



CHƯƠNG 2

CÁC MÔ HÌNH DỮ LIỆU

(DATA MODELS)

chuc.nv@due.edu.vn

Chương 2. Các mô hình dữ liệu

Mô hình dữ liệu (data model): *Hệ thống các ký hiệu mô tả dữ liệu và một tập hợp các phép toán để thao tác trên dữ liệu đó.*

Có rất nhiều Data model như mô hình Thực thể - Mối quan hệ, mô hình dữ liệu quan hệ, mô dữ liệu mạng, mô hình phân cấp,... Mỗi mô hình có những ưu và nhược điểm riêng.

2 mô hình dữ liệu được sử dụng phổ biến trong phân tích và thiết kế CSDL hiện nay đó là mô hình Thực thể - Mối quan hệ (Entity-Relationship Model) và mô hình dữ liệu quan hệ (Relational Data Model)

2.1. Mô hình Thực thể-Mối quan hệ

Mô hình Thực thể-Mối quan hệ (Entity–Relationship Model, E-R) mô hình hóa thế giới thực bằng các khái niệm như tập thực thể và các mối quan hệ giữa chúng.

Ưu điểm của mô hình E-R là biểu diễn trực quan, dễ phát hiện những thiếu sót, xung đột dữ liệu trong quá trình khảo sát hệ thống.

Một số khái niệm của mô hình E-R

Thực thể (Entity): Là các đối tượng tồn tại và phân biệt được

Ví dụ: Mỗi sinh viên, mỗi chiếc xe máy là một thực thể.

Tập thực thể (Entity Set, Entity Type): Tập các thực thể tương tự nhau tạo thành một tập thực thể.

Ví dụ: Tập các chiếc xe máy của hãng Honda, Tập các SV của trường ĐHKT Đà Nẵng

2.1.Mô hình Thực thể-Mối quan hệ

Một số khái niệm của mô hình E-R (Cont)

Thuộc tính (Attributes):

Một tập thực thể có một tập hợp các tính chất đặc trưng và mỗi tính chất đặc trưng này được đặt bởi một tên gọi là thuộc tính.

Thông tin về mỗi thực thể trong tập thực thể được xác định bởi một bộ giá trị các thuộc tính của tập thực thể đó.

Ví dụ: Tập thực thể SINH_VIÊN có 3 thuộc tính là Họ_Tên, Lớp và Khoa. Khi đó một sinh viên cụ thể là (Nguyễn Văn Chung, 37K14, TKTH).

Thuộc tính đa trị (Multi-valued Attribute): Thuộc tính có thể nhận nhiều giá trị với mỗi thực thể

Thuộc tính hợp thành (Composite Attribute): Thuộc tính được hợp thành từ nhiều thuộc tính con

Thuộc tính dẫn xuất (Derived attribute): Thuộc tính được suy ra từ các thuộc tính khác.

2.1.Mô hình Thực thể-Mối quan hệ

Một số khái niệm của mô hình E-R (Cont)

Miền giá trị của thuộc tính (Domain Value):

Miền giá trị của một thuộc tính là tập các giá trị mà thuộc tính có thể nhận được.

Ví dụ: Họ tên có miền giá trị là các xâu ký tự độ dài tối đa 50, giới tính chỉ nhận giá trị là {Nam, Nữ}

Khóa (Key): Một thuộc tính hay một tập tối thiểu các thuộc tính mà các giá trị của nó xác định duy nhất một thực thể trong một tập thực thể gọi là khóa (key) cho tập thực thể đó.

Ví dụ: Số chứng minh nhân dân, Biển số xe, Số tài khoản ngân hàng

2.1.Mô hình Thực thể-Mối quan hệ

Một số khái niệm của mô hình E-R (Cont)

Mối quan hệ (Relationship):

Mối quan hệ R trên các tập thực thể E_1, E_2, \dots, E_n là một danh sách có thứ tự giữa các tập thực thể đó.

$$R = \langle E_1, E_2, \dots, E_n \rangle$$

R là mối quan hệ giữa n tập thực thể gọi là quan hệ n ngôi. Một thể hiện (instance) của R là một tập các bộ gồm n thành phần (e_1, e_2, \dots, e_n) , gọi là n_bộ (n_tuple), trong đó $e_i \in E_i$ ($i=1..n$).

2.1.Mô hình Thực thể-Mối quan hệ

Một số khái niệm của mô hình E-R (Cont)

Mối quan hệ ISA (Is a):

Cho hai tập thực thể E1 và E2, ta nói rằng E1 có mối quan hệ **ISA** với E2, ký hiệu **E1 ISA E2**, nếu mỗi thực thể của E1 là một thực thể của E2.

Ví dụ:

E1:MANAGERS (Title)

E2: EMPLOYEES (name,Salary, address, Gender)

E1 ISA E2

Nếu E1 ISA E2 thì E1 bao gồm (thừa hưởng) các thuộc tính của E2 đồng thời E1 có thêm các thuộc tính khác mà E2 không có.

2.1.Mô hình Thực thể-Mối quan hệ

Một số khái niệm của mô hình E-R (Cont)

Quan hệ Một - Một (One-to-One): Mỗi quan hệ R giữa 2 tập thực thể E1 và E2 được gọi là mối quan hệ một-một nếu mỗi thực thể của E1 có quan hệ với nhiều nhất một thực thể của E2 và ngược lại mỗi thực thể của E2 có quan hệ nhiều nhất với một thực thể của E1.

Ví dụ: E1: MANAGERS; E2:DEPARTMENTS

R: MANAGES

2.1.Mô hình Thực thể-Mối quan hệ

Một số khái niệm của mô hình E-R (Cont)

Quan hệ Nhiều-Một (Many-to-One): Mỗi quan hệ R giữa giữa hai tập thực thể E1 và E2 được gọi là mối quan hệ nhiều- một **từ E1 vào E2**. Nếu:

- Mỗi thực thể trong tập thực thể E1 có quan hệ với nhiều nhất 1 thực thể trong tập thực thể E2
- Mỗi thực thể trong tập thực thể E2 có thể có quan hệ với 0, 1 hoặc nhiều thực thể trong tập thực thể E1.

Ví dụ: E1:EMPLOYEES; E2: DEPARTMENTS; R: WORK_IN

Khi đó quan hệ WORK_IN là quan hệ Nhiều-Một từ EMPLOYEES Vào DEPARTMENTS.

2.1.Mô hình Thực thể-Mối quan hệ

Một số khái niệm của mô hình E-R (Cont)

Quan hệ Nhiều - Nhiều (Many-to-Many):

Mối quan hệ R giữa giữa hai tập thực thể E1 và E2 được gọi là mối quan hệ nhiều- nhiều Nếu:

- Mỗi thực thể trong tập thực thể E1 có thể có quan hệ với 0, 1 hoặc nhiều thực thể trong tập thực thể E2.
- Mỗi thực thể trong tập thực thể E2 có thể có quan hệ với 0, 1 hoặc nhiều thực thể trong tập thực thể E1.

Ví dụ: E1:Courses; E2:Students; R: Enrolled

Quan hệ **Enrolled (đăng ký học)** là quan hệ nhiều – nhiều vì rằng một sinh viên có thể đăng ký 0 hoặc nhiều môn học và ngược lại mỗi môn học có thể không có sinh viên nào đăng ký học hoặc có nhiều sinh viên đăng ký.

2.1.Mô hình Thực thể-Mối quan hệ

Sơ đồ Thực thể- Mối quan hệ (Entity – Relationship Diagram)

Sử dụng các ký hiệu trực quan (hình vẽ) để mô tả hệ thống

Ưu điểm:

- Đơn giản, dễ biểu diễn
- Trực quan, dễ phát hiện thiếu sót thông tin
- Gần với hoạt động của thế giới thực









Nhược điểm:

- Chỉ chú ý đến việc mô tả hệ thống
- Chưa chú ý đến tính cấu trúc dữ liệu
- Chưa cài đặt được trên thiết bị nhớ

2.1.Mô hình Thực thể-Mối quan hệ

Qui ước biểu diễn sơ đồ Thực thể-mối quan hệ
(First introduced by Peter Chen - 1976)



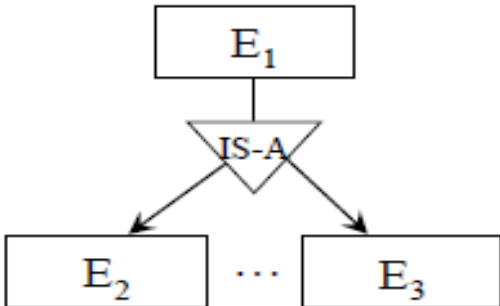
Graphical Notations of Entity-Relationship Model

<u>Symbol</u>	<u>Meaning</u>
	Entity type
	Weak entity type
	Association relationship type
	Identifying relationship
	Single-valued attribute
	Key attribute
	Partial key
	Multi-valued attribute

2.1.Mô hình Thực thể-Mối quan hệ

Qui ước biểu diễn sơ đồ Thực thể-Mối quan hệ

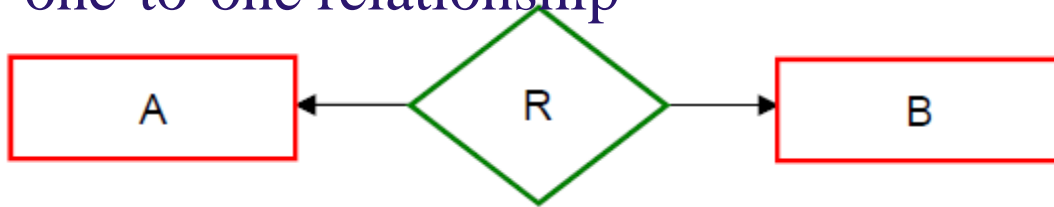
Graphical Notations of Entity-Relationship Model (Cont'd)

<u>Symbol</u>	<u>Meaning</u>
	Composite attribute
	Derived attribute
	Specialization relationship type (where E_1 is the superclass while E_2 and E_3 are the subclasses of E_1)

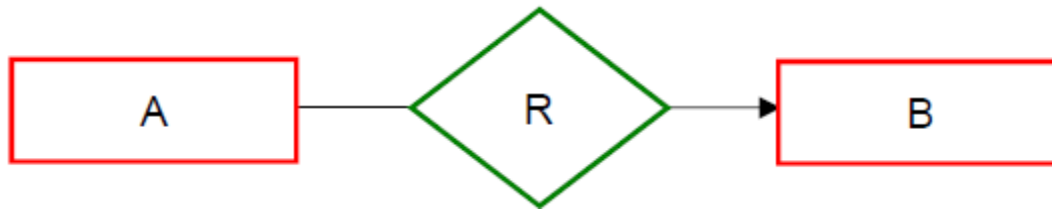
2.1.Mô hình Thực thể-Mối quan hệ

Biểu diễn mối quan hệ

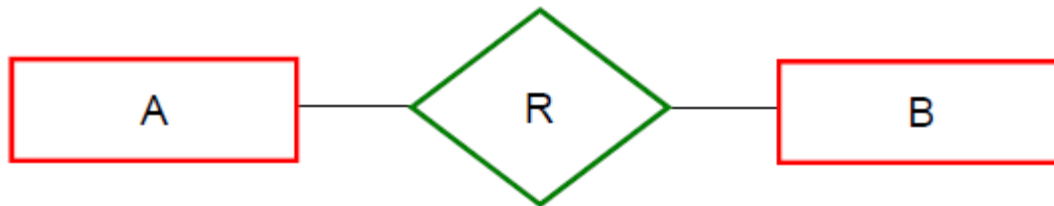
one-to-one relationship



many-to-one relationship from A to B

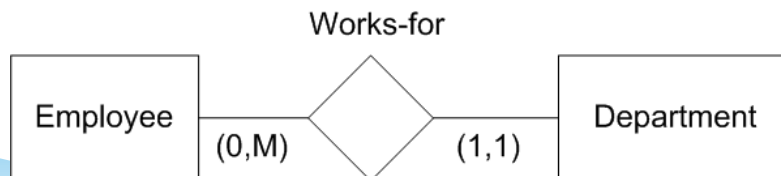
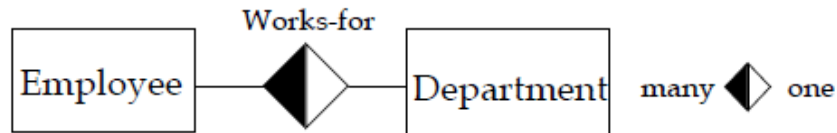
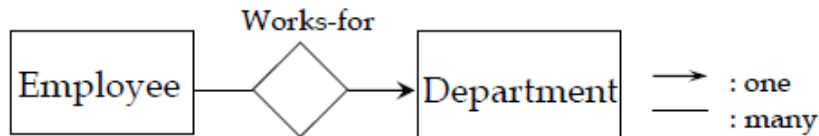
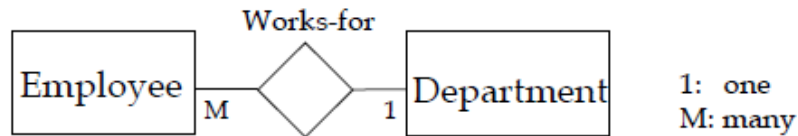


many-to-many relationship



2.1.Mô hình Thực thể-Mối quan hệ

Một số qui ước biểu diễn sơ đồ Thực thể-Mối quan hệ khác Different Graphical Notations for Cardinality



2.2. Mô hình dữ liệu quan hệ (Relational Data Model)

- Mô hình Thực thể - Mỗi quan hệ chỉ quan tâm đến việc mô tả hệ thống chứ chưa quan tâm đến cấu trúc và tổ chức dữ liệu vật lý nên không thể cài đặt trên thiết bị nhớ.
- Mô hình dữ liệu được sử dụng rộng rãi cho các hệ QTCSDL hiện nay là mô hình dữ liệu quan hệ do E.F CODE đề xuất năm 1970.
- Mô hình dữ liệu quan hệ tổ chức dữ liệu dưới dạng các bảng gồm các dòng (rows) và các cột (Columns)
- Mô hình dữ liệu quan hệ được xây dựng dựa trên lý thuyết tập hợp nên dễ hiểu và dễ biểu diễn bằng toán học

2.2. Mô hình dữ liệu quan hệ (Relational Data Model)

Định nghĩa quan hệ (relation):

Quan hệ là tập con của tích Descartes của một hoặc nhiều miền.

Quan hệ có thể có hữu hạn hoặc vô hạn số phần tử (chúng ta chỉ quan tâm đến các quan hệ có số phần tử hữu hạn)

D1: TênSV = Text(50), D2: Lớp = Text (5), D3: Giới Tính = {Nam, Nữ}

Sau đây là một quan hệ trên 3 miền D1, D2 và D3

TênSV	Lớp	Giới Tính
Nguyễn Văn Châu	42K14	Nam
Ngô Văn Dũng	43K21	Nam
Nguyễn Thị Hạnh	42K22	Nữ

2.2. Mô hình dữ liệu quan hệ (Relational Data Model)

Một số chú ý:

- Mỗi lược đồ quan hệ có rất nhiều quan hệ.
- Mỗi quan hệ còn được gọi là thể hiện (instance) hay hiện hành của lược đồ quan hệ

Ví dụ: Với lược đồ quan hệ SinhViên(MãSV, Họ tên, Lớp) có 2 quan hệ như sau:

r1			r2		
MãSV	Họ tên	Lớp	MãSV	Họ tên	Lớp
SV001	Nguyễn Văn Chung	42K14	SV007	Nguyễn Văn Ban	43K14
SV002	Trần Thị Hồng	43K21	SV008	Trần Thị Nga	42K14
SV003	Ngô Văn Chương	41K22	SV009	Lê Thị Nhung	41K22
SV004	Thái Viết Châu	42K14	SV010	Ngô Thị Hải	43K22

- Tập tất cả các lược đồ quan hệ của một cơ sở dữ liệu gọi là lược đồ CSDL quan hệ (Relational Data Schema)

2.2. Mô hình dữ liệu quan hệ (Relational Data Model)

Khóa của LĐ quan hệ (Key of Relation)

Siêu khóa (SuperKey):

Cho LĐQH $R(A_1, A_2, \dots, A_n)$
và tập thuộc tính

$$S \subseteq \{A_1, A_2, \dots, A_n\}$$

S được gọi là siêu khóa của $R(A_1, A_2, \dots, A_n)$ nếu các thuộc tính của S xác định duy nhất các bộ trong mỗi quan hệ của LĐQH R .

Hoặc có thể định nghĩa như sau:

$$\forall r \in R, \forall u, v \in r : u \neq v \Rightarrow A \in S : u(A) \neq v(A)$$

Lưu ý rằng theo tính chất của quan hệ thì mỗi bộ là duy nhất nên tập tất cả các thuộc tính của quan hệ là siêu khóa

2.2. Mô hình dữ liệu quan hệ (Relational Data Model)

Khóa (Key):

Tập K các thuộc tính của lược đồ quan hệ R được gọi là khóa của R nếu K là siêu khóa cực tiểu. Tức là mọi tập con thực sự của K không phải là siêu khóa.

Nói cách khác:

Cho một lược đồ quan hệ $R(U)$, K là một tập thuộc tính của U . K được gọi là khoá của $R(U)$ nếu thoả mãn hai điều kiện sau:

- i. Với mọi quan hệ $r \in R(U)$, không có hai bộ khác nhau nào của r trùng nhau trên tất cả các thuộc tính trong K . Điều này có nghĩa là không tồn tại hai bộ $u, v \in r$ sao cho $u \neq v$ nhưng $u(K) = v(K)$.
- ii. Không có tập con thực sự nào của K có tính chất (i) trên.

2.2. Mô hình dữ liệu quan hệ (Relational Data Model)

Khoá ngoại (Foreign Key)

Định nghĩa khóa ngoại:

Cho lược đồ R và lược đồ Q. Tập con H các thuộc tính của R gọi là khóa ngoại của R tham chiếu đến lược đồ Q, nếu Q có khóa chính là K gồm các thuộc tính của H thỏa mãn:

Với mọi r và q lần lượt là các quan hệ trên các lược đồ quan hệ R và Q. Khi đó ta có: $\forall u \in r, \exists v \in q : u(H) = v(K)$

Chú ý: Phân biệt khóa chính (Primary Key – PK) và khóa ngoại (Foreign Key- FK)

2.2. Mô hình dữ liệu quan hệ (Relational Data Model)

Lược đồ **Employees** có khoá ngoại **Dno** tham chiếu đến khoá chính **Dno** của lược đồ **Departments**

Employees

	Eno	Ename	Gender	Skill	DateOfBirth	Salary	Dno
1	E0001	Khoa	1	C#	1980-01-20	960	D001
2	E0002	Nga	0	SQL	1985-06-15	720	D002
3	E0003	Chuc	1	SQL	1988-02-14	1000	D001
4	E0004	Ngoc	0	Java	1989-03-15	900	D002
5	E0005	Chau	1	C#	1985-02-18	1200	D001
6	E0006	Thanh	1	mySQL	1982-02-18	1100	D001

Departments

	Dno	Dname	Location
1	D001	IT	Room A301
2	D002	HR	Room B202
3	D003	MAR	Room B201

2.2. Mô hình dữ liệu quan hệ (Relational Data Model)

Giá trị Null (Null value):

- Null là giá trị đặt biệt trong CSDL, dùng để chỉ giá trị của một số thuộc tính chưa có, chưa biết hoặc chưa dùng. Khi nhập dữ liệu vào các thuộc tính của một đối tượng, các thuộc tính cho phép null có thể bỏ qua.
- Cần phân biệt giá trị null với giá trị zero (với thuộc tính kiểu số) hay giá trị rỗng (với thuộc tính kiểu ký tự).

Eno	Ename	Gender	Skill	DateOfBirth	Salary	Dno
E0001	Khoa	TRUE	C#	1980-01-20	960	D001
E0002	Nga	FALSE	NULL	1985-06-15	NULL	D002
E0003	Chuc	TRUE	SQL	1988-02-14	1000	D001
E0004	Ngoc	FALSE	Java	NULL	900	D002
E0005	Chau	TRUE	C#	1985-02-18	1200	D001
E0006	Thanh	TRUE	mySQL	1982-02-18	1100	D001

2.2. Mô hình dữ liệu quan hệ (Relational Data Model)

Các qui tắc toàn vẹn dữ liệu (Integrity Rules)

Qui tắc toàn vẹn (integrity rule) là các ràng buộc trên dữ liệu đảm bảo trạng thái nhất quán của cơ sở dữ liệu.

Có các loại qui tắc toàn vẹn sau:

- Toàn vẹn thực thể (Entity integrity)
- Toàn vẹn miền giá trị (Domain value integrity)
- Toàn vẹn tham chiếu (Referential integrity)
- Thao tác bất lỗi (Triggering operations).

2.2. Mô hình dữ liệu quan hệ (Relational Data Model)

Chuyển đổi từ mô hình E-R sang CSDL quan hệ

Nguyên tắc chuyển đổi:

Chuyển đổi các tập thực thể:

Mỗi tập thực thể được chuyển thành một lược đồ quan hệ cùng tên, cùng số thuộc tính và giữ nguyên khóa.

Chuyển đổi các mối quan hệ (Xác định khóa ngoại):

Nếu R là mối quan hệ Một-Một giữa 2 tập thực thể E1 và E2 thì có thể lấy khóa của E1 đưa vào E2 làm khóa ngoại hoặc ngược lại.

Nếu R là mối quan hệ Nhiều-một từ E1 vào E2 thì lấy khóa của E2 đưa vào E1 làm khóa ngoại.

2.2. Mô hình dữ liệu quan hệ (Relational Data Model)

Nguyên tắc chuyển đổi (Cont)

Nếu R là mối quan hệ Nhiều-Nhiều giữa 2 tập thực thể E1 và E2 thì tạo lược đồ quan hệ trung gian có tên R gồm tất cả các khóa của E1 và E2, ngoài ra R có thể có thêm một số thuộc tính khác.

Ví dụ: Mối quan hệ DK_Hoc là mối quan hệ nhiều nhiều giữa 2 tập Thực thể SinhVien và MonHoc. Khi đó mối quan hệ DK_Hoc được tạo thành 1 lược đồ quan hệ có tên DK_Hoc gồm các thuộc tính MaSV là khóa của SinhVien và MaMH là khóa của MonHoc, ngoài ra lược đồ quan hệ DK_Hoc còn có thể có thêm một số thuộc tính khác như ThoiGianDK, HocKy, NamHoc.

DK_Hoc(MaSV, MaMH, ThoiGianDK, HocKy, NamHoc)

2.2. Mô hình dữ liệu quan hệ (Relational Data Model)

Đại số quan hệ (Relational Algebra):

Đại số quan hệ dùng trong lý thuyết cơ sở dữ liệu quan hệ, gồm một bộ các toán tử (Operators) và các quy tắc tương ứng có thể được sử dụng để thao tác trên các toán hạng là các quan hệ (*relation*) và tạo ra kết quả là một quan hệ khác.

Đại số quan hệ là nền tảng cho các ngôn ngữ truy vấn CSDL quan hệ.

Các phép toán chính của đại số quan hệ:

Union (hợp), Intersection (giao), difference (hiệu), *Cartesian product* (cross product), Selection (chọn), Projection (chiếu), Join (nối)

2.2. Mô hình dữ liệu quan hệ (Relational Data Model)

Đại số quan hệ (Relational Algebra):

Union (hợp): $Q = R \cup S = \{t | t \in R \vee t \in S\}$

Intersection (giao): $Q = R \cap S = \{t | t \in R \wedge t \in S\}$

Difference (hiệu): $Q = R - S = \{t | t \in R \wedge t \notin S\}$

Cartesian product (cross product)

$$Q = R \times S = \{tq | t \in R \wedge q \in S\}$$

Selection (chọn): $Q = \sigma_{\theta}(R) = \{t | t \in R \wedge \theta(t) = \text{true}\}$

Projection (chiếu): Tập con S có m thuộc tính A_1, A_2, \dots, A_m

$$Q = \pi_S(R) = \{m_bộ\ t | \exists v \in R : v(S) = t\}$$

2.2. Mô hình dữ liệu quan hệ (Relational Data Model)

Đại số quan hệ (Relational Algebra):

Join (nối): Cho 2 quan hệ R và S , quan hệ R có k thuộc tính.

Phép nối 2 quan hệ R và S sao cho cột thứ i của quan hệ R có quan hệ β ($>$, $>=$, $<$, $<=$, $=$, $<>$) với thành phần thứ j trong quan hệ S , được định nghĩa như sau:

$$R \bowtie_{i \beta j} S = \sigma_{i \beta (k+j)} (R \times S)$$

Nối tự nhiên (Natural Join): Nối bằng trên các thuộc tính cùng tên của 2 quan hệ. Ký hiệu $(R \bowtie S)$.

2.2. Mô hình dữ liệu quan hệ (Relational Data Model)

Đại số quan hệ (Relational Algebra):

Một số hàm thống kê:

$Sum(R,A)$: Tính tổng trên thuộc tính A của quan hệ R

$AVG(R,A)$: Tính trung bình trên thuộc tính A của quan hệ R

$Max(R,A)$: Tìm Max trên thuộc tính A của quan hệ R

$Min(R,A)$: Tìm Min trên thuộc tính A của quan hệ R