PHP 04

Databases

MySQL

PHPMyAdmin

PHP and MySQL

Buttons as Images

Based on Rasmus Lerdorf& Kevin Tatroe: Programming PHP. Sebastopol: O'Reilly, 2002; David Lane, Hugh E. Williams: Web Database Application with PHP and MySQL, 2nd Edition. Sebastopol: O'Reilly, 2004; http://php.net

W05/161B/Sauter

Databases

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, 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 Terminology

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.

(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.

Relational Database

Tables

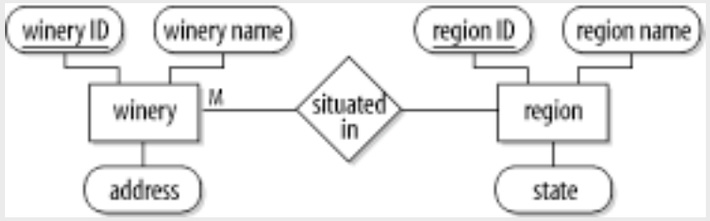
Winery Table

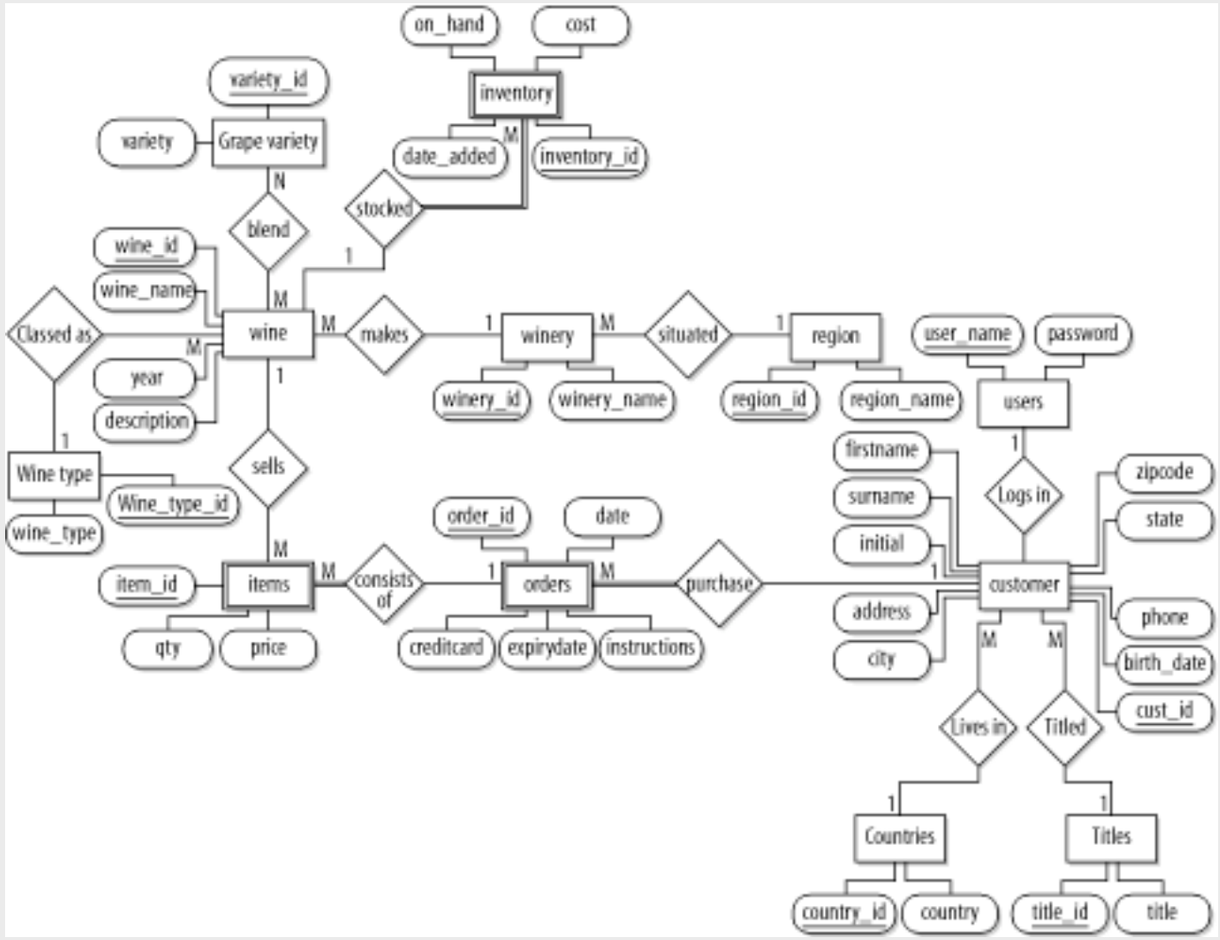
Winery ID	Winery name	Address	Region ID
1	Moss Brothers	Smith Rd.	3
2	Hardy Brothers	Jones St.	1
3	Penfolds	Arthurton Rd.	1
4	Lindemans	Smith Ave.	2
5	Orlando	Jones St.	1

Region Table

Region ID	Region name	State
1	Barossa Valley	South Australia
2	Yarra Valley	Victoria
3	Margaret River	Western Australia

ER Model (Entity Relationship)




on_hand
cost
variety_id
inventory
variety
Grape variety
date_added
inventory_id
N
M
stocked
blend
wine_id
wine_name
1
M
Classed as
wine
M
makes
1
winery
M
situated
1
region
user_name
password
M
year
1
winery_id
winery_name
region_id
region_name
users
description
1
sells
firstname
1
Wine type
Wine_type_id
surname
Logs in
zipcode
wine_type
order_id
date
initial
state
M
item_id
items
M
consists of
1
orders
M
purchase
1
customer
address
phone
qty
price
creditcard
expirydate
instructions
city
M
M
birth_date
cust_id
Lives in
Titled
1
1
Countries
Titles
country_id
country
title_id
title

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;
```

phpMyAdmin

web34db1 (18)
- calendar
- category
- connecter
- cursos
- ejercicios
- fotos
- images
- medialibrary
- media_library
- movies
- navigation
- personas
- profesores
- projects
- subscribers
- ta_alumnos_ejercicios
- ta_profesores_cursos
- upcoming

Server: localhost ▶ Database: web34db1 ▶ Table: images

Structure | Browse | SQL | Search | Insert | Export | Operations | Empty | Drop

	Field	Type	Attributes	Null	Default	Extra	Action
☐	id	int(4)		No		auto_increment	
☐	name	varchar(40)		Yes	*NULL*		
☐	path	varchar(30)		Yes	0		
☐	codec	tinytext		No			
☐	info	tinytext		Yes	*NULL*		
☐	txt	text		No			
☐	project_id	tinyint(4)		No	0		
☐	size	tinytext		No			
☐	thumbnail	tinyint(1)		No	0		
☐	image_order	smallint(2)		No	99		

Check All / Uncheck All *With selected:*

Print view | Relation view | Propose table structure

Add new field: 1 At End of Table At Beginning of Table After id Go

Indexes:

Keyname	Type	Cardinality	Action	Field
PRIMARY	PRIMARY	176		id
id	UNIQUE	176		id
id_2	INDEX	None		id
id_3	INDEX	None		id
name	FULLTEXT	None		name
info	FULLTEXT	None		info 1
fulltext_index_images	FULLTEXT	None		info 1

Create an index on 1 columns Go

Space usage:

Type	Usage
Data	15,568 Bytes
Index	48,128 Bytes
Total	63,696 Bytes

Row Statistic:

Statements	Value
Format	dynamic
Rows	176
Row length ø	88
Row size ø	362 Bytes
Next Autoindex	217
Creation	Jul 14, 2004 at 10:35 PM
Last update	Jan 17, 2005 at 06:43 AM
Last check	Jul 14, 2004 at 10:35 PM

Run SQL query/queries on database web34db1

```
SELECT * FROM `images` WHERE 1
```

Fields: id, name, path, codec, info, txt, project_id

<< Go

Show this query here again

Types

String Types

1. Char - Maximum 255 bytes
2. Varchar - Maximum 255 characters in a string
3. Tiny Blob, Tiny Text - Maximum 200,000,000 bytes
4. Bob, Text - Maximum 20,000,000,000,000,000 byes
5. Medium Blob, Medium Text - Maximum 2 x 10^24 bytes
6. Long Blob, Long Text - Maximum 2 x 10^32 bytes
7. Enum - Maximum 2 bytes with a Maximum 65,535 values total
8. Set - Maxmimum 8 bytes with a Maximum 64 values total

Numeric types

1. Tinyint - 1 byte
2. Smallint - 2 bytes
3. Mediumint - 3 bytes
4. Int - 4 bytes
5. Integer - 4 bytes
6. Bigint - 8 bytes
7. Float(x) - 4 if x^&24 or 8 if 25^=x^= 53
8. Float - 4 bytes
9. Double - 8 bytes
10. Double Precision - 8 bytes

11. Real - 8 bytes
12. Decimal(M,D) - M+2 bytes if D ⌣ 0, M+1 bytes if D = 0 (D+2, if M ⌒ D)
13. Numeric(M,D) - M+2 bytes if D ⌣ 0, M+1 bytes if D = 0 (D+2, if M ⌒ D)

And finally the date/time types:

1. Date - 3 bytes
2. DateTime - 8 bytes
3. TimeStamp - 4 bytes
4. Time - 3 bytes
5. Year - 1 bytes

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

```
// Server connect
$host = 'users.design.ucla.edu';
$usr = 'dsauter';
$pwd = 'myPassword';

$db = 'dsauter';

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.

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

or

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

SELECT Statements

```
SELECT surname, firstname FROM customer;

+-----------+-----------+
| surname   | firstname |
+-----------+-----------+
| Marzalla  | Dimitria  |
| LaTrobe   | Anthony   |
| Fong      | Nicholas  |
| Stribling | James     |
+-----------+-----------+
4 rows in set (0.04 sec)
```

```
SELECT * FROM region;

+-----------+--------------------+
| region_id | region_name        |
+-----------+--------------------+
|         1 | All                |
|         2 | Goulburn Valley    |
|         3 | Rutherglen         |
|         4 | Coonawarra         |
|         5 | Upper Hunter Valley |
|         6 | Lower Hunter Valley |
|         7 | Barossa Valley     |
|         8 | Riverland          |
|         9 | Margaret River     |
|        10 | Swan Valley        |
+-----------+--------------------+
10 rows in set (0.01 sec)
```

SELECT Statements

```
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

```
SELECT surname, firstname FROM customer WHERE city = 'Portsea' and
firstname = 'James' ORDER by surname;
```

```
SELECT * FROM customer WHERE city='Melbourne' ORDER BY surname DESC;
```

```
SELECT * FROM customer LIMIT 5;
```

```
SELECT * FROM customer LIMIT 100,5;
```

Reading out values

```
while (list($id, $date, $title, $text, $url) = mysql_fetch_row($news)) {
    echo "ID: $id <br>";
   echo"DATE: $date <br>";
   echo"DATE: $date <br>";
   echo"TITLE: $title <br>";
   echo"TEXT: $text <br>";
   echo"URLO: $url <br>";
}
```

Inserting/Creating Rows with PHP

```
$insert = "INSERT upcoming (date, title, text, url)
VALUES ('$date', '$title', '$text', '$url')";

mysql_query ($insert);
```

Deleting Rows

```
$delete_news = "DELETE FROM upcoming WHERE id = '$update_id'";
mysql_query ($delete_news);
```

Updating Rows

```
$update_news = "UPDATE upcoming SET date = '$date', title =
'$title', text = '$text', url = '$url' WHERE id =
'$update_id'";

mysql_query ($update_news);
```

MySQL Resource

Please refer to: http://www.mysql.com/

for a detaild MySQL Reference.

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