tiêu của các dạng chuẩn trên quan hệ là hạn chế những dư thừa trong dữ liệu lưu trữ.
- Những dữ thừa dữ liệu là do phát sinh từ mối liên quan giữa các mục dữ liệu thể hiện qua các phụ thuộc hàm.
- Dữ thừa dữ liệu dẫn đến những dị thường khi thêm, xoá, cập nhật dữ liệu.

Tổng kết về chuẩn hoá CSDL (tt)

- Trong 2NF và 3NF vẫn còn dư thừa dữ liệu
 - 2NF: loại được sự phụ thuộc riêng phần vào khoá, vẫn tồn tại phụ thuộc bắc cầu vào khoá.
 - 3NF: loại được sự phụ thuộc bắc cầu vào khoá đối với những thuộc tính không nguyên tố.
- BCNF: loại bỏ mọi dư thừa dữ liệu mà phụ thuộc hàm có thể tạo ra (không thể suy ra giá trị tại một ô nào trong bảng bằng cách chỉ sử dụng các phụ thuộc hàm cùng với các số liệu đang có trong bảng).
- Dạng chuẩn của 1 lược đồ csdl bằng với dạng chuẩn của lược đồ quan hệ có dạng chuẩn thấp nhất

Tổng kết về chuẩn hoá CSDL (tt)

- Để đạt được dạng chuẩn cao (ít dư thừa dữ liệu), một lược đồ quan hệ thường được phân rã thành nhiều lược đồ quan hệ con.
- Một lược đồ quan hệ luôn có thể được phân rã thành các lược đồ quan hệ con đạt 3NF vừa bảo toàn nội dung vừa bảo toàn phụ thuộc hàm.
- Một lược đồ quan hệ luôn có thể được phân rã thành các lược đồ quan hệ con đạt BCNF bảo toàn nội dung nhưng có thể không bảo toàn được tập phụ thuộc hàm.