

Introduzione a SQL

Corso di Laboratorio di Informatica

Prof. Dei Rossi, Leonardo Essam



Introduzione

Definizione 1: Linguaggio SQL

SQL è un linguaggio dichiarativo ad alto livello, standardizzato, per l'interazione con basi di dati relazionali, che permette la definizione delle strutture dati e l'accesso ai dati secondo il modello relazionale.

Domande:

1. Quale sotto-insieme di SQL permette la definizione di strutture dati?
2. Quale sotto-insieme di SQL permette l'accesso dai dati?

Gestione del server MySQL (1)

All'interno di XAMPP è integrato un server MySQL¹ e un'interfaccia Web per una gestione semplice dei vari database/tabelle.

Dalla console di XAMPP avviare sia Apache che MySQL:

Service	Module	PID(s)	Port(s)	Actions
<input type="checkbox"/>	Apache	10492 6896	80, 443	<input type="button" value="Stop"/> <input type="button" value="Admin"/> <input type="button" value="Config"/> <input type="button" value="Logs"/>
<input type="checkbox"/>	MySQL	23960	3306	<input type="button" value="Stop"/> <input type="button" value="Admin"/> <input type="button" value="Config"/> <input type="button" value="Logs"/>

¹Server: 10.4.32-MariaDB [UTF-8 Unicode (utf8mb4)]

Gestione del server MySQL (2)

Per accedere all'interfaccia Web di phpMyAdmin è sufficiente accedere al link:

<http://localhost/phpmyadmin/>

Osservare che phpMyAdmin *sembrerebbe* una sotto-cartella del nostro server Web, ma se si osserva attentamente il contenuto della cartella htdocs si può notare che la cartella /phpmyadmin non è presente.

Questo perché, onde evitare possibili eliminazioni accidentali, la cartella è stata spostata in un posto più sicuro e non direttamente accessibile; per poi essere resa disponibile all'utente tramite un Alias.

Gestione del server MySQL (3)

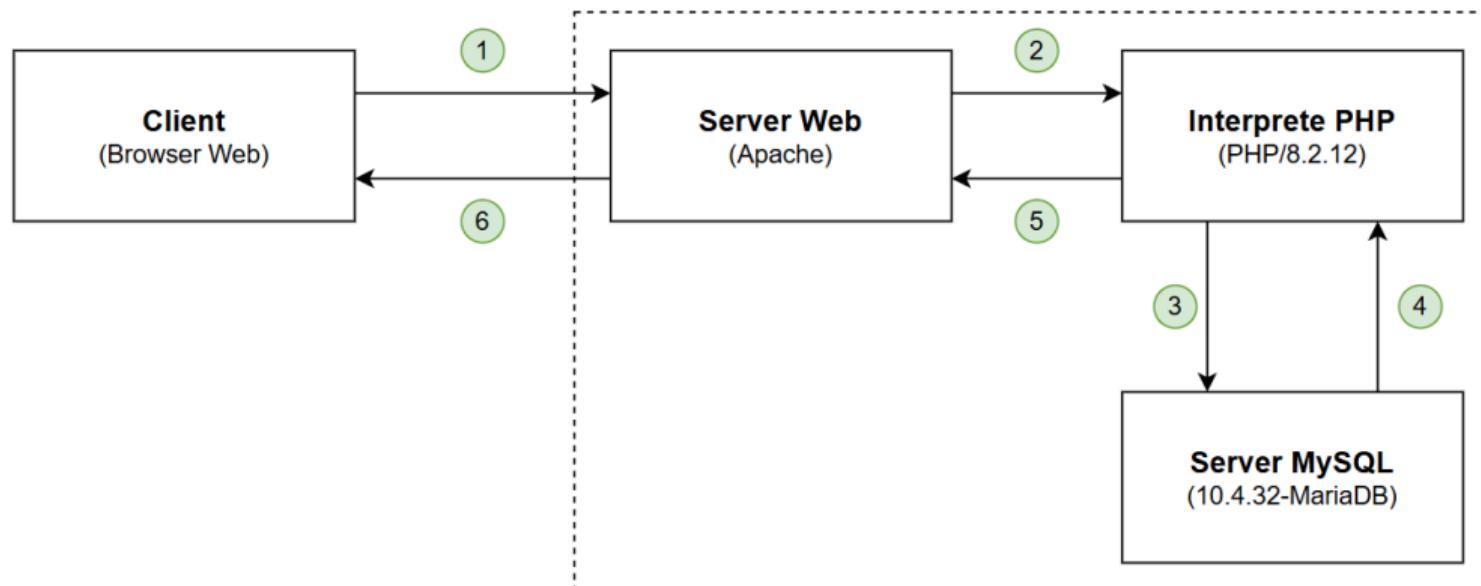
Ad esempio (solo per completezza):

```
Alias /5inb "D:/2025-2026/5INB/GPOI/ProgettiStudenti"

<Directory "D:/2025-2026/5INB/GPOI/ProgettiStudenti">
    Options Indexes FollowSymLinks
    AllowOverride All
    Require all granted
</Directory>
```

In questo modo la cartella D:/2025-2026/5INB/GPOI/ProgettiStudenti verrà resa disponibile tramite il server Web ma senza che la cartella si trovi all'interno di htdocs.

Schema di funzionamento



L'estensione MySQLi (1)

MySQLi, acronimo di MySQL Improved, è l'estensione di PHP progettata per consentire l'accesso ai database MySQL e MariaDB in modo moderno, sicuro ed efficiente.

È stata introdotta per superare i limiti della vecchia estensione `mysql`, oggi completamente rimossa, e rappresenta uno degli strumenti principali per lo sviluppo di applicazioni web dinamiche in PHP.

L'estensione MySQLi (2)

L'obiettivo principale di MySQLi è fornire un'interfaccia più robusta e sicura rispetto alle soluzioni precedenti.

L'estensione supporta nativamente i *prepared statement*, che consentono di separare la struttura delle query dai dati inseriti dall'utente, riducendo drasticamente il rischio di *SQL Injection*².

MySQLi supporta inoltre transazioni, gestione avanzata degli errori e comunicazione efficiente con il motore MySQL, sfruttando le funzionalità introdotte dalle versioni più recenti del database.

L'estensione MySQLi (3)

MySQLi può essere utilizzata secondo due stili di programmazione.

Lo stile procedurale utilizza funzioni globali ed è più vicino alla sintassi del C, risultando spesso più immediato per chi proviene da una programmazione tradizionale.

Lo stile orientato agli oggetti, invece, incapsula connessione e operazioni in oggetti PHP, risultando più leggibile, modulare e in linea con il PHP moderno.

Dal punto di vista funzionale le due modalità sono equivalenti: la scelta riguarda principalmente lo stile di progettazione del codice.

Connessione al database (1)

La connessione al database MySQL avviene tramite la creazione di una nuova istanza della classe mysqli.

Esempio 1: Istanza della classe mysqli

```
<?php
$connessione = new mysqli(
    "localhost",                      // Host
    "root",                            // User
    "",                                // Password
    "celin"                            // Database
);
?>
```

Connessione al database (2)

Il costruttore della classe `mysqli` presenta diversi argomenti:

1. L'indirizzo del server (`hostname`);
2. L'utente del server (`user`);
3. La password dell'utente (`password`);
4. Il nome del database (`database`).

Connessione al database (3)

L'indirizzo del server (hostname)

Può essere un nome host o un indirizzo IP, quando si passa null, il valore viene recuperato da `mysqli.default_host`.

Quando possibile, verranno utilizzate le pipe³ al posto del protocollo TCP/IP.

Il protocollo TCP/IP viene utilizzato se vengono forniti insieme un nome host e un numero di porta, ad esempio `localhost:3308`.

³In Windows, socket interni al sistema (come i socket locali su Linux).

Connessione al database (4)

L'utente del server (user)

Il nome utente MySQL oppure null per assumere il nome utente in base all'opzione `mysqli.default_user`.

La password dell'utente (password)

La password MySQL o null per assumere la password in base all'opzione `mysqli.default_pw`.

Connessione al database (5)

Il nome del database (database)

Il database predefinito da utilizzare quando si eseguono query o null.

Connessione al database (6)

Il costruttore della classe `mysqli` in ogni caso restituirà un oggetto di tipo `mysqli`, anche in caso di errore all'atto della connessione.

Viene lasciata quindi allo sviluppatore la responsabilità di gestire l'errore all'atto della connessione.

Connessione al database (7)

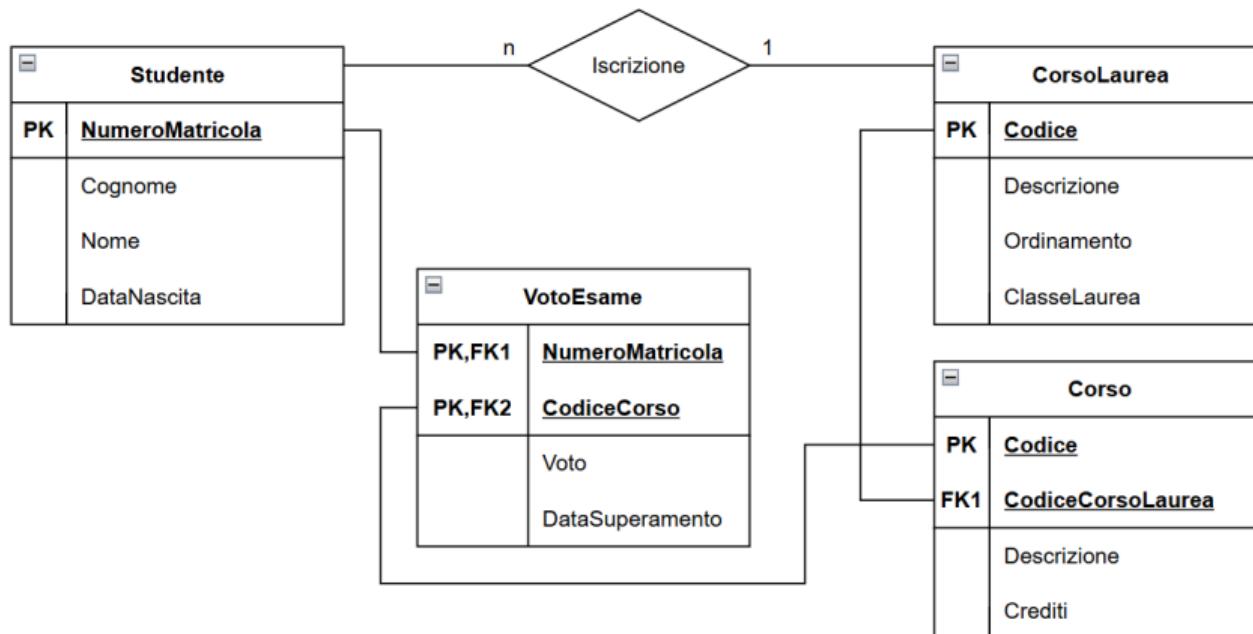
Esempio 2: Errore di connessione

```
<?php
$mysqli = new mysqli("localhost", "root", "", "celin");
if ($mysqli->connect_errno) {
    echo("Errore: " . $mysqli->connect_error);
}
?>
```

- `$mysqli->connect_errno`: codice di errore numerico dell'ultimo errore che si è verificato;
- `$mysqli->connect_error`: descrizione testuale dell'ultimo errore che si è verificato.

Primi esempi di query (1)

Si consideri il seguente schema:



Primi esempi di query (2)

Si realizzi un database chiamato Esse3 che contenga le tabelle di cui sopra.

Si inseriscano i seguenti dati nelle varie tabelle: ([link](#))

Allegare al compito pubblicato su Classroom una copia delle query SQL svolte.

Primi esempi di query (3)

Svolgere le seguenti query:

- Stampare i voti degli esami dello studente con matricola 20240001;
- Stampare il nome del corso di laurea dello studente con matricola 20240002;
- Stampare il codice, la descrizione e il numero di crediti dei corsi che lo studente con matricola 20240001 ha superato;
- Stampare il codice, la descrizione e il numero di crediti dei corsi che lo studente con matricola 20240002 non ha superato;
- Stampare i voti degli studenti per il corso con codice 145008 ordinati per data di superamento e per voto (decrescente).

Query in PHP (1)

Esempio 3: Esecuzione di una query

```
<?php
$mysqli = new mysqli("localhost", "root", "", "celin");
$result = $mysqli->query(
    "SELECT_*_FROM_Corso_WHERE_Codice_=145008"
);

$row = $result->fetch_assoc();
?>
```

Query in PHP (2)

Il metodo `query([...])` dell'oggetto `mysqli` ammette e richiede come parametro una stringa che rappresenta la query da inviare al server MySQL.

Restituisce a sua volta un oggetto di tipo `mysqli_result`.

Per accedere agli elementi, vale il metodo:

```
$row = $result->fetch_assoc();
```

In questo modo si va a leggere la prima tupla ricavata dalla query SQL.

Per iterare tutte le tuple ricavate, vale:

```
while ($row = $result->fetch_assoc()) {  
    // ...  
}
```

Documentazione

1. Classe `mysqli`: ([link](#))
2. Classe `mysqli_result`: ([link](#))

