

Kỹ thuật phần mềm ứng dụng

Chương 3

Mô hình dữ liệu quan hệ (Phần 4)

Nội dung trình bày

1. *Khái niệm chuẩn hóa*
2. *Mục đích của chuẩn hóa*
3. *Các dạng chuẩn*
4. *Lược đồ khái niệm*
5. *Các phép toán trên CSDL quan hệ*
6. *Các bước đưa về dạng chuẩn 3NF /BCNF*
7. *Các quy tắc toàn vẹn dữ liệu*

1. Khái niệm chuẩn hóa

- Chuẩn hóa: Mô hình quan hệ chỉ chấp nhận duy nhất các quan hệ thoả mãn điều kiện “Mọi giá trị trong quan hệ - nghĩa là tất cả các giá trị của các thuộc tính trong tất cả các bộ - đều phải là nguyên tố (atomic) (nghĩa là không chia nhỏ được nữa)”.
- Nói một cách khác, tại mỗi vị trí cắt nhau của một hàng và một cột trong bảng tồn tại đúng một giá trị và không bao giờ là một tập các giá trị.
- Dạng chuẩn (normal form): Là trạng thái của một quan hệ có được do áp dụng những quy tắc liên quan đến phụ thuộc hàm của quan hệ đó.
- Chuẩn hoá dữ liệu: 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)

2. Mục đích của việc chuẩn hóa

- *Loại bỏ các dữ liệu dư thừa trong các bảng.*
- *Xóa bỏ các dị thường trong các thao tác cập nhật dữ liệu.*
- *Tối ưu hóa cấu trúc của các bảng.*

3 Các dạng chuẩn

- *Dạng chuẩn 1(1 Normal Form - 1NF): Một bảng đạt chuẩn 1 khi và chỉ khi tất cả các cột trong bảng chỉ chứa giá trị nguyên tố, nghĩa là không có nhóm lặp (các cột) bên trong một hàng của bảng.*

BEFORE

S#	PQ	
	P#	QTY
S1	P1	300
	P2	200
	P3	400
	P4	200
	P5	100
	P6	100
S2	P1	300
	P2	400
S3	P2	200
S4	P2	200
	P4	300
	P5	400

AFTER

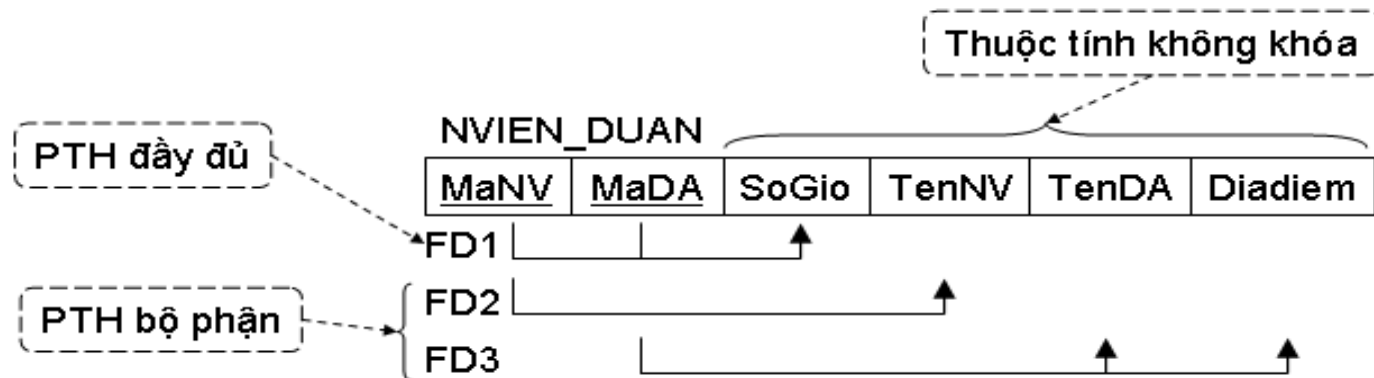
S#	P#	QTY
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400

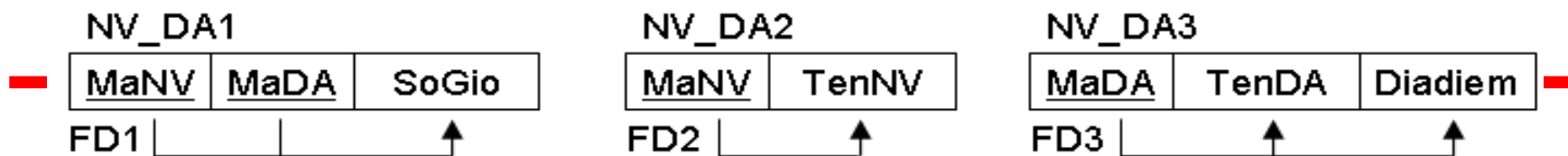
3 Các dạng chuẩn (tiếp)

- *Kết luận:*
 - *Mọi lược đồ quan hệ đều thuộc dạng chuẩn 1.*
 - *Dạng chuẩn 1 có thể dẫn đến sự trùng lặp dữ liệu. Do đó gây ra các dị thường về cập nhật dữ liệu.*

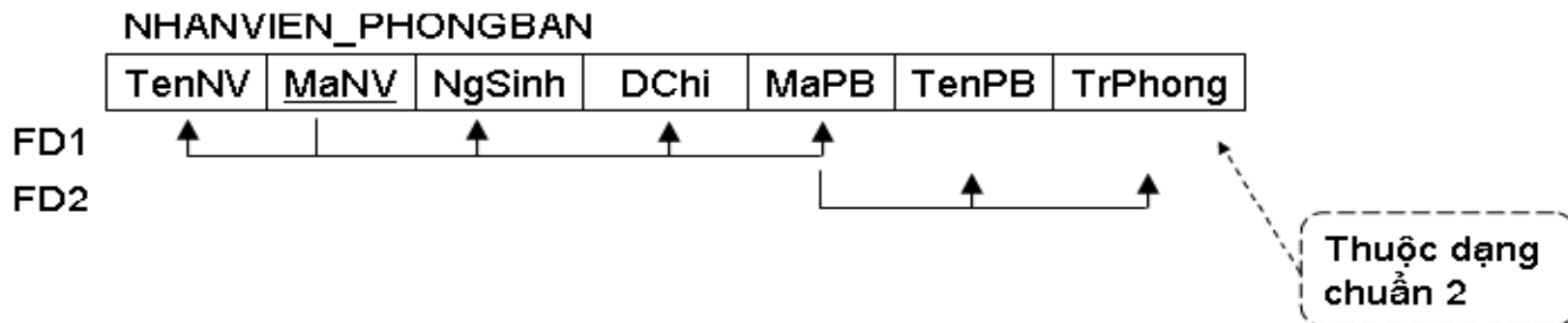
3. Các dạng chuẩn (tiếp)

- *Dạng chuẩn 2 (2 Normal Form - 2NF):* Lược đồ quan hệ R được gọi là thuộc dạng chuẩn 2 nếu nó đã ở dạng chuẩn 1 và mọi thuộc tính không khóa của R phụ thuộc hàm đầy đủ vào khóa chính của R .





3 lược đồ NV_DA1, NV_DA2, NV_DA3 thuộc dạng chuẩn 2



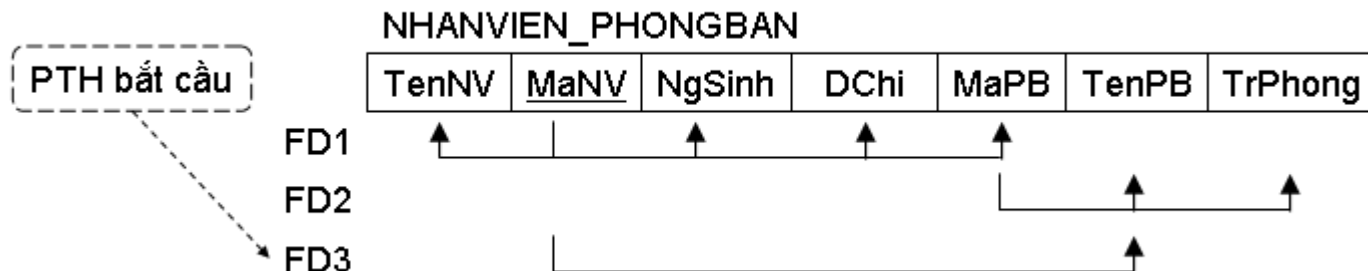
■ **Kết luận:**

- Mọi lược đồ quan hệ thuộc dạng chuẩn 2 cũng thuộc dạng chuẩn 1.
- Còn xuất hiện sự trùng lặp dữ liệu. Do đó gây ra các dị thường về cập nhật dữ liệu.

3. Các dạng chuẩn (tiếp)

- *Dạng chuẩn 3 (3 Normal Form - 3NF):* Lược đồ quan hệ R được gọi là thuộc dạng chuẩn 3 nếu:
 - R thuộc dạng chuẩn 2.
 - Mọi thuộc tính không khóa của R không phụ thuộc bắc cầu vào khóa chính của R .

Ví dụ: Lược đồ dưới đây thuộc dạng chuẩn 2 nhưng không thuộc dạng chuẩn 3, vì tồn tại phụ thuộc hàm bắc cầu:

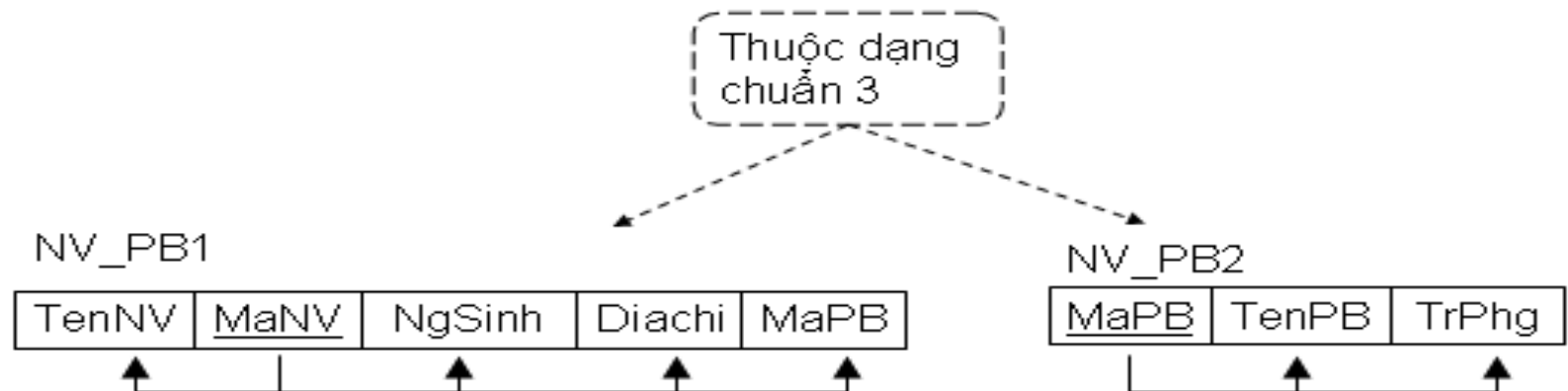


Rõ ràng: $\text{MaNV} \rightarrow \text{MaPB}$, $\text{MaPB} \rightarrow \text{TenPB} \Rightarrow \text{MaNV} \rightarrow \text{TenPB}$, Sau khi chuẩn hóa sang dạng 3 chúng ta nhận được các lược đồ sau:

3. Các dạng chuẩn (tiếp)

- **Kết luận:**

- Mọi lược đồ quan hệ thuộc dạng chuẩn 3 cũng thuộc dạng chuẩn 2.
- PTH bắt cầu là nguyên nhân dẫn đến trùng lặp dữ liệu.
- Dạng chuẩn 3 là dạng chuẩn tối thiểu trong thiết kế CSDL



3. Các dạng chuẩn (tiếp)

- *Dạng chuẩn Boyce-Codd(BCNF):* Lược đồ quan hệ R được gọi là đạt BCNF khi và chỉ khi với mọi FD $X \rightarrow A$ đúng trong R thì X là siêu khóa

Ví dụ: Cho lược đồ quan hệ $R(A,B,C,D)$, với tập phụ thuộc hàm

$F=\{A \rightarrow BCD, BC \rightarrow AD, D \rightarrow B\}$

4 Lược đồ khái niệm

S	S#	SNAME	STATUS	CITY
	S1	Smith	20	London
	S2	Jones	10	Paris
	S3	Blake	30	Paris
	S4	Clark	20	London
	S5	Adams	30	Athens

P	P#	PNAME	COLOR	WEIGHT	CITY
	P1	Nut	Red	12	London
	P2	Bolt	Green	17	Paris
	P3	Screw	Blue	17	Rome
	P4	Screw	Red	14	London
	P5	Cam	Blue	12	Paris
	P6	Cog	Red	19	London

SP	S#	P#	QTY
	S1	P1	300
	S1	P2	200
	S1	P3	400
	S1	P4	200
	S1	P5	100
	S1	P6	100
	S2	P1	300
	S2	P2	400
	S3	P2	200
	S4	P2	200

4 Lược đồ khái niệm(tiếp)

<u>DOMAIN</u>	S#	<u>CHARACTER</u> (5)
<u>DOMAIN</u>	SNAME	<u>CHARACTER</u> (20)
<u>DOMAIN</u>	STATUS	<u>NUMERIC</u> (3)
<u>DOMAIN</u>	CITY	<u>CHARACTER</u> (5)
<u>DOMAIN</u>	P#	<u>CHARACTER</u> (6)
<u>DOMAIN</u>	PNAME	<u>CHARACTER</u> (20)
<u>DOMAIN</u>	COLOR	<u>CHARACTER</u> (6)
<u>DOMAIN</u>	WEIGHT	<u>NUMERIC</u> (4)
<u>DOMAIN</u>	QTY	<u>NUMERIC</u> (5)
<u>RELATION</u>	S (S#, SNAME, STATUS, CITY) <u>KEY</u> (S#)	
<u>RELATION</u>	P (P#, PNAME, COLOR, WEIGHT, CITY) <u>KEY</u> (P#)	
<u>RELATION</u>	SP (S#, P#, QTY) <u>KEY</u> (S#, P#)	

5 Các phép toán trên CSDL quan hệ

- *Chèn (insert)*
- *Loại bỏ (delete)*
- *Thay đổi (change)*
- *Trong mô hình CSDL quan hệ được nêu trên, các phép toán này được áp dụng cho từng bộ của các quan hệ lưu trữ trong máy*

5 Các phép toán trên CSDL quan hệ(tiếp)

1. Phép chèn (INSERT)

- Phép chèn thêm một bộ t vào quan hệ r $\{A1, ..., An\}$ có dạng:

$$r = r \cup t$$

- Ta có thể biểu diễn chi tiết phép chèn như sau:

$INSERT(r; A1 = d1, A2 = d2, ..., An = dn)$

- Trong đó: - Ai với $i = 1, ..., n$ là tên các thuộc tính và di với $di \in dom(Ai)$ là các giá trị thuộc miền trị tương ứng của thuộc tính Ai .

- Ví dụ:

- Để thêm một bộ $t4 = (S4, Clark, 20, London)$ vào quan hệ S trong hình 1.2 ta làm như sau:

$INSERT(S; S\# = S4, SNAME = Clark, STATUS = 20, CITY = London)$

5 Các phép toán trên CSDL quan hệ(tiếp)

- *Nếu xem thứ tự các trường là cố định, khi đó có thể biểu diễn phép chèn dưới dạng không tường minh như sau:*
$$\text{INSERT}(r; d1, d2, \dots, dn)$$
 - *Mục đích của phép chèn là thêm một bộ vào một quan hệ nhất định*
- *Kết quả của phép toán này có thể gây ra một số sai sót với những lý do sau đây:*
 - *Bộ mới cần thêm vào không phù hợp với lược đồ quan hệ đã cho*
 - *Một số giá trị của một số thuộc tính nằm ngoài miền giá trị của thuộc tính đó*
 - *Giá trị khoá của bộ mới có thể là giá trị đã có trong quan hệ đang lưu trữ*
- *Do vậy, tùy từng hệ thống cụ thể sẽ có những cách khắc phục riêng*

5. Các phép toán trên CSDL quan hệ(tiếp)

2. Phép loại bỏ (DEL)

- Phép loại bỏ (DEL) là phép xoá một bộ ra khỏi một quan hệ cho trước
- Giống như phép chèn, phép loại bỏ có dạng:

$$r = r \setminus t$$

- Ta có thể biểu diễn chi tiết phép loại bỏ như sau:

$$DEL(r; A1 = d1, A2 = d2, \dots, An = dn)$$

- Hoặc nếu xem thứ tự các trường là cố định thì:

$$DEL(r; d1, d2, \dots, dn)$$

- Ví dụ:

- Cần loại bỏ bộ tương ứng với P2, S3 trong quan hệ SP trên hình 1.2 ta làm như sau:

$$DEL(SP; S\# = S3, P\# = P2, QTY = 200)$$

5. Các phép toán trên CSDL quan hệ(tiếp)

- *Tuy nhiên, không phải lúc nào phép loại bỏ cũng cần đầy đủ tất cả các loại thông tin của bộ cần loại bỏ*
- *Nếu có giá trị về bộ đó tại các thuộc tính khoá $K = \{B1, B2, \dots, Bj\}$ thì phép loại bỏ chỉ cần viết:*

$$DEL(r; B1 = f1, B2 = f2, \dots, Bi = fj)$$

- *Ví dụ, trong phép loại bỏ nêu trên, vì $(S\#, P\#)$ là khoá của quan hệ SP nên ta chỉ cần viết:*

$$DEL(SP; S\# = S3, P\# = P2)$$

5. Các phép toán trên CSDL quan hệ(tiếp)

3. Phép thay đổi (CHANGE hoặc UPDATE)

- Trong thực tế không phải lúc nào cũng chỉ dùng phép chèn hoặc loại bỏ đi một bộ mà nhiều khi chỉ cần sửa đổi một số giá trị nào đó tại một số thuộc tính, lúc đó cần thiết phải sử dụng phép thay đổi ("Change" hay viết tắt là CH)
- Gọi tập $\{C1, \dots, Cp\} \subseteq \{A1, \dots, An\}$ là tập các thuộc tính mà tại đó các giá trị của bộ cần thay đổi, khi đó phép thay đổi có dạng:

$$r = r \setminus t \cup t'$$

- Hay ta có thể biểu diễn chi tiết phép loại bỏ như sau:
 $CH(r; A1 = d1, A2 = d2, \dots, An = dn; C1 = e1, C2 = e2, \dots, Cp = ep)$

5. Các phép toán trên CSDL quan hệ(tiếp)

- Nếu $K = \{B1, ..., Bm\}$ là khoá của quan hệ, khi đó chỉ cần viết:

$CH(r; B1 = f1, B2 = f2, ..., Bm = fm; C1 = e1, C2 = e2, ..., Cp = ep)$

- Ví dụ: Cần thay đổi địa chỉ hãng cung cấp S1 từ London thành Amsterdam trong quan hệ S. Khi đó phép thay đổi có dạng như sau:

$CH(S; S\# = S1; CITY = Amsterdam)$

- Cũng có thể không dùng phép thay đổi mà dùng tổ hợp của phép loại bỏ và phép chèn một bộ mới
 - Do vậy, sai sót đối với phép thay đổi cũng sẽ xảy ra tương tự như với phép chèn và phép loại bỏ

6. Các bước đưa về dạng chuẩn 3NF.

- *Bước 1: Chuyển quan hệ chưa chuẩn hóa thành 1 tập quan hệ tương đương ở dạng 1NF.*
 - *2 phương pháp:*
 - *Cách 1: Loại bỏ các nhóm lặp bằng cách đưa dữ liệu thích hợp vào các cột trống của các hàng chứa dữ liệu lặp*
 - *Cách này thường được gọi là “làm phẳng” (flattening)*
 - *Bằng cách này sự dư thừa đã được đưa vào quan hệ kết quả, tuy nhiên sự dư thừa này sẽ được loại bỏ trong quá trình chuẩn hóa*
 - *Cách 2:*
 - *Chỉ định một hoặc một nhóm các thuộc tính làm khóa chính cho bảng chưa chuẩn hóa*
 - *Loại bỏ các nhóm lặp bằng cách mang nhóm các thuộc tính lặp lại cùng bản copy của khóa chính sang một quan hệ tách biệt*
 - *Nhận diện các khóa chính của các quan hệ mới*
 - *Nếu bảng chưa chuẩn hóa có chưa hơn một nhóm lặp lại hay các nhóm lặp lại lồng nhau thì cách này được lặp đi lặp lại cho đến khi không còn nhóm lặp nữa.*

6. Các bước đưa về dạng chuẩn 3NF.

- Ví dụ: Xét hoạt động của một công ty cho thuê văn phòng.
- Với giả thiết:
 - Mỗi khách hàng chỉ được thuê một văn phòng 1 lần
 - Mỗi khách hàng chỉ được thuê tối đa 1 văn phòng tại 1 thời điểm,

DreamHome						
Customer Rental Details						
Customer Name		Customer Number				
John Kay		CR76				
Property Number	Property Address	Rent Start	Rent Finish	Rent	Owner Number	Owner Name
PG4	6 Lawrence Street, Glasgow	1-Jul-93	31-Aug-95	350	CO40	Tina Murphy
PG16	5 Novar Drive, Glasgow	1-Sep-95	1-Sep-96	450	CO93	Tony Shaw

6. Các bước đưa về dạng chuẩn 3NF.

- Giả sử có 2 khách hàng, thông tin về 2 khách hàng thuê được biểu diễn dưới dạng bảng *Customer_Rental*:

Customer_Rental

Customer_No	CName	Property_No	PAddress	RentStart	RentFinish	Rent	Owner_No	OName
CR76	John Kay	PG4	6 Lawrence Street, Glasgow	1-Jul-93	31-Aug-95	350	CO40	Tina Murphy
		PG16	5 Novar Drive, Glasgow	1-Sep-95	1-Sep-96	450	CO93	Tony Shaw
CR56	Aline Stewart	PG4	6 Lawrence Street, Glasgow	1-Sep-92	10-June-93	350	CO40	Tina Murphy
		PG36	2 Manor Road, Glasgow	10-Oct-93	1-Dec-94	375	CO93	Tony Shaw
		PG16	5 Novar Drive, Glasgow	1-Jan-95	10-Aug-95	450	CO93	Tony Shaw

6. Các bước đưa về dạng chuẩn 3NF.

- Thuộc tính khóa chính được nhận diện cho bảng chưa chuẩn hóa **Customer_Rental** là **Customer_No**.
- Nhóm lặp lại: (**Property_No**, **PAddress**, **RentStart**, **RentFinish**, **Rent**, **Owner_No**, **OName**).
- Để chuyển thành 1NF, phải đảm bảo chỉ có một giá trị tại mỗi ô của bảng.
- 2 Cách chuyển thành 1NF:
 - Cách 1: Loại bỏ nhóm lặp bằng cách đưa dữ liệu tương ứng của khách hàng vào từng hàng của bảng.

Customer_Rental (1NF)

Customer_No	Property_No	CName	PAddress	RentStart	RentFinish	Rent	Owner_No	OName
CR76	PG4	John Kay	6 Lawrence Street, Glasgow	1-Jul-93	31-Aug-95	350	CO40	Tina Murphy
CR76	PG16	John Kay	5 Novar Drive, Glasgow	1-Sep-95	1-Sep-96	450	CO93	Tony Shaw
CR56	PG4	Aline Stewart	6 Lawrence Street, Glasgow	1-Sep-92	10-June-93	350	CO40	Tina Murphy
CR56	PG36	Aline Stewart	2 Manor Road, Glasgow	10-Oct-93	1-Dec-94	375	CO93	Tony Shaw
CR56	PG16	Aline Stewart	5 Novar Drive, Glasgow	1-Jan-95	10-Aug-95	450	CO93	Tony Shaw

6. Các bước đưa về dạng chuẩn 3NF

- *Cách 2: Loại bỏ nhóm lặp lại bằng cách mang dữ liệu lặp lại cùng bản copy của thuộc tính khoá gốc (Customer_No) sang một quan hệ tách biệt được gọi là Customer như trên hình 3.20*
- *Sau đó nhận diện khoá chính cho mỗi quan hệ*

6. Các bước đưa về dạng chuẩn 3NF

Customer	Customer_No	CName
	CR76	John Kay
	CR56	Aline Stewart

Prop_Rental_Owner

Customer_No	Property_No	PAddress	RentStart	RentFinish	Rent	Owner_No	OName
CR76	PG4	6 Lawrence Street, Glasgow	1-Jul-93	31-Aug-95	350	CO40	Tina Murphy
CR76	PG16	5 Novar Drive, Glasgow	1-Sep-95	1-Sep-96	450	CO93	Tony Shaw
CR56	PG4	6 Lawrence Street, Glasgow	1-Sep-92	10-June-93	350	CO40	Tina Murphy
CR56	PG36	2 Manor Road, Glasgow	10-Oct-93	1-Dec-94	375	CO93	Tony Shaw
CR56	PG16	5 Novar Drive, Glasgow	1-Jan-95	10-Aug-95	450	CO93	Tony Shaw

6. Các bước đưa về dạng chuẩn 3NF

- *Bước 2. Tạo các ánh xạ của quan hệ gốc 1NF để hạn chế mọi phụ thuộc hàm không đầy đủ vào khoá chính để tạo ra một tập các quan hệ 2NF theo các bước sau:*
 - *Từ quan hệ ban đầu, đối với mỗi thuộc tính của khoá chính, ta tạo một quan hệ tách biệt*
 - *Tiếp tục tạo các quan hệ tách biệt mới với tất cả các kết hợp của các thuộc tính của khoá chính cho đến khi tạo được quan hệ với toàn bộ khoá chính gốc*
 - *Chỉ định thuộc tính hoặc kết hợp các thuộc tính này làm khoá chính cho các quan hệ mới*
 - *Trong quan hệ gốc, tìm sự phụ thuộc hàm đầy đủ của từng thuộc tính trong các thuộc tính còn lại vào các khoá chính mới tìm ra và đặt nó vào quan hệ tương ứng với khoá chính mới này*
- *Khi đã xét hết các thuộc tính còn lại trong quan hệ gốc thì bỏ đi các quan hệ mà chỉ có các thuộc tính khoá mà không có các thuộc tính không khoá khác*

6. Các bước đưa về dạng chuẩn 3NF

– Ví dụ, xét quan hệ 1NF tên là *ORDER* như sau:

ORDER(Order_Num, *Order_Date*, Product_Num, *Product_Desc*, *Num_Ordered*)

- Khoá chính của quan hệ này là kết hợp của hai thuộc tính (*Order_Num*, *Product_Num*)
- Thuộc tính *Num_Ordered* phụ thuộc vào toàn bộ khoá chính
- Thuộc tính *Order_Date* chỉ phụ thuộc vào *Order_Num*
- Thuộc tính *Product_Desc* chỉ phụ thuộc vào *Product_Num*

⇒ Như vậy, quan hệ *ORDER* không phải ở dạng 2NF

6. Các bước đưa về dạng chuẩn 3NF

- *Theo thủ tục chuyển thành 2NF, tiến hành như sau:*
 - *Từ quan hệ ban đầu, với mỗi thuộc tính thuộc khoá chính tạo được các quan hệ tách biệt:*
 - *(Order_Num,*
 - *(Product_Num,*
 - *Kết hợp tiếp theo của các thuộc tính trong khoá chính là hai thuộc tính Order_Num và Product_Num, vì vậy, có thêm quan hệ:*
 - *(Order_Num, Product_Num,*
 - *Đặt các thuộc tính còn lại trong quan hệ ban đầu ORDER vào các quan hệ mới tương ứng:*
 - *ORDER(Order_Num, Order_Date)*
 - *PRODUCT(Product_Num, Product_Desc)*
 - *ORDER_LINE(Order_Num, Product_Num, Num_Ordered)*
 - *Như vậy, thay vì một quan hệ ORDER ban đầu, đã nhận được ba quan hệ ở dạng 2NF*

6. Các bước đưa về dạng chuẩn 3NF

- *Bước 3: Tạo các ánh xạ của các quan hệ 2NF này để hạn chế mọi phụ thuộc hàm bậc cầu của các thuộc tính không khoá vào khoá chính. Kết quả sẽ tạo nên một tập các quan hệ 3NF.*
 - *Từ quan hệ 2NF ban đầu, mang bản copy của thuộc tính quyết định cùng tất cả các thuộc tính không khoá mà phụ thuộc hàm vào thuộc tính quyết định này sang một quan hệ mới với thuộc tính quyết định sẽ được chỉ định làm khoá chính của quan hệ mới*

6. Các bước đưa về dạng chuẩn 3NF

- Ví dụ, xét quan hệ *CUSTOMER* như sau:
 - *CUSTOMER* (*Customer_Num*, *Customer_Name*, *Address*, *Sales_Rep_Num*, *Sales_Rep_Name*)
- Quan hệ này chưa ở dạng 3NF vì thuộc tính *Sales_Rep_Name* phụ thuộc hàm vào một thuộc tính là *Sales_Rep_Num*
- Theo thủ tục trên sẽ nhận được 2 quan hệ ở dạng 3NF:
 - *CUSTOMER*(*Customer_Num*, *Customer_Name*, *Address*, *Sales_Rep_Num*)
 - *SALES_REP* (*Sales_Rep_Num*, *Sales_Rep_Name*)

6. Các bước đưa về dạng chuẩn 3NF

- *Lưu ý:*
 - *Trường hợp quan hệ đã ở dạng 3NF nhưng có các khoá ứng cử phủ nhau thì chưa thể khẳng định là đã ở dạng chuẩn BCNF*
 - *Nếu tồn tại thuộc tính quyết định mà không phải là khoá ứng cử thì phải mang bản copy của thuộc tính quyết định cùng các thuộc tính phụ thuộc hàm đầy đủ vào nó sang một quan hệ mới*
- *Mục đích chính của quá trình này là giảm được sự dư thừa thông tin, và do đó tránh được các vấn đề dị thường với các phép toán lưu trữ như đã bàn đến ở trên*

7. Các quy tắc toàn vẹn dữ liệu

C.J. Date:

- *Qui tắc Toàn vẹn thực thể (Entity Integrity Rule - EI):*
 - *Không một thành phần nào của các thuộc tính dùng làm khoá chính trên một bảng dữ liệu được là null*
- *Qui tắc Toàn vẹn quy chiếu (Referential Integrity Rule - RI):*
 - *Nếu một bảng dữ liệu bao gồm một khoá ngoại lai (FK) khớp với một khoá chính (PK) trên một bảng dữ liệu T nào đó, thì mỗi trị của FK phải hoặc:*
 - *bằng trị của khoá chính trên một hàng nào đó của bảng dữ liệu T, hoặc*
 - *hoàn toàn null (Điều này không nên có vì không kết nối vào đâu)*

Cần tuân thủ các quy tắc toàn vẹn dữ liệu này trong 3 phép toán thao tác CSDL

7. Các quy tắc toàn vẹn dữ liệu(tiếp)

Phép toán Chèn (Insert):

- Nếu người sử dụng cố thử đưa vào một bộ với khoá ngoại lai không khớp với trị của khoá chính của bảng dữ liệu tương ứng thì để đảm bảo các quy tắc toàn vẹn dữ liệu, ta có thể cho áp dụng một trong 3 nguyên tắc sau đây:
 - Gạt bỏ việc thêm hàng này và yêu cầu người sử dụng đưa vào thông tin của khoá chính trên bảng dữ liệu tương ứng
 - Nguyên tắc này được gọi là nguyên tắc phụ thuộc (DEPENDENT)
 - Tự động chuyển người sử dụng qua một chương trình khác để cho phép người sử dụng thêm vào quan hệ tương ứng một hàng mới với khoá chính cần thiết
 - Nguyên tắc này được gọi là nguyên tắc tự động (AUTOMATIC)
 - Chấp nhận bộ mới với khoá ngoại lai là null, nếu đồng ý chấp nhận null đối với khoá ngoại lai
 - Nguyên tắc này được gọi là nguyên tắc làm rỗng (NULLIFY)

7. Các quy tắc toàn vẹn dữ liệu(tiếp)

Phép toán Thay đổi (Update):

- Để xác định sẽ xử lý như thế nào khi người sử dụng cố tình thay đổi khoá chính có liên hệ với một khoá ngoại lai, có thể chọn một trong 3 hướng giải quyết sau đây:
 - Tự động thay đổi giá trị của khoá ngoại lai cho khớp với giá trị mới của khoá chính
 - Nguyên tắc này được gọi là nguyên tắc thác đổ hay dây chuyền (CASCADE)
 - Chỉ cho phép thay đổi giá trị khoá chính ở hàng hiện không có giá trị của khoá chính khớp với giá trị hiện có của khoá ngoại lai tương ứng
 - Nguyên tắc này được gọi là nguyên tắc hạn chế (RESTRICT)
 - Cho khoá ngoại lai về null đối với các hàng có giá trị của khoá ngoại lai hiện khớp với giá trị của khoá chính cần thay đổi, nếu đồng ý chấp nhận null đối với khoá ngoại lai
 - Đây là nguyên tắc làm rỗng (NULLIFY)

7. Các quy tắc toàn vẹn dữ liệu(tiếp)

Phép toán Xoá (Delete):

- Nếu người sử dụng cố tình xoá một hàng chứa giá trị của khoá chính mà ít nhất có một bảng dữ liệu hiện đang sử dụng giá trị này cho khoá ngoại lai thì quy tắc toàn vẹn dữ liệu có thể tăng cường bằng một trong ba nguyên tắc sau đây (cũng tương tự như với phép thay đổi ở trên):
 - Tự động xoá tất cả các hàng trong tất cả các bảng dữ liệu có giá trị của khoá ngoại lai trùng với giá trị của khoá chính cần xoá
 - Nguyên tắc này được gọi là nguyên tắc thác đổ hay dây chuyền (CASCADE)
 - Chỉ cho phép xoá giá trị khoá chính ở hàng hiện không có giá trị của khoá chính khớp với giá trị hiện có của khoá ngoại lai tương ứng
 - Nguyên tắc này được gọi là nguyên tắc hạn chế (RESTRICT)
 - Cho khoá ngoại lai về null đối với các hàng có giá trị của khoá ngoại lai hiện khớp với giá trị của khoá chính cần xoá, nếu đồng ý chấp nhận null đối với khoá ngoại lai
 - Đây là nguyên tắc làm rỗng (NULLIFY)