SQL - Structured Query Language

Alessandro Lori

Università di Pisa

16 marzo 2012

Query su più tabelle

Sintassi **SELECT**

```
SELECT Attributo1, Attributo2, ...
FROM Tabella1, Tabella2, ...
```

- Cosa significa inserire più di una tabella nel **FROM**?
- La query viene eseguita sul *prodotto cartesiano* delle tabelle
- Nel caso di due tabelle, ogni riga della prima tabella viene collegata ad ogni riga della seconda

Prodotto cartesiano

- Sia data la tabella a destra (Persone)
- *Esempio 1:* Costruire una query che produca come risultato tutte le coppie ordinate di persone. Per ogni coppia di persone visualizzare il nome di ciascuna delle persone della coppia

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

Esempio 1

Soluzione

```
SELECT P1.Nome, P2.Nome
FROM Persone AS P1,
     Persone AS P2
```

Nome	Nome
Aldo	Aldo
Andrea	Aldo
Filippo	Aldo
Gigi	Aldo
...	...

E se volessi ottenere solo le coppie non ordinate?

Esempio 1

Soluzione

```
SELECT P1.Nome, P2.Nome
FROM Persone AS P1,
     Persone AS P2
```

Nome	Nome
Aldo	Aldo
Andrea	Aldo
Filippo	Aldo
Gigi	Aldo
...	...

E se volessi ottenere solo le coppie non ordinate?

Soluzione

```
SELECT P1.Nome, P2.Nome
FROM Persone AS P1,
     Persone AS P2
WHERE P1.Nome < P2.Nome
```

Nome	Nome
Aldo	Andrea
Aldo	Anna
Andrea	Anna
Aldo	Filippo
...	...

Esercizio 1

Trovare tutte le persone che hanno la stessa età. Ogni riga del risultato dovrà mostrare una coppia di persone (distinte) che hanno la stessa età e la loro età.

Esercizio 1

Trovare tutte le persone che hanno la stessa età. Ogni riga del risultato dovrà mostrare una coppia di persone (distinte) che hanno la stessa età e la loro età.

Soluzione

```
SELECT P1.Nome, P2.Nome, P1.Eta
FROM Persone P1, Persone P2
WHERE P1.Nome < P2.Nome
AND   P1.Eta = P2.Eta
```

Nome	Nome	Eta
Filippo	Gigi	26
Anna	Luigi	50

Join

- Progettando le basi di dati in maniera oculata le informazioni vengono disperse su più tabelle.
- Nelle query si può aver necessità di ricostruire l'informazione dispersa utilizzando il **`JOIN`**
- Col **`JOIN`** si collegano solo le righe di tabelle diverse che hanno lo stesso valore su uno o più campi
- Si considerino le seguenti tabelle
  - Persone(Nome, Età, Reddito)
  - Maternità(Madre, Figlio)
  - Paternità(Padre, Figlio)

Esercizio 2

Costruire una query che produca come risultato i padri di persone che hanno un reddito superiore a 20

Esercizio 2

Costruire una query che produca come risultato i padri di persone che hanno un reddito superiore a 20

Soluzione

```
SELECT DISTINCT Padre
FROM Persone, Paternita
WHERE Figlio = Nome
AND Reddito > 20
```

Padre
Franco
Luigi

Esercizio 2 - Parola chiave `JOIN`

Costruire una query che produca come risultato i padri di persone che hanno un reddito superiore a 20

Esercizio 2 - Parola chiave `JOIN`

Costruire una query che produca come risultato i padri di persone che hanno un reddito superiore a 20

Soluzione

```
SELECT DISTINCT Padre
FROM Persone JOIN Paternita
ON Figlio=Nome
WHERE Reddito > 20
```

Padre
Franco
Luigi

Esercizio 3

Costruire una query che produca come risultato il padre e la madre di ogni persona

Esercizio 3

Costruire una query che produca come risultato il padre e la madre di ogni persona

Soluzione

```
SELECT Padre, Madre, P.Figlio
FROM Paternita P, Maternita M
WHERE P.Figlio = M.Figlio
```

Esercizio 3

Costruire una query che produca come risultato il padre e la madre di ogni persona

Soluzione

```
SELECT Padre, Madre, P.Figlio
FROM Paternita P, Maternita M
WHERE P.Figlio = M.Figlio
```

Soluzione alternativa

```
SELECT Padre, Madre, P.Figlio
FROM Paternita P JOIN Maternita M
ON P.Figlio = M.Figlio
```

Natural Join

Soluzione alternativa

```
SELECT Padre, Madre, P.Figlio
FROM Paternita P NATURAL JOIN Maternita
```

Il **NATURAL JOIN** effettua il join sugli attributi che hanno lo stesso nome

Esercizio 4

Considerare le persone che guadagnano più dei rispettivi padri. Costruire una query che produca come risultato nome del figlio, reddito del figlio e reddito del padre

Esercizio 4

Considerare le persone che guadagnano più dei rispettivi padri. Costruire una query che produca come risultato nome del figlio, reddito del figlio e reddito del padre

Soluzione

```
SELECT F.Nome, F.Reddito, P.Reddito AS RedditoPadre
FROM Persone P, Paternita Pa, Persone F
WHERE P.Nome = Pa.Padre AND F.Nome = Pa.Figlio
AND F.Reddito > P.Reddito
```

Esercizio 4

Considerare le persone che guadagnano più dei rispettivi padri. Costruire una query che produca come risultato nome del figlio, reddito del figlio e reddito del padre

Soluzione

```
SELECT F.Nome, F.Reddito, P.Reddito AS RedditoPadre
FROM Persone P, Paternita Pa, Persone F
WHERE P.Nome = Pa.Padre AND F.Nome = Pa.Figlio
AND F.Reddito > P.Reddito
```

Soluzione alternativa con JOIN

```
SELECT F.Nome, F.Reddito, P.Reddito AS RedditoPadre
FROM (Persone P JOIN Paternita Pa ON
P.Nome = Pa.Padre) JOIN Persone F ON
Pa.Figlio = F.Nome WHERE F.Reddito > P.Reddito
```

INNER JOIN - OUTER JOIN

- Di default viene usato l'**INNER JOIN**. Fanno parte del risultato solo le righe della prima e della seconda tabella che verificano la condizione
- Con l'**OUTER JOIN** si possono includere nel risultato anche le righe che non hanno una corrispondenza nell'altra tabella
  - **LEFT** [**OUTER**] **JOIN** Partecipano anche le righe della prima tabella che non hanno una corrispondenza. Gli attributi della seconda tabella sono fissati a **NULL**
  - **RIGHT** [**OUTER**] **JOIN** Partecipano anche le righe della seconda tabella che non hanno una corrispondenza. Gli attributi della prima tabella sono fissati a **NULL**
  - **FULL** [**OUTER**] **JOIN** Partecipano tutte le righe che non hanno una corrispondenza. Gli altri attributi sono fissati a **NULL**. Non supportato da `MySQL`

INNER JOIN - OUTER JOIN - Esempi

LEFT JOIN

```
SELECT *
FROM Maternita Ma LEFT JOIN Persone Pe
ON Ma.Figlio = Pe.Nome
```

INNER JOIN - OUTER JOIN - Esempi

LEFT JOIN

```
SELECT *
FROM Maternita Ma LEFT JOIN Persone Pe
ON Ma.Figlio = Pe.Nome
```

RIGHT JOIN

```
SELECT *
FROM Maternita Ma RIGHT JOIN Persone Pe
ON Ma.Figlio = Pe.Nome
```

Operatori aggregati

- Si possono usare degli operatori che effettuano calcoli sui valori di una colonna appartenenti a più righe
  - **COUNT** conta il numero di valori in un insieme
  - **MIN** calcola il minimo un insieme di valori
  - **MAX** calcola il massimo un insieme di valori
  - **AVG** calcola la media di un insieme di valori
  - **SUM** calcola la somma di un insieme di valori
- L'operatore aggregato **COUNT** supporta una lista di attributi (e quindi anche un attributo singolo)
- Gli operatori aggregati **MIN**, **MAX**, **SUM** e **AVG** supportano solo un singolo attributo

Operatori aggregati - Esempi

Calcolare la somma dei redditi delle persone con meno di 40 anni

Operatori aggregati - Esempi

Calcolare la somma dei redditi delle persone con meno di 40 anni

Soluzione

```
SELECT SUM(Reddito)
FROM Persone
WHERE Eta<40
```

Operatori aggregati - Esempi

Calcolare la somma dei redditi delle persone con meno di 40 anni

Soluzione

```
SELECT SUM(Reddito)
FROM Persone
WHERE Eta<40
```

Calcolare quante persone guadagnano fra 30 e 40

Operatori aggregati - Esempi

Calcolare la somma dei redditi delle persone con meno di 40 anni

Soluzione

```
SELECT SUM(Reddito)
FROM Persone
WHERE Eta<40
```

Calcolare quante persone guadagnano fra 30 e 40

Soluzione

```
SELECT COUNT(*)
FROM Persone
WHERE Reddito BETWEEN 30 AND 40
```

L'operatore COUNT

La query

```
SELECT COUNT(*)
FROM Persone
WHERE Reddito BETWEEN 30 AND 40
```

si ottiene applicando **COUNT** al risultato della query

```
SELECT *
FROM Persone
WHERE Reddito BETWEEN 30 AND 40
```

L'operatore **COUNT** e i valori **NULL**

- **SELECT COUNT**(∗) **FROM** Persone
  - Conta il numero di righe nella tabella
- **SELECT COUNT**(Reddito) **FROM** Persone
  - Conta il numero di righe con Reddito diverso da **NULL**
- **SELECT COUNT**(**DISTINCT** Reddito) **FROM** Persone
  - Conta il numero di valori diversi nella colonna Reddito
- **SELECT COUNT**(**ALL** Reddito) **FROM** Persone
  - Conta il numero di righe hanno un Reddito diverso da **NULL**. La parola chiave **ALL** può essere omessa

L'operatore `COUNT` e i valori `NULL` - Esempi

Data la tabella

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

- **SELECT COUNT**(∗) **FROM** Persone
- **SELECT COUNT**(Reddito) **FROM** Persone
- **SELECT COUNT**(**DISTINCT** Reddito) **FROM** Persone

L'operatore `COUNT` e i valori `NULL` - Esempi

Data la tabella

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

- **`SELECT COUNT`**$(*)$ **`FROM`** Persone **4**
- **`SELECT COUNT`**(Reddito) **`FROM`** Persone
- **`SELECT COUNT`**(**`DISTINCT`** Reddito) **`FROM`** Persone

L'operatore `COUNT` e i valori `NULL` - Esempi

Data la tabella

Nome	Eta	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

L'operatore `COUNT` e i valori `NULL` - Esempi

Data la tabella

Nome	Eta	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 **2**

Esercizio

Costruire una query che produca come risultato la media dei redditi dei figli di Franco

Esercizio

Costruire una query che produca come risultato la media dei redditi dei figli di Franco

Soluzione

```
SELECT AVG(Reddito)
FROM Persone JOIN Paternita
ON Nome=Figlio
WHERE Padre='Franco'
```

Eventuali valori nulli non vengono considerati nel calcolo della media