SQL - Structured Query Language

Alessandro Lori

Università di Pisa

18 marzo 2011

SQL - operazioni sui dati

SQL permette di svolgere operazioni di vario tipo sulle basi di dati

Operazioni di interrogazione svolte mediante il costrutto **`SELECT`**

Operazioni di modifica dati svolte mediante i costrutti **`INSERT`**, **`DELETE`**, **`UPDATE`**

Sintassi del costrutto **SELECT**

Sintassi

```
SELECT Attributo1, Attributo2, ...
FROM Tabella1, Tabella2, ...
[ WHERE condizione ]
```

SELECT indica quali attributi produrre in uscita

FROM indica quali tabelle utilizzare per produrre il risultato

WHERE permette di indicare le condizioni che devono essere rispettate dalle tuple. Più condizioni possono essere composte usando gli operatori logici **AND**, **OR**, **NOT**. Se non ci sono condizioni da imporre, la clausola WHERE può essere omessa.

Esempio 1

- Sia data la tabella a destra (Persone)
- Costruire una query che produca come risultato nome e reddito delle persone con meno di trenta anni

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 Nome, Reddito
FROM Persone
WHERE Eta < 30;
```

Nome	Reddito
Aldo	15
Andrea	21
Filippo	30
Gigi	21

Costrutto `DISTINCT`

- SQL non elimina i duplicati nel risultato (operazione costosa), va chiesto esplicitamente
- I duplicati possono essere eliminati utilizzando il costrutto `DISTINCT`
- I duplicati vengono eliminati dal risultato (solo dopo che la query ha prodotto il risultato)

Costrutto DISTINCT

Con duplicati

```
SELECT Reddito
FROM Persone
WHERE Eta < 30
```

Costrutto DISTINCT

Con duplicati

```
SELECT Reddito
FROM Persone
WHERE Eta < 30
```

Reddito
15
21
30
21

Costrutto DISTINCT

Con duplicati

```
SELECT Reddito
FROM Persone
WHERE Eta < 30
```

Reddito
15
21
30
21

Senza duplicati

```
SELECT DISTINCT Reddito
FROM Persone
WHERE Eta < 30
```

Costrutto DISTINCT

Con duplicati

```
SELECT Reddito
FROM Persone
WHERE Eta < 30
```

Reddito
15
21
30
21

Senza duplicati

```
SELECT DISTINCT Reddito
FROM Persone
WHERE Eta < 30
```

Reddito
15
21
30

Tutti gli attributi

Per visualizzare tutti gli attributi delle tabelle contenute nella clausola **FROM** si può usare il carattere $*$

Tutti gli attributi

Per visualizzare tutti gli attributi delle tabelle contenute nella clausola **FROM** si può usare il carattere $*$

```
SELECT Nome, Eta, Reddito
FROM Persone
WHERE Eta < 30
```

```
SELECT *
FROM Persone
WHERE Eta < 30
```

Risultato ordinato

- I risultati di una query possono essere ordinati utilizzando la clausola **ORDER BY**

Esempio

```
SELECT *
FROM Persone
ORDER BY Eta
```

Ordina le persone dalla più giovane alla più vecchia

- Più criteri possono essere specificati e si può anche ordinare in ordine decrescente

Risultato ordinato - 2

Esempio: più criteri

```
SELECT *
FROM Persone
ORDER BY Eta, Nome
```

Ordina le persone dalla più giovane alla più vecchia. A parità di età vengono ordinate alfabeticamente per nome

Esempio: ordine decrescente

```
SELECT *
FROM Persone
ORDER BY Eta DESC, Nome
```

Ordina le persone dalla più vecchia alla più giovane. A parità di età vengono ordinate alfabeticamente per nome

Costrutto `BETWEEN` ... `AND`

- L'operatore **`BETWEEN`** ... **`AND`** seleziona un intervallo di dati tra due valori (estremi inclusi).
- Tali valori possono essere:
  - Numeri
  - Testo
  - Date

Costrutto `BETWEEN` ... `AND`

- L'operatore **`BETWEEN`** ... **`AND`** seleziona un intervallo di dati tra due valori (estremi inclusi).
- Tali valori possono essere:
  - Numeri
  - Testo
  - Date

Esercizio 2

Scrivere una query che restituisce nome e reddito delle persone con età compresa tra i trenta anni e i sessanta anni e reddito maggiore di 25.

Esercizio 2 - Soluzione

Con **BETWEEN**

```
SELECT Nome, Reddito
FROM Persone
WHERE Reddito > 25
AND Eta BETWEEN 30 AND 60
```

Esercizio 2 - Soluzione

Con **BETWEEN**

```
SELECT Nome, Reddito
FROM Persone
WHERE Reddito > 25
AND Eta BETWEEN 30 AND 60
```

... oppure senza

```
SELECT Nome, Reddito
FROM Persone
WHERE Reddito > 25
AND Eta >= 30
AND Eta <= 60
```

Esercizio 3

Costruire una query che produca come risultato nome e reddito delle persone con età minore di trenta o maggiore di sessanta anni e reddito maggiore di 25.

Esercizio 3

Costruire una query che produca come risultato nome e reddito delle persone con età minore di trenta o maggiore di sessanta anni e reddito maggiore di 25.

```
SELECT Nome, Reddito
FROM Persone
WHERE Reddito > 25
AND (Eta NOT BETWEEN 30 AND 60)
```

Esercizio 3

Costruire una query che produca come risultato nome e reddito delle persone con età minore di trenta o maggiore di sessanta anni e reddito maggiore di 25.

```
SELECT Nome, Reddito
FROM Persone
WHERE Reddito > 25
AND (Eta NOT BETWEEN 30 AND 60)
```

```
SELECT Nome, Reddito
FROM Persone
WHERE Reddito > 25
AND ( Eta < 30 OR Eta > 60)
```

Confronti con il testo: operatore `LIKE`

- L'operatore **`LIKE`** serve per esprimere l'operazione di *pattern matching* su stringhe.
- Tale condizione permette di indicare i caratteri (o gruppi di caratteri) che devono essere contenuti in una stringa:
  - _ indica un singolo carattere non specificato
  - % indica un gruppo di caratteri non specificato

Esempi

La condizione Nome **`LIKE`** '%o' restituisce vero per i nomi che terminano per 'o'
La condizione Nome **`LIKE`** '`____`' restituisce vero per i nomi composti di 4 lettere

Esercizio 4

Costruire una query che produca come risultato nome, età e reddito delle persone che hanno un nome che inizia per ’A’ e ha una ’d’ come terza lettera.

Esercizio 4

Costruire una query che produca come risultato nome, età e reddito delle persone che hanno un nome che inizia per 'A' e ha una 'd' come terza lettera.

Soluzione

```
SELECT *
FROM Persone
WHERE Nome LIKE 'A_d%'
```

Metacaratteri e caratteri

- Come posso fare a considerare i metacaratteri come caratteri semplici?
- Ad esempio se voglio trovare tutti i nomi che iniziano per il carattere '_' devo indicare che il carattere non deve essere interpretato come metacarattere.
- Devo inserire un carattere di escape, scelto opportunamente:

Metacaratteri e caratteri

- Come posso fare a considerare i metacaratteri come caratteri semplici?
- Ad esempio se voglio trovare tutti i nomi che iniziano per il carattere '_' devo indicare che il carattere non deve essere interpretato come metacarattere.
- Devo inserire un carattere di escape, scelto opportunamente:

Esempio

```
SELECT *
FROM Persone
WHERE Nome LIKE '!_%' ESCAPE('!')
```

I valori NULL

- La base di dati può contenere al suo interno dei valori **NULL**
- Un valore **NULL** indica che un valore sconosciuto o inesistente
- Questi valori devono essere presi esplicitamente in considerazione nel momento in cui vengono costruite le condizioni da inserire nel costrutto **WHERE**
- Una qualsiasi condizione che coinvolge un campo **NULL** restituiscono un valore sconosciuto che nella pratica viene trattato come un valore falso

I valori `NULL` - Esempio

Matricola	Cognome	Filiale	Eta
1000	Verdi	NULL	56
5998	Neri	Milano	45
7309	Rossi	Roma	32
9553	Bruni	Milano	NULL

- Si elenchino gli impiegati con età superiore ai 40

I valori `NULL` - Esempio

Matricola	Cognome	Filiale	Eta
1000	Verdi	NULL	56
5998	Neri	Milano	45
7309	Rossi	Roma	32
9553	Bruni	Milano	NULL

- Si elenchino gli impiegati con età superiore ai 40

Soluzione

```
SELECT *
FROM Impiegati
WHERE Eta > 40
```

Bruni non compare nel risultato

L'operatore `IS NULL` - Esempio

Matricola	Cognome	Filiale	Eta
1000	Verdi	NULL	56
5998	Neri	Milano	45
7309	Rossi	Roma	32
9553	Bruni	Milano	NULL

- Si elenchino gli impiegati con età sconosciuta o superiore ai 40

Soluzione

```
SELECT *
FROM Impiegati
WHERE Eta > 40
OR Eta IS NULL
```

Stavolta compare anche Bruni nel risultato

Ridenominazioni: l'operatore **AS**

- I campi del risultato e le tabelle possono essere rinominate (temporaneamente) nella query utilizzando l'operatore **AS**
- Sarà soprattutto utile con query più complesse (Più volte la stessa tabella o campi dallo stesso nome in tabelle diverse)

Esempio

```
SELECT X.Att1 , Y.Att4
FROM   Tabella AS X, Tabella AS Y
WHERE  X.Att2 = Y.Att3
```

Espressioni nella `SELECT`

- È possibile introdurre all'interno della clausola SELECT alcune semplici espressioni (somma, moltiplicazione, sottrazione, divisione).

Esempio

```
SELECT Reddito/2 AS RedditoSemestrale
FROM Persone
WHERE Nome = 'Luigi'
```