

Home
Announcements
Modules
Tasks, Tests and Surveys
Discussion and Private Messages
Gradebook
Site Info

users present:

Michael Aubrey
Martha Raup
Henri Salles
Scott Wakefield

MODULES

[View](#) [Preferences](#)[Prev](#) | [Table Of Contents](#) | [Next](#)[Week 9](#) » [Week 9 Tasks](#)

Week 9

Week 9 Tasks

1. Assignment: Read Topic: Introduction to MySQL
2. Read **"Additional Reading"** materials (please scroll down the page).

3. Lab Work: For instructions on what to submit for lab work, please refer to "Tasks, Tests and Surveys", "Lab 3."

Please post all your questions, answers, and helpful tips in "Discussion and Private Messages."

If you know the answers to your classmate's questions in "Discussion and Private Messages," please feel free to post your answers.

Additional Reading

Please note: Web sites can change over time, and as a result some of the links below may be inactive since the publication of this course. If this is the case, try searching for a similar Web site using Yahoo!, Google, or another Internet search engine.

http://www.webdevelopersnotes.com/tutorials/sql/mysql_database_introduction_mysql_beginners_tutorial.php3

<http://www.tizag.com/mysqlTutorial/>

<http://www.mysql.com>

<http://dev.mysql.com/doc/>

<http://www.devshed.com/c/b/MySQL>

<http://www.onlamp.com/onlamp/general/mysql.csp>

<http://www.analysisandsolutions.com/code/mybasic.htm>

<http://www.securityfocus.com/infocus/1726>

<http://databases.about.com/od/shareware/>

<http://hotwired.lycos.com/webmonkey/programming/php/tutorials/tutorial4.html>

<http://www.freewebmasterhelp.com/tutorials/phpmysql>

<http://databases.about.com/od/shareware/MySQL.htm>

Lecture Notes:

Working with Databases and MySQL

In this lesson, we take a break from PHP to learn how to work with MySQL databases. Your goal is to learn the basics of database manipulation. Despite the fact that many people mistakenly believe the MySQL is part of PHP, it is just one of many databases that PHP can manipulate directly or through ODBC. MySQL is primarily used for Web applications and is extremely popular for several reasons, first, and foremost is that it's open source and free.

Introduction to Databases

Formally defined, a database is an ordered collection of information from which a computer program can quickly access information. The information stored in computer databases is actually stored similar to spreadsheets.

Each row in a database table is called a record. A record in a database, for instance, is a single complete set of related information. Each recipe in a recipe database, for instance, is a single database record.

Each column in a database table is called a field. Fields are the individual categories of information stored in a record. Examples of fields that might exist in a recipe database include ingredients, cooking time, cooking temperature, and so on.

A flat-file database stores information in a single table. For simple collections of information, flat-file database can become unwieldy. A better solution for large and complex databases is a relational database. A relational database stores information across multiple related tables.

Understanding Relational Database

A primary table is the main table in a relationship that is referenced by another table. A related table (also called a child table) references a primary table in a relational database. Tables in a relationship are connected using primary and foreign keys.

A primary key is a field that contains a unique identifier for each record in a primary table. A primary key is a field that contains a unique identifier for each record in a primary table. A primary key is a type of index, which identifies records in a database to make retrievals and sorting faster.

An index can consist of just a primary key, or it can be a combination of multiple fields. A foreign key is a field in a related table that refers to the primary key in a primary table. Primary and foreign keys link records across multiple tables in a relational database.

There are three basic types of relationships within a relational database: one-to-one, one-to-many, and many-to-many. A one-to-one relationship exists between two tables when a related table contains exactly one record for each record in the primary table.

A one-to-many relationship exists in a relational database when one record in a primary table has many related records in a related table. Breaking tables into multiple related tables to reduce redundant and duplicate information is called normalization.

A many-to-many relationship exists in a relational database when many records in one table are related to many records in another table.

A junction table creates a one-to-many relationship for each of the two tables in a many-to-many relationship.

Working the Database Management Systems

An application or collection of applications used to access and manage a database is called a database management system, or DBMS. A DBMS is also used to define a database's schema, which is the structure of a database, including its tables, fields, and relationships. A database management system that stores data in a flat-file format is called a flat-file database management system. A database management system that stores data in a relational format is called a relational database management system, or RDBMS.

A query is a structured set of instructions and criteria for retrieving, adding, modifying, and deleting database information. Most database management systems use a data manipulation language or DML for creating queries. Different database management systems support different data manipulation languages. However, structured query language, or SQL (pronounced sequel), has become somewhat of a standard data manipulation language among many database management systems.

Open database connectivity, or ODBC, allows ODBC-compliant applications to access any data source for which there is an ODBC driver. ODBC uses SQL commands (known as ODBC SQL) to allow an ODBC-compliant application to access a database.

Querying Databases with Structured Query Language

In 1991, the X/Open and SQL Access Group created a standardized version of SQL known as the Common Applications Environment (CAE) SQL draft specification. MySQL corresponds primarily to the ANSI SQL standard, although it includes a few of its own extensions to the language.

Logging in to MySQL

To use MySQL Monitor to log in to the MySQL database server, enter the following command:

```
mysql -h host -u user -p
```

You know you're successfully logged on when see the mysql> prompt.

When you first install MySQL, two accounts are created: an anonymous user account and a root account. The anonymous user account allows anyone to log in to MySQL without specifying a username or password. The root account is the primary administrative account for MySQL. It is initially created without a password, so you can log in to the MySQL Monitor with the following command:

```
mysql-u root
```

When you finish working with the MySQL Monitor, you can log out by entering the exit or quit commands.

Working with the MySQL Monitor

The mysql> command prompt in the MySQL Monitor is where most of the action occurs when you create or manipulate databases in MySQL. The SQL keywords you enter in the MySQL Monitor are not case sensitive.

Understanding MySQL Identifiers

In MySQL, you must define identifiers (names) for databases, table, fields, indexes, and aliases. An alias is an alternate name that you can use to refer to a table or field in SQL statements.

Getting Help with MySQL Commands

Most of the commands you enter in the MySQL Monitor are SQL commands. If you type help; or ? at the MySQL command prompt, you should see several support URLs along with command descriptions.

Working with MySQL Databases

When you first install MySQL, two databases are installed: mysql and test. The mysql database contains user accounts and other information that is required for your installation of the MySQL database server. The MySQL installation program installs the test database to ensure that the database server is working properly.

Selecting a Database

No database is selected when you first log in to MySQL. To work with a database, you must first select it by executing the use database statement.

Creating a Database

You use the CREATE DATABASE statement to create a new database. Keep in mind that the CREATE DATABASE statement only creates a new directory for the specified database.

Deleting a Database

To delete a database, you execute the DROP DATABASE statement, which removes all tables from the database and deletes the database itself.

Securing the Initial MySQL Accounts

As you learned earlier, when you first install MySQL, two accounts are created: an anonymous user account and a root account. The anonymous user account allows anyone to log in to MySQL without specifying a username or password, whereas the root account is created without a password. Instead of allowing anyone to access your databases anonymously, you need to delete the anonymous account and assign a password to the root account.

Deleting the Anonymous User Account

You can delete the anonymous account in several ways. The easiest way is to log in with the root account and execute specific commands: DELETE FROM and FLUSH PRIVILEGES. The FLUSH PRIVILEGES statement reloads the MySQL database server's grant tables, which contain the permissions assigned to each user account. In this case, the FLUSH PRIVILEGES statement removes the permissions assigned to the anonymous user account.

Assigning a Password to the Root Account

You have multiple ways to assign a password to the root account. The easiest way is to log in with the root account and execute specific commands: UPDATE and FLUSH PRIVILEGES. UPDATE statement uses the UPDATE keyword to update the password for the root account. The FLUSH PRIVILEGES statement reloads the grant tables.

Creating Users

Typically, for security purposes, you create an account with a password for every user that needs access to your database. Since you'll be building Web applications with PHP, you can't determine how many people will need access to the database. Therefore, instead of creating a separate database account for each visitor, you only need to create a single account that a PHP script uses to access the database for a user by proxy. The term proxy refers to someone or something that acts or performs a request for another person.

Granting Privileges

Use the GRANT statement to create user account and assign privileges, which are the operations that a user can perform with the database. The GRANT statement creates the user account if it does not exist and assigns the specified privileges. If the user account already exists, the GRANT statement just updates the privileges.

Revoking Privileges

Use the REVOKE statement to take away privileges from an existing user account for a specified table or database.

The REVOKE ALL PRIVILEGES statement removes all privileges from a user account for a specified table or database.

Deleting Users

To delete an existing user, execute the DROP USER statement. Be sure that before you attempt to delete a user, you first revoke all privileges assigned to the user account for all databases.

Specifying Field Data Types

MySQL includes numerous data types that are categorized in numeric types, string types, and date/time types. You can find a complete listing of MySQL data types in the MySQL Reference Manual at <http://dev.mysql.com/doc/mysql/en/index.html>.

Creating Tables

To create a table, you use the CREATE TABLE statement, which specifies the table and column names and that data type for the each column.

Deleting Tables

To delete a table, you execute the DROP TABLE statement, which removes all data and the table definition.

Adding Records

You add individual records to a table with the INSERT statement. In most cases, you will probably add individual records to a database with the INSERT INTO statement.

Retrieving Records

Use a SELECT statement to retrieve records from a table. The basic syntax for a SELECT statement is as follows:

```
SELECT criteria FROM table_name;
```

MIN() and MAX()

These functions are very similar and select the lowest and highest figure respectively from a result set.

Syntax:

```
MIN(field)
MAX(field)
```

So a simple example would be to display the least number and most number of tracks that any cd in the database has.

```
mysql> SELECT MIN(cds.tracks), MAX(cds.tracks)
-> FROM cds;
+-----+-----+
| MIN(cds.tracks) | MAX(cds.tracks) |
+-----+-----+
| 11 | 58 |
+-----+-----+
1 row in set (0.00 sec)
```

Sorting Query Results

Use the ORDER BY keyword with SELECT statement to perform an alphanumeric sort of the result returned from a query. To perform a reverse sort, add the DESC keyword after that name of the field by which you want to perform the sort.

Filtering Query Results

The criteria portion of the SELECT statement determines which field to retrieves from a table. You can also specify which records to return by using the WHERE keyword. SQL includes keywords AND and OR that you can use to specify more detailed conditions about the records you want to return.

Updating Records

If you need to update records in a table, you use the UPDATE statement. The UPDATE keyword specifies the name of the table to update, and the SET keyword specifies the value to assign to the fields in the records that match the condition in the WHERE keyword.

Deleting Records

To delete records in a table, you use the DELETE statement. Be careful when you use the DELETE statement because it deletes all records that match the condition. Therefore, be certain to carefully construction assigned to the

Powered by Sakai

(c) 2008, 2009 Etudes, Inc. All rights reserved.
ETUDES - 2.3.x - Sakai 2.3.x - Server "artemis68"