- Mô hình dữ liệu quan hệ:

Quan hệ(~Bảng) : D1,D2,…Dn làcác miền giá trị

Bộ(~Bản ghi~dòng) :

Thuộc tính(~Trường~cột) :

Bậc : bậc của 1 quan hệ bằng số cột

Lực lượng : lực lượng của 1 quan hệ bằng số bản ghi (số dòng)

- Mô hình thực thể liên kết:

# **Chương 2 :**

## **- Phân loại các ngôn ngữ truy vấn**

+, Ngôn ngữ đại số

+, Ngôn ngữ tính toán vị từ

## **- Phân loại các phép toán đại số quan hệ**

+, Phép toán quan hệ : phép chiếu, chọn kết nối, chia

+, Phép toán tập hợp: hợp, giao, trừ, tích đề-các

## **- Quan hê khả hợp :**

Định nghĩa : 2 quan hệ r và s được gọi là khả hợp nếu chúng được xác định trên cùng 1 miền giá trị

VD : r xác định trên D1 x D2 x … x Dn

s xác định trên D1’ x D2’ x … x Dn’

-> Di = Di’ và n=m

## **- Phép hợp :**

+, Định nghĩa : gồm các bộ thuộc ít nhất 1 trong 2 quan hệ đầu vào

+, 2 quan hệ đầu vào phải là khả hợp

+, Cú pháp : R = R1 R2

## **- Phép giao :**

+, Định nghĩa : gồm các bộ thuộc cả hai quan hệ đầu vào

+, Cú pháp : R1 R2

## **- Phép trừ :**

+, Định nghĩa : gồm các bộ thuộc quan hệ thứ nhất nhưng không thuộc quan hệ thứ hai

+, 2 quan hệ đầu vào phải là khả hợp

+, Cú pháp : R1 \ R2

## **- Phép tích đề các :**

+, Định nghĩa : là kết nối giữa từng bộ của quan hệ thứ nhất với mỗi bộ của quan hệ thứ hai

+, Cú pháp : R = R1 x R2

## **- Phép chiếu**

+, Định nghĩa : Lựa chọn một số thuộc tính từ một quan hệ

+, Cú pháp : A1,A2,…(R)

+, Ví dụ : đưa ra danh sách tên của tất cả các sinh viên : Name(Student)

## **- Phép chọn :**

+, Định nghĩa : Lựa chọn các bộ trong một quan hệ thoả mãn điều kiện cho trước

+, Cú pháp : <condition>(R)

+, Ví dụ : đưa ra danh sách những sinh viên sống ở Bundoora : Suburb = ‘Bundoora’ (Student)

- Ví dụ phép chọn và chiếu

Đưa ra tên của các sinh viên sống ở Bundoora : Name(Suburb = ‘Bundoora’Student)

## **- Phép kết nối 2 quan hệ :**

+, Khái niệm ghép bộ : u = (a1,…,an), v = (b1,…,bn) => (u,v) = (a1,…,an,b1,…,bn)

+,Phép kết nối 2 quan hệ thực chất là phép ghép các cặp bộ của 2 quan hệ thoả mãn 1 điều kiện nào đó trên chúng

+, Biểu thức kết nối là phép hội của các toán hạng, mỗi toán hạng là 1 phép so sánh đơn giản giữa 1 thuộc tính của quan hệ r và 1 thuộc tính của quan hệ s

+, Cú pháp R1 <<điều kiện>> R2

+, Ví dụ : đưa ra danh sách các sinh viên và mã khoá học mà sinh viên đó tham gia :

=> Student Id=SID Enrol

## **- Phép kết nối bằng – kết nối tự nhiên**

+, Định nghĩa : Nếu phép so sánh trong điều kiện kết nối là phép so sánh bằng thì kết nối gọi là kết nối bằng

+, Định nghia : Phép kết nối bằng trên các thuộc tính cùng tên của 2 quan hệ và sau khi kết nối 1 thuộc tính trong 1 cặp thuộc tính trùng tên đó sẽ bị loại khỏi quan hệ kết quả thì phép kết nối đó gọi là kết nối tự nhiên

+, Cú pháp phép kết nối tự nhiên : R1 \* R2

+, Ví dụ :

Table

Description automatically generated with medium confidence

## **- Ví dụ phép chọn, chiếu, kết nối**

Đưa ra tên của các sinh viên sống ở Bundoora và mã khoá học mà sinh viên đó đăng ký

=> Name, Course(Suburb = ‘Bundoora’ (Student Id=SID Enrol))

## **- Phép kết nối ngoài :**

Diagram

Description automatically generated

+, Ví dụ :

Table

Description automatically generated with medium confidence

## **- Phép chia :**

+, Định nghĩa : Phép chia giữa 1 quan hệ r bậc n và quan hệ s bậc m (m<n) với sơ đồ quan hệ của s là tập con của sơ đồ quan hệ của r là một tập các (n-m)-bộ sao cho khi ghép mọi bộ thuộc s với t thì ta đều có mọi bộ thuộc r

+, Cú pháp : R = R1 : R2

=> r s = {t | v s => (t,v) r}

# **Chương 3 : Ngôn ngữ SQL**

Char(n) : sử dụng khi trường dữ liệu đấy mang tính ổn định

Varchar(n) : sử dụng khi trường dữ liệu không ổn định, lúc dữ liệu ngắn, lúc dữ liệu lại

# **Chương 4 :**

## **- Phụ thuộc hàm :**

+, Định nghĩa : Cho R(U) là một sơ đồ quan hệ với U là tập thuộc tính {A1,A2,…,An}. X,Y là tập con không rỗng của U.  
Nói X xác định hàm Y, hay Y là phụ thuộc hàm vào X (viết : X -> Y) nếu với một quan hệ r xác định trên R(U) và với 2 bộ bất kỳ t1, t2 thuộc r mà t1[X] = t2[X] thì t1[Y] = t2[Y]

Phụ thuộc hàm là một trường hợp của rang buộc toàn vẹn, tổng quát hoá khái niệm khoá.

## **- Hệ tiên đề Amstrong đối với phụ thuộc hàm**

Cho :

R(U) là 1 sơ đồ quan hệ, U là tập các thuộc tính

X,Y,Z,W U

Ký hiệu XY = X Y

+, Phản xạ (reflexivity) Nếu Y X thì X -> Y

+, Tăng trưởng (augumentation) Nếu X -> Y thì XZ -> YZ

+, Bắc cầu (transitivity) Nếu X -> Y, Y -> Z thì X -> Z

+, Luật hợp (union) Nếu X -> Y, X -> Z thì X -> YZ

+, Luật tựa bắc cầu (pseudo-transitivity) Nếu X -> Y, WY -> Z thì XW -> Z

+, Luật tách (decomposition) Nếu X -> Y, Z Y thì X -> Z

## **- Bao đóng của một tập phụ thuộc hàm**

+, Định nghĩa : Cho F là một tập phụ thuộc hàm. Bao đóng của F ký hiệu là F+ là tập lớn nhất chứa các phụ thuộc hàm có thể được suy ra từ các phụ thuộc hàm trong F

+, Đặc điểm của bao đóng của một tập phụ thuộc hàm : Có thể rất lớn, chi phí rất tốn kém cho việc tìm kiếm

+, Vấn đề đặt ra: Kiểm tra xem một PTH có được suy diễn từ một tập PTH có sẵn không => sử dụng bao đóng của một tập thuộc tính đối với tập PTH.

## **- Bao đóng của một tập các thuộc tính đối với một tập các PTH**

+, Định nghĩa: Cho một sơ đồ quan hệ R(U), F là một tập PTH trên U. X là tập con của U. Bao đóng của tập thuộc tính X đối với Tập F, Ký hiệu là X+F (X+), là tập tất cả các thuộc tính được xác định hàm bởi X thông qua tập F

X+ = {A U | X -> A F+}

+, Có thể thấy, định nghĩa về bao đóng của một tập thuộc tính dựa trên bao đóng của tập PTH

=> Thuật toán xác định bao đóng của một tập thuộc tính

## **- Bổ đề**

+, X -> Y được suy diễn từ tập F dựa trên hệ tiên đề Amstrong khi và chỉ khi Y X+F

Ví dụ : Cho F = {A -> B, B -> C}, f : A -> C. Hỏi f F+ hay không?

Ta có A+F = ABCC => Kết luận: f F+ hay F f: A -> C

## **- Khoá tối thiểu**

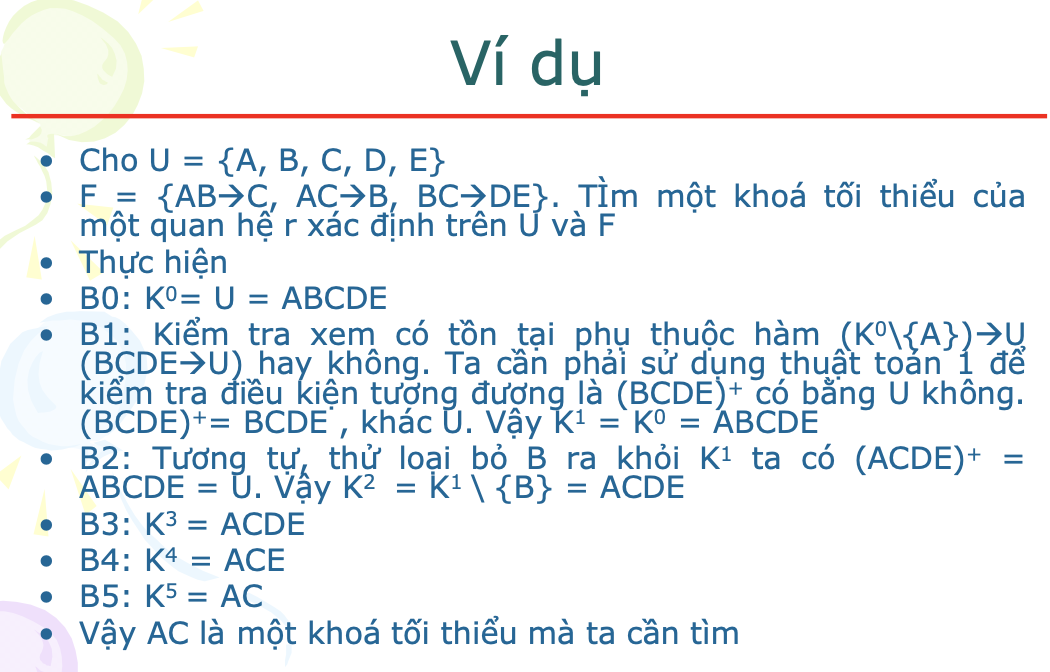
+, Định nghĩa: Cho lược đồ quan hệ R=<U,F>, U là tập thuộc tính, F làmột tập các phụ thuộc hàm xác định trên U. K được gọi là khoá tối thiểu của R nếu:

K U

K -> U F+

Với K’ K thì K’ -> U F+

+, Với những gì đã đề cập trong phần bao đóng ở trên, có thể nói, để thoả mãn là một khoá tối thiểu thì K+ = U và K là tập thuộc tính nhỏ nhất có tính chất này



Text, letter

Description automatically generated

Lưu ý: Loại bỏ dần các thuộc tính, có thể từ trái -> phải, phải -> trái hoặc từ giữa đi ra nên có thể có các kết quả khác nhau

## **- Thuật toán tìm tất cả các khoá trong lược đồ quan hệ**

U là tập tất cả các thuộc tính CSDL

F là tập phụ thuộc hàm

L(left): là các thuộc tính xuất hiện bên trái

R(right): là các thuộc tính xuất hiện ở vế phải • S(superkey): là tập các siêu khóa

K(key) : là tập các khóa

**Tập thuộc tính nguồn (TN)**: gồm các thuộc tính chỉ xuất hiện ở vế trái, không xuất hiện ở vế phải của F và các thuộc tính không xuất hiện ở cả vế trái và vế phải của F. Vậy TN = U \ R

**Tập thuộc tính đích (TĐ)**: gồm các thuộc tính chỉ xuất hiện ở R, không xuất hiện ở L.   
Vậy TĐ = R \ L

**Tập thuộc tính trung gian (TG)**: chứa các thuộc tính xuất hiện ở cả L và R

Thuật toán:

• Bước 1: Tìm tập thuộc tính nguồn TN và tập thuộc tính trung gian TG

* Bước 2: Nếu TG = ∅ thì K(Key) = TN, và kết thúc thuật toán, xuất ra K của tập cơ sở dữ liệu <U,F>
* Ngược lại, nếu TG ≠ ∅ thì qua bước 3
* Bước 3: Tìm tất cả các tập con Xi của TG
* Bước 4: Tìm Siêu khóa (Si)
* Với ∀Xi, nếu (TN U Xi)+ = U thì khi đó Si = TN U Xi
* Bước 5: Tìm Khóa (Ki) bằng cách loại bỏ các siêu khóa không
* tối thiểu
* Với mọi Si Sj thuộc S, nếu Si chứa trong Sj thì loại bỏ Sj ra khỏi tập siêu khóa. Khi đó, tập S còn lại chính là tập khóa cần tìm

A piece of paper with writing on it

Description automatically generated with medium confidence

## **- Hai tập phụ thuộc hàm tương đương**

+, Định nghĩa : Tập PTH F là phủ của tập PTH G, hay G là phủ của F, hay F và G tương đương nếu F+ = G+, kí hiệu là F G

+, Kiểm tra tính tương đương của 2 tập PTH:

Bước 1. Nếu với ∀ pth fi ∈ F, fi có dạng Xfi → Yfi, mà fi ∈ G+ thì F+⊆ G+.Kiểm tra fi ∈ G+ bằng cách kiểm tra Yfi ⊆ (Xfi)+ G

Bước2.Tương tự,nếu ∀ pth gj ∈ G,mà gj ∈ F+ thì G+ ⊆ F+

Bước3. Nếu F+ ⊆ G+ và G+ ⊆ F+ thì F ≅ G

Text, application

Description automatically generated

## **- Tập phụ thuộc hàm không dư thừa**

+, Định nghĩa : Tập phụ thuộc hàm F là không dư thừa nếu không X -> Y F sao cho F\{X -> Y} F

## **- Phủ tối thiểu của 1 tập phụ thuộc hàm**

+, Định nghĩa: FC được gọi là phủ tối thiểu của 1 tập PTH F nếu thoả mãn 3 điều kiện sau

(Đk1) Với ∀ f ∈ Fc, f có dạng X -> A, trong đó A là 1 thuộc tính.

(Đk2) Với ∀ f = X◊Y ∈ Fc, !∃ A ∈X (A là 1 thuộc tính) mà (Fc \ f) U {(X \ A) ->Y} ≅ Fc

(Đk3) !∃ X -> A ∈ Fc mà Fc \ {X -> A} ≅ Fc

## **- Tìm phủ tối thiểu Fc của một tập PTH**

B.1. Biến đổi F về dạng F1={Li -> Aj} trong đó Aj là 1 thuộc tính bất kỳ thuộc U (thoả mãn đk1)

B.2. Loại bỏ thuộc tính thừa trong vế trái của các phụ thuộc hàm.

Lần lượt giản ước từng thuộc tính trong vế trái của từng phụ thuộc hàm trong F1 thu được F1’. Nếu F1’ ≅ F1 thì loại bỏ thuộc tính đang xét  
Khi không có sự giản ước nào xảy ra nữa, thu được F2 thỏa mãn đk2  
B.3. Loại bỏ phụ thuộc hàm dư thừa

Lần lượt kiểm tra từng phụ thuộc hàm f .Nếu F2 \f ≅ F2 thì loại bỏ f. Khi không còn phụ thuộc hàm nào có thể loại bỏ thì thu đươc F3 thoả mãn đk3

B.4. Fc= F3

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

## **- Phép tách các sơ đồ quan hệ**

+, Mục đích: Thay thế một sơ đồ quan hệ R(A1, A2, …, An) bằng một tập các sơ đồ con {R1, R2,…,Rk} trong đó Ri  và R = R1 R2 … Rk

+, Yêu cầu của phép tách:

Bảo toàn thuộc tính, ràng buộc

Bảo toàn dữ liệu

### **\* Phép tách không mất mát thông tin**

+, Định nghĩa: Cho lược đồ quan hệ R(U) phép tách R thành các sơ đồ con {R1, R2,…,Rk} được gọi là phép tách không mất ámt thông tin đối với một tập phụ thuộc hàm F nếu với mọi quan hệ r xác định trên R thảo mãn F thì:

r=ΠR1(r)⋈ ΠR2(r)⋈...⋈ ΠRk (r)

+, Định lý tách đôi:

Cho lược đồ quan hệ R(U), tập pth F, phép tách R thành R1(U), R2(U) là một phép tách không mất mát thông tin nếu 1 trong 2 phụ thuộc hàm sau là thoả mãn trên F+:

U1 ∩ U2 -> U1 - U2

U1 ∩ U2 -> U2 - U1

Hệ quả: Cho lược đồ quan hệ R(U) và phụ thuộc hàm X -> Y thoả mãn trên R(U). Phép tách R thành 2 lược đồ con R1(U1), R2(U2) là một phép tách không mất mát thông tin với:

U1 = XY  
U2 = XZ  
Z = U \ XY

### **\* Phép tách bảo toàn tập phụ thuộc hàm**

+, Hình chiếu của tập phụ thuộc hàm

Cho sơ đồ quan hệ R, tập phụ thuộc hàm F, phép tách {R1, R2,…,Rk} của R trên F.

Hình chiếu Fi của F trên Ri là tập tất cả X -> Y F+ : XY Ri

+, Phép tách sơ đồ quan hệ R thành {R1,R2,…,Rk} là một phép tách bảo toàn tập phụ thuộc hàm F nếu:

(F1 F2 … Fk)+ = F+

Hay hợp của tất cả các phụ thuộc hàm trong các hình chiếu của F lên các sơ đồ con sẽ suy diễn ra các phụ thuộc hàm trong F

## **- Các dạng chuẩn đối với SĐQH**

### **\* Dạng chuẩn 1 (1NF)**

+, Định nghĩa: Một sơ đồ quan hệ R được gọi là ở dạng chuẩn 1 nếu tất cả các miền giá trị của các thuộc tính trong R đều chỉ chứa giá trị nguyên tố

Giá trị nguyên tố là giá trị mà không thể chia nhỏ ra được nữa

+, Một quan hệ r xác định trên sơ đồ quan hệ ở dạng chuẩn 1 thì quan hệ đấy là ở dạng chuẩn 1

### **\* Dạng chuẩn 2 (2NF)**

+, Định nghĩa: Một sơ đồ quan hệ R được coi là ở dạng chuẩn 2 nếu:

Sơ đồ quan hệ này ở dạng 1NF

Tất cả các thuộc tính không khoá đều phụ thuộc hàm đầy đủ vào khoá chính

(Lưu ý, A là một thuộc tính khoá nếu A thuộc một khoá tối thiểu nào đó của R. Ngược lại A là thuộc tính không khoá)

- Phụ thuộc hàm đầy đủ

+, Định nghĩa: Cho lược đồ quan hệ R(U), F là tập phụ thuộc hàm trên R. X, Y U. Y được gọi là phụ thuộc đầy đủ vào X nếu:

X -> Y thuộc F+

! X’ X : X’ -> Y F+

+, Các phụ thuộc hàm không đầy đủ gọi là phụ thuộc hàm bộ phận

### **\* Dạng chuẩn 3 (3NF)**

+, Định nghĩa: Một sơ đồ quan hệ được coi là dạng chuẩn 3 nếu:

Sơ đồ quan hệ này ở dạng 2NF

Mọi thuộc tính không khoá đều không phụ thuộc bắc cầu vào khoá chính

## **- Phụ thuộc bắc cầu:**

+, Định nghĩa : Cho sơ đồ quan hệ R(U). F là tập pth trên R(U). X,Y,Z U. Nói Z là phụ thuộc bắc cầu vào X nếu ta có X->Y, Y->Z thuộc F+. Ngược lại, ta nói Z không phụ thuộc bắc cầu vào X

## **- Tách bảo toàn tập phụ thuộc hàm về 3NF:**

Input: R(U), F(giả thiết F là phủ tối thiểu)

Output: Phép tách bảo toàn tập phụ thuộc hàm về 3NF

B1: Với các Ai thuộc U, Ai F thì loại Ai khỏi R và lập 1 quan hệ mới cho các Ai

B2: Nếu tồn tại f thuộc F, f chứa tất cả các thuộc tính của R(đã bỏ các Ai ở bước trên) thì kết quả là R

B3: Ngược lại, với mỗi X -> A F, xác định một qua hệ Ri(XA)

Nếu tồn tại X -> Ai, X -> Aj thì tạo một quan hệ chung R’(XAiAj)Text, timeline

Description automatically generated

## **- Tách không mất mát thông tin và bảo toàn tập phụ thuộc hàm về 3NF**

• Yêu cầu:

– Bảo toàn tập phụ thuộc hàm (như thuật toán trên)

– Đảm bảo là có một lược đồ con chứa khoá của lược đồ được tách

• Các bước tiến hành  
B1. Tìm một khoá tối thiểu của lược đồ quan hệ R đã cho  
B2. Tách lược đồ quan hệ R theo phép tách bảo toàn tập phụ thuộc hàm.  
B3. Nếu 1 trong các sơ đồ con có chứa khoá tối thiểu thì kết quả của B2 là kết quả cuối cùng

Ngược lại, thêm vào kết quả đó một sơ đồ quan hệ được tạo bởi khoá tối thiểu tìm được ở 1

Text

Description automatically generated