


UNIVERSITÀ DEGLI STUDI DI SALERNO

BASI DI DATI
.....
SQL-DML Avanzato - terza parte




DIPARTIMENTO DI
INFORMATICA, ELETTRONICA E MATEMATICA APPLICATA

Know MIS
Giuseppe D'Aniello
Researcher, PHD – Member IEEE

Atzeni, Ceri, Fraternali, Paraboschi, Torlone
Basi di dati *Quinta edizione*
McGraw-Hill Education, 2018

1

SINTASSI COMPLETA

SELECT (lista espressioni)
FROM (lista tabelle)
WHERE (condizioni)
GROUP BY (lista attributi)
HAVING (condizioni)
ORDER BY (lista attributi);

2

Semantica delle espressioni “correlate”

- ❖ L'interrogazione interna viene eseguita una volta per ciascuna ennupla della query esterna

3

Esempio Interrogazioni nidificate

- Nome e reddito dei padri di persone che guadagnano più di 20

```

select distinct P.Nome, P.Reddito
from Persone P, Paternita, Persone F
where P.Nome = Padre and Figlio = F.Nome
and F.Reddito > 20


```

```

select Nome, Reddito
from Persone
where Nome in (select Padre
from Paternita
where Figlio = any (select Nome
from Persone
where Reddito > 20))

```

notare la **distinct**



SELECT Tabella di riferimento			
Maternità		Paternità	
Madre	Figlio	Padre	Figlio
Luisa	Maria	Luisa	Maria
Anna	Olga	Anna	Olga
Maria	Andre	Maria	Andre
Maria	Aldo	Maria	Aldo
Padre	Figlio	Padre	Figlio
Sergio	Franco	Sergio	Franco
Luisa	Olga	Luisa	Olga
Luisa	Filippo	Luisa	Filippo
Franco	Andre	Franco	Andre
Franco	Aldo	Franco	Aldo

- Nome e reddito dei padri di persone che guadagnano più di 20

```
select distinct P.Nome, P.Reddito
from Persone P, Paternita, Persone F
where P.Nome = Padre and Figlio = F.Nome
and F.Reddito > 20
```

```
select Nome, Reddito
from Persone
where Nome in (select Padre
               from Paternita
               where Figlio = any (select Nome
                                   from Persone
                                   where Reddito > 20))
```

notare la **distinct**

SELECT Tabella di riferimento

Maternità	Madre		Persone		
	Nome	Età	Reddito		
Maternità	Luisa	Luigi	Andrea	27	21
	Anna	Ogla	Alido	25	15
	Anna	Filippo	Maria	55	42
	Maria	Andre	Maria	50	35
	Maria	Alido	Filippo	26	30
	Padre	Figlio	Luigi	50	40
Paternità	Sergio	Franco	Luigi	60	20
	Luigi	Ogla	Ogla	30	41
	Luigi	Filippo	Sergio	85	35
	Franco	Andre	Sergio	85	35
	Franco	Alido	Luisa	75	87

4

Esempio Interrogazioni nidificate

- Nome e reddito dei padri di persone che guadagnano più di 20, **con indicazione del reddito del figlio**

```
select distinct P.Nome, P.Reddito, F.Reddito
from Persone P, Paternita, Persone F
where P.Nome = Padre and Figlio = F.Nome
and F.Reddito > 20
```

```
select Nome, Reddito, ????
from Persone
where Nome in (select Padre
               from Paternita
               where Figlio = any (select Nome
                                   from Persone
                                   where Reddito > 20))
```

SELECT Tabelle di riferimento

Maternità	Madre	Figlio	Persone
	Luisa	Maria	Nome Età Reddito
	Luisa	Luigi	Andrea 27 21
	Anna	Olga	Aldo 25 15
	Anna	Filippo	Maria 55 42
	Maria	Andre	Anna 50 35
	Maria	Aldo	Filippo 26 30
Paternità	Padre	Figlio	Luigi 50 40
	Sergio	Franco	Franco 60 20
	Luigi	Olga	Olga 30 41
	Luigi	Filippo	Sergio 85 35
	Franco	Andre	Luisa 75 87
	Franco	Aldo	

5

Interrogazioni nidificate

- regole di visibilità:
 - non è possibile fare riferimenti a variabili definite in blocchi più interni
 - se un nome di variabile è omesso, si assume riferimento alla variabile più “vicina”
- in un blocco si può fare riferimento a variabili definite in blocchi più esterni; la semantica base (prodotto cartesiano, selezione, proiezione) non funziona più, vedremo presto

6

Esempio Visibilità

- **scorretta:**

```
select *  
from Impiegato  
where Dipart in (select Nome  
                 from Dipartimento D1 (Stesso Livello)  
                 where Nome = 'Produzione') or  
Dipart in (select Nome  
           from Dipartimento D2  
           where D2.Citta = D1.Citta) (Stesso Livello)
```

7

Quantificazione esistenziale

- Ulteriore tipo di condizione
 - **EXISTS** (Sottoespressione)

8

Esempio Quantificazione esistenziale

- I padri i cui figli guadagnano tutti più di 20

```
select distinct Padre
from Paternita Z
where not exists (
  select *
  from Paternita W, Persone
  where W.Padre = Z.Padre
        and W.Figlio = Nome
        and Reddito <= 20)
```

SELECT Tabelle di riferimento

Maternità	Madre	Figlio	Nome	Età	Reddito
	Luisa	Maria			
	Luisa	Luigi	Andrea	27	21
	Anna	Olga	Aldo	25	15
	Anna	Filippo	Maria	55	42
	Maria	Andre	Anna	50	35
	Maria	Aldo	Anna	50	35
Paternità	Padre	Figlio	Filippo	26	30
	Sergio	Franco	Luigi	50	40
	Luigi	Olga	Franco	60	20
	Luigi	Filippo	Olga	30	41
	Franco	Andre	Sergio	85	35
	Franco	Aldo	Luisa	75	87

padre
character varying (50)
1 Luigi

9

Esempio Quantificazione esistenziale

- I padri i cui figli guadagnano tutti più di 20

```
select distinct Padre
from Paternita
where not exists (
  select *
  from Persone
  where Figlio = Nome
        and Reddito <= 20)
```

NO!!!

SELECT Tabelle di riferimento

Maternità	Madre	Figlio	Nome	Età	Reddito
	Luisa	Maria			
	Luisa	Luigi	Andrea	27	21
	Anna	Olga	Aldo	25	15
	Anna	Filippo	Maria	55	42
	Maria	Andre	Anna	50	35
	Maria	Aldo	Anna	50	35
Paternità	Padre	Figlio	Filippo	26	30
	Sergio	Franco	Luigi	50	40
	Luigi	Olga	Franco	60	20
	Luigi	Filippo	Olga	30	41
	Franco	Andre	Sergio	85	35
	Franco	Aldo	Luisa	75	87

padre
character varying (50)
1 Franco
2 Luigi

10

Disgiunzione e unione (ma non sempre)

```
select * from Persone where Reddito > 30
union
select F.*
from Persone F, Paternita, Persone P
where F.Nome = Figlio and Padre = P.Nome
and P.Reddito > 30
```

```
select *
from Persone F
where Reddito > 30 or
exists (select *
        from Paternita, Persone P
        where F.Nome = Figlio and Padre = P.Nome
        and P.Reddito > 30)
```

11

Esercizio

```
-- =====
-- DROP delle tabelle FORNITORI, PRODOTTI e CATALOGO
-- =====

DROP TABLE IF EXISTS Fornitori CASCADE;
DROP TABLE IF EXISTS Prodotti CASCADE;
DROP TABLE IF EXISTS Catalogo CASCADE;
-- =====

-- Creazione delle tabelle
CREATE TABLE Fornitori( fid          CHAR(2) PRIMARY KEY,
  nome          CHAR(20),
  indirizzo     CHAR(20) );

CREATE TABLE Prodotti( pid          CHAR(3) PRIMARY KEY,
  nome          CHAR(20),
  colore        CHAR(20) );
CREATE TABLE Catalogo( fid          CHAR(2),
  pid           CHAR(3),
  costo         REAL,
  FOREIGN KEY (fid) REFERENCES FORNITORI(fid),
  FOREIGN KEY (pid) REFERENCES PRODOTTI(pid),
  PRIMARY KEY(fid,pid) );
```

12

Esercizio				Fornitori, 5 Righe			
Prodotti, 16 Righe							
	pid [PK] character (3)	nome character (20)	colore character (20)		fid [PK] character (2)	nome character (20)	indirizzo character (20)
1	P1	Volante	Nero	1	F1	ACME	via Hollywood
2	P2	Volante	Rosso	2	F2	Ingegneria	via Eudossiana
3	P3	Carrozzeria	Nero	3	F3	Sapienza	via Scarpa
4	P4	Carrozzeria	Rosso	4	F4	DIS	via Ariosto
5	P5	Carrozzeria	Verde	5	F5	Gest	via Buonarroti
6	P6	Cerchione	Nero				
7	P7	Cerchione	Rosso				
8	P8	Ruota	Nero				
9	P9	Sedile	Nero				
10	P10	Sedile	Rosso				
11	P11	Sedile	Verde				
12	P12	Tappetino	Nero				
13	P13	Tappetino	Rosso				
14	P14	Tappetino	Verde				
15	P15	Casco	Rosso				
16	P16	Casco	Verde				
Script "negozio.sql"							

	fid [PK] character (2)	pid [PK] character (3)	costo real
1	F1	P1	100
2	F1	P2	100
3	F1	P3	500
4	F1	P4	500
5	F1	P5	500
6	F1	P6	70
7	F1	P7	70
8	F1	P8	180
9	F1	P9	220
10	F1	P10	220
11	F1	P11	220
12	F1	P12	50
13	F1	P13	50
14	F1	P14	50
15	F1	P15	90
16	F1	P16	90
17	F2	P2	120
18	F2	P3	550
19	F2	P4	550
20	F2	P5	550

Catalogo
32 Righe

13

SQL DML – Query			
Prodotti			Fornitori
pid [PK] character (3)	nome character (20)	colore character (20)	fid [PK] character (2)
1 P1	Volante	Nero	1 F1
2 P2	Volante	Rosso	2 F2
3 P3	Carrozzeria	Nero	3 F3

nome character (20)	indirizzo character (20)
ACME	via Hollywood
Ingegneria	via Eudossiana
Sapienza	via Scarpa

Catalogo		
fid [PK] character (2)	pid [PK] character (3)	costo real
1 F1	P1	100
2 F1	P2	100
3 F1	P3	500
4 F1	P4	500

Trovare i prodotti forniti dalla ACME e da nessun altro, senza fare uso di Differenza insiemistica: utilizzare un approccio con uso di Exist o Not Exist

14

SQL DML – Query Nidificate – Costruzione – Fase 1

Prodotti

	pid [PK] character (3)	nome character (20)	colore character (20)
1	P1	Volante	Nero
2	P2	Volante	Rosso
3	P3	Carrozzeria	Nero

Fornitori

	fid [PK] character (2)	nome character (20)	indirizzo character (20)
1	F1	ACME	via Hollywood
2	F2	Ingegneria	via Eudossiana
3	F3	Sapienza	via Scarpa

Catalogo

	fid [PK] character (2)	pid [PK] character (3)	costo real
1	F1	P1	100
2	F1	P2	100
3	F1	P3	500
4	F1	P4	500

Trovare i prodotto forniti dalla ACME e da nessun altro – Come la costruisco?
Accorgimenti per non incappare in errori
Trova tutti i prodotti forniti da ACME

```
SELECT P.pid, P.nome, P.colore
FROM
    Prodotti P,
    Catalogo C,
    Fornitori F
WHERE P.pid=C.pid
    AND C.fid=F.fid
    AND F.nome = 'ACME'
```

Questa query (che se lanciata, va in errore – non avendo la Tabella Prodotti P) trova i fornitori diversi da ACME che vendono lo stesso prodotto P.PID. Il risultato non è vuoto se il prodotto P.pid è venduto da un fornitore diverso da ACME !!!

```
SELECT *
FROM
    Catalogo C1,
    Fornitori F1
WHERE P.pid = C1.pid
    AND C1.fid=F1.fid
    AND F1.nome <> 'ACME'
```

15

SQL DML – Query Nidificate – Costruzione – Fase 2

Trovare i prodotto forniti dalla ACME e da nessun altro

COME METTO INSIEME LE DUE QUERY ?

Trova tutti i prodotti forniti da ACME

```
SELECT P.pid, P.nome, P.colore
FROM
    Prodotti P,
    Catalogo C,
    Fornitori F
WHERE P.pid=C.pid
    AND C.fid=F.fid
    AND F.nome = 'ACME'
```

Questa query trova i fornitori diversi da ACME che vendono lo stesso prodotto P.PID. Il risultato non è vuoto se il prodotto P.pid è venduto da un fornitore diverso da ACME

```
SELECT *
FROM
    Catalogo C1,
    Fornitori F1
WHERE P.pid = C1.pid
    AND C1.fid=F1.fid
    AND F1.nome <> 'ACME'
```

16

SQL DML – Query Nidificate - Soluzione

Prodotti

	pid [PK] character (3)	nome character (20)	colore character (20)
1	P1	Volante	Nero
2	P2	Volante	Rosso
3	P3	Carrozzeria	Nero

Fornitori

	fid [PK] character (2)	nome character (20)	indirizzo character (20)
1	F1	ACME	via Hollywood
2	F2	Ingegneria	via Eudossiana
3	F3	Sapienza	via Scarpa

Catalogo

	fid [PK] character (2)	pid [PK] character (3)	costo real
1	F1	P1	100
2	F1	P2	100
3	F1	P3	500
4	F1	P4	500

Trovare i prodotti forniti dalla ACME e da nessun altro

```
SELECT P.pid, P.nome, P.colore
FROM Prodotti P, Catalogo C, Fornitori F
WHERE P.pid=C.pid
      AND C.fid=F.fid
      AND F.nome = 'ACME'
      AND NOT EXISTS (
        SELECT *
        FROM Catalogo C1, Fornitori F1
        WHERE P.pid = C1.pid
              AND C1.fid=F1.fid
              AND F1.nome <> 'ACME' )
```

Data Output	Explain	Messages	Notifications
pid [PK] character (3)	nome character (20)	colore character (20)	
1 P6	Cerchione	Nero	
2 P9	Sedile	Nero	
3 P11	Sedile	Verde	
4 P16	Casco	Verde	

Questa query trova i fornitori diversi da ACME che vendono lo stesso prodotto P.PID. Il risultato non è vuoto se il prodotto P.pid è venduto da un fornitore diverso da ACME

17

SQL DML – Query Nidificate

Prodotti

	pid [PK] character (3)	nome character (20)	colore character (20)
1	P1	Volante	Nero
2	P2	Volante	Rosso
3	P3	Carrozzeria	Nero

Fornitori

	fid [PK] character (2)	nome character (20)	indirizzo character (20)
1	F1	ACME	via Hollywood
2	F2	Ingegneria	via Eudossiana
3	F3	Sapienza	via Scarpa

Catalogo

	fid [PK] character (2)	pid [PK] character (3)	costo real
1	F1	P1	100
2	F1	P2	100
3	F1	P3	500
4	F1	P4	500

Trovare i nomi dei fornitori che forniscono ogni prodotto

18

SQL DML – Query Nidificate

Prodotti

	pid [PK] character (3)	nome character (20)	colore character (20)
1	P1	Volante	Nero
2	P2	Volante	Rosso
3	P3	Carrozzeria	Nero

Fornitori

	fid [PK] character (2)	nome character (20)	indirizzo character (20)
1	F1	ACME	via Hollywood
2	F2	Ingegneria	via Eudossiana
3	F3	Sapienza	via Scarpa

Catalogo

	fid [PK] character (2)	pid [PK] character (3)	costo real
1	F1	P1	100
2	F1	P2	100
3	F1	P3	500
4	F1	P4	500

Trovare i nomi dei fornitori che forniscono ogni prodotto

Idea: dall'elenco di tutti i fornitori non considero (NOT EXISTS) i fornitori che non forniscono tutti i prodotti

```
SELECT DISTINCT F.nome
FROM Fornitori F
WHERE NOT EXISTS (
    -- Trovare i fornitori che non forniscono tutti i prodotti
)
```

19

SQL DML – Query Nidificate

Prodotti

	pid [PK] character (3)	nome character (20)	colore character (20)
1	P1	Volante	Nero
2	P2	Volante	Rosso
3	P3	Carrozzeria	Nero

Fornitori

	fid [PK] character (2)	nome character (20)	indirizzo character (20)
1	F1	ACME	via Hollywood
2	F2	Ingegneria	via Eudossiana
3	F3	Sapienza	via Scarpa

Catalogo

	fid [PK] character (2)	pid [PK] character (3)	costo real
1	F1	P1	100
2	F1	P2	100
3	F1	P3	500
4	F1	P4	500

Trovare i nomi dei fornitori che forniscono ogni prodotto

Idea: dall'elenco di tutti i fornitori non considero (NOT EXISTS) i fornitori che non forniscono tutti i prodotti.

Quindi per trovare i fornitori che forniscono tutti i prodotti, parto dalla lista di tutti i prodotti e tolgo i prodotti forniti dal fornitore considerato (nella query esterna)

```
SELECT DISTINCT F.nome
FROM Fornitori F
WHERE NOT EXISTS (
    -- Trovare i fornitori che non forniscono tutti i prodotti
    SELECT P.pid FROM Prodotti P
    ..... – come li trovo ? )
```

20

SQL DML – Query Nidificate

Prodotti

	pid [PK] character (3)	nome character (20)	colore character (20)
1	P1	Volante	Nero
2	P2	Volante	Rosso
3	P3	Carrozzeria	Nero

Fornitori

	fid [PK] character (2)	nome character (20)	indirizzo character (20)
1	F1	ACME	via Hollywood
2	F2	Ingegneria	via Eudossiana
3	F3	Sapienza	via Scarpa

Catalogo

	fid [PK] character (2)	pid [PK] character (3)	costo real
1	F1	P1	100
2	F1	P2	100
3	F1	P3	500
4	F1	P4	500

Trovare i nomi dei fornitori che forniscono ogni prodotto

Per trovare i fornitori che forniscono tutti i prodotti, parto dalla lista di tutti i prodotti e tolgo i prodotti forniti dal fornitore considerato (nella query esterna)

La query più interna trova i prodotti forniti dal fornitore.

SELECT DISTINCT F.nome

FROM Fornitori F

WHERE NOT EXISTS (

-- Trovare i fornitori che non forniscono tutti i prodotti

SELECT P.pid FROM Prodotti P

WHERE P.pid NOT IN

(SELECT C.pid
FROM Catalogo C
WHERE C.fid = F.fid)
)

Verifico che un P.pid di
un prodotto non è
presente tra tutti quelli
che il fornitore fornisce
IN QUESTO CASO LO
INSERISCO NELLA
PROIEZIONE

21

BASI DI DATI

Materiale utilizzato e bibliografia

➤ Le slide utilizzate dai docenti per le attività frontali sono in gran parte riconducibili e riprese dalle slide originali (con alcuni spunti parziali ripresi dai libri indicati) realizzate da:

✓ autori del libro Basi di Dati (Atzeni e altri) testo di riferimento del corso Basi di Dati e sono reperibili su internet su molteplici link oltre che laddove indicato dagli stessi autori del libro;

✓ Prof.ssa Tiziana Catarci e dal dott. Ing. Francesco Leotta – corso di Basi di Dati dell'Università degli Studi La Sapienza di Roma al seguente link ed altri: <http://www.dis.uniroma1.it/~catarci/basidatGEST.html> (molto Interessanti anche le lezioni su YouTube).

✓ Prof. Luca Allulli e Umberto Nanni, Libro Fondamenti di basi di dati, editore HOEPLI (testo di facile lettura ed efficace).

➤ Diverse slide su specifici argomenti utilizzate dai docenti per le attività frontali sono anche in parte riconducibili e riprese dalle slide originali – facilmente reperibili e accessibili su internet - realizzate da:

Prof.ssa Roberta Aiello – corso Basi di Dati dell'Università di Salerno

Prof. Dario Maio - corso Basi di Dati dell'Università di Bologna al seguente link ed altri: <http://bias.csr.unibo.it/maio>

Prof. Marco Di Felice - corso Basi di Dati dell'Università di Bologna al seguente link ed altri: <http://www.cs.unibo.it/difelice/dbsi/>

Prof. Marco Maggini e prof. Franco Scarselli - corso Basi di Dati dell'Università di Siena al seguenti link ed altri: [http://staff.icar.cnr.it/pontieri/didattica/LabSI/lezioni/_preliminari-DB1%20\(Maggini\).pdf](http://staff.icar.cnr.it/pontieri/didattica/LabSI/lezioni/_preliminari-DB1%20(Maggini).pdf)

Prof. Fabio A. Schreiber - corso Basi di Dati del Politecnico di Milano al seguente link ed altri: <https://schreiber.faculty.polimi.it/BasidiDati0607/LucidTeoria/IntroduzioneCR.pdf>

Prof.ssa Raffaella Gentilini - corso Basi di Dati dell'Università di Perugia al seguente link ed altri: <http://www.dmi.unipg.it/raffaella.gentilini/BD.htm>

Prof. Enrico Giunchiglia - corso Basi di Dati dell'Università di Genova al seguente link ed altri: <http://www.star.dist.unige.it/~enrico/BasidiDati/>

Prof. Maurizio Lenzerini - corso Basi di Dati dell'Università degli Studi La Sapienza di Roma al seguente link ed altri: <http://didattica.info.altervista.org/Quinta/Database2.pdf>

Prof.ssa Claudia D'Amato - corso Basi di Dati dell'Università di Bari al seguente link ed altri: <http://www.di.uniba.it/~cdamato/>

Atzeni, Ceri, Fraternali, Paraboschi, Torlone
Basi di dati *Quinta edizione*
McGraw-Hill Education, 2018

22