



# Google Maps with MySQL and PHP

# Disclaimer

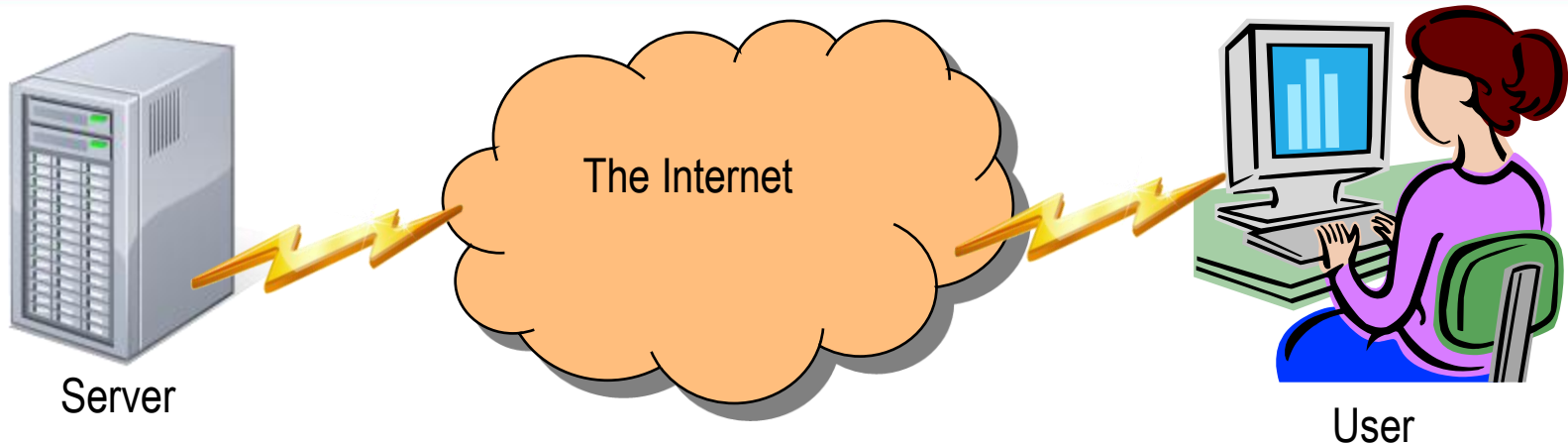
- Though SSC Pacific makes every effort to perform quality assurance on its training materials, the material in this presentation may inadvertently include technical inaccuracies or other errors. We would be grateful if users notify us of any errors or inaccuracies they may find.
- The presentation contains references to links and to third-party websites. These are provided for the convenience and interest of users and this implies neither responsibility for, nor approval of, information contained in these websites on the part of the U.S. Government. The USG makes no warranty, either express or implied, as to the accuracy, availability or content of information, text, graphics in the links/third party websites. The USG has not tested any software located at these sites and does not make any representation as to the quality, safety, reliability or suitability of such software, nor does this presentation serve to endorse the use of such sites.

# Overview

---

- Layout of system
- Installation of Software
- Configuration of MySQL
- Our First Map
- A Look Through the Code

# Layout of System



## L/WAMP Stack

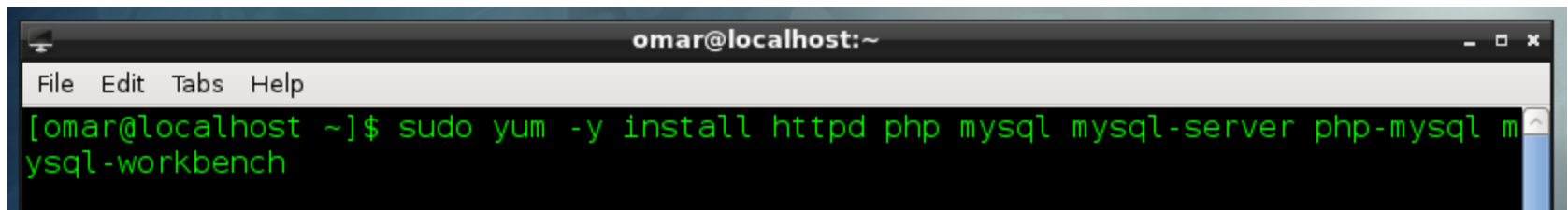
- Linux/Windows
  - Operating System
- Apache HTTP Server
  - Serves web-pages to the WWW
- MySQL Database
  - Used for storing the AIS data
- PHP Scripting Language
  - Used to query the database given criteria selected by the user

## •Internet Browser

- Used to view our Google Map with AIS data on it

# Installation of Software

- Enter the following command in a command shell:
- `sudo yum -y install httpd php mysql mysql-server php-mysql mysql-workbench`

A screenshot of a terminal window titled 'omar@localhost:~'. The window has a menu bar with 'File', 'Edit', 'Tabs', and 'Help'. The command prompt shows '[omar@localhost ~]\$' followed by the command 'sudo yum -y install httpd php mysql mysql-server php-mysql mysql-workbench' entered in green text. The command is split across two lines: 'mysql mysql-server php-mysql m' on the first line and 'ysql-workbench' on the second line.

```
omar@localhost:~  
File Edit Tabs Help  
[omar@localhost ~]$ sudo yum -y install httpd php mysql mysql-server php-mysql m  
ysql-workbench
```

# Installation of Software

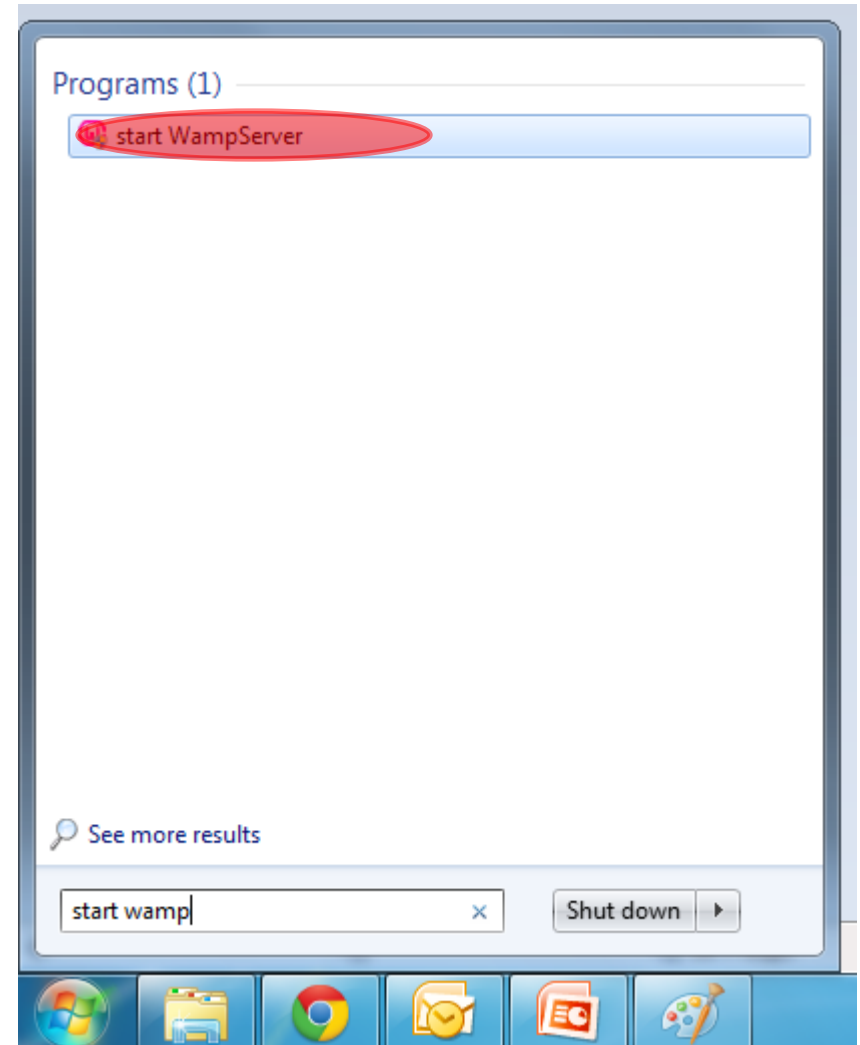
- Download and install WAMP stack (Windows, Apache, MySQL, PHP) wampserver2.2e-php5.3.13-httpd2.2.22-mysql5.5.24-x64
  - <http://www.wampserver.com/en/>
  - <http://www.wampserver.com/es/>
- Optionally install MySQL Workbench
  - <http://www.mysql.com/downloads/workbench/>

# Configuration of MySQL

- Start MySQL service by typing
- `sudo service mysqld start`
- Run mysql-workbench by entering
- `sudo mysql-workbench`

# Configuration of MySQL

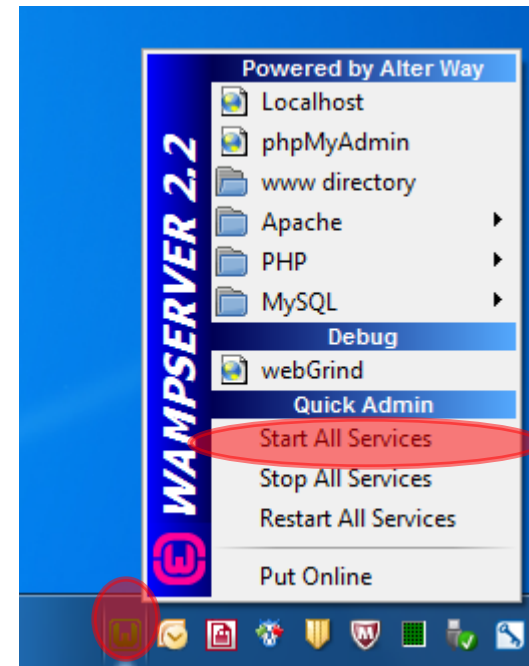
- Start Wamp Server





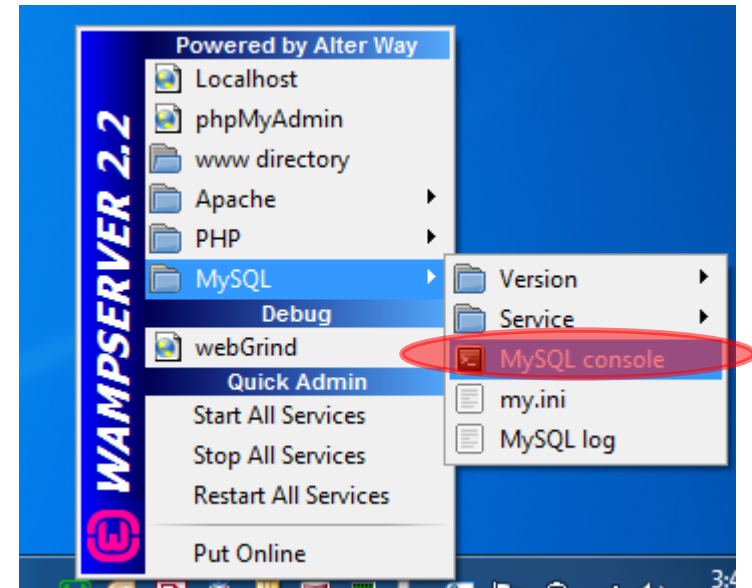
# Configuration of MySQL

- Start Apache, and MySQL services



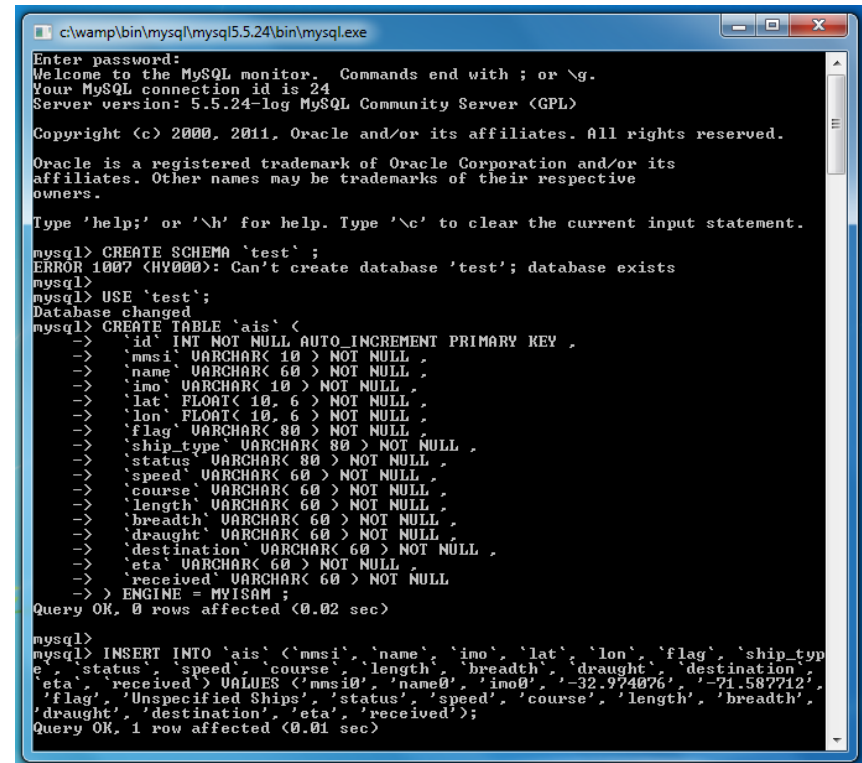
# Configuration of MySQL

- Run SQL script to create a database, create a table, and populate the table
- To do this we need to open the MySQL console.



# Configuration of MySQL

- When prompted for the password, press Enter
  - There is no password by default
- Copy and paste content of SetupDBForICODE.sql into the MySQL Console



```
c:\wamp\bin\mysql\mysql5.5.24\bin\mysql.exe
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 24
Server version: 5.5.24-log MySQL Community Server (GPL)

Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.

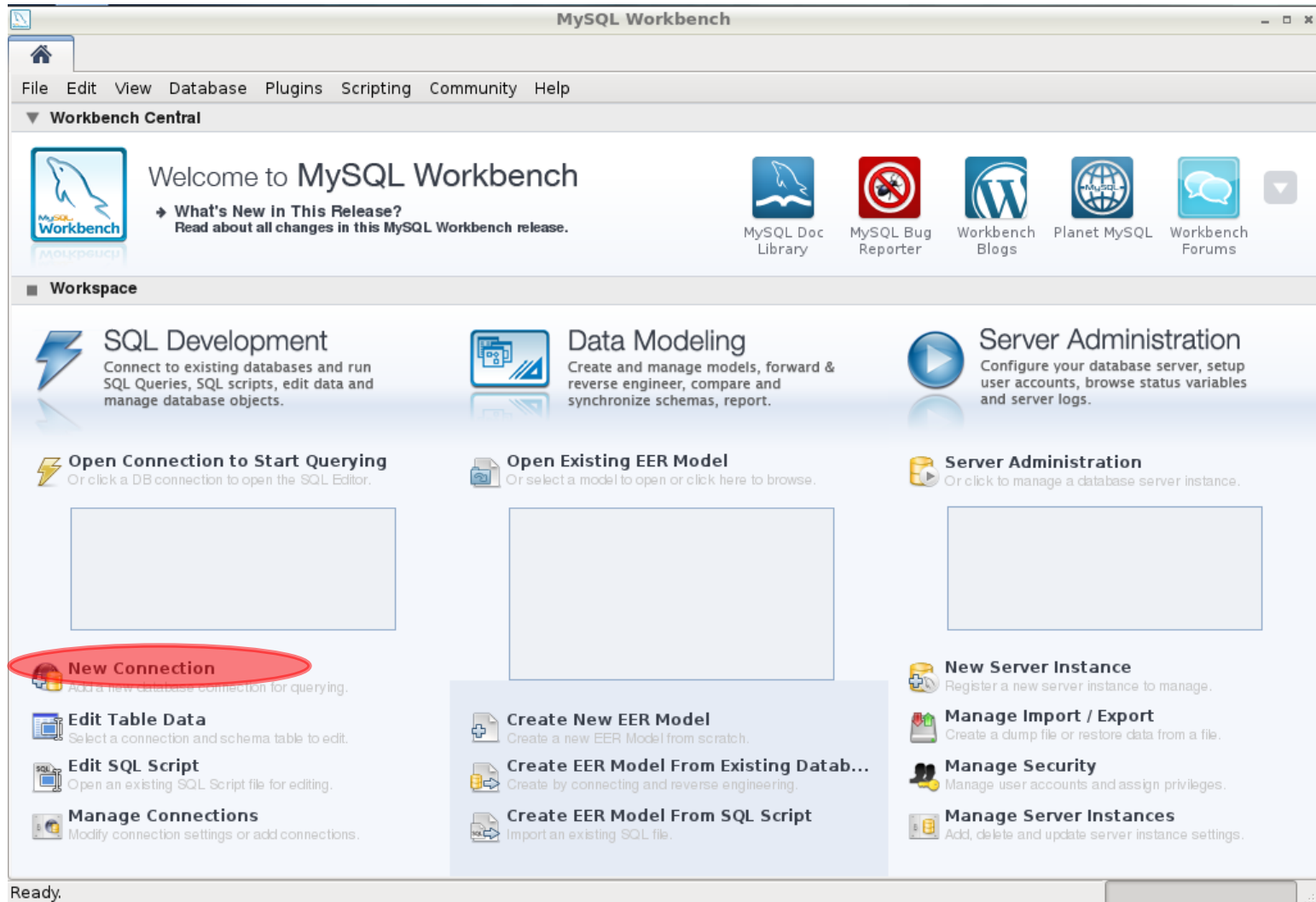
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

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

mysql> CREATE SCHEMA `test` ;
ERROR 1007 (HY000): Can't create database 'test'; database exists
mysql>
mysql> USE `test`;
Database changed
mysql> CREATE TABLE `ais` (
  ->   `id` INT NOT NULL AUTO INCREMENT PRIMARY KEY ,
  ->   `mmsi` VARCHAR(10) > NOT NULL ,
  ->   `name` VARCHAR(60) > NOT NULL ,
  ->   `imo` VARCHAR(10) > NOT NULL ,
  ->   `lat` FLOAT(10, 6) > NOT NULL ,
  ->   `lon` FLOAT(10, 6) > NOT NULL ,
  ->   `flag` VARCHAR(80) > NOT NULL ,
  ->   `ship_type` VARCHAR(80) > NOT NULL ,
  ->   `status` VARCHAR(80) > NOT NULL ,
  ->   `speed` VARCHAR(60) > NOT NULL ,
  ->   `course` VARCHAR(60) > NOT NULL ,
  ->   `length` VARCHAR(60) > NOT NULL ,
  ->   `breadth` VARCHAR(60) > NOT NULL ,
  ->   `draught` VARCHAR(60) > NOT NULL ,
  ->   `destination` VARCHAR(60) > NOT NULL ,
  ->   `eta` VARCHAR(60) > NOT NULL ,
  ->   `received` VARCHAR(60) > NOT NULL
  -> ) ENGINE = MYISAM ;
Query OK, 0 rows affected (0.02 sec)

mysql>
mysql> INSERT INTO `ais` (`mmsi`,`name`,`imo`,`lat`,`lon`,`flag`,`ship_type`,`status`,`speed`,`course`,`length`,`breadth`,`draught`,`destination`,`eta`,`received`) VALUES (`mmsi0`,`name0`,`imo0`,`-32.974076`,`-71.587712`,`flag`,`Unspecified Ships`,`status`,`speed`,`course`,`length`,`breadth`,`draught`,`destination`,`eta`,`received`);
Query OK, 1 row affected (0.01 sec)
```

# Configuration of MySQL



# Configuration of MySQL

The screenshot shows a 'Setup New Connection' dialog box with the following fields and options:

- Connection Name:** A text field containing 'AisDatabase', which is circled in red.
- Connection Method:** A dropdown menu set to 'Standard (TCP/IP)'.
- Parameters tab:** The 'Parameters' tab is selected, showing fields for Hostname, Port, Username, Password, and Default Schema.
- Hostname:** A text field containing '127.0.0.1'.
- Port:** A text field containing '3306'.
- Username:** A text field containing 'root'.
- Password:** A section with a 'Store in Keychain ...' button, a 'Clear' button, and a text field.
- Default Schema:** An empty text field.
- Buttons:** 'Test Connection', 'Cancel', and 'OK' buttons are at the bottom right. The 'OK' button is circled in red.

# Configuration of MySQL

The screenshot displays the MySQL Workbench application window. The title bar reads 'MySQL Workbench'. The menu bar includes 'File', 'Edit', 'View', 'Database', 'Plugins', 'Scripting', 'Community', and 'Help'. Below the menu bar is the 'Workbench Central' section, which features a 'Welcome to MySQL Workbench' message, a link to 'What's New in This Release?', and several utility icons: 'MySQL Doc Library', 'MySQL Bug Reporter', 'Workbench Blogs', 'Planet MySQL', and 'Workbench Forums'. The main area is the 'Workspace', which is divided into three columns. The left column, 'SQL Development', includes a description of connecting to databases and running queries, a button to 'Open Connection to Start Querying' (with a red circle highlighting the 'AisDatabase' connection), and links for 'New Connection', 'Edit Table Data', 'Edit SQL Script', and 'Manage Connections'. The middle column, 'Data Modeling', includes a description of creating and managing models, a button to 'Open Existing EER Model', and links for 'Create New EER Model', 'Create EER Model From Existing Datab...', and 'Create EER Model From SQL Script'. The right column, 'Server Administration', includes a description of configuring the database server, a button to 'Server Administration', and links for 'New Server Instance', 'Manage Import / Export', 'Manage Security', and 'Manage Server Instances'.

**MySQL Workbench**

File Edit View Database Plugins Scripting Community Help

▼ **Workbench Central**

Welcome to **MySQL Workbench**

→ **What's New in This Release?**  
Read about all changes in this MySQL Workbench release.

MySQL Doc Library MySQL Bug Reporter Workbench Blogs Planet MySQL Workbench Forums

■ **Workspace**

**SQL Development**  
Connect to existing databases and run SQL Queries, SQL scripts, edit data and manage database objects.

**Data Modeling**  
Create and manage models, forward & reverse engineer, compare and synchronize schemas, report.

**Server Administration**  
Configure your database server, setup user accounts, browse status variables and server logs.

**Open Connection to Start Querying**  
Or click a DB connection to open the SQL Editor.

**Open Existing EER Model**  
Or select a model to open or click here to browse.

**Server Administration**  
Or click to manage a database server instance.

**AisDatabase**  
User: root Host: 127.0.0.1:3306

**New Connection**  
Add a new database connection for querying.

**Edit Table Data**  
Select a connection and schema table to edit.

**Edit SQL Script**  
Open an existing SQL Script file for editing.

**Manage Connections**  
Modify connection settings or add connections.

**Create New EER Model**  
Create a new EER Model from scratch.

**Create EER Model From Existing Datab...**  
Create by connecting and reverse engineering.

**Create EER Model From SQL Script**  
Import an existing SQL file.

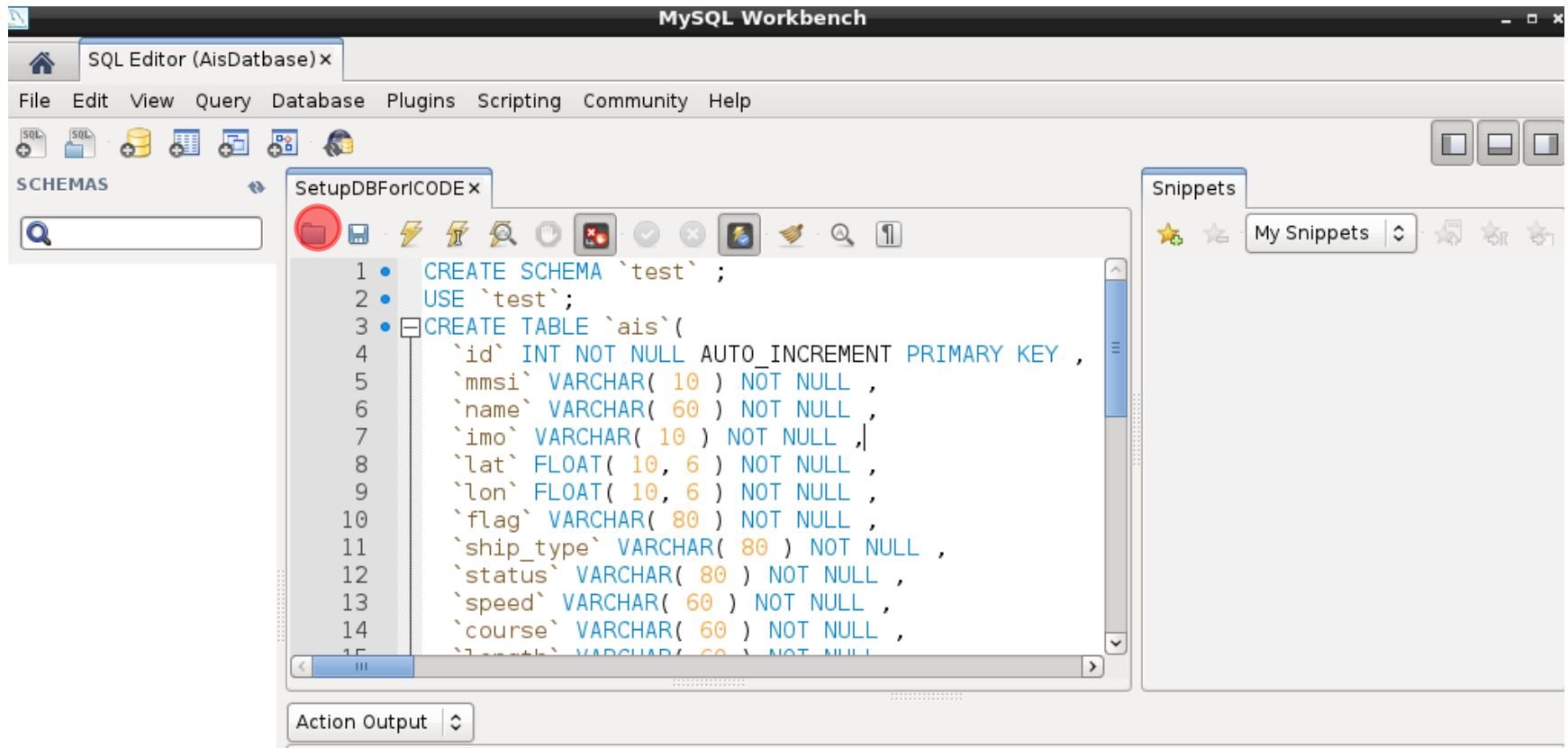
**New Server Instance**  
Register a new server instance to manage.

**Manage Import / Export**  
Create a dump file or restore data from a file.

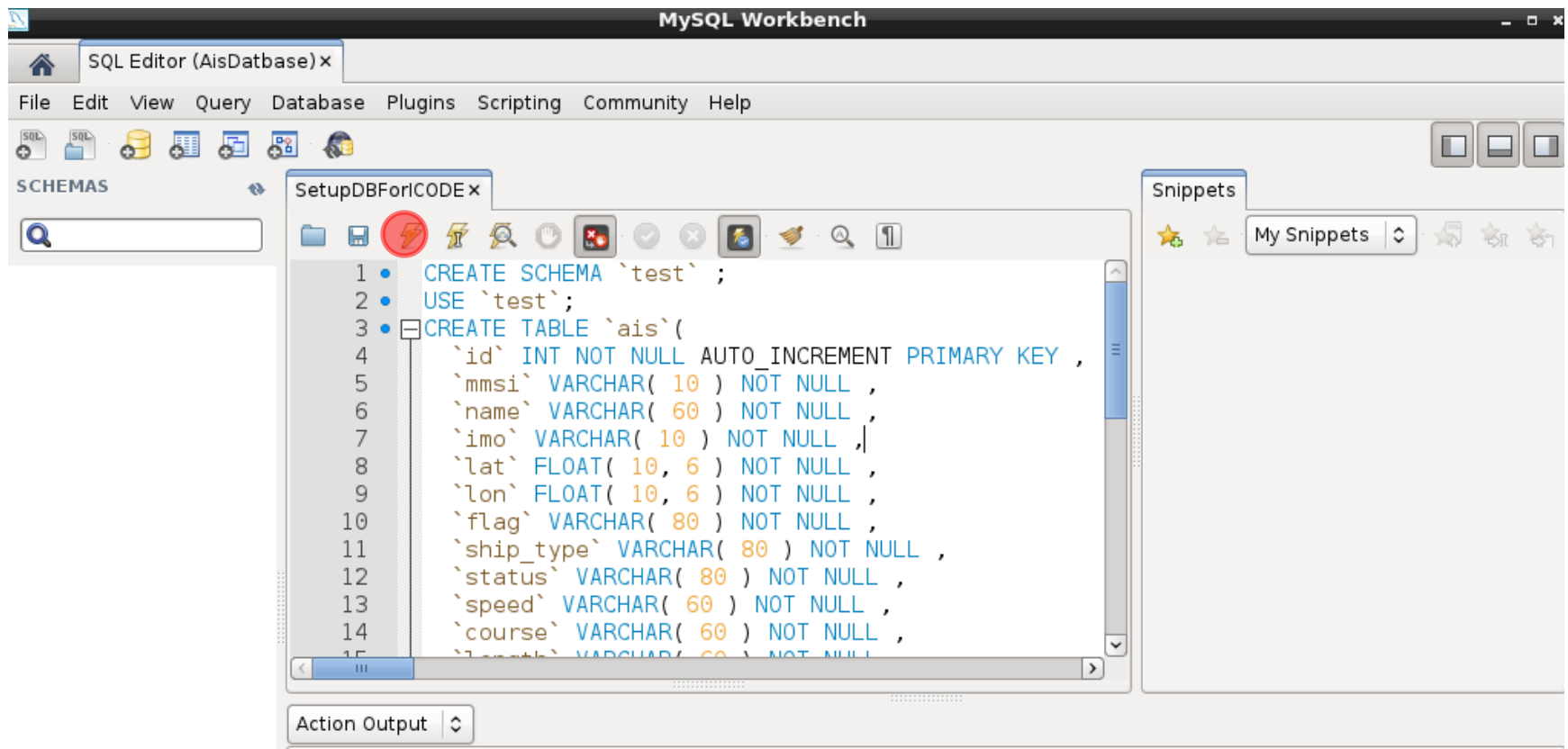
**Manage Security**  
Manage user accounts and assign privileges.

**Manage Server Instances**  
Add, delete and update server instance settings.

# Configuration of MySQL



# Configuration of MySQL





# Configuration of MySQL

- The MySQL database is now setup and has some sample data in it for us to display on our map

# Our First Map

- **Index.html**
  - Contains layout of page and tells the browser to use the code found in icode-maps.js, and also the Google Maps javascript
- **icode-maps.js**
  - Uses Google Maps API to display the map with AIS data displayed
- **icode\_example\_db\_query.php**
  - Contains code to query the MySQL database. This is called from within the icode-maps.js file

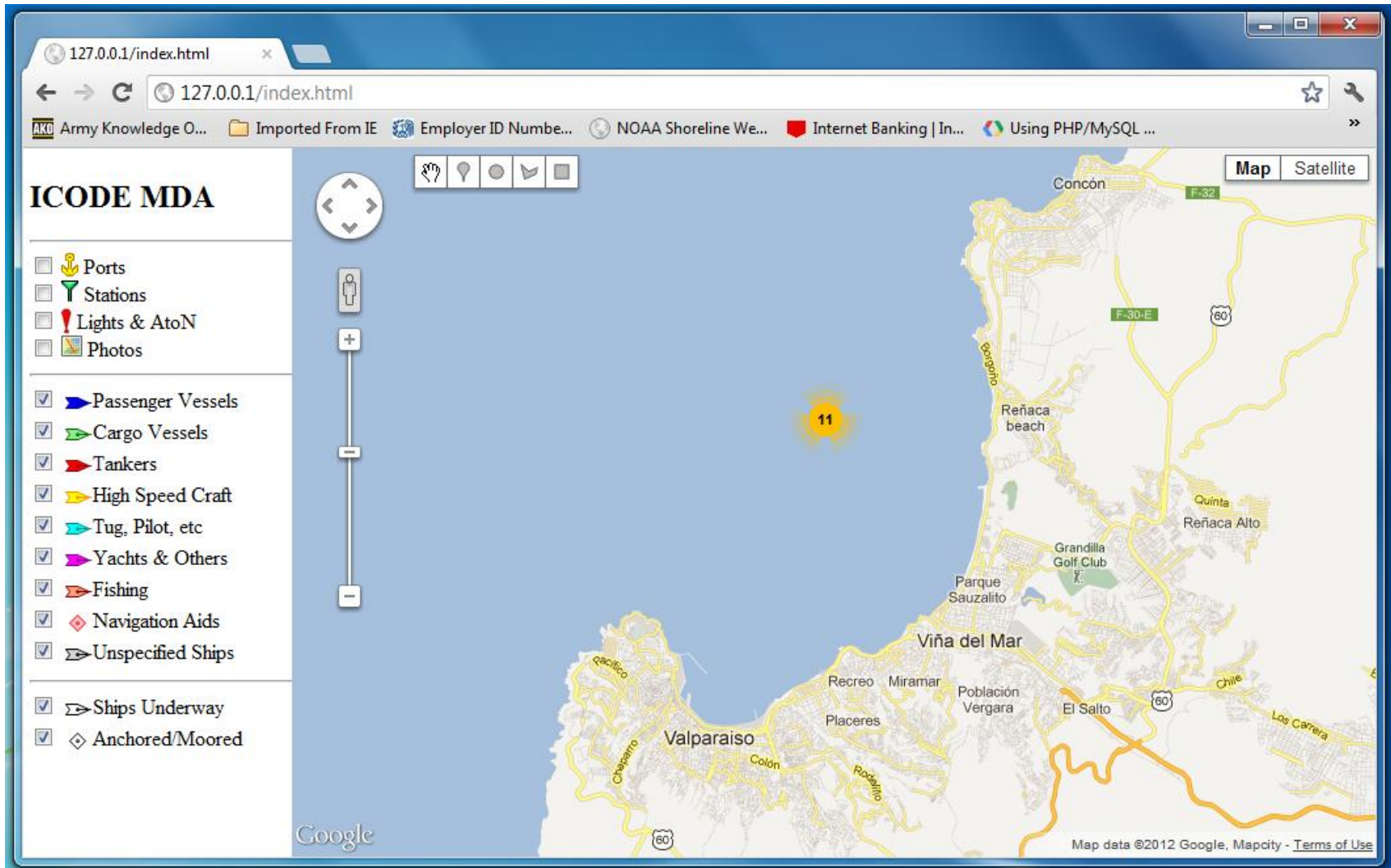
# Our First Map

```
omar@localhost:/home/omar/Desktop/GoogleMaps-ICODE
File Edit Tabs Help
[root@localhost GoogleMaps-ICODE]# ls
icode_example_db_query.php  index.html          SetupDBForICODE.sql
icode-maps.js               install_lamp.txt
[root@localhost GoogleMaps-ICODE]# cp * /var/www/html/
```

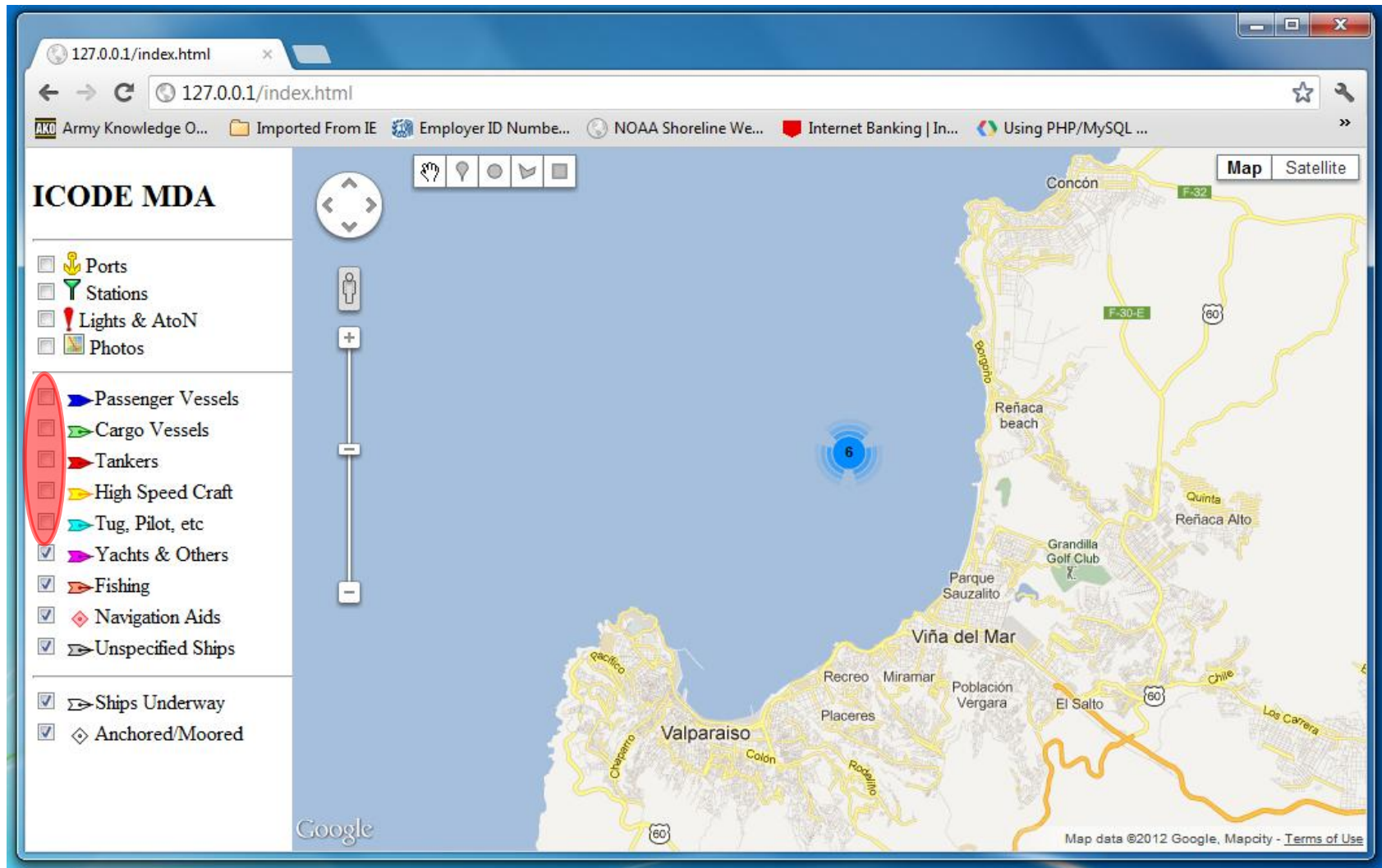
# Our First Map

- Copy the following files to C:\wamp\www
  - shipicons (folder of images)
  - icode\_example\_db\_query.php
  - icode-maps.js
  - index.html
  - markerclusterer.js

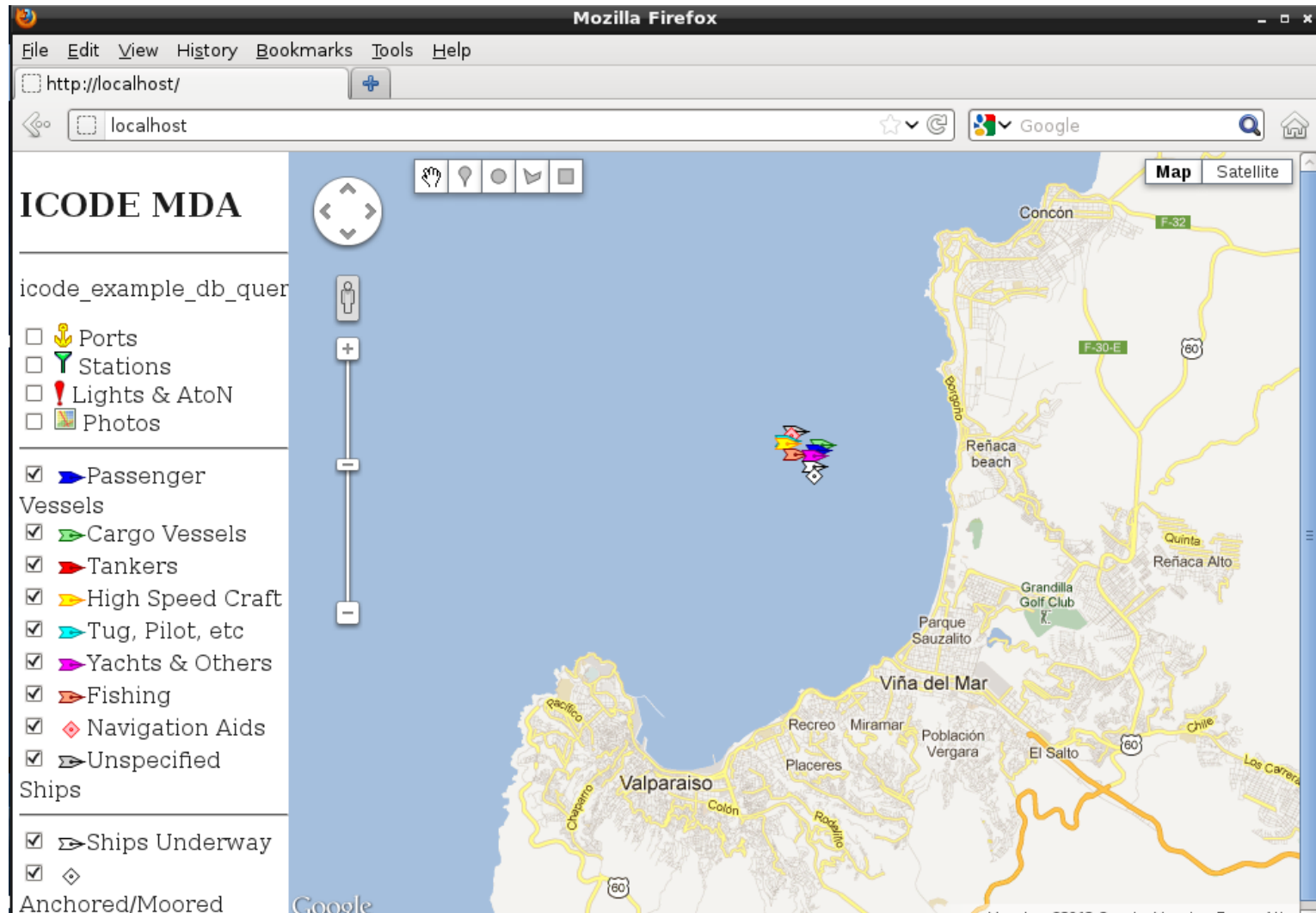
# Our First Map



# Our First Map

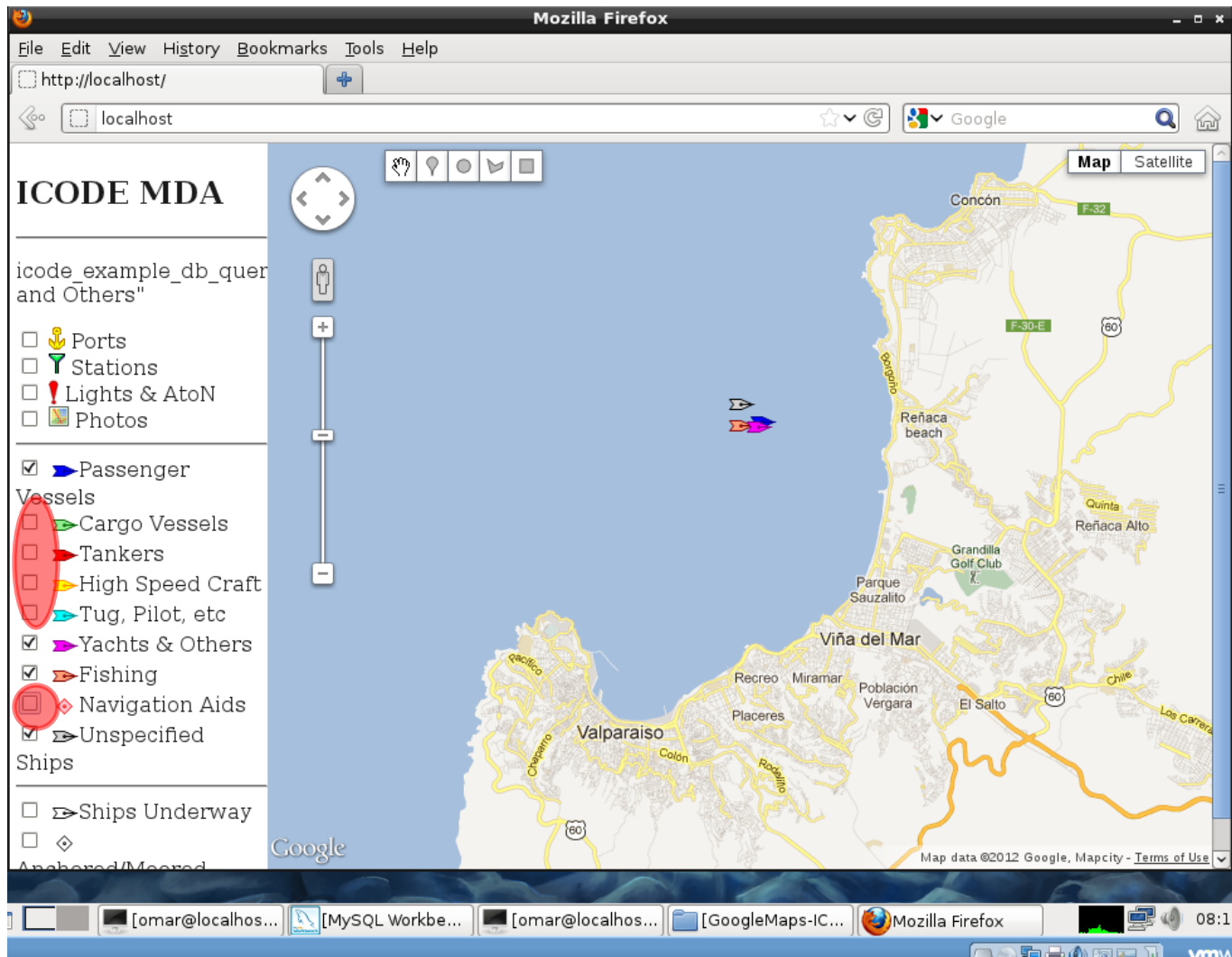


# Our First Map





# Our First Map





# Our First Map

The screenshot shows a web browser window at `http://localhost/` displaying a map application. The browser's address bar shows `localhost` and the search engine is set to Google. The application interface includes a sidebar on the left with the title "ICODE MDA" and a list of filters. The main area displays a map with a blue overlay representing water. A popup window is open over a vessel icon, displaying detailed information.

**ICODE MDA**

icode\_example\_db\_query and Others"

- ☐ Ports
- ☐ Stations
- ☐ Lights & AtoN
- ☐ Photos

---

☒ Passenger

Vessels

- ☐ Cargo Vessels
- ☐ Tankers
- ☐ High Speed Craft
- ☐ Tug, Pilot, etc
- ☒ Yachts & Others
- ☒ Fishing
- ☐ Navigation Aids
- ☒ Unspecified

Ships

- ☐ Ships Underway

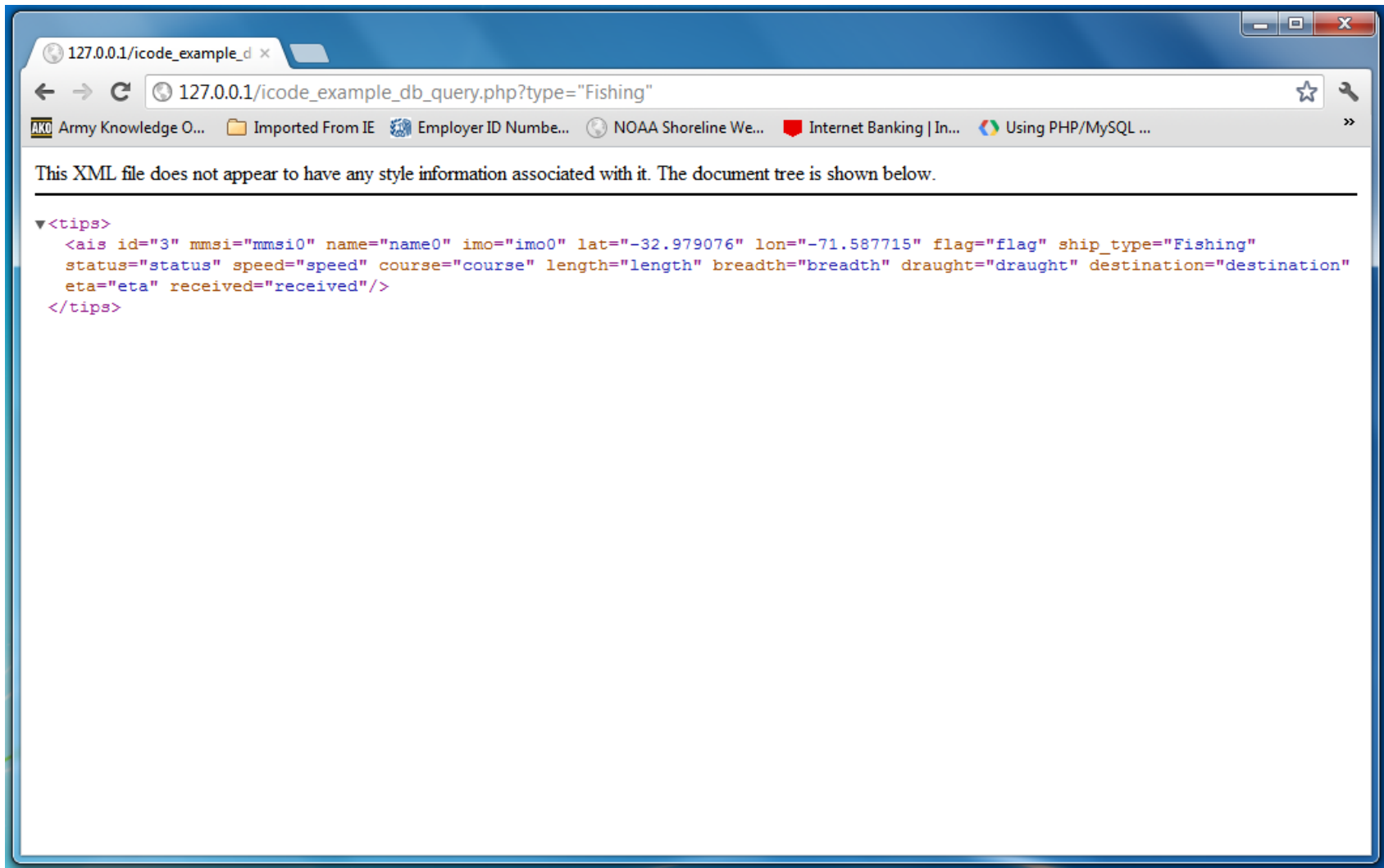
**name5**

Flag: flag  
 Ship Type: Yachts and Others  
 Status: status  
 Speed/Course: speed / course  
 Length x Breadth: length X breadth  
 Draught: draught  
 Destination: destination  
 ETA: eta  
 Received: received

# A Look Through the Code

- The PHP file creates an XML file.
- You can see this by browsing to:
  - [http://127.0.0.1/icode\\_example\\_db\\_query.php?type=%22Fishing%22](http://127.0.0.1/icode_example_db_query.php?type=%22Fishing%22)
- The JavaScript calls this file with an argument in the “type” field for each ship type that is checked on the left side of the map
- The XML that is returned of the ships currently in the database is parsed and the points are displayed on the map

# A Look Through the Code



The screenshot shows a web browser window with the address bar displaying `127.0.0.1/icode_example_db_query.php?type="Fishing"`. The browser's address bar also shows several bookmarks: "Army Knowledge O...", "Imported From IE", "Employer ID Numbe...", "NOAA Shoreline We...", "Internet Banking | In...", and "Using PHP/MySQL ...". The main content area of the browser displays a message: "This XML file does not appear to have any style information associated with it. The document tree is shown below." Below this message, the XML document tree is displayed, showing a root element `<tips>` containing a single `<ais>` element. The `<ais>` element has the following attributes: `id="3"`, `mmsi="mmsi0"`, `name="name0"`, `imo="imo0"`, `lat="-32.979076"`, `lon="-71.587715"`, `flag="flag"`, `ship_type="Fishing"`, `status="status"`, `speed="speed"`, `course="course"`, `length="length"`, `breadth="breadth"`, `draught="draught"`, `destination="destination"`, `eta="eta"`, and `received="received"`. The XML code is displayed in a monospaced font with syntax highlighting.

```
<tips>
  <ais id="3" mmsi="mmsi0" name="name0" imo="imo0" lat="-32.979076" lon="-71.587715" flag="flag" ship_type="Fishing"
    status="status" speed="speed" course="course" length="length" breadth="breadth" draught="draught" destination="destination"
    eta="eta" received="received"/>
</tips>
```

# A Look Through the Code

- The PHP file creates an XML file.
- You can see this by browsing to:
  - [http://localhost/icode\\_example\\_db\\_query.php?type=%22Unspecified%20Ships%22](http://localhost/icode_example_db_query.php?type=%22Unspecified%20Ships%22)
- The JavaScript calls this file with an argument in the “type” field for each ship type that is checked on the left side of the map
- The XML that is returned of the ships currently in the database is parsed and the points are displayed on the map

# A Look Through the Code

## HTML

- **Index.html**
  - Contains a header that points to the javascript files
  - Contains a body that defines how the page looks and the JavaScript function to run when it loads
    - There is a map area and an area for the buttons

# A Look Through the Code

## HTML

- `<script type="text/javascript" src="http://maps.googleapis.com/maps/api/js?libraries=geometry,drawing,weather&sensor=false"></script>`
- `<script type="text/javascript" src='icode-maps.js' ></script>`
- `<body onload="initialize()">`

# A Look Through the Code HTML

- `<div id="map_canvas" style="height: 100%; position: relative; background-color: rgb(229, 227, 223); overflow: hidden; "></div>`

# A Look Through the Code

## JavaScript

- Create a Basemap
  - `var map = new google.maps.Map(document.getElementById("map_canvas"), myOptions);`
- Get the data from the database
  - `var phpWithArg = "icode_example_db_query.php?type=\" + typesSelected[i] + "\";`
  - `downloadUrl(phpWithArg, function(data) {`



# A Look Through the Code PHP

- Get the ship\_types passed from the JavaScript call to this file
  - `$type = $_GET["type"];`
- Select all the rows with the correct ship\_type
  - `$query = "SELECT * FROM ais WHERE ship_type = $type";`