SQL

Laboratorio di
Progettazione di Basi di Dati (CdS in Informatica e TPS)

a.a. 2014/2015

http://www.di.uniba.it/~lisi/courses/basi-dati/bd2014-15.htm

dott.ssa Francesca A. Lisi
francesca.lisi@uniba.it

Orario di ricevimento: giovedì ore 10-11

Sommario (IV parte)

- Interrogazioni complesse in SQL
  - Operatori aggregati
  - Interrogazioni con raggruppamento
  - Interrogazioni di tipo insiemistico
  - Interrogazioni nidificate

Riferimenti

- capp. 3-4 di Pratt
- capp. 6-8 di “MySQL Tutorial”
- cap. 8, in particolare 8.5, di Elmasri & Navathe
- cap. 4, in particolare 4.3.3-4.3.6, di Atzeni et al.

Maternità

Madre	Figlio
Luisa	Maria
Luisa	Luigi
Anna	Olga
Anna	Filippo
Maria	Andrea
Maria	Aldo

Paternità

Padre	Figlio
Sergio	Franco
Luigi	Olga
Luigi	Filippo
Franco	Andrea
Franco	Aldo

Persone

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

Operatori aggregati

- Nelle espressioni della target list possiamo avere anche espressioni che calcolano valori a partire da insiemi di ennuple:
  - conteggio (**`COUNT`**)
  - minimo (**`MIN`**)
  - massimo (**`MAX`**)
  - media (**`AVG`**)
  - totale (**`SUM`**)

Operatori aggregati (II)

Es. Contare il numero di figli di Franco

```
SELECT COUNT(*) AS NumFigliDiFranco
FROM Paternita
WHERE Padre = 'Franco'
```

2

- l'operatore aggregato viene applicato al risultato dell'interrogazione:

```
SELECT *
FROM Paternita
WHERE Padre = 'Franco'
```

Operatori aggregati (III)

Es. Trovare la media dei redditi dei figli di Franco

```
SELECT AVG(reddito)
FROM persone JOIN paternita ON
  nome=figlio
WHERE padre='Franco'
```

Operatori aggregati (IV)

```
SELECT eta, MAX(reddito)
FROM persone
```

- La target list deve essere omogenea!

```
SELECT MIN(eta), MAX(reddito)
FROM persone
```

Operatori aggregati e valori nulli

Persone

Nome	Età	Reddito
Andrea	27	21
Aldo	25	NULL
Maria	55	21
Anna	50	35

```
SELECT COUNT(*) FROM persone
   4
SELECT COUNT(reddito) FROM persone
   3
SELECT COUNT(DISTINCT reddito) FROM persone
  3
```

Operatori aggregati e valori nulli (II)

```
SELECT AVG(reddito)
FROM persone
```

25,6

```
SELECT SUM(reddito)/COUNT(*)
FROM persone
```

19,25

Raggruppamenti

- Gli operatori aggregati possono essere applicati a partizioni di una relazione (ottenute mediante la clausola **`GROUP BY`**)

- Come si risponde a questo tipo di interrogazioni?

1. Si esegue l'interrogazione *trascurando momentaneamente* raggruppamenti ed operatori aggregati
2. Si formano i gruppi nella relazione risultato secondo la condizione di raggruppamento e si applica l'operatore aggregato a ciascun gruppo

Raggruppamenti (II)

Es. Trovare il numero di figli di ciascun padre

```
SELECT padre, COUNT(*) AS NumFigli
FROM paternita
GROUP BY padre
```

Paternita

Padre	Figlio
Sergio	Franco
Luigi	Olga
Luigi	Filippo
Franco	Andrea
Franco	Aldo

Padre	NumFigli
Sergio	1
Luigi	2
Franco	2

Raggruppamenti (III)

Es. Estrarre i padri i cui figli hanno un reddito medio maggiore di 25

```
SELECT padre, AVG(f.reddito)
FROM persone f
  JOIN paternita ON figlio = nome
GROUP BY padre
HAVING AVG(f.reddito)> 25
```

Raggruppamenti (IV)

Es. Estrarre i padri i cui figli sotto i 30 anni hanno un reddito medio maggiore di 25

```
SELECT padre, AVG(f.reddito)
FROM persone f
  JOIN paternita ON figlio = nome
WHERE eta < 30
GROUP BY padre
HAVING AVG(f.reddito)> 25
```

Raggruppamenti (V)

```
SELECT padre, AVG(f.reddito), p.reddito
FROM persone f JOIN
  paternita ON figlio = nome JOIN
  persone p ON padre = p.nome
GROUP BY padre
```

```
SELECT padre, AVG(f.reddito), p.reddito
FROM persone f JOIN
  paternita ON figlio = nome JOIN
  persone p ON padre = p.nome
GROUP BY padre, p.reddito
```

Sintassi, riassumiamo

SelectSQL ::=

SELECT *ListaAttributiOEspressioni*

FROM *ListaTabelle*

[**WHERE** *CondizioniSemplici*]

[**GROUP BY** *ListaAttributiDiRaggruppamento*]

[**HAVING** *CondizioniAggregate*]

[**ORDER BY** *ListaAttributiDiOrdinamento*]

Unione

```
SELECT padre
FROM paternita
     UNION
SELECT madre
FROM maternita
```

```
Sergio
Luigi
Franco
Luisa
Anna
Maria
```

- quali nomi per gli attributi del risultato?
  - nessuno
  - quelli del primo operando
  - …

Unione (II)

```
SELECT padre, figlio
FROM paternita
     UNION
SELECT madre, figlio
FROM maternita
```

```
SELECT padre, figlio
FROM paternita
          UNION
SELECT figlio, madre
FROM maternita
```

Padre	Figlio
Sergio	Franco
Luigi	Olga
Luigi	Filippo
Franco	Andrea
Franco	Aldo
Luisa	Maria
Luisa	Luigi
Anna	Olga
Anna	Filippo
Maria	Andrea
Maria	Aldo

Unione (III)

```
SELECT padre AS genitore, figlio
FROM paternita
                UNION
SELECT figlio, madre AS genitore
FROM maternita
```

```
SELECT padre AS genitore, figlio
FROM paternita
                UNION
SELECT madre AS genitore, figlio
FROM maternita
```

Unione (IV)

Es. Estrarre le persone che o hanno esse stesse un reddito superiore ai 30 milioni o sono padri di figli con un reddito superiore ai 30 milioni

```
SELECT *
FROM Persone
WHERE Reddito > 30
  UNION
SELECT P.*
FROM Persone F, Paternita, Persone P
WHERE F.Nome = Figlio AND
  Padre = P.Nome AND F.Reddito > 30
```

Differenza

Es. Estrarre tutti i nomi che non siano cognomi

```
SELECT Nome
FROM Impiegato
     EXCEPT
SELECT Cognome AS Nome
FROM Impiegato
```

Intersezione

Es. Estrarre tutti i nomi che siano anche cognomi

```
SELECT Nome
FROM Impiegato
     INTERSECT
SELECT Cognome AS Nome
FROM Impiegato
```

- equivale a

```
SELECT I.Nome
FROM Impiegato I, Impiegato J
WHERE I.Nome = J.Cognome
```

Interrogazioni nidificate

- Le condizioni atomiche permettono anche
  - il confronto fra un attributo (o più, vedremo poi) e il risultato di una sotto-interrogazione
  - quantificazioni esistenziali
- La forma nidificata è "meno dichiarativa", ma talvolta più leggibile (richiede meno variabili)
- La forma piana e quella nidificata possono essere combinate
- Le sotto-interrogazioni non possono contenere operatori insiemistici (ma tale limitazione non è significativa)

Interrogazioni nidificate (II)

Es. Estrarre nome e reddito del padre di Franco

```
SELECT Nome, Reddito
FROM Persone, Paternita
WHERE Nome=Padre AND Figlio = 'Franco'
```

```
SELECT Nome, Reddito
FROM Persone
WHERE Nome = (SELECT Padre
              FROM Paternita
              WHERE Figlio = 'Franco')
```

Interrogazioni nidificate (III)

Es. Estrarre le persone con il reddito massimo

```
SELECT *
FROM persone
WHERE reddito =
      (SELECT MAX(reddito)
       FROM persone)
```

```
SELECT *
FROM persone
WHERE reddito >= ALL
      (SELECT reddito
       FROM persone)
```

soluzioni equivalenti

Interrogazioni nidificate (IV)

Es. Estrarre nome e reddito dei padri di persone che guadagnano più di 20 milioni

```
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))
```

Interrogazioni nidificate (V)

Es. Estrarre nome e reddito dei padri di persone che guadagnano più di 20 milioni

```
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, Persone
   WHERE Figlio = Nome AND Reddito > 20)
```

Interrogazioni nidificate (VI)

Es. Estrarre nome e reddito dei padri di persone che guadagnano più di 20 milioni, 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))
```

- regole di visibilità:
  - si può fare riferimento *solo* a variabili definite in blocchi più esterni
  - se un nome di variabile è omesso, si assume riferimento alla variabile più "vicina"

Interrogazioni nidificate (VII)

```
SELECT *
FROM Impiegato
WHERE Dipart IN
   (SELECT Nome
   FROM Dipartimento D1
   WHERE Nome = 'Produzione') OR
         Dipart IN
   (SELECT Nome
   FROM Dipartimento D2
   WHERE D2.Citta = D1.Citta)
```

Interrogazioni nidificate (VII)

Es. Estrarre le persone che hanno almeno un figlio

```
SELECT *
FROM Persone
WHERE EXISTS (SELECT *
              FROM Paternita
              WHERE Padre = Nome) OR
      EXISTS (SELECT *
              FROM Maternita
              WHERE Madre = Nome)
```

Interrogazioni nidificate (VIII)

Es. Estrarre i padri i cui figli guadagnano tutti più di venti milioni

```
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)
```

Interrogazioni nidificate (IX)

Es. Estrarre le persone che o hanno esse stesse un reddito superiore ai 30 milioni o sono padri di figli con un reddito superiore ai 30 milioni

```
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)
```

Interrogazioni nidificate (X)

Es. Estrarre tutti i nomi che non siano anche cognomi

```
SELECT Nome
FROM Impiegato I
WHERE NOT EXISTS
          (SELECT *
          FROM Impiegato
          WHERE Cognome = I.Nome)
```

Esercitazione con MySQL

- Esercizi 10, 14-22 sul database Prodotti Premiere da cap. 3., pagg. 65-66, di Pratt;
- Esercizi 4-19 sul database Prodotti Premiere da cap. 4., pag. 91-2 di Pratt;

- N.B. MySQL 5.5 supporta tutti gli operatori aggregati, le interrogazioni nidificate ma solo **`UNION`** fra gli operatori insiemistici. In particolare, un **`FULL OUTER JOIN`** (non supportato in MySQL 5.5) si esprime come **`UNION`** di un **`LEFT OUTER JOIN`** e di un **`RIGHT OUTER JOIN`**.