

Introduzione a MySQL



Moreno Marzolla
Dipartimento di Informatica
Università Ca' Foscari di Venezia
marzolla@dsi.unive.it
<http://www.dsi.unive.it/~marzolla>

Definizione

- Una tabella (o relazione) è costituita da
 - Un insieme di colonne
 - Ciascuna colonna è etichettata con un nome, e tutti i dati di quella colonna hanno lo stesso tipo (intero, stringa, ecc.)
 - Un insieme di righe
 - Ciascuna riga prende il nome di *record*, e rappresenta una informazione memorizzata nel database
- Una base di dati *relazionale* (database relazionale) è un insieme di tabelle

SQL

- SQL è un linguaggio standard per la manipolazione di database relazionali
 - Creazione/cancellazione/modifica di tabelle
 - Creazione/cancellazione/modifica di colonne da tabelle
 - Inserimento e rimozione di record dalle tabelle
 - Interrogazioni complesse per recuperare tutte le informazioni in una o più tabelle che soddisfano determinati criteri
 - “Quale è l'articolo più costoso che abbiamo in magazzino?”

Esempio

Codice	Autore	Titolo	Prezzo
1	Manzoni	I Promessi Sposi	10.00
2	Asimov	Io, Robot	12.00
3	Alighieri	La Divina Commedia	8.00
4	Omero	Iliade	10.00
5	Omero	Odissea	11.00

Creare tabelle con SQL

```
create table nome_tabella (  
  nome_col_1 tipo attributi,  
  nome_col_2 tipo attributi,  
  nome_col_3 tipo attributi,  
  primary key (nome_col),  
  index nome_indice(nome_col)  
)
```

Tipi di colonne di testo

- **char(lunghezza)**
 - testo di lunghezza fissa, max 255 caratteri
- **varchar(lunghezza)**
 - testo di lunghezza variabile, max 255 caratteri
- **tinytext**
 - Equivalente a varchar(255)
- **text**
 - testo di lunghezza variabile, max 65535 caratteri
- **mediumtext, longtext**
 - testo di lunghezza ancora più lunga

Tipi di colonne numerici / 1

- **int [unsigned]**
 - numero intero compreso tra 0 e 4294967295 se unsigned, altrimenti tra -2147483648 e 2147483647
- **tinyint [unsigned]**
 - numero intero compreso tra 0 e 255 se unsigned, altrimenti tra -128 e 127
- **mediumint [unsigned] / bigint [unsigned]**
 - altri intervalli...
- **float / double**
 - numero con la virgola; double ha precisione (numero di cifre dopo la virgola) doppia rispetto a float

Tipi di colonne per data e ora

- **date**
 - Data in formato AAAA-MM-GG
- **datetime**
 - Data in formato AAAA-MM-GG OO:MM:SS
- **timestamp(size)**
 - Memorizza automaticamente l'ora della modifica più recente apportata ad una riga. La dimensione è un numero tra 2 e 14 (vedi libro)
- **time**
 - Archivia l'ora nel formato OO:MM:SS
- **year[(2|4)]**
 - Memorizza un anno in formato a 2 oppure 4 cifre

Esempio / 1

Codice	Autore	Titolo	Prezzo
1	Manzoni	I Promessi Sposi	10.00
2	Asimov	Io, Robot	12.00
3	Alighieri	La Divina Commedia	8.00
4	Omero	Iliade	10.00
5	Omero	Odissea	11.00

```
CREATE TABLE libri (  
  Codice INT UNSIGNED not null AUTO_INCREMENT,  
  Autore VARCHAR (255) not null ,  
  Titolo VARCHAR (255) not null ,  
  Prezzo FLOAT not null ,  
  PRIMARY KEY (Codice),  
  INDEX (Codice)  
)
```

Inserire valori nelle tabelle

```
insert into nome_tabella  
  (nome_col_1, nome_col_2, ...) values  
  (valore_1, valore_2, ...)
```

```
insert into libri (Autore, Titolo, Prezzo) values  
  ("Verga", "I Malavoglia", 18.0)
```

Interrogare un database

- **SELECT**
 - Viene utilizzata per estrarre delle colonne da una o più tabelle
 - Le colonne così selezionate sono usate per comporre una tabella (che però non esiste realmente nel database)

Esempio / 1

- Le seguenti interrogazioni sono equivalenti:
 - `select * from libri`
 - `select Codice, Autore, Titolo, Prezzo from libri`
- Entrambe producono:

Codice	Autore	Titolo	Prezzo
1	Manzoni	I Promessi Sposi	10.00
2	Asimov	Io, Robot	12.00
3	Alighieri	La Divina Commedia	8.00
4	Omero	Iliade	10.00
5	Omero	Odissea	11.00

Esempio / 2

- select Autore, Prezzo from libri

Autore	Prezzo
Manzoni	10.00
Asimov	12.00
Alighieri	8.00
Omero	10.00
Omero	11.00

La clausola where

- E' possibile dire alla select che deve mostrare solo le righe nella tabella risultato che soddisfano una determinata condizione, definita dalla clausola where
- select *nome_col_1, nome_col_2, ...* from *tabella* where *condizione*

Esempio / 1

- Tutti i libri che costano più di 10 euro
 - select * from libri where Prezzo > 10.0

Codice	Autore	Titolo	Prezzo
2	Asimov	Io, Robot	12.00
5	Omero	Odissea	11.00

- Tutti i libri scritti da Omero
 - select * from libri where Autore = "Omero"

Codice	Autore	Titolo	Prezzo
4	Omero	Iliade	10.00
5	Omero	Odissea	11.00

Esempio / 2

- Tutti i libri di prezzo compreso tra 9.0 e 12.0 euro
 - select * from libri where Prezzo > 9.0 and Prezzo < 12.0

Codice	Autore	Titolo	Prezzo
1	Manzoni	I Promessi Sposi	10.00
4	Omero	Iliade	10.00
5	Omero	Odissea	11.00

Esempio / 3

- Il titolo e l'autore di tutti i libri scritti da Manzoni oppure da Omero
 - `select Autore, Titolo`
`from libri`
`where Autore="Manzoni" or Autore="Omero"`

Autore	Titolo
Manzoni	I Promessi Sposi
Omero	Iliade
Omero	Odissea

Forme Normali / Prima FN

- Ogni colonna deve contenere valori atomici
 - Solo un valore per cella
 - Niente array simili
- Ogni colonna deve avere un nome univoco
- Ogni tabella deve avere un insieme di valori che identificano univocamente ciascuna riga
 - Questa è nota come la chiave primaria della tabella
- Non devono esserci righe duplicate
- Non devono esserci gruppi di dati ripetuti

Forme Normali / Seconda FN

- Il valore di ciascuna colonna deve dipendere da tutta la chiave primaria
 - Non solo da una sua parte

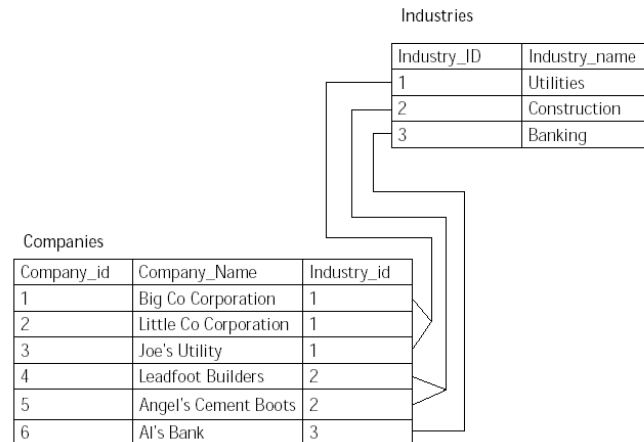
<u>company_name</u>	<u>company_location</u>	company_ceo	company_addr
BigCo Company	San Francisco	Bill Hurt	1121 43rd St
LittleCo Company	LA	LittleCo Company	4444 44th st

- Esempio: Chiave primaria multicolonna
 - Se BigCo Company apre una nuova filiale in un'altra città, il capo della compagnia è comunque lo stesso e va ripetuto

Forme normali / Terza FN

- Tutte le colonne devono dipendere direttamente dalla chiave primaria
 - Non sono ammesse dipendenze transitive

Relazione uno-a-molti

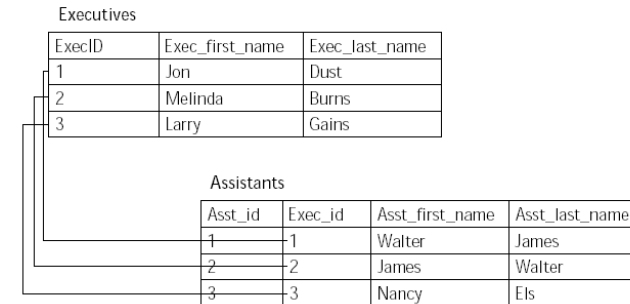


Moreno Marzolla

Tecnologie Web

21

Relazioni uno-a-uno

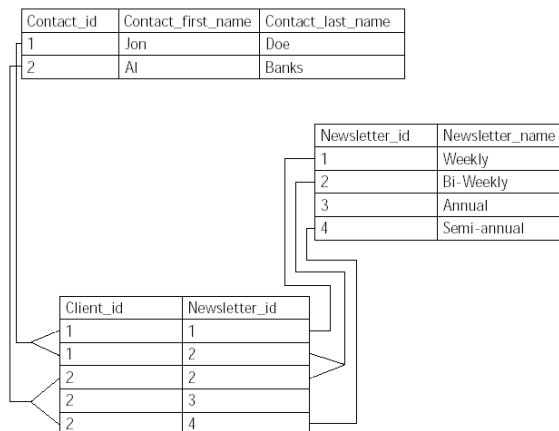


Moreno Marzolla

Tecnologie Web

22

Relazioni multi-a-molti



Moreno Marzolla

Tecnologie Web

23

Connessione al DB con PHP

```
mysql_connect("localhost", "nobody","password") or
die ("Could not connect to database");
mysql_select_db("guestbook") or
die ("Could not select database");
```

- La connessione dura solo per la pagina PHP corrente
 - Se l'utente apre una nuova pagina, occorre nuovamente connettersi al DB
 - PHP può anche gestire connessioni "persistenti"

Moreno Marzolla

Tecnologie Web

24

Inserire informazioni

```
$query = "insert into guestbook
(name,location,email,url,comments) values
(`$name`, `$location`, `$email`,
`$url`, `$comments`);"
mysql_query($query) or die (mysql_error());
```

- Ricordarsi di connettersi prima al DB!

Interrogare il DB

```
$result = mysql_query("select * from guestbook") or
die (mysql_error());
while ($row = mysql_fetch_array($result))
{
    echo "<b>Name:</b>";
    echo $row["name"];
    echo $row["location"];
    echo $row["email"];
    echo $row["url"];
    echo $row["comments"];
}
mysql_free_result($result);
?>
```

Nomi di colonne del DB

Interrogare il DB / 1

```
$db = mysql_connect("localhost","root", "")
or die ("Could not connect to localhost");
mysql_select_db("test", $db)
or die ("Could not find test");
$db_name = "topics";
$query = "select * from $db_name";
$result = mysql_query($query);
$num_fields = mysql_num_fields($result);
```

Interrogare il DB / 2

```
//create table header
echo "<table border = 1>";
echo "<tr>";
for ($i=0; $i<$num_fields; $i++)
{
    echo "<th>";
    echo mysql_field_name ($result, $i);
    echo "</th>";
}
echo "</tr>";
//end table header
```

Interrogare il DB / 3

```
//create table body
echo "<tr>";
for ($i=0; $i<$num_fields; $i++)
{
    echo "<td valign = top>";
    echo mysql_field_type ($result, $i) . "<br> \n";
    echo "(" . mysql_field_len ($result, $i) . ")<br> \n";
    echo mysql_field_flags ($result, $i) . "<br> \n";
    echo "</td>";
}
echo "</tr>";
//end table body
echo "</table>";
```