VISUAL QUICKPRO GUIDE

COVERS PHP 5 & 7

PHP and MySQL for Dynamic Web Sites

Fifth Edition

LARRY ULLMAN

LEARN THE QUICK AND EASY WAY!

VISUAL QUICKPRO GUIDE

PHP and MySQL for Dynamic Web Sites

Fifth Edition

LARRY ULLMAN

Peachpit Press

To establish a table's type:

1. Find your MySQL server's available table types **A**:

   `SHOW ENGINES;`

   The **`SHOW ENGINES`** command, when executed on the MySQL server, will reveal not only the available storage engines but also the default storage engine. It will help to know this information when it's time to choose a table type for your database.

2. If any of your tables requires a **`FULLTEXT`** index and you're not using MySQL 5.6.4 or greater, make it a MyISAM table.

   Again, **`FULLTEXT`** indexes and searches are discussed in the next chapter, but I'll say now that the *messages* table in the *forums* example will require a **`FULLTEXT`** index. Therefore, this table can use InnoDB if you're using MySQL 5.6.4 or greater but must be MyISAM if you're not.

3. If any of your tables requires support for transactions, make it an InnoDB table.

   Yes, again, transactions are discussed in the next chapter, but the storage engines ought to be determined now. Neither the *forums* nor *users* tables in the *forums* database will require transactions.

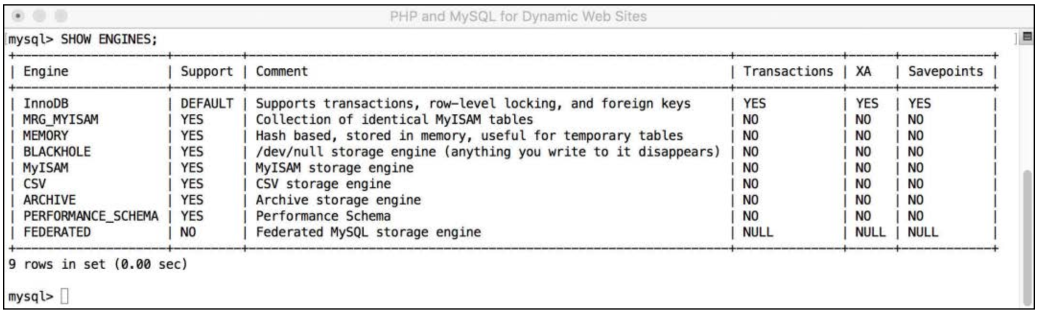
4. If neither of the above applies to a table, use the default storage engine.

   Table 6.7 identifies the storage engines to be used by the tables in the *forums* database with the caveat that if you're not using MySQL 5.6.4 or greater, the *messages* table should be MyISAM.

TIP MySQL has several other table types, but MyISAM and InnoDB are the two most important, by far. The `MEMORY` type creates the table in memory, making it an extremely fast table but with absolutely no permanence.

TABLE 6.7 The Forum Database Table Types

Table	Table Type
`forums`	`InnoDB`
`messages`	`InnoDB`
`users`	`InnoDB`



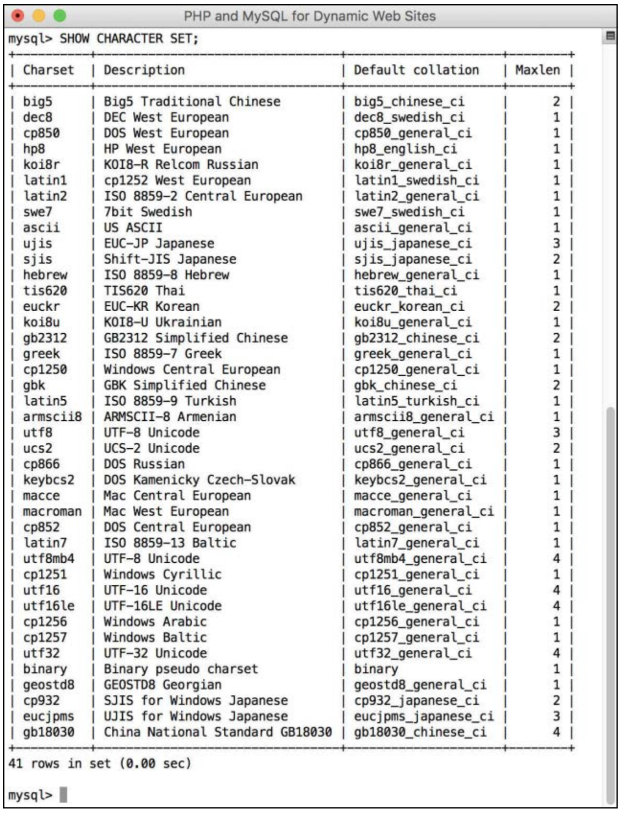
```
mysql> SHOW ENGINES;
+--------------------+---------+----------------------------------------------------------------+--------------+------+------------+
| Engine             | Support | Comment                                                        | Transactions | XA   | Savepoints |
+--------------------+---------+----------------------------------------------------------------+--------------+------+------------+
| InnoDB             | DEFAULT | Supports transactions, row-level locking, and foreign keys     | YES          | YES  | YES        |
| MRG_MYISAM         | YES     | Collection of identical MyISAM tables                          | NO           | NO   | NO         |
| MEMORY             | YES     | Hash based, stored in memory, useful for temporary tables      | NO           | NO   | NO         |
| BLACKHOLE          | YES     | /dev/null storage engine (anything you write to it disappears) | NO           | NO   | NO         |
| MyISAM             | YES     | MyISAM storage engine                                          | NO           | NO   | NO         |
| CSV                | YES     | CSV storage engine                                             | NO           | NO   | NO         |
| ARCHIVE            | YES     | Archive storage engine                                         | NO           | NO   | NO         |
| PERFORMANCE_SCHEMA | YES     | Performance Schema                                             | NO           | NO   | NO         |
| FEDERATED          | NO      | Federated MySQL storage engine                                 | NULL         | NULL | NULL       |
+--------------------+---------+----------------------------------------------------------------+--------------+------+------------+
9 rows in set (0.00 sec)

mysql> 
```

A To confirm what table types your MySQL installation supports, run this command (in the mysql client, here, or phpMyAdmin).

Languages and MySQL

Chapter 1, “Introduction to PHP,” briefly introduced the concept of *encodings*. An HTML page or PHP script can specify its encoding, which dictates what characters, and therefore languages, are supported. The same is true for a MySQL database: by setting your database’s encoding, you can impact what characters can be stored in it. To see a list of encodings supported by your version of MySQL, run a **`SHOW CHARACTER SET`** command **A**. Note that the phrase *character set* is being used in MySQL to mean *encoding* (which I’ll generally follow in this section to be consistent with MySQL).

Each character set in MySQL has one or more *collations*. Collation refers to the rules used for comparing characters in a set. It’s like alphabetization, but it considers numbers, spaces, and other characters as well. Collation is tied to the character set being used, reflecting both the kinds of characters present in that language and the cultural habits of people who generally use the language. For example, how text is sorted in English is not the same as it is in traditional Spanish or in Arabic. Other considerations include: Are upper- and lowercase versions of a character considered to be the same or different (i.e., is it a case-sensitive comparison)? How do accented characters get sorted? Is a space counted or ignored?



```
mysql> SHOW CHARACTER SET;
+----------+---------------------------------+---------------------+--------+
| Charset  | Description                     | Default collation   | Maxlen |
+----------+---------------------------------+---------------------+--------+
| big5     | Big5 Traditional Chinese        | big5_chinese_ci     |      2 |
| dec8     | DEC West European               | dec8_swedish_ci     |      1 |
| cp850    | DOS West European               | cp850_general_ci    |      1 |
| hp8      | HP West European                | hp8_english_ci      |      1 |
| koi8r    | KOI8-R Relcom Russian           | koi8r_general_ci    |      1 |
| latin1   | cp1252 West European            | latin1_swedish_ci   |      1 |
| latin2   | ISO 8859-2 Central European     | latin2_general_ci   |      1 |
| swe7     | 7bit Swedish                    | swe7_swedish_ci     |      1 |
| ascii    | US ASCII                        | ascii_general_ci    |      1 |
| ujis     | EUC-JP Japanese                 | ujis_japanese_ci    |      3 |
| sjis     | Shift-JIS Japanese              | sjis_japanese_ci    |      2 |
| hebrew   | ISO 8859-8 Hebrew               | hebrew_general_ci   |      1 |
| tis620   | TIS620 Thai                     | tis620_thai_ci      |      1 |
| euckr    | EUC-KR Korean                   | euckr_korean_ci     |      2 |
| koi8u    | KOI8-U Ukrainian                | koi8u_general_ci    |      1 |
| gb2312   | GB2312 Simplified Chinese       | gb2312_chinese_ci   |      2 |
| greek    | ISO 8859-7 Greek                | greek_general_ci    |      1 |
| cp1250   | Windows Central European        | cp1250_general_ci   |      1 |
| gbk      | GBK Simplified Chinese          | gbk_chinese_ci      |      2 |
| latin5   | ISO 8859-9 Turkish              | latin5_turkish_ci   |      1 |
| armscii8 | ARMSCII-8 Armenian              | armscii8_general_ci |      1 |
| utf8     | UTF-8 Unicode                   | utf8_general_ci     |      3 |
| ucs2     | UCS-2 Unicode                   | ucs2_general_ci     |      2 |
| cp866    | DOS Russian                     | cp866_general_ci    |      1 |
| keybcs2  | DOS Kamenicky Czech-Slovak      | keybcs2_general_ci  |      1 |
| macce    | Mac Central European            | macce_general_ci    |      1 |
| macroman | Mac West European               | macroman_general_ci |      1 |
| cp852    | DOS Central European            | cp852_general_ci    |      1 |
| latin7   | ISO 8859-13 Baltic              | latin7_general_ci   |      1 |
| utf8mb4  | UTF-8 Unicode                   | utf8mb4_general_ci  |      4 |
| cp1251   | Windows Cyrillic                | cp1251_general_ci   |      1 |
| utf16    | UTF-16 Unicode                  | utf16_general_ci    |      4 |
| utf16le  | UTF-16LE Unicode                | utf16le_general_ci  |      4 |
| cp1256   | Windows Arabic                  | cp1256_general_ci   |      1 |
| cp1257   | Windows Baltic                  | cp1257_general_ci   |      1 |
| utf32    | UTF-32 Unicode                  | utf32_general_ci    |      4 |
| binary   | Binary pseudo charset           | binary              |      1 |
| geostd8  | GEOSTD8 Georgian                | geostd8_general_ci  |      1 |
| cp932    | SJIS for Windows Japanese       | cp932_japanese_ci   |      2 |
| eucjpms  | UJIS for Windows Japanese       | eucjpms_japanese_ci |      3 |
| gb18030  | China National Standard GB18030 | gb18030_chinese_ci  |      4 |
+----------+---------------------------------+---------------------+--------+
41 rows in set (0.00 sec)

mysql> 
```

A The list of character sets supported by this MySQL installation.

To view MySQL's available collations, run this query **B**, replacing *charset* with the proper value from the result in the last query **A**:

```
SHOW COLLATION LIKE 'charset%'
```

The results of this query will also indicate the default collation for that character set. The names of collations use a concluding *ci* to indicate case-insensitivity, *cs* for case-sensitivity, and *bin* for binary.

Generally speaking, I recommend using the UTF-8 character set, with its default collation. More importantly, *the character set in use by the database should match that of your PHP scripts*. If you're not using UTF-8 in your PHP scripts, use the matching encoding in the database. If the default collation doesn't adhere to the conventions of the language primarily in use, then adjust the collation accordingly.

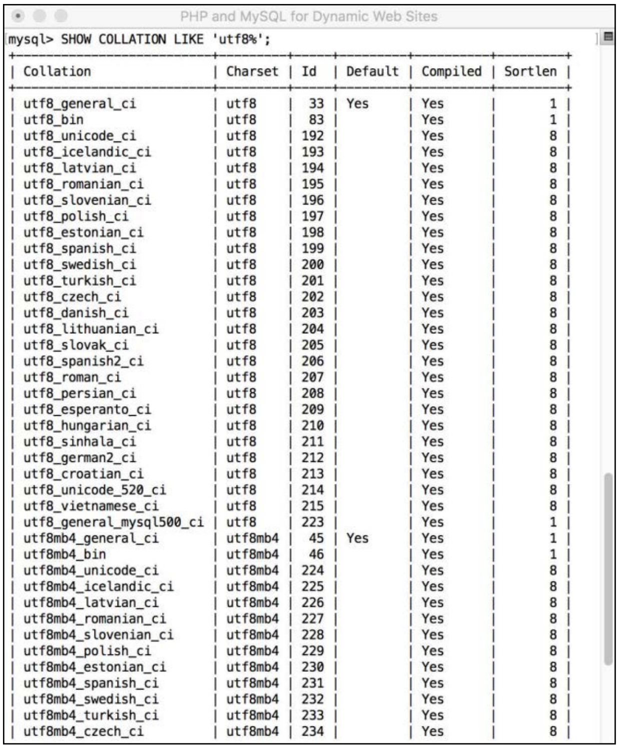
In MySQL, the server as a whole, each database, each table, and even every string column can have a defined character set and collation. To set these values when you create a database, use

```
CREATE DATABASE name
CHARACTER SET charset
COLLATE collation
```

To set these values when you create a table, use

```
CREATE TABLE name (
column definitions
)
CHARACTER SET charset
COLLATE collation
```

continues on next page



```
mysql> SHOW COLLATION LIKE 'utf8%';
+--------------------------+---------+-----+---------+----------+---------+
| Collation                | Charset | Id  | Default | Compiled | Sortlen |
+--------------------------+---------+-----+---------+----------+---------+
| utf8_general_ci          | utf8    |  33 | Yes     | Yes      |       1 |
| utf8_bin                 | utf8    |  83 |         | Yes      |       1 |
| utf8_unicode_ci          | utf8    | 192 |         | Yes      |       8 |
| utf8_icelandic_ci        | utf8    | 193 |         | Yes      |       8 |
| utf8_latvian_ci          | utf8    | 194 |         | Yes      |       8 |
| utf8_romanian_ci         | utf8    | 195 |         | Yes      |       8 |
| utf8_slovenian_ci        | utf8    | 196 |         | Yes      |       8 |
| utf8_polish_ci           | utf8    | 197 |         | Yes      |       8 |
| utf8_estonian_ci         | utf8    | 198 |         | Yes      |       8 |
| utf8_spanish_ci          | utf8    | 199 |         | Yes      |       8 |
| utf8_swedish_ci          | utf8    | 200 |         | Yes      |       8 |
| utf8_turkish_ci          | utf8    | 201 |         | Yes      |       8 |
| utf8_czech_ci            | utf8    | 202 |         | Yes      |       8 |
| utf8_danish_ci           | utf8    | 203 |         | Yes      |       8 |
| utf8_lithuanian_ci       | utf8    | 204 |         | Yes      |       8 |
| utf8_slovak_ci           | utf8    | 205 |         | Yes      |       8 |
| utf8_spanish2_ci         | utf8    | 206 |         | Yes      |       8 |
| utf8_roman_ci            | utf8    | 207 |         | Yes      |       8 |
| utf8_persian_ci          | utf8    | 208 |         | Yes      |       8 |
| utf8_esperanto_ci        | utf8    | 209 |         | Yes      |       8 |
| utf8_hungarian_ci        | utf8    | 210 |         | Yes      |       8 |
| utf8_sinhala_ci          | utf8    | 211 |         | Yes      |       8 |
| utf8_german2_ci          | utf8    | 212 |         | Yes      |       8 |
| utf8_croatian_ci         | utf8    | 213 |         | Yes      |       8 |
| utf8_unicode_520_ci      | utf8    | 214 |         | Yes      |       8 |
| utf8_vietnamese_ci       | utf8    | 215 |         | Yes      |       8 |
| utf8_general_mysql500_ci | utf8    | 223 |         | Yes      |       1 |
| utf8mb4_general_ci       | utf8mb4 |  45 | Yes     | Yes      |       1 |
| utf8mb4_bin              | utf8mb4 |  46 |         | Yes      |       1 |
| utf8mb4_unicode_ci       | utf8mb4 | 224 |         | Yes      |       8 |
| utf8mb4_icelandic_ci     | utf8mb4 | 225 |         | Yes      |       8 |
| utf8mb4_latvian_ci       | utf8mb4 | 226 |         | Yes      |       8 |
| utf8mb4_romanian_ci      | utf8mb4 | 227 |         | Yes      |       8 |
| utf8mb4_slovenian_ci     | utf8mb4 | 228 |         | Yes      |       8 |
| utf8mb4_polish_ci        | utf8mb4 | 229 |         | Yes      |       8 |
| utf8mb4_estonian_ci      | utf8mb4 | 230 |         | Yes      |       8 |
| utf8mb4_spanish_ci       | utf8mb4 | 231 |         | Yes      |       8 |
| utf8mb4_swedish_ci       | utf8mb4 | 232 |         | Yes      |       8 |
| utf8mb4_turkish_ci       | utf8mb4 | 233 |         | Yes      |       8 |
| utf8mb4_czech_ci         | utf8mb4 | 234 |         | Yes      |       8 |
```

B The list of collations available in the UTF-8 character set. The first one, *utf_general_ci*, is the default.

To establish the character set and collation for a column, add the right clause to the column's definition (you'd only use this for text types):

```
CREATE TABLE name (
something TEXT
CHARACTER SET charset
COLLATE collation
...)
```

In each of these cases, both clauses are optional. If omitted, a default character set or collation will be used.

Establishing the character set and collation when you define a database affects what data can be stored; you cannot store a character in a column if its encoding doesn't support that character. A second issue is the encoding used to communicate with MySQL. If you want to store Chinese characters in a table with a Chinese encoding, those characters will need to be transferred using the same encoding. To do so within the mysql client, set the encoding using just

```
CHARSET charset
```

With phpMyAdmin, the encoding to be used is established in the application itself (i.e., written in the configuration file).

At this point in time, every aspect of the database design for the *forums* example has been covered, so let's create that database in MySQL, including its indexes, storage engines, character sets, and collations.

To assign character sets and collations:

1. Access MySQL using whatever client you prefer.

   Like the preceding chapter, this one will also use the mysql client for all of its examples. You are welcome to use phpMyAdmin or other tools as the interface to MySQL.

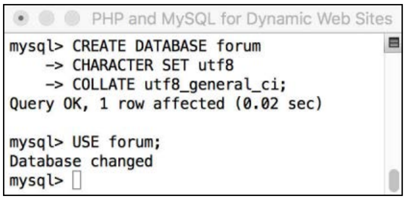
2. Create the *forum* database **C**:

   ```
   CREATE DATABASE forum
   CHARACTER SET utf8
   COLLATE utf8_general_ci;
   USE forum;
   ```

   Depending on your setup, you may not be allowed to create your own databases. If that's the case, just use the database provided to you and add the following tables to it. Note that in the **CREATE DATABASE** command, the character set and collation are also defined. By doing so at this point, you ensure that every table will use those settings.

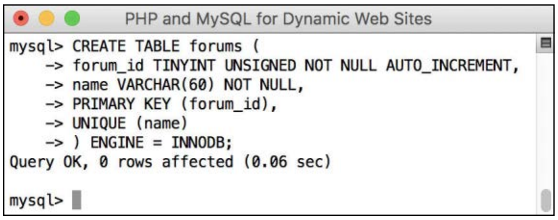
3. Create the *forums* table **D**:

   ```
   CREATE TABLE forums (
   forum_id TINYINT UNSIGNED NOT
   → NULL AUTO_INCREMENT,
   name VARCHAR(60) NOT NULL,
   PRIMARY KEY (forum_id),
   UNIQUE (name)
   ) ENGINE = INNODB;
   ```



```
mysql> CREATE DATABASE forum
    -> CHARACTER SET utf8
    -> COLLATE utf8_general_ci;
Query OK, 1 row affected (0.02 sec)

mysql> USE forum;
Database changed
mysql> 
```

C The first steps are to create and select the database.



```
mysql> CREATE TABLE forums (
    -> forum_id TINYINT UNSIGNED NOT NULL AUTO_INCREMENT,
    -> name VARCHAR(60) NOT NULL,
    -> PRIMARY KEY (forum_id),
    -> UNIQUE (name)
    -> ) ENGINE = INNODB;
Query OK, 0 rows affected (0.06 sec)

mysql> 
```

D Creating the first table.

It does not matter in what order you create your tables, but I'll make the *forums* table first. Remember that you can enter your SQL queries over multiple lines for convenience.

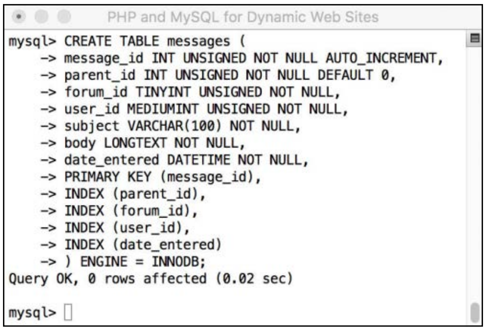
This table contains only two columns (which will happen frequently in a normalized database). Because I don't expect there to be many forums, the primary key is a really small type (**TINYINT**). If you wanted to add descriptions of each forum, a **VARCHAR(255)** or **TINYTEXT** column could be added to this table. This table uses the InnoDB storage engine.

4. Create the *messages* table **E**:

```
CREATE TABLE messages (
message_id INT UNSIGNED
→ NOT NULL AUTO_INCREMENT,
parent_id INT UNSIGNED
→ NOT NULL DEFAULT 0,
forum_id TINYINT UNSIGNED
→ NOT NULL,
user_id MEDIUMINT UNSIGNED
→ NOT NULL,
subject VARCHAR(100) NOT NULL,
body LONGTEXT NOT NULL,
date_entered DATETIME NOT NULL,
PRIMARY KEY (message_id),
INDEX (parent_id),
INDEX (forum_id),
INDEX (user_id),
INDEX (date_entered)
) ENGINE = INNODB;
```

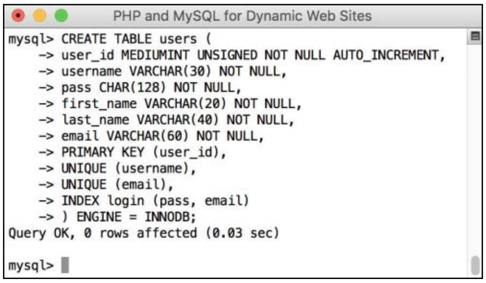
The primary key for this table has to be big, since it could have lots and lots of records. The three foreign key columns—*forum_id*, *parent_id*, and *user_id*—will all be the same size and type as their primary key counterparts. The subject is limited to 100 characters and the body of each message can be a lot of text. The *date_entered* field is a **DATETIME** type.

All three tables use the InnoDB storage engine, unless you're using an older version of MySQL, in which case you'll probably need to make this one MyISAM.



```
mysql> CREATE TABLE messages (
    -> message_id INT UNSIGNED NOT NULL AUTO_INCREMENT,
    -> parent_id INT UNSIGNED NOT NULL DEFAULT 0,
    -> forum_id TINYINT UNSIGNED NOT NULL,
    -> user_id MEDIUMINT UNSIGNED NOT NULL,
    -> subject VARCHAR(100) NOT NULL,
    -> body LONGTEXT NOT NULL,
    -> date_entered DATETIME NOT NULL,
    -> PRIMARY KEY (message_id),
    -> INDEX (parent_id),
    -> INDEX (forum_id),
    -> INDEX (user_id),
    -> INDEX (date_entered)
    -> ) ENGINE = INNODB;
Query OK, 0 rows affected (0.02 sec)

mysql>
```

E Creating the second table.



```
mysql> CREATE TABLE users (
    -> user_id MEDIUMINT UNSIGNED NOT NULL AUTO_INCREMENT,
    -> username VARCHAR(30) NOT NULL,
    -> pass CHAR(128) NOT NULL,
    -> first_name VARCHAR(20) NOT NULL,
    -> last_name VARCHAR(40) NOT NULL,
    -> email VARCHAR(60) NOT NULL,
    -> PRIMARY KEY (user_id),
    -> UNIQUE (username),
    -> UNIQUE (email),
    -> INDEX login (pass, email)
    -> ) ENGINE = INNODB;
Query OK, 0 rows affected (0.03 sec)

mysql>
```

F The database's third and final table.

5. Create the *users* table **F**:

```
CREATE TABLE users (
user_id MEDIUMINT UNSIGNED
→ NOT NULL AUTO_INCREMENT,
username VARCHAR(30) NOT NULL,
pass CHAR(128) NOT NULL,
first_name VARCHAR(20) NOT NULL,
last_name VARCHAR(40) NOT NULL,
email VARCHAR(60) NOT NULL,
PRIMARY KEY (user_id),
UNIQUE (username),
UNIQUE (email),
INDEX login (pass, email)
) ENGINE = INNODB;
```

continues on next page

Most of the columns here mimic those in the *sitename* database's *users* table, created in the preceding two chapters. The pass column is defined as **`CHAR(128)`**, because the **`SHA2()`** function will be used and it always returns a string 128 characters long (see Chapter 5).

This table uses the InnoDB engine.

6. If desired, confirm the database's structure **G**:

```
SHOW TABLES;
SHOW COLUMNS FROM forums;
SHOW COLUMNS FROM messages;
SHOW COLUMNS FROM users;
```

The **`SHOW`** command reveals information about a database or a table. This step is optional because MySQL reports on the success of each query as it is entered. Still, it's always nice to remind yourself of a database's structure.

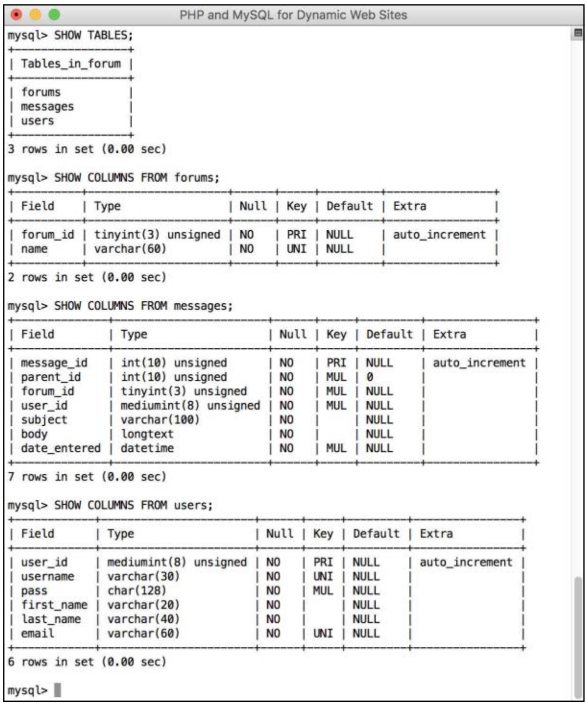
TIP Collations in MySQL can also be specified within a query, to affect the results:

```
SELECT ... ORDER BY column
COLLATE collation
SELECT ... WHERE column LIKE 'value'
COLLATE collation
```

TIP The `CONVERT()` function can convert text from one character set to another.

TIP You can change the default character set or collation for a database or table using an ALTER command, discussed in Chapter 7.

TIP Because different character sets require more space to represent a string, you will likely need to increase the size of a column for UTF-8 characters. Do this before changing a column's encoding so that no data is lost.



```
mysql> SHOW TABLES;
+-----------------+
| Tables_in_forum |
+-----------------+
| forums          |
| messages        |
| users           |
+-----------------+
3 rows in set (0.00 sec)

mysql> SHOW COLUMNS FROM forums;
+----------+---------------------+------+-----+---------+----------------+
| Field    | Type                | Null | Key | Default | Extra          |
+----------+---------------------+------+-----+---------+----------------+
| forum_id | tinyint(3) unsigned | NO   | PRI | NULL    | auto_increment |
| name     | varchar(60)         | NO   | UNI | NULL    |                |
+----------+---------------------+------+-----+---------+----------------+
2 rows in set (0.00 sec)

mysql> SHOW COLUMNS FROM messages;
+--------------+-----------------------+------+-----+---------+----------------+
| Field        | Type                  | Null | Key | Default | Extra          |
+--------------+-----------------------+------+-----+---------+----------------+
| message_id   | int(10) unsigned      | NO   | PRI | NULL    | auto_increment |
| parent_id    | int(10) unsigned      | NO   | MUL | 0       |                |
| forum_id     | tinyint(3) unsigned   | NO   | MUL | NULL    |                |
| user_id      | mediumint(8) unsigned | NO   | MUL | NULL    |                |
| subject      | varchar(100)          | NO   |     | NULL    |                |
| body         | longtext              | NO   |     | NULL    |                |
| date_entered | datetime              | NO   | MUL | NULL    |                |
+--------------+-----------------------+------+-----+---------+----------------+
7 rows in set (0.00 sec)

mysql> SHOW COLUMNS FROM users;
+------------+-----------------------+------+-----+---------+----------------+
| Field      | Type                  | Null | Key | Default | Extra          |
+------------+-----------------------+------+-----+---------+----------------+
| user_id    | mediumint(8) unsigned | NO   | PRI | NULL    | auto_increment |
| username   | varchar(30)           | NO   | UNI | NULL    |                |
| pass       | char(128)             | NO   | MUL | NULL    |                |
| first_name | varchar(20)           | NO   |     | NULL    |                |
| last_name  | varchar(40)           | NO   |     | NULL    |                |
| email      | varchar(60)           | NO   | UNI | NULL    |                |
+------------+-----------------------+------+-----+---------+----------------+
6 rows in set (0.00 sec)

mysql> 
```

G Check the structure of any database or table using **`SHOW`**.