Databases
MySQL
PHPMyAdmin
PHP and MySQL

Peter Cho | 161B | Winter 06

Databases are useful!

Most of the services we use on the Web are provided by web database applications. Web-based email, online shopping, forums and bulletin boards, corporate web sites, and news portals are all database-driven.

The use of databases has several potential advantages.

– separation of design and content, by working with templates
– content often outlasts the design of a Web site
– search and sort capabilities (access to all columns of a DB)
– easy backup and recovery
Databases

PHP supports over 20 types of databases, both commercial and open source.

In this class we are focusing on the MySQL relational database system, using the Structured Query Language (SQL) to communicate with the database.

In a Database Management System (DBMS), running on a database server, the data is structured into tables where each table has some number of columns, each of which has a name and a type (e.g. one table might keep track of all purchased items in an e-business where another table stores the billing and shipping address of the customer, connected through a key)
Database terms

Database
A repository to store data. For example, a database might store all of the data associated with finance in a large company, information about your CD and DVD collection, or the records of an online store.

Table
A part of a database that stores data related to an object, thing, or activity. For example, a table might store data about customers. A table has columns, fields, or attributes. The data is stored as rows or records.

Attributes
The columns in a table. All rows in a table have the same attributes. For example, a customer table might have the attributes name, address, and city. Each attribute has a data type such as string, integer, or date.

Rows
The data entries stored in a table. Rows contain values for each attribute. For example, a row in a customer table might contain the values “Matthew Richardson,” “Punt Road,” and “Richmond.” Rows are also known as records.

Relational model
A formal model that uses database, tables, and attributes to store data and manages the relationship between tables.
Database terms, cont.

(Relational) database management system (DBMS)

A software application that manages data in a database and is based on the relational model. Also known as a database server.

SQL

A standard query language that interacts with a database server. SQL is a set of statements to manage databases, tables, and data. Despite popular belief, SQL does not stand for Structured Query Language and isn’t pronounced Sequel: it’s pronounced as the three-letter acronym S-Q-L and it doesn’t stand for anything.

Primary key

One or more attributes that contain values that uniquely identify each row. For example, a customer table might have the primary key named cust_ID. The cust_ID attribute is then assigned a unique value for each customer. A primary key is a constraint of most tables.

Index

A data structure used for fast access to rows in a table. An index is usually built for the primary key of each table and can then be used to quickly find a particular row. Indexes are also defined and built for other attributes when those attributes are frequently used in queries.
<table>
<thead>
<tr>
<th>Winery ID</th>
<th>Winery name</th>
<th>Address</th>
<th>Region ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Moss Brothers</td>
<td>Smith Rd.</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>Hardy Brothers</td>
<td>Jones St.</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Penfolds</td>
<td>Arthurton Rd.</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>Lindemans</td>
<td>Smith Ave.</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>Orlando</td>
<td>Jones St.</td>
<td>1</td>
</tr>
</tbody>
</table>
Relational database

Table

Winery Table

<table>
<thead>
<tr>
<th>Winery ID</th>
<th>Winery name</th>
<th>Address</th>
<th>Region ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Moss Brothers</td>
<td>Smith Rd.</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>Hardy Brothers</td>
<td>Jones St.</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Penfolds</td>
<td>Arthurton Rd.</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>Lindemans</td>
<td>Smith Ave.</td>
<td>2</td>
</tr>
<tr>
<td>5</td>
<td>Orlando</td>
<td>Jones St.</td>
<td>1</td>
</tr>
</tbody>
</table>

Region Table

<table>
<thead>
<tr>
<th>Region ID</th>
<th>Region name</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Barossa Valley</td>
<td>South Australia</td>
</tr>
<tr>
<td>2</td>
<td>Yarra Valley</td>
<td>Victoria</td>
</tr>
<tr>
<td>3</td>
<td>Margaret River</td>
<td>Western Australia</td>
</tr>
</tbody>
</table>
A more complex entity-relationship model
web-based admin for your database

MySQL 4.1.12
Creating tables with SQL

CREATE TABLE customer (  
cust_id int(5) NOT NULL,  
surname varchar(50),  
firstname varchar(50),  
initial char(1),  
title_id int(3),  
address varchar(50),  
city varchar(50),  
state varchar(20),  
zipcode varchar(10),  
country_id int(4),  
phone varchar(15),  
birth_date char(10),  
PRIMARY KEY (cust_id)  
) type=MyISAM;
Common SQL data types

more at http://dev.mysql.com/doc/, Ch. 11

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>int(length)</code></td>
<td>Integer with a max length; for IDs, age, counters, etc.</td>
</tr>
<tr>
<td><code>decimal(width[,decimal_digits])</code></td>
<td>A number with a <code>width</code> including an optional number of <code>decimal_digits</code> after the decimal point; used for currency, measurements, etc.</td>
</tr>
<tr>
<td><code>datetime</code></td>
<td>Stores a date and time in the format YYYY-MM-DD HH:MM:SS</td>
</tr>
<tr>
<td><code>time</code></td>
<td>Stores a time in the format HH:MM:SS</td>
</tr>
<tr>
<td><code>date</code></td>
<td>Stores a date in the format YYYY-MM-DD</td>
</tr>
<tr>
<td><code>timestamp</code></td>
<td>Stores the date and time in the format YYYYMMDDHHMMSS</td>
</tr>
<tr>
<td><code>varchar(length)</code></td>
<td>Unpadded, variable-length text string with a specific maximum <code>length</code></td>
</tr>
<tr>
<td><code>char(length)</code></td>
<td>Padded, fixed-length text string of size <code>length</code></td>
</tr>
<tr>
<td><code>blob</code></td>
<td>Stores up to 64 KB of data</td>
</tr>
</tbody>
</table>
Basic SQL statements

// Creating a new entry (row) in a table
INSERT INTO items VALUES (0, 'screwdriver', 293848, 29.95, '04-12-01')

// Deleting a row in a table
DELETE FROM items WHERE number=223344

// Updating values in a specific row or multiple rows
UPDATE items SET date='05-01-12' where id=0

// Reading out rows where the condition is true
SELECT * FROM items WHERE date >= '04-08-01' AND price <= 50

// Reading out specific fields/values where the condition is true
SELECT items.title, items.price, customers.firstName, customers.lastName, customer.zipCode WHERE items.number=293848
Connecting to the MySQL server with PHP

// server connect
$host = 'users.design.ucla.edu';
$usr = 'petercho';
$pwd = 'myPassword';
$db = 'petercho';

mysql_connect($host, $usr, $pwd) or die(mysql_error());
mysql_select_db($db);
SELECT statement

SELECT is used to retrieve rows selected from one or more tables.

```php
$news = mysql_query("SELECT id, date, title, text, url FROM upcoming ORDER BY date");

$news = mysql_query("SELECT id, date, title, text, url FROM upcoming WHERE title="Talk" ORDER BY date DESC");
```
SELECT statements

```sql
SELECT surname, firstname FROM customer;
```

```
+-----------+-----------+
<table>
<thead>
<tr>
<th>surname</th>
<th>firstname</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marzalla</td>
<td>Dimitria</td>
</tr>
<tr>
<td>LaTrobe</td>
<td>Anthony</td>
</tr>
<tr>
<td>Fong</td>
<td>Nicholas</td>
</tr>
<tr>
<td>Stribling</td>
<td>James</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
</tr>
</tbody>
</table>
```

4 rows in set (0.04 sec)

```sql
SELECT * FROM region;
```

```
+-----------+---------------------+
<table>
<thead>
<tr>
<th>region_id</th>
<th>region_name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>All</td>
</tr>
<tr>
<td>2</td>
<td>Goulburn Valley</td>
</tr>
<tr>
<td>3</td>
<td>Rutherglen</td>
</tr>
<tr>
<td>4</td>
<td>Coonawarra</td>
</tr>
<tr>
<td>5</td>
<td>Upper Hunter Valley</td>
</tr>
<tr>
<td>6</td>
<td>Lower Hunter Valley</td>
</tr>
<tr>
<td>7</td>
<td>Barossa Valley</td>
</tr>
<tr>
<td>8</td>
<td>Riverland</td>
</tr>
<tr>
<td>9</td>
<td>Margaret River</td>
</tr>
<tr>
<td>10</td>
<td>Swan Valley</td>
</tr>
</tbody>
</table>
+-----------+---------------------+
```

10 rows in set (0.01 sec)
SELECT * FROM region WHERE region_id <= 3;
+-----------+-----------------+
| region_id | region_name     |
| 1         | All             |
| 2         | Goulburn Valley |
| 3         | Rutherglen      |
+-----------+-----------------+
3 rows in set (0.03 sec)

SELECT region_name FROM region WHERE region_id <= 3;
+-----------------+
| region_name     |
| All             |
| Goulburn Valley |
| Rutherglen      |
+-----------------+
3 rows in set (0.01 sec)

SELECT * FROM customer WHERE surname='Marzalla' AND firstname='Dimitria';

SELECT cust_id FROM customer WHERE (surname='Marzalla' AND firstname LIKE 'M%') OR birth_date='1980-07-14';
**LIMIT**

The **LIMIT** operator is used to control the size of the output. Row numbering begins at row zero.

// returns only the first five rows from the customer table

SELECT * FROM customer LIMIT 5;

// returns the 100th to 104th rows from the customer table

SELECT * FROM customer LIMIT 100,5;

// set the second parameter to -1 to get all rows after a particular row

SELECT * FROM customer LIMIT 600,-1;
while (list($id, $date, $name, $statement, $url) = mysql_fetch_row($webstudent)) {
    echo "ID: $id <br>");
    echo "DATE: $date <br>");
    echo "NAME: $name <br>");
    echo "STATEMENT: $statement <br>");
    echo "URL: $url <br>");
}
**Inserting/adding rows (PHP)**

```php
$insert_webstudent = "INSERT webstudent (date, name, statement, url)
    VALUES ('$date', '$name', '$statement', '$url')";
mysql_query ($insert_webstudent);
```

**Deleting rows (PHP)**

```php
$delete_webstudent = "DELETE FROM webstudent WHERE id = '$update_id'";
mysql_query ($delete_webstudent);
```
$update_webstudent = "UPDATE webstudent SET date = '$date',
    name = '$name', statement = '$statement', url = '$url'
WHERE id = '$update_id'";
mysql_query ($update_webstudent);
MySQL resources


Also see http://us3.php.net/manual/en/ref.mysql.php for MySQL functions in PHP.

O'Reilly offers a variety of books on this subject available online through the UCLA proxy server.