

Corso sul linguaggio SQL

Modulo L2B (SQL)

4.1 – Operazioni insiemistiche

M. Malatesta SQL4.1 Operazioni insiemistiche-07

1
01/01/2015

Prerequisiti

- Creazione e gestione tabelle
- Uso ambiente SQL
- Cenni di algebra relazionale

M. Malatesta SQL4.1 Operazioni insiemistiche-07

2
01/01/2015

Introduzione

In questa Unità iniziamo a descrivere i comandi SQL con i quali è possibile svolgere operazioni sulle tabelle presenti in un DB.

In particolare, esaminiamo le **operazioni insiemistiche**.

M. Malatesta SQL4.1 Operazioni insiemistiche-07

3
01/01/2015

Informazioni generali

N.B. – A solo scopo didattico:

- i caratteri MAIUSCOLI indicano parole chiave del linguaggio;
- i caratteri *corsivi* indicano elementi che dovranno essere specificati dal programmatore;
- le parentesi quadre indicano opzione
- la barra verticale “|” indica alternativa.

M. Malatesta SQL4.1 Operazioni insiemistiche-07

4
01/01/2015

Operare sui dati

Le operazioni insiemistiche sulle tabelle di un DB possono essere effettuate mediante un unico potente comando: **SELECT**.

Questo comando presenta molte varianti (**clausole**) con le quali è possibile generare tutte le altre operazioni che vedremo successivamente.

Il risultato di un'operazione insiemistica su due tabelle è ancora una tabella temporanea che viene visualizzata in righe e colonne, come le tabelle del DB.

M. Malatesta SQL4.1 Operazioni insiemistiche-07

5
01/01/2015

Il comando **SELECT**

Vedremo il comando **SELECT** nei vari casi e poi ne daremo la sintassi generale che li riunisce tutti.

Nelle slide che seguono, illustriamo il comando **SELECT** per le seguenti operazioni:

- **Operazioni insiemistiche** (in questa Unità)
- **Ridenominazione** (in successive Unità)
- **Selezione** (in successive Unità)
- **Proiezione**
- **Congiunzione**
- **Ordinamento**

M. Malatesta SQL4.1 Operazioni insiemistiche-07

6
01/01/2015

Operazioni insiemistiche

Tramite il comando **SELECT** è possibile eseguire le operazioni insiemistiche:

- **Unione**
- **Intersezione**
- **Differenza**
- **Prodotto cartesiano**

M. Malatesta SQL4.1 Operazioni insiemistiche-07

7
01/01/2015

Operazioni insiemistiche

-Unione

Consideriamo le seguenti tabelle

PATERNITA

| Padre | Figlio |
|--------------|---------------|
| Sergio | Franco |
| Luigi | Olga |
| Luigi | Filippo |
| Franco | Andrea |
| Franco | Aldo |
| Antonio | Luigi |
| Marco | Anna |

MATERNITA

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

M. Malatesta SQL4.1 Operazioni insiemistiche-07

8
01/01/2015

Operazioni insiemistiche

-Unione

La **SELECT** può realizzare l'unione tra tabelle mediante l'operatore **UNION**.

Sintassi

```
(SELECT lista_attributi FROM tab1)
UNION
(SELECT lista_attributi FROM tab2);
```

dove

- *lista_attributi* è l'elenco di attributi da considerare
- *tab1*, *tab2* sono le tabelle coinvolte nell'operazione

Operazioni insiemistiche

-Unione

ATTIVITA': generare dalle tabelle *Paternita* e *Maternita* la tabella unione delle due tabelle, che riporti tutti gli attributi.

```
mysql> (SELECT padre, figlio FROM Paternita)
→ UNION
→ (SELECT madre, figlio FROM Maternita)
→ );
mysql> _
```

PATERNITA \cup **MATERNITA**

| Padre | Figlio |
|---------|---------|
| Sergio | Franco |
| Luigi | Olga |
| Luigi | Filippo |
| Franco | Andrea |
| Franco | Aldo |
| Antonio | Luigi |
| Marco | Anna |
| Luisa | Maria |
| Luisa | Luigi |
| Anna | Olga |
| Anna | Filippo |
| Maria | Andrea |
| Maria | Aldo |

Operazioni insiemistiche

-Unione

PATERNITA \cup MATERNITA

ATTIVITA': generare dalle tabelle *Paternita* e *Maternita* la tabella con i nomi dei soli genitori.

```
mysql> (SELECT padre FROM paternita)
  → UNION
  → (SELECT madre FROM maternita);
mysql> _
```

| Padre |
|---------|
| Sergio |
| Luigi |
| Franco |
| Antonio |
| Marco |
| Luisa |
| Anna |
| Maria |

SELECT

-Intersezione

La **SELECT** può realizzare l'intersezione tra tabelle mediante l'operatore **INTERSECT**.

Sintassi

```
(SELECT lista_attributi FROM tab1)
INTERSECT
(SELECT lista_attributi FROM tab2);
```

dove

- *lista_attributi* è l'elenco di attributi da considerare
- *tab1*, *tab2* sono le tabelle coinvolte nell'operazione

SELECT

-Intersezione

Ad esempio, date le tabelle seguenti, l'operazione di intersezione individua i dipendenti che svolgono turni di mattina e di pomeriggio.

```
mysql> SELECT cognome, nome FROM mattino
  → INTERSECT
  → SELECT cognome, nome FROM pomeriggio;
mysql> _
```

Non in tutte le implementazioni il comando **INTERSECT** viene implementato, per cui ci sono modi alternativi di realizzazione.

M. Malatesta SQL4.1 Operazioni insiemistiche-07

MATTINO

| Cognome | Nome |
|---------|---------|
| Bianchi | Mario |
| Alfonsi | Luigi |
| Verdi | Antonio |
| Magni | Piero |
| Liberti | Michele |

POMERIGGIO

| Cognome | Nome |
|---------|---------|
| Verdi | Antonio |
| Liberti | Michele |
| Pani | Elisa |
| Visca | Stefano |
| Pratesi | Giulio |

13
01/01/2015

SELECT

-Intersezione

Due esempi alternativi per ottenere lo stesso risultato.

MATTINO

```
mysql> SELECT mattino.cognome, mattino.nome
  → FROM mattino, pomeriggio WHERE
  → mattino.cognome=pomeriggio.cognome
  → AND mattino.nome=pomeriggio.nome;
mysql> _
```

POMERIGGIO

```
mysql> SELECT cognome FROM mattino WHERE
  → cognome IN
  → (SELECT cognome FROM pomeriggio);
mysql> _
```

M. Malatesta SQL4.1 Operazioni insiemistiche-07

| Cognome | Nome |
|---------|---------|
| Bianchi | Mario |
| Alfonsi | Luigi |
| Verdi | Antonio |
| Magni | Piero |
| Liberti | Michele |

| Cognome | Nome |
|---------|---------|
| Verdi | Antonio |
| Liberti | Michele |
| Pani | Elisa |
| Visca | Stefano |
| Pratesi | Giulio |

14
01/01/2015

SELECT

-Differenza

L'operatore differenza si realizza con la clausola **EXCEPT**.

Sintassi:

```
(SELECT lista_attributi FROM tab1)  
EXCEPT  
(SELECT lista_attributi FROM tab2);
```

dove

- *lista_attributi* è l'elenco di attributi da considerare
- *tab1, tab2* sono le tabelle coinvolte nell'operazione

SELECT

-Differenza

Nelle implementazioni in cui la clausola non è implementata, l'effetto si può ottenere ugualmente con:

```
SELECT lista_attributi FROM tab1  
WHERE lista_attributi NOT IN  
(SELECT lista_attributi FROM tab2);
```


SELECT

-Prodotto cartesiano

Il **prodotto cartesiano** di due tabelle si esegue con il comando **SELECT** nel seguente modo:

```
SELECT * FROM paternita, maternita;
```

Si noti la presenza di due tabelle in ingresso.

Argomenti

- Operare sui dati
- Il comando **SELECT**
- Operazioni insiemistiche
 - Unione
 - Intersezione
 - Differenza
 - Prodotto cartesiano

Altre fonti di informazione

- Atzeni, Ceri, Paraboschi, Torlone, Basi di dati - McGraw-Hill, 1996-2002
- A. Lorenzi-D.Rossi – Le basi di dati e il linguaggio SQL – ed. ATLAS