*Trương Nguyễn Thủy Tiên*

*0912463*

Topic 1: **TRÙNG LẮP DỮ LIỆU**

*Có 4 tiêu chuẩn để đánh giá một thiết kế lược đồ quan hệ:*

* *Semantic of the attribute – ngữ nghĩa của thuộc tính.*
* *Reducing the redundant values in tuples – giảm sự dư thừa giá trị trong các bộ.*
* *Reducing the null values in tuples – giảm các giá trị rỗng trong các bộ.*
* *Disallowing the possibility of generating spurious tuple – không để phát sinh các bộ sai.*

1. **Sematic of the Relation Attributes**

* Khi ta gom nhóm các thuộc tính để tạo thành một lược đồ quan hệ, ta giả định rằng các thuộc tính thuộc về một quan hệ nào đó có nghĩa thực và cách hiểu thích hợp để liên kết chúng với nhau. Nghĩa của các thuộc tính hay còn gọi là ngữ nghĩa, cho ta biết cách hiểu giá trị của thuộc tính lưu trữ trong một bộ của quan hệ - nói cách khác là cách giá trị của các thuộc tính trong một bộ có liên quan với các thuộc tính khác. Nếu lược đồ quan niệm được thiết kế cẩn thận , sau đó chuyển thành lược đồ quan hệ một cách có hệ thống, thì hầu hết các ngữ nghĩa sẽ được phân tích và bản thiết kế cuối cùng sẽ có ý nghĩa rõ rang.
* Trong thực tế, càng dễ dàng giải thích các ngữ nghĩa của một quan hệ thì lược đồ quan hệ được thiết kế sẽ càng tốt hơn.
* Việc hiểu rõ ngữ nghĩa của các thuộc tính trong một quan hệ tỉ lệ thuận với việc thiết kế lược đồ qua hệ tốt.
* EX:

EMPLOYEE (ENAME, **SSN**, BDATE, ADDRESS, DNUMBER)

DEPARTMENT (DNAME, **DNUMBER**, DMGRSSN)

DEPT\_LOCATIONS (**DNUMBER, DLOCATION**) 🡪 multivalues attribute of DEPARTMENT

PROJECT (PNAME, **PNUMBER**, PLOCATION, DNUM)

WORKS\_ON ( **SSN, PNUMBER**, HOURS) 🡪 M:N relationship between EMPLOYEE and PROJECT

* **GUIDELINE 1**

Mục tiêu của việc thiết kế lược đồ quan hệ là dễ dàng hiểu và giải thích ý nghĩa của các quan hệ. Vì thế ta không nên kết hợp các thuộc tính từ những thực thể hay quan hệ phức tạp (đa giá trị) vào trong một quan hệ đơn.

Nếu như ta xem một quan hệ tương đương hay đại diện cho một thực thể hay một quan hệ nào đó thì việc giải thích nghĩa của nó rất dễ dàng. Ngược lại nếu một quan hệ tương đương với một hỗn hợp các thuộc tính và quan hệ đa nghĩa, đa giá trị, sự nhầm lẫn và mập mờ trong ngữ nghĩa sẽ dẫn đến việc quan hệ không thể được giải thích rõ ràng.

1. **Redundant Information in Tuples and Update Anomalies**

* Một trong những mục tiêu của việc thiết kế lược đồ là giảm đến mức tối thiểu khoảng không gian lưu trữ sử dụng bởi lược đồ quan hệ (và các file có liên quan).
* Việc gom nhóm các thuộc tính trong một lược đồ quan hệ có ảnh hưởng rất lớn đến không gian lưu trữ 🡪 xem ví dụ trong file pdf.
* Update anomalies – tạm dịch là các vấn đề cập nhật.
  + Gồm 3 loại: vấn đề khi thêm mới, xóa, thay đổi.
  + Khi thêm mới: có 2 loại
    - Phải điền thong tin chính xác vào mỗi bộ sao cho đồng nhất giá trị với các bộ khác có cùng thông tin.
    - Nếu trong quan hệ mà các thuộc tính của đối tượng đó không phải là khóa chính thì việc muốn thêm 1 bộ mới là rất khó khăn, đòi hỏi phải có đối tượng của thực thể có thuộc tính làm khóa chính phù hợp đi kèm khi thêm mới.
  + Khi xóa: nếu ta xóa 1 bộ của thực thể chính mà nó chứa thông tin của một thực thể khác -> thông tin của thực thể đó sẽ mất nếu không còn thực thể chính nào liên quan đến nó.
  + Khi sửa: khi sửa thông tin của một thực thể nào đó thì phải sửa lại toàn bộ để giữ được tính nhất quán của thông tin.
* **GUIDELINE 2**
* Thiết kế một lược đồ quan hệ sao cho không có vấn đề hay sự cố bất thường nào xảy ra khi ta thêm, xóa hay sửa các giá trị của các thuộc tính trong các quan hệ. Nếu có bất cứ vấn đề nào xảy ra, chú ý đến nó và chắc chắn rằng chương trình sẽ cập nhật dữ liệu một cách đúng đắn.
* Nguyên tắc thứ 2 này nhất quán với nguyên tắc 1 hay nói cách khác, đã khẳng định lại nguyên tắc 1.
* Tuy nhiên, một điều quan trọng ta cần phải chú ý. Đó là những nguyên tắc này đôi khi có thể sẽ bị vi phạm để cải thiện sự thực thi của câu truy vấn 🡪 giải pháp: sử dụng các câu lệnh ràng buộc (trigger hay stores procedures).
* Nói chung, ta nên sử dụng các quan hệ và khung nhìn gồm các mối kết hợp để thay thế các thuộc tính thường xuyên liên quan đến các truy vấn quan trọng. Điều này sẽ giảm số lượng các phép Join trong các truy vấn và dễ dàng viết các câu lệnh. Trong nhiều trường hợp còn nâng cao quá trình thực thi các câu lệnh.

1. **Null Values in Tuples**

* Trong một số lược đồ quan hệ, ta có thể gom nhiều thuộc tính vào một quan hệ lớn. Nếu nhiều thuộc tính không dung cho tất cả các bộ trong quan hệ, ta sẽ gặp phải nhiều giá trị null trong các bộ đó. Điều này có thể gây ra sự lãng phí không gian của tầng lưu trữ và cũng có thể dẫn đến vấn đề với việc hiểu nghĩa của thuộc tính và với phép JOIN trong tầng logic.
* Một vấn đề khác của null là giải thích thế nào khi kết hợp các phép như COUNT và SUM. Ngoài ra giá trị null có thể có nhiều cách hiểu như:
  + Thuộc tính không được dung cho bộ này.
  + Giá trị của thuộc tính cho bộ này không xác định (unknown).
  + Giá trị của thuộc tính chưa xác định được, chưa được thu thập.
* **GUIDELINE 3**
* Nếu có thể ta nên tránh các thuộc tính trong một quan hệ mà giá trị của nó thường xuyên bị rỗng. Nếu giá trị null là không thể tránh khỏi, đảm bảo rằng chúng chỉ sử dụng trong các trường hợp đặc biệt (ngoại lệ) và không dùng như phần chính của bộ trong quan hệ.
* Sử dụng không gian hiệu quả và tránh liên kết là hai tiêu chí rất quan trọng để xác định có nên thêm cột có thể có giá trị null vào một quan hệ hay có một quan hệ riêng cho những cột đó (với key thích hợp).

1. **Generation of Spuripus Tuples**

* **GUIDELINE 4**

Thiết kế lược đồ quan hệ sao cho ta có thể kết bằng trên các thuộc tính là khóa chính hay khóa ngoại để bảo đảm rằng không có những bộ sai bị phát sinh. Tránh các quan hệ chứa các thuộc tính kết mà không phải là khóa chính hay khóa ngoại, bởi vì nếu thực hiện phép kết trên các thuộc tính đó có thể sinh ra các bộ sai.

1. **Summary**

* Các vấn đề bất thường có thể gây ra sự dư thừa thông tin khi thêm hay sửa quan hệ, và có thể gây ra sự mất thông tin khi xóa một bộ nào đó trong quan hệ.
* Sự lãng phí không gian lưu trữ bởi các giá trị null và sự khó khăn khi thực hiện phép kết khi có giá trị null.
* Sự phát sinh các dữ liệu sai khi thực hiện phép kết trên các quan hệ không thích hợp.