

## Database Systems Website Project

**Due Monday, December 1**

The purpose of this assignment is to give you practical experience in developing a website with a database backend. This will be done using the XAMPP package, which can be obtained at <https://www.apachefriends.org/download.html>. There are versions for Windows, Mac OS, and Linux. This package contains everything you need to do this project, except an IDE, if you choose to use one. The XAMPP package includes an implementation of the PHP language running on the Apache webserver and MariaDB, which is an open-source version of the MySQL database management system. As an IDE, my preference is IntelliJ PhpStorm, which can be obtained at <https://www.jetbrains.com/phpstorm/>. Students can get this free of charge. An important advantage of IntelliJ IDEs is that they provide support. Most other IDEs (Eclipse, Netbeans, Visual Studio) only have user communities, which are not always responsive or helpful, or charge money for support. Instructions on how to install and use XAMPP can be found at <https://www.simplilearn.com/tutorials/php-tutorial/php-using-xampp>.

The website will comprise a real estate listing service. In this assignment you are given a database schema, including a set of integrity constraints, and asked to implement and use this schema in the MySQL RDMS. Everyone is expected to use the same schema. This includes information about two kinds of properties, houses and businesses, and for each type some relevant information, e.g., for houses the price and numbers of bedrooms and bathrooms. There will also be information about real estate agents and agencies, and about buyers, including buyers' preferences. Once the database is created, then you are asked to create a website using PHP with an HTML interface that allows the user to query the database. A collection of queries is provided, e.g., searching for houses within a certain price range and having a specified number of bedrooms and bathrooms. Some queries will be generated automatically by the user's selecting items and check boxes on the HTML interface, and more complex queries can be entered manually in a text box. Details regarding this interface are provided in the following. In creating the database, it is necessary to enter enough data so that all the queries return results.

The current MySQL reference manual can be obtained at <https://dev.mysql.com/doc/refman/9.4/en/>. There are numerous sources of information regarding PHP, including especially <https://www.php.net/>. Tutorials can be found at <https://www.tutorialspoint.com/php/index.htm> and <https://www.w3schools.com/php/>. A brief introduction to using PHP and MySQL in XAMPP will be included in the lectures, and a simple example has been posted in the Modules section on Canvas, but you will be mostly on your own. Anyone that knows C++ should have no trouble picking up PHP.

It is permissible to work in groups of up to four people per group. Turn in one assignment for each group, showing all the peoples' names in the submitted documents.

### Problem Context

A real estate multiple listing service (MLS) keeps track of the houses and business properties for sale, the real estate agents that posts the listing to the database, and the potential buyers that are working with the agents. Houses and business properties are kinds of property. A listing entry shows information about the house or business property and the agent that posted the listing. Each agent works for a real estate firm. Buyers who are looking for houses or business properties work with a real estate agent who has access to the MLS. Following is an analysis of the information requirements and integrity constraints.

### Information Requirements

- 1) *Property*: Has an address, owner's name, and price.
- 2) *House*: Inherits the information required by a property and in addition has the number of bedrooms, number of bathrooms, and size in square feet. Make the House address a foreign key reference to a Property address.
- 3) *Business Property*: Inherits the information required by a property and in addition has the type of business, e.g., gas station, store front, office space, etc., and the size in square feet. Make the Business Property address a foreign key reference to a Property address.
- 4) *Agent*: Has an id, a name and phone number. An agent's employment record contains the date when his/her employment started.
- 5) *Firm*: Has an id, a name, and an address.
- 6) *Buyer*: Has an id, a name, a phone number, and a set of preferences consisting of the type of property in which the buyer is interested (house or business property) and, if a house, the desired number of bedrooms and bathrooms, and, if a business, the desired type of business property, and, for either a house or business property, the desired price range.
- 7) *Listing*: Has an MLS number, the property address, the agent's id, and the date the listing was posted. The MLS number serves as an id.

### Database Schema

- Property ( address: VARCHAR(50), ownerName: VARCHAR(30), price INTEGER )
- House ( bedrooms: INTEGER, bathrooms: INTEGER, size INTEGER, address: VARCHAR(50) )  
[Make the House address a foreign key reference to a Property address.]
- BusinessProperty ( type: CHAR(20), size: INTEGER, address: VARCHAR(50) )  
[Make the BusinessProperty address a foreign key reference to a Property address.]
- Firm ( id: INTEGER, name: VARCHAR(30), address VARCHAR(50) )
- Agent ( agentId: INTEGER, name VARCHAR(30), phone CHAR(12), firmId INTEGER, dateStarted DATE )
- Listings ( mlsNumber: INTEGER, address: VARCHAR(50), agentId: INTEGER, dateListed DATE )
- Buyer ( id: INTEGER, name VARCHAR(30), phone CHAR(12), propertyType: CHAR(20), bedrooms INTEGER, bathrooms INTEGER, businessPropertyType: CHAR(20), minimumPreferredPrice: INTEGER, maximumPreferredPrice: INTEGER )
- Works\_With( buyerId: INTEGER, agentID:INTEGER )

### Integrity Constraints

- 1) A property is identified by its address.
- 2) A property can be listed by at most one agent.
- 3) A listing is identified by its MLS number.
- 4) An agent works for one, and only one, firm.
- 5) Address in Listings is a foreign key reference to Address in Property.

### Data requirements

- 1) Insert sufficiently many records into your tables to return at least 1 record for each of the queries below.

- 2) In any case, insert at least 5 records into each table.
- 3) Indicate the primary keys, foreign keys, and any non-null fields appropriately.

**You are to turn in:**

- 1) An ER diagram that translates into this schema.
- 2) A MySQL script file that creates the tables that implement the given relation schemas and populates these tables with test data according to the instructions given above.
- 3) A MySQL script file containing SQL instructions that implement the queries described below.
- 4) A document showing the output resulting from each of the ten queries shown below. This means that must enter them manually into MySQL, i.e., not using the HTML interface, or run a script file that contains all the queries.
- 5) The source code for a PHP program that provides the user interface described in the following Website Interface Specifications.
- 6) Screen shots showing the results your program produces for the six actions described in the Website Interface Specifications.

**The queries are:**

- 1) Find the addresses of all houses currently listed.
- 2) Find the addresses and MLS numbers of all houses currently listed.
- 3) Find the addresses of all 3-bedroom, 2-bathroom houses currently listed.
- 4) Find the addresses and prices of all 3-bedroom, 2-bathroom houses with prices in the range \$100,000 to \$250,000, with the results shown in descending order of price.
- 5) Find the addresses and prices of all business properties that are advertised as office space in descending order of price.
- 6) Find all the ids, names and phones of all agents, together with the names of their firms and the dates when they started.
- 7) Find all the properties currently listed by agent with id "001" (or some other suitable id).
- 8) Find all Agent.name-Buyer.name pairs where the buyer works with the agent, sorted alphabetically by Agent.name.
- 9) For each agent, find the total number of buyers currently working with that agent, i.e., the output should be Agent.id-count pairs.
- 10) For some buyer that is interested in a house, where the buyer is identified by an id (e.g., "001"), find all houses that meet the buyer's preferences, with the results shown in descending order of price.

**Website Interface Specifications**

Your website should be able to do the following:

1. Display all listings and their associated property info. Break out separate lists for houses and business properties.
2. Search houses based on price range, number of bedrooms, and number of bathrooms.
3. Search business properties based on price range and range of size in square feet.
4. Display all agents and their associated info.
5. Display all buyers and their associated info, including preferences.

6. Provide a textbox where the user can manually enter any database query, and return the query results as a separate HTML page.

### **Some basic facts about MySQL**

When you install XAMPP, MySQL is automatically installed with default user “root” and a blank password. You can access this through the “Shell” in the XAMPP interface or via the Command Prompt or Powershell on Windows, or a terminal on Mac OS, or an xterm in Linux. To log into MySQL enter

```
mysql -u root -p
```

You will be prompted for the password. Since the password is blank, just press Enter. Here are some useful MySQL commands.

1. To see the databases in your account enter:

```
show databases;
```

2. To use a particular database enter:

```
use <database-name>;
```

3. To see all the tables in your database (after doing the above) enter:

```
show tables;
```

4. To see the schema for a particular table enter:

```
describe table-name;
```

5. A MySQL statement can span multiple lines, and it must end with “;”.

For example:

|                       |                                      |
|-----------------------|--------------------------------------|
| select * from //Enter | leads to a new line                  |
| property; //Enter     | causes the statement to be executed; |

6. Exit MySQL with:

```
exit;//Enter
```

### **How to create and execute a script file:**

- 1) Write your SQL commands into a text file with the extension .sql (the extension isn’t actually necessary, but it is customary to use this).
- 2) Put the (script).sql file in some directory.
- 3) Navigate to that directory, log into your MySQL database using the instructions given above, and then execute either “source (script).sql” or “./ (script).sql”.