

1. L'ambiente phpMyAdmin

Oltre alla modalità di interazione tramite linea comandi, è possibile utilizzare un database MySQL attraverso programmi che forniscono un'interfaccia grafica. Tra questi il più diffuso è il programma **phpMyAdmin**, che si chiama così perché il suo codice è scritto nel linguaggio *Php*.

Questo sistema è un software **free** (libero), distribuito con la licenza **GNU** (*General Public License*), che consente di amministrare le basi di dati create con MySQL. Si può prelevare la versione più recente di questo software dal sito

www.phpmyadmin.net

phpMyAdmin è composto da un insieme di script *Php* che permettono di gestire un server MySQL tramite un'interfaccia composta da pagine Web. Questo significa che per poter usare questo programma, di cui esistono sia le versioni Linux che Windows, occorre installare l'interprete del linguaggio *Php* e attivare un Web server, Apache per Linux o IIS (*Internet Information Services*) per Windows.

L'interprete del **linguaggio Php** può essere scaricato dal sito

www.php.net

Per il sistema operativo Windows è consigliabile scaricare la versione *Installer*, che installa l'interprete *Php* e configura anche il Web server IIS.

L'installazione di phpMyAdmin richiede di scaricare il pacchetto e di scompattarlo in una directory (per esempio, *phpMyAdmin*) nella root del Web server, tipicamente */var/www/html* per il sistema operativo Linux e *\inetpub\wwwroot* per Windows: usando l'interfaccia grafica, basta fare doppio clic sull'icona del file.

Occorre poi modificare, con un editor di testi, il file *config.inc.php3*, specificando la directory di installazione nella linea

```
$cfg['PmaAbsoluteUri'] = 'http://localhost/phpMyAdmin/'
```

(si faccia attenzione alla barra finale).

Si deve quindi avviare il Web server e il server MySQL:

- in Linux con i comandi:

```
/etc/init.d/httpd start  
/etc/init.d/mysqld start
```

- in Windows con *Pannello di controllo*, *Strumenti di amministrazione*, *Internet Information Services*, *Siti Web*, *Default Web Site* e con il comando per l'avvio del server MySQL

```
c:\mysql\bin\mysqld
```

Dopo aver installato il programma e aver avviato i server, phpMyAdmin viene aperto da browser scrivendo nella casella Indirizzo:

```
http://localhost/phpMyAdmin/
```

Per Windows occorre essere certi che tra le pagine avviate automaticamente dal browser sia compresa anche la pagina del tipo **index.php** (*Pannello di controllo*, *Strumenti di amministrazione*, *Internet Information Services*, *Siti Web*, tasto destro su *Default Web Site*, scelta *Proprietà*, scheda *Documenti* nella finestra *Proprietà*, clic sul pulsante *Aggiungi*, inserire *index.php*).

In caso contrario, occorre scrivere l'indirizzo per esteso:

```
http://localhost/phpMyAdmin/index.php
```

Nel programma **phpMyAdmin** si possono attivare alcune funzionalità aggiuntive per la gestione dei database relazionali **MySQL**: in particolare la memorizzazione di query in vista di un riutilizzo successivo, l'impostazione delle associazioni tra le tabelle e la compilazione del dizionario del database in formato *pdf*.

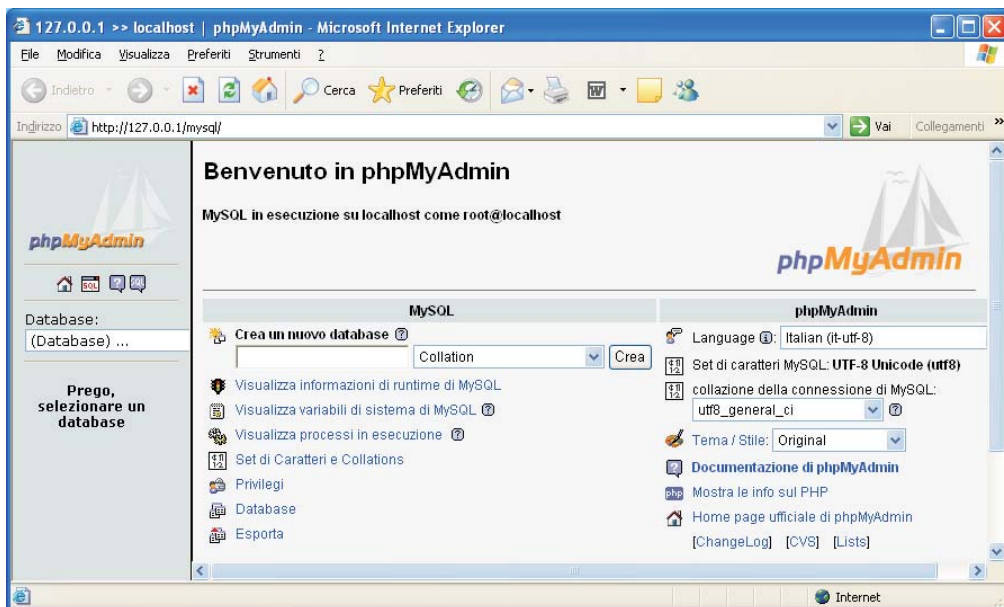
Per attivare queste opzioni occorre creare in MySQL un database di supporto contenente alcune tabelle predefinite. Le operazioni da compiere sono due:

1. modificare, con un editor di testo, le seguenti righe del file **config.inc.php** che si trova nella directory di installazione di phpMyAdmin, scrivendo tra apici i nomi delle tabelle di servizio:

```
$cfg['Servers'][$i]['pmadb'] = 'phpmyadmin';  
$cfg['Servers'][$i]['bookmarktable'] = 'pma_bookmark';  
$cfg['Servers'][$i]['relation'] = 'pma_relation';  
$cfg['Servers'][$i]['table_info'] = 'pma_table_info';  
$cfg['Servers'][$i]['table_coords'] = 'pma_table_coords';  
$cfg['Servers'][$i]['pdf_pages'] = 'pma_pdf_pages';  
$cfg['Servers'][$i]['column_info'] = 'pma_column_info';  
$cfg['Servers'][$i]['history'] = 'pma_history';
```

2. eseguire lo script **create_tables.sql** che si trova nella sottodirectory **scripts** all'interno della directory di installazione di phpMyAdmin: l'esecuzione può essere attivata dalla linea comandi di MySQL con il comando **source**, oppure da phpMyAdmin stesso.

La figura mostra la pagina di accesso a phpMyAdmin. Essa è suddivisa in due frame: quello di sinistra, di minore dimensione, è presente in tutte le pagine di phpMyAdmin, mentre quello di destra cambia in funzione della pagina visualizzata.



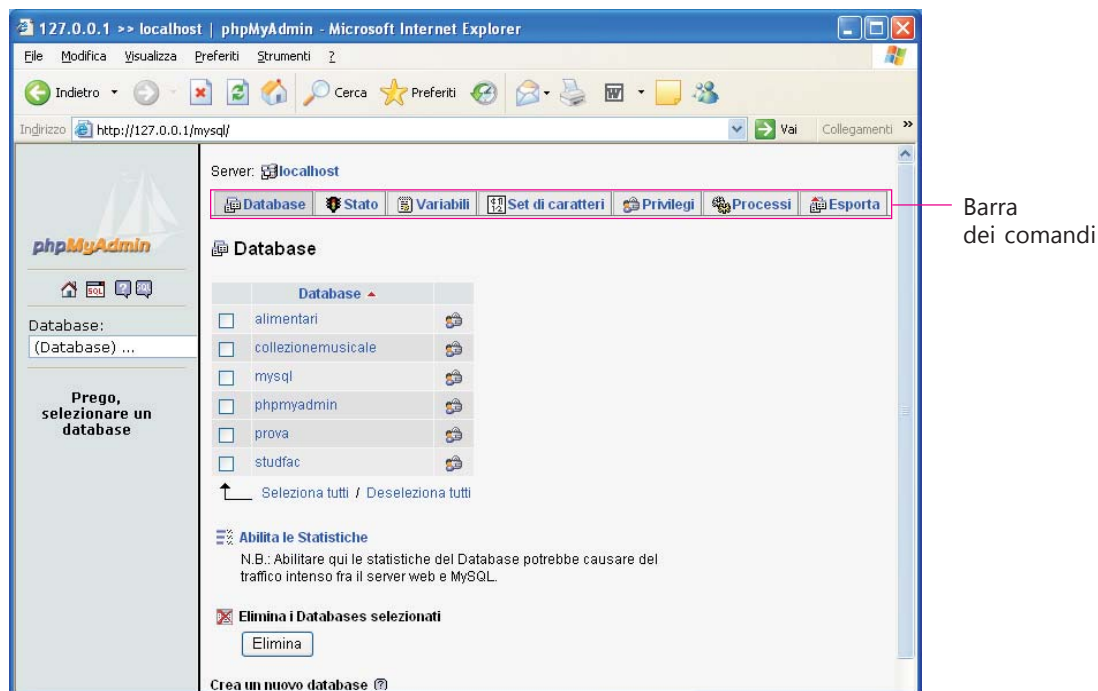
Il frame di sinistra contiene nella parte centrale la casella combinata **Database** che consente di aprire un database di MySQL, scegliendolo da un elenco di database disponibili sul server.

Il numero tra parentesi che appare accanto al nome di ogni database indica il numero di tabelle che esso contiene. Con phpMyAdmin è possibile accedere a ogni database di MySQL compresi quelli come *CollezioneMusicale* e *StudFac* che non sono stati creati tramite l'interfaccia grafica di phpMyAdmin, ma dalla linea comandi di MySQL o con altre interfacce grafiche.

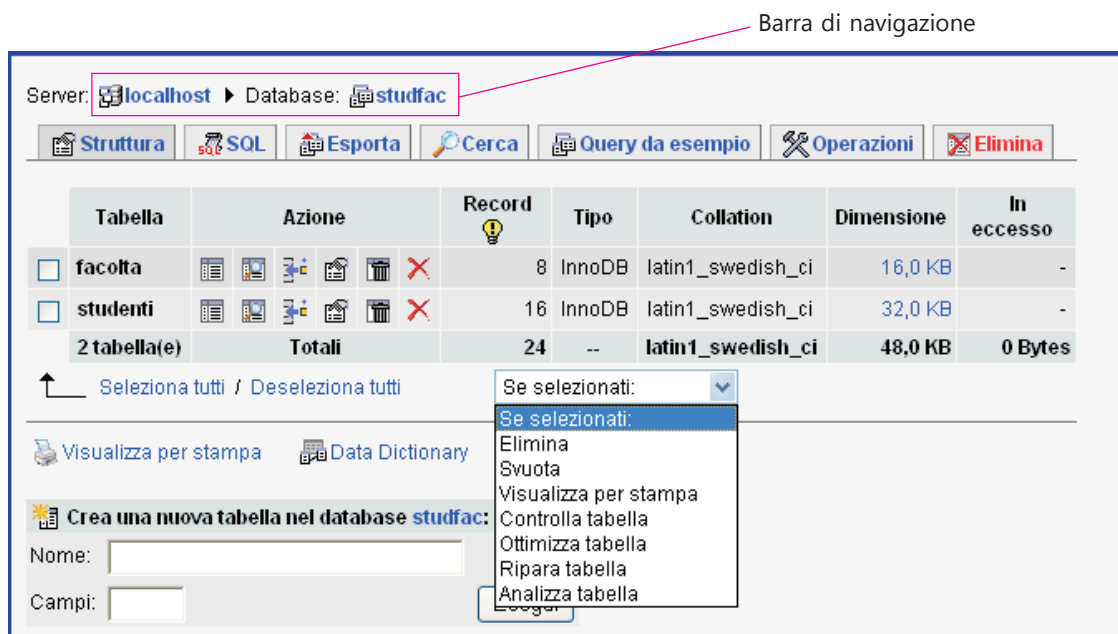
Nel frame principale della pagina di accesso, nella colonna intitolata MySQL, compare un menu di scelte per accedere ai servizi offerti da phpMyAdmin. Le scelte possibili sono:

- **Crea un nuovo database** per creare un nuovo database specificandone il nome;
- **Visualizza ...** per visualizzare informazioni sul server MySQL quali: traffico sul server e statistiche sulle query effettuate, valore assunto dalle variabili di sistema di MySQL, elenco dei processi in esecuzione, dettagli sui set di caratteri disponibili;
- **Privilegi** per assegnare permessi agli utenti e, più in generale, gestirli: con questa scelta è possibile creare nuovi utenti, eliminarli e modificare le autorizzazioni loro concesse;
- **Esporta** per esportare dati, creando file di backup secondo diverse modalità;
- **Database** l'opzione principale dal punto di vista operativo perchè con questa scelta si accede ai diversi database presenti sul server per eseguire le seguenti operazioni:
 - creare e cancellare database,
 - creare, copiare, cancellare e modificare le tabelle,
 - cancellare, modificare e aggiungere campi a una tabella,
 - eseguire comandi SQL,
 - gestire le chiavi e gli indici associate ai campi,
 - popolare le tabelle con i dati,
 - importare ed esportare i dati.

La scelta **Database** porta a una pagina che elenca i database sul server e permette di gestirli. Si noti nella figura una *barra dei comandi* con i pulsanti: **Database**, **Stato**, **Variabili**, **Set di caratteri**, **Privilegi**, **Processi**, **Esporta** che permettono di accedere agli stessi servizi attivabili dalla pagina di accesso a phpMyAdmin.



Da questa finestra si possono eliminare database, crearne di nuovi, visualizzare le autorizzazioni concesse per ognuno dei database elencati oppure passare alla finestra che descrive la struttura di uno specifico database. Per esempio, facendo clic su *StudFac*, phpMyAdmin visualizza una pagina dove sono elencate le tabelle che compongono il database selezionato.



Si noti, nel frame principale, una *barra di navigazione* che indica il percorso seguito per arrivare al database *StudFac* e la barra dei comandi con i pulsanti **Struttura**, **SQL**, **Esporta**, **Cerca**, **Query da progetto**, **Operazioni** ed **Elimina**, che attivano funzioni relative all'intero database.

La finestra visualizza la struttura del database scelto, con le tabelle che lo compongono e alcune informazioni di sintesi sulle tabelle. Dalla figura si nota, per esempio, che le tabelle *Facolta* e *Studenti* sono di tipo InnoDB. La struttura del database con informazioni di carattere generale e il dizionario dei dati con le informazioni di dettaglio sulla struttura delle tabelle sono visualizzabili e, successivamente, stampabili facendo clic su: **Visualizza per stampa** e **Data Dictionary**



Da questa finestra è possibile accedere e manipolare ogni tabella del database, facendo clic sulle icone dei *pulsanti di azione* che compaiono di seguito al nome delle tabelle, per eseguire le seguenti operazioni:

- Mostra:** visualizza il contenuto delle prime 30 righe della tabella selezionata
- Cerca:** per fare ricerche all'interno dei campi della tabella selezionata
- Inserisci:** visualizza una pagina che permette di inserire nuove righe in una tabella
- Struttura:** visualizza la struttura della tabella con i campi che la compongono e le relative definizioni
- Svuota:** per cancellare tutte le righe della tabella senza però eliminare la tabella dal database
- Elimina:** per cancellare la tabella e il relativo contenuto dal database.

Facendo clic su **Struttura**, per la tabella *Studenti*, viene visualizzata la pagina in figura che descrive la struttura della tabella *Studenti* con i campi che la compongono e le relative caratteristiche.

Si noti la barra di navigazione che permette di ritornare alla pagina di visualizzazione della struttura di *StudFac* o alla pagina di partenza e la barra dei comandi con i pulsanti **Struttura**, **Mostra**, **SQL**, **Cerca**, **Inserisci**, **Esporta**, **Operazioni**, **Svuota** ed **Elimina** per gestire la tabella *Studenti*.

Server: localhost Database: studfac Tabella: studenti

Struttura Mostra SQL Cerca Inserisci Esporta Operazioni Svuota Elimina

InnoDB free: 3072 kB; ("CodFac") REFER `studfac/facolta` ("Co"); InnoDB free: 3072 kB; ("CodFac") REFER `studfac/facolta` ("CodFac")

	Campo	Tipo	Collation	Attributi	Null	Predefinito	Extra	Azione
☐	Matricola	varchar(5)	latin1_swedish_ci		No			     
☐	Cognome	varchar(30)	latin1_swedish_ci		No			     
☐	Nome	varchar(20)	latin1_swedish_ci		No			     
☐	Indirizzo	varchar(30)	latin1_swedish_ci		No			     
☐	CodFac	varchar(5)	latin1_swedish_ci		No			     

Seleziona tutti / Deseleziona tutti Se selezionati:     

Visualizza per stampa Vedi relazioni Proponi la struttura della tabella

Aggiungi 1 campo() Alla fine della tabella All'inizio della tabella Dopo Matricola Esegui

Indici					Spazio utilizzato:		Statistiche righe:	
Nome chiave	Tipo	Cardinalità	Azione	Campo	Tipo	Utilizzo	Istruzioni	Valore
PRIMARY	PRIMARY	16	 	Matricola	Dati	16.384 Bytes	Formato	dinamico
CodFac	INDEX	16	 	CodFac	Indice	16.384 Bytes	Collation	latin1_swedish_ci
Crea un indice su 1 columns Esegui					Totale	32.768 Bytes	Creazione	03 Gen, 2006 at 10:18 AM

Esegui la query SQL sul database studfac

SELECT * FROM `studenti` WHERE 1

Campi: Matricola Cognome

La finestra visualizza la struttura della tabella scelta e permette di modificarla eliminando o aggiungendo campi, oltre a modificare le caratteristiche dei campi che la compongono.

Come si può vedere dalla figura, in questa pagina sono descritti tutti gli elementi che caratterizzano una tabella: i campi che la compongono e il loro tipo, la chiave primaria, gli indici e i collegamenti tra campi della tabella e campi di altre tabelle del database.

Nella parte inferiore della figura si intravede anche una finestra nella quale si può scrivere ed eseguire una query SQL.

Per esaminare nei dettagli le funzionalità di phpMyAdmin faremo uso del seguente esempio.

ESEMPIO 1 - Un database raccoglie le informazioni sui farmaci in vendita, con l'indicazione della ditta produttrice e del principio attivo.

Un database con queste caratteristiche è indicato con il nome *Prontuario dei farmaci*. I farmaci sono classificati secondo il principio attivo, cioè la loro composizione chimica. Il principio attivo è identificato attraverso una codifica internazionale, rappresentata dalla classificazione *ATC (Anatomica Terapeutica Chimica)*.

I farmaci sono approvati dal Ministero della Salute, dal cui sito sono stati prelevati alcuni dati di prova per il database utilizzato nell'esempio.

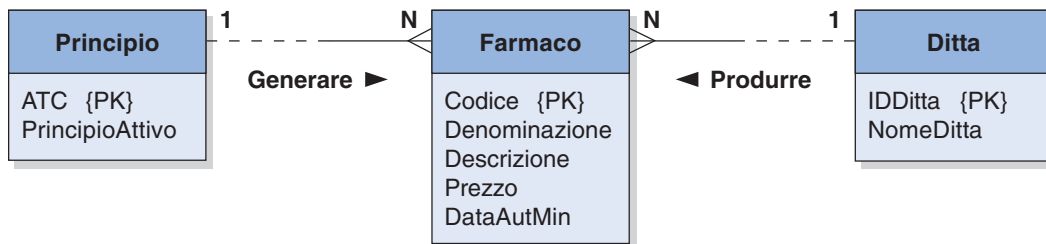
Le entità individuate sono:

- *Principio* con le informazioni sui principi attivi: la codifica ATC e la descrizione estesa;
- *Ditta* con le informazioni sulle aziende farmaceutiche: il codice e la denominazione;
- *Farmaco* con le informazioni sulle diverse confezioni dei farmaci in vendita: il codice del farmaco (indicato sulle confezioni con la sigla *AIC, Autorizzazione Immissione in Commercio*), la sua denominazione commerciale, la descrizione della confezione (bustine, compresse, soluzioni, ecc.), il prezzo e la data dell'Autorizzazione ministeriale.

Tra *Principio* e *Farmaco* esiste un'associazione *Generare* di tipo uno a molti, perché ogni principio attivo può dare origine a diversi prodotti commerciali, ma ogni farmaco ha un solo principio attivo nella sua composizione. La partecipazione di *Principio* è facoltativa, mentre la partecipazione di *Farmaco* è obbligatoria.

La seconda associazione è *Produrre* che mette in relazione una ditta con i farmaci da essa venduti. È un'associazione uno a molti perché ogni azienda può produrre molti farmaci, mentre ogni farmaco deve essere riferito a una sola *Ditta* produttrice. La partecipazione di *Ditta* è facoltativa, mentre la partecipazione di *Farmaco* è obbligatoria.

Rappresentiamo l'analisi dei dati con il seguente diagramma E/R:



Verifichiamo il modello costruito con le regole di lettura, ottenendo le seguenti frasi:

Ogni principio può generare uno o più farmaci; ogni farmaco deve essere generato da uno e un solo principio attivo.

Ogni ditta può produrre più farmaci; ogni farmaco deve essere prodotto da una sola ditta.

Gli attributi di *Principio* sono: *ATC* e *PrincipioAttivo*, con *ATC* chiave primaria dell'entità.

Gli attributi di *Ditta* sono: *IDDitta* e *NomeDitta*, con *IDDitta* chiave primaria.

Gli attributi di *Farmaco* sono: *Codice* (chiave primaria), *Denominazione*, *Descrizione*, *Prezzo* e *DataAutMin*.

Il passaggio dal modello concettuale al modello relazionale permette di definire le due associazioni con il seguente schema, dove l'attributo sottolineato rappresenta la chiave primaria, mentre quello in corsivo è la chiave esterna della relazione:

Principi (ATC, PrincipioAttivo)

Ditte (IDDitta, NomeDitta)

Farmaci (Codice, Denominazione, Descrizione, Prezzo, *ATC*, *IDDitta*, DataAutMin)

2. La creazione del database, delle tabelle e delle chiavi esterne

La creazione di un database si effettua specificandone il nome nella casella **Crea un nuovo database** della pagina iniziale di phpMyAdmin e inviando la richiesta al sistema con un clic sul bottone **Crea** posto sulla stessa riga.

ESEMPIO 2 - Creare il database Prontuario e le tabelle che lo compongono.

Per creare il database *Prontuario* si effettuano le operazioni sopra indicate e il sistema visualizza una pagina nella quale compare il comando SQL eseguito per creare il database. La figura mostra la parte centrale della pagina visualizzata, con la casella che permette di creare una nuova tabella nel database *Prontuario*.

Server: localhost ▶ Database: Prontuario

Struttura SQL Esporta Cerca Query da esempio Operazioni Elimina

Database Prontuario è stato creato.

query SQL:
`CREATE DATABASE `Prontuario` ;`

[Modifica] [Crea il codice PHP]

Non ci sono tabelle nel database.

Crea una nuova tabella nel database Prontuario:

Nome:

Campi:

Per creare la tabella *Ditte* si specifica il suo nome e il numero dei campi che la compongono, nelle due caselle etichettate con **Nome** e **Campi** e si fa clic sul bottone **Esegui**.

phpMyAdmin visualizza una pagina nella quale si definiscono le caratteristiche dei campi che compongono la tabella. La figura mostra come sono stati definiti i campi *IDDitta* e *NomeDitta* di *Ditte*.

Il sistema permette di precisare anche altre caratteristiche che non compaiono in figura quali, per esempio, la indicizzazione di un campo, l'univocità dei suoi valori oltre all'inserimento di note per descrivere i campi. Per farlo bisogna posizionarsi sulle caselle con il tasto **Tab** o usare la barra di scorrimento orizzontale posta alla base della pagina che permette di accedere alle caselle che non sono visualizzate.

Server: localhost ▶ Database: Prontuario ▶ Tabella: Ditte

Campo	Tipo ?	Lunghezza/Set *	Collation	Attributi	Null	Predefinito **	Extra
IDDitta	INT				not null		auto_increment
NomeDitta	VARCHAR	255			not null		

Commenti sulla tabella:

Aggiungi

Tipo tabella: Collation:

* Se il tipo di campo è VARCHAR, CHAR, TINYTEXT, TEXT, MEDIUMTEXT o LONGTEXT, immettere i valori usando il formato: 'a','b','c'...
Se comunque si vogliono usare backslashes ("\" o dei single quote ("') davanti a questi valori, backslasharli (per esempio \"xyz\" o 'a'b').
** Per i valori predefiniti, immettere un singolo valore, senza backslashes escaping o virgolette, utilizzando questo formato: a

La creazione della tabella avviene facendo clic sul pulsante **Salva**.

Il sistema crea la tabella e ne visualizza la struttura assieme a un messaggio che informa dell'avvenuta creazione e al comando SQL che l'ha creata. La figura seguente, per esempio, mostra la parte centrale della pagina che viene visualizzata dopo la creazione della tabella *Principi*, composta dai due campi *ATC* e *PrincipioAttivo*, entrambi di tipo *Varchar*.

Server: localhost ▶ Database: Prontuario ▶ Tabella: Principi

Struttura Mostra SQL Cerca Inserisci Esporta Operazioni Svota Elimina

Tabella Principi è stato creato.

query SQL:

```
CREATE TABLE `Principi` (  
  `ATC` VARCHAR(10) NOT NULL,  
  `PrincipioAttivo` VARCHAR(255) NOT NULL,  
  PRIMARY KEY (`ATC`)  
)
```

[Modifica] [Crea il codice PHP]

Campo	Tipo	Collation	Attributi	Null	Predefinito	Extra	Azione
☐ ATC	varchar(10)	latin1_swedish_ci		No			
☐ PrincipioAttivo	varchar(255)	latin1_swedish_ci		No			

Seleziona tutti / Deseleziona tutti Se selezionati:

Visualizza per stampa Vedi relazioni Proponi la struttura della tabella

Aggiungi 1 campo() Alla fine della tabella All'inizio della tabella Dopo ATC Esegui

I campi che compongono una tabella possono essere modificati con azioni avviate da pulsanti. I pulsanti di azione posti accanto alla definizione di un campo permettono infatti di:

- modificarne la definizione;
- eliminare un campo;
- definirlo come componente della chiave;
- indicizzare il campo;
- impedire l'immissione di valori duplicati in quel campo.

L'ultima tabella di *Prontuario* è la tabella *Farmaci*. Questa tabella contiene, come chiavi esterne, i due campi *ATC* e *IDDitta* che devono essere posti in relazione con gli omonimi campi delle altre tabelle del database. Per potere definire le chiavi esterne e i vincoli di integrità referenziale la tabella *Farmaci* deve essere di tipo *InnoDB* e bisogna indicizzare i campi *IDDitta* e *ATC*.

Server: localhost ▶ Database: Prontuario ▶ Tabella: Farmaci

Campo	Tipo	Lunghezza/Set	Collation
Codice	VARCHAR	20	
Denominazione	VARCHAR	40	
Descrizione	VARCHAR	255	
Prezzo	DECIMAL	12,2	
ATC	VARCHAR	10	
IDDitta	INT		
DataAutMin	DATE		

Commenti sulla tabella:

Tipo tabella: INNO DB

Aggiungi 1 campo() Esegui

Salva

Per creare i vincoli di integrità referenziale bisogna che *tutte* le tabelle poste in relazione siano di tipo *InnoDB*.

Bisogna allora modificare il tipo delle tabelle *Ditte* e *Principi* che sono state create per default di tipo *MyISAM* che è il tipo predefinito per le tabelle.

Per farlo si accede, nella pagina di visualizzazione delle proprietà della tabella *Ditte* (e della tabella *Principi*) ai servizi attivati con il pulsante **Operazioni** della barra dei comandi. In questa finestra è possibile modificare il tipo della tabella per mezzo di una casella combinata di nome **Tipo Tabella**.

Dopo aver modificato il tipo delle tabelle *Ditte* e *Principi* con questa procedura, nella pagina di visualizzazione della struttura del database *Prontuario* appaiono le seguenti informazioni:

Tabella	Azione	Record	Tipo	Collation	Dimensione	In eccesso
☐ ditte		0	InnoDB	latin1_swedish_ci	16,0 KB	-
☐ farmaci		0	InnoDB	latin1_swedish_ci	48,0 KB	-
☐ principi		0	InnoDB	latin1_swedish_ci	16,0 KB	-
3 tabella(e)	Totali	0	--	latin1_swedish_ci	80,0 KB	0 Bytes

/

Dalla figura si nota che il database *Prontuario* con le tre tabelle previste è stato creato e che le tabelle sono vuote.

Il database va popolato con dati. Prima di immettere record di dati nelle tabelle è bene definire le chiavi esterne e i vincoli di integrità referenziale. In questo modo il sistema impedirà l'immissione di una riga di *Farmaci* con riferimenti a righe di *Ditte* o *Principi* inesistenti.

ESEMPIO 3 - Definire chiavi esterne e vincoli di integrità referenziale nella tabella *Farmaci*.

Per definire le chiavi esterne e i vincoli di integrità referenziale bisogna fare clic su: **Vedi relazioni** in visualizzazione struttura della tabella *Farmaci*. Si accede così a una pagina che nel frame principale contiene la finestra in figura.

Server: localhost Database: Prontuario Tabella: farmaci

Collegamenti a

Relazioni interne		InnoDB	
Codice	--	ON DELETE	ON UPDATE
Denominazione	--		
Descrizione	--		
Prezzo	--		
ATC	principi->ATC	ON DELETE	ON UPDATE
IDDitta	ditte->IDDitta	ON DELETE	ON UPDATE
DataAutMin	--		

In questa finestra è possibile stabilire collegamenti esterni per quei campi di *Farmaci* per i quali sono stati costruiti indici. Nel caso in esame tali collegamenti possono essere definiti per i soli tre campi *Codice*, *ATC* e *IDDitta* di *Farmaci* in quanto questi sono i soli campi per i quali sono stati definiti degli indici.

Il collegamento avviene scegliendo il campo da mettere in relazione per mezzo di caselle combinate che elencano tutti i campi del database collegabili. La figura della pagina precedente mostra che il campo *ATC* di *Farmaci* è stato messo in relazione con il campo *ATC* di *Principi* e che il campo *IDDitta* di *Farmaci* è stato messo in relazione con il campo *IDDitta* di *Ditte*, mentre non è stata costruita alcuna associazione per il campo *Codice* di *Farmaci*. Le opzioni per **ON DELETE**, **ON UPDATE** sono quelle di *default* e quindi vale la dichiarazione **NO ACTION**.

La creazione delle chiavi esterne avviene facendo clic sul pulsante **Esegui**. L'avvenuta creazione delle chiavi esterne e dei vincoli di integrità referenziale è confermata dalla presenza in visualizzazione struttura della tabella *Farmaci* di un messaggio come quello in figura, che descrive i collegamenti con i campi di altre tabelle.

Server: localhost ▶ Database: Prontuario ▶ Tabella: farmaci

Struttura Mostra SQL Cerca Inserisci Esporta Operazioni Svuota Elimina

InnoDB free: 3072 kB; ('IDDitta') REFER `prontuario/ditte`('IDDitta'); ('ATC') REFER `prontuario/principi`('ATC')

Per verificare gli effetti delle azioni eseguite popoliamo con alcuni dati di prova le tabelle *Ditte* e *Principi* e, successivamente, inseriamo un record nella tabella *Farmaci*.

Per inserire dati in una tabella si fa clic pulsante **Inserisci** nella barra dei comandi della tabella oppure sul pulsante di azione **Inserisci** in corrispondenza della tabella scelta. Nel caso della tabella *Ditte* si accede alla pagina in figura che permette di inserire righe nella tabella digitando i valori dei campi che la compongono nelle caselle **Valore**. Ai valori inseriti è anche possibile applicare una funzione, per esempio la funzione *LCASE* che trasforma i caratteri di una stringa nei corrispondenti caratteri minuscoli, scegliendola da un elenco proposto nella casella combinata di nome **Funzione**.

Server: localhost ▶ Database: Prontuario ▶ Tabella: ditte

Struttura Mostra SQL Cerca Inserisci Esporta Operazioni Svuota Elimina

InnoDB free: 3072 kB

Campo	Tipo	Funzione	Null	Valore
IDDitta	int(11)		1	
NomeDitta	varchar(255)		Ditta di prova 1	

☐ Ignora

Campo	Tipo	Funzione	Null	Valore
IDDitta	int(11)		2	
NomeDitta	varchar(255)		Ditta di prova 2	

Inserisci come nuova riga -- E --

☒ Indietro
☒ Oppure
☐ Inserisci un nuovo record

 Esegui Riavvia

L'immissione delle due righe con i dati di prova avviene facendo clic sul bottone **Esegui**. Dopo avere inserito alcuni dati di prova anche nella tabella *Principi* si vogliono inserire righe nella tabella dei farmaci accedendo a una pagina come quella mostrata in figura.

Server: localhost ▶ Database: prontuario ▶ Tabella: farmaci

Struttura
 Mostra
 SQL
 Cerca
 Inserisci
 Esporta
 Operazioni
 Svuota

InnoDB free: 3072 kB; ('IDDitta') REFER 'prontuario/ditte'('IDDitta'); ('ATC') REFER 'prontuario/principi'('ATC')

Campo	Tipo	Funzione	Null	Valore
Codice	varchar(20)			Codice 1
Denominazione	varchar(40)			Denominazione 1
Descrizione	varchar(255)			Descrizione 1
Prezzo	decimal(12,2)			10
ATC	varchar(10)			ATC1 -
IDDitta	int(11)			2 -
DataAutMin	date			2006-10-00

La figura mostra i valori immessi in corrispondenza dei campi di *Farmaci*. Si nota, in corrispondenza dei campi *ATC* e *IDDitta* di *Farmaci*, la presenza di caselle combinate che obbligano a inserire i valori da un elenco di valori ammissibili per quel campo. Nel caso in esame la casella combinata corrispondente a *IDDitta* conterrà solo 1 e 2 che sono i soli valori attribuiti al campo *IDDitta* di *Ditte*.

In breve: il sistema obbliga a inserire solo quei valori per i quali sono rispettati i vincoli di integrità referenziale.

I dati immessi in una tabella possono essere visualizzati, per esempio, partendo dalla visualizzazione struttura del database e facendo clic sul pulsante di azione **Mostra** posto accanto al nome della tabella. Nel caso della tabella *Farmaci* viene visualizzato l'elenco:

			Codice	Denominazione	Descrizione	Prezzo	ATC	IDDitta	DataAutMin
☐			Codice 1	Denominazione 1	Descrizione 1	10.00	ATC1	2	2006-00-00
Seleziona tutti / Deseleziona tutti Se selezionati:									

Si noti che i valori corrispondenti alle chiavi esterne sono visualizzati in azzurro. Facendo clic su questi valori il sistema mostra la riga corrispondente della tabella associata. Per esempio facendo clic su **2** nella colonna *IDDitta* viene visualizzata, in una nuova pagina, la riga che corrisponde all'interrogazione:

```
SELECT * FROM Ditte WHERE IDDitta = 2;
```

Nell'Esempio 3 abbiamo visto come si fa a inserire righe in una tabella. L'intero database potrà allora essere popolato immettendo i dati riga per riga nelle tre tabelle che lo compongono. Nel caso del database *Prontuario*, avendo impostato l'integrità referenziale, bisognerà inserire prima i dati delle ditte e dei principi e successivamente quelli dei farmaci che li referenziano.

I record immessi in una tabella possono essere modificati e cancellati. Per farlo bisogna utilizzare i pulsanti di azione posti accanto a ogni record di una tabella:  per modificare i valori introdotti in una riga,  per cancellare l'intera riga. Le stesse operazioni si possono eseguire su più record evidenziandoli con il segno di spunta e utilizzando i pulsanti di azione di fianco alla scritta: **Se selezionati**.

Volendo cancellare tutte le righe di una tabella conviene usare il pulsante **Svuota** nella barra dei comandi. In tale caso il sistema utilizza il comando **TRUNCATE** per svuotare una tabella da ogni contenuto senza però eliminarla. Usiamo questa funzionalità per svuotare le tabelle dai dati di prova precedentemente immessi. Queste operazioni si devono eseguire rispettando i vincoli di integrità referenziale. Per farlo bisogna eliminare prima i dati di *Farmaci* e successivamente quelli delle altre due tabelle.

3. Importazione di dati da file esterni

Mostriamo, con un esempio, come inserire dati in una tabella caricandoli direttamente da un file esterno che contiene i dati da immettere. Per esempio, il file *Ditte.txt* contiene i dati della tabella *Ditte* del database *Prontuario* nel seguente formato:

```
6;"A.M.S.A. S.R.L.";
42;"AVENTIS PHARMA S.P.A.";
69;"BRACCO S.P.A.";
70;"BRISTOL MYERS SQUIBB S.R.L.";
140;"FOURNIER PHARMA S.P.A.";
. . .
```

ESEMPIO 4 - Popolare la tabella *Ditte* con i dati del file *Ditte.txt*.

Per caricare dati da un file esterno bisogna fare clic su  **Inserisci un file di testo nella tabella** che si trova ai piedi della pagina di visualizzazione della struttura della tabella.



InnoDB free: 3072 kB		
Percorso del file	c:\PhpMysql\Dat\Ditte.txt	Sfoggia... (Dimensione massima: 2.048KB)
Sostituisci i dati della tabella col file	☐ Sostituisci	Il contenuto del file sostituisce le righe della tabella con la stessa chiave primaria o chiave unica.
Campo terminato da	;	Il carattere terminatore dei campi.
Campo composto da	" ☐ OPZIONALE	In genere da doppi apici (virgolette). OPZIONALE indica che solo i campi <i>char</i> e <i>varchar</i> devono essere delimitati dal carattere indicato.
Campo impedito da	\	Opzionale. Questo carattere controlla come scrivere o leggere i caratteri speciali.
Linee terminate da	\n	Ritorno carrello: \r Fine riga: \n
Nomi delle colonne		Per caricare i dati solo per alcune colonne della tabella, specificare la lista dei campi (separati da virgole).
Metodo di CARICAMENTO	☐ ...DATA ☒ ...DATA LOCAL	Il metodo migliore è selezionato di default, ma lo potete cambiare se fallisce.
?		
Invia Riavvia		

La pagina visualizzata permette di precisare il file con i dati digitandone il nome, o di cercarlo con **Sfoglia**, e di precisare tutti i parametri che influenzano il caricamento.

Come si nota dalla figura, i commenti posti accanto alle caselle per l'immissione dei parametri descrivono in modo puntuale il significato dei diversi campi.

I valori in figura sono quelli proposti dal sistema e si rivelano adatti al caricamento dei dati nel formato del file *Ditte.txt*. Per avviare la funzione di caricamento bisogna fare clic sul bottone **Invia**.

Il sistema carica i dati usando il comando **LOAD DATA INFILE** e informa sui risultati del caricamento.

La figura mostra i risultati dell'operazione sopra eseguita. Si noti che il sistema visualizza il numero di righe immesse nella tabella (o un messaggio di errore se il caricamento fallisce per una qualsiasi ragione) e il comando eseguito per caricare i dati.

Struttura

Mostra

SQL

Cerca

Inserisci

Esporta

Operazioni

Svuota

Elimina

Righe inserite: 13 (La query ha impiegato 0.0019 sec)

query SQL:

LOAD DATA LOCAL INFILE 'C:/Program Files/EasyPHP1-8/tmp/php3A.tmp' INTO TABLE 'Ditte' FIELDS TERMINATED BY ';' ENCLOSED BY '"' ESCAPED BY '\\' LINES TERMINATED BY '\n'

Adottando la medesima procedura mettiamo i dati nelle tabelle *Principi* e *Ditte*. Accedendo alla pagina che descrive la struttura del database *Prontuario* si visualizzano le informazioni in figura dove si nota, per esempio, che la tabella *Farmaci* contiene 20 righe e la tabella *Principi* è formata da 12 record.

Server: localhost Database: prontuario

Struttura

SQL

Esporta

Cerca

Query da esempio

Operazioni

Elimina

Tabella	Azione	Record	Tipo	Collation	Dimensione	In eccesso
☐ ditte	 	13	InnoDB	latin1_swedish_ci	16,0 KB	-
☐ farmaci	 	20	InnoDB	latin1_swedish_ci	48,0 KB	-
☐ principi	 	12	InnoDB	latin1_swedish_ci	16,0 KB	-
3 tabella(e)	Totali	45	--	latin1_swedish_ci	80,0 KB	0 Bytes

Seleziona tutti / Deseleziona tutti

Se selezionati:

Visualizza per stampa

Data Dictionary

	ATC	PrincipioAttivo
☐  	B03BB01	Acido folico
☐  	D06AX02	Cloramfenicolo
☐  	G03BA03	Testosterone
☐  	G03CA57	Estrogeni coniugati naturali
☐  	G03GA01	Gonadotropina corionica
☐  	J01GA01	Streptomina
☐  	N01AF03	Tiopental
☐  	N03AA02	Fenobarbital
☐  	N03AB02	Fenitoina
☐  	N04AA01	Triesifenidile
☐  	P01CB01	Meglumina
☐  	R03DA04	Teofillina

Il contenuto delle tabelle può essere visualizzato con i pulsanti di azione posti accanto al nome delle tabelle.

←T→	Codice	Denominazione	Descrizione	Prezzo	ATC	IDDitta	DataAutMin
☐	001537012	FARMOTAL	500 MG POLVERE PER SOLUZIONE PER INFUSIONE 1 FLAC...	6.15	N01AF03	310	1996-08-03
☐	001801012	GLUCANTIM	1,5 G/5 ML SOLUZIONE INIETTABILE PER USO INTRAMUSC...	3.82	P01CB01	42	1997-05-12
☐	002021069	TEFAMIN	ELISIR 1 FLAC. 200 ML 1,58%	3.06	R03DA04	334	2002-03-15
☐	002129017	DINTOINA	30 COMPRESSE RIVESTITE 100 MG	1.86	N03AB02	334	2001-11-05
☐	002309033	FOLINA	15 MG/2 ML SOLUZIONE INIETTABILE PER USO I.M. 5 FI...	2.16	B03BB01	355	1998-04-14
☐	002309045	FOLINA	5 MG CAPSULE RIGIDE 20 CAPSULE	1.63	B03BB01	355	2001-11-01
☐	002347019	PENTOTHAL SODIUM	1 FLACONE 0,5 G	6.01	N01AF03	168	1999-08-11
☐	002347021	PENTOTHAL SODIUM	1 FLACONE 1 G	11.48	N01AF03	168	1999-02-21
☐	002792188	PREMARIN	0,625 MG COMPRESSE 28 COMPRESSE	4.49	G03CA57	401	1997-01-17
☐	002860017	LUMINALE	200 MG SOLUZIONE INIETTABILE PER USO INTRAMUSCOLAR...	5.33	N03AA02	69	1995-04-01
☐	002860031	LUMINALE	20 COMPRESSE 100 MG	1.60	N03AA02	69	1998-09-25
☐	002860043	LUMINALE	LUMINALETTE 30 COMPRESSE 15 MG	1.55	N03AA02	69	1999-03-10
☐	002866236	CHEMICETINA	POLVERE CHIRURGICA 1 G	2.12	D06AX02	140	2001-09-04
☐	002992016	STREPTOMICINA SOLFATO SQUIBB	1 FLACONE 1.000.000 UI	2.48	J01GA01	70	2002-09-03
☐	003247032	PROFASI HP	2000 U.I. POLVERE E SOLVENTE PER SOLUZIONE INIETTA...	5.09	G03GA01	180	1997-01-19
☐	003247044	PROFASI HP	5000 U.I. POLVERE E SOLVENTE PER SOLUZIONE INIETTA...	5.72	G03GA01	180	2002-01-26
☐	003488018	ARTANE	2 MG COMPRESSE 50 COMPRESSE	2.53	N04AA01	377	2004-06-23
☐	003559059	TESTOVIS	IM 2 FIALE 2 ML 100 MG	2.17	G03BA03	228	2002-07-22
☐	003763125	GONASI HP	5000 U.I./2 ML POLVERE E SOLVENTE	5.72	G03GA01	6	2000-04-04

←T→	IDDitta	NomeDitta
☐	6	A.M.S.A. S.R.L.
☐	42	AVENTIS PHARMA S.P.A.
☐	69	BRACCO S.P.A.
☐	70	BRISTOL MYERS SQUIBB S.R.L.
☐	140	FOURNIER PHARMA S.P.A.
☐	168	HOSPIRA S.P.A.
☐	180	INDUSTRIA FARMACEUTICA SERONO S.P.A.
☐	228	LABORATORIO FARMACEUTICO SIT SPECIALITA' IGIENICO ...
☐	310	PHARMACIA ITALIA S.P.A.
☐	334	RECORDATI INDUSTRIA CHIMICA FARMACEUTICA S.P.A.
☐	355	SCHWARZ PHARMA S.P.A.
☐	377	TEOFARMA S.R.L.
☐	401	WYETH MEDICA IRELAND

4. Esportazione di tabelle in file esterni

Con phpMyAdmin si possono esportare dati in file esterni al database. Per farlo bisogna fare clic sul pulsante **Esporta** nella barra dei comandi di una pagina di visualizzazione della tabella che si desidera esportare.

Il sistema propone una pagina dalla quale è possibile effettuare diverse scelte:

- Il tipo di esportazione che si desidera effettuare. Per esempio i soli dati della tabella, in diversi formati, oppure dati e struttura della tabella.
- Il nome del file esterno dove salvare il risultato dell'esportazione. È anche possibile limitarsi a visualizzare i risultati dell'esportazione in una finestra di phpMyAdmin.
- La compatibilità del formato di esportazione nei confronti di diversi DBMS fra cui: lo standard ANSI, DB2, ORACLE, e altri.

Il prossimo esempio mostra come usare il sistema di esportazione per salvare in un file esterno i soli dati della tabella *Principi*.

ESEMPIO 5 - Salvare i dati della tabella Principi in un file esterno all'ambiente phpMyAdmin.

Per operare l'esportazione dei dati di *Principi* bisogna fare clic su **Esporta**, in una pagina di visualizzazione della tabella *Principi*, per accedere alla pagina con le diverse modalità di esportazione e si sceglie l'opzione **dati CSV**. Il sistema visualizza una pagina come quella in figura per precisare le opzioni di salvataggio di un file CSV.

Dall'esame della figura si nota che le opzioni proposte sono le medesime del precedente Esempio 4, nel caso dell'importazione di dati.

Server: localhost ▶ Database: prontuario ▶ Tabella: principi

Struttura Mostra SQL Cerca Inserisci Esporta Operazioni Svuota

InnoDB free: 3072 kB

Visualizza dump (schema) della tabella

Esporta

- ☐ SQL
- ☐ LaTeX
- ☐ CSV per dati MS Excel
- ☒ dati CSV
- ☐ XML

Opzioni CSV

Campo terminato da ;

Campo composto da "

Campo impedito da \

Linee terminate da \r\n

Sostituisci NULL con NULL

☐ Mette i nomi delle colonne alla prima riga

Dump di 12 righe a partire dalla riga 0.

☐ Salva con nome...

Nome file template: Dump__TABLE__ (☒ ricorda il template)*

Compressione

- ☒ Nessuno
- ☐ "compressato con zip"
- ☐ "compressato con gzip"

Esegui

* Utilizza __DB__ come nome per il DataBase, __TABLE__ come nome per la tabella e any strftime opzioni per per le specifiche del tempo, l'estensione sarà aggiunta automaticamente. Qualsiasi altro testo sarà conservato.

Si osservi il segno di spunta accanto a **Salva con nome**.

Omettendo il segno di spunta il comando viene eseguito: il risultato non viene memorizzato in un file, ma è solamente visualizzato in una nuova pagina.

Nella casella sottostante a **Salva con nome** è stato impostato uno scheletro di nome per il file di salvataggio completando la proposta del sistema (`__TABLE__`), con il termine *Dump*.

In questo modo il file esterno con i dati avrà nome *Dumpprincipi.csv*.

Si osservi infine che è possibile, mettendo il segno di spunta nell'apposito riquadro, generare un file che riporta nella prima riga il nome delle colonne che compongono la tabella.

Facendo clic su **Esegui** il sistema visualizza una finestra di dialogo nella quale si precisa la collocazione su disco del file *Dumpprincipi.csv*. Il risultato dell'esportazione è visualizzato nel seguente riquadro.

```
"B03BB01";"Acido folico"  
"D06AX02";"Cloramfenicolo"  
"G03BA03";"Testosterone"  
"G03CA57";"Estrogeni coniugati naturali"  
"G03GA01";"Gonadotropina corionica"  
"J01GA01";"Streptomicina"  
"N01AF03";"Tiopental"  
"N03AA02";"Fenobarbital"  
"N03AB02";"Fenitoina"  
"N04AA01";"Triesifenidile"  
"P01CB01";"Meglumina"  
"R03DA04";"Teofillina"
```

Partendo da questo file è possibile popolare di dati la tabella *Principi* con la procedura descritta nel precedente approfondimento o trasferirla a un altro applicativo che supporta il formato CSV dei dati.

L'esportazione dei dati in un file esterno è quindi un modo per produrre una copia dei dati di una tabella con la quale è possibile sia trasferire i dati di quella tabella in un altro database, sia ripopolare di dati la tabella salvata nel caso che, per una qualsiasi ragione, i dati in essa contenuta debbano essere ripristinati.

Il salvataggio appena effettuato si limita a ricopiare i dati contenuti nella tabella. Supponiamo allora che una tabella del database, per esempio la tabella *Farmaci*, sia stata cancellata dal database. Per poterla ricostruire bisogna come prima cosa creare la tabella definendone i campi e le chiavi esterne e, successivamente, popolarla di dati caricandoli da un file esterno costruito, per esempio, esportando i dati.

È anche possibile esportare una tabella ricopiandola in un file esterno in modo che il file di salvataggio contenga non solo i dati della tabella ma anche tutte le informazioni per poterla ricostruire automaticamente partendo da tale file.

ESEMPIO 6 - Esportare dati e struttura della tabella Farmaci.

Visualizza dump (schema) della tabella

Esporta

- ☒ SQL
- ☐ LaTeX
- ☐ CSV per dati MS Excel
- ☐ dati CSV
- ☐ XML

Opzioni SQL

Aggiunge un commento personalizzato all'header (\n per tornare a capo):

☐ Includi export in una transazione

☐ Disabilita i controlli sulle chiavi straniere

☒ **Struttura:**

- ☐ Aggiungi DROP TABLE
- ☐ Aggiunge IF NOT EXISTS
- ☒ Aggiungi valore AUTO_INCREMENT
- ☒ Usa i backquotes con i nomi delle tabelle e dei campi

Aggiungi nei commenti

- ☐ Creazione/Aggiornamento/Controllo date
- ☐ Relazioni
- ☐ Commenti
- ☐ tipo MIME

Compatibilità dell'esportazione SQL: NONE

☒ **Dati:**

- ☐ Inserimenti completi
- ☐ Inserimenti estesi
- ☐ Utilizza inserimenti ritardati
- ☐ Utilizza gli IGNORE INSERTS
- ☒ Usa l'esadecimale per i dati bin

Tipo di esportazione: INSERT

Dump di 20 righe a partire dalla riga 0

☐ **Salva con nome...**

Nome file template: Dump__TABLE__ (☒ ricorda il template)*

Per esportare dati e struttura di *Farmaci* si procede come nel caso precedente, facendo clic su **Esporta** in una pagina della tabella *Farmaci*, ma si sceglie l'opzione **SQL** come è mostrato nella figura.

La figura mostra i valori proposti dal sistema.

Si noti la possibilità di aggiungere commenti nelle intestazioni e di inserire diverse opzioni nel comando di creazione della tabella.

In figura si nota una casella combinata tramite la quale si richiede al sistema di generare un comando SQL per la creazione della tabella che sia compatibile con la sintassi di diversi DBMS. Fra i sistemi di gestione di database compaiono anche precedenti versioni di MySQL.

Si noti che mancando il segno di spunta accanto a **Salva con nome..** non verrà creato alcun file esterno con il risultato dell'esportazione.

In questo caso il dump della tabella viene eseguito e il risultato sarà visualizzato in una nuova pagina.

Per attivare l'esportazione della tabella bisogna fare clic sul bottone **Esegui** posto ai piedi della pagina.

Il risultato dell'operazione è mostrato nel seguente riquadro dove, per brevità, sono state eliminate alcune righe di commento e molti comandi INSERT per l'inserimento di righe nella tabella.

```
--
-- Struttura della tabella 'farmaci'
--
CREATE TABLE 'farmaci' (
  'Codice' varchar(20) NOT NULL default "",
  'Denominazione' varchar(40) NOT NULL default "",
  'Descrizione' varchar(255) NOT NULL default "",
  'Prezzo' decimal(12,2) NOT NULL default '0.00',
  'ATC' varchar(10) NOT NULL default "",
  'IDDI' int(11) NOT NULL default '0',
  'DataAutMin' date NOT NULL default '0000-00-00',
  PRIMARY KEY ('Codice'),
  KEY 'ATC' ('ATC','IDDI'),
  KEY 'IDDI' ('IDDI')
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Dump dei dati per la tabella 'farmaci'
--
INSERT INTO 'farmaci' VALUES ('001537012', 'FARMOTAL', '500 MG POLVERE PER
SOLUZIONE PER INFUSIONE 1 FLACONE 500 MG', 6.15, 'N01AF03', 310, '1996-08-
03');
...
...
...

INSERT INTO 'farmaci' VALUES ('003763125', 'GONASI HP', '5000 U.I./2 ML
POLVERE E SOLVENTE PER SOLUZIONE INIETTABILE PER USO INTRAMUSCOLARE O
SOTTOCUTANEO 1 FLACONCINO POLVERE + 1 FIALA
SOLVENTE DA 2 ML', 5.72, 'G03GA01', 6, '2000-04-04');

--
-- Limiti per la tabella 'farmaci'
--
ALTER TABLE 'farmaci'
  ADD CONSTRAINT 'farmaci_ibfk_2' FOREIGN KEY ('IDDI')
  REFERENCES 'ditte' ('IDDI'),
  ADD CONSTRAINT 'farmaci_ibfk_1' FOREIGN KEY ('ATC')
  REFERENCES 'principi' ('ATC');
```

Si noti che il dump della tabella *Farmaci* è composto da tre comandi SQL introdotti, in figura, da commenti evidenziati in neretto: un primo comando CREATE TABLE crea la tabella *Farmaci*. A questo segue un comando INSERT (o meglio una serie di comandi INSERT) per immettere dati nella tabella e infine un comando ALTER TABLE che definisce le chiavi esterne. Se il risultato dell'esportazione è salvato in un file esterno esso ha il nome definito dell'utente e ha estensione **.sql**.

L'esempio mostra che l'esportazione di una tabella con le modalità sopra descritte è attuata da phpMyAdmin costruendo un file composto dai comandi SQL che devono essere digitati per creare e immettere i dati nella tabella così salvata.

È allora possibile utilizzare il comando **SOURCE** che permette di eseguire comandi SQL in modalità batch, per creare una nuova tabella partendo dal file ottenuto esportando la tabella stessa.

Si noti che l'esportazione di un intero database, per esempio il database *Prontuario*, viene attuata facendo clic su **Esporta** nella barra dei comandi della pagina che visualizza la struttura del database e si compone dei soli comandi SQL per la definizione e il popolamento delle tabelle che compongono il database.

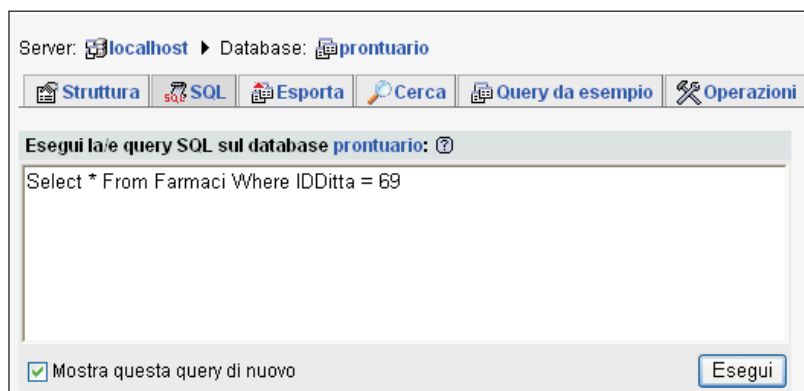
Per esempio, se si esporta il database *Prontuario* nel file *SalvaProntuario.sql*, tale file contiene i comandi per la definizione e il popolamento delle tabelle *Farmaci*, *Ditte*, *Principi* ma non il comando per la creazione del database *Prontuario*.

Utilizzando invece la funzionalità **Esporta** della home page di phpMyAdmin il file che viene generato, oltre ai comandi per la creazione e il popolamento delle tabelle del database esportato, conterrà anche il comando CREATE DATABASE che crea il database stesso.

La creazione di una o più tabelle di un database partendo da file esterni può essere effettuata per mezzo del comando SOURCE dalla linea comandi di MySQL oppure direttamente da phpMyAdmin.

5. Interrogazioni sul database

Il database può essere interrogato scrivendo una query SQL nella finestra che viene visualizzata facendo clic su **SQL** nella barra dei comandi del database.

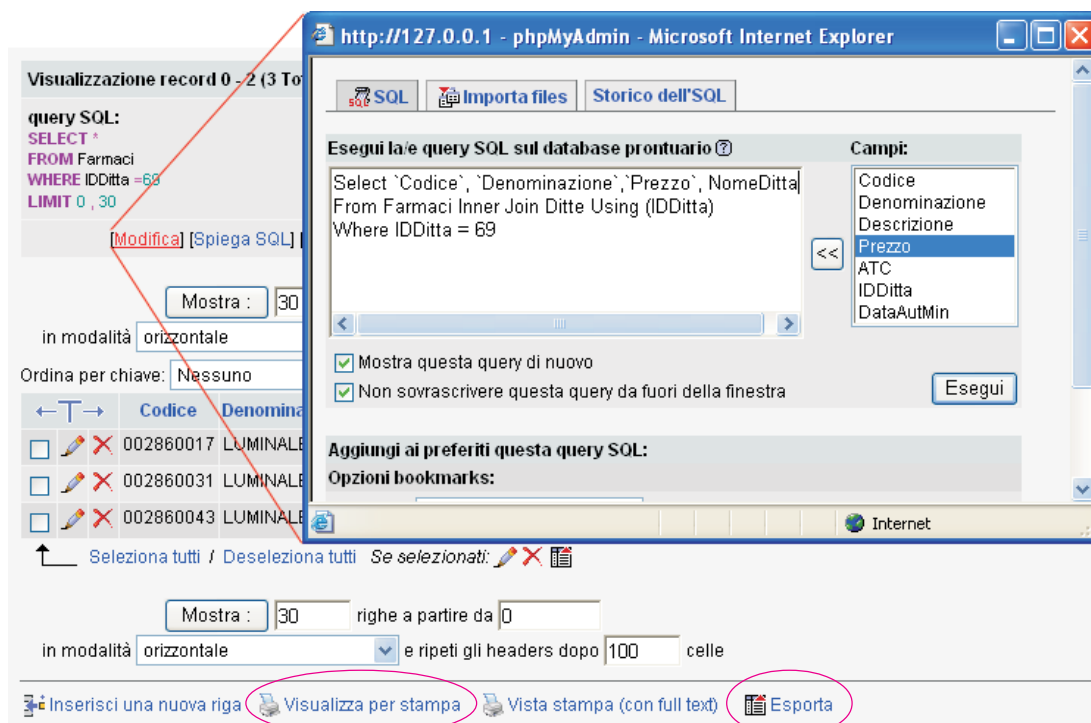


La figura mostra una query che ricerca i farmaci prodotti dalla ditta di codice 69 e che viene eseguita facendo clic su **Esegui**.

L'elenco è visualizzato in un'altra finestra assieme al comando che l'ha generato.

Nella pagina di visualizzazione dell'elenco è poi possibile modificare la query appena eseguita facendo clic sul pulsante **Modifica** posto alla base della finestra che mostra il testo del comando SQL.

La seguente figura mostra appunto l'elenco generato dalla query precedente e una nuova interrogazione che affina la precedente. La seconda interrogazione è stata composta sfruttando la possibilità di inserire nella query i nomi dei campi e facendo clic su << dopo avere selezionato i campi desiderati nella finestra **Campi**.



Alla base della figura si notano quattro icone e tra queste una etichettata **Visualizza per stampa** che permette di visualizzare e stampare, opportunamente formattato, l'elenco generato dalla query, ed **Esporta** per esportare in un file esterno i dati generati dall'interrogazione.

La figura mostra la pagina che viene visualizzata facendo clic su **Visualizza per stampa** con le informazioni sulla query che ha generato l'elenco.

Risultato SQL						
Host: localhost						
Database: prontuario						
Generato il: 19 Feb, 2006 at 05:20 PM						
Generato da: phpMyAdmin 2.6.1 / MySQL 4.1.9-max						
query SQL: Select * From Farmaci Where IDDDitta = 69 LIMIT 0, 30;						
Righe: 3						
Codice	Denominazione	Descrizione	Prezzo	ATC	IDDDitta	DataAutMin
002860017	LUMINALE	200 MG SOLUZIONE INIETTABILE PER USO INTRAMUSCOLAR...	5.33	N03AA02	69 [->]	1995-04-01
002860031	LUMINALE	20 COMPRESSE 100 MG	1.60	N03AA02	69 [->]	1998-09-25
002860043	LUMINALE	LUMINALETTE 30 COMPRESSE 15 MG	1.55	N03AA02	69 [->]	1999-03-10

phpMyAdmin permette di costruire interrogazioni anche in modalità **QBE** (Query By Example). Secondo questa modalità la query viene costruita in modalità grafica inserendo in una griglia i nomi dei campi che compongono l'interrogazione e i criteri che devono essere soddisfatti perché un campo faccia parte dei risultati della query.

Questa modalità viene attivata facendo clic sul pulsante **Query da progetto** nella barra dei comandi del database, visualizzando la pagina in figura. Il sistema propone, in partenza, una griglia con tre campi, per ognuno dei quali si possono definire due criteri di selezione composti secondo l'operazione logica di OR (si noti in figura il segno di spunta sul bottone **Oppure**) mentre i criteri per campi diversi sono composti secondo l'AND logico (segno di spunta su **E**).

Il numero di campi da includere nella query e il numero di criteri che si possono esprimere per ogni campo è modificabile con le caselle combinate: **Aggiungi/Cancella criterio**, **Aggiungi/Cancella campo**, oppure ponendo il segno di spunta sui bottoni **Aggiungi** e **Cancella**, in corrispondenza delle righe, per i criteri, e delle colonne, per i campi. Per modificare il layout della griglia, dopo avere operato le scelte opportune, bisogna fare clic su **Aggiorna Query**.

Caselle per aggiungere o eliminare un criterio o un campo

L'inserimento dei campi nella griglia avviene selezionandone il nome da una casella combinata come quella mostrata in figura che include i campi delle tabelle scelte nella finestra **Utilizza tabelle**.

Dopo avere inserito campi e criteri nella griglia, facendo clic su **Aggiorna Query**, il corrispondente comando SQL viene visualizzato nella finestra sotto la griglia. L'interrogazione costruita viene eseguita facendo clic sul pulsante **Invia Query**.

ESEMPIO 7 - Elencare Descrizione, Denominazione e Prezzo dei farmaci di costo superiore a 6 euro.

Si tratta di una semplice interrogazione basata sulla sola tabella *Farmaci*. Dal punto di vista dell'algebra relazionale l'elenco richiesto si ottiene eseguendo una selezione su *Farmaci*, per ricercare i soli farmaci di costo superiore a 6 euro, seguita da una proiezione sui campi che si vogliono visualizzare.

Per ottenere l'elenco richiesto bisogna inserire nella griglia della query i campi di nome *Descrizione*, *Denominazione* e *Prezzo*, avendo cura di mettere il segno di spunta nella casella **Mostra**, e di scrivere il criterio di selezione (>6) nella riga **Criterio** in corrispondenza del campo *Prezzo*.

Server: localhost Database: prontuario

Struttura SQL Esporta Cerca Query da esempio Operazioni Elimina

Campo:	'farmaci'.`Descrizione`	'farmaci'.`Denominazi`	'farmaci'.`Prezzo`
Ordinamento:			Crescente
Mostra:	☒	☒	☒
Criterio:			>6
Aggiungi: ☐ E: ☐			
Cancella: ☐ Oppure: ☒			
Modifica:	Oppure: ☐ E: ☒ Aggiungi ☐ Cancella ☐	Oppure: ☐ E: ☒ Aggiungi ☐ Cancella ☐	Oppure: ☐ E: ☒ Aggiungi ☐ Cancella ☐

Aggiungi/Cancella criterio: 0 Aggiungi/Cancella campo: 0 Aggiorna Query

Utilizza tabelle:

- ☒ farmaci
- ☐ ditte
- ☐ principi

Aggiorna Query

SQL-query sul database prontuario:

```
SELECT
'farmaci'.`Descrizione`,
'farmaci'.`Denominazione`,
'farmaci'.`Prezzo`
FROM `farmaci`
WHERE ('farmaci'.`Prezzo` >
6)
```

Invia Query

Volendo visualizzare i risultati per valori crescenti di prezzo si seleziona il valore *Crescente* nella riga **Ordinamento** del campo *Prezzo*.

La figura mostra, nella finestra **SQL-query sul database prontuario**, il comando SELECT che il sistema costruisce, dopo avere fatto clic su **Aggiorna Query**, sulla base del contenuto della griglia della query.

Il comando viene eseguito inviandolo a MySQL con **Invia Query**. Il risultato dell'interrogazione è visualizzato in un'altra finestra.

Il precedente esempio può essere preso come modello per costruire selezioni e proiezioni in modalità QBE. La congiunzione di tabelle si realizza scrivendo il criterio di congiunzione nella riga **Criteri** di uno dei campi che compaiono nella condizione di congiunzione.

Per esempio, se si congiungono le tabelle *Farmaci* e *Ditte*, secondo il criterio: *Ditte.IDDitta*=*Farmaci.IDDitta*, il join tra le due tabelle si può ottenere scrivendo la condizione =*Farmaci.IDDitta* nella riga **Criteri** del campo *Ditte.IDDitta*.

ESEMPIO 8 - Elencare Codice, Descrizione e Prezzo dei farmaci che hanno come principio attivo l'Acido Folico.

La sequenza di operazioni relazionali che produce l'elenco richiesto è la seguente:

1. σ_P Principi Temp1 = Selezione di Principi per PrincipioAttivo = 'Acido Folico'
2. Temp1_A $\bowtie$ Farmaci_A Temp2 = Congiunzione di Temp1 e Farmaci su ATC
3. Π_L Temp2 Proiezione di Temp2 su Codice, Descrizione, Prezzo

Per costruire questa interrogazione oltre ai campi che devono essere visualizzati (*Codice*, *Descrizione* e *Prezzo*) bisogna usare anche il campo *PrincipioAttivo* per limitare l'elenco ai soli farmaci che usano acido folico come principio attivo e *Principi.ATC* per costruire la condizione di congiunzione delle tabelle *Farmaci* e *Principi*: *Principi.ATC* = *Farmaci.ATC*.

Per realizzare l'interrogazione in modalità QBE bisogna aumentare il numero di campi della griglia della query portandolo a 5. Per farlo si può, per esempio, selezionare il valore 2 tra quelli proposti nella casella combinata **Aggiungi/Cancel**la campi e poi fare clic su **Aggiorna Query**. Nella griglia della query si inseriscono quindi i valori mostrati in figura.

Database: **prontuario**

Esporta Cerca Query da esempio Operazioni Elimina

'farmaci`.`Codice`	'farmaci`.`Descrizione`	'farmaci`.`Prezzo`	'principi`.`ATC`	'principi`.`PrincipioAttivo`
☒	☒	☒	☐	☐
			=Farmaci.ATC	=Acido Folico

Oppure: ☐ E: ☒ Oppure: ☐ E: ☒ Oppure: ☐ E: ☒ Oppure: ☐ E: ☒ Oppure: ☐ E: ☒

Aggiungi ☐ Cancella ☐ Aggiungi ☐ Cancella ☐ Aggiungi ☐ Cancella ☐ Aggiungi ☐ Cancella ☐ Aggiungi ☐ Cancella ☐

0 Aggiungi/Cancella campo: 0 **Aggiorna Query**

SOL-query sul database prontuario:

```
SELECT 'farmaci`.`Codice` ,
'farmaci`.`Descrizione` ,
'farmaci`.`Prezzo`
FROM 'principi` , 'farmaci'
WHERE (('principi`.`ATC`
=Farmaci.ATC) AND
('principi`.`PrincipioAttivo`
='Acido Folico'))
```

Invia Query

Visualizzazione record 0 - 1 (2 Totali, La query ha impiegato 0.0007 sec)

query SQL:

```
SELECT 'farmaci`.`Codice` , 'farmaci`.`Descrizione` , 'farmaci`.`Prezzo`
FROM 'principi` , 'farmaci'
WHERE (
('principi`.`ATC` = Farmaci.ATC
)
AND (
'principi`.`PrincipioAttivo` = 'Acido Folico'
)
)
LIMIT 0 , 30
```

[Modifica] [Spiega SQL] [Crea il codice PHP] [Aggiorna]

Nelle prime tre posizioni della griglia sono inseriti i nomi dei campi che devono essere visualizzati. L'ultima colonna della griglia è stata utilizzata per selezionare l'acido folico come principio attivo e la penultima per realizzare la condizione di congiunzione tra le due tabelle *Farmaci* e *Principi*. Si noti l'assenza del segno di spunta su **Mostra** in corrispondenza dei campi *ATC* e *PrincipioAttivo* di *Principi*, in quanto questi campi sono utilizzati per la costruzione delle condizioni di selezione e di congiunzione tra tabelle e non devono essere visualizzati.

La query viene eseguita facendo clic su **Invia Query** e il risultato è visualizzato in una nuova pagina assieme al comando SQL che l'ha generato.

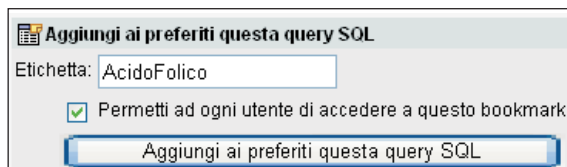
Nella figura è stato inserita, nella parte inferiore a destra, la piccola finestra della pagina di visualizzazione dei risultati che mostra il comando **SELECT** corrispondente alla query costruita in modalità QBE.

I precedenti progetti mostrano le limitate possibilità offerte da phpMyAdmin per costruire query in modalità QBE. Con la scelta **Query da Progetto** si possono realizzare in modalità interattiva proiezioni, selezioni e congiunzioni tra tabelle. L'altra funzionalità offerta dal sistema è di ordinare l'elenco per valori crescenti o decrescenti, su più campi. Per qualsiasi altra esigenza, per esempio per eseguire raggruppamenti, calcoli o applicare le funzioni di aggregazione, dopo avere costruito una versione semplificata dell'interrogazione in modalità grafica, bisogna ricorrere al linguaggio SQL per completare la query con le funzionalità richieste.

6. Salvataggio ed esecuzione di query

Una query eseguita con successo può essere salvata per successive esecuzioni inserendola in una speciale cartella abbinata al database. A ogni query si assegna un nome simbolico specificandolo nella casella **Etichetta** nella piccola finestra **Aggiungi ai preferiti questa query SQL** posta alla fine della pagina di visualizzazione dei risultati della query.

La finestra in figura mostra l'etichetta assegnata alla query dell'esempio precedente (*AcidoFolico*).



L'inserimento nella cartella dei preferiti avviene facendo clic sul pulsante **Aggiungi ai preferiti questa query SQL**.

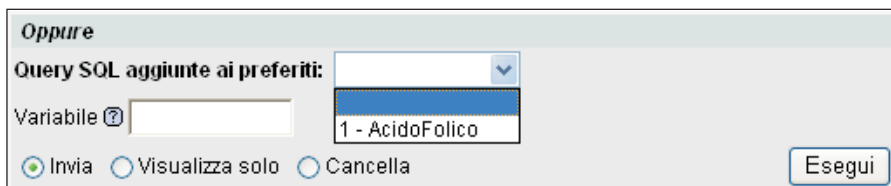
Si noti che, a differenza di quanto accade per i dati di una tabella o di una query, phpMyAdmin non ha funzionalità specifiche per salvare il codice SQL di un'interrogazione in un file esterno. Per farlo bisogna ricopiare il comando da una finestra di visualizzazione del codice SQL della query e incollarlo in un file esterno.

I prossimi progetti mostrano come eseguire una query aggiunta ai preferiti e come sia possibile caricare e eseguire una query il cui codice SQL si trova in un file esterno all'ambiente phpMyAdmin.

ESEMPIO 9 - Eseguire la query di etichetta *AcidoFolico* tra i preferiti del database *Prontuario*.

Facendo clic su **SQL** nella barra dei comandi del database *Prontuario* si accede a una pagina che permette di specificare il codice SQL dell'interrogazione oppure eseguirla da un file esterno al sistema o, infine, ricercare la query fra quelle presenti nei *preferiti* (indicato anche come *storico dell'SQL*).

La figura evidenzia la finestra che permette di scegliere la query desiderata da una casella combinata che elenca le query presenti nello storico dell'SQL.



Si notino nella parte inferiore della finestra tre bottoni che permettono alternativamente: di eseguire la query scelta, oppure di visualizzarne il codice o di eliminarla dai preferiti.

Dopo avere scelto la query dall'elenco e messo il segno di spunta su **Invia**, facendo clic sul pulsante **Esegui** la query viene eseguita e il risultato è visualizzato in una nuova pagina.

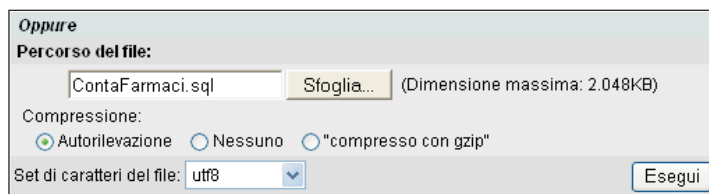
Supponiamo di volere conoscere il numero di farmaci prodotti dalle diverse ditte presenti nel database. Desiderando eseguire la ricerca sia dalla linea comandi che da phpMyAdmin o da altri ambienti, si prepara il comando in un file esterno in modo da poterlo sottomettere a MySQL con il comando **SOURCE**. L'esempio mostra come eseguire il comando da phpMyAdmin.

ESEMPIO 10 - Salvare in un file esterno un comando SELECT che conta i farmaci prodotti da ogni ditta e, successivamente, eseguire il comando da phpMyAdmin.

Il comando SELECT da eseguire è il seguente:

```
SELECT Ditte.IDDitta, NomeDitta, Count(Codice)
FROM Ditte INNER JOIN Farmaci USING (IDDitta)
GROUP BY Ditte.IDDitta
```

Si prepara il comando con un editor di testi e lo si salva sul disco in un file di nome *ContaFarmaci.sql*. Per mandarlo in esecuzione si procede come nel caso dell'esempio precedente facendo clic su **SQL** nella barra dei comandi del database *Prontuario*. Nella pagina che si apre, subito dopo la finestra per l'immissione del codice SQL della query, c'è un riquadro come quello in figura.



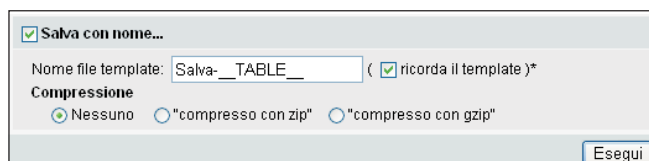
Per eseguire una query salvata in precedenza si specifica il nome del file che la contiene, in genere tramite il pulsante **Sfoglia**, nella casella **Percorso del file**, e la si manda in esecuzione facendo clic su **Esegui**.

L'ultimo esempio mostra come sia possibile eseguire un comando SQL salvato in un file esterno. Il file dell'esempio contiene un comando SELECT, ma l'approccio è più generale e, operando come sopra, è possibile eseguire qualsiasi comando SQL (o un insieme di comandi SQL) salvato in un file esterno.

È possibile quindi sottoporre al sistema un file esterno con il salvataggio dei dati e della struttura di una tabella del database per ricrearla in caso di necessità.

ESEMPIO 11 - Salvare la struttura e i dati della tabella Farmaci esportandola in un file esterno di nome *Salva-farmaci.sql* e, dopo avere eliminato la tabella dal database, ricrearla utilizzando il file *Salva-farmaci.sql*.

Per creare il file di salvataggio dei dati e della struttura della tabella *Farmaci* si esegue l'esportazione dei dati di una tabella: si fa clic su **Esporta** nella barra dei comandi di una finestra di visualizzazione della tabella *Farmaci*, si sceglie la modalità di esportazione **SQL**, e si accettano le opzioni proposte dal sistema facendo attenzione a mettere il segno di spunta su **Salva con nome** e di precisare il nome del file di *dump* come nella seguente figura.



Il file di viene creato dopo avere fatto clic su **Esegui** e aver confermato la scelta in una nuova finestra.

Per eliminare la tabella *Farmaci* si può usare il pulsante **Elimina** nella barra dei comandi, in corrispondenza di una pagina di visualizzazione di questa tabella.

Il sistema prima di eseguire il comando DROP TABLE che elimina la tabella *Farmaci* dal database chiede conferma in una finestra come quella nella figura a lato.



Per ripristinare lo stato del database ricostruendo struttura e dati della tabella eliminata si usa il file *Salva-farmaci.sql* e lo si sottomette al sistema, come nell'esempio precedente, mandandolo in esecuzione come una qualsiasi query.

Il sistema esegue i comandi contenuti nel file e informa dell'avvenuta esecuzione visualizzando, in una nuova pagina, sia il comando eseguito sia il successo o meno dell'esecuzione.

```
La query è stata eseguita con successo

query SQL:
-- phpMyAdmin SQL Dump
-- version 2.5.1
-- http://www.phpmyadmin.net
--
-- Host: localhost
-- Generato il: 03 Gen, 2006 at 10:58 AM
-- Versione MySQL: 4.1.9
-- Versione PHP: 4.3.10
--
-- Database: `prontuario`
--
--
-- Struttura della tabella `farmaci`
--
CREATE TABLE `farmaci` (
  `Codice` VARCHAR( 20 ) NOT NULL DEFAULT "",
  `Denominazione` VARCHAR( 40 ) NOT NULL DEFAULT "",
  `Descrizione` VARCHAR( 255 ) NOT NULL DEFAULT "",
  `Prezzo` DECIMAL( 12, 2 ) NOT NULL DEFAULT '0.00',
  `ATC` VARCHAR( 10 ) NOT NULL DEFAULT "",
  `IDDitta` INT( 11 ) NOT NULL DEFAULT '0',
  `DataAutMin` DATE NOT NULL DEFAULT '0000-00-00',
  PRIMARY KEY ( `Codice` ),
  KEY `ATC` ( `ATC` , `IDDitta` ),
  KEY `IDDitta` ( `IDDitta` )
) ENGINE = INNODB DEFAULT CHARSET = latin1; # MySQL ha restituito un insieme vuoto (i.e. zero righe).
--
-- Dump dei dati per la tabella `farmaci`
--
INSERT INTO `farmaci`
VALUES (
  '001537012', 'FARMOTAL', '500 MG POLVERE PER SOLUZIONE PER INFUSIONE 1 FLACONE 500 MG', 6.15, 'N01AF03', 310, '1996-08-03'
); # Righe interessate:1

--
-- Limiti per le tabelle scaricate
--
--
-- Limiti per la tabella `farmaci`
--
ALTER TABLE `farmaci` ADD CONSTRAINT `farmaci_ibfk_1` FOREIGN KEY ( `ATC` ) REFERENCES `principi` ( `ATC` ),
ADD CONSTRAINT `farmaci_ibfk_2` FOREIGN KEY ( `IDDitta` ) REFERENCES `ditte` ( `IDDitta` ); # Righe interessate:20

[Modifica] [Crea il codice PHP]
```

Nella figura dove sono stati eliminati, per brevità, numerosi comandi INSERT, è riportato il messaggio di esecuzione del file *Salva-farmaci.sql* con il listato del comando eseguito.

7. Strumenti per la documentazione

Si desidera controllare che la tabella *Farmaci* dell'Esempio 11 sia stata correttamente ricreata nella struttura e nei dati, riportando il database *Prontuario* nella situazione di partenza. Dalla seguente figura, che visualizza la struttura del database *Prontuario*, si nota per esempio che la tabella *Farmaci* contiene 20 record ed è di tipo InnoDB.

	Tabella	Azione	Record	Tipo	Collation	Dimensione	In eccesso
☐	ditte	      	13	InnoDB	latin1_swedish_ci	16,0 KB	-
☐	farmaci	      	20	InnoDB	latin1_swedish_ci	48,0 KB	-
☐	principi	      	12	InnoDB	latin1_swedish_ci	16,0 KB	-
	3 tabella(e)	Totali	45	--	latin1_swedish_ci	80,0 KB	0 Bytes
Seleziona tutti / Deseleziona tutti Se selezionati:							
Visualizza per stampa Data Dictionary							

Per controllare che la tabella *Farmaci* sia stata ricreata con i campi correttamente definiti assieme alle chiavi esterne, esaminiamo le informazioni prodotte dalla funzionalità attivata facendo clic su **Data Dictionary**.

La funzionalità **Data Dictionary** visualizza una pagina come quella in figura dove sono descritti i dati delle tabelle che compongono il database. La figura mostra che il dizionario dei dati, dopo la visualizzazione, può anche essere stampato con il bottone **Stampa** in basso.

Server:  localhost ▶ Database:  prontouario						
ditte						
Commenti sulla tabella: InnoDB free: 3072 kB						
Campo	Tipo	Null	Predefinito	Collegamenti a	Commenti	MIME
IDDitta	int(11)	No				
NomeDitta	varchar(255)	No				
farmaci						
Commenti sulla tabella: InnoDB free: 3072 kB; ('ATC') REFER `prontuario/principi`('ATC'); ('IDDitta') REFER `prontuario/ditte`('IDDitta')						
Campo	Tipo	Null	Predefinito	Collegamenti a	Commenti	MIME
Codice	varchar(20)	No				
Denominazione	varchar(40)	No				
Descrizione	varchar(255)	No				
Prezzo	decimal(12,2)	No	0.00			
ATC	varchar(10)	No		principi -> ATC		
IDDitta	int(11)	No	0	ditte -> IDDitta		
DataAutMin	date	No	0000-00-00			
principi						
Commenti sulla tabella: InnoDB free: 3072 kB						
Campo	Tipo	Null	Predefinito	Collegamenti a	Commenti	MIME
ATC	varchar(10)	No				
PrincipioAttivo	varchar(255)	No				
Stampa						

L'esame del dizionario dei dati evidenzia che la tabella *Farmaci* è stata ricreata esattamente, con le chiavi esterne correttamente associate alle altre tabelle del database.

Un importante strumento di documentazione messo a disposizione da phpMyAdmin produce il dizionario dei dati in pagine *pdf*. Mostriamo, per esempio, come ottenerlo per il database *Prontuario*.

ESEMPIO 12 - Preparare un file pdf con il dizionario dei dati del database Prontuario.

Nella finestra del database bisogna fare clic sul pulsante **Operazioni** della barra dei comandi. Nella parte bassa della finestra compare il riquadro **PDF**. Facendo clic su **Modifica pagine PDF**, si apre una nuova finestra come quella in figura.

Server: localhost Database: prontuario

Prego scegliere una Page da modificare 6: Schema di Prontuario ☒ Modifica ☐ Cancella

Crea una nuova Page ☐ (Impaginazione automatica)

Seleziona Tables

Tabella	Cancella	X	Y
ditte	☐ Cancella	20	20
farmaci	☐ Cancella	110	60
principi	☐ Cancella	200	20
--	☐ Cancella	200	20

Mostra lo schema del PDF:

☐ Mostra la griglia

☒ Mostra il colore

☐ Mostra la dimensione delle tabelle

☐ mostra tutte le Tabelle con la stessa larghezza?

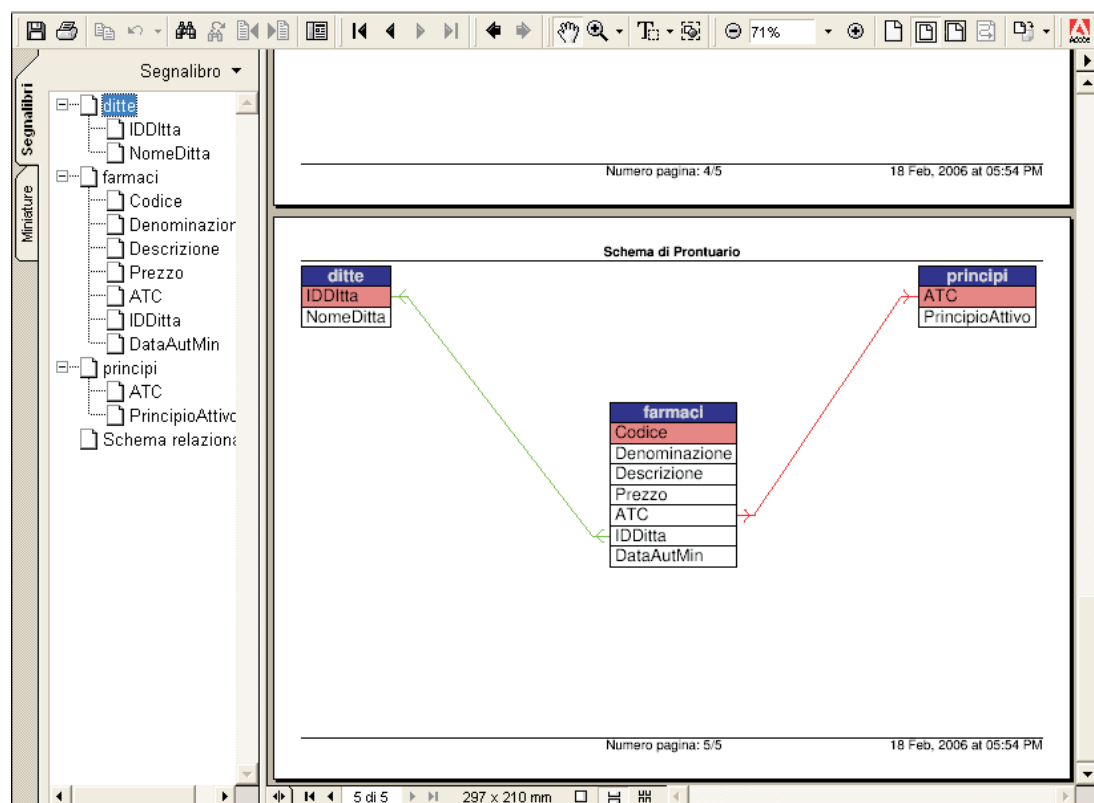
☒ Data Dictionary

Formato del Data Dictionary Orizzontale

Dimensioni carta A4

A questo punto bisogna assegnare un nome alla pagina da creare nella casella **Crea una nuova Page** e fare clic su **Esegui** e, successivamente, selezionare la pagina appena creata scegliendola tra quelle della casella combinata **Prego scegliere una Page da modificare**. Per costruire il dizionario dei dati bisogna selezionare le tabelle che devono comparire nel dizionario, scegliendo nell'elenco e confermando con il primo tasto **Esegui**; impostare anche, nelle colonne successive dello schema, le coordinate **X** e **Y** (esprese in mm rispetto all'angolo in alto a sinistra del foglio). La posizione delle tabelle può essere successivamente modificata in modo visuale facendo clic sul pulsante **(dis)attiva scratchboard** e trascinando le tabelle con il mouse all'interno dell'area del foglio.

Facendo clic sul secondo pulsante **Esegui**, in basso, si ottiene l'anteprima del documento *pdf*: esso contiene i metadati del database, riferiti alle tabelle selezionate, e lo schema relazionale. Il sistema genera un rapporto composto da cinque pagine: nella prima pagina c'è un indice, seguito dai metadati delle tre tabelle del database, una pagina per tabella e nell'ultima pagina è visualizzato lo schema relazionale del database con evidenziate graficamente le associazioni tra le tabelle. La figura mostra l'ultima pagina del rapporto.



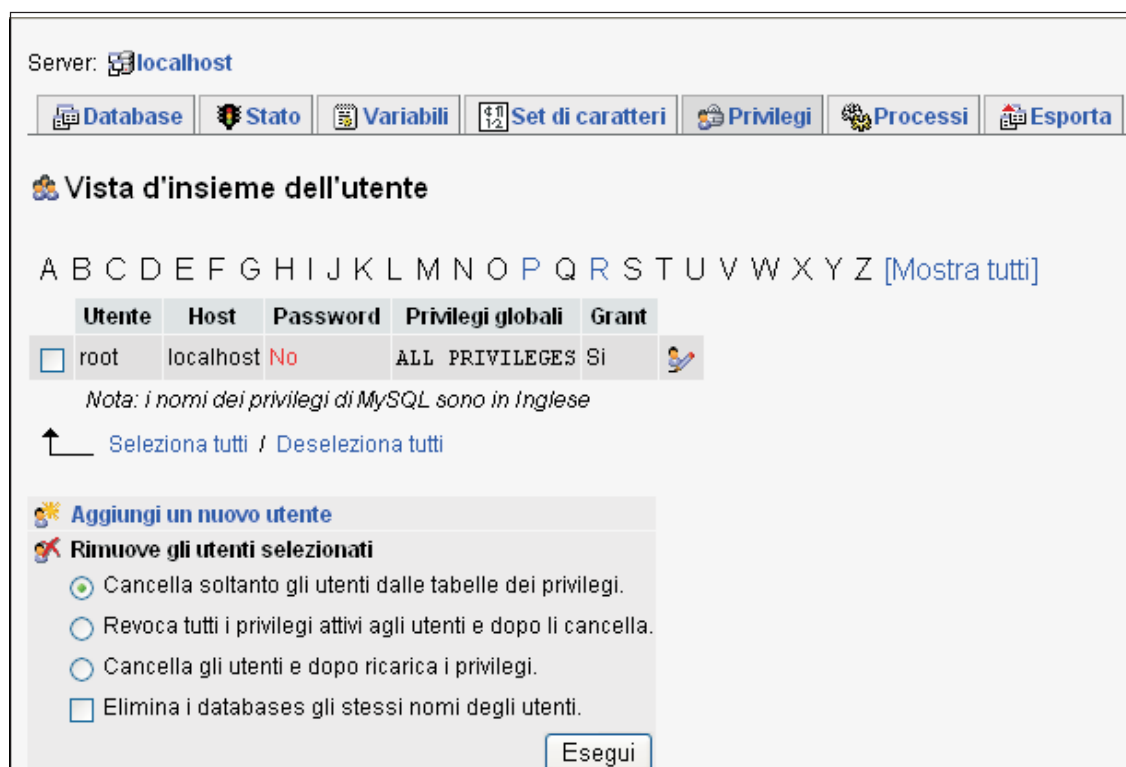
Il dizionario dei dati può essere stampato o salvato in un file agendo sulle piccole icone nella barra superiore della finestra dell'anteprima del documento.

8. Amministrazione degli utenti

In questo approfondimento esamineremo alcune funzionalità di phpMyAdmin che permettono di gestire gli utenti del database, ossia di creare nuovi utenti, concedendo loro le necessarie autorizzazioni per accedere e modificare i dati e la struttura delle tabelle del database.

Facendo clic sul pulsante **Privilegi** nella finestra di visualizzazione del server si accede alla pagina in figura dalla quale si attivano tutte le funzionalità per amministrare gli utenti.

La finestra elenca gli utenti in una tabella che mostra le principali caratteristiche di ogni utente e permette, facendo clic sull'icona  **Modifica Privilegi**, di modificare tutte le caratteristiche e le autorizzazioni concesse a un utente.



Nella parte inferiore della pagina si nota una finestra con le scelte: **Aggiungi un nuovo utente** che permette di creare nuovi utenti e **Rimuove gli utenti selezionati** per eliminare utenti dal sistema. Nel prossimo esempio si farà uso di queste funzionalità per creare un nuovo utente abilitandolo a manipolare le tabelle di *Prontuario*.

ESEMPIO 13 - Creare un utente di nome *Utente 1* per il database *Prontuario*. Utente 1 deve essere abilitato a eseguire solo operazioni di interrogazione e manipolazione dei dati del database. In altre parole: *Utente 1* deve poter visualizzare, inserire, cancellare e modificare dati di tutte le tabelle di *Prontuario*, ma non può né eliminarle né modificarne la struttura.

Facendo clic su **Aggiungi un nuovo utente** viene visualizzata una pagina nella quale si indicano le caratteristiche generali dell'utente, quali nome, server al quale può accedere, password, e assegnargli *permessi* (o *privilegi*) *globali*.

I permessi globali sono permessi relativi a tutti i database presenti sul server e, come si vede dalla figura, sono suddivisi in permessi sui dati, sulla struttura della tabelle e all'amministrazione dei database. Se, per esempio, a un utente è concesso un permesso globale CREATE significa che potrà eseguire il comando CREATE TABLE per ogni database del server.

Poiché *Utente 1* deve poter accedere al solo database *Prontuario* non gli viene concesso alcun permesso globale. La creazione di un utente con le caratteristiche precisate in questa pagina avviene facendo clic sul pulsante **Esegui** alla base della pagina.

🔧 Aggiungi un nuovo utente

Informazioni di Login		
Nome utente:	Utilizza campo text: ▼	Utente 1
Host:	Locale ▼	localhost
Password:	Utilizza campo text: ▼	•••••
Reinserisci:		•••••

Privilegi globali		
<i>Nota: i nomi dei privilegi di MySQL sono in Inglese</i>		
Seleziona tutti Deseleziona tutti		
Dati	Struttura	Amministrazione
☐ SELECT	☐ CREATE	☐ GRANT
☐ INSERT	☐ ALTER	☐ SUPER
☐ UPDATE	☐ INDEX	☐ PROCESS
☐ DELETE	☐ DROP	☐ RELOAD
☐ FILE	☐ CREATE TEMPORARY TABLES	☐ SHUTDOWN
		☐ SHOW DATABASES
		☐ LOCK TABLES
		☐ REFERENCES
		☐ EXECUTE
		☐ REPLICATION CLIENT
		☐ REPLICATION SLAVE

Limiti di risorse	
<i>N.B.: 0 (zero) significa nessun limite.</i>	
MAX QUERIES PER HOUR	0
MAX UPDATES PER HOUR	0
MAX CONNECTIONS PER HOUR	0

Il sistema visualizza il comando con il quale ha creato l'utente, con i permessi concessi, in una nuova pagina dalla quale è poi possibile modificare e dettagliare ulteriormente i permessi dell'utente. Per farlo si accede alla sezione **Privilegi specifici al database** per definire i permessi relativi al database *Prontuario* che viene scelto in una casella combinata con l'elenco di tutti i database.

🔧 Utente 'Utente 1'@'localhost' - Database prontuario

• Modifica Privilegi

Privilegi specifici al database		
<i>Nota: i nomi dei privilegi di MySQL sono in Inglese</i>		
Seleziona tutti Deseleziona tutti		
Dati	Struttura	Amministrazione
☒ SELECT	☐ CREATE	☐ GRANT
☒ INSERT	☐ ALTER	☐ LOCK TABLES
☒ UPDATE	☐ INDEX	☐ REFERENCES
☒ DELETE	☐ DROP	
	☐ CREATE TEMPORARY TABLES	

Per autorizzare l'*Utente 1* a eseguire interrogazioni sul database *Prontuario* e manipolare i dati delle tabelle che lo compongono si mettono i segni di spunta come nella figura precedente e si fa clic sul pulsante **Esegui**.

phpMyAdmin visualizza il comando che ha modificato i permessi di *Utente 1* e i permessi concessi, che possono essere ulteriormente dettagliati per definire, per esempio, permessi per una specifica tabella o per i singoli campi di una tabella.

Hai aggiornato i permessi per 'Utente 1'@'localhost'.

query SQL:
`GRANT SELECT , INSERT , UPDATE , DELETE ON `prontuario` . * TO 'Utente 1'@'localhost';`

Se a questo punto si torna alla pagina di partenza della gestione degli utenti si visualizzano le caratteristiche generali dell'utente *Utente 1*.

	Utente	Host	Password	Privilegi globali	Grant	
☐	Utente 1	localhost	Si	USAGE	No	
☐	root	localhost	No	ALL PRIVILEGES	Si	