

Database Management Systems

April 12, 2025

1 ERD

1.1 Thực thể (Entity)

- Thực thể
- Thực thể yếu (Weak)

1.2 Thuộc tính (Attribute)

- Đơn trị
- Đa trị
- Kết hợp
- Suy diễn
- Khoá chính

1.3 Mỗi quan hệ (Relationship)

- 1-1
- 1-n
- n-n
- Vòng

2 RDM

2.1 Thành phần

- Lược đồ quan hệ (Schema Relational)
- Thuộc tính (Attributes)
- Miền giá trị (Domain)
- Bộ (Tuples)
- Lược đồ CSDL (Schema DBMS)
- Mô hình dữ liệu quan hệ (RDM)

2.2 Ký hiệu

$$R(A_1 : D_1, A_2 : D_2, \dots, A_n : D_n)$$

$$R(A_1, A_2, \dots, A_n)$$

$$DOM(A) \text{ or } MGT(A)$$

$$t[A] \text{ or } t.A$$

3 ERD to RDM

3.1 Mỗi quan hệ

IDEA: Lấy khoá của Rx làm khoá chính Ry

3.1.1 1-1

Đối xứng.

3.1.2 1-n

Khoá chính bên **nhiều** đặt làm khoá ngoại bên **ít**.

3.1.3 n-n

$$R(< \text{Khoá ngoại 1} >, < \text{Khoá ngoại 2} >, \dots, < \text{Thuộc tính của quan hệ (nếu có)} >, \dots)$$

3.2 Thực thể yếu

$$R(< \text{key chính của entity phụ thuộc vào} >, A_1, A_2, \dots)$$

- Thực thể R liên kết với thực thể yếu TTY : R chứa khoá chính của tất cả khoá của TTY .

E.g. $TTY(A_1, A_2, \dots, A_n) \rightarrow R(A_1, A_2, \dots, A_n)$.

- Dựa vào phân biệt của thực thể khác \rightarrow thực thể yếu.

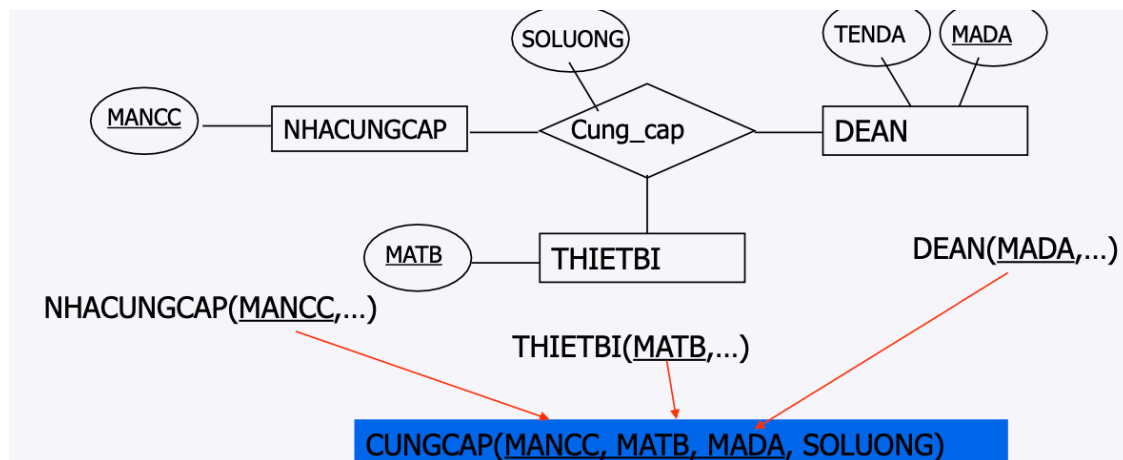
3.3 Thuộc tính đa trị

$$R(< \text{Key chính của Tuple} >, A_1, A_2, \dots)$$



3.4 Liên kết đa ngôi

$R(< KEY1 >, < KEY2 >, < KEY3 >, \dots, < Thuộc tính của quan hệ (nếu có) >, \dots)$



4 Relational Algebra

4.1 Tính khả hợp (Union compatibility)

$$R(A_1, A_2, \dots, A_n) S(B_1, B_2, \dots, B_n)$$

- Cùng bậc n (n attribute).
- $DOM(A_i) = DOM(B_i) \quad \forall i \in \overline{1, n}$

4.2 Phép toán

4.2.1 Giao

- Tập hợp: \cap
- Mệnh đề: \wedge

4.2.2 Hợp

- Tập hợp: \cup
- Mệnh đề: \vee

4.2.3 Trừ

- Tập hợp, mệnh đề: $-$

4.2.4 Chọn

$$\sigma_p(R)$$

- p : Điều kiện (Mệnh đề \leftarrow Phép giao hợp trừ).
- Có tính giao hoán.

4.2.5 Chiếu

$$\pi_{A_1, A_2, \dots, A_k}(R)$$

- A_1, A_2, \dots, A_k : Attribute.
- Trong quá trình chiếu loại bỏ bộ (tuples) trùng nhau. $\left(t[\pi_{A_1, A_2, \dots, A_k}(R)] \leq t[R] \right)$
- Không có tính giao hoán.

4.2.6 Chiếu tổng quát

$$\pi_{F_1(A_1), F_2(A_2), \dots, F_k(A_k)}(R)$$

4.2.7 Gán

$$\leftarrow$$

E.g. $S \leftarrow \sigma_p(R)$, $S \leftarrow \pi_{A_1, A_2, \dots, A_n}(R)$.

4.2.8 Đổi tên

$$\rho_S(R)$$

- R : Quan hệ, với $R(B, C, D)$.
- S : Tên thay thế cho R .
- $R(B, C, D) \rightarrow S(B, C, D)$.

$$\rho_{X, C, D}(R)$$

- $R(B, C, D) \rightarrow R(X, C, D)$.

$$\rho_{S(X, C, D)}(R)$$

- $R(B, C, D) \rightarrow S(X, C, D)$.

4.2.9 Tích Cartesian

$$R \times S$$

- Xuất ra table (tổ hợp) mới với
 - $n + m$ attribute.
 - $u \times v$ tuple.

4.2.10 Kết

Theta

$$R \bowtie_{\text{DK}} S$$

- Xuất ra table mới.

Bằng

$$R \bowtie_{R.A=S.B} S$$

- Xuất ra table mới.

Tự nhiên

$$R \bowtie S$$

- Attribute:
 - Cùng NAME.
 - Cùng *DOM*.
- Tự **delete** Attribute trùng tên.
- Xuất ra table mới.

4.2.11 Chia

$$R(Z) \div S(X)$$

- $X \subseteq Z$.
- Xuất ra $Q(Z - X)$. (Loại bỏ A_i của S trong R)
- Dùng khi: Lấy ra cái này **thoả hết tất cả** cái kia.

E.g. $R(A, B, C, D, E)$ và $S(D, E)$. $R \div S$ ra $Q(A, B, C)$

4.2.12 Hàm kết hợp

- $SUM(A_i)$
- $AVG(A_i)$
- $COUNT(A_i)$
- $MIN(A_i)$
- $MAX(A_i)$

4.2.13 Gom nhóm

$$G_1, G_2, \dots, G_n \mathfrak{J}_{F_1(A_1), F_2(A_2), \dots, F_n(A_n)}(R)$$

- G_i : Attribute cần gom.
- $F_i(A_i)$: Sử dụng hàm kết hợp lên A_i .
- R : Table nguồn.
- Xuất ra table mới.

4.3 Thao tác

4.3.1 Thêm

$$R_{\text{new}} \leftarrow R_{\text{old}} \cup E$$

- E & R khả hợp.
- E : Biểu thức Đại số quan hệ.

E.g. PHANCONG \leftarrow PHANCONG('Bancie', 20, 10).

4.3.2 Xóa

$$R_{\text{new}} \leftarrow R_{\text{old}} - E$$

E.g. PHANCONG \leftarrow PHANCONG $- \sigma_{\text{MANV}='123'}(\text{PHANCONG})$.

4.3.3 Sửa

$$R_{\text{new}} \leftarrow \pi_{F_1(A_1), F_2(A_2), \dots, F_n(A_n)}(R_{\text{old}})$$

4.4 Kiểu thực hiện

4.4.1 Lồng

$$\sigma_p(\pi_{A_1, A_2, \dots, A_k}(\sigma_p(R)))$$

4.4.2 Từng bước

$$\text{B1: } R2 \leftarrow \sigma_p(R1)$$

$$\text{B2: } R3 \leftarrow \pi_{A_1, A_2, \dots, A_k}(R2)$$

\vdots

B_n

5 SQL

5.1 Set up

5.1.1 Tạo bảng

- Constraint **Trực tiếp**

```
CREATE TABLE TableName (  
    Column1 DataType Constraints,  
    Column2 DataType Constraints,  
    ...  
)
```

- Constraint **Tự định nghĩa**

```
CREATE TABLE TableName (
    Column1 DataType,
    Column2 DataType,
    Constraints ConstraintName ConstraintType,
    ...
)
```

- CHAR(x): Số lượng ký tự cố định.
- VARCHAR(x): Số lượng ký tự động (Auto xoá).
- NVARCHAR(x): Tương tự VARCHAR(x) nhưng có dấu được.

5.1.2 RBTV (Constraint)

Tên Con-straint	Ý nghĩa tiếng Việt	Cú pháp (Syntax)	Ví dụ (Example)
PRIMARY KEY	Khóa chính – định danh duy nhất cho mỗi dòng	PRIMARY KEY (ColumnName) hoặc ngay sau tên cột	StudentID INT PRIMARY KEY
FOREIGN KEY	Khóa ngoại – liên kết với khóa chính ở bảng khác	FOREIGN KEY (Column) REFERENCES OtherTable(Column)	CustomerID INT FOREIGN KEY REFERENCES Customers(CustomerID)
NOT NULL	Không cho phép NULL – bắt buộc nhập dữ liệu	ColumnName DataType NOT NULL	Name NVARCHAR(100) NOT NULL
UNIQUE	Giá trị không được trùng lặp trong cột	ColumnName DataType UNIQUE	Email NVARCHAR(100) UNIQUE
CHECK	Kiểm tra điều kiện ràng buộc	CHECK (condition) hoặc ColumnName DataType CHECK (...)	GPA FLOAT CHECK (GPA BETWEEN 0 AND 4)
DEFAULT	Gán giá trị mặc định nếu không nhập	ColumnName DataType DEFAULT (value)	CreatedAt DATETIME DEFAULT GETDATE()
IDENTITY	Tự động tăng giá trị, thường dùng cho khóa chính	ColumnName INT IDENTITY(seed, increment)	OrderID INT PRIMARY KEY IDENTITY(1,1)

- GETDATE(): E.g. output: 2025-04-06 14:32:17.897

5.1.3 Sửa Table

Thêm

- Column

```
ALTER TABLE TableName
ADD ColumnName DataType [CONSTRAINT ...]
```

- Constraint

```
ALTER TABLE TableName
ADD CONSTRAINT ConstraintName ConstraintType
```

Xoá

- Column

```
ALTER TABLE TableName
DROP COLUMN ColumnName
```

- Constraint

```
ALTER TABLE TableName
DROP CONSTRAINT ConstraintName
```

Mở rộng

```
ALTER TABLE TableName
ALTER COLUMN ColumnName NewDomainType
```

5.1.4 Xoá Table

```
DROP TABLE TableName
```

5.2 Truy vấn

$$\pi_A(\sigma_C(B))$$

```
SELECT A FROM B WHERE C
```

$$\wedge \vee \neg$$

```
AND
OR
NOT
```

5.2.1 SELECT

- Chọn Attribute A

```
SELECT A
```

- Chọn hết

```
SELECT *
```

DISTINCT

- Loại bỏ dòng trùng

```
SELECT DISTINCT Cot1, Cot2, ...
FROM TenBang
```


AS

- Đổi tên

```
TenCu AS TenMoi
```

```
TenCu TenMoi
```

Tính toán

```
SELECT <PHEPTOAN(Attribute)> AS KQ
FROM B
```

- E.g.

```
SELECT (Price * Quantity) * (1 - Discount) AS FinalPrice
FROM Products
```

Nối chuỗi

```
SELECT FirstName + ' ' + LastName AS FullName
FROM Employees
```

```
SELECT CONCAT(FirstName, ' ', LastName) AS FullName
FROM Employees
```

5.2.2 FROM

Đơn Table

```
FROM Table1
```

Đa Table

$$TABLE_1 \times TABLE_2 \times TABLE_3 \times \dots$$

```
FROM TABLE1, TABLE2 , TABLE3,...
```

Lưu ý

$$A(C)B(D)$$

```
SELECT *
FROM A, B
WHERE A.C = B.D
```

- Or

```
SELECT *
FROM A AS x, B AS y
WHERE x.C = y.D
```

- Lưu ý

- Số điều kiện dùng để loại **tuple trùng lặp** của $A \times B$ = Số quan hệ (loại trừ quan hệ ko liên quan) giữa A và B .

- A, B, C, \dots trùng Attribute $\rightarrow A.Attribute, B.Attribute, C.Attribute, \dots$

5.2.3 WHERE

BETWEEN

- $a < x < b$

WHERE Column BETWEEN a AND b

- $x < a \wedge x > b$

WHERE Column NOT BETWEEN a AND b

LIKE

- Bằng

WHERE Column LIKE 'xyz'

- Không bằng

WHERE Column NOT LIKE 'xyz'

- Lưu ý
 - Tiếng Việt: N'xyz'
 - Ký tự:
 - * '%xyz%': Chứa xyz.
 - * 'xyz%': Bắt đầu là xyz.
 - * '%xyz': Kết thúc là xyz.

NULL

- Lấy ra Attribute Null trong Column

WHERE Column IS NULL

- Lấy ra Attribute Not Null trong Column

WHERE Column IS NOT NULL

5.3 OTHERS

5.3.1 ORDER BY

SELECT

FROM

WHERE

ORDER BY A, B, ...

- Tăng ASC

ORDER BY A ASC

- Giảm DESC

ORDER BY A DESC

5.3.2 Set

- \cup UNION
- \cap INTERSECT
- $-$ EXCEPT

```
SELECT
FROM
WHERE
<SET>
SELECT
FROM
WHERE
```

- Giữ lại bộ trùng <SET> + ALL

5.3.3 Subquery

Phân cấp (non-correlated)

- Subquery **độc lập** với Outer query.

```
SELECT
FROM
WHERE A <operator> (
    SELECT
    FROM
    WHERE B <operator> (
        ...
    )
)
```

- Lưu ý: Subquery *khả hợp* với Outer query.

Opertators

- IN / NOT IN
- ALL
- ANY
- SOME

Tương quan (non-correlated)

- Attribute trong Subquery **phụ thuộc** Outer query.

```

SELECT
FROM
WHERE <operator> (
    SELECT 1
    FROM
    WHERE <operator> (
        ...
    )
)

```

Opertators

- EXISTS / NOT EXISTS

5.3.4 Hàm kết hợp

```
SELECT <Hàm kết hợp> (AS TEN)
```

- COUNT()
 - COUNT(*)
 - COUNT(A)
 - COUNT(DISTINCT A)
- MIN()
- MAX()
- SUM()
- AVG()

5.3.5 GROUP BY

```

SELECT A, <Hàm kết hợp>
FROM
WHERE
GROUP BY A

```

- Kèm điều kiện

```

SELECT A, <Hàm kết hợp>
FROM
WHERE
GROUP BY A
HAVING <Condition>

```

5.3.6 JOIN

Kiểu kết	Command
$R_1 \bowtie R_2$: Khớp cả 2 bảng	FROM R1 JOIN R2 ON <Condition> or FROM R1 INNER JOIN R2 ON <Condition>
All bên trái + bên phải (nếu khớp)	FROM R1 LEFT JOIN R2 ON <Condition>
All bên phải + bên trái (nếu khớp)	FROM R1 RIGHT JOIN R2 ON <Condition>
All cả 2 bảng	FROM R1 FULL OUTER JOIN R2 ON <Condition>
$R_1 \times R_2$: Tích Descartes	FROM R1 CROSS JOIN R2
$R \bowtie R$: Tự kết	SELECT R R1, R R2 FROM R1 (Kết ngoài) JOIN R2 ON <Condition>

5.3.7 CASE

CASE

```

WHEN condition1 THEN result1
WHEN condition2 THEN result2
...
ELSE result_default

```

END

Vị trí dùng CASE	Ý nghĩa
SELECT	Hiển thị dữ liệu theo điều kiện
ORDER BY	Sắp xếp theo logic tùy chỉnh
WHERE	Tạo điều kiện phức tạp (ít dùng hơn)
GROUP BY	Kết hợp với biểu thức tính nhóm

5.4 STRUCTURE ORDERING

```

SELECT
FROM
WHERE
GROUP BY
HAVING
ORDER BY

```