

Esempio Pilota: QuerySuSingolaTabella

basato sul database di esempio "TestateFumetti"
(con un'unica tabella "Testate")

Esempio 1

/*

Esempio 1:
Uso di un semplice comando SELECT nella forma semplificata:
SELECT <elenco-campi> **FROM** <tabella>
WHERE <condizione> **ORDER BY** <elenco-campi>

TESTO:
Elencare Testata, Tipo, AnnoDiPrimaPubblicazione delle sole testate con AnnoDiPrimaPubblicazione dal 2000 in poi, ordinate per Anno di Prima Pubblicazione decrescente e, a parità di anno, per Testata crescente.
*/

```
SELECT Testata, Tipo, AnnoDiPrimaPubblicazione
FROM Testate
WHERE AnnoDiPrimaPubblicazione >= 2000
ORDER BY AnnoDiPrimaPubblicazione DESC, Testata ASC
```

-- Nota Bene:
-- Nella clausola ORDER BY si possono specificare più livelli di ordinamento.
-- DESC significa decrescente, ASC significa crescente.
-- Se non si specifica, si intende ASC

Esempio 2

/*

Esempio 2:
Ricerca su campi testuali e uso del singolo apice per delimitare valori string

TESTO:
Elencare tutti i campi delle sole testate di tipo spillato, ordinate per Testata in ordine alfabetico decrescente.
*/

```
SELECT *
FROM Testate
WHERE Tipo = 'spillato'
ORDER BY Testata DESC
```

-- Nota Bene:
-- il simbolo '*' (asterisco), usato nell'elenco dei campi, significa 'tutti i campi'.
-- il campo Tipo (char(15)) contiene spazi finali fittizi, ma essi vengono ignorati.
-- la differenza fra maiuscole e minuscole viene ignorata.

Esempio 3

/*

Esempio 3:
Uso dell'operatore Like e del carattere jolly '%'

TESTO:
Elencare Testata, Tipo, Numero di Uscite delle sole testate che iniziano per 'BATMAN' ordinate dalla testata con maggior Numero di Uscite a scendere.
*/

```
SELECT Testata, Tipo, NumeroDiUscite
FROM Testate
WHERE Testata LIKE 'BATMAN%'
ORDER BY NumeroDiUscite DESC
```

-- Nota Bene: Per ottenere le testate che "contengono" la stringa 'BATMAN' -----
-- utilizzare la condizione: Testate LIKE '%BATMAN%'

Esempio 4

/*

Esempio 4:
Campi calcolati e Uso delle funzioni UPPER, LTRIM, RTRIM

TESTO:
Generare un elenco la cui prima colonna sia la Testata (priva di spazi iniziali o finali) con fra parentesi la Casa Editrice e come seconda colonna il Tipo espresso in maiuscolo. Elencare solo le testate uscite dal 2005 in poi, ordinate secondo la prima colonna calcolata.
*/

```
SELECT RTRIM(LTRIM(Testata)) +
        '(' + RTRIM(CasaEditrice) + ')' AS TestataCorretta,
        UPPER(Tipo)
FROM Testate
WHERE AnnoDiPrimaPubblicazione >= 2005
ORDER BY TestataCorretta ASC
```

-- Nota Bene:
-- la prima colonna è un CAMPO CALCOLATO alla quale è stato attribuito il nome TestataCorretta (alias).
-- la seconda colonna è anch'essa un CAMPO CALCOLATO alla quale NON è stato attribuito il nome.

Esempio 5

/*

Esempio 5:
Isolare anno, mese o giorno del tipo data e controllarne il formato di visualizzazione

TESTO:
Generare un elenco con Testata, in cui la seconda colonna sia solo l'ANNO di Ultima Uscita e infine con la data completa di ultima uscita isualizzata nel formato abituale usato in Italia. Elencare solo le testate la cui ultima uscita cade nel 1° semestre del 2005. Ordinare per data di ultima uscita decrescente.
*/

```
SELECT Testata, AnnoDiPrimaPubblicazione,
        YEAR(DataUltimaUscita) as AnnoUltimaUscita,
        FORMAT(DataUltimaUscita,
                'ddd dd MMMM yyyy', 'it-it') as DataUltimaUscita
FROM Testate
WHERE (DataUltimaUscita >= Convert(Date, '01-01-2005')) AND
        (DataUltimaUscita <= Convert(Date, '06-30-2005'))
ORDER BY AnnoUltimaUscita DESC
```

/* Nota Bene:
1) la funzione YEAR restituisce, come numero, l'anno della data indicata.
2) la funzione FORMAT converte la data nel formato che desideriamo. Il parametro 'it-it' indica che la data va espressa in italiano (es., i mesi saranno GENNAIO, FEBBRAIO, ecc.), mentre la "stringa di formato" indica in che modo desideriamo esprimere la data (es. dd=day, MM=mounth in 2 cifre, MMM=mounth in 3 lettere, MMMM=mounth per esteso, yy=year a 2 cifre, yyyy=year esteso, ddd=day della settimana in 3 lettere, dddd=day per esteso, ecc.)

3) Per esprimere un valore di tipo data, conviene scriverlo come stringa nella forma "mm-gg-aaaa" e convertirlo in data con la funzione:

Convert(Date, <data-espressa-come-stringa>)

4) Evitare di confrontare Date direttamente con stringhe perchè tutto viene convertito in stringa e con la variabilità dei formati, il risultato è difficilmente controllabile.

5) Nella condizione della clausola WHERE è possibile usare operatori logici come AND, OR, NOT.

6) Funzioni analoghe esistono per gli orari.

7) E' possibile usare come alias un nome di campo esistente, come se si ridefinisse il campo.

*/

Esempio 6

/*

Esempio 6:

Uso del tipo bit e valori true, false e NULL

TESTO:

Elencare Testata, Tipo, FormatoMaxi e NumeroDiUscite delle testate che hanno NULL nel campo Tipo e sono in formato maxi. Ordinare per Testata.

*/

```
SELECT Testata, Tipo, FormatoMaxi, NumeroDiUscite
FROM Testate
WHERE (Tipo IS NULL) AND
      (FormatoMaxi = 'true')
ORDER BY Testata
```

/* Nota Bene:

- Per verificare se un campo è NULL usare la condizione: <campo> IS NULL (oppure IS NOT NULL)

- Il valore (0 o 1) campo di tipo bit, se confrontato con una stringa viene implicitamente convertito in 'false' (se è 0) o 'true' (se è 1) per cui è possibile usare la condizione: <campo> = 'false' (o 'true')

- Ovviamente è anche possibile scrivere: <campo> = 0 (... oppure... = 1)

*/