

Unità **A2**

PHP e MySQL



CASE STUDY

Costruire una videoteca con le tabelle



FILE SORGENTI

In questa unità imparerai...

- A installare MySQL
- A creare e a gestire tabelle con MySQL
- A utilizzare MySQL con PHP

1 Gestione di database con MySQL

I DBMS relazionali sono lo strumento oggi universalmente utilizzato per implementare basi di dati di qualsiasi dimensione. Ogni linguaggio lato server deve necessariamente disporre delle primitive per l'interazione con un database al fine di consentire la realizzazione di programmi lato server completi e professionali.

PHP ha la possibilità di connettersi con un numero elevatissimo di database server (MySQL, PostgreSQL, Oracle, Microsoft SQL Server, Access, Sybase, Informix, mSQL, e altri ancora). In questa unità prendiamo in esame l'interazione tra **PHP** e **MySQL**.

MySQL è il gestore di database che si è affermato prepotentemente negli ultimi anni per la sua velocità, facilità d'utilizzo, stabilità, e per il fatto di essere **open source**. Il connubio tra PHP e MySQL consente di esprimere al meglio le potenzialità di entrambi gli strumenti.

In questa unità ci occuperemo delle tecniche utilizzate per:

- stabilire una connessione al server MySQL, prima da un client MySQL e in seguito da uno script PHP;
- visualizzare e manipolare i dati contenuti nel database.

Daremo come prerequisito la conoscenza sia pur superficiale – in questo contesto – di SQL, il linguaggio standard di manipolazione dei database relazionali.

Ricorda

PHP dispone di numerose librerie per la gestione dei differenti tipi di database. Inizialmente, per ognuno di essi, erano disponibili funzioni ad hoc; in seguito, con l'avvento della programmazione a oggetti, l'utilizzo di tali funzioni è stato sostituito dai metodi delle rispettive classi.

Installare e configurare MySQL

MySQL è stato pensato per essere utilizzato all'interno di un contesto web.

Fino a ora abbiamo utilizzato esclusivamente come server il server web; da ora in poi avremo la necessità di integrare all'interno del linguaggio PHP delle funzionalità che gestiscono le basi di dati e, per fare ciò, utilizzeremo MySQL come server DB. Tale server è già incluso all'interno della suite XAMPP: non ci resta che attivarlo dall'apposito tasto Start.

Ricorda

Il pannello di controllo di XAMPP dispone di differenti pulsanti per l'avvio e l'arresto dei differenti server.

Ricorda

Spesso ci si dimentica di **avviare** il server MySQL. Il mancato avvio causerà errori durante l'esecuzione, che saranno segnalati dagli script PHP.

Prima di entrare nel merito dell'interazione fra PHP e le basi di dati, prendiamo confidenza con il server MySQL e le istruzioni SQL. Per fare ciò, possiamo utilizzare l'interfaccia testuale e quella grafica: le esamineremo entrambe.

Una volta attivato il server MySQL, dobbiamo utilizzare un client per accedere al DBMS, cosa che possiamo fare attraverso due approcci:

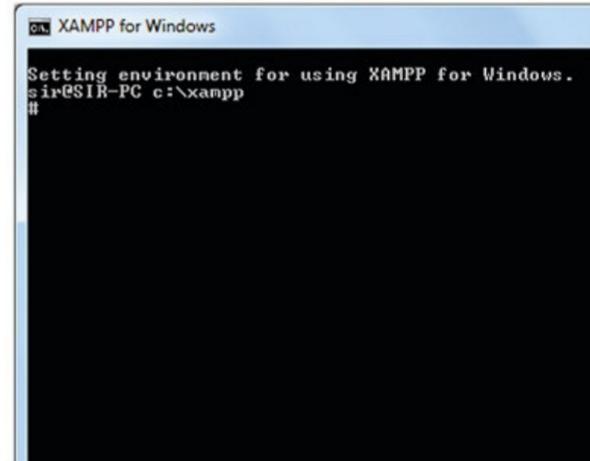
- client esclusivo MySQL, che consente il dialogo esclusivamente attraverso comandi SQL;
- client incorporato nel server web (Apache), attraverso PHP, nel caso di realizzazione di pagine web dinamiche. Questa soluzione prevede naturalmente l'ulteriore attivazione del server web Apache.

La prima soluzione prevede l'utilizzo di un ambiente testuale in cui impartire le istruzioni a riga di comando; tutti i sistemi operativi dispongono di un'interfaccia testuale denominata **shell**.

Facendo riferimento al pannello di controllo di XAMPP, notiamo il pulsante  **Shell**, la cui attivazione apre una finestra nera in attesa dell'inserimento dei comandi.

Non resta che impartire i comandi giusti per attivare il client MySQL:

```
mysql -h <NomeHost> -u <NomeUtente> -p <Password>
```



```
mysql -h <NomeHost> -u <NomeUtente> -p <Password>
```

- L'opzione *-h* richiede di specificare il nome dell'host o l'indirizzo IP del computer su cui è in esecuzione il server MySQL;
- l'opzione *-u* richiede di specificare il nome dell'utente riconosciuto dal sistema;
- l'opzione *-p* richiede che venga specificata la password di accesso.

Se il server MySQL è presente sullo stesso computer del client, non è necessario specificare l'opzione *-h* e basterà inserire il comando come utente *root* (utente amministratore).

Se al momento dell'installazione non è stata specificata alcuna password per tale utente, il comando diventa:

```
mysql -u root
```

Rifletti

Osservando la figura, si può notare che viene fornito un messaggio di benvenuto da MariaDB, nonostante il comando impartito sia stato *mysql*. MariaDB è semplicemente una particolare versione del DBMS MySQL utilizzato all'interno di XAMPP.

Per **operazioni di creazione di database e tabelle**, occorre connettersi come **root** o come **utente** con permessi equivalenti.

Se il server MySQL è avviato correttamente, il programma client MySQL andrà in esecuzione visualizzando un prompt come quello mostrato di seguito.

```
sir@SIR-PC c:\xampp  
# mysql -u root  
Welcome to the MariaDB monitor. Commands end with ; or \g.  
Your MariaDB connection id is 152  
Server version: 10.1.33-MariaDB mariadb.org binary distribution  
  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
MariaDB [(none)]> _
```

Ogni comando deve terminare con il punto e virgola (;) e premendo *Invio*. Se premiamo *Invio* senza aver inserito il punto e virgola, aggiungeremo una nuova riga contenente il carattere -> a indicare che il comando non è ancora terminato. In tale situazione, si può sempre terminare il comando digitando, anche successivamente, il punto e virgola. Un primo importante comando è **HELP**, che ci permette di vedere la sintassi di alcuni comandi di uso comune.

Per uscire dal client si può digitare **QUIT** oppure **EXIT** oppure **\q**.

Per **vedere la lista dei database presenti**, digitiamo:

```
SHOW DATABASES;
```

Nella lista appaiono inizialmente alcuni database di sistema, in seguito appariranno anche quelli creati dall'utente.

Per **creare un nuovo database**, utilizziamo la sintassi:

```
CREATE DATABASE <NomeDatabase>;
```

dove *NomeDatabase* è il nome che vogliamo assegnare al nuovo database.

Per **creare il database** relativo a un negozio online utilizzeremo:

```
CREATE DATABASE negozio;
```

Rifletti

Il client MySQL non è *case sensitive*; pertanto, i comandi possono essere inseriti sia in minuscolo sia in maiuscolo.

XAMPP for Windows - mysql -u root

```
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| phpmyadmin |
| test |
+-----+
5 rows in set (0.00 sec)

MariaDB [(none)]> -
```

Per **cancellare un database esistente**, utilizziamo la sintassi:

```
DROP DATABASE <NomeDatabase>;
```

dove <NomeDatabase> è il nome del database che vogliamo cancellare.

Per **selezionare un database tra quelli esistenti**, utilizziamo la sintassi:

```
USE <NomeDatabase>;
```

dove <NomeDatabase> è il nome del database che vogliamo selezionare. Per selezionare il database del negozio, utilizzeremo:

```
USE negozio;
```

2 Creare e gestire le tabelle in MySQL

Creare una tabella in un database

Una volta creato e selezionato il database *negozi*, possiamo creare le tabelle PRODOTTO e CATEGORIA utilizzando l'istruzione CREATE TABLE di SQL:

```
CREATE TABLE CATEGORIA (
    CodCat CHAR(2) NOT NULL,
    DescrCat CHAR(20) NOT NULL,
    PRIMARY KEY (CodCat)
);

CREATE TABLE PRODOTTO (
    CodProd CHAR(5) NOT NULL,
    CodCat CHAR(2) NOT NULL,
    DescrProd CHAR(30) NOT NULL,
    PrezzoProd DEC(5, 2) NOT NULL,
    QtaRes INT(3),
    PRIMARY KEY (CodProd),
    FOREIGN KEY (CodCat)
        REFERENCES Categoria (CodCat)
        ON DELETE RESTRICT
        ON UPDATE CASCADE
);
```

The screenshot shows a Windows command-line window titled "XAMPP for Windows - MYSQL -u root". The MySQL prompt is at the top. The user has run the command "use negozio;" followed by "show tables;". The output shows a table structure with two rows: "categoria" and "prodotto". The entire output is as follows:

```
MariaDB [none]> use negozio;
Database changed
MariaDB [negozio]> show tables;
+-----+
| Tables_in_negozio |
+-----+
| categoria          |
| prodotto           |
+-----+
2 rows in set (0.00 sec)

MariaDB [negozio]>
```

Puoi sicuramente notare che abbiamo sviluppato l'istruzione CREATE su più righe e l'abbiamo fatta terminare con il punto e virgola.

Nella tabella PRODOTTO abbiamo specificato, oltre alla chiave primaria, la chiave esterna *CodCat* che fa riferimento alla tabella CATEGORIA.

Visualizzare le tabelle di un database

Per visualizzare l'elenco delle tabelle presenti in un database, utilizziamo il comando seguente:

```
SHOW TABLES;
```

La figura qui a sinistra mostra che cosa viene visualizzato quando eseguiamo il comando per il database *negozio*.

Visualizzare la struttura di una tabella

Per visualizzare la struttura di una tabella, utilizziamo il comando DESCRIBE con la seguente sintassi:

```
DESCRIBE <NOMETABELLA>;
```

Per visualizzare la struttura della tabella *prodotto*, digiteremo:

```
DESCRIBE prodotto;
```

ottenendo quanto è visualizzato nella figura.

The screenshot shows a terminal window titled "XAMPP for Windows - mysql - u root". The command "describe prodotto;" is entered, followed by its output. The output is a table with columns: Field, Type, Null, Key, Default, and Extra. The rows represent the columns of the 'prodotto' table, with types like char(5), char(2), decimal(5,2), and int(3). Primary key (PRI) and multiple key (MUL) are indicated in the Key column. The 'QtaRes' column has YES in the Null column and NULL in the Default column.

Field	Type	Null	Key	Default	Extra
CodProd	char(5)	NO	PRI	NULL	
CodCat	char(2)	NO	MUL	NULL	
DescrProd	char(30)	NO		NULL	
PrezzoProd	decimal(5,2)	NO		NULL	
QtaRes	int(3)	YES		NULL	

5 rows in set (0.13 sec)

MariaDB [negozi]> _

Lo stesso risultato si può ottenere con il comando SHOW COLUMNS:

```
SHOW COLUMNS FROM prodotto;
```

Cancellare una tabella

Per cancellare una tabella da un database, digitiamo il comando SQL:

```
DROP TABLE <NOMETABELLA>;
```

<NOMETABELLA> è il nome della tabella da eliminare.

Per eliminare, ad esempio, la tabella TEMP1, digiteremo:

```
DROP TABLE TEMP1;
```

Copiare una tabella

Per copiare una tabella da una già esistente, digitiamo il comando SQL:

```
CREATE TABLE <NOMENUOVATABELLA> SELECT * FROM <NOMETABELLAORIGINARIA>;
```

<NOMENUOVATABELLA> è il nome della tabella da creare, <NOMETABELLA-ORIGINARIA> è il nome della tabella da cui copiare il contenuto.

Per copiare, ad esempio, la tabella VECCHICLIENTI nella tabella NUOVICLIENTI, digiteremo quanto segue:

```
CREATE TABLE NUOVICLIENTI SELECT * FROM VECCHICLIENTI;
```

Ricorda

Tieni presente che il linguaggio SQL non è *case sensitive*; pertanto scrivere PRODOTTO o prodotto è indifferente.

Inserire e modificare i dati presenti in una tabella

In modo interattivo, utilizzando le istruzioni SQL, è possibile effettuare qualsiasi operazione di manipolazione sui dati presenti in una tabella, dall'inserimento, alla modifica, alla cancellazione del contenuto, e così via. Ad esempio, per inserire un nuovo prodotto nella tabella PRODOTTO, utilizzeremo l'istruzione INSERT di SQL e digiteremo:

```
INSERT INTO PRODOTTO (CodProd, CodCat, DescrProd, PrezzoProd, QtaRes)
VALUES ("P0025", "08", "Memoria RAM", 120, 5);
```

Query di aggiornamento e query di interrogazione

È opportuno distinguere tra:

- **query di aggiornamento** (del tipo: INSERT, UPDATE, DELETE), che utilizziamo per inserire un nuovo prodotto nel nostro negozio;
- **query di interrogazione** (del tipo: SELECT, SHOW, EXPLAIN, DESCRIBE), che utilizziamo per visualizzare i prodotti presenti nel nostro negozio.

3 phpMyAdmin

phpMyAdmin è uno strumento software gratuito scritto in PHP, destinato a gestire l'amministrazione di database MySQL attraverso l'interfaccia web.

phpMyAdmin supporta una vasta gamma di operazioni su MySQL: gestione dei database, tabelle, colonne, relazioni, indici, utenti, permessi, ecc. Il tutto può essere eseguito tramite l'interfaccia utente grafica. Inoltre phpMyAdmin consente di eseguire direttamente qualsiasi istruzione SQL.

Caratteristiche principali

Interfaccia web intuitiva.

Supporto per la maggior parte delle caratteristiche di MySQL:

- sfogliare ed eliminare database, tabelle, viste, campi e indici;
- creare, copiare, eliminare, rinominare e modificare database, tabelle, campi e indici;
- eseguire, modificare e memorizzare qualsiasi istruzione SQL;
- gestire MySQL account utente e privilegi;
- gestire procedure e trigger.

Può importare dati da CSV e SQL.

Può esportare i dati in vari formati: CSV, SQL, XML, PDF, ISO / IEC 26300 - OpenDocument Text e foglio di calcolo, Word e altri.

Consente di amministrare più server.

Interfaccia phpMyAdmin

All'avvio di phpMyAdmin (digitando nel browser `http://localhost/phpmyadmin`) ci troveremo di fronte la schermata di un applicativo a interfaccia web che ci consentirà di gestire i nostri database.

Per accedere all'applicativo è necessario essere un utente del sistema; per default l'utente amministratore di phpMyAdmin è contraddistinto dal termine *root* privo di password. Successivamente sarà possibile creare nuovi utenti, ai quali concedere privilegi per i differenti database da implementare. In questa sede tralasseremo aspetti quali la gestione degli utenti e aspetti avanzati, concentrando la nostra attenzione esclusivamente sulla gestione di database e di tabelle, sull'inserimento e sull'aggiornamento di dati e sulle esecuzioni di *query sql*.

La figura seguente mostra la pagina di apertura (*home*); nella colonna a sinistra sono elencati i database presenti, la parte centrale è contraddistinta in alto da un menu orizzontale per accedere alle varie funzionalità e da alcuni elementi di impostazione, quali password, lingua, dimensione caratteri, ecc., mentre nella parte destra compaiono alcune informazioni che riguardano il web server e il server del database.



localhost / 127.0.0.1 | ph Pasquale

localhost/phpmyadmin/ Server: 127.0.0.1

phpMyAdmin Database SQL Stato Account utenti Esporta Importa Impostazioni Replicazione Variabili Più

Nuovo information_schema mysql negozio performance_schema phpmyadmin test

Impostazioni generali Collation della connessione del server: utf8mb4_unicode_ci

Impostazioni di presentazione Lingua - Language: Italiano - Italian Tema: pmahomme Dimensione font: 82% Ulteriori impostazioni

Server del Database

- Server: 127.0.0.1 via TCP/IP
- Tipo di server: MariaDB
- Connessione Server: SSL inattivo
- Versione del server: 10.1.33-MariaDB - mariadb.org binary distribution
- Versione protocollo: 10
- Utente: root@localhost
- Codifica caratteri del server: UTF-8 Unicode (utf8)

Web server

- Apache/2.4.33 (Win32) OpenSSL/1.1.0h PHP/7.2.6
- Versione del client del database: libmysql - mysqlnd 5.0.12-dev - 20150407 - \$Id: 38fea24f2847fa7519001be390c98ae0acafe387 \$
- Estensione PHP: mysqli curl mbstring
- Versione PHP: 7.2.6

phpMyAdmin

- Informazioni sulla versione: 4.8.1, versione stabile più recente: 4.8.2
- Documentazione
- Home page ufficiale di phpMyAdmin
- Contribuisci
- Ricevi aiuto
- Lista dei cambiamenti
- Licenza

Console

The screenshot shows the phpMyAdmin interface with the title 'phpMyAdmin'. Below the title is a horizontal menu bar with icons for Home, Import, Export, Structure, SQL, and Help. Underneath the menu is a dropdown menu labeled '(Tabelle recenti) ...'. The main area displays a tree view of databases and their structures. Databases listed include: assenze, biblio (circled with a red circle), discografia, Nuova, afferisce, album, autore, brano, Campi, New, Cod_brano, Durata, Genere, Musicista, Titolo, contiene, genere, musicista, scrive, dogana, film, information_schema, mysql, performance_schema, test, and univ. The 'Campi' node under 'brano' is expanded, showing its sub-tables: assenze, biblio, discografia, dogana, film, information_schema, mysql, performance_schema, test, and univ.

Il simbolo + accanto a ogni database consente di espandere il database, e cioè di visualizzare le tabelle che lo compongono. È possibile espandere anche le tabelle, per visualizzare possibili campi e indici presenti.

Creare un database

La creazione di un database è un'operazione semplice; dal menu orizzontale si accede alla voce *Database*: una casella di testo indica di inserire il nome del nuovo database.

The screenshot shows the phpMyAdmin interface with the title 'phpMyAdmin' and the URL 'localhost'. Below the title is a horizontal menu bar with tabs: Database (circled with a red circle), SQL, Stato, Utenti, and Esporta. The 'Database' tab is active. On the left, there is a sidebar with the same list of databases as the previous screenshot. The main content area has a heading 'Database' and a sub-section 'Crea un nuovo database' with a text input field containing 'istruzione'. Below this, a table lists existing databases with their replication status and privilege control options. The table includes:

Database	Replicazione master
assenze	✓ Replicato <input type="checkbox"/> Controlla i privilegi
biblio	✓ Replicato <input type="checkbox"/> Controlla i privilegi
discografia	✓ Replicato <input type="checkbox"/> Controlla i privilegi
dogana	✓ Replicato <input type="checkbox"/> Controlla i privilegi
film	✓ Replicato <input type="checkbox"/> Controlla i privilegi
information_schema	✓ Replicato <input type="checkbox"/> Controlla i privilegi
mysql	✓ Replicato <input type="checkbox"/> Controlla i privilegi
performance_schema	✓ Replicato <input type="checkbox"/> Controlla i privilegi
test	✓ Replicato <input type="checkbox"/> Controlla i privilegi
univ	✓ Replicato <input type="checkbox"/> Controlla i privilegi

Totale: 10

Il nuovo database sarà visibile all'interno dell'elenco complessivo dei database.

Creare una tabella

Al click sul nome del database viene attivata la pagina che chiede la descrizione della sua struttura: bisogna creare e gestire le tabelle. Si inserisce il nome della tabella e il numero dei campi: il sistema predisporrà l'inserimento dei campi.

Per ciascun campo è necessario compilare alcuni aspetti quali il nome, la tipologia del valore, l'eventuale lunghezza e la chiave primaria (dicitura PRIMARY sulla colonna indice).



Le successive tabelle possono essere in alternativa costruite direttamente dall'elenco di sinistra, accedendo alla voce *Nuova* del relativo database.

Nome tabella: SCUOLA Aggiungi 1 campo/ Esegui

Struttura

Nome	Tipo	Lunghezza/Valori	Predefinito	Codifica caratteri	Attributi	Null	Indice
CodScuola	VARCHAR	20	Nessuno			<input checked="" type="checkbox"/>	PRIMARY
NomScuola	VARCHAR	30	Nessuno			<input type="checkbox"/>	...
Citta	VARCHAR	30	Nessuno			<input type="checkbox"/>	...

Commenti alla tabella:

Motore di Memorizzazione: InnoDB

Codifica caratteri:

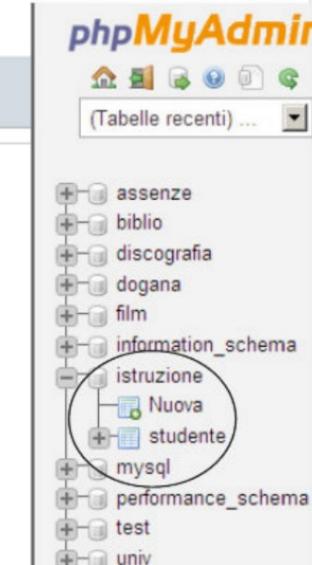
Definizione PARTITION:

Chiave primaria

Chiavi esterne

Supponiamo di voler realizzare il database i cui schemi, concettuale e logico, sono rappresentati di seguito.

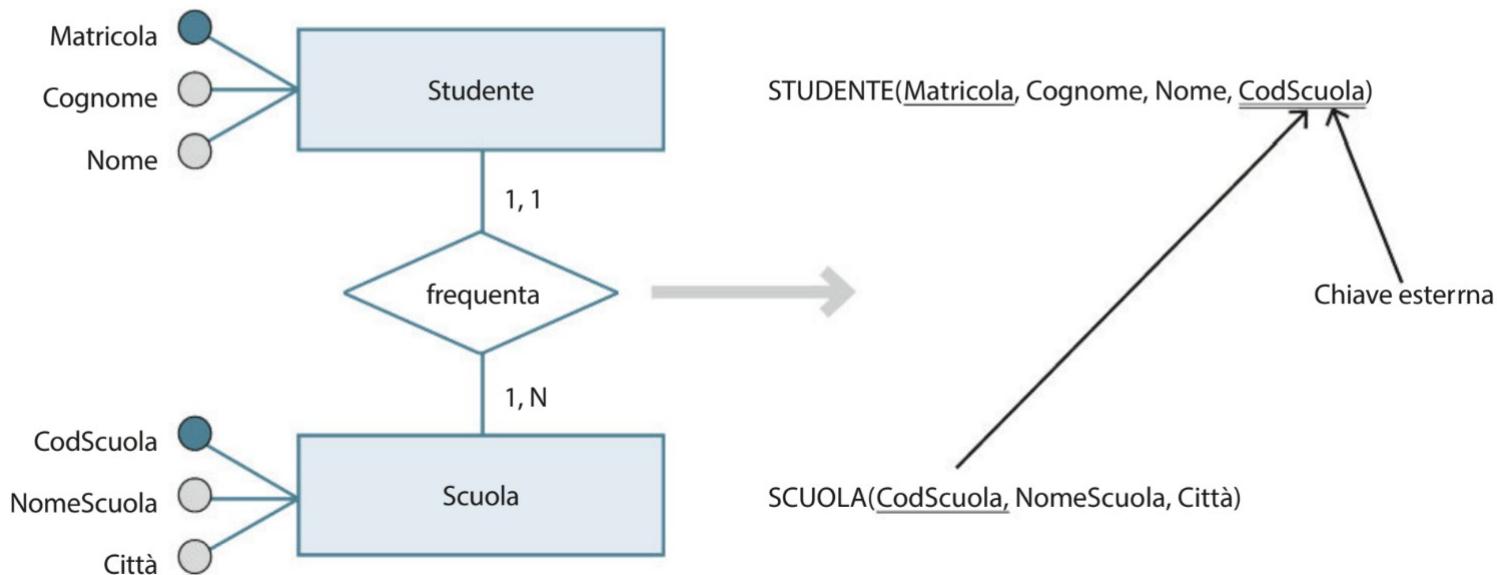
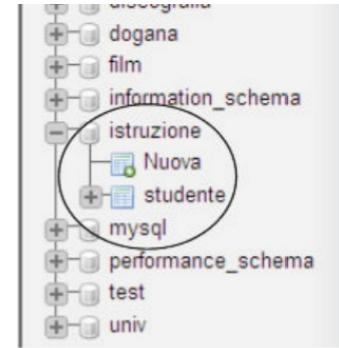
Nasce pertanto l'esigenza di inserire all'interno della tabella STUDENTE il campo CodScuola come chiave esterna in riferimento alla tabella SCUOLA.



Chiavi esterne

Supponiamo di voler realizzare il database i cui schemi, concettuale e logico, sono rappresentati di seguito.

Nasce pertanto l'esigenza di inserire all'interno della tabella STUDENTE il campo CodScuola come chiave esterna in riferimento alla tabella SCUOLA.



In fase di definizione dei campi è necessario selezionare, come voce di *Indice*, il valore **INDEX** in corrispondenza del campo *CodScuola* il quale, come si evince dallo sche-

localhost > istruzione

Struttura SQL Cerca Query da esempio Esporta Importa Operazioni Privilegi Routine Più

Nome tabella: STUDENTE Aggiungi 1 campo/ Esegui

Struttura

Nome	Tipo	Lunghezza/Valori	Predefinito	Codifica caratteri	Attributi	Null	Indice
Matricola	VARCHAR	15	Nessuno			<input type="checkbox"/>	PRIMARY
Cognome	VARCHAR	30	Nessuno			<input type="checkbox"/>	
Nome	VARCHAR	20	Nessuno			<input type="checkbox"/>	
CodScuola	VARCHAR	20	Nessuno			<input checked="" type="checkbox"/> INDEX	

Commento alla tabella:

Motore di Memorizzazione: InnoDB

Codifica caratteri:

Definizione PARTITION:

Salva

Valore INDEX

ma logico, rappresenta la chiave esterna da collegare alla chiave primaria della tabella SCUOLA.

I nomi delle tabelle vengono riportati dal sistema sempre con le lettere minuscole, per convenzione e rispettando le indicazioni dello schema logico utilizzato nel testo; noi faremo riferimento ai nomi delle tabelle con le lettere maiuscole.

A questo punto è possibile stabilire la relazione tra le due tabelle; pertanto attraverso la struttura della tabella STUDENTE è possibile accedere alle relazioni.

localhost / 127.0.0.1 / ne... Pasquale X

localhost/phpmyadmin/tbl_structure.php?db=negozi&table=studente

phpMyAdmin

Server: 127.0.0.1 » Database: negozi » Tabella: studente

Mostra Struttura SQL Cerca Inserisci Esporta Importa Privilegi Operazioni Più

Struttura della tabella Vista relazioni

#	Nome	Tipo	Codifica caratteri	Attributi	Null	Predefinito	Commenti	Extra	Azione
1	Matricola	varchar(15)	latin1_swedish_ci		No	Nessuno			Modifica Elimina Più
2	cognome	varchar(30)	latin1_swedish_ci		No	Nessuno			Modifica Elimina Più
3	nome	varchar(30)	latin1_swedish_ci		No	Nessuno			Modifica Elimina Più
4	CodScuola	varchar(20)	latin1_swedish_ci		No	Nessuno			Modifica Elimina Più

Selezione tutto Se selezionati: Mostra Modifica Elimina Primaria Unica Indice Testo completo

Aggiungi ai campi centrali Rimuovi dai campi centrali

Stampa Proponi la struttura della tabella Controllo tabella Muovi campi Normalizza

Aggiungi 1 campo dopo CodScuola Esegui

Indici

Azione	Chiave	Tipo	Unica	Compresso	Colonna	Cardinalità	Codifica caratteri	Null	Commenti
Modifica Elimina	PRIMARY	BTREE	Si	No	Matricola	0	A	No	

Crea un indice su 1 campi Esegui

In corrispondenza della colonna CodScuola (ricorda che siamo all'interno della tabella STUDENTE) è possibile selezionare il campo della tabella SCUOLA che funge da chiave primaria (anche in questo caso il nome del campo è CodScuola).

The screenshot shows the phpMyAdmin interface for the 'negozio' database, specifically the 'studente' table. The 'Struttura della tabella' (Table Structure) tab is selected. In the 'Relazioni' (Relationships) section, there is a 'Vincoli della foreign key' (Foreign Key Constraints) panel. This panel shows a constraint named 'CodScuola' that links the 'scuola' column in the 'studente' table to the 'scuola' column in the 'scuola' table via the 'negozio' database. The 'ON DELETE' and 'ON UPDATE' actions are both set to 'RESTRICT'. There are three dropdown menus for selecting columns: 'Colonna' (Column), 'Database' (Database), and 'Tabella' (Table). Arrows point from the text 'Colonna' to each of these three dropdowns. Below the main panel, there are buttons for adding a new constraint ('+ Aggiungi vincolo') and viewing internal relationships ('+ Relazioni interne'). At the bottom, there are buttons for previewing SQL code ('Anteprima SQL') and saving changes ('Salva').

Inserimento dati

Create le tabelle, e quindi le rispettive strutture, è arrivato il momento di popolarle, inserendo i dati. Facendo click sulla tabella si attiva il relativo menu orizzontale; attraverso la voce *Inserisci* è possibile immettere i dati nei rispettivi campi.

The screenshot shows the phpMyAdmin interface with the following details:

- Left Panel (Database Structure):** Shows the database structure with the 'istruzione' schema selected. Inside 'istruzione', the 'Nuova' table is highlighted with a green circle. Other tables shown are assenze, biblio, discografia, dogana, film, information_schema, scuola, and studente.
- Top Bar:** The title bar shows 'localhost > istruzione > scuola'. The tabs include Mostra, Struttura, SQL, Cerca, Inserisci (which is highlighted with a blue circle), Esporta, and Importa.
- Table Insertion Form (Top Section):** This section is for the 'scuola' table. It has columns: CodScuola (varchar(20)), NomScuola (varchar(30)), and Citta (varchar(30)). The 'NomScuola' field contains 'I.T. E. Fermi'. The 'Citta' field contains 'Roma'. The 'CodScuola' field contains 'XYZ45'. An arrow points from the 'NomScuola' field to the 'I.T. E. Fermi' label. Another arrow points from the 'Citta' field to the 'Roma' label. A third arrow points from the 'Esegui' button to the 'NomScuola' field.
- Table Insertion Form (Bottom Section):** This section is for the 'Nuova' table. It has columns: CodScuola (varchar(20)) and NomScuola (varchar(30)). Both fields are currently empty.
- Checkboxes:** A checkbox labeled 'Ignora' is checked.

SQL

Abbiamo visto, durante l'utilizzo di phpMyAdmin, che per ogni comando impartito il sistema ha visualizzato la relativa istruzione SQL; per esempio, quella rappresentata in figura è relativa all'inserimento di una tupla all'interno della tabella SCUOLA.



The screenshot shows the phpMyAdmin interface with the following details:

- Left sidebar:** Shows the database structure with databases like assenze, biblio, discografia, dogana, film, information_schema, istruzione, scuola, and studente.
- Top menu:** Includes links for Mostra, Struttura, SQL, Cerca, Inserisci, Esporta, Importa, Operazioni, and Trigger.
- Message bar:** A green bar at the top indicates "1 riga inserita." (1 row inserted).
- SQL tab content:** Displays the SQL query used for insertion:

```
INSERT INTO `istruzione`.`scuola` (
    `CodScuola`,
    `NomScuola`,
    `Citta`
)
VALUES (
    'XYZ45', 'I.T. E. Fermi', 'Roma'
);
```
- Bottom right:** Buttons for [Modifica] and [Crea il codice PH].

Accedendo, pertanto, al link SQL, è possibile utilizzare un ambiente all'interno del quale costruire le nostre query indipendentemente che esse siano del tipo DDL (Data Definition Language, query di struttura) o DML (Data Manipulation Language, query sui dati).

Se volessimo inserire un'altra tupla all'interno della tabella SCUOLA, in alternativa al percorso visto prima, potremmo realizzare direttamente l'istruzione SQL tramite il comando INSERT.

phpMyAdmin

Server: 127.0.0.1 > Database: negozio > Tabella: studente

Mostra Struttura SQL Cerca Inserisci Esporta Importa Privilegi Operazioni Più

Esegui la/le query SQL sulla tabella negozio.studente:

```
1 INSERT INTO `studente`(`Matricola`, `cognome`, `nome`, `cod_scuola`) VALUES ([value-1],[value-2],[value-3],[value-4])
```

Campi

Matricola
cognome
nome
cod_scuola

SELECT* SELECT INSERT UPDATE DELETE Cancellà Formato

Ottieni la query auto-memorizzata

Parametri Bind

Aggiungi ai preferiti questa query SQL:

[Delimitatori :] Mostra di nuovo questa query Nascondi riquadro query SQL Annulla le modifiche quando hai finito Abilita i controlli sulle chiavi esterne Esegui

Viene riproposta un'istruzione SQL all'interno della quale dobbiamo aver cura di inserire i valori corretti relativi al campo VALUES, rispettando l'ordine dei campi; ricordiamo che nel caso di valori di tipo CHAR/VARCHAR, i valori devono essere posti tra apici.