

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

Nguyễn Hồng Phương

phuongnh@soict.hut.edu.vn

<http://is.hut.edu.vn/~phuongnh>

Bộ môn Hệ thống thông tin
Viện Công nghệ thông tin và Truyền thông
Đại học Bách Khoa Hà Nội

1

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 (*Query By Example*)
 - SQL (*Structured Query Language*)
- Kết luận

2

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

3

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

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

Product

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

4

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

5

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 ra 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

6

Phân loại các ngôn ngữ truy vấn

- Ngôn ngữ đại số
 - 1 câu hỏi = 1 tập các phép toán trên các quan hệ
 - Được biểu diễn bởi một biểu thức đại số (quan hệ)
- Ngôn ngữ tính toán vị từ
 - 1 câu hỏi = 1 mô tả của các bộ mong muốn
 - Được đặc tả bởi một vị từ mà các bộ phải thỏa mãn
 - Phân biệt 2 lớp:
 - ngôn ngữ tính toán vị từ biến bộ
 - ngôn ngữ tính toán vị từ biến miền

7

Ngôn ngữ đại số quan hệ

8

Tổng quan

- Gồm các phép toán tương ứng với các thao tác trên các quan hệ
- Mỗi phép toán
 - 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

9

Phân loại các phép toán đại số quan hệ

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

10

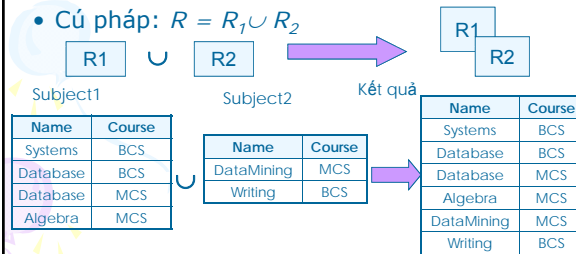
Phép toán tập hợp

- Định nghĩa: Quan hệ khả hợp
 - 2 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 $D_1 \times D_2 \times \dots \times D_n$
 - s xác định trên $D'_1 \times D'_2 \times \dots \times D'_m$
 - $\rightarrow D_i = D'_i$ và $n=m$

11

Phép 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: $R = R_1 \cup R_2$



Phép giao

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



Subject1		Subject2		Kết quả	
Name	Course	Name	Course	Name	Course
Systems	BCS	DataMining	MCS	Systems	BCS
Database	BCS	Database	MCS	Database	MCS
Database	MCS	Systems	BCS		
Algebra	MCS	Writing	BCS		

13

Phép trừ

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

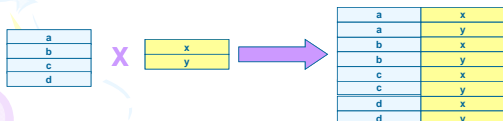


Subject1		Subject2		Kết quả	
Name	Course	Name	Course	Name	Course
Systems	BCS	DataMining	MCS	Database	BCS
Database	BCS	Database	MCS	Database	MCS
Database	MCS	Systems	BCS	Algebra	MCS
Algebra	MCS	Writing	BCS		

14

Phép tích Đề-các

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



15

Ví dụ phép tích Đề-các

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

Student_Sport

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

16

Phép chiếu

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



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

Student			$\Pi_{name}(Student)$	
Id	Name	Suburb	Name	
1108	Robert	Kew	Robert	
3936	Glen	Bundoora	Glen	
8507	Norman	Bundoora	Norman	
8452	Mary	Balwyn	Mary	

17

Phép chọn

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



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

Song & Bundoora

Student

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

18

Phép chọn - Điều kiện ?

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

19

Ví dụ: chọn và chiếu

- Đưa ra tên của các sinh viên sống ở Bundoora

$$\Pi_{name}(\sigma_{suburb="Bundoora"} Student)$$

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

→

Name
Glen
Norman

20

Phép kết nối (join) 2 quan hệ r và s

- Khái niệm ghép bộ: $u = (a_1, \dots, a_n); v = (b_1, \dots, b_m)$
 $(u, v) = (a_1, \dots, a_n, b_1, \dots, b_m)$
- 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: $R_1 \bowtie_{<join_condition>} R_2$

a	r			
b	r			
c	v			

⋈

r	x			
s	v			
t	z			

→

a	r	x		
b	r	x		

21

Phép kết nối - Ví dụ:

- Đưa ra danh sách các sinh viên và mã khóa học mà sinh viên đó tham gia

Student			Enrol	
Id	Name	Suburb	SID	Course
1108	Robert	Kew	3936	101
3936	Glen	Bundoora	1108	113
8507	Norman	Bundoora	8507	101
8452	Mary	Balwyn		

⋈
Id=SID

Kết quả

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

22

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: $R_1 * R_2$

23

Phép kết nối tự nhiên - Ví dụ:

Takes		Enrol				
SID	SNO	SID	Course	SID	SNO	Course
1108	21	3936	101	1108	21	113
1108	23	1108	113	1108	23	113
8507	23	8507	101	8507	23	101
8507	29			8507	29	101

24

Ví dụ: chọn, chiếu, kết nối

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

$$\Pi_{name, Course} (\sigma_{suburb="Bundoora"} (Student \bowtie_{id=SID} Enrol))$$

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

25

Phép kết nối ngoài

- Phép kết nối ngoài trái

a	r
b	r
c	v

 \bowtie

r	x
s	y
t	z

 \rightarrow

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

 \rightarrow

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

26

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

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

27

Phép chia

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

- Cú pháp: $R = R_1 \div R_2$

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

28

Phép chia (tiếp)

a	x
a	y
a	z
b	x
c	y

 \div

x
z

 \rightarrow

a

- 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

29

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 ∪ 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

30

Luyện tập

• Phép giao (intersection)

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

Ví dụ:

r	(A	B	C)	s	(A	B	C)
	a ₁	b ₁	c ₁		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 ₁	b ₂	c ₂
	a ₂	b ₂	c ₂				
	a ₃	b ₂	c ₂				

r ∩ s = g	(A	B	C)
	a ₁	b ₁	c ₁
	a ₁	b ₂	c ₂

31

Luyện tập

• Phép trừ (minus)

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

Ví dụ:

r	(A	B	C)	s	(A	B	C)
	a ₁	b ₁	c ₁		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 ₁	b ₂	c ₂
	a ₂	b ₂	c ₂				
	a ₃	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

32

Luyện tập

• Phép tích Đề - Các (Cartesian Product)

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

Ví dụ:

r	(A	B	C)	s	(D	E)	r × s = p	(A	B	C	D	E)
	a ₁	b ₁	1		1	e ₁		a ₁	b ₁	1	1	e ₁
	a ₁	b ₁	2		2	e ₂		a ₁	b ₁	1	2	e ₂
	a ₂	b ₂	2		3	e ₃		a ₁	b ₁	1	3	e ₃
	a ₂	b ₂	2		2	e ₂		a ₂	b ₂	2	1	e ₁
	a ₂	b ₂	2		3	e ₃		a ₂	b ₂	2	2	e ₂
	a ₃	b ₃	3		1	e ₁		a ₂	b ₂	2	3	e ₃
	a ₃	b ₃	3		2	e ₂		a ₃	b ₃	3	1	e ₁
	a ₃	b ₃	3		3	e ₃		a ₃	b ₃	3	2	e ₂
								a ₃	b ₃	3	3	e ₃

33

Luyện tập

• Phép chiếu (Projection)

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

Ví dụ:

$$X = \{ A, B \}; Y = \{ C \}$$

r	(A	B	C)	$\Pi_X(r) = s_1$	(A	B)	$\Pi_Y(r) = s_2$	(C)
	a ₁	b ₁	c ₁		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 ₁
	a ₂	b ₂	c ₂		a ₃	b ₂		c ₁
	a ₃	b ₂	c ₂					c ₁

34

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)	$\sigma_{A=a1 \wedge C=c2}(r) = r_2$	(A	B	C)
	a ₁	b ₁	c ₁		a ₁	b ₁	c ₂
	a ₁	b ₂	c ₂		a ₁	b ₂	c ₂
	a ₁	b ₁	c ₂				

35

Luyện tập

• Phép kết nối (join)

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

Ví dụ:

r	(A	B	C)	s	(D	E)	$r \bowtie_F s = k$	(A	B	C	D	E)
	a ₁	b ₁	1		1	e ₁		a ₁	b ₁	1	1	e ₁
	a ₁	b ₁	1		2	e ₂		a ₁	b ₁	1	2	e ₂
	a ₁	b ₁	1		3	e ₃		a ₁	b ₁	1	3	e ₃
	a ₂	b ₂	2		2	e ₂		a ₂	b ₂	2	2	e ₂
	a ₂	b ₂	2		3	e ₃		a ₂	b ₂	2	3	e ₃
	a ₃	b ₃	3		3	e ₃		a ₃	b ₃	3	3	e ₃

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

36

Luyện tập

• Kết nối tự nhiên (natural join)

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

p	(A B C)	q	(C D)	p*q = z	(A B C D)
a ₁	b ₁ 1	1	d ₁	a ₁	b ₁ 1 d ₁
a ₂	b ₁ 1	1	d ₂	a ₁	b ₁ 1 d ₂
a ₃	b ₂ 2	2	d ₂	a ₂	b ₁ 1 d ₁
				a ₂	b ₁ 1 d ₂
				a ₃	b ₂ 2 d ₂

37

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)	s	(D E)	p+s = q	(A B C)
a ₁	b ₁ 1 1 e ₁	1	e ₁	a ₁	b ₁ 1
a ₁	b ₁ 1 2 e ₂	2	e ₂	a ₂	b ₂ 2
a ₁	b ₁ 1 3 e ₃	3	e ₃	a ₃	b ₃ 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				

38

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)	SP(S# P# QTY)
S1 Smith 20 London	S1 P1 300
S2 Jones 10 Paris	S1 P2 200
S3 Black 30 Paris	S1 P3 400
	S2 P1 300
	S2 P2 400
	S3 P2 200

P(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

39

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.

40

Lời giải của bài tập

- Đưa ra danh sách các mặt hàng màu đỏ:

$$\sigma_{\text{COLOR} = \text{'red'}}(P)$$
- Cho biết S# của các hãng cung ứng mặt hàng 'P1' hoặc 'P2':

$$\Pi_{S\#}(\sigma_{P\# = \text{'P1'}} \vee P\# = \text{'P2'}}(SP))$$
- Liệt kê S# của các hãng cung ứng cả hai mặt hàng 'P1' và 'P2':

$$\Pi_{S\#}(\sigma_{P\# = \text{'P1'}}(SP)) \cap \Pi_{S\#}(\sigma_{P\# = \text{'P2'}}(SP))$$
- Đưa ra S# của các hãng cung ứng ít nhất một mặt hàng màu đỏ:

$$\Pi_{S\#}(SP * \sigma_{\text{COLOR} = \text{'red'}}(P))$$
- Đưa ra S# của các hãng cung ứng tất cả các mặt hàng:

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

41

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

42

Bài tập về nhà

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

43

Bài tập về nhà

- Đưa ra {sid} của các hãng cung ứng tất cả các mặt hàng màu đỏ
- Đưa ra {sname} của các hãng cung ứng ít nhất một mặt hàng màu đỏ hoặc màu xanh
- Đưa ra {sname} của các hãng cung ứng ít nhất 1 mặt hàng màu đỏ và một mặt hàng màu xanh
- Đưa ra {sid} của các hãng không cung ứng mặt hàng nào

44



45

Ngôn ngữ QBE

46

QBE (Query-By-Example)

- Là một ngôn ngữ truy vấn dữ liệu
- Các câu truy vấn được thiết lập bởi một giao diện đồ họa
- 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, ...

47

Truy vấn trên một quan hệ

- P.~ Print

Student	ID	Name	Suburb
		P_x	Bundoora

- Biểu thức đại số quan hệ tương đương

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

48

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

49

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	

50

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.

- G. - Grouping

51

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
	_id		G.P.

Condition
COUNT._id > 5

52

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

53

Tính đầy đủ của QBE

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

54

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

55

Định nghĩa dữ liệu trong QBE (tiếp)

- Các khung nhìn

I.View V	I.	ID	Name	Course
	I.	_id	_name	_course

Student	ID	Name	Suburb	Enrol	SID	Course
	_id	_name			_id	_course

56

Ngôn ngữ SQL

57

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



ORACLE

58

Các thành phần của SQL

- Ngôn ngữ định nghĩa dữ liệu (*Data Definition Language*)
 - 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 (*Data Manipulation Language*)
 - Thêm, xóa, sửa dữ liệu trong CSDL
 - Truy vấn dữ liệu
- Ngôn ngữ điều khiển dữ liệu (*Data Control Language*)
 - Khai báo bảo mật thông tin
 - Quyền hạn của người dùng trong khai thác CSDL

59

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

60

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

61

Cú pháp

• Tạo bảng

```
CREATE TABLE tab(
  col1 type1(size1)[NOT NULL], ...,
  col2 type2(size2)[NOT NULL], ...,
  ....
  [CONSTRAINT <constraint name> <constraint
  type> clause]
  ...
);
```

• Xoá bảng

```
DROP TABLE tab
```

62

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)
);
```

63

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)
);
```

64

Kiểu ràng buộc

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

CONSTRAINT <name>

CHECK <condition>

- RBTv về khoá ngoại hay phụ thuộc tồn tại

CONSTRAINT <name> **FOREIGN KEY** (fk1,fk2,...)
REFERENCES tab(k1,k2);

65

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>;
```

66

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);

67

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>

68

Ngôn ngữ truy vấn dữ liệu

• Cú pháp câu lệnh SQL:

SELECT [DISTINCT] <DS> | * | <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>]

69

Truy vấn không điều kiện trên một bảng

• Tìm thông tin từ các cột của bảng

> SELECT ColumnName, ColumnName, ...
FROM TableName;
> SELECT *
FROM TableName;

• Ví dụ

SELECT Name
FROM Student;

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

$\Pi_{\text{name}}(\text{Student})$

Name
Robert
Glen
Norman
Mary

70

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;

71

Truy vấn có điều kiện trên 1 bảng

• Chọn các bản ghi (dòng)

SELECT ColumnName, ColumnName, ...
FROM TableName
WHERE condition_expression;

• Ví dụ

SELECT *
FROM Student
WHERE suburb='Bundoora';

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

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

Id	Name	Suburb
3936	Glen	Bundoora
8507	Norman	Bundoora

72

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;
```

73

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 (\Leftrightarrow (attr>=val1) and (attr<=val2))
 - attr IN (val1, val2, ...) (\Leftrightarrow (attr=val1) or (attr=val2) or ...)
 - Kiểu dữ liệu chuỗi
 - LIKE: sử dụng đối sánh mẫu chuỗi với các ký tự % hoặc _ (thay thế cho 1 ký tự bất kỳ), * hay % (thay thế cho 1 chuỗi ký tự bất kỳ)

74

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');
```

75

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%';
```

76

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)
```

77

### Truy vấn phức tạp trên nhiều bảng

- Điều kiện kết nối  

```
SELECT T1.C1,T1.C2,T2.C1,T2.C4, ...
FROM T1, T2
WHERE condition_expression
```
- Ví dụ: đưa ra danh sách mã sinh viên (Id), tên sinh viên (Name), thành phố (Suburb), mã khóa học (Course) mà các sinh viên đã đăng ký  

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

78

## Truy vấn phức tạp trên nhiều bảng Một số ví dụ khác:

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

79

## Loại trừ các bản ghi trùng nhau

- Từ khoá **DISTINCT**

```
SELECT DISTINCT <bt1>, <bt2>, ...
FROM <bang1>, <bang2>, ...
```

- 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
```

80

## Tìm kiếm có sắp xếp

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

```
SELECT <bt1>, <bt2>, ...
FROM <bang1>, <bang2>, ...
[WHERE <điều kiện chọn>]
ORDER BY <tt1> | <biểu thức số 1> [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
```

81

## Phân nhóm các bản ghi kết quả

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

```
SELECT <bt1>, <bt2>, ...
FROM <bang1>, <bang2>, ...
[WHERE <điều kiện chọn>]
[GROUP BY <tt1>, <tt2>, ...]
```

- Cột được chỉ ra trong 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
```

82

## Điều kiện hiển thị các bản ghi kết quả

- Lựa chọn các bản ghi kết quả để hiển thị

```
SELECT <bt1>, <bt2>, ...
FROM <bang1>, <bang2>, ...
[WHERE <điều kiện chọn>]
HAVING <điều kiện in kết quả>
```

- 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
```

83

## Các phép toán tập hợp: UNION, MINUS, INTERSECT

- 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
```

84

## 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, )
    - vd: `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

85

## 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');
```

86

## 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 đồng 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');
```

87

## 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

88

## 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);
```

89

## Các hàm thư viện

- Hàm tính toán trên nhóm các bản ghi
  - MAX/MIN
  - SUM
  - AVG
  - COUNT
- Hàm tính toán trên bản ghi
  - Hàm toán học: ABS, SQRT, LOG, EXP, SIGN, ROUND
  - Hàm xử lý chuỗi ký tự: LEN, LEFT, RIGHT, MID
  - Hàm xử lý thời gian: DATE, DAY, MONTH, YEAR, HOUR, MINUTE, SECOND
  - Hàm chuyển đổi kiểu giá trị: FORMAT

90

### 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';
```

91

### Một số truy vấn phức tạp

- Đưa ra tên của hãng S1 và tổng số 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
```

92

### Các câu lệnh cập nhật dữ liệu

- Thêm  

```
>INSERT INTO table[(col1,col2,...)]
VALUES (exp1,exp2,...)
>INSERT INTO table[(col1,col2,...)]
SELECT col1,col2, ...
FROM tab1, tab2, ...
WHERE <dieu_kien>
```
- Ví dụ  

```
>INSERT INTO Student(Id, Name, Suburb)
VALUES ('1179','David','Evr')
```

93

### Các câu lệnh cập nhật dữ liệu

- Xóa dữ liệu:  

```
DELETE FROM <Tên bảng>
WHERE <Điều kiện xóa>;
```
- Ví dụ:  

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

94

### Các câu lệnh cập nhật dữ liệu

- Sửa đổi dữ liệu:  

```
UPDATE <tên bảng> SET (<Tên cột> = Giá trị mới , ...)
[WHERE <Điều kiện sửa đổi>;]
```
- 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  

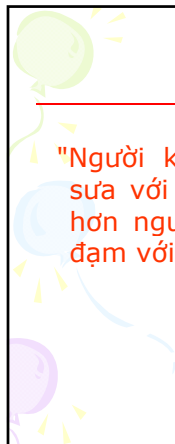
```
UPDATE SupplyProduct SET quantity = quantity * 1.5
WHERE quantity < 100;
```

95



96





## 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

97