* Phép toán Đại số quan hệ
* Định nghĩa: Miền (Domain)

Miền là một tập các giá trị

Tích Descarter của các miền D1 , D2 , D3 , ..., Dn được kí hiệu: D1 x D2 x ...x Dn là tập tất cả các bộ (v1 ,v2 ,....,vn ) trong đó vi ∈ Di

* Định nghĩa: Quan hệ (Relation)

Một quan hệ r với các thuộc tính U = { A1 , A2 , ... , An }, ký hiệu là r(U) là một tập các ánh xạ

Mỗi ánh xạ được gọi là một bộ của quan hệ r. Mỗi quan hệ r(U) có hình ảnh là một bảng, mỗi cột ứng với một thuộc tính, mỗi dòng là một bộ

* Định nghĩa : Lược đồ quan hệ

Một lược đồ quan hệ R là một cặp có thứ tự R=(U,F), trong đó U là tập hữu hạn các thuộc tính của quan hệ, F là tập các điều kiện giữa các thuộc tính

* Ràng buộc toàn vẹn

Một ràng buộc toàn vẹn trong một CSDL là một quy luật bất biến mà tất cả các quan hệ trong CSDL ấy phải tuân theo

+ Ràng buộc miền giá trị

Giá trị của mỗi bộ tại một thuộc tính phải thuộc một miền nào đó được xác định trước

+ Ràng buộc liên thuộc tính

Trong r(R) có thể tồn tại 2 thuộc tính A, B mà giá trị mỗi bộ tại 2 thuộc tính có liên quan nhau theo một quy tắc nào đó

+ Ràng buộc toàn vẹn trên nhiều lĩnh vực quan hệ

Ràng buộc khóa ngoại: một lược đồ R có K1 là khóa ngoại, nghĩa là K1 là một khóa của lược đồ Ri nào đó

* Các phép toán quan hệ:

1. Các phép toán logic

- Phép hợp (Union) : của r(R) và s(R) , kí hiệu r ∪ s

- Phép giao (Intersection) : của r(R) và s(R) , kí hiệu r∩s

- Phép hiệu (Set difference) : của r(R) và s(R) kí hiệu r – s

1. Phép chọn

Cho 1 quan hệ r(R), phép chọn trên quan hệ r đã cho thỏa biểu thức F ký hiệu δF (r) xác định như sau:

δF (r) = { t ∈ r | F(t) đúng}

F là một công thức gồm có:

i/ Các toán hạng, hằng, hoặc số hiệu các thành phần (thành phần i được ký hiệu là $i)

ii/ Các phép so sánh số học , ≤, ≥, ≠

iii/ Các toán tử logic (and), (or) , (not)

1. Phép chiếu

Giả sử r (R), với R = A1A2 ...An , tập thuộc tính X⊆R, phép chiếu của r lên X ký hiệu Πx (r ) được xác định : Πx (r ) = r’ (X) = { t [X] | t∈r }

1. Tích Descartes

Xét r(R), s(K) .Tích Descartes của r và s kí hiệu r x s được xác định:

r x s = {t/ ∃u∈r, v∈s, t=uv}

1. Phép kết nối tự nhiên

Xét hai quan hệ r(R), s(S). Kết nối tự nhiên của r và s kí hiệu r ⋈ s là một quan hệ được xác định như sau : r ⋈ s = ΠR∪S (δ r[C]=s[C] rxs), trong đó C=R∩S

1. Phép chia

Xét 2 quan hệ r(R) và s(S) , với S⊆R. đặt R’= R-S phép chia r cho s ký hiệu :

r ÷ s là quan hệ

* Ngôn ngữ SQL

SQL viết tắt của Structured Query languge là ngôn ngữ hỏi của công ty IBM

SanJoe năm 1981. Phiên bản gốc của ngôn ngữ này là ngôn ngữ SEQUEL được giới

thiệu năm 1976.

SQL là 1 ngôn ngữ phi thủ tục, chuẩn mực và điển hình. Để xây dựng các ứng dụng phức

tạp, SQL được nhúng vào các ngôn ngữ thế hệ thứ 3 như : COBOL,PL/1. Hiện nay rất

nhiều sản phẩm phần mềm thương mại có cài đặt SQL như Oracle, Ingres, DB2,… so sánh với các ngôn ngữ thế hệ thứ 4, SQL chưa có chức năng đồ họa và phương tiện hỏi giúp quyết định.

Tên gọi SQL = Structured Query Language, nghĩa là ngôn ngữ truy vấn (thông tin) có cấu trúc.

Tuy nhiên, không chỉ có Query mà SQL còn thực hiện nhiều chức năng khác như bổ sung thêm dữ liệu, sửa đổi cập nhật thông tin, huỷ thông tin, liên kết lưu trữ thông tin.

Vậy phải gọi là :

Structured Add, Modify, Delete, Join, Store, Trigger & Query Language.

Khả năng của SQL gồm :

1. Tạo dữ liệu, hỏi dữ liệu, cập nhật dữ liệu

2. Sắp xếp dữ liệu

3. Trình bày dữ liệu

Một số khả năng, tiện ích về cơ sở dữ liệu quan hệCơ sở dữ liệu 75 Nguyễn Đức Thuần

---------------------------------------------------------------------------------------------------------

· Tạo sinh báo cáo

· Chuyển đổi ngôn ngữ

· Chức năng trợ giúp

Chú ý : Các thuật ngữ trong CSDL quan hệ như quan hệ, thuộc tính, bộ ... trong phần

trình bày sau được thay thế bằng các thuật ngữ như bảng (table), cột (column), bản ghi

(record) hoặc hàng (row) nhằm trực quan.

SQL trong chương trình ứng dụng.

Ba kiểu tương tác với ứng dụng

- Modun : các thủ tục làm sẵn, thông qua các tham số, gọi và trả kết quả về.

- Nhúng trực tiếp : các câu lệnh SQL trong chương trình.

- Gọi trực tiếp : tương tác.

Nhúng là hình thức làm việc tĩnh – static : câu lệnh được biên dịch ra, không thay

đổi được khi chạy.

Hiện nay Dynamic SQL : cho phép chuẩn bị câu lệnh SQL trong khi chạy. Các thành

phần của câu lệnh SQL được ghép nối lại trong thời gian chạy, cho phép mức linh hoạt

rất cao trong khai thác CSDL.

Gọi trực tiếp quá trình xử lý tương tự như câu lệnh DOS.

IV.1 Câu lệnh tạo dữ liệu

a.Tạo bảng

Cú pháp : CREATE TABLE <tên quan hệ>(<tên thuộc tính> <kiểu dữ liệu>

[not null][, <tên thuộc tính> <kiểu dữ liệu> [not null][,...]])

Trong đó :

- Tên quan hệ : là 1 chuỗi ký tự bất kỳ không có ký hiệu trống, không trùng với

các từ khóa

- Tên thuộc tính : là chuỗi ký tự bất kỳ không có ký hiệu trống.( thứ tự của các

thuộc tính không là quan trọng).

- Kiểu dữ liệu : mô tả kiểu dữ liệu của thuộc tính, gồm có:

Số nguyên:

Integer:(-2147483648->2147483647)

Smallinteger : (-32768 -> 32767)

Số thực :

Decimal(n,p): số thập phân có độ dài tối đa là n ký số có p chữ số thập phân

(không tính dấu chấm thập phân).

Float : số thực dấu chấm động

Ký tự :

Char(n) : chuỗi ký tự có độ dài cố định n, n<=255

Varchar(n) : chuỗi ký tự có độ dài biến đổi từ 0 đến n.

Longvarchar : xâu ký tự có độ dài không cố định

(độ dài này từ 4Kbyte đến 32 Kbytes)

Ngày tháng :Cơ sở dữ liệu 76 Nguyễn Đức Thuần

---------------------------------------------------------------------------------------------------------

Date : dữ liệu ngày tháng (yy-mm-dd)

Trong mô tả các thuộc tính trên, NOT NULL được dùng với nghĩa thuộc tính

có mô tả này nhất thiết phải có dữ liệu.

Tác dụng : Tạo các quan hệ (bảng) theo cấu trúc được chỉ ra.

Ví dụ 1:

1. Create Table CONGTY(S# char(5) not null, tên char(25) not null, Status

integer, CITY char(40))

2. Create Table NHANVIEN(NA# char(4) not null, Hoten Varchar(30),

Ngsinh date, baclg decimal(5,2))

3. Xét 3 lược đồ quan hệ

NHACUNGCAP(Mã số, tên, tình trạng, địa chỉ)

MATHANG(số hiệu, tên mặt hàng, màu sắc, trọng lượng, thành phố sản

xuất mặt hàng)

MATHANGCCAP(Mã số nhà cung cấp, số hiệu mặt hàng, số lượng hàng

gởi đi, ngày gởi, đơn giá)

Các bảng tương ứng được tạo lặp :

CREATE TABLE S(S# integer not null, Sname varchar(8) not null, Status

integer not null, city varchar(30) not null)

CREATE TABLE P(P# integer not null, Pname varchar(10) not null, weight

integer not null, Colour varchar(10) not null, city varchar(30) not null)

CREATE TABLE SP(S# integer not null, P# integer not null, QTY integer,

Sdate date not null, Price Decimal(8,2)).

b. Đặt bí danh :

Cú pháp : CREATE SYNONYM <bí danh> FOR <tên quan hệ>

Tác dụng : để đặt bí danh ( để ngắn gọn) cho tên bảng sử dụng câu lệnh:

Ví dụ 2 : Create synonym H For CONGTY

c. Xóa dữ liệu :

Cú pháp : DROP TABLE <tên quan hệ>

Tác dụng : để xóa bảng sử dụng câu lệnh :

Ví dụ 3: DROP TABLE CONGTY

V.2 Thao tác trên bảng dữ liệu:

a. Bổ sung :

Cú pháp : INSERT INTO <tên quan hệ> [(<ds thuộc tính>)] [VALUES (bộ giá

trị)][<câu hỏi>]

Tác dụng : chèn 1 bảng ghi vào quan hệ có tên được chỉ ra

Ví dụ 4 : INSERT INTO CONGTY(S#, Ten, STATUS, CITY) VALUES ('

S4', 'John', Null, Null)Cơ sở dữ liệu 77 Nguyễn Đức Thuần

---------------------------------------------------------------------------------------------------------

b. Xóa dữ liệu :

Cú pháp : DELETE [FROM] <tên quan hệ>WHERE <mệnh đề logic>

Tác dụng : xóa các bản ghi thỏa mệnh đề logic được chỉ ra

Ví dụ 5 : DELETE CONGTY WHERE S# = 'S1'

c. Sửa đổi dữ liệu :

Cú pháp : UPDATE <tên quan hệ> SET <tên thuộc tính><biểu thức>[,…]

WHERE < mệnh đề logic>

Tác dụng : Thay giá trị của thuộc tính bởi giá trị của biểu thức tương ứng của

các bản ghi thỏa điều kiện được chỉ ra :

Ví dụ 6: UPDATE CONGTY SET CITY = ' London' WHERE

S# = 'S1' CITY = 'PARIS'

d. Thay đổi cấu trúc của bảng :

Cú Pháp : ALTER TABLE <tên quan hệ> ADD <tên thuộc tính> <k. dữ

liệu>

Tác dụng : Bổ sung 1 thuộc tính vào quan hệ.

Ví dụ 7: ALTER TABLE NHANVIEN ADD Diachi Varchar (45)

IV.3 Hỏi đáp dữ liệu

a. Câu lệnh Select

Cú pháp :

SELECT [\*[DISTINCT][<danh sách thuộc tính>

FROM <danh sách bảng>

[WHERE <mệnh đề điều kiện>]

[GROUP BY <danh sách thuộc tính>]

[HAVING <mệnh đề điều kiện> ]

[ORDER BY <tên cột>| <biểu thức>|<số thứ tự cột> [ASC \ DESC]]

Trong đó các mệnh đề WHERE có dạng :

WHERE [NOT] <mệnh đề logic>

WHERE [NOT] <tên thuộc tính> [NOT] LIKE <chuỗi ký tự/ mẫu>

WHERE [NOT] <bthức> [NOT] BETWEEN [bthức1> and <bthức2>

WHERE [NOT] <bthức> [NOT] IN (<dsách> | <câu hỏi>)

WHERE [NOT] EXITS (câu hỏi)

WHERE [NOT] <mđề logic> ANY | ALL (câu hỏi)

WHERE [NOT] bthức AND | OR [NOT] bthức2

Có thể kết nối nhiều kết quả truy vấn bằng cách dùng các phép toán tập hợp:

UNION: hợp, MINUS: hiệu, INTERSECT : giao, CONTAINS:

chứadạng:

Select .. From .. Where .. Group by ..order

UNION / MINUS/ INTERSECT/ CONTAINS

Select .. From .. Where .. Group by ..orderCơ sở dữ liệu 78 Nguyễn Đức Thuần

---------------------------------------------------------------------------------------------------------

Tác dụng :

Truy vấn không điều kiện :

Với SELECT \* : Chọn tất cả các cột thuộc bảng được chỉ ra .

SELECT DISTINCT : Chọn các giá trị phân biệt của thuộc tính được chỉ ra.

SELECT <ds thuộc tính> : Chọn các thuộc tính có tên được chỉ ra.

FROM <ds bảng> : Danh sách bảng chứa các thuộc tính.

Ví dụ 8: Hiển thị danh sách nhân viên

SELECT \*

FROM NHANVIEN

Hiển thị danh sách họ tên nhân viên

SELECT hoten

FROM NHANVIEN

Chú ý : Kết quả câu lệnh trên không tương đương với biểu thức quan hệ :

? hoten (NHANVIEN)

Vì câu lệnh trên hiển thị họ tên của các nhân viên có thể trùng nhau.

Câu lệnh tương đương với biểu thức quan hệ trên

SELECT DISTINCT hoten

FROM NHANVIEN

Truy vấn có điều kiện :

Sử dụng WHERE <bthức logic>

Ví dụ 9 : Hiển thị danh sách họ tên nhân viên có bậc lương = 2.4

SELECT DISTINCT hoten

FROM NHANVIEN

WHERE baclg = 2.4

Biểu thức đại số quan hệ tương đương :

?hoten (?baclg = 2.4 (NHANVIEN))

Các tham số chọn của câu lệnh và hàm mẫu :

· ORDER BY <tên cột> | <bthức> | <số thứ tự cột> [ASC \ DESC]

Sắp xếp thứ tự theo <tên cột> | <bthức> | <số thứ tự cột> theo thứ tự tăng dần (mặc

nhiên).

ASC : Xếp theo thứ tự tăng dần

DESC : Xếp theo thứ tự giảm dần

· GROUP BY <danh sách thuộc tính>

Phân nhóm theo danh sách thuộc tính.

· HAVING <mđề đkiện> : Thường được sử dụng cùng mệnh đề GROUP

BY. Mệnh đề điều kiện không tác động vào toàn bảng được chỉ ra ở mệnh đề

FROM, mà chỉ tác động lần lượt từng nhóm các bảng ghi đã chỉ ra ở mệnh đề

GROUP BY.

· Hàm mẫu : SQL có các hàm mẫu sau :

COUNT (<tên cột>) : Cho số phần tử thuộc cột được chỉ ra.

COUNT (\*) : Cho số bản ghi thỏa yêu cầu tìm kiếm mà không cần quan tâm đến

bất kỳ cột nào.

SUM (<tên cột>) : Cho tổng giá trị các cột có tên được chỉ ra (có dữ liệu là kiểu số)

AVG (<tên cột>) : Cho giá trị trung bình của cột có tên được chỉ ra.Cơ sở dữ liệu 79 Nguyễn Đức Thuần

---------------------------------------------------------------------------------------------------------

MIN (<tên cột>) : Cho giá trị bé nhất của cột có tên được chỉ ra.

MAX (<tên cột>) : Cho giá trị lớn nhất của cột có tên được chỉ ra.

SET (<tên cột>) : Tập giá trị của cột có tên được chỉ ra.

Ví dụ 10:

a. Cho biết số lần mặt hàng P2 đã được cung cấp

SELECT COUNT (\*)

FROM SP

WHERE P# = 'P2'

b. Tính giá trị trung bình trọng lượng các mặt hàng

SELECT AVG(WEIGHT)

FROM P

c. Hãy đưa ra danh sách nhà cung ứng mặt hàng P1 :

SELECT S.SNAME

FROM S

WHERE S.S# IN

SELECT SP.S#

FROM SP

WHERE SP.P# = 'P1'

d. Hãy đưa ra dánh sách số hiệu hãng cung ứng tất cả các mặt hàng

( Biểu thức đại số quan hệ : ?S#, P# (SP) ? ?P# (P)).

SELECT S#

FROM SP

GROUP BY S# CONTAINS

SELECT P#

FROM P

e. Đưa ra tất cả mặt hàng có tên bắt đầu là BOT

SELECT \*

FROM S

WHERE SNAME LIKE 'BOT %'

Chú ý : Ký hiệu % : Thay thế cho chuỗi con bất kỳ.

?, - : Thay cho 1 ký tự.

f. Đưa ra những mặt hàng bán trước ngày 00-08-25 là 15 ngày

SELECT P#

FROM SP

WHERE '00-08-25' - sdate = 15

Đưa ra tên mặt hàng có đơn giá từ 1000 đến 2000

SELECT PNAME

FROM P, SP

WHERE P.P# = SP.P# and PRICE BETWEEN 1000 and 2000Cơ sở dữ liệu 80 Nguyễn Đức Thuần

---------------------------------------------------------------------------------------------------------

h. Đưa ra mã hiệu của những nhà cung ứng không cung ứng mặt hàng

nào

SELECT S#

FROM S

WHERE S# NOT IN

SELECT S#

FROM SP

hoặc : SELECT S#

FROM S

MINUS /\* hiệu \*/

SELECT S#

FROM SP

( ?S#(SP) - ?s# (P))

i. Đưa ra danh sách số hiệu nhà cung cấp cả mặt hàng P1 và P2

SELECT S#

FROM SP

WHERE P# = 'P1'

INTERSECT /\* giao \*/

SELECT S#

FROM SP

WHERE P# = 'P2'

(?S#,(?P#='P1' (SP) ? ?S#(?P#='P2' (SP))

k. Đưa ra tên các mặt hàng màu xanh và sắp xếp theo thứ tự giảm dần

của mã số mặt hàng

SELECT PNAME, P#

FROM P

WHERE COLOUR = 'Blue''

ORDER BY P# ASC

Chú ý :

l. Đưa ra những nhà cung cấp đã cung cấp ít nhất 1 mặt hàng nào đó

SELECT \*

FROM S

WHERE EXITS

( SELECT \*

FROM SP

WHERE SP.S# = S.S# )

n. Đưa ra tên mặt hàng có mã số mặt hàng của 1 mặt hàng nào đó mà

hãng S1 đã bán

SELECT PNAME

FROM P

WHERE P# = ANY SELECT P#

FROM SPCơ sở dữ liệu 81 Nguyễn Đức Thuần

---------------------------------------------------------------------------------------------------------

WHERE S# = 'S1'

o. Đưa ra mã số những nhà cung cấp có số lượng của 1 mặt hàng nào

đó lớn hơn bằng số lượng cung cấp mỗi lần của tất cả các hãng

SELECT S#

FROM SP

WHERE QTY >= ALL

SELECT QTY

FROM SP

hoặc : SELECT S#

FROM SP

WHERE QTY = (SELECT MAX (QTY)

FROM SP)

* Phụ thuộc hàm
* Định nghĩa: phụ thuộc hàm

Cho r là một quan hệ trên lược đồ R, X và Y là các tập con của R. Nói rằng X xác định hàm Y hoặc Y phụ thuộc hàm. Ký hiệu X 🡪 Y

* Định nghĩa: bao đóng tập phụ thuộc hàm

Cho F là tập các phụ thuộc hàm xác định trên lược đồ quan hệ R, bao đóng tập phụ thuộc hàm F ký hiệu F+ là tập phụ thuộc hàm bé nhất chứa F sao cho thỏa hệ tiên đề Armstrong:

a. tính phản xạ: Y⊆X ⇒ X→Y∈ F+

b. Tính tăng trưởng: ∀ X, Y, Z ⊆ R: X →Y ⇒ XZ→ YZ ∈ F+

c. Tính bắc cầu: ∀ X, Y, Z ⊆ R: X → Y, Y → Z ⇒ X → Z

Ký hiệu F ╞═ X→Y nói rằng F suy dẫn logic X→Y hay X→Y được suy dẫn bởi hệ tiên đề Armstrong từ F

* Định nghĩa: Bao đóng tập thuộc tính

Cho tập thuộc tính R, F là tập phụ thuộc hàm xác định trên R. X⊆R Bao đóng của tập thuộc tính X theo F ký hiệu : X+ = { A∈ R ⏐ F ╞═ X→Y}

* Tính đúng và đủ của hệ tiên đề Armstrong

+ Đúng: F+ ⊆ X+ ( những gì suy dẫn bằng Armstrong được thì suy dẫn bằng quan hệ được)

+ Đủ: F+ ⊆ F\*

* Phủ tối thiểu
* Định nghĩa: tập phụ thuộc hàm cực tiểu

Một tập hợp phụ thuộc hàm F gọi là cực tiểu nếu:

1) Vế phải của mỗi phụ thuộc trong F chỉ có một thuộc tính độc nhất.

2) Không tồn tại bất kỳ một X→A nào trong F mà tập (F-{X→A}) + = F+

3) Không tồn tại một phụ thuộc hàm X→A thuộc F và 1 tập con Z của X:

F + = (F – {X→A} ∪ {Z→A})

Ý nghĩa:

- Điều kiện 2 để đảm bảo không có phụ thuộc hàm nào trên F là dư thừa.

- Điều kiện 3 để đảm bảo không có thuộc tính nào ở vế trái là dư thừa.

- Điều kiện 1 để đảm bảo không có thuộc tính nào ở vế phải là dư thừa.

* Định nghĩa: phủ cực tiểu của một tập phụ thuộc hàm

Cho F là tập phụ thuộc hàm, G là tập phụ thuộc hàm, G là cực tiểu và G ≡ F thì G được gọi là một phủ cực tiểu của F.

Mỗi phụ thuộc hàm đều có một phủ cực tiểu

* Khóa của quan hệ, lược đồ quan hệ
* Khóa của một quan hệ:

Siêu khóa: Siêu khóa của một quan hệ r(R) là một tập các thuộc tính của lược đồ R xác định duy nhất một bộ trong quan hệ r(R).

Một quan hệ có thể có nhiều siêu khóa

Khóa: Khóa (key) của một quan hệ r(R) là một siêu khóa của r(R) sao cho mọi tập con thực sự của nó không là siêu khóa

* Khóa của một lược đồ quan hệ :

Siêu khóa: Siêu khóa của một lược đồ quan hệ R là siêu khóa của bất kỳ quan hệ r(R).

Khóa: Khóa của một lược đồ quan hệ R là khóa của bất kỳ quan hệ r(R).

Khóa ngoại : là tập thuộc tính khóa của lược đồ quan hệ khác.

* Lược đồ quan hệ

Định nghĩa: là tập các thuộc tính của một quan hệ

Nếu ký hiệu A1, A2 , ... , Ak là các thuộc tính của một quan hệ r thì lược đồ quan hệ tương ứng với r là R = {A1, A2, ..., Ak}

* Phân rã bảo toàn thông tin

Phân rã một lược đồ quan hệ R = A1A2…An là việc thay lược đồ quan hệ R bằng một tập các lược đồ R1, R2,….,Rn

* Phân rã không tổn thất thông tin

Phân rã một lược đồ quan hệ R thành R1,R2…, Rn gọi là không tổn thất nếu:

∀r( R): r = ΠR1 ( r) ⋈ ΠR2 ( r) ⋈ ...ΠRn ( r)

* Định lý: phép tách bảo toàn

Nếu ρ=(R1,R2) là một phép tách của R và F là tập phụ thuộc hàm thì phép tách không mất thông tin đối với F khi và chỉ khi R1∩R2→R1\R2 hoặc R1∩R2 →R2\R1

* Phân rã bảo toàn phụ thuộc hàm
* Chuẩn hóa dữ liệu
* Ba dạng chuẩn:
* Dạng chuẩn 1 (1NF)

Định nghĩa: Một lược đồ quan hệ R được gọi là ở dạng chuẩn 1 (1NF) nếu và chỉ nếu toàn bộ các miền có mặt trong R đều chỉ chứa các giá trị nguyên tố

Một thuộc tính là nguyên tố nếu không là:

- Đa trị (là một danh sách (list) hay tập các giá trị hoặc hợp các giá trị khác)

- Phân rã được thành các thuộc tính nhỏ hơn

- Có thể được kết xuất từ các thuộc tính khác.

Chú ý: Một thuộc tính là nguyên tố hay không còn phụ thuộc yêu cầu và ngữ cảnh của bài toán

* Dạng chuẩn 2 (2NF)

Định nghĩa: Cho tập phụ thuộc hàm F , phụ thuộc hàm X → Y ∈ F+ , Y được gọi là phụ thuộc bộ phận vào X nếu ∃ X’⊂ X, X’ →Y ∈ F+ ngược lại Y phụ thuộc đầy đủ vào X

Định nghĩa: Một sơ đồ quan hệ s ={R,F} được gọi là ở dạng chuẩn 2 (2NF), nếu nó ở dạng chuẩn 1 và nếu mỗi thuộc tính không khóa của R phụ thuộc hàm đầy đủ vào khóa

* Dạng chuẩn 3 (3NF)

Định nghĩa : Cho sơ đồ quan hệ s ={R,F}, một tập thuộc tính X ⊆ R, A∈ R, A được gọi là phụ thuộc bắc cầu vào X trên s, nếu tồn tại một tập con Y⊆ R mà trên F+ : X →Y,Y → X, Y→ A và A ∉ XY