

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

1

Nội dung

- Các cách tiếp cận đối với thiết kế ngôn ngữ cho mô hình quan hệ
 - Giới thiệu một số ngôn ngữ hình thức, trừu tượng với mô hình quan hệ
 - So sánh và đánh giá
- Một số ngôn ngữ 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 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 khóa 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	BCE	CSCE
101	MCS	CSCE

6

Các ngôn ngữ truy vấn với mô hình quan hệ

- Ngôn ngữ đại số
 - 1 câu hỏi được biểu diễn bởi 1 tập các phép toán trên các quan hệ hay một biểu thức đại số trên các quan hệ.
 - biểu thức đại số trên các quan hệ trả lời cho câu hỏi How?
- Ngôn ngữ đại số thuộc lớp ngôn ngữ thủ tục

7

Các ngôn ngữ truy vấn với mô hình quan hệ...

- Ngôn ngữ tính toán vị từ
 - 1 câu hỏi được biểu diễn bởi 1 mô tả tập các bộ mong muốn thông qua cách đặc tả các vị từ hay một biểu thức logic mà các bộ phải thỏa mãn
 - biểu thức logic trên các bộ hay các miền thuộc tính của các quan hệ trả lời cho câu hỏi What?
- Ngôn ngữ tính toán vị từ thuộc lớp ngôn ngữ phi thủ tục
- Phân biệt 2 lớp con của Ngôn ngữ tính toán vị từ :
 - Ngôn ngữ tính toán vị từ biến bộ
 - Ngôn ngữ tính toán vị từ biến miền

8

Các ngôn ngữ truy vấn với mô hình quan hệ...

- Hai ngôn ngữ truy vấn trừu tượng:
Ngôn ngữ đại số và
Ngôn ngữ tính toán vị từ
có khả năng biểu diễn như nhau và
được xem như một chuẩn đối với việc
đánh giá các ngôn ngữ truy vấn được cài
đặt trong 1 DBMS quan hệ.

9

Ngôn ngữ đại số quan hệ

10

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 gồm 1 hay nhiều phép toán lồng nhau theo một thứ tự xác định
- Kết quả thực hiện một biểu thức đại số là một quan hệ
- Được cài đặt trong phần lớn các DBMS hiện nay

11

Các phép toán đại số quan hệ

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

12

Các phép toán tập hợp

- Quan hệ khả hợp

Định nghĩa 1: 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 $n=m$

13

Các phép toán tập hợp

- Quan hệ khả hợp

Định nghĩa 2: 2 quan hệ r và s được gọi là khả hợp nếu chúng được xác định trên cùng tập thuộc tính và các thuộc tính cùng tên có cùng miền giá trị

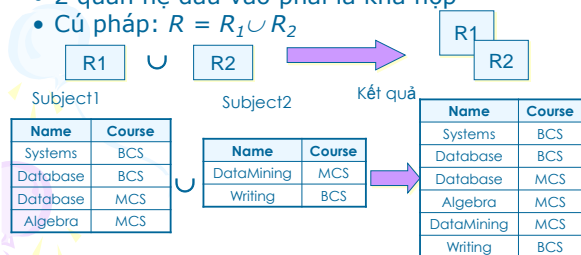
- 2 định nghĩa 1 và 2 về Quan hệ khả hợp là tương đương

14

Phép hợp

Định nghĩa 3: Phép hợp của 2 quan hệ khả hợp là 1 quan hệ gồm các bộ thuộc ít nhất 1 trong 2 quan hệ đầu vào

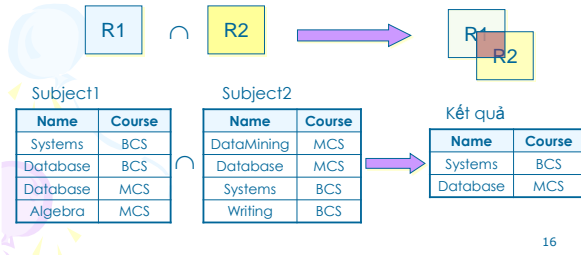
- 2 quan hệ đầu vào phải là khả hợp
- Cú pháp: $R = R_1 \cup R_2$



Phép giao

Định nghĩa 4: Phép giao của 2 quan hệ khả hợp là 1 quan hệ gồm các bộ thuộc cả 2 quan hệ đầu vào

- 2 quan hệ đầu vào phải là khả hợp
- Cú pháp: $R_1 \cap R_2$

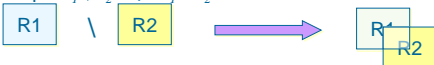


16

Phép trừ

Định nghĩa 5: Phép trừ của 2 quan hệ khả hợp là 1 quan hệ gồm các bộ thuộc quan hệ thứ nhất nhưng không thuộc quan hệ thứ hai

- 2 quan hệ đầu vào 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

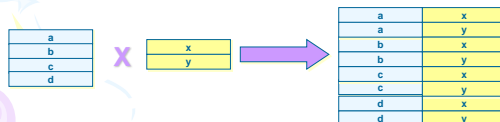
Name	Course
Database	BCS
Algebra	MCS

17

Phép tích Đề-các

Định nghĩa 6: Phép tích Đề-các của 2 quan hệ là phép ghép mỗi một bộ của quan hệ thứ nhất với mỗi một bộ của quan hệ thứ hai

- Cú pháp: $R = R_1 \times R_2$
- Khái niệm ghép bộ: $u = (a_1, \dots, a_n); v = (b_1, \dots, b_m)$ thì phép ghép u và v là: $(u, v) = (a_1, \dots, a_n, b_1, \dots, b_m)$
- (u, v) được gọi là một cặp bộ của 2 quan hệ tương ứng



18

Ví dụ phép tích Đề-các

Student

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

Sport

SportID	Sport
05	Swimming
09	Dancing

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

19

Phép chiếu

Định nghĩa 7: Phép chiếu trên 1 quan hệ là phép toán chỉ chọn một số thuộc tính từ quan hệ.

- Cú pháp: $\Pi_{A_1, A_2, \dots, (R)}$



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

Student

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

 $\Pi_{name}(Student)$

Name
Robert
Glen
Norman
Mary

20

Phép chọn

Định nghĩa 8: Phép chọn trên 1 quan hệ là phép toán lọc ra một tập con các bộ trong quan hệ thoả mãn các điều kiện cho trước.

- Cú pháp: $\sigma_{\langle \text{condition} \rangle}(R)$



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

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

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

Id	Name	Suburb
3936	Glen	Bundoora
8507	Norman	Bundoora

21

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

22

Ví dụ: chọn và chiếu

- Đưa ra tên của các sinh viên sống ở Bundoora

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

Student

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

Name
Glen
Norman

23

Phép kết nối

Định nghĩa 9: Phép kết nối 2 quan hệ là phép ghép các cặp bộ của 2 quan hệ thoả mãn 1 điều kiện nào đó trên chúng.

- Điều kiện kết nối còn gọi là Biểu thức kết nối
- 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 này với 1 thuộc tính của quan hệ kia.
- Cú pháp:

$$R_1 \bowtie_{\langle \text{join_condition} \rangle} R_2$$



24

Phép kết nối - Ví dụ:

- Đưa ra danh sách các sinh viên cùng với khoá học sinh viên đăng ký

$Student \bowtie_{Id=SID} Enrol$

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		

Kết quả

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

25

Phép kết nối bảng & kết nối tự nhiên

- Định nghĩa 10:** Nếu phép so sánh trong điều kiện kết nối là phép so sánh bằng thì phép kết nối này được gọi là **kết nối bằng**
- Định nghĩa 11:** 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: $R_1 * R_2$

26

Phép kết nối tự nhiên - Ví dụ:

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

*

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

27

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

Kết quả

Name	Course
Glen	101
Norman	101

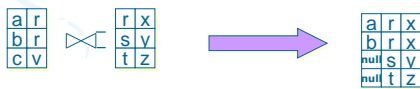
28

Phép kết nối ngoài

- Phép kết nối ngoài trái



- Phép kết nối ngoài phải



29

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

Kết quả

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

30

Phép chia

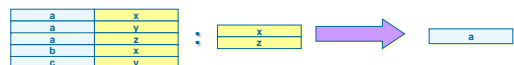
- Định nghĩa 12:** Phép chia quan hệ r bậc n cho quan hệ s bậc m ($m < n$) với sơ đồ quan hệ của s là tập con của sơ đồ quan hệ của r là một tập các $(n-m)$ -bộ t sao cho khi ghép mọi bộ thuộc s với t thì ta đều có một bộ thuộc r

- Cú pháp: $R = R_1 : R_2$

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

31

Phép chia (tiếp)



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

Subject		Course		Kết quả	
Name	Course	Course		Name	
Systems	BCS	BCS		Database	
Database	BCS	MCS			
Database	MCS				
Algebra	MCS				

32

Luyện tập

• Phép hợp (Union)

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

Ví dụ:

r	(A B C)	s	(A B C)	$r \cup s = h$	(A B C)
a ₁	b ₁ c ₁	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 ₁ c ₂
a ₁	b ₂ c ₂	a ₁	b ₂ c ₂	a ₁	b ₂ c ₁
a ₂	b ₂ c ₂			a ₁	b ₂ c ₂
a ₃	b ₂ c ₂			a ₂	b ₂ c ₂
				a ₃	b ₂ c ₂

33

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		

$r \cap s = g$	(A B C)
a ₁	b ₁ c ₁
a ₁	b ₂ c ₂

34

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

35

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)	$r \times s = p$	(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 ₁	b ₁ 3	3	e ₃	a ₁	b ₁ 1 3 e ₃
a ₂	b ₂ 2			a ₂	b ₂ 2 1 e ₁
a ₂	b ₂ 2			a ₂	b ₂ 2 2 e ₂
a ₂	b ₂ 2			a ₂	b ₂ 2 3 e ₃
a ₃	b ₃ 3			a ₃	b ₃ 3 1 e ₁
a ₃	b ₃ 3			a ₃	b ₃ 3 2 e ₂
a ₃	b ₃ 3			a ₃	b ₃ 3 3 e ₃

36

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 ₂
	a ₂ b ₂ c ₂		a ₂ b ₂		c ₂
	a ₃ b ₂ c ₂		a ₃ b ₂		c ₂

37

Luyện tập

• Phép chọn (Selection)

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

Ví dụ:

r	(A B C)	$\sigma_{A=a1}(r) = r_1$	(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 ₁		a ₁ b ₁ c ₂	
	a ₁ b ₁ c ₂	a ₁ b ₁ c ₂		a ₁ b ₂ c ₂	
	a ₁ b ₂ c ₂	a ₁ b ₂ c ₂			
	a ₂ b ₂ c ₂				
	a ₃ b ₂ c ₂				

38

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:

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

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

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 ₂ b ₂ 2		2 e ₂		a ₂ b ₂ 2 2 e ₂
	a ₃ b ₃ 3		3 e ₃		a ₃ b ₃ 3 3 e ₃

39

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)	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 ₁ b ₁ 1		2 d ₂		a ₁ b ₁ 1 d ₂
	a ₂ b ₂ 2		1 d ₁		a ₂ b ₂ 2 d ₁
	a ₂ b ₂ 2		2 d ₂		a ₂ b ₂ 2 d ₂

40

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

$p \div s = q$	(A	B	C)
	a_1	b_1	1
	a_2	b_2	2
	a_3	b_3	3

41

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

42

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.

43

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:

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

44

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

SupplyProduct

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

Product

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

45

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

46

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

47

Ngôn ngữ tính toán vị từ

48

Tổng quan

• Ứng dụng logic toán vào CSDL

- Sử dụng các vị từ hay các công thức nguyên tố để biểu diễn các điều kiện của truy vấn
- Công thức nguyên tố : là phép so sánh giữa 2 biến hay giữa 1 biến và 1 hằng, hay 1 phép kiểm tra bộ thuộc quan hệ và luôn nhận 1 trong 2 giá trị ĐÚNG hoặc SAI
 - **Biến**: 1 đại lượng biến thiên trong 1 miền giá trị
 - **Hằng**: 1 đại lượng không đổi
- **Biểu thức logic** : là 1 công thức được thiết lập từ các công thức nguyên tố sử dụng các phép toán logic và phép lượng hóa với các lượng từ
 - **Phép toán logic**: phủ định (\neg) kéo theo (\Rightarrow), và (\wedge) hoặc (\vee)
 - **Lượng từ**: với mọi (\forall), tồn tại (\exists)

• Hai loại:

- Phép tính vị từ biến bộ
- Phép tính vị từ biến miền

49

Phép tính vị từ biến bộ

- Đ/n: là phép tính vị từ với các biến nhận giá trị là các bộ hay còn gọi là biến bộ.

• Định nghĩa hình thức

- 1 biểu thức phép tính vị từ biến bộ có dạng: $\{ t \mid P(t) \}$

- mô tả tập các bộ kết quả t sao cho vị từ P là đúng đối với t
- P là một biểu thức có duy nhất 1 biến tự do t

• Một số quy ước:

- $t[A]$: giá trị của bộ t tại thuộc tính A
- $t[X]$: giá trị của bộ t trên tập các thuộc tính X
- $t \in R$: bộ t là một bộ trong quan hệ R

50

Công thức nguyên tố

• $t \in R$

- t là một biến bộ
- R là một quan hệ (không sử dụng phép toán \neq)

• $t[x] \theta u[y]$

- t và u là các biến bộ
- x và y lần lượt là 1 thuộc tính mà trên đó t và u được xác định
- θ là một phép toán so sánh ($<$, $=$, $>$, \leq , \neq , \geq)

❖ Ví dụ:

$s \in \text{Student}$
 $e \in \text{Enrol}$
 $s[id] = e[sid]$

51

Công thức nguyên tố (2)

• $t[x] \theta c$

- t là một biến bộ
- x là một thuộc tính mà trên đó u xác định
- θ là một phép so sánh
- c là một hằng trong miền của thuộc tính x .

❖ Ví dụ

$s[\text{Suburb}] = \text{"Bundoora"}$

52

Công thức

- Một công thức nguyên tố là một công thức.
- $P1$ là công thức
 $\Rightarrow \neg P1, (P1)$ là các công thức
- $P1$ và $P2$ là công thức
 $\Rightarrow P1 \wedge P2, P1 \vee P2, P1 \Rightarrow P2$ là các công thức
- $P1$ là công thức chứa 1 biến bộ tự do u , và R là 1 quan hệ
 $\Rightarrow \exists u \in r (P1(u)), \forall u \in r (P1(u))$ cũng là các công thức

53

Các phép biến đổi tương đương

- $P1 \wedge P2 \Leftrightarrow \neg(\neg P1 \vee \neg P2)$
- $t \in r (P1(t)) \Leftrightarrow \neg \exists t \in r (\neg P1(t))$
- $P1 \Rightarrow P2 \Leftrightarrow \neg P1 \vee P2$

54

Bài tập

- Biểu diễn câu hỏi bằng ngôn ngữ tính toán vị từ biến bộ

55

Tính an toàn của các biểu thức

- Đặt vấn đề:
 $\{t\} \neg(t \in r)$
- K/n miền giá trị của biểu thức: $DOM(P)$
 - Các hằng xuất hiện trong P
 - Các giá trị của các thuộc tính của các bộ của các quan hệ xuất hiện trong P
- Ví dụ: $P(t) = t \in Sport \wedge t[Sport] \neq "Football"$
 $DOM(P) = \{"05", "09", "Swimming", "Dancing", "Football"\}$

Sport

SportID	Sport
05	Swimming
09	Dancing

56

Biểu thức an toàn

- Đ/n : $\{t \mid P(t)\}$ là an toàn nếu tất cả các giá trị xuất hiện trong kết quả là các giá trị từ $\text{DOM}(P)$

❖ Ví dụ

- ✓ $P(t) = t \in \text{Sport} \wedge t[\text{Sport}] \neq \text{"Football"}$ là an toàn
- ✗ $P(t) = \neg(t \in \text{Sport})$ là không an toàn

57

Phép tính vị từ biến miền

- Đ/n: là phép tính vị từ với các biến nhận giá trị trong miền giá trị của một thuộc tính hay còn gọi là biến miền.

- Định nghĩa hình thức

– 1 biểu thức phép tính vị từ biến miền có dạng: $\{ \langle x_1, \dots, x_n \rangle \mid P(x_1, \dots, x_n) \}$

- mô tả tập các bộ kết quả (x_1, \dots, x_n) sao cho vị từ P là đúng đối với x_1, \dots, x_n
- x_1, \dots, x_n là các biến miền hay các hằng miền
- P là một biểu thức chỉ có các biến tự do x_i

58

Công thức nguyên tố

- $\langle x_1, \dots, x_n \rangle \in r$
 - r là 1 quan hệ trên n thuộc tính
 - x_1, \dots, x_n là các biến miền hay các hằng miền.
- $x \theta y$
 - x và y là các biến miền
 - θ là một phép so sánh đơn giản ($<, =, >, \leq, \neq, \geq$).

❖ Ví dụ

$\langle x, y, z \rangle \in \text{Student}$
 $\langle u, v \rangle \in \text{Enrol}$
 $x = u$

59

Công thức nguyên tố(2)

- $x \theta c$
 - x là một biến miền
 - θ là một phép so sánh
 - c là một hằng trong miền của thuộc tính của x

❖ Ví dụ

$Z = \text{"Bundoora"}$

60

Công thức

- Một công thức nguyên tố là một công thức.
- $P1$ là 1
 $\Rightarrow \neg P1, (P1)$ là công thức
- $P1$ và $P2$ là công thức
 $\Rightarrow P1 \wedge P2, P1 \vee P2, P1 \Rightarrow P2$ là công thức
- $P1(x)$ là một công thức với biến tự do x
 $\Rightarrow \exists x (P1(x)), \forall x (P1(x))$ là công thức

61

Tính an toàn của các biểu thức

- Đ/n: một biểu thức $\{ \langle x_1, \dots, x_n \rangle \mid P(x_1, \dots, x_n) \}$ là an toàn nếu tất cả các giá trị xuất hiện trong kết quả là các giá trị từ $\text{DOM}(P)$

62

Bài tập

- Biểu diễn câu hỏi bằng ngôn ngữ tính toán vị từ biến miền

63

Nhận xét

- Sự tương đương của 3 ngôn ngữ
 - Đại số quan hệ
 - Phép tính vị từ biến bộ hạn chế với các biểu thức an toàn
 - Phép tính vị từ biến miền hạn chế với các biểu thức an toàn
- So sánh đặc điểm của 3 ngôn ngữ

64

Khả năng bổ sung của các ngôn ngữ

- Tính toán số học: các phép toán số học $+, -, *, /$
- Lệnh gán và hiển thị: hiển thị quan hệ kết quả hay gán một quan hệ đã được tính toán đến một tên quan hệ khác.
- Hàm tập hợp: tính giá trị trung bình, tính tổng, chọn giá trị nhỏ nhất hay lớn nhất

65

Bài tập biến đổi tương đương

- Viết định nghĩa các phép toán ĐSQH với các biểu thức tính toán vị từ
- Tìm biểu thức tương đương trong ngôn ngữ ĐSQH và ngôn ngữ tính toán vị từ

66



67

Ngôn ngữ QBE

68

QBE (Query-By-Example)

- Là một ngôn ngữ truy vấn dữ liệu 2 chiều dựa trên khái niệm bảng khung
- 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: IBMTM (IBM Query Management Facility), Paradox, MS. Access, ...

69

Truy vấn trên một quan hệ

- P. ~ Print

Student	ID	Name	Suburb
		P._x	Bundoora

- Phép tính vị từ biến miền tương đương:
 $\{ \langle x \rangle \mid \exists i, x, s (i, x, s) \in \text{Student} \wedge s = \text{"Bundoora"} \}$
- Biểu thức đại số quan hệ tương đương

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

70

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

71

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	Enrol	SID	Course
	_id	P._name			_id	

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

Student	ID	Name	Suburb	Enrol	SID	Course
	_id	P._name		\neg	_id	

72

Các tính toán tập hợp

- Các phép toán: AVG, CNT, 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
	P.CNT.ALL_id		P. G.

- G. ~ Grouping

73

Hộp điều kiện

- Được sử dụng để biểu diễn các ràng buộc tổng quát trên các biến miền
- Ví dụ: đưa ra danh sách các thành phố không phải Bundoora có ít nhất 2 sinh viên

Student	ID	Name	Suburb	Conditions
	_id ¬_id		P._x _x	$x \neg = \text{Bundoora}$

74

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

75

Tính đầy đủ của QBE

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

76

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

77

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

78

Ngôn ngữ SQL

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



80

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, xóa, 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

81

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

82

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

83

Cú pháp

- Tạo bảng


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


```
DROP TABLE tab
```

84

Tạo bảng - Ví dụ:

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

85

Tạo bảng - Ví dụ (tiếp)

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

86

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);

87

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> <kiểu dữ liệu mới>;

88

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);

89

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>

90

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>]

91

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

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

Name
Robert
Glen
Norman
Mary

92

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;`

93

Truy vấn có điều kiện trên 1 bảng

- Chọn các dòng thỏa mãn điều kiện

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

- Ví dụ

SELECT *
FROM Student
WHERE suburb='Bundoora';

Student

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

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

Id	Name	Suburb
3936	Glen	Bundoora
8507	Norman	Bundoora

94

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;`

95

Biểu diễn điều kiện chọn trên 1 bảng

- Các phép toán so sánh: =, !=, <, >, <=, >=
- 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)

96

Biểu diễn điều kiện 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 SupplierProduct
WHERE pid = 'P1' OR pid = 'P2';
```
 - Cách 2:

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

97

Biểu diễn điều kiện 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%';
```

98

Truy vấn phức tạp trên nhiều bảng

- Cú pháp tổng quát:


```
SELECT T1.C1,T1.C2,T2.C1,T2.C4, ...
FROM T1, T2,.....
WHERE complex_condition_expression
```
- `complex_condition_expression` thông thường chứa cả `selection_condition_expression` và `join_condition_expression`
- 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
```

99

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';
```

100

Loại bỏ các dòng 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
```

101

Tìm kiếm có sắp xếp

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

```
SELECT <bt1>, <bt2>, ...
FROM <bang1>,<bang2>, ...
[WHERE <Điều kiện tìm kiếm>]
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
```

102

Tìm kiếm nhờ mệnh đề chia nhóm GROUP BY

- Các dòng thỏa mãn <Điều kiện tìm kiếm> được phân thành các nhóm theo giá trị của 1 hoặc nhiều cột chia nhóm

```
SELECT <tt1>, <tt2>, ..., <bt1>, <bt2>, ...
FROM <bang1>,<bang2>, ...
[WHERE <Điều kiện tìm kiếm>]
GROUP BY <tt1>, <tt2>, ...
[HAVING <điều kiện đại với nhóm>]
```

- Các cột được chỉ ra trong mệnh đề GroupBy được sử dụng làm cơ sở để chia nhóm. Các cột này bắt buộc phải được chỉ ra trong mệnh đề Select

- <điều kiện đại với nhóm> được kiểm tra với mỗi nhóm sau khi đã chia nhóm
- Ví dụ: với mỗi thành phố, cho biết số sinh viên ở thành phố này

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

103

Tìm kiếm nhờ mệnh đề chia nhóm GROUP BY (tiếp)

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

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

104

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
```

105

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**

106

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');
```

107

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');
```

108

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

109

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);
```

110

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 HãngOParis, size AS SoNhanVien
FROM Supplier
```

```
WHERE city = 'Paris';
```

```
SELECT SID, Stud.Name as $Name,
Sub.Name as Subject
```

```
FROM Student as Stud,Takes,
```

```
Subject as Sub
```

```
WHERE (Id=SID) and (SNO = No)
```

111

Các hàm thư viện

- Hàm tính toán trên nhóm các dòng

- MAX/MIN
- SUM
- AVG
- COUNT

- Hàm tính toán trên các cột của dòng

- 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

112

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';
```

113

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
```

114

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')
```

115

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";
```

116

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;`

117

Các điểm cần lưu ý

- Các ngôn ngữ với mô hình quan hệ
 - ĐSQH vs. vị từ
 - QBE vs. SQL
- Sự tương đương của các ngôn ngữ
 - Ngôn ngữ ĐSQH và ngôn ngữ vị từ
 - Biến đổi giữa câu truy vấn SQL và biểu thức đại số quan hệ

118



119