Home Page
Novità
Aiuto

Il linguaggio SQL
L'istruzione SELECT

http://www.vbsimple.net/database/db_08_01.htm

Difficoltà: 3 / 5

Sicuramente la prima e la più utilizzata tra le istruzioni di questo linguaggio. Serve letteralmente a selezionare un gruppo di valori da un insieme di tabelle, specificando opzionalmente dei criteri di selezione, di raggruppamento e di ordinamento. La sintassi generale dell'istruzione è:

```
SELECT <campi>
[ INTO <tabella> ]
FROM <origini>
[ JOIN <origine> ON <criteri> ]
[ WHERE <criteri> ]
[ GROUP BY <campi> ]
[ HAVING <criteri> ]
[ ORDER BY <criteri di ordinamento> ]
```

Delle 5 righe indicate soltanto la prima è obbligatoria; le successive rappresentano ulteriori clausole oppure operatori SQL che vedremo in seguito. Nella sua forma più semplice si compone del nome dei campi da scegliere e dal nome di una tabella come origine:

```
SELECT id, nome, cognome FROM dipendenti
```

L'esempio estrae i campi ID, NOME e COGNOME dalla tabella di nome *dipendenti*; nessun filtro sarà operato ma saranno quindi restituite tutte le righe presenti all'interno della tabella indicata. Se dovesse essere necessario estrarre **tutti i campi** dall'origine indicata può essere utilizzato il **simbolo *** come segue:

```
SELECT * FROM dipendenti
```

Si tratta di una pratica generalmente da evitare se non è davvero necessario. In tal modo infatti l'intero contenuto della tabella sarà recuperato e trasferito lungo la rete al richiedente. Generalmente basterà selezionare solo i campi interessati.

I dati saranno restituiti in forma tabellare (quindi a righe e colonne) ed i nomi dei campi saranno quelli originali recuperati dalla tabella di origine. In caso di necessità sarà possibile modificare il nome dei singoli campi restituiti utilizzando l'operatore **AS**:

```
SELECT id AS id_cliente, nome AS nome_cliente, cognome AS cognome_cliente
FROM clienti
```

La riga precedente estrae i tre campi ID, NOME e COGNOME e li assegna ai campi di nome *ID_CLIENTE*, *NOME_CLIENTE* e *COGNOME_CLIENTE*. Sebbene sia possibile inserire degli spazi nei nomi dei campi si consiglia di evitarlo sia per diminuire la probabilità di errori sia perché non tutti i database lo consentono. Esistono due sintassi

molto comuni per racchiudere i nomi dei campi quando questi contengono degli spazi o altri simboli che potrebbero arrecare disturbo all'elaborazione. I due simboli sono le parentesi quadre **[ ]** ed il carattere di accento ` (non si tratta dell'apostrofo della tastiera italiana) riproducibile con la sequenza di tasti **ALT+096** oppure con l'apposito tasto sotto ESC nella tastiera americana. Alcuni database preferiscono le parentesi quadre mentre altri l'accento, pertanto si rimanda alla documentazione del singolo database.

```
SELECT id AS [id cliente], nome AS [nome cliente], cognome AS [cognome
cliente] FROM clienti
```

oppure

```
SELECT id AS `id cliente`, nome AS `nome cliente`, cognome AS `cognome
cliente` FROM clienti
```

Alla stessa maniera naturalmente vanno racchiusi i nomi dei campi di origine, cioè quelli che si trovano alla sinistra di AS, quando questo si renda necessario per la presenza di spazi o altri simboli particolari. Solitamente i programmi che generano istruzioni SQL racchiudono tutti i nomi dei campi tra i precedenti simboli anche quando non è strettamente necessario.

Quando un campo può riferirsi a più di una tabella dell'origine (vedremo subito dopo come specificare più di un'origine), è necessario fornire il nome dell'origine cui il campo si riferisce, ma è comunque possibile farlo anche quando si tratta di una sola origine:

```
SELECT dipendenti.id, dipendenti.nome, dipendenti.cognome FROM dipendenti
```

È anche possibile rinominare il nome delle singole origini, semplicemente specificando il nuovo nome immediatamente dopo il nome originale:

```
SELECT dip.id, dip.nome, dip.cognome FROM dipendenti dip
```

Tutte le espressioni che abbiamo visto finora riportavano i campi di tutti i record di una sola tabella nell'**ordine casuale** con cui le righe vengono estratte. Utilizzando la clausola WHERE che vedremo in seguito è possibile filtrare la scelta dei record da estrarre. La clausola ORDER BY invece consente di ordinare i risultati. Tralasciamo per il momento le forme più avanzate di uso dell'istruzione SELECT per concentrarci sui fondamenti.

Specificando come origine più di un nome di tabella si ottiene una permutazione di righe per ciascuna tabella, ad esempio date due tabelle di nome dipendenti e dettaglio, con un campo ciascuna di nome ID e quattro righe in entrambe le tabelle, la selezione:

```
SELECT dipendenti.id AS id1, saldi.id AS id2 FROM dipendenti, saldi
```

Saranno restituite 16 righe (4*4):

id1	id2
1	1
1	2

1	3
1	4
2	1
2	2
2	3
2	4
3	1
3	2
3	3
3	4
4	1
4	2
4	3
4	4

Naturalmente se le tabelle fossero state 3, le righe restituite saranno 64 (4*4*4). Questa scelta potrà sembrare una sciocchezza ma in realtà si tratta di un'ampia possibilità: sono infatti presentate tutte le possibili combinazioni con i due gruppi di 4 numeri; sarà sempre possibile scegliere solo quelli che necessitano. Questa operazione è alla base dell'operatore JOIN che unisce più origini con un certo criterio di scelta, ma può essere risolta anche con l'uso della clausola WHERE.

Abbiamo lasciato per ultimo un operatore particolare utilizzabile nelle istruzioni di selezione: si tratta di **DISTINCT**, dedicato a filtrare le righe ripetute. La sua sintassi è molto semplice:

```
SELECT DISTINCT <campi> FROM <origini>
```

Essendo posto prima dei campi, la sua azione è valida su tutti i campi; saranno infatti eliminate le righe dei risultati completamente uguali tra loro. Vediamo un esempio per semplificare la comprensione del problema:

```
SELECT DISTINCT nome FROM dipendenti
```

Restituirà tutti i nominativi unici dei dipendenti; saranno ovvero esclusi i nomi duplicati. Dipendenti con nome uguale e cognome differente saranno comunque annoverati nell'unica riga con il loro nome.

```
SELECT DISTINCT nome, cognome FROM dipendenti
```

Tuttavia in questo caso, se due dipendenti avessero lo stesso nome, sarebbero comunque riportati entrambi perché l'esclusione dei duplicati agisce soltanto **sulle righe intere dei risultati** e non sui dati da recuperare dall'origine. Sarebbero allora esclusi soltanto quei dipendenti con lo stesso nome e lo stesso cognome, caso assai difficile in una singola azienda.

Fibia FBI
14 Marzo 2004

Torna all'indice della sezione Database
