

Esercitazione SQL

Il linguaggio SQL (structured query language) è il linguaggio standard per la gestione dei database relazionali. Con l'uso di SQL è possibile definire strutture, manipolare e interrogare il db.

In questa esercitazione impareremo ad interrogare una tabella attraverso l'uso del comando select e ad integrarne l'utilizzo nel codice ADO di VB ("SQL embedded").

La sintassi del comando è

```
SELECT nomecampo1, nomecampo2, nomecampo3  
FROM nometabella;
```

l'esecuzione genera una tabella "virtuale", cioè temporanea, estraendo dalla tabella originaria solo i campi elencati.

Il linguaggio SQL è ad "alto livello", il programmatore viene liberato dalla costruzione di codice procedurale, le istruzioni sono relative a **COSA** si deve ottenere, e non **COME** ottenerlo.

Qualora volessimo estrarre, dalla tabella originaria, solo alcune righe, aggiungeremo la clausola where

```
SELECT nomecampo1, nomecampo2, nomecampo3  
FROM nometabella  
WHERE condizione;
```

dove per condizione si intende nomecampo=valore, con possibilità di avere condizioni composte con l'uso degli operatori logici (and, or).

Per capire meglio significato ed utilizzo passiamo subito ad un esempio pratico.

Creiamo una tabella di nome iscritti così strutturata

Nick	Testo
Pw	testo
Email	testo
Data	Date data di iscrizione, cioè di memorizzazione del record
Livello	Numerico 0 Administrator super, 1 administrator, 2 membro senior, 3 membro junior

Apriamo il programma VB e scriviamo il codice per la connessione al database.

Quindi inseriamo e colleghiamo un nuovo form dove via via andremo a provare le nuove istruzioni, scegliendo un controllo per attivare l'esecuzione.

Vogliamo stampare l'elenco dei nick, delle pw e del livello. Seguiamo le istruzioni

Private Sub.....

```
Dim rs As New adodb.Recordset
```

```
' dichiariamo l'oggetto recordset cui assegnare il risultato della interrogazione
```

```
Dim query As String
```

```
'dichiaro una variabile stringa per costruire il comando
```

```
query = "select nick, pw, livello from iscritti;"
```

```
'istruzione SQL il nome dei campi è quello dichiarato nella tabella
```

```
rs.Open (query), conn
```

```
' eseguiamo la query ed otteniamo in rs il recordset
```

```
while not rs.eof
```

```
'procediamo con il normale codice
Label1 = rs!nick
label2=rs!pw
msgbox("next")
rs.movenext
wend
End Sub
```

Per evidenziare pienamente le enormi potenzialità di questo linguaggio proviamo a risolvere altre interrogazioni, anche con l'aiuto di nuove "clausole".

Elenco ordinato per livello

Utilizziamo la clausola **ORDER BY** del comando select modificando la stringa

- query = "select nick, pw, livello from iscritti **order by** livello;"
- per avere nel recordset i record ordinati secondo il criterio specificato

Elenco dei soli amministratori

utilizziamo la clausola **where** con l'ausilio dell'operatore or

- query = "select nick, pw from iscritti where livello=0 or livello=1;"

Elenco dei livelli effettivamente memorizzati

In tal caso l'uso del predicato **DISTINCT** elimina la ripetizione di righe uguali, specificando una sola volta il livello anche nel caso si ripeta all'interno della tabella

- query = "select **distinct** livello from iscritti;"

Possiamo costruire il comando anche utilizzando confronti con valori acquisiti in input, ad es. trovare la pw di un nick scelto

- query = "select pw from iscritti where nick='" + Text1 + "';"

naturalmente il risultato dell'esecuzione sarà un solo record, se esiste.

E' evidente quindi la possibilità di utilizzare tali istruzioni per innumerevoli problematiche, tra le quali vi è sicuramente il controllo delle operazioni di login ad un'area riservata.

Provate a costruire la query che regoli gli accessi.

All'interno del comando select è possibile utilizzare diverse **funzioni di aggregazione** predefinite, che svolgono dei calcoli sulla tabella e restituiscono il risultato dell'operazione.

La funzione **COUNT** restituisce il numero di righe presenti in una tabella

- query="select **count**(*) as conteggio from iscritti;"

fornisce in uscita il numero di record memorizzati in iscritti

'per poter prelevare tale valore nella query gli abbiamo assegnato un identificatore

```
rs.open(query), conn
```

```
label1=rs!conteggio
```

- query="select **count**(*) as conteggio from iscritti where livello=0;"

In tal caso il valore estratto è il conteggio dei record avententi livello 0

Altre funzioni utilizzabili sono

SUM(campo) restituisce la somma dei valori contenuti in una colonna

AVG(campo) restituisce la media dei valori contenuti in una colonna

MIN(campo) restituisce il valore minimo tra quelli contenuti in una colonna

MAX(campo) restituisce il valore massimo tra quelli contenuti in una colonna