

Chapter 3

Introduction to relational databases and MySQL

Objectives

Applied

1. Use phpMyAdmin to review the data and structure of the tables in a database, to import and run SQL scripts that create databases, and to create users with limited privileges.
2. Code simple SELECT, INSERT, UPDATE, and DELETE statements, and use phpMyAdmin to run them.

Objectives (continued)

Knowledge

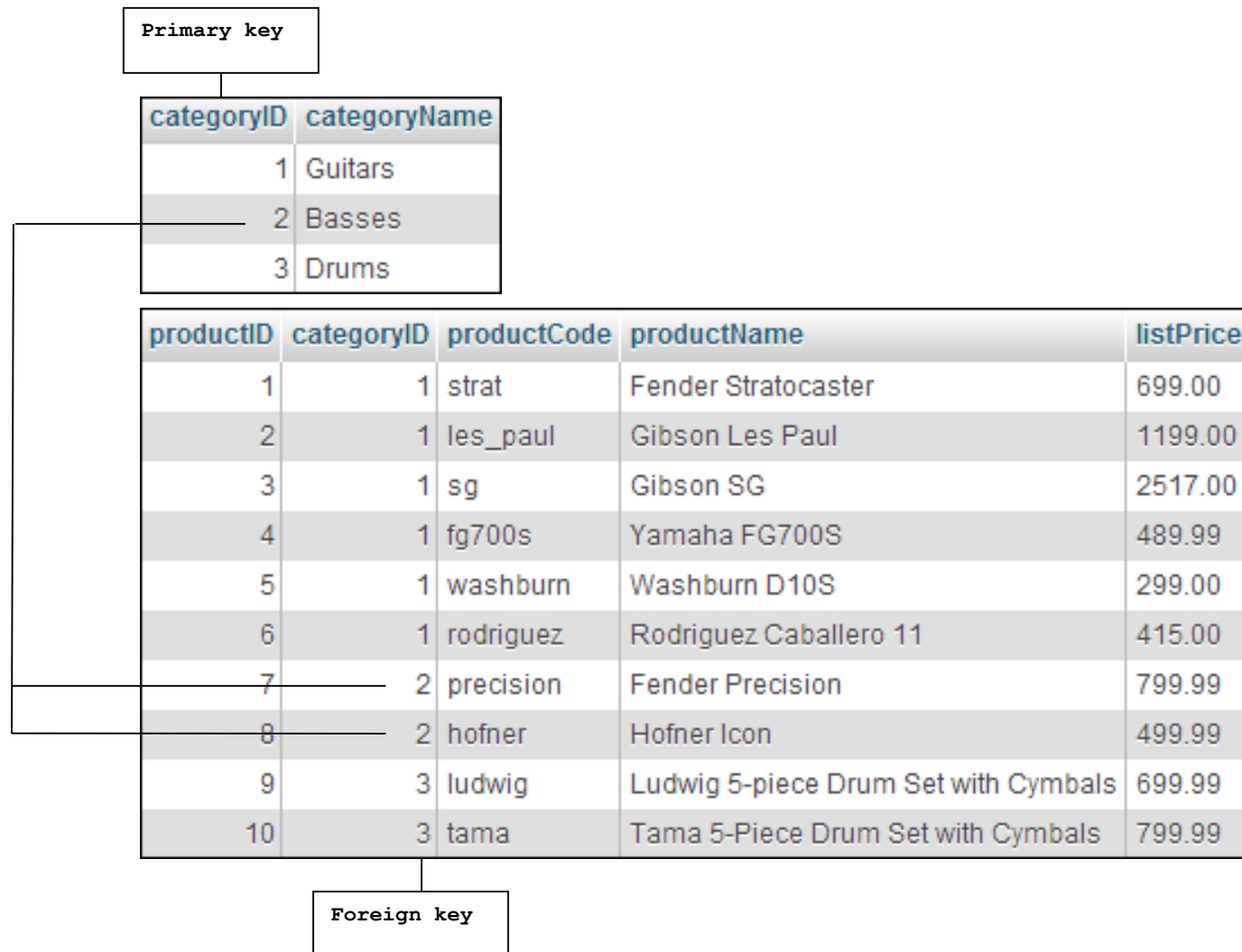
1. Describe the structure of a database table.
2. Describe how the tables in a relational database are related using these terms: primary key and foreign key.
3. Identify the three types of relationships that can exist between two tables.
4. Describe the way the columns in a table are defined using these terms: data type, NULL value, default value, and auto-increment column.
5. Describe the use of SELECT statements, including the use of inner joins.
6. Describe the use of INSERT, UPDATE, and DELETE statements.
7. Describe the way the creation of users and the assignment of privileges affect how a MySQL database can be used.

A products table

The diagram illustrates a database table structure. A box labeled 'Primary key' points to the 'productID' column. A box labeled 'Columns' points to the header row. A box labeled 'Rows' points to the data rows.

| productID | categoryID | productCode | productName | listPrice |
|-----------|------------|-------------|--------------------------------------|-----------|
| 1 | 1 | strat | Fender Stratocaster | 699.00 |
| 2 | 1 | les_paul | Gibson Les Paul | 1199.00 |
| 3 | 1 | sg | Gibson SG | 2517.00 |
| 4 | 1 | fg700s | Yamaha FG700S | 489.99 |
| 5 | 1 | washburn | Washburn D10S | 299.00 |
| 6 | 1 | rodriguez | Rodriguez Caballero 11 | 415.00 |
| 7 | 2 | precision | Fender Precision | 799.99 |
| 8 | 2 | hofner | Hofner Icon | 499.99 |
| 9 | 3 | ludwig | Ludwig 5-piece Drum Set with Cymbals | 699.99 |
| 10 | 3 | tama | Tama 5-Piece Drum Set with Cymbals | 799.99 |

The relationship between two tables in a database



Terms

- foreign key
- one-to-many relationship
- one-to-one relationship
- many-to-many relationship

The columns of the products table

| Name | Type | Collation | Attributes | Null | Default | Extra |
|-------------------------|---------------|-------------------|------------|------|-------------|----------------|
| <u>productID</u> | int(11) | | | No | <i>None</i> | AUTO_INCREMENT |
| categoryID | int(11) | | | No | <i>None</i> | |
| productCode | varchar(10) | latin1_swedish_ci | | No | <i>None</i> | |
| productName | varchar(255) | latin1_swedish_ci | | No | <i>None</i> | |
| listPrice | decimal(10,2) | | | No | <i>None</i> | |

MySQL Numeric Data Types

from <http://www.tutorialspoint.com/mysql/mysql-data-types.htm>

- **INT** - You can specify a width of up to 11 digits.
- **FLOAT(M,D)** - A floating-point number that cannot be unsigned. You can define the display length (M) and the number of decimals (D). This is not required and will default to 10,2, where 2 is the number of decimals and 10 is the total number of digits (including decimals). Decimal precision can go to 24 places for a FLOAT.
- **DOUBLE(M,D)** - A double precision floating-point number that cannot be unsigned. You can define the display length (M) and the number of decimals (D). This is not required and will default to 16,4, where 4 is the number of decimals. Decimal precision can go to 53 places for a DOUBLE. REAL is a synonym for DOUBLE.
- **DECIMAL(M,D)** - An unpacked floating-point number that cannot be unsigned. In unpacked decimals, each decimal corresponds to one byte. Defining the display length (M) and the number of decimals (D) is required. NUMERIC is a synonym for DECIMAL.

MySQL String Data Types

from <http://www.tutorialspoint.com/mysql/mysql-data-types.htm>

- **CHAR(M)** - A fixed-length string between 1 and 255 characters in length (for example CHAR(5)), right-padded with spaces to the specified length when stored. Defining a length is not required, but the default is 1.
- **VARCHAR(M)** - A variable-length string between 1 and 255 characters in length; for example VARCHAR(25). You must define a length when creating a VARCHAR field.
- **BLOB or TEXT** - A field with a maximum length of 65535 characters. BLOBs are "Binary Large Objects" and are used to store large amounts of binary data, such as images or other types of files. Fields defined as TEXT also hold large amounts of data; the difference between the two is that sorts and comparisons on stored data are case sensitive on BLOBs and are not case sensitive in TEXT fields. You do not specify a length with BLOB or TEXT.
- **ENUM** - An enumeration, which is a fancy term for list. When defining an ENUM, you are creating a list of items from which the value must be selected (or it can be NULL). For example, if you wanted your field to contain "A" or "B" or "C", you would define your ENUM as ENUM ('A', 'B', 'C') and only those values (or NULL) could ever populate that field.

MySQL Date & Time Data Types

from <http://www.tutorialspoint.com/mysql/mysql-data-types.htm>

- **DATE** - A date in YYYY-MM-DD format, between 1000-01-01 and 9999-12-31. For example, December 30th, 1973 would be stored as 1973-12-30.
- **DATETIME** - A date and time combination in YYYY-MM-DD HH:MM:SS format, between 1000-01-01 00:00:00 and 9999-12-31 23:59:59. For example, 3:30 in the afternoon on December 30th, 1973 would be stored as 1973-12-30 15:30:00.
- **TIMESTAMP** - A timestamp between midnight, January 1, 1970 and sometime in 2037. This looks like the previous DATETIME format, only without the hyphens between numbers; 3:30 in the afternoon on December 30th, 1973 would be stored as 19731230153000 (YYYYMMDDHHMMSS).
- **TIME** - Stores the time in HH:MM:SS format.
- **YEAR(M)** - Stores a year in 2-digit or 4-digit format. If the length is specified as 2 (for example YEAR(2)), YEAR can be 1970 to 2069 (70 to 69). If the length is specified as 4, YEAR can be 1901 to 2155. The default length is 4.

The SELECT statement syntax for all columns

```
SELECT *  
FROM table-1  
[WHERE selection-criteria]  
[ORDER BY column-1 [ASC|DESC]  
          [, column-2 [ASC|DESC] ...] ]
```

A SELECT statement that gets all columns

```
SELECT * FROM products  
WHERE categoryID = 2
```

The result table

| productID | categoryID | productCode | productName | listPrice |
|-----------|------------|-------------|------------------|-----------|
| 7 | 2 | precision | Fender Precision | 799.99 |
| 8 | 2 | hofner | Hofner Icon | 499.99 |

The syntax for selected columns

```
SELECT column-1 [, column-2] ...  
FROM table-1  
[WHERE selection-criteria]  
[ORDER BY column-1 [ASC|DESC]  
          [, column-2 [ASC|DESC] ...] ]
```

A statement that gets selected columns and rows

```
SELECT productName, listPrice  
FROM products  
WHERE listPrice < 500  
ORDER BY listPrice ASC
```

The result table

| productName | listPrice |
|------------------------|-----------|
| Washburn D10S | 299.00 |
| Rodriguez Caballero 11 | 415.00 |
| Yamaha FG700S | 489.99 |
| Hofner Icon | 499.99 |

A statement that gets data from two related tables

```
SELECT categoryName, productName, listPrice
FROM categories
    INNER JOIN products
        ON categories.categoryID = products.categoryID
WHERE listPrice > 800
ORDER BY listPrice ASC
```

The result table

| categoryName | productName | listPrice |
|--------------|-----------------|-----------|
| Guitars | Gibson Les Paul | 1199.00 |
| Guitars | Gibson SG | 2517.00 |

The syntax for the INSERT statement

```
INSERT INTO table-name [(column-list)]  
VALUES (value-list)
```

A statement that adds one row to a table

```
INSERT INTO products  
    (categoryID, productCode, productName, listPrice)  
VALUES  
    (1, 'tele', 'Fender Telecaster', 599.00)
```

A statement that uses the MySQL NOW function to get the current date

```
INSERT INTO orders (customerID, orderDate)  
VALUES (1, NOW())
```

The syntax for the UPDATE statement

```
UPDATE table-name  
SET expression-1 [, expression-2] ...  
WHERE selection-criteria
```

A statement that updates a column in one row

```
UPDATE products  
SET productName =  
    'Ludwig 5-Piece Kit with Zildjian Cymbals'  
WHERE productCode = 'ludwig'
```

A statement that updates multiple rows

```
UPDATE products  
SET listPrice = 299  
WHERE categoryID = 1
```

The syntax for the DELETE statement

```
DELETE FROM table-name  
WHERE selection-criteria
```

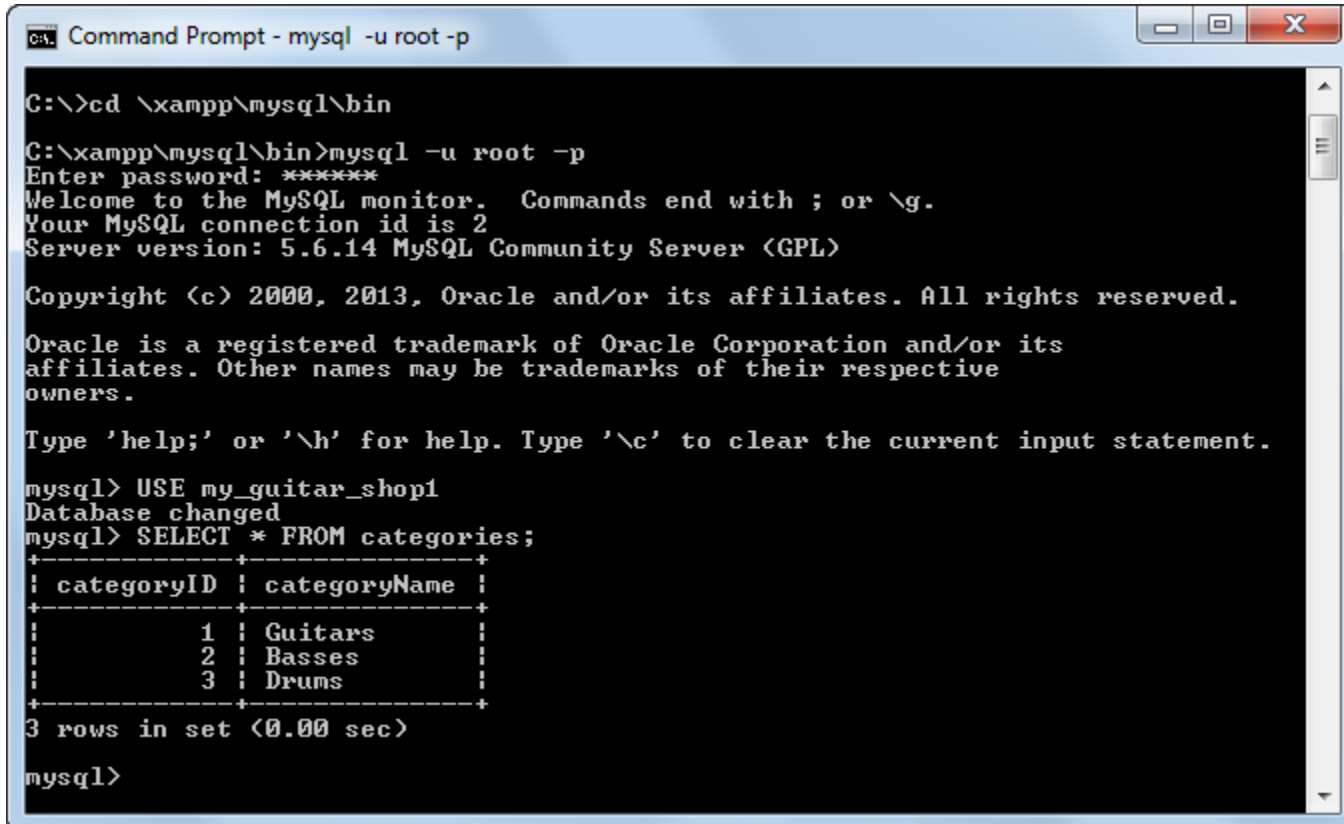
A statement that deletes one row from a table

```
DELETE FROM products  
WHERE productID = 1
```

A statement that deletes multiple rows

```
DELETE FROM products  
WHERE listPrice > 200
```


A command-line client



```
CA: Command Prompt - mysql -u root -p

C:\>cd \xampp\mysql\bin

C:\xampp\mysql\bin>mysql -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.6.14 MySQL Community Server (GPL)

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

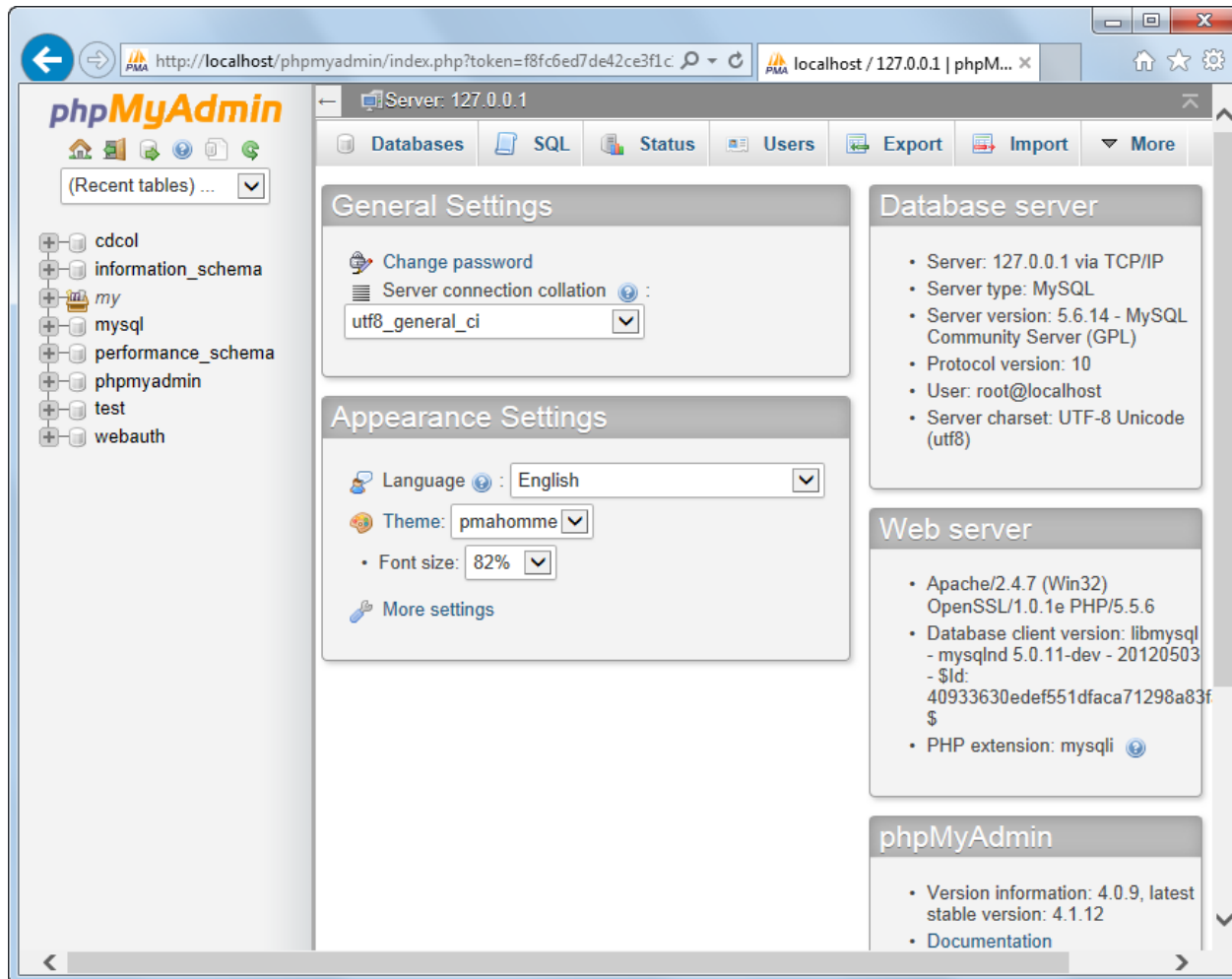
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

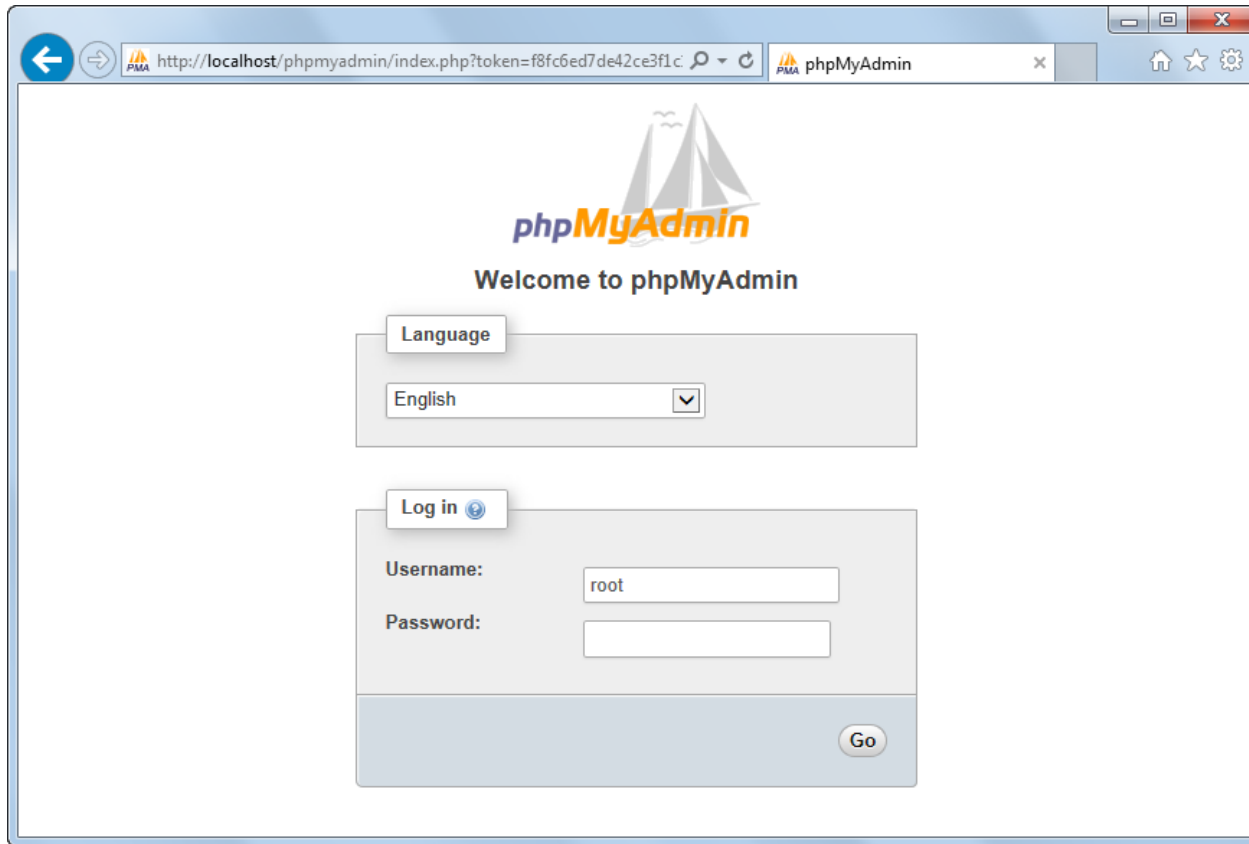
mysql> USE my_guitar_shop1
Database changed
mysql> SELECT * FROM categories;
+-----+-----+
| categoryID | categoryName |
+-----+-----+
| 1 | Guitars |
| 2 | Bases |
| 3 | Drums |
+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

A web-based client



The Welcome page



The Home and Logout buttons at the top of the sidebar on most pages



How to change your password

- Click the Home button (the house icon). Then, click the Change Password link.
- On the Change Password page, enter and re-enter your new password, and click the Go button.

Running a SQL script that creates a database

The screenshot shows the phpMyAdmin interface. On the left is a sidebar with a tree view of databases and tables. The main area is titled 'Importing into the table "categories"' and contains several sections for configuring the import process.

Server: 127.0.0.1 » Database: my_guitar_shop1 » Table: categories

Tools: Browse | Structure | SQL | Search | Insert | Export | Import | More

(Recent tables) ...

- cdcol
- information_schema
- my
- mysql
- New
- performance_schema
- phpmyadmin
- test
- webauth

File to Import:

File may be compressed (gzip, bzip2, zip) or uncompressed.
A compressed file's name must end in `.[format].[compression]`. Example: `.sql.zip`

Browse your computer: create_db.sql (Max: 2,048KiB)

Character set of the file:

Partial Import:

☒ Allow the interruption of an import in case the script detects it is close to the PHP timeout limit. *(This might be a good way to import large files, however it can break transactions.)*

Skip this number of queries (for SQL) or lines (for other formats), starting from the first one:

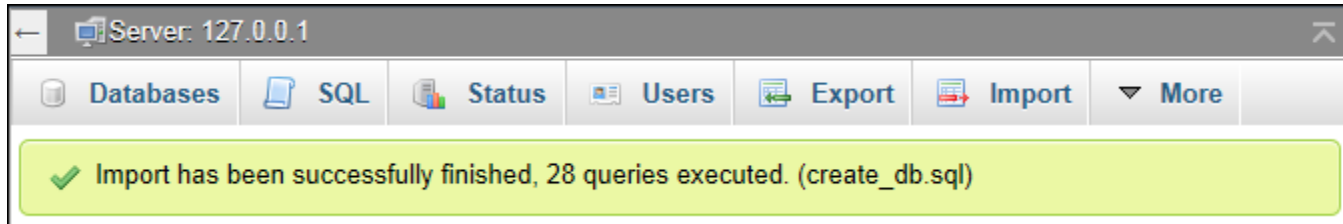
Format:

Format-Specific Options:

SQL compatibility mode:

☒ Do not use `AUTO_INCREMENT` for zero values

A success message after importing a SQL script



How to import and run a SQL script

1. Click the Import tab, go to the “File to Import” section, click the Browse button, and select the file that contains the script.
2. Click the Go button. This runs the script that’s in the file.

The Structure tab for the my_guitar_shop 1 database

| | | | | | | | | | | | | | | | | | |
|---|------------|--------|-----------|--------|--------|-------|------|--------|--------|-------------------|--------|------------|--|------------|--|----------|--|
| Server: 127.0.0.1 » Database: my_guitar_shop1 | | | | | | | | | | | | | | | | | |
| Structure | | SQL | | Search | | Query | | Export | | Import | | Operations | | Privileges | | Routines | |
| Table | | Action | | | | | | Rows | Type | Collation | Size | Overhead | | | | | |
| <input type="checkbox"/> | categories | Browse | Structure | Search | Insert | Empty | Drop | ~3 | InnoDB | latin1_swedish_ci | 16 KiB | - | | | | | |
| <input type="checkbox"/> | orders | Browse | Structure | Search | Insert | Empty | Drop | ~0 | InnoDB | latin1_swedish_ci | 16 KiB | - | | | | | |
| <input type="checkbox"/> | products | Browse | Structure | Search | Insert | Empty | Drop | ~10 | InnoDB | latin1_swedish_ci | 32 KiB | - | | | | | |
| 3 tables | | Sum | | | | | | 13 | InnoDB | latin1_swedish_ci | 64 KiB | 0 B | | | | | |

The Browse tab for the categories table

Server: 127.0.0.1 » Database: my_guitar_shop1 » Table: categories

Browse Structure SQL Search Insert Export Import

✓ Showing rows 0 - 2 (3 total, Query took 0.0010 sec)

```
SELECT *  
FROM `categories`  
LIMIT 0, 30
```

☐ Profiling [[Inline](#)]

Show : Start row: Number of rows: Headers every rows

Sort by key: ▼

+ Options

| | | | categoryID | categoryName |
|--------------------------|------|------|------------|--------------|
| <input type="checkbox"/> | Edit | Copy | Delete | 1 Guitars |
| <input type="checkbox"/> | Edit | Copy | Delete | 2 Basses |
| <input type="checkbox"/> | Edit | Copy | Delete | 3 Drums |

↑ ☐ Check All With selected: Delete Export

The SQL tab with a statement ready to run

The screenshot shows a web-based SQL client interface. At the top, a status bar indicates the server is '127.0.0.1' and the database is 'my_guitar_shop1'. Below this is a navigation bar with tabs for 'Structure', 'SQL', 'Search', 'Query', 'Export', 'Import', 'Operations', and a 'More' dropdown. The 'SQL' tab is active. A message box says 'Run SQL query/queries on database my_guitar_shop1:'. The main area contains a text editor with the following SQL query:

```
1 SELECT categoryName, productName, listPrice
2 FROM categories
3 INNER JOIN products ON categories.categoryID = products.categoryID
4 WHERE listPrice > 800
5 ORDER BY listPrice ASC
```

Below the text editor is a 'Clear' button. Underneath that is a label 'Bookmark this SQL query:' followed by an empty text input field. At the bottom, there is a footer bar with a 'Delimiter' dropdown set to ';', a checked checkbox for 'Show this query here again', an unchecked checkbox for 'Retain query box', and a 'Go' button.