

Istruzioni SQL

Istruzioni SQL	1
Query di selezione	2
Istruzione SELECT	2
Istruzione SELECT DISTINCT	2
ORDER BY	3
WHERE	3
La condizione LIKE	4
BETWEEN ... AND	5
Query di inserimento	5
INSERT INTO	5
Query di cancellazione	6
DELETE	6
Alias	6
Alias di campi	6
Funzioni SQL	6

La sintassi delle istruzioni SQL varia leggermente in base al software che utilizzo. In linea di massima comunque la struttura è identica per tutti. In questi esempi mi sono basato sulla sintassi di MySQL.

Query di selezione

Istruzione SELECT

L'istruzione Select è usata per selezionare dei dati da una tabella. Il risultato sarà inserito in una tabella chiamata result-set.

```
SELECT campo1,campo2,... FROM NomeDB,NomeTabella;
```

Questa istruzione ricerca i campi campo1, campo 2 ... nella tabella NomeTabella del Dbase NomeDB.

Un esempio pratico potrebbe essere:

```
SELECT id,nome,telefono FROM DBtelefoni.telefoni;
```

Id	nome	telefono
1	Pippo	0444/543188
2	Pluto	0444/991761
3	Paperino	0444/746111
4	Topolino	0444/519922

Se volessi invece vedere tutti i campi del Dbase posso utilizzare il simbolo * in questo modo

```
SELECT * FROM DBtelefoni.telefoni;
```

Id	nome	indirizzo	telefono
1	Pippo	Via Rossi, 4	0444/543188
2	Pluto	Via Piave, 11	0444/991761
3	Paperino	Via Roma, 1	0444/746111
4	Topolino	Via Verdi, 6	0444/519922

Vengono così mostrati tutti i valori di campi di tutti i valori della tabella telefoni del database Dbtelefoni.

Istruzione SELECT DISTINCT

La parola chiave DISTINCT è utilizzata per restituire soltanto dei valori distinti.

L'istruzione SELECT fornisce informazioni circa il contenuto delle tabelle. Ma se volessimo selezionare soltanto elementi distinti? In SQL, occorre aggiungere la parola chiave DISTINCT all'istruzione SELECT:

Ad esempio
SELECT Società FROM Ordini;
restituirebbe

Società
Microsoft
Oracle
Microsoft
HP

Mentre

```
SELECT DISTINCT Società FROM Ordini;
```

restituirebbe

Società
Microsoft
Oracle
HP

Adesso "Microsoft" compare una volta soltanto nel result-set.

ORDER BY

E' possibile ordinare l'elenco ottenuto con il comando ORDER BY NomeCampo. Ad esempio

```
SELECT id,nome,telefono FROM DBtelefoni.telefoni ORDER BY nome;  
restituirebbe
```

Id	nome	telefono
3	Paperino	0444/746111
1	Pippo	0444/543188
2	Pluto	0444/991761
4	Topolino	0444/519922

Mentre

```
SELECT id,nome,telefono FROM DBtelefoni.telefoni ORDER BY nome DESC;  
diventerebbe
```

Id	nome	telefono
4	Topolino	0444/519922
2	Pluto	0444/991761
1	Pippo	0444/543188
3	Paperino	0444/746111

WHERE

Con questi sistemi è possibile però mostrare solo tutti i record del Dbase. Se volessi restringere la ricerca uno dei sistemi potrebbe essere quello di usare l'istruzione WHERE in questo modo.

```
SELECT campo1,campo2,campoX FROM NomeDB,NomeTabella WHERE campo1 = 'valore';
```

In questo modo verrebbero mostrati solo i record per i quali il campo campo1 è uguale a valore.

Ad esempio

```
SELECT id,nome,telefono FROM DBtelefoni.telefoni WHERE nome = 'Topolino' ORDER BY nome DESC;
```

Id	nome	telefono
4	Topolino	0444/519922

E' inoltre possibile usare gli operatori logici per restringere ancora di più i risultati. Ad esempio

SELECT id,nome,telefono FROM DBtelefoni.telefoni WHERE nome = 'Topolino' AND id >= 4;

Id	nome	telefono
4	Topolino	0444/519922
9	Topolino	0444/214569

Oppure

SELECT id,nome,telefono FROM DBtelefoni.telefoni WHERE nome = 'Topolino' OR id >= 4;

Id	nome	telefono
4	Topolino	0444/519922
45	Paperoga	0445/124589
9	Topolino	0444/214569
12	Clarabella	0444/927856

Da notare che se il valore che viene cercato è una stringa è necessario inserirlo tra apici.

SELECT id,nome,telefono FROM DBtelefoni.telefoni WHERE nome = Topolino OR id >= 4;

restituirebbe un errore perché Topolino è una stringa.

La condizione LIKE

Se non sono sicuro del valore che cerco, ma ne conosco solo una parte posso utilizzare LIKE al posto di = e il simbolo % per indicare la parte non nota della stringa.

Ad esempio

SELECT id,nome FROM telefoni.telefoni WHERE nome LIKE '%ert%';

id	nome
405	Roberto
421	Alberto
437	Roberta
440	Alberto
783	Umberto
1008	Roberto

Vengono mostrati tutti i record nei quali il campo nome contiene la stringa 'ert'

Se scriviamo invece

SELECT id,nome FROM telefoni.telefoni WHERE nome LIKE 'Pa%';

id	nome
465	Paola
987	Paolo
1005	Paola
1016	Patrizio
1027	Paolo
1170	Paola
1202	Pasquale
1210	Pasquale

1364	Patrizia
------	----------

Ritroveremmo tutti quei record nei quali il campo nome contiene una stringa che inizia per 'Pa'.

Allo stesso modo il comando

```
SELECT id,nome FROM telefoni.telefoni WHERE nome LIKE '%erto';
```

id	nome
405	Roberto
421	Alberto
440	Alberto
783	Umberto
1008	Roberto
1073	Roberto

Restituirebbe i record nei quali il campo nome contiene stringhe che finiscono per 'erto'.

BETWEEN ... AND

L'operatore BETWEEN ... AND seleziona un intervallo di dati tra due valori. Questi valori possono essere numeri, testo o date.

```
SELECT Campo1 FROM NomeDb.NomeTabella WHERE Campo2 BETWEEN valore1 AND valore2
```

Ad esempio se voglio selezionare dei record con i valori di 'id' tra 4 e 22 (Compresi) scriverò:

```
SELECT * FROM telefoni WHERE id BETWEEN 4 AND 22;
```

Se volessi tutti gli altri valori scriverei

```
SELECT * FROM telefoni WHERE id NOT BETWEEN 4 AND 22;
```

Query di inserimento

INSERT INTO

E' sempre possibile inserire in un dBase dei record con comandi SQL. La sintassi è la seguente:

```
INSERT INTO NomeDbase,NomeTabella (campo1,campo2,...) VALUES (valore1,valore2,...);
```

facendo attenzione a inserire tra apici il valore se questo fosse una stringa. Ad esempio se volessimo inserire un record nella tabella

IDUser	FullName	Indirizzo	Interno
1	Mario Bianchi	Via Verdi, 5	7894
2	Carlo Verdi	Via Roma ,45	1546

Possiamo operare con l'istruzione:

```
INSERT INTO dbase.tabella_dipendenti (IDUser,FullName,Interno) values (3,'Antonio Rossi',1234);
```

Il risultato sarebbe

IDUser	FullName	Indirizzo	Interno
1	Mario Bianchi	Via Verdi, 5	7894
2	Carlo Verdi	Via Roma ,45	1546
3	Antonio Rossi		1234

UPDATE ... SET ... WHERE

Questa istruzione serve per modificare i dati in una tabella. La sintassi è

UPDATE NomeDb.NomeTabella SET Campo1 = valore1 WHERE Campo2 = valore2

Ad esempio se vogliamo aggiungere il nome 'Paperone' al record col cognome uguale a 'De Paperoni' scriveremo:

UPDATE Persone SET Nome = 'Paperone' WHERE Cognome = 'De Paperoni';

Query di cancellazione

DELETE

Per cancellare un record in un Dbase la sintassi è:

DELETE FROM NomeDb.NomeTabella WHERE campo1 = 'Valore';

Ad esempio per cancellare Paperon de Paperoni dalla tabella Persone del Dbase DB

DELETE FROM DB.Persone WHERE Cognome = 'De Paperoni';

Alias

Gli Alias SQL sono usati per dare dei nomi ai campi e alle tabelle.

Alias di campi

Sintassi

SELECT campo AS alias_campo FROM tabella;

Alias di tabelle

Sintassi

SELECT campo1, campo2 FROM NomeTabella AS AliasTabella;

Funzioni SQL

Sintassi

SELECT funzione(cmpo) FROM NomeTabella;

Elenco funzioni SQL

Funzione	Descrizione
AVG(column)	Restituisce il valore medio di una colonna
COUNT(column)	Restituisce il numero di righe (escludendo quelle con valore NULL) di una colonna
COUNT(*)	Restituisce il numero di righe selezionate
FIRST(column)	Restituisce il valore del primo record di un campo specificato
LAST(column)	Restituisce il valore dell'ultimo record di un campo specificato
MAX(column)	Il valore massimo in una colonna
MIN(column)	Il valore minimo
SUM(column)	La somma dei valori di una colonna