# Chương 4: Mô hình dữ liệu quan hệ

GV: Nguyễn Hồng Hạnh
Email: hanhnh@nuce.edu.vn

Bộ môn Công nghệ Phần mềm
Khoa Công nghệ Thông Tin, Đại học Xây dựng

1

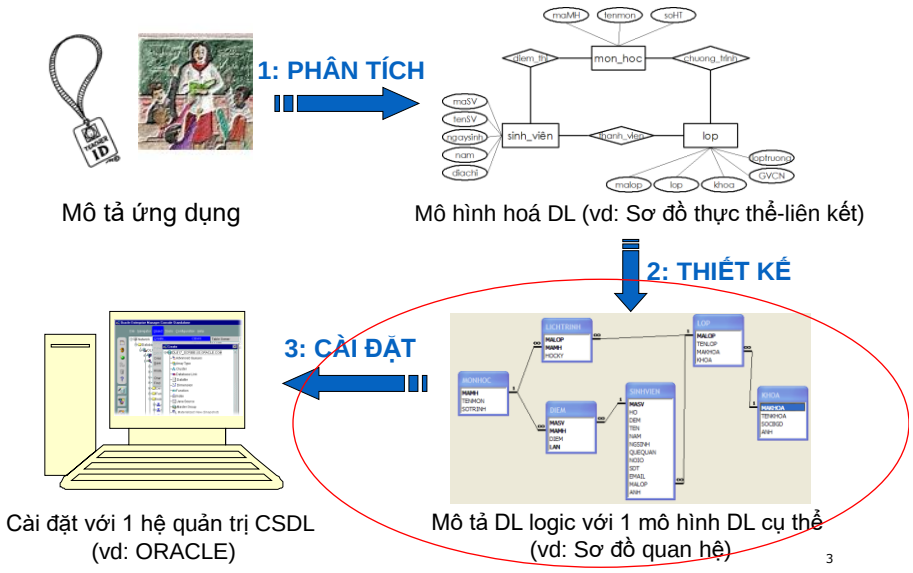
1

## Nội dung chương:

1. Khái niệm cơ bản của mô hình quan hệ
2. Chuyển lược đồ ER sang mô hình quan hệ
3. Ngôn ngữ định nghĩa và thao tác dữ liệu đối với mô hình quan hệ

2

## Các bước xây dựng một hệ CSDL




3

3

## Mô hình dữ liệu quan hệ

- Sự ra đời: vào năm 1970[Codd, 1970]
- Dữ liệu được biểu diễn dưới dạng bảng
- Là mô hình dữ liệu khái niệm phổ biến cho đến tận thời điểm hiện tại
- Dựa trên lý thuyết toán học, đồng thời cũng gần với cấu trúc tệp và cấu trúc dữ liệu nên có hai loại thuật ngữ liên quan:
  - Thuật ngữ toán học: quan hệ, bộ, thuộc tính
  - Thuật ngữ hướng dữ liệu: bảng, bản ghi, trường

4

4

Ví dụ mô hình dữ liệu quan hệ

MON_HOC

| maMH | tenmon | soHT |
|---|---|---|
| CNTT01 | Nhập môn CSDL | 4 |
| CNTT02 | Truyền DL và mạng | 4 |
| CNTT03 | Phân tích và thiết kế hệ thống | 4 |
| HTTT01 | Quản lý dự án | 3 |

LOP

| malop | lop | khoa | GVCN | loptruong |
|---|---|---|---|---|
| IT4 | Tin 4 | CNTT | Ng. V. Anh | Trần T. Bình |
| IT5 | Tin 5 | CNTT | Lê A. Văn | Ng. Đ. Trung |
| IT6 | Tin 6 | CNTT | Ng. T. Thảo | Trần M. Quế |
| IT7 | Tin 7 | CNTT | Ng. V. Quý | Ng. T. Phương |

SINH_VIEN

| maSV | tenSV | ngaysinh | gt | diachi | malop |
|---|---|---|---|---|---|
| SV0011 | Trần T. Bình | 1/4/1981 | 0 | 21 T. Q. B | IT4 |
| SV0025 | Ng. Đ. Trung | 3/2/1980 | 1 | 56 Đ. C. V | IT5 |
| SV0067 | Trần M. Quế | 26/3/1982 | 0 | 45 H. B. T | IT6 |
| SV0034 | Ng. T. Phương | 29/2/1980 | 0 | 86 L. T. N | IT7 |

5

## 1. Khái niệm cơ bản của mô hình quan hệ

### *1.1 Quan hệ*

- Các thông tin lưu trữ trong CSDL được tổ chức thành <u>bảng (table) 2 chiều</u> gọi là quan hệ
- Một quan hệ gồm
  - ***Tên***
  - ***Tập hợp các cột***
  - ***Tập hợp các dòng (tập thực thể) :*** Thay đổi theo thời gian

1 cột là 1 thuộc tính của nhân viên

| TENNV | HONV | NS | DIACHI | GT | LUONG | PHG |
|---|---|---|---|---|---|---|
| Tung | Nguyen | 12/08/1955 | 638 NVC Q5 | Nam | 40000 | 5 |
| Hang | Bui | 07/19/1968 | 332 NTH Q1 | Nu | 25000 | 4 |
| Nhu | Le | 06/20/1951 | 291 HVH QPN | Nu | 43000 | 4 |
| Hung | Nguyen | 09/15/1962 | Ba Ria VT | Nam | 38000 | 5 |

1 dòng là 1 nhân viên

Tên quan hệ là NHANVIEN

6

## 1. Khái niệm cơ bản của mô hình quan hệ

### *1.2 Thuộc tính*

*a. Khái niệm*

*là một tính chất riêng biệt của một đối tượng cần được lưu trữ trong CSDL để phục vụ cho việc khai thác dữ liệu về đối tượng*

*b. Các đặc trưng của thuộc tính*

*- Tên gọi (Name)*

*- Kiểu dữ liệu (Data Type)*

*- Miền giá trị (Domain)*

7

## 1. Khái niệm cơ bản của mô hình quan hệ

### *1.2 Thuộc tính (tiếp)*

***/ Tên gọi:** Do người dùng đặt,có tính chất gợi nhớ, không quá dài, đặt theo quy định của HQT CSDL*

*Lưu ý: Nếu không cần lưu ý đến ngữ nghĩa thì tên thuộc tính thường được ký hiệu bằng các chữ cái A, B, C, D còn X, Y, Z, W, … dùng thay cho 1 nhóm thuộc tính.*

***/ Kiểu dữ liệu:** Mỗi thuộc tính khi định nghĩa đều thuộc về một kiểu dữ liệu xác định*

*- Kiểu văn bản (text)*

*- Kiểu số (number)*

*- Kiểu logic (boolean)*

*- Kiểu ngày giờ (datetime)*

8

## 1. Khái niệm cơ bản của mô hình quan hệ

### *1.2 Thuộc tính*

***/ Miền giá trị(Domain):***

- *Tập hợp các giá trị mà một thuộc tính A có thể nhận được gọi là miền giá trị của A.*
- *Ký hiệu: Dom(A), MGT(A). Ví dụ:* dom(MãSV) = {char(5)}

***Miền thuộc tính:*** *là tập con các giá trị của một kiểu dữ liệu, mà từ đó có thể rút ra các giá trị thực sự xuất hiện trong một thuộc tính của một quan hệ*

*VD: Thuộc tính:DiemTB - MGT(DiemTB)= float*

*- MTT(DiemTB) = (0..10)*

9

## 1. Khái niệm cơ bản của mô hình quan hệ

### *1.3 Định nghĩa hình thức*

- **Lược đồ quan hệ:** là sự trừu tượng hóa của quan hệ ở mức độ cấu trúc của một bảng hai chiều. Ký hiệu $R(A_1:D_1, A_2:D_2, \ldots, A_n:D_n)$ là một lược đồ quan hệ

  Trong đó:
  - $A_1, A_2, \ldots, A_n$ là các thuộc tính
  - Có các miền giá trị $D_1, D_2, \ldots, D_n$ tương ứng

  **Bậc** của lược đồ quan hệ là số lượng thuộc tính trong lược đồ
- Ví dụ: NHANVIEN(MANV:integer, TENNV:string, HONV:string, NGSINH:date, DCHI:string, GT:string, LUONG:integer, DONVI:integer)
  - NHANVIEN là một lược đồ bậc 8 mô tả đối tượng nhân viên
  - MANV là một thuộc tính có miền giá trị là số nguyên
  - TENNV là một thuộc tính có miền giá trị là chuỗi ký tự

10

## 1. Khái niệm cơ bản của mô hình quan hệ

### *1.3 Định nghĩa hình thức*

- **Bộ giá trị (Tuple):** là các thông tin của một đối tượng thuộc lược đồ quan hệ, còn gọi là mẫu tin (record- bản ghi) hay dòng (row)
- Ví dụ: NHANVIEN(MANV:integer, TENNV:string, HONV:string, NGSINH:date, DCHI:string, GT:string, LUONG:integer, DONVI:integer)
- Bộ giá trị của quan hệ NHANVIEN:

t= (1234, "Hoàng", "Nguyễn Văn", 01/06/1993, "Hà Nội", "Nam", 10000000, 34)

11

## 1. Khái niệm cơ bản của mô hình quan hệ

### *1.3 Định nghĩa hình thức*

- **Quan hệ** (thể hiện quan hệ - instance relation)
  - Một quan hệ r của lược đồ quan hệ $R(A_1, A_2, \ldots, A_n)$, ký hiệu r(R), là một tập hợp các bộ giá trị $r = \{t_1, t_2, \ldots, t_k\}$ tại một thời điểm nhất định
  - Trong đó mỗi $t_i$ là 1 danh sách <u>có thứ tự</u> của n giá trị $t_i=<v_1, v_2, \ldots, v_n>$
  - Tại những thời điểm khác nhau thì quan hệ có những thể hiện khác nhau

| | TENNV | HONV | NGSINH | DCHI | PHAI | LUONG | PHG |
|---|---|---|---|---|---|---|---|
| $t_1$ | Tung | Nguyen | 12/08/1955 | 638 NVC Q5 | Nam | 40000 | 5 |
| $t_2$ | Hang | Bui | 07/19/1968 | 332 NTH Q1 | Nu | 25000 | 4 |
| $t_3$ | Nhu | Le | 06/20/1951 | 291 HVH QPN | Nu | 43000 | 4 |
| $t_4$ | Hung | Nguyen | 09/15/1962 | null | Nam | 38000 | 5 |

$v_i$

12

## 1. Khái niệm cơ bản của mô hình quan hệ

### *1.4 Khái niệm về khóa*

- Định nghĩa
  - Gọi K là một tập con khác rỗng các thuộc tính của R
  - K là khóa nếu thỏa đồng thời 2 điều kiện:
  - $\forall r, \forall t1,t2 \in r, t1 \neq t2 \Rightarrow t1[K] \neq t2[K]$
  - $\forall K' \subset K, K' \neq K, K'$ không phải là khóa của R
- Nhận xét
  - Giá trị của khóa dùng để nhận biết một bộ trong quan hệ
  - Khóa là một đặc trưng của lược đồ quan hệ, không phụ thuộc vào thể hiện quan hệ
  - Khóa được xây dựng dựa vào ý nghĩa của một số thuộc tính trong quan hệ
  - Lược đồ quan hệ có thể có nhiều khóa

13

## 1. Khái niệm cơ bản của mô hình quan hệ

### *1.4 Khái niệm về khóa*

- **Khóa chính**

  Xét quan hệ

  NHANVIEN(MANV, TENNV, HONV, NS, DCHI, GT, LUONG, PHG)
  - Có 2 khóa
    - MANV
    - HONV, TENNV, NS
  - Khi cài đặt quan hệ thành bảng (table)
    - Chọn 1 khóa làm cơ sở để nhận biết các bộ (chọn Khóa có ít thuộc tính hơn)
  - Khóa được chọn gọi là khóa chính (PK - primary key)
    - Các thuộc tính khóa chính phải có giá trị khác null
    - Các thuộc tính khóa chính thường được gạch dưới
    - Thuộc tính có tham gia vào 1 khóa gọi là thuộc tính khóa, ngược lại – thuộc tính không khóa.

  NHANVIEN(<u>MANV</u>, TENNV, HONV, NS, DCHI, GT, LUONG, PHG)

14

# 1. Khái niệm cơ bản của mô hình quan hệ

## *1.4 Khái niệm về khóa*

- **Tham chiếu**
  - Một bộ trong quan hệ R, tại thuộc tính A nếu nhận một giá trị từ một thuộc tính B của quan hệ S, ta gọi R tham chiếu S
  - Bộ được tham chiếu phải tồn tại trước

S

| TENPHG | MAPHG |
|---|---|
| Nghien cuu | **5** |
| Dieu hanh | 4 |
| Quan ly | 1 |

R

| TENNV | HONV | NS | DCHI | GT | LUONG | PHG |
|---|---|---|---|---|---|---|
| Tung | Nguyen | 12/08/1955 | 638 NVC Q5 | Nam | 40000 | **5** |
| Hang | Bui | 07/19/1968 | 332 NTH Q1 | Nu | 25000 | 4 |
| Nhu | Le | 06/20/1951 | 291 HVH QPN | Nu | 43000 | 4 |
| Hung | Nguyen | 09/15/1962 | Ba Ria VT | Nam | 38000 | 5 |

15

# 1. Khái niệm cơ bản của mô hình quan hệ

## *1.4 Khái niệm về khóa*

- **Khóa ngoại**
  - Xét 2 lược đồ R và S
    - Gọi FK là tập thuộc tính khác rỗng của R
    - FK là khóa ngoại (Foreign Key) của R khi
    - Các thuộc tính trong FK phải có cùng miền giá trị với các thuộc tính khóa chính của S
    - Giá trị tại FK của một bộ $t_1 \in R$
      - Hoặc bằng giá trị tại khóa chính của một bộ $t_2 \in S$
      - Hoặc bằng giá trị rỗng
  - Ví dụ

Quan hệ tham chiếu

NHANVIEN(MANV, TENNV, HONV, NS, DCHI, GT, LUONG, *PHG*)

PHONGBAN(TENPHG, MAPHG)

Khóa ngoại

Quan hệ bị tham chiếu

Khóa chính

16

## 1. Khái niệm cơ bản của mô hình quan hệ

### *1.5 Các đặc trưng của quan hệ*

- Thứ tự các bộ trong quan hệ là không quan trọng

| HONV | TENNV | NGSINH | DCHI | PHAI | LUONG | PHG |
|---|---|---|---|---|---|---|
| Bui | Hang | 07/19/1968 | 332 NTH Q1 | Nu | 25000 | 4 |
| Le | Nhu | 06/20/1951 | 291 HVH QPN | Nu | 43000 | 4 |

- Thứ tự giữa các giá trị trong một bộ là quan trọng

Bộ <Nguyen, Tung, 12/08/1955, 638 NVC Q5, ***Nam, 40000***, 5>

khác

Bộ <Nguyen, Tung, 12/08/1955, 638 NVC Q5, ***40000, Nam***, 5>

- Không có bộ nào trùng nhau trong cùng một quan hệ

17

## Nội dung chương:

**1.Khái niệm cơ bản của mô hình quan hệ**

**2.Chuyển lược đồ ER sang mô hình quan hệ**

**3.Ngôn ngữ định nghĩa và thao tác dữ liệu đối với mô hình quan hệ**

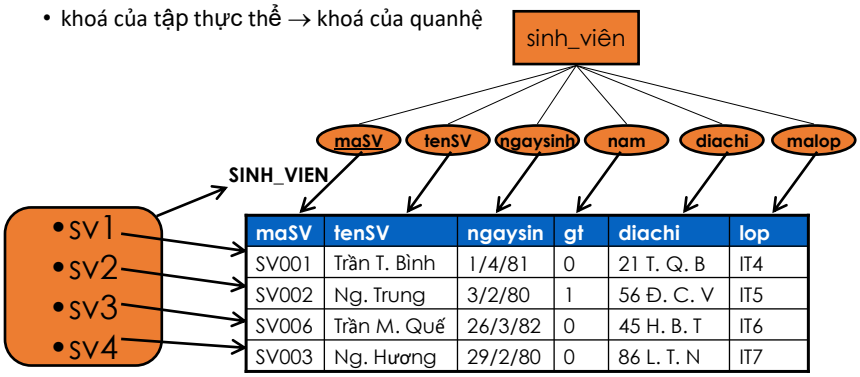
18

## 2. Chuyển lược đồ ER sang mô hình quan hệ

### *Các quy tắc chuyển đổi*

(QT1) Chuyển đối đối với Tập thực thể (trừ thực thể yếu)

- 1 tập thực thể → 1 quan hệ
  - thuộc tính → thuộc tính (trường)
  - 1 thực thể → 1 bộ
  - khoá của tập thực thể → khoá của quanhệ




| maSV | tenSV | ngaysin | gt | diachi | lop |
|---|---|---|---|---|---|
| SV001 | Trần T. Bình | 1/4/81 | 0 | 21 T. Q. B | IT4 |
| SV002 | Ng. Trung | 3/2/80 | 1 | 56 Đ. C. V | IT5 |
| SV006 | Trần M. Quế | 26/3/82 | 0 | 45 H. B. T | IT6 |
| SV003 | Ng. Hương | 29/2/80 | 0 | 86 L. T. N | IT7 |

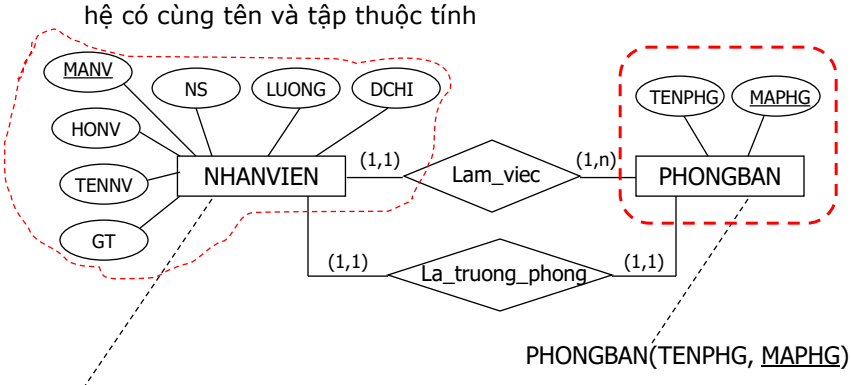
19

## 2. Chuyển lược đồ ER sang mô hình quan hệ

### *Các quy tắc chuyển đổi*

(QT1) Chuyển đối đối với Tập thực thể (tiếp)

- Các tập thực thể (trừ tập thực thể yếu) chuyển thành các quan hệ có cùng tên và tập thuộc tính




PHONGBAN(TENPHG, MAPHG)

NHANVIEN(MANV, TENNV, HONV, NS, DCHI, GT, LUONG)

20

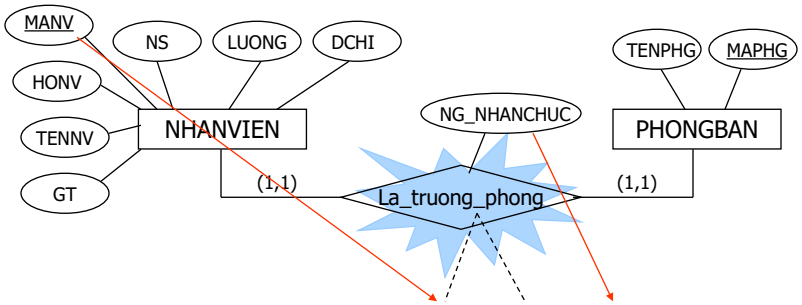
## 2. Chuyển lược đồ ER sang mô hình quan hệ

### *Các quy tắc chuyển đổi*

(QT2) Chuyển đổi đối với Mối quan hệ (Liên kết)

- (2a) Một-Một

  Tạo khóa ngoài: Thêm vào quan hệ này thuộc tính khóa chính của quan hệ kia cùng các thuộc tính của bản thân liên kết (nếu có)




PHONGBAN(MAPHG, TENPHG, **MANV, NG_NHANCHUC**)

21

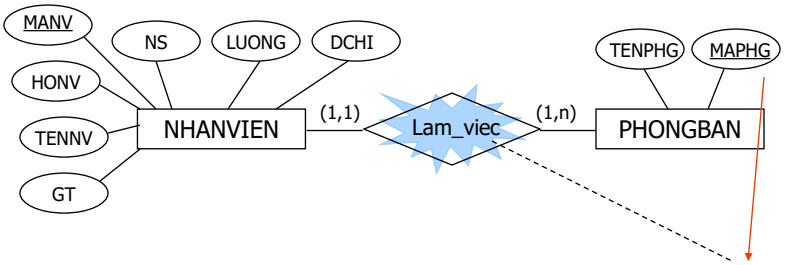
## 2. Chuyển lược đồ ER sang mô hình quan hệ

### *Các quy tắc chuyển đổi*

(QT2) Chuyển đổi đối với Mối quan hệ (Liên kết)

- (2b) Một-Nhiều
  - Thêm vào quan-hệ-nhiều thuộc tính khóa của quan-hệ-một

(Lấy khóa chính của quan hệ bên 1 đưa vào làm khóa ngoài của quan hệ bên n)




NHANVIEN(MANV, TENNV, HONV, NS, DCHI, GT, LUONG, **MAPHG**)

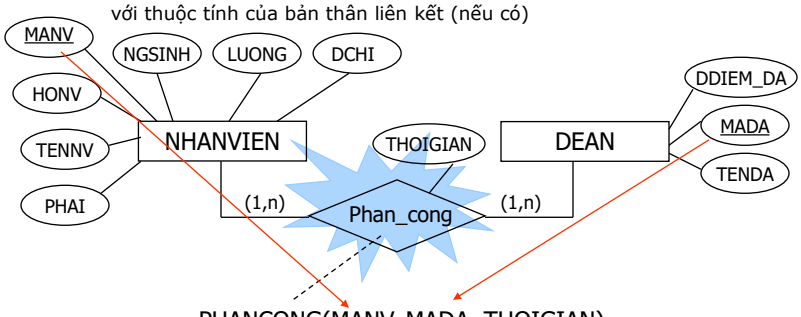
22

## 2. Chuyển lược đồ ER sang mô hình quan hệ

### *Các quy tắc chuyển đổi*

(QT2) Chuyển đổi đối với Mối quan hệ (Liên kết)

- (2c) Nhiều-Nhiều
  - Tạo một quan hệ mới có
    - Tên quan hệ thường là tên của mối quan hệ
    - Thuộc tính là những thuộc tính khóa của các tập thực thể liên quan cùng với thuộc tính của bản thân liên kết (nếu có)




PHANCONG(MANV, MADA, THOIGIAN)

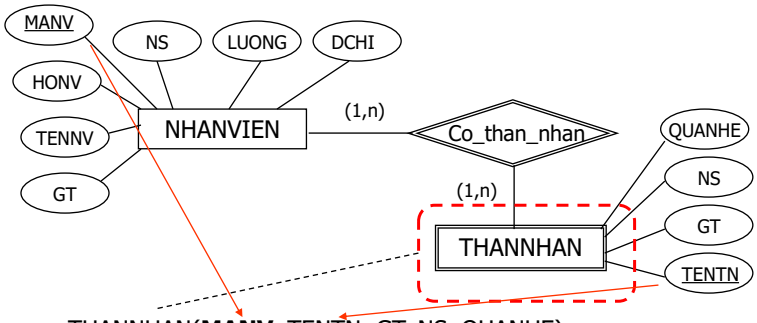
23

## 2. Chuyển lược đồ ER sang mô hình quan hệ

### *Các quy tắc chuyển đổi*

(QT3) Chuyển đổi đối với Thực thể yếu

- Chuyển thành một quan hệ
  - Có cùng tên với thực thể yếu
  - Thêm vào thuộc tính khóa của quan hệ liên quan




THANNHAN(**MANV**, TENTN, GT, NS, QUANHE)

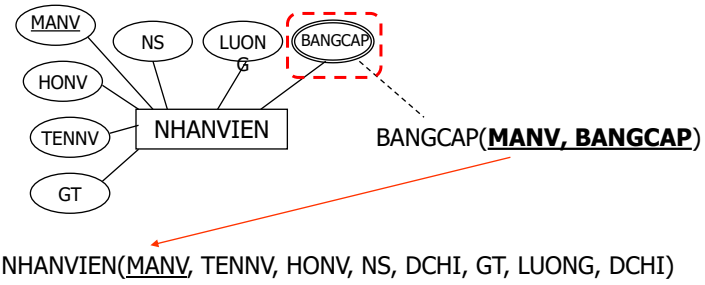
24

## 2. Chuyển lược đồ ER sang mô hình quan hệ

### *Các quy tắc chuyển đổi*

(QT4) Chuyển đổi đối với Thuộc tính đa trị

- Chuyển thành một quan hệ
  - Có cùng tên với thuộc tính đa trị
  - Thuộc tính khóa của quan hệ này được xác định bởi thuộc tính đa trị và khóa của tập thực thể tương ứng.



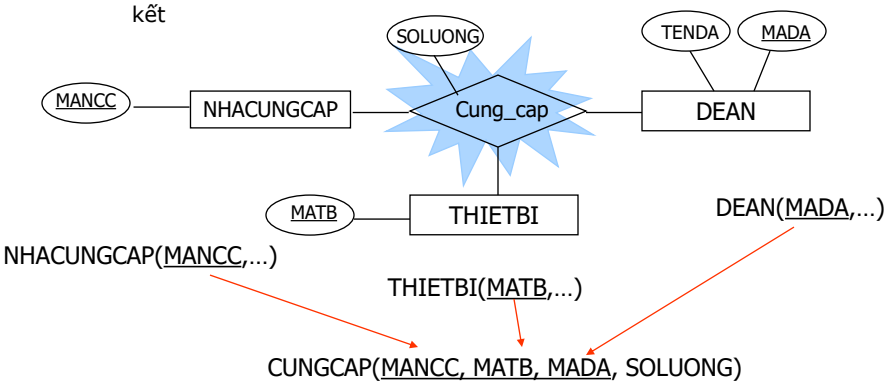

25

## 2. Chuyển lược đồ ER sang mô hình quan hệ

### *Các quy tắc chuyển đổi*

(QT5) Chuyển đổi đối với Liên kết đa ngôi (n>2)

- Chuyển thành một quan hệ
  - Có cùng tên với tên mối liên kết đa ngôi
  - Khóa chính là tổ hợp các khóa của tập các thực thể tham gia liên kết



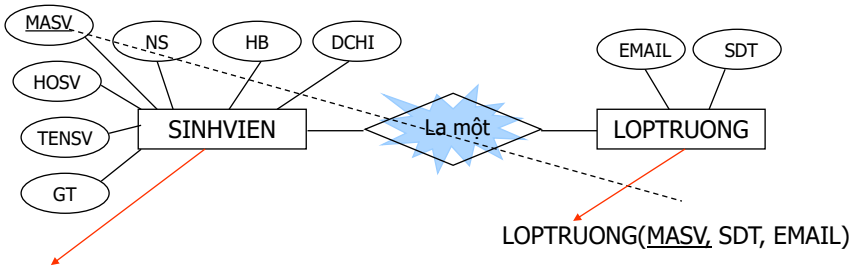

26

## 2. Chuyển lược đồ ER sang mô hình quan hệ

### *Các quy tắc chuyển đổi*

(QT6) Chuyển đổi đối với phân cấp là một (isa)

Một tập thực thể B được xác định thông qua liên kết "là một" với một tập thực thể A sẽ tạo thành một quan hệ mới có tên trùng với tập thực thể B và chứa khóa là khóa của tập thực thể A




LOPTRUONG(MASV, SDT, EMAIL)

SINHVIEN(MASV, HOSV, TENSV, GT, NS, HB, DCHI)

27

## Nội dung chương:

**1.Khái niệm cơ bản của mô hình quan hệ**

**2.Chuyển lược đồ ER sang mô hình quan hệ**

**3.Ngôn ngữ định nghĩa và thao tác dữ liệu đối với mô hình quan hệ**

28

# Nội dung

- Các cách tiếp cận đối với thiết kế ngôn ngữ của CSDL quan hệ
  - Giới thiệu một số ngôn ngữ và phân loại
  - ➢So sánh và đánh giá
- Một số ngôn ngữ dữ liệu mức cao
  - QBE (***Q**uery **B**y **E**xample*)
  - SQL (***S**tructured **Q**uery **L**anguage*)
- Kết luận

29

29

# CSDL ví dụ 1

Student

| Id | Name | Suburb |
|---|---|---|
| 1108 | Robert | Kew |
| 3936 | Glen | Bundoora |
| 8507 | Norman | Bundoora |
| 8452 | Mary | Balwyn |

Takes

| SID | SNO |
|---|---|
| 1108 | 21 |
| 1108 | 23 |
| 8507 | 23 |
| 8507 | 29 |

Enrol

| SID | Course |
|---|---|
| 3936 | 101 |
| 1108 | 113 |
| 8507 | 101 |

Course

| No | Name | Dept |
|---|---|---|
| 113 | BCS | CSCE |
| 101 | MCS | CSCE |

Subject

| No | Name | Dept |
|---|---|---|
| 21 | Systems | CSCE |
| 23 | Database | CSCE |
| 29 | VB | CSCE |
| 18 | Algebra | Maths |

30

30

# CSDL ví dụ 2

Supplier

| SID | SNAME | SIZE | CITY |
|---|---|---|---|
| S1 | Dustin | 100 | London |
| S2 | Rusty | 70 | Paris |
| S3 | Lubber | 120 | London |
| S4 | M&M | 60 | NewYork |
| S5 | MBI | 1000 | NewOrlean |
| S6 | Panda | 150 | London |

Product

| PID | PNAME | COLOR |
|---|---|---|
| P1 | Screw | red |
| P2 | Screw | green |
| P3 | Nut | red |
| P4 | Bolt | blue |
| P5 | Plier | green |
| P6 | Scissors | blue |

SupplyProduct

| SID | PID | QUANTITY |
|---|---|---|
| S1 | P1 | 500 |
| S1 | P2 | 400 |
| S1 | P4 | 100 |
| S2 | P3 | 250 |
| S2 | P4 | 50 |
| S3 | P1 | 300 |
| S3 | P2 | 350 |
| S3 | P6 | 200 |
| S4 | P1 | 10 |
| S5 | P2 | 200 |

31

31

# Đặt vấn đề: các câu hỏi

- Tìm tên của các sinh viên nào sống ở Bundoora
  - Tìm các bộ của bảng Student có Suburb = Bundoora
  - Đưa ra các giá trị của thuộc tính Name của các bộ này

Student

| Id | Name | Suburb |
|---|---|---|
| 1108 | Robert | Kew |
| 3936 | Glen | Bundoora |
| 8507 | Norman | Bundoora |
| 8452 | Mary | Balwyn |

32

32

# Câu hỏi (tiếp)

- Tìm các sinh viên đăng ký khoá học có mã số 113
  - Tìm các giá trị SID trong bảng Enrol có Course tương ứng là 113
  - Đưa các bộ của bảng Student có SID trong các giá trị tìm thấy ở trên

Student

| Id | Name | Suburb |
|---|---|---|
| 1108 | Robert | Kew |
| 3936 | Glen | Bundoora |
| 8507 | Norman | Bundoora |
| 8452 | Mary | Balwyn |

Enrol

| SID | Course |
|---|---|
| 3936 | 101 |
| 1108 | 113 |
| 8507 | 101 |

Course

| No | Name | Dept |
|---|---|---|
| 113 | BCS | CSCE |
| 101 | MCS | CSCE |

33

33

## Nội dung chương:

**1.Khái niệm cơ bản của mô hình quan hệ**

**2.Chuyển lược đồ ER sang mô hình quan hệ**

**3.Ngôn ngữ định nghĩa và thao tác dữ liệu đối với mô hình quan hệ**

- **3.1 Ngôn ngữ Đại số quan hệ**
- **3.2 Ngôn ngữ QBE**
- **3.3 Ngôn ngữ truy vấn hướng cấu trúc SQL**

34

# Ngôn ngữ đại số quan hệ

35

35

# Tổng quan

- Gồm các phép toán tương ứng với các thao tác trên các quan hệ
- Mỗi phép toán ngo
  - Đầu vào: một hay nhiều quan hệ
  - Đầu ra: một quan hệ
- Biểu thức đại số quan hệ = chuỗi các phép toán
- Kết quả thực hiện một biểu thức đại số là một quan hệ
- Được cài đặt trong phần lớn các hệ CSDL hiện nay

=>Ngôn ngữ đại số quan hệ bao gồm các phép toánn và phép toán tương ứng trên các bảng.

36

36

## Phân loại các phép toán đại số quan hệ

- Phép toán tập hợp
  - Phép hợp (*union*)
  - Phép giao (*intersection*)
  - Phép trừ (*difference*)
  - Phép tích đề-các (*cartesian product*)
- Phép toán quan hệ
  - Phép chiếu (*projection*)
  - Phép chọn (*selection*)
  - Phép kết nối (*join*)
  - Phép chia (*division*)

37

37

## Nhóm phép toán tập hợp

- Định nghĩa: **Quan hệ khả hợp** ( là có khả năng kết hợp)

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

- r xác định trên $D_1 x\ D_2 x \ldots x\ D_n$
- s xác định trên $D'_1 x\ D'_2 x \ldots x\ D'_m$
- → $D_i = D'_i$ và n=m

38

38

# Phép hợp (union)

là phép ghép các bản ghi thuộc ít nhất 1 trong 2 bảng đầu vào, và 2 bảng đầu vào phải là 2 quan hệ khả hợp

- Đ/n: gồm các bộ thuộc ít nhất 1 trong 2 quan hệ đầu vào
- 2 quan hệ đầu vào phải là khả hợp
- Cú pháp: $R = R_1 \cup R_2$

Subject1

| Name | Course |
|---|---|
| Systems | BCS |
| Database | BCS |
| Database | MCS |
| Algebra | MCS |

∪

Subject2

| Name | Course |
|---|---|
| DataMining | MCS |
| Writing | BCS |

Kết quả

| Name | Course |
|---|---|
| Systems | BCS |
| Database | BCS |
| Database | MCS |
| Algebra | MCS |
| DataMining | MCS |
| Writing | BCS |

39

# Phép giao (intersection)

điều kiện: phép giao cũng làm việc trên 2 quan hệ khả hợp.

- Đ/n: gồm các bộ thuộc cả hai quan hệ đầu vào
- Cú pháp: $R_1 \cap R_2$

Subject1

| Name | Course |
|---|---|
| Systems | BCS |
| Database | BCS |
| Database | MCS |
| Algebra | MCS |

∩

Subject2

| Name | Course |
|---|---|
| DataMining | MCS |
| Database | MCS |
| Systems | BCS |
| Writing | BCS |

Kết quả

| Name | Course |
|---|---|
| Systems | BCS |
| Database | MCS |

40

40

# Phép trừ (minus) điều kiện: phép giao cũng làm việc trên 2 quan hệ khả hợp.

- Đ/n: gồm các bộ thuộc quan hệ thứ nhất nhưng không thuộc quan hệ thứ hai
  - 2 quan hệ phải là khả hợp
- Cú pháp: $R_1 \setminus R_2$ hoặc $R_1$ - $R_2$

Subject1

| Name | Course |
|---|---|
| Systems | BCS |
| Database | BCS |
| Database | MCS |
| Algebra | MCS |

\

Subject2

| Name | Course |
|---|---|
| DataMining | MCS |
| Database | MCS |
| Systems | BCS |
| Writing | BCS |

Kết quả

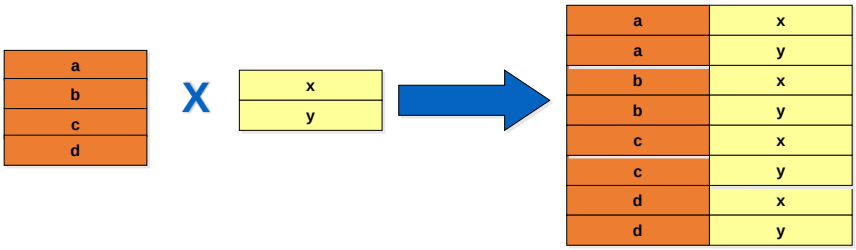
| Name | Course |
|---|---|
| Database | BCS |
| Algebra | MCS |

41

41

# Phép tích Đề-các (Cartesian Product)

- Đ/n: là kết nối giữa từng bộ của quan hệ thứ nhất với mỗi bộ của quan hệ thứ hai.
- Cú pháp: $R = R_1 \times R_2$




42

42

# Ví dụ phép tích Đề-các

Student

| Id | Name | Suburb |
|---|---|---|
| 1108 | Robert | Kew |
| 3936 | Glen | Bundoora |
| 8507 | Norman | Bundoora |
| 8452 | Mary | Balwyn |

x

Sport

| SportID | Sport |
|---|---|
| 05 | Swimming |
| 09 | Dancing |

Student_Sport

| Id | Name | Suburb | SportID | Sport |
|---|---|---|---|---|
| 1108 | Robert | Kew | 05 | Swimming |
| 1108 | Robert | Kew | 09 | Dancing |
| 3936 | Glen | Bundoora | 05 | Swimming |
| 3936 | Glen | Bundoora | 09 | Dancing |
| 8507 | Norman | Bundoora | 05 | Swimming |
| 8507 | Norman | Bundoora | 09 | Dancing |
| 8452 | Mary | Balwyn | 05 | Swimming |
| 8452 | Mary | Balwyn | 09 | Dancing |

43

43

# Nhóm phép toán quan hệ

Là nhóm các phép toán làm việc trên một hoặc nhiều các quan hệ (bảng)

- Phép chiếu (*projection*)
- Phép chọn (*selection*)
- Phép kết nối (*join*)
- Phép chia (*division*)

44

44

# Phép chiếu (projection)

- Đ/n: Lựa chọn một số thuộc tính từ một quan hệ.
- Cú pháp: $\prod_{A1,A2,...}(R)$

❖Ví dụ: đưa ra danh sách tên của tất cả các sinh viên

$$\prod_{Name}(Student)$$

Student

| Id | Name | Suburb |
|---|---|---|
| 1108 | Robert | Kew |
| 3936 | Glen | Bundoora |
| 8507 | Norman | Bundoora |
| 8452 | Mary | Balwyn |

| Name |
|---|
| Robert |
| Glen |
| Norman |
| Mary |

45

45

# Phép chọn

- Đ/n: Lựa chọn các bộ trong một quan hệ thoả mãn điều kiện cho trước.
- Cú pháp: $\sigma_{<condition>}(R)$

- Ví dụ: đưa ra danh sách những sinh viên sống ở Bundoora

$$\sigma_{suburb='Bundoora'}(Student)$$

Student

| Id | Name | Suburb |
|---|---|---|
| 1108 | Robert | Kew |
| 3936 | Glen | Bundoora |
| 8507 | Norman | Bundoora |
| 8452 | Mary | Balwyn |

| Id | Name | Suburb |
|---|---|---|
| 3936 | Glen | Bundoora |
| 8507 | Norman | Bundoora |

46

46

# Phép chọn - Điều kiện (conditional)

- Điều kiện chọn còn gọi là biểu thức chọn.
- Biểu thức chọn F:

*Là một tổ hợp logic của các toán hạng. Mỗi toán hạng là một phép so sánh đơn giản giữa 2 biến là hai thuộc tính hoặc giữa 1 biến là 1 thuộc tính và 1 giá trị hằng.*

- Các phép so sánh trong F: $<, =, >, \leq, \geq, \neq$
- Các phép toán logic trong F: $\wedge, \vee, \neg$

47

47

# Ví dụ: kết hợp chọn và chiếu

- Viết biểu thức ĐSQH: Đưa ra tên của các sinh viên sống ở Bundoora

$$\prod_{Name}(\sigma_{suburb='Bundoora'} Student)$$

Student

| Id | Name | Suburb |
|---|---|---|
| 1108 | Robert | Kew |
| 3936 | Glen | Bundoora |
| 8507 | Norman | Bundoora |
| 8452 | Mary | Balwyn |

| Name |
|---|
| Glen |
| Norman |

48

48

## Phép kết nối (join) 2 quan hệ R và S

- Khái niệm ghép bộ: u = $(a_1,..,a_n)$; v=$(b_1,..,b_m)$
  (u,v) = $(a_1,..,a_n,b_1,..,b_m)$
- Phép kết nối 2 quan hệ thực chất là phép ghép các cặp bộ của 2 quan hệ **thỏa 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: R $\bowtie_{<<\text{điều kiện}>>}$ S

49

49

## Phép kết nối - Ví dụ:

- Đưa ra danh sách các sinh viên và mã khoá học mà sinh viên đó tham gia: Student $\bowtie_{Id=SID}$ Enrol

Student

| Id | Name | Suburb |
|---|---|---|
| 1108 | Robert | Kew |
| 3936 | Glen | Bundoora |
| 8507 | Norman | Bundoora |
| 8452 | Mary | Balwyn |

$\bowtie_{Id=SID}$

Enrol

| SID | Course |
|---|---|
| 3936 | 101 |
| 1108 | 113 |
| 8507 | 101 |

Kết quả

| SID | Id | Name | Suburb | Course |
|---|---|---|---|---|
| 1108 | 1108 | Robert | Kew | 113 |
| 3936 | 3936 | Glen | Bundoora | 101 |
| 8507 | 8507 | Norman | Bundoora | 101 |

50

50

## 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 nghĩa: 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 gọi là **kết nối tự nhiên**
- Cú pháp phép kết nối tự nhiên: $R_1 * R_2$

51

51

## Phép kết nối tự nhiên - Ví dụ:

Takes

| SID | SNO |
|---|---|
| 1108 | 21 |
| 1108 | 23 |
| 8507 | 23 |
| 8507 | 29 |

*

Enrol

| SID | Course |
|---|---|
| 3936 | 101 |
| 1108 | 113 |
| 8507 | 101 |

| SID | SNO | Course |
|---|---|---|
| 1108 | 21 | 113 |
| 1108 | 23 | 113 |
| 8507 | 23 | 101 |
| 8507 | 29 | 101 |

52

52

# Ví dụ: kết hợp chọn, chiếu, kết nối

- Viết BT ĐSQH đưa ra tên của các sinh viên sống ở Bundoora và mã khoá học mà sinh viên đó đăng ký:

$$\prod_{Name,Course}(\sigma_{Suburb='Bundoora'}(Student \bowtie_{Id=SID} Enrol))$$

Student

| Id | Name | Suburb |
|---|---|---|
| 1108 | Robert | Kew |
| 3936 | Glen | Bundoora |
| 8507 | Norman | Bundoora |
| 8452 | Mary | Balwyn |

Enrol

| SID | Course |
|---|---|
| 3936 | 101 |
| 1108 | 113 |
| 8507 | 101 |

Kết quả

| Name | Course |
|---|---|
| Glen | 101 |
| Norman | 101 |

53

53

# Phép kết nối ngoài

- Phép kết nối ngoài trái

| | |
|---|---|
| a | r |
| b | r |
| c | v |

$\bowtie$

| | |
|---|---|
| r | x |
| s | y |
| t | z |

| | | |
|---|---|---|
| a | r | x |
| b | r | x |
| c | v | null |

- Phép kết nối ngoài phải

| | |
|---|---|
| a | r |
| b | r |
| c | v |

$\bowtie$

| | |
|---|---|
| r | x |
| s | y |
| t | z |

| | | |
|---|---|---|
| a | r | x |
| b | r | x |
| null | s | y |
| null | t | z |

54

54

# Phép kết nối ngoài - Ví dụ:

- Đưa ra danh sách các sinh viên và mã khoá học mà sinh viên đó đăng ký nếu có

Student

| ID | Name | Suburb |
|---|---|---|
| 1108 | Robert | Kew |
| 3936 | Glen | Bundoora |
| 8507 | Norman | Bundoora |
| 8452 | Mary | Balwyn |

$\bowtie$

Enrol

| SID | Course |
|---|---|
| 3936 | 101 |
| 1108 | 113 |
| 8507 | 101 |

Kết quả

| ID | Name | Suburb | Course |
|---|---|---|---|
| 1108 | Robert | Kew | 113 |
| 3936 | Glen | Bundoora | 101 |
| 8507 | Norman | Bundoora | 101 |
| 8452 | Mary | Balwyn | **null** |

55

55

# Phép chia (division)

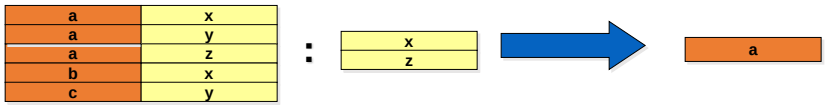
- Định nghĩa: Phép chia giữa 1 quan hệ r bậc n và quan hệ s bậc m (m<n) với sơ đồ quan hệ của s là tập con của sơ đồ quan hệ của r là một tập các (n-m)-bộ sao cho khi ghép mọi bộ thuộc s với t thì ta đều có một bộ thuộc r
- Cú pháp: $R = R_1 : R_2$

$$r \div s = \{ t \mid \forall\, v \in s \Rightarrow (t,v) \in r \}$$

56

56

# Phép chia (tiếp)




- Ví dụ: Đưa ra môn học được dạy ở tất cả các khoá học

Subject

| Name | Course |
|---|---|
| Systems | BCS |
| Database | BCS |
| Database | MCS |
| Algebra | MCS |

:

Course

| Course |
|---|
| BCS |
| MCS |

Kết quả

| Name |
|---|
| Database |

57

57

# Luyện tập

- Phép hợp (Union)

$$r \cup s = \{ t \mid t \in r \vee t \in s \}$$

Ví dụ:

r (A B C)

| | | |
|---|---|---|
| $a_1$ | $b_1$ | $c_1$ |
| $a_1$ | $b_1$ | $c_2$ |
| $a_1$ | $b_2$ | $c_2$ |
| $a_2$ | $b_2$ | $c_2$ |
| $a_3$ | $b_2$ | $c_2$ |

s (A B C)

| | | |
|---|---|---|
| $a_1$ | $b_1$ | $c_1$ |
| $a_1$ | $b_2$ | $c_1$ |
| $a_1$ | $b_2$ | $c_2$ |

$r \cup s = h$ (A B C)

| | | |
|---|---|---|
| $a_1$ | $b_1$ | $c_1$ |
| $a_1$ | $b_1$ | $c_2$ |
| $a_1$ | $b_2$ | $c_1$ |
| $a_1$ | $b_2$ | $c_2$ |
| $a_2$ | $b_2$ | $c_2$ |
| $a_3$ | $b_2$ | $c_2$ |

58

58

# Luyện tập

- Phép giao (intersection)

$$r \cap s = \{ t \mid t \in r \wedge t \in s \}$$

Ví dụ:

r

| (A | B | C) |
|---|---|---|
| $a_1$ | $b_1$ | $c_1$ |
| $a_1$ | $b_1$ | $c_2$ |
| $a_1$ | $b_2$ | $c_2$ |
| $a_2$ | $b_2$ | $c_2$ |
| $a_3$ | $b_2$ | $c_2$ |

s

| (A | B | C) |
|---|---|---|
| $a_1$ | $b_1$ | $c_1$ |
| $a_1$ | $b_2$ | $c_1$ |
| $a_1$ | $b_2$ | $c_2$ |

$r \cap s = g$

| (A | B | C) |
|---|---|---|
| $a_1$ | $b_1$ | $c_1$ |
| $a_1$ | $b_2$ | $c_2$ |

59

59

# Luyện tập

- Phép trừ (minus)

$$r - s = \{ t \mid t \in r \wedge t \notin s \}$$

Ví dụ:

r

| (A | B | C) |
|---|---|---|
| $a_1$ | $b_1$ | $c_1$ |
| $a_1$ | $b_1$ | $c_2$ |
| $a_1$ | $b_2$ | $c_2$ |
| $a_2$ | $b_2$ | $c_2$ |
| $a_3$ | $b_2$ | $c_2$ |

s

| (A | B | C) |
|---|---|---|
| $a_1$ | $b_1$ | $c_1$ |
| $a_1$ | $b_2$ | $c_1$ |
| $a_1$ | $b_2$ | $c_2$ |

$r - s = t$

| (A | B | C) |
|---|---|---|
| $a_1$ | $b_1$ | $c_2$ |
| $a_2$ | $b_2$ | $c_2$ |
| $a_3$ | $b_2$ | $c_2$ |

60

60

# Luyện tập

- Phép tích Đề - Các (Cartesian Product)

$$r \times s = \{t \mid t = (a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_m) \wedge (a_1, a_2, \ldots, a_n) \in r \wedge (b_1, b_2, \ldots, b_m) \in s\}$$

Ví dụ:

r

| (A | B | C) |
|---|---|---|
| $a_1$ | $b_1$ | 1 |
| $a_2$ | $b_2$ | 2 |
| $a_3$ | $b_3$ | 3 |

s

| (D | E) |
|---|---|
| 1 | $e_1$ |
| 2 | $e_2$ |
| 3 | $e_3$ |

r×s = p

| (A | B | C | D | E) |
|---|---|---|---|---|
| $a_1$ | $b_1$ | 1 | 1 | $e_1$ |
| $a_1$ | $b_1$ | 1 | 2 | $e_2$ |
| $a_1$ | $b_1$ | 1 | 3 | $e_3$ |
| $a_2$ | $b_2$ | 2 | 1 | $e_1$ |
| $a_2$ | $b_2$ | 2 | 2 | $e_2$ |
| $a_2$ | $b_2$ | 2 | 3 | $e_3$ |
| $a_3$ | $b_3$ | 3 | 1 | $e_1$ |
| $a_3$ | $b_3$ | 3 | 2 | $e_2$ |
| $a_3$ | $b_3$ | 3 | 3 | $e_3$ |

61

61

# Luyện tập

- Phép chiếu (Projection)

$$\prod_X (r) = \{ t[X] \mid t \in r \}$$

Ví dụ: $X = \{A, B\}$ ; $Y = \{C\}$

r

| (A | B | C) |
|---|---|---|
| $a_1$ | $b_1$ | $c_1$ |
| $a_1$ | $b_1$ | $c_2$ |
| $a_1$ | $b_2$ | $c_2$ |
| $a_2$ | $b_2$ | $c_2$ |
| $a_3$ | $b_2$ | $c_2$ |

$\Pi_X(r) = s_1$

| (A | B) |
|---|---|
| $a_1$ | $b_1$ |
| $a_1$ | $b_2$ |
| $a_2$ | $b_2$ |
| $a_3$ | $b_2$ |

62

62

# Luyện tập

- Phép chọn (Selection)

$$\sigma_F(r) = \{ t \mid t \in r \wedge F(t) = \text{đúng} \}$$

Ví dụ:

r

| A | B | C |
|---|---|---|
| $a_1$ | $b_1$ | $c_1$ |
| $a_1$ | $b_1$ | $c_2$ |
| $a_1$ | $b_2$ | $c_2$ |
| $a_2$ | $b_2$ | $c_2$ |
| $a_3$ | $b_2$ | $c_2$ |

$\sigma_{A=a1}(r) = r_1$

| A | B | C |
|---|---|---|
| $a_1$ | $b_1$ | $c_1$ |
| $a_1$ | $b_2$ | $c_2$ |
| $a_1$ | $b_1$ | $c_2$ |

$\sigma_{A=a1 \wedge C=c2}(r) = r_2$

| A | B | C |
|---|---|---|
| $a_1$ | $b_1$ | $c_2$ |
| $a_1$ | $b_2$ | $c_2$ |

63

63

# Luyện tập

- Phép kết nối (join)

$$r \bowtie_F s = \{ t \mid t = (u,v) \wedge u \in r \wedge v \in s \wedge F(t) = \text{đúng} \}$$

Ví dụ:

$F = (C \le D)$; $F' = (C = D)$

r

| A | B | C |
|---|---|---|
| $a_1$ | $b_1$ | 1 |
| $a_2$ | $b_2$ | 2 |
| $a_3$ | $b_3$ | 3 |

s

| D | E |
|---|---|
| 1 | $e_1$ |
| 2 | $e_2$ |
| 3 | $e_3$ |

$r \bowtie_F s = k$

| A | B | C | D | E |
|---|---|---|---|---|
| $a_1$ | $b_1$ | 1 | 1 | $e_1$ |
| $a_1$ | $b_1$ | 1 | 2 | $e_2$ |
| $a_1$ | $b_1$ | 1 | 3 | $e_3$ |
| $a_2$ | $b_2$ | 2 | 2 | $e_2$ |
| $a_2$ | $b_2$ | 2 | 3 | $e_3$ |
| $a_3$ | $b_3$ | 3 | 3 | $e_3$ |

$r \bowtie_{F'} s = k'$

| A | B | C | D | E |
|---|---|---|---|---|
| $a_1$ | $b_1$ | 1 | 1 | $e_1$ |
| $a_2$ | $b_2$ | 2 | 2 | $e_2$ |
| $a_3$ | $b_3$ | 3 | 3 | $e_3$ |

64

64

## Luyện tập

- Kết nối tự nhiên (natural join)

$$r(U) * s(V) = \{ t[U \cup V] \mid t[U] \in r \wedge t[V] \in s \}$$

p

| (A | B | C) |
|---|---|---|
| $a_1$ | $b_1$ | 1 |
| $a_2$ | $b_1$ | 1 |
| $a_3$ | $b_2$ | 2 |

q

| (C | D) |
|---|---|
| 1 | $d_1$ |
| 1 | $d_2$ |
| 2 | $d_2$ |

p*q = z

| (A | B | C | D) |
|---|---|---|---|
| $a_1$ | $b_1$ | 1 | $d_1$ |
| $a_1$ | $b_1$ | 1 | $d_2$ |
| $a_2$ | $b_1$ | 1 | $d_1$ |
| $a_2$ | $b_1$ | 1 | $d_2$ |
| $a_3$ | $b_2$ | 2 | $d_2$ |

65

65

## Luyện tập

- Phép chia (Division)

$$r \div s = \{ t \mid \forall\ v \in s \Rightarrow (t,v) \in r \}$$

Ví dụ:

p

| (A | B | C | D | E) |
|---|---|---|---|---|
| $a_1$ | $b_1$ | 1 | 1 | $e_1$ |
| $a_1$ | $b_1$ | 1 | 2 | $e_2$ |
| $a_1$ | $b_1$ | 1 | 3 | $e_3$ |
| $a_2$ | $b_2$ | 2 | 1 | $e_1$ |
| $a_2$ | $b_2$ | 2 | 2 | $e_2$ |
| $a_2$ | $b_2$ | 2 | 3 | $e_3$ |
| $a_3$ | $b_3$ | 3 | 1 | $e_1$ |
| $a_3$ | $b_3$ | 3 | 2 | $e_2$ |
| $a_3$ | $b_3$ | 3 | 3 | $e_3$ |

s

| (D | E) |
|---|---|
| 1 | $e_1$ |
| 2 | $e_2$ |
| 3 | $e_3$ |

p ÷ s = q

| (A | B | C) |
|---|---|---|
| $a_1$ | $b_1$ | 1 |
| $a_2$ | $b_2$ | 2 |
| $a_3$ | $b_3$ | 3 |

66

66

# Bài tập

- Cho CSDL gồm 3 quan hệ sau: S(Các hãng cung ứng), P (các mặt hàng), SP(các sự cung ứng).

S (S#, SNAME, STATUS, CITY)

| S# | SNAME | STATUS | CITY |
|---|---|---|---|
| S1 | Smith | 20 | London |
| S2 | Jones | 10 | Paris |
| S3 | Black | 30 | Paris |

SP (S#, P#, QTY)

| S# | P# | QTY |
|---|---|---|
| S1 | P1 | 300 |
| S1 | P2 | 200 |
| S1 | P3 | 400 |
| S2 | P1 | 300 |
| S2 | P2 | 400 |
| S3 | P2 | 200 |

P (P#, PNAME, COLOR, WEIGHT, CITY)

| P# | PNAME | COLOR | WEIGHT | CITY |
|---|---|---|---|---|
| P1 | Nut | red | 12 | London |
| P2 | Bolt | green | 17 | Paris |
| P3 | Screw | blue | 17 | Rom |
| P4 | Screw | red | 14 | London |

67

67

# Yêu cầu của bài tập

- Biểu diễn các truy vấn sau bằng đại số quan hệ:
  - Đưa ra danh sách các mặt hàng màu đỏ
  - Cho biết S# của các hãng cung ứng mặt hàng 'P1' hoặc 'P2'
  - Liệt kê S# của các hãng cung ứng cả hai mặt hàng 'P1' và 'P2'
  - Đưa ra S# của các hãng cung ứng ít nhất một mặt hàng màu đỏ
  - Đưa ra S# của các hãng cung ứng tất cả các mặt hàng.

68

68

# Lời giải của bài tập

- Đưa ra danh sách các mặt hàng màu đỏ:

$$\sigma_{COLOR = 'red'}(P)$$

- Cho biết S# của các hãng cung ứng mặt hàng 'P1' hoặc 'P2':

$$\Pi_{S\#}(\sigma_{P\# = 'P1' \vee P\# = 'P2'}(SP))$$

- Liệt kê S# của các hãng cung ứng cả hai mặt hàng 'P1' và 'P2':

$$\Pi_{S\#}(\sigma_{P\# = 'P1'}(SP)) \cap \Pi_{S\#}(\sigma_{P\# = 'P2'}(SP))$$

- Đưa ra S# của các hãng cung ứng ít nhất một mặt hàng màu đỏ:

$$\Pi_{S\#}(SP * \sigma_{COLOR = 'red'}(P))$$

- Đưa ra S# của các hãng cung ứng tất cả các mặt hàng:

$$\Pi_{S\#,P\#}(SP) \div \Pi_{P\#}(P)$$

69

69

# Bài tập về nhà

- Cho các quan hệ sau:

Supplier

| sid | sname | size | city |
|---|---|---|---|
| S1 | Dustin | 100 | London |
| S2 | Rusty | 70 | Paris |
| S3 | Lubber | 120 | London |

Product

| pid | pname | colour |
|---|---|---|
| P1 | Screw | red |
| P2 | Screw | green |
| P3 | Nut | red |
| P4 | Bolt | blue |

SupplyProduct

| sid | pid | quantity |
|---|---|---|
| S1 | P1 | 500 |
| S1 | P2 | 400 |
| S1 | P3 | 100 |
| S2 | P2 | 200 |
| S3 | P4 | 100 |
| S2 | P3 | 155 |

70

70

## Bài tập về nhà

- Biểu diễn các truy vấn sau bằng biểu thức đại số quan hệ:
  1) Đưa ra {sid,sname,size,city} của các Supplier có trụ sở tại London
  2) Đưa ra {pname} của tất cả các mặt hàng
  3) Đưa ra {sid} của các Supplier cung cấp mặt hàng P1 hoặc P2
  4) Đưa ra {sname} của các Supplier cung cấp mặt hàng P3
  5) Đưa ra {sname} của các hãng cung ứng ít nhất một mặt hàng màu đỏ

71

71

## Bài tập về nhà

6) Đưa ra {sid} của các hãng cung ứng tất cả các mặt hàng màu đỏ
7) Đưa ra {sname} của các hãng có cung ứng mặt hàng màu đỏ hoặc màu xanh
8) Đưa ra {sname} của các hãng cung ứng ít nhất một mặt hàng màu đỏ và ít nhất một mặt hàng màu xanh
9) Đưa ra {sid} của các hãng không cung ứng mặt hàng nào

72

72

73

73

## Nội dung chương:

1. Khái niệm cơ bản của mô hình quan hệ
2. Chuyển lược đồ ER sang mô hình quan hệ
3. Ngôn ngữ định nghĩa và thao tác dữ liệu đối với mô hình quan hệ
   - 3.1 Ngôn ngữ Đại số quan hệ
   - 3.2 Ngôn ngữ QBE
   - 3.3 Ngôn ngữ truy vấn hướng cấu trúc SQL

74

# Ngôn ngữ QBE





## QBE (*Query-By-Example*)

- Là một ngôn ngữ truy vấn dữ liệu
- Các câu truy vấn được thiết lập bởi một giao diện đồ hoạ
- Phù hợp với các câu truy vấn đơn giản, tham chiếu đến ít bảng
- Một số sản phẩm: IBM™ (IBM Query Management Facility), Paradox, MS. Access, ...

## Truy vấn trên một quan hệ

- P.~ Print

| Student | ID | Name | Suburb |
|---|---|---|---|
| | | P._x | Bundoora |

- Biểu thức đại số quan hệ tương đương

$$\sigma_{suburb='Bundoora'}(Student)$$

77

77

## Truy vấn trên một quan hệ (tiếp)

- Lựa chọn tất cả các cột

| Student | ID | Name | Suburb |
|---|---|---|---|
| P. | | | Bundoora |

- Sắp xếp

| Student | ID | Name | Suburb |
|---|---|---|---|
| | | P.AO(1) | P.AO(2) |

- AO: sắp xếp tăng dần
- DO: sắp xếp giảm dần

78

78

# Truy vấn trên nhiều quan hệ

- Đưa ra tên của các sinh viên có đăng ký ít nhất một khoá học

| Student | ID | Name | Suburb |
|---|---|---|---|
| | _id | P._name | |

| Enrol | SID | Course |
|---|---|---|
| | _id | |

- Đưa ra tên các sinh viên không đăng ký một khoá học nào

| Student | ID | Name | Suburb |
|---|---|---|---|
| | _id | P._name | |

| Enrol | SID | Course |
|---|---|---|
| ¬ | _id | |

79

79

# Các tính toán tập hợp

- Các phép toán: AVG, COUNT, MAX, MIN, SUM
- Ví dụ: đưa ra tên các thành phố và số lượng sinh viên đến từ thành phố đó

| Student | ID | Name | Suburb | |
|---|---|---|---|---|
| | _id | | G.P. | P.COUNT._id |

- G. ~ Grouping

80

80

# Hộp điều kiện

- Được sử dụng để biểu diễn
  - Điều kiện trên nhiều hơn 1 thuộc tính
  - Điều kiện trên các trường tính toán tập hợp
- Ví dụ: đưa ra danh sách các thành phố có nhiều hơn 5 sinh viên

| Student | ID | Name | Suburb | Condition |
|---|---|---|---|---|
| | _id | | G.P. | COUNT._id > 5 |

81

81

# Các thao tác thay đổi dữ liệu

- Xóa

| Student | ID | Name | Suburb |
|---|---|---|---|
| D. | 1108 | | |

- Thêm

| Student | ID | Name | Suburb |
|---|---|---|---|
| I. | 1179 | David | Evry |

- Sửa

| Student | ID | Name | Suburb |
|---|---|---|---|
| | 1179 | | U.Paris |

82

82

# Tính đầy đủ của QBE

- Có thể biểu diễn cả 5 phép toán đại số cơ sở ($\sigma$,$\Pi$,$\cup$,\,x)

83

83

# Định nghĩa dữ liệu trong QBE

- Sử dụng cùng qui cách và giao diện đồ họa như đối với truy vấn.

| I.Student | I. | ID | Name | Suburb |
|---|---|---|---|---|
| KEY | I. | Y | N | N |
| TYPE | I. | CHAR(5) | CHAR(30) | CHAR(30) |
| DOMAIN | I. | Sid | SName | Surb |
| INVERSION | I. | Y | N | N |

84

84

## Nội dung chương:

1. Khái niệm cơ bản của mô hình quan hệ
2. Chuyển lược đồ ER sang mô hình quan hệ
3. Ngôn ngữ định nghĩa và thao tác dữ liệu đối với mô hình quan hệ
   - 3.1 Ngôn ngữ Đại số quan hệ
   - 3.2 Ngôn ngữ QBE
   - 3.3 Ngôn ngữ truy vấn hướng cấu trúc SQL

85

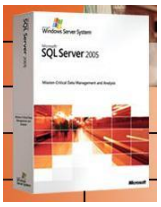
# Ngôn ngữ SQL

86

86

# SQL (Structured Query Language)

- 1975: SEQUEL
  - System-R
- 1976: SEQUEL2
- 1978/79: SQL
  - System-R
- 1986: chuẩn SQL-86
- 1989: chuẩn SQL-89
- 1992: chuẩn SQL-92
- 1996: chuẩn SQL-96




87

87

# Các thành phần của SQL

- Ngôn ngữ định nghĩa dữ liệu (***D**ata **D**efinition **L**anguage*)
  - Cấu trúc các bảng CSDL
  - Các mối liên hệ của dữ liệu
  - Quy tắc, ràng buộc áp đặt lên dữ liệu
- Ngôn ngữ thao tác dữ liệu (***D**ata **M**anipulation **L**anguage*)
  - Thêm, xoá, sửa dữ liệu trong CSDL
  - Truy vấn dữ liệu
- Ngôn ngữ điều khiển dữ liệu (***D**ata **C**ontrol **L**anguage*)
  - Khai báo bảo mật thông tin
  - Quyền hạn của người dùng trong khai thác CSDL

88

88

# Ngôn ngữ định nghĩa dữ liệu

- Các thông tin được định nghĩa bao gồm
  - Sơ đồ quan hệ
  - Kiểu dữ liệu hay miền giá trị của mỗi thuộc tính
  - Các ràng buộc toàn vẹn
  - Các chỉ số đối với mỗi bảng
  - Thông tin an toàn và ủy quyền đối với mỗi bảng
  - Cấu trúc lưu trữ vật lý của mỗi bảng trên đĩa
- ➢Được biểu diễn bởi các lệnh định nghĩa dữ liệu

89

89

# Quy ước đặt tên và kiểu dữ liệu

- Quy ước đặt tên
  - 32 ký tự: chữ cái, số, dấu _
- Kiểu dữ liệu (SQL-92)
  - char(n)
  - varchar(n)
  - int
  - smallint
  - numeric(p,d)
  - real, double
  - float(n)
  - date
  - time

90

90

# Quy ước đặt tên và kiểu dữ liệu

## Microsoft Access Data Types

| Data type | Description | Storage |
|---|---|---|
| Text | Use for text or combinations of text and numbers. 255 characters maximum | |
| Memo | Memo is used for larger amounts of text. Stores up to 65,536 characters. **Note:** You cannot sort a memo field. However, they are searchable | |
| Byte | Allows whole numbers from 0 to 255 | 1 byte |
| Integer | Allows whole numbers between -32,768 and 32,767 | 2 bytes |
| Long | Allows whole numbers between -2,147,483,648 and 2,147,483,647 | 4 bytes |
| Single | Single precision floating-point. Will handle most decimals | 4 bytes |
| Double | Double precision floating-point. Will handle most decimals | 8 bytes |
| Currency | Use for currency. Holds up to 15 digits of whole dollars, plus 4 decimal places. **Tip:** You can choose which country's currency to use | 8 bytes |
| AutoNumber | AutoNumber fields automatically give each record its own number, usually starting at 1 | 4 bytes |
| Date/Time | Use for dates and times | 8 bytes |
| Yes/No | A logical field can be displayed as Yes/No, True/False, or On/Off. In code, use the constants True and False (equivalent to -1 and 0). **Note:** Null values are not allowed in Yes/No fields | 1 bit |
| Ole Object | Can store pictures, audio, video, or other BLOBs (Binary Large OBjects) | up to 1GB |
| Hyperlink | Contain links to other files, including web pages | |
| Lookup Wizard | Let you type a list of options, which can then be chosen from a drop-down list | 4 bytes |

91

# Quy ước đặt tên và kiểu dữ liệu

## MySQL Data Types

**Text types:**

| Data type | Description |
|---|---|
| CHAR(size) | Holds a fixed length string (can contain letters, numbers, and special characters). The fixed size is specified in parenthesis. Can store up to 255 characters |
| VARCHAR(size) | Holds a variable length string (can contain letters, numbers, and special characters). The maximum size is specified in parenthesis. Can store up to 255 characters. **Note:** If you put a greater value than 255 it will be converted to a TEXT type |
| TINYTEXT | Holds a string with a maximum length of 255 characters |
| TEXT | Holds a string with a maximum length of 65,535 characters |
| BLOB | For BLOBs (Binary Large OBjects). Holds up to 65,535 bytes of data |
| MEDIUMTEXT | Holds a string with a maximum length of 16,777,215 characters |
| MEDIUMBLOB | For BLOBs (Binary Large OBjects). Holds up to 16,777,215 bytes of data |
| LONGTEXT | Holds a string with a maximum length of 4,294,967,295 characters |
| LONGBLOB | For BLOBs (Binary Large OBjects). Holds up to 4,294,967,295 bytes of data |
| ENUM(x,y,z,etc.) | Let you enter a list of possible values. You can list up to 65535 values in an ENUM list. If a value is inserted that is not in the list, a blank value will be inserted.<br>**Note:** The values are sorted in the order you enter them.<br>You enter the possible values in this format: ENUM('X','Y','Z') |
| SET | Similar to ENUM except that SET may contain up to 64 list items and can store more than one choice |

92

# Quy ước đặt tên và kiểu dữ liệu

## MySQL Data Types

**Number types:**

| Data type | Description |
|---|---|
| TINYINT(size) | -128 to 127 normal. 0 to 255 UNSIGNED*. The maximum number of digits may be specified in parenthesis |
| SMALLINT(size) | -32768 to 32767 normal. 0 to 65535 UNSIGNED*. The maximum number of digits may be specified in parenthesis |
| MEDIUMINT(size) | -8388608 to 8388607 normal. 0 to 16777215 UNSIGNED*. The maximum number of digits may be specified in parenthesis |
| INT(size) | -2147483648 to 2147483647 normal. 0 to 4294967295 UNSIGNED*. The maximum number of digits may be specified in parenthesis |
| BIGINT(size) | -9223372036854775808 to 9223372036854775807 normal. 0 to 18446744073709551615 UNSIGNED*. The maximum number of digits may be specified in parenthesis |
| FLOAT(size,d) | A small number with a floating decimal point. The maximum number of digits may be specified in the size parameter. The maximum number of digits to the right of the decimal point is specified in the d parameter |
| DOUBLE(size,d) | A large number with a floating decimal point. The maximum number of digits may be specified in the size parameter. The maximum number of digits to the right of the decimal point is specified in the d parameter |
| DECIMAL(size,d) | A DOUBLE stored as a string , allowing for a fixed decimal point. The maximum number of digits may be specified in the size parameter. The maximum number of digits to the right of the decimal point is specified in the d parameter |

93

# Quy ước đặt tên và kiểu dữ liệu

## MySQL Data Types

**Date types:**

| Data type | Description |
|---|---|
| DATE() | A date. Format: YYYY-MM-DD<br><br>**Note:** The supported range is from '1000-01-01' to '9999-12-31' |
| DATETIME() | *A date and time combination. Format: YYYY-MM-DD HH:MM:SS<br><br>**Note:** The supported range is from '1000-01-01 00:00:00' to '9999-12-31 23:59:59' |
| TIMESTAMP() | *A timestamp. TIMESTAMP values are stored as the number of seconds since the Unix epoch ('1970-01-01 00:00:00' UTC). Format: YYYY-MM-DD HH:MM:SS<br><br>**Note:** The supported range is from '1970-01-01 00:00:01' UTC to '2038-01-09 03:14:07' UTC |
| TIME() | A time. Format: HH:MM:SS<br><br>**Note:** The supported range is from '-838:59:59' to '838:59:59' |
| YEAR() | A year in two-digit or four-digit format.<br><br>**Note:** Values allowed in four-digit format: 1901 to 2155. Values allowed in two-digit format: 70 to 69, representing years from 1970 to 2069 |

94

# Quy ước đặt tên và kiểu dữ liệu

## SQL Server Data Types

**Character strings:**

| Data type | Description | Storage |
|---|---|---|
| char(n) | Fixed-length character string. Maximum 8,000 characters | n |
| varchar(n) | Variable-length character string. Maximum 8,000 characters | |
| varchar(max) | Variable-length character string. Maximum 1,073,741,824 characters | |
| text | Variable-length character string. Maximum 2GB of text data | |

**Unicode strings:**

| Data type | Description | Storage |
|---|---|---|
| nchar(n) | Fixed-length Unicode data. Maximum 4,000 characters | |
| nvarchar(n) | Variable-length Unicode data. Maximum 4,000 characters | |
| nvarchar(max) | Variable-length Unicode data. Maximum 536,870,912 characters | |
| ntext | Variable-length Unicode data. Maximum 2GB of text data | |

**Binary types:**

| Data type | Description | Storage |
|---|---|---|
| bit | Allows 0, 1, or NULL | |
| binary(n) | Fixed-length binary data. Maximum 8,000 bytes | |
| varbinary(n) | Variable-length binary data. Maximum 8,000 bytes | |
| varbinary(max) | Variable-length binary data. Maximum 2GB | |
| image | Variable-length binary data. Maximum 2GB | |

95

# Quy ước đặt tên và kiểu dữ liệu

**Number types:**

| Data type | Description | Storage |
|---|---|---|
| tinyint | Allows whole numbers from 0 to 255 | 1 byte |
| smallint | Allows whole numbers between -32,768 and 32,767 | 2 bytes |
| int | Allows whole numbers between -2,147,483,648 and 2,147,483,647 | 4 bytes |
| bigint | Allows whole numbers between -9,223,372,036,854,775,808 and 9,223,372,036,854,775,807 | 8 bytes |
| decimal(p,s) | Fixed precision and scale numbers.<br>Allows numbers from -10^38 +1 to 10^38 –1.<br>The p parameter indicates the maximum total number of digits that can be stored (both to the left and to the right of the decimal point). p must be a value from 1 to 38. Default is 18.<br>The s parameter indicates the maximum number of digits stored to the right of the decimal point. s must be a value from 0 to p. Default value is 0 | 5-17 bytes |
| numeric(p,s) | Fixed precision and scale numbers.<br>Allows numbers from -10^38 +1 to 10^38 –1.<br>The p parameter indicates the maximum total number of digits that can be stored (both to the left and to the right of the decimal point). p must be a value from 1 to 38. Default is 18.<br>The s parameter indicates the maximum number of digits stored to the right of the decimal point. s must be a value from 0 to p. Default value is 0 | 5-17 bytes |
| smallmoney | Monetary data from -214,748.3648 to 214,748.3647 | 4 bytes |
| money | Monetary data from -922,337,203,685,477.5808 to 922,337,203,685,477.5807 | 8 bytes |
| float(n) | Floating precision number data from -1.79E + 308 to 1.79E + 308.<br>The n parameter indicates whether the field should hold 4 or 8 bytes. float(24) holds a 4-byte field and float(53) holds an 8-byte field. Default value of n is 53. | 4 or 8 bytes |
| real | Floating precision number data from -3.40E + 38 to 3.40E + 38 | 4 bytes |

96

## Quy ước đặt tên và kiểu dữ liệu

**Date types:**

| Data type | Description | Storage |
|---|---|---|
| datetime | From January 1, 1753 to December 31, 9999 with an accuracy of 3.33 milliseconds | 8 bytes |
| datetime2 | From January 1, 0001 to December 31, 9999 with an accuracy of 100 nanoseconds | 6-8 bytes |
| smalldatetime | From January 1, 1900 to June 6, 2079 with an accuracy of 1 minute | 4 bytes |
| date | Store a date only. From January 1, 0001 to December 31, 9999 | 3 bytes |
| time | Store a time only to an accuracy of 100 nanoseconds | 3-5 bytes |
| datetimeoffset | The same as datetime2 with the addition of a time zone offset | 8-10 bytes |
| timestamp | Stores a unique number that gets updated every time a row gets created or modified. The timestamp value is based upon an internal clock and does not correspond to real time. Each table may have only one timestamp variable | |

**Other data types:**

| Data type | Description |
|---|---|
| sql_variant | Stores up to 8,000 bytes of data of various data types, except text, ntext, and timestamp |
| uniqueidentifier | Stores a globally unique identifier (GUID) |
| xml | Stores XML formatted data. Maximum 2GB |
| cursor | Stores a reference to a cursor used for database operations |
| table | Stores a result-set for later processing |

97

## Cú pháp

- Tạo bảng

**CREATE TABLE** tên-bảng(
    cột-1 kiểu-dữ-liệu-1 [**NOT NULL**], ...,
    cột-2 kiểu-dữ-liệu-2 [**NOT NULL**], ...,
        .....
    [**CONSTRAINT** tên-ràng-buộc kiểu-ràng-buộc]
    ...
);

- Xoá bảng

**DROP TABLE** tên-bảng

98

98

# Tạo bảng - Ví dụ:

```
CREATE TABLE Supplier(
        sid char(4) NOT NULL,
        sname varchar(30) NOT NULL,
        size smallint,
        city varchar(20),
  CONSTRAINT KhoachinhS primary key(sid)
);
```

99

99

# Tạo bảng - Ví dụ (tiếp)

```
CREATE TABLE Product(
    pid char(4) NOT NULL,
    pname varchar(30) NOT NULL,
    colour char(8),
    weight int,
    city varchar(20),
CONSTRAINT KhoachinhP primary key(pid)
);
```

100

100

# Tạo bảng - Ví dụ (tiếp)

```
CREATE TABLE SupplyProduct(
        sid char(4) NOT NULL,
        pid char(4) NOT NULL,
        quantity smallint,
        primary key(sid,pid),
  foreign key(sid) references Supplier(sid),
  foreign key(pid) references Product(pid),
  check(quantity >0)
  );
```

101

101

# Kiểu ràng buộc

- Ràng buộc toàn vẹn (RBTV) về giá trị miền

  **CONSTRAINT** <tên ràng buộc>

  **CHECK** <điều kiện>

- RBTV về khoá ngoại hay phụ thuộc tồn tại

  **CONSTRAINT** <tên ràng buộc>

  **FOREIGN KEY** ($fk_i$) **REFERENCES** tên-bảng($k_i$);

102

102

## Thêm/xoá/sửa cột của các bảng

- Thêm

  **ALTER TABLE** <tên bảng>
  **ADD COLUMN** <tên cột> <kiểu dữ liệu> [NOT NULL];

- Xoá

  **ALTER TABLE** <tên bảng>
  **DROP COLUMN** <tên cột>;

- Sửa

  **ALTER TABLE** <tên bảng>
  **CHANGE COLUMN** <tên cột> **TO** <kiểu dữ liệu mới>;

103

103

## Ví dụ:

- ALTER TABLE SupplyProduct ADD COLUMN price real NOT NULL;
- ALTER TABLE SupplyProduct DROP COLUMN price;
- ALTER TABLE Supplier CHANGE COLUMN sname TO varchar(20);

104

104

## Thêm/xóa các ràng buộc

- Thêm

  **ALTER TABLE** <tên bảng>
  **ADD CONSTRAINT** <tên ràng buộc> <kiểu ràng buộc>

- Xóa

  **ALTER TABLE** <tên bảng>
  **DROP CONSTRAINT** <tên ràng buộc>

105

105

## Ngôn ngữ truy vấn dữ liệu

- Cú pháp câu lệnh SQL:

**SELECT** [**DISTINCT**] <DS cột>|*|<Biểu thức>|<Hàm TV>

**FROM** <DS bảng>

[**WHERE** <Điều kiện tìm kiếm>]

[**GROUP BY** <DS cột> [**HAVING** <Điều kiện>]]

[**ORDER BY** <Danh sách cột> [**ASC** | **DESC**]]

[**UNION** |**INTERSECT**| **MINUS** <Câu truy vấn khác>]

106

106

## Truy vấn không điều kiện trên một bảng

- Tìm thông tin từ các cột của bảng
  - **SELECT** <DS cột>
    **FROM** <Tên bảng>;
  - **SELECT** *
    **FROM** <Tên bảng>;
- Ví dụ

  **SELECT** Name **FROM** Student;

Student

| Id | Name | Suburb |
|---|---|---|
| 1108 | Robert | Kew |
| 3936 | Glen | Bundoora |
| 8507 | Robert | Bundoora |
| 8452 | Mary | Balwyn |

$$\prod_{Name}(Student)$$

| Name |
|---|
| Robert |
| Glen |
| Mary |

107

107

## Truy vấn không điều kiện trên một bảng

**Một số ví dụ khác:**

- Đưa ra tên của các mặt hàng
  SELECT pname FROM Product;
- Đưa ra tên khác nhau của các mặt hàng
  SELECT DISTINCT pname
  FROM Product;
- Đưa ra toàn bộ thông tin về các hãng cung ứng
  SELECT * FROM Supplier;
- Đưa ra mã số hãng cung ứng, mã mặt hàng được cung ứng và 10 lần số lượng mặt hàng đã được cung ứng
  SELECT sid, pid, quantity*10
  FROM SupplyProduct;

108

108

# Truy vấn có điều kiện trên 1 bảng

- Chọn các bản ghi (dòng)

```
SELECT  <DS cột>
FROM    <Tên bảng>
WHERE   <Điều kiện tìm kiếm>
```

- Ví dụ

```
SELECT  *
FROM    Student
WHERE   Suburb='Bundoora' ;
```

Student

| Id | Name | Suburb |
|---|---|---|
| 1108 | Robert | Kew |
| 3936 | Glen | Bundoora |
| 8507 | Robert | Bundoora |
| 8452 | Mary | Balwyn |

$$\sigma_{Suburb='Bundoora'}(Student)$$

| Id | Name | Suburb |
|---|---|---|
| 3936 | Glen | Bundoora |
| 8507 | Robert | Bundoora |

109

109

# Truy vấn có điều kiện trên 1 bảng

**Một số ví dụ khác:**

- Đưa ra tên của các hãng cung ứng có trụ sở tại London

```
SELECT sname FROM Supplier
WHERE city = 'London';
```

- Đưa ra mã số và tên của các hãng cung ứng nằm ở London và có số nhân viên lớn hơn 75

```
SELECT sid, sname FROM Supplier
WHERE city = 'London' AND size > 75;
```

110

110

# Biểu diễn điều kiện lựa chọn

- Các phép toán quan hệ: =, !=, <, >, <=, >=
- Các phép toán logic: NOT, AND, OR
- Phép toán phạm vi: BETWEEN, IN, LIKE
  - Kiểu dữ liệu số
    - attr **BETWEEN** val1 **AND** val2 (⇔ (attr>=val1) and (attr<=val2) )
    - attr **IN** (val1, val2, ...) (⇔ (attr=val1) or (attr=val2) or ... )
  - Kiểu dữ liệu xâu
    - **LIKE**: sử dụng đối sánh mẫu xâu với các ký tự thay thế cho 1 ký tự bất kỳ (_, ?), thay thế cho 1 xâu ký tự bất kỳ (*, %)

    (PostGreSQL sử dụng dấu % và dấu _ )

111

111

# Biểu diễn điều kiện lựa chọn - Ví dụ:

- Đưa ra thông tin của các hãng cung ứng có số nhân viên trong khoảng từ 100 đến 150

```
SELECT * FROM Supplier
WHERE size BETWEEN 100 AND 150;
```

- Đưa ra mã số của hãng cung ứng mặt hàng P1 hoặc P2
  - Cách 1:

```
SELECT sid FROM SupplyProduct
WHERE pid = 'P1' OR pid = 'P2';
```

  - Cách 2:

```
SELECT sid FROM SupplyProduct
WHERE pid IN ('P1', 'P2');
```

112

112

## Biểu diễn điều kiện lựa chọn - Ví dụ (tiếp)

- Đưa ra thông tin của hãng sản xuất có trụ sở đặt tại thành phố bắt đầu bằng chữ New

```
SELECT * FROM SUPPLIER
WHERE city LIKE 'New%';
```

New York, New Jersey, New Mexico, New Hampshire

113

113

## Loại trừ các bản ghi trùng nhau

- Từ khoá **DISTINCT**

```
SELECT DISTINCT  <DS cột>
FROM  <DS bảng>
```

- Ví dụ: đưa ra danh sách tên các khoa (Dept) tương ứng với các khoá học (Course). Mỗi giá trị chỉ hiện thị một lần

```
SELECT DISTINCT  Dept
FROM  Course
```

114

114

## Truy vấn có sử dụng phép toán đổi tên

- SQL cho phép đổi tên các bảng và các cột trong một câu truy vấn (sau mệnh đề SELECT và FROM) sử dụng cấu trúc:

<tên cũ> AS <tên mới>

- Đưa ra tên và số nhân viên của các hãng cung ứng ở Paris

```
SELECT sname AS HangOParis, size AS SoNhanVien
FROM Supplier
WHERE city = 'Paris';
SELECT      SID , Stud.Name as SName,
            Sub.Name as Subject
FROM        Student as Stud,Takes,
        Subject as Sub
WHERE       (Id=SID) and (SNO = No)
```

115

115

## Truy vấn phức tạp trên nhiều bảng

- Điều kiện kết nối

```
SELECT    <DS cột>
FROM      <DS bảng>
WHERE     <Điều kiện tìm kiếm>
```

- Ví dụ: đưa ra danh sách mã sinh viên (Id), tên sinh viên (Name), thành phố (Suburb), mã khoá học (Course) mà các sinh viên đã đăng ký

```
SELECT    Id, Name, Suburb, Course
FROM      Student, Enrol
WHERE     Id=SID;
```

116

116

# Kết nối tự nhiên

SELECT <DS cột>
FROM A, B, C
WHERE A.CộtX = B.CộtX AND B.CộtY = C.CộtY

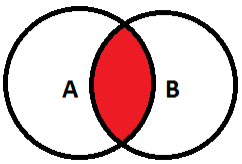
SELECT <DS cột>
FROM A NATURAL JOIN B NATURAL JOIN C

117

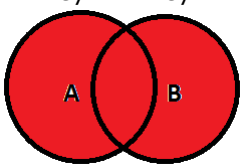
117

# Kết nối hai bảng

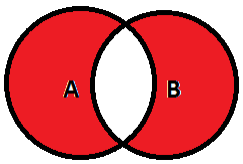
**Kết nối trong**
SELECT <DS cột>
FROM A INNER JOIN B
ON A.Key = B.Key




**Kết nối OUTER JOIN**
SELECT <DS cột>
FROM A FULL OUTER JOIN B
ON A.Key = B.Key




**Kết nối OUTER JOIN**
SELECT <DS cột>
FROM A FULL OUTER JOIN B
ON A.Key = B.Key
WHERE A.Key IS NULL OR
B.Key IS NULL

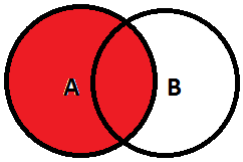



118

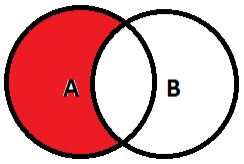
118

# Kết nối hai bảng

**Kết nối ngoài trái**
SELECT <DS cột>
FROM A LEFT JOIN B
ON A.Key = B.Key




**Kết nối ngoài trái**
SELECT <DS cột>
FROM A LEFT JOIN B
ON A.Key = B.Key
WHERE B.Key IS NULL

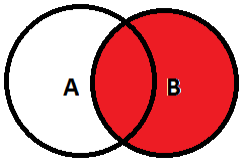



119

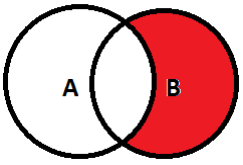
119

# Kết nối hai bảng

**Kết nối ngoài phải**
SELECT <DS cột>
FROM A RIGHT JOIN B
ON A.Key = B.Key




**Kết nối ngoài phải**
SELECT <DS cột>
FROM A RIGHT JOIN B
ON A.Key = B.Key
WHERE A.Key IS NULL




120

120

## Truy vấn phức tạp trên nhiều bảng

- Đưa ra tên của hãng có cung ứng mặt hàng P1

```
SELECT sname
FROM Supplier S, SupplyProduct SP
WHERE S.sid = SP.sid AND SP.pid = 'P1';
```

- Đưa ra tên và mã số của hãng cung ứng ít nhất một mặt hàng màu đỏ

```
SELECT sname, S.sid
FROM Supplier S, SupplyProduct SP, Product P
WHERE S.sid = SP.sid AND P.pid = SP.pid AND
  P.colour = 'red';
```

121

121

## Tìm kiếm có sắp xếp

- Sắp xếp các bản ghi kết quả theo một thứ tự cho trước

```
SELECT      <DS cột>
FROM  <DS bảng>
[WHERE      <Điều kiện tìm kiếm>]
ORDER BY  <DS cột> [ASC | DESC]
```

- Ví dụ: đưa ra danh sách tên các sinh viên theo thứ tự tăng dần

```
SELECT      Name
FROM  Student
ORDER BY Name ASC
```

122

122

# Phân nhóm các bản ghi kết quả

- Phân nhóm các bản ghi kết quả theo giá trị của 1 hoặc nhitroều thuộc tính

```
SELECT      <DS cột>
FROM        <DS bảng>
[WHERE      <Điều kiện tìm kiếm>]
[GROUP BY   <DS cột>]
```

- Cột được chỉ ra ng mệnh đề GroupBy được sử dụng làm cơ sở để chia nhóm. Cột này cũng bắt buộc phải được chỉ ra trong mệnh đề Select
- Ví dụ đưa ra tên các sinh viên nhóm theo thành phố của sinh viên đó

```
SELECT  Suburb, Name
FROM    Student
GROUP BY  Suburb
```

```
SELECT  Suburb, Count(Id)
FROM    Student
GROUP BY  Suburb
```





# Phân nhóm các bản ghi kết quả

Orders

| O_Id | OrderDate | OrderPrice | Customer |
|---|---|---|---|
| 1 | 2008/11/12 | 1000 | Hansen |
| 2 | 2008/10/23 | 1600 | Nilsen |
| 3 | 2008/09/02 | 700 | Hansen |
| 4 | 2008/09/03 | 300 | Hansen |
| 5 | 2008/08/30 | 2000 | Jensen |
| 6 | 2008/10/04 | 100 | Nilsen |

```
SELECT Customer,SUM(OrderPrice) FROM Orders
```

The result-set will look like this:

| Customer | SUM(OrderPrice) |
|---|---|
| Hansen | 5700 |
| Nilsen | 5700 |
| Hansen | 5700 |
| Hansen | 5700 |
| Jensen | 5700 |
| Nilsen | 5700 |

```
SELECT Customer,SUM(OrderPrice) FROM Orders
GROUP BY Customer
```

The result-set will look like this:

| Customer | SUM(OrderPrice) |
|---|---|
| Hansen | 2000 |
| Nilsen | 1700 |
| Jensen | 2000 |

## Điều kiện hiển thị các bản ghi kết quả

- Lựa chọn các bản ghi kết quả để hiển thị

```
SELECT        <DS cột>
FROM          <DS bảng>
[WHERE        <Điều kiện tìm kiếm>]
GROUP BY <Ds cột> HAVING <Điều kiện>
```

- Ví dụ: đưa ra tên các thành phố có nhiều hơn 3 sinh viên

```
SELECT    Suburb, COUNT(ID)
FROM  Student
GROUP BY Suburb
HAVING    COUNT(ID) > 3
```

125

125

## Điều kiện hiển thị các bản ghi kết quả

*Ví dụ :* Điều kiện trên nhóm HAVING

Bảng Orders

Tìm và đưa ra tất cả khách hàng có tổng giá trị các hóa đơn <2000

```
SELECT Customer,SUM(OrderPrice) FROM Orders
GROUP BY Customer
HAVING SUM(OrderPrice)<2000
```

The result-set will look like this:

| Customer | SUM(OrderPrice) |
|---|---|
| Nilsen | 1700 |

Tìm khách hàng có tên “Hansen” hoặc “Jensen có tổng giá trị các hóa đơn

```
SELECT Customer,SUM(OrderPrice) FROM Orders
WHERE Customer='Hansen' OR Customer='Jensen'
GROUP BY Customer
HAVING SUM(OrderPrice)>1500
```

→

| Customer | SUM(OrderPrice) |
|---|---|
| Hansen | 2000 |
| Jensen | 2000 |

126

## Các phép toán tập hợp:

- SQL có cài đặt các phép toán tập hợp
  - Hợp (UNION)
  - Giao (INTERSECT)
  - Trừ (EXCEPT)
- Kết quả trả về là tập hợp
  - Loại bỏ các bộ trùng nhau
  - Để giữ lại các bộ trùng nhau phải sử dụng từ khóa
    - UNION ALL
    - INTERSECT ALL
    - EXCEPT ALL

127

## Các phép toán tập hợp:

Ví dụ UNION

"Employees_Norway":

| E_ID | E_Name |
|---|---|
| 01 | Hansen, Ola |
| 02 | Svendson, Tove |
| 03 | Svendson, Stephen |
| 04 | Pettersen, Kari |

"Employees_USA":

| E_ID | E_Name |
|---|---|
| 01 | Turner, Sally |
| 02 | Kent, Clark |
| 03 | Svendson, Stephen |
| 04 | Scott, Stephen |

```
SELECT E_Name FROM Employees_Norway
UNION
SELECT E_Name FROM Employees_USA
```

| E_Name |
|---|
| Hansen, Ola |
| Svendson, Tove |
| Svendson, Stephen |
| Pettersen, Kari |
| Turner, Sally |
| Kent, Clark |
| Scott, Stephen |

128

## Các phép toán tập hợp:

Ví dụ: UNION ALL

"Employees_Norway":

| E_ID | E_Name |
|---|---|
| 01 | Hansen, Ola |
| 02 | Svendson, Tove |
| 03 | Svendson, Stephen |
| 04 | Pettersen, Kari |

"Employees_USA":

| E_ID | E_Name |
|---|---|
| 01 | Turner, Sally |
| 02 | Kent, Clark |
| 03 | Svendson, Stephen |
| 04 | Scott, Stephen |

```
SELECT E_Name FROM Employees_Norway
UNION ALL
SELECT E_Name FROM Employees_USA
```

↓

| E_Name |
|---|
| Hansen, Ola |
| Svendson, Tove |
| Svendson, Stephen |
| Pettersen, Kari |
| Turner, Sally |
| Kent, Clark |
| Svendson, Stephen |
| Scott, Stephen |

129

## Các phép toán tập hợp:

- Ví dụ: đưa ra danh sách tên các môn học không có sinh viên nào tham dự

```
    SELECT DISTINCT Subject.Name
    FROM Subject
MINUS
    SELECT DISTINCT Subject.Name
    FROM Student, Takes, Subject
    WHERE Student.Id = Takes.SID and Takes.SNO = Subject.No
```

- Tìm sid của hãng cung ứng đồng thời 2 mặt hàng P1 và P2

```
  SELECT sid FROM SupplyProduct WHERE pid = 'P1'
INTERSECT
  SELECT sid FROM SupplyProduct WHERE pid = 'P2'
```

- Tìm mã số của hãng không cung ứng mặt hàng nào

```
  SELECT sid FROM Supplier
MINUS
  SELECT sid FROM SupplyProduct
```

130

130

## Các câu truy vấn lồng nhau

- Là trường hợp các câu truy vấn (con) được viết lồng nhau
- Thường được sử dụng để
  - Kiểm tra thành viên tập hợp (**IN, NOT IN**)
  - So sánh tập hợp (**>ALL, >=ALL, <ALL,<=ALL,=ALL, NOT IN,SOME,** )

    Ví dụ :SELECT *
    FROM Supplier
    WHERE SIZE>=ALL(SELECT SIZE FROM Supplier);

  - Kiểm tra các bảng rỗng (**EXISTS** hoặc **NOT EXISTS**)
- Các truy vấn con lồng nhau thông qua mệnh đề **WHERE**

131

131

## Các câu truy vấn lồng nhau

❖Truy vấn lồng

❖Cú pháp :

```
SELECT <danh sách các cột>
FROM <danh sách các bảng>
WHERE < điều kiện so sánh> (SELECT....
                            FROM ....
                            WHERE... )
```

T/vấn ngoài

Truy vấn con

❖Các câu lệnh SELECT có thể lồng nhau ở nhiều mức

❖Câu truy vấn con thường trả về một tập các giá trị

❖Kết quả của câu truy vấn con sẽ làm điều kiện cho câu truy vấn ngoài, thứ tự thực thi từ trong ra ngoài

132

## Các câu truy vấn lồng nhau (tiếp)

- Kiểm tra thành viên tập hợp với IN và NOT IN:
  - Đưa ra mã số của các hãng cung ứng đồng thời 2 mặt hàng P1 và P2:

```
SELECT DISTINCT sid FROM SupplyProduct
WHERE pid = 'P1' AND
        sid IN (SELECT sid FROM SupplyProduct SP2 WHERE
  SP2.pid = 'P2');
```

  - Đưa ra sid của các hãng không cung ứng mặt hàng P3:

```
SELECT sid FROM SupplyProduct
WHERE sid NOT IN (SELECT sid From SupplyProduct SP2 WHERE
  SP2.pid = 'P3');
```

133

133

## Các câu truy vấn lồng nhau (tiếp)

- **So sánh tập hợp:** Sử dụng các phép toán <,>, >=,<=,=,!= (<>) kèm với các mệnh đề ANY và ALL
  - Đưa ra tên của các hãng có số nhân viên đông nhất:

```
SELECT sname FROM Supplier
WHERE size >= ALL(SELECT size FROM Supplier)
```

  - Đưa ra sid của hãng cung ứng một mặt hàng với số lượng bằng ít nhất 1 trong số lượng các mặt hàng được cung ứng bởi S2

```
SELECT sid FROM SupplyProduct
WHERE sid != 'S2' AND quantity = ANY(SELECT quantity
  FROM SupplyProduct SP2 WHERE SP2.sid = 'S2');
```

134

134

## Các câu truy vấn lồng nhau (tiếp)

- Kiểm tra tập hợp rỗng với **EXISTS** và **NOT EXISTS**
  - **EXISTS** (câu truy vấn con): nhận giá trị đúng khi câu truy vấn con cho ra kết quả là một quan hệ khác rỗng
  - **NOT EXISTS** (câu truy vấn con): nhận giá trị đúng khi câu truy vấn con cho ra kết quả là một quan hệ rỗng

135

135

## Các câu truy vấn lồng nhau (tiếp)

- Đưa ra thông tin của các nhà cung cấp đã cung ứng ít nhất một mặt hàng

```
SELECT * FROM Supplier S
WHERE EXISTS (SELECT sid FROM SupplyProduct SP
              WHERE S.sid = SP.sid);
```

- Đưa ra thông tin của các nhà cung cấp không cung ứng mặt hàng nào

```
SELECT * FROM Supplier S
WHERE NOT EXISTS (SELECT * FROM SupplyProduct SP
                  WHERE S.sid = SP.sid);
```

136

136

# Các hàm thư viện

- Hàm tính toán trên nhóm các bản ghi

❖COUNT()

- ❖ COUNT(*) đếm số dòng
- ❖ COUNT(<tên thuộc tính>) đếm số giá trị khác NULL của thuộc tính
- ❖ COUNT(DISTINCT <tên thuộc tính>) đếm số giá trị khác nhau và khác NULL của thuộc tính

❖MIN(<tên cột>) trả về giá trị cực tiểu chứa trong cột

❖MAX(<tên cột>) trả về giá trị cực đại chứa trong cột

❖SUM(<tên cột>) tổng giá trị trong cột

❖AVG(<tên cột>) giá trị trung bình của cột

137

137

# Các hàm thư viện (tiếp)

- Hàm tính toán trên bản ghi
  - Hàm toán học: ABS, SQRT, LOG, EXP, SIGN, ROUND
  - Hàm xử lý xâu ký tự: LEN, LEFT, RIGHT, MID
  - Hàm xử lý thời gian: DATE, DAY, MONTH, YEAR, HOUR, MINUTE, SECOND, DATEPART
  - Hàm chuyển đổi kiểu giá trị: FORMAT

138

138

## Một số ví dụ với các hàm thư viện

- Có bao nhiêu mặt hàng khác nhau được cung ứng

```
SELECT COUNT(DISTINCT pid)
FROM SupplyProduct;
```

- Có tổng cộng bao nhiêu nhân viên làm cho các hãng ở Paris

```
SELECT SUM(size) FROM Supplier
WHERE city = 'Paris';
```

- Đưa ra số lượng mặt hàng trung bình mà hãng S1 cung ứng

```
SELECT AVG(quantity)
FROM SupplyProduct
WHERE sid = 'S1';
```

139

139

## Một số truy vấn phức tạp

- Đưa ra tên của hãng S1 và tổng số lượng các mặt hàng mà hãng đó cung ứng

```
SELECT sname, SUM(quantity)
FROM Supplier S, SupplyProduct SP
WHERE S.sid = SP.sid AND S.sid = 'S1'
GROUP BY sname;
```

- Đưa ra mã số các hãng cung ứng và số lượng trung bình các mặt hàng được cung ứng bởi từng hãng

```
SELECT sid, AVG(quantity) FROM SupplyProduct
GROUP BY sid;
```

- Đưa ra mã số các hãng cung ứng mà số lượng mặt hàng trung bình được cung cấp bởi hãng đó là trong khoảng từ 75 đến 100

```
SELECT sid, AVG(quantity) FROM SupplyProduct
GROUP BY sid HAVING AVG(quantity) BETWEEN 75 AND 100
```

140

140

## Thêm bản ghi vào bảng (INSERT)

- **Lệnh INSERT**
  - Dùng để thêm 1 hay nhiều dòng vào bảng
  - Để thêm dữ liệu
    - Tên quan hệ
    - Danh sách các thuộc tính cần thêm dữ liệu
    - Danh sách các giá trị tương ứng
  - Thứ tự các giá trị phải trùng với thứ tự các cột
  - Có thể thêm giá trị NULL ở những thuộc tính không là khóa chính và NOT NULL
  - Câu lệnh INSERT sẽ gặp lỗi nếu vi phạm RBTV
    - Khóa chính
    - Tham chiếu
    - NOT NULL - các thuộc tính có ràng buộc NOT NULL bắt buộc phải có giá trị

141

141

## Thêm bản ghi vào bảng (tiếp)

Persons

| P_Id | LastName | FirstName | Address | City |
|---|---|---|---|---|
| 1 | Hansen | Ola | Timoteivn 10 | Sandnes |
| 2 | Svendson | Tove | Borgvn 23 | Sandnes |
| 3 | Pettersen | Kari | Storgt 20 | Stavanger |

```
INSERT INTO Persons
VALUES (4,'Nilsen', 'Johan', 'Bakken 2', 'Stavanger')
```

The "Persons" table will now look like this:

| P_Id | LastName | FirstName | Address | City |
|---|---|---|---|---|
| 1 | Hansen | Ola | Timoteivn 10 | Sandnes |
| 2 | Svendson | Tove | Borgvn 23 | Sandnes |
| 3 | Pettersen | Kari | Storgt 20 | Stavanger |
| 4 | Nilsen | Johan | Bakken 2 | Stavanger |

142

142

## Thêm bản ghi vào bảng (tiếp)

**Cú pháp:**

- **INSERT INTO** table[(col1,col2,…)]
  **VALUES** (exp1,exp2,…)

- **INSERT INTO** table[(col1,col2,…)]
  **SELECT** col1,col2, …
  **FROM** tab1, tab2, …
  **WHERE** <dieu_kien>

143

143

## Thêm bản ghi vào bảng (tiếp)

- Ví dụ
  - **INSERT INTO** Student
    **VALUES** ('1179','Jane','California');
  - **INSERT INTO** Student(Id, Name, Suburb)
    **VALUES** ('1180','David','NewYork');
  - **INSERT INTO** Student(Name, Id, Suburb)
    **VALUES** ('Mary','1181','Texas');
  - **INSERT INTO** Student(Id, Name, Suburb)
    **VALUES** ('1182','John','Ohio'), ('1183','Tom','Georgia'), ('1184','Declan','Arizona');

144

144

## Xóa bản ghi trong bảng (DELETE)

**Lệnh UPDATE**

- Dùng để xóa các dòng của bảng (xóa mọi dòng, xóa có điều kiện)
- Cú pháp

```
DELETE FROM <tên bảng>
[WHERE <điều kiện xóa>]
```

- Có thể dùng DELETE FROM <table> : xóa mọi dòng
- Ví dụ:

```
DELETE FROM SupplyProduct
WHERE sid = 'S4';
DELETE FROM  Student
WHERE  Suburb = 'Indiana';
```

145

145

## Sửa dữ liệu trong bảng (UPDATE)

**Lệnh UPDATE**

- Dùng để thay đổi giá trị của thuộc tính cho các dòng của bảng
- Cú pháp

```
UPDATE <tên bảng>
SET <tên thuộc tính>=<giá trị mới>,
    <tên thuộc tính>=<giá trị mới>,
    ...
[WHERE <điều kiện sửa đổi>]
```

146

146

# Sửa dữ liệu trong bảng

- Ví dụ:
  - Hãng S1 chuyển tới Milan

```
UPDATE Supplier
SET city = 'Milan'
WHERE sid = 'S1';
```

  - Tất cả các mặt hàng được cung cấp với số lượng nhỏ hơn 100 đều tăng số lượng lên 1.5 lần

```
UPDATE SupplyProduct
SET quantity = quantity * 1.5
WHERE quantity < 100;
```

147

147

148

148

# Lời hay ý đẹp

"Người kém thông minh nhưng say sưa với công việc, tiến mạnh và xa hơn người cực thông minh mà lãnh đạm với công việc".

J. Deval



149