

Database Systems MySQL Practice

Due Friday 4/20

The purpose of this assignment is to give you practical experience with a relational database management system, namely the well-known and popular MySQL. You are allowed to work in teams, with a maximum of four people per team.

Each student will be assigned a MySQL account on the departmental machine dbsrv where an empty database will have been created for you by the departmental systems group. If you prefer, however, you can go to www.mysql.com and download the current version of MySQL for use on your own computer. There are implementations for Windows and most varieties of Linux. If you go this route, then you will need to create your own database. This is very simple to do: once inside MySQL, just execute “create database <dbname>;”. Some instructions for logging into MySQL and performing various tasks appear at the end of this document. This may not cover everything you need, however, for which reason you might want to consult the MySQL 5.7 reference manual at <https://dev.mysql.com/doc/refman/5.7/en/>. This contains a tutorial, which you might find helpful.

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. Given that an empty database has been created, this entails creating the necessary tables, initializing them with some test data, and executing a variety of queries.

You are also asked to draw the associated ER model. This admittedly is doing things a bit in reverse, since the normal procedure is to create the ER model first, and then use this to produce the database schema. A problem with this is that there is more than one way to do the latter, and we want to be sure that all students work with the same schema.

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*: Has the information required by a property plus number of bedrooms, number of bathrooms, and size in square feet.
- 3) *Business Property*: Has information required by a property plus the type of business, e.g., gas station, store front, office space, etc., and the size in square feet.
- 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)
In addition, House inherits the attributes of Property, i.e., these need to be included in the House schema.
- BusinessProperty (type: CHAR(20), size: INTEGER)
In addition, BusinessProperty inherits the attributes of Property, i.e., these need to be included in the BusinessProperty schema.
- Firm (id: INTEGER, name: VARCHAR(30), address VARCHAR(50))
- Agent (agentId: INTEGER, name VARCHAR(30), phone CHAR(12), firmId INTEGER, dateStarted DATE)
- Listings (address: VARCHAR(50), agentId: INTEGER, mlsNumber: 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) No two listings can have the same MLS number.
- 4) An agent works for one, and only one, firm.

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 which 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.

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.

How to connect to the database

In class you will be given an accountName and password.

1. Use your computer science department account to connect to linprog.cs.fsu.edu or any other server in the Computer Science department.
2. Connect to the database server by entering the following:

```
mysql -h dbsrv2.cs.fsu.edu -u <accountName> -p <accountName>db
```

For example, if your accountName is cop4710t01, enter:

```
mysql -h dbsrv2.cs.fsu.edu -u cop4710t01 -p cop4710t01db
```

You will be prompted for a password. Enter the password you were given with your accountName in class. This will enter you into the database with name <accountName>db.

Some basic facts about MySQL

1. To see the databases in your account use:

show databases;
2. To use a particular database use:

use <database-name>;
3. To see all the tables in your database (after doing the above) use:

show tables;

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

`describe table-name;`

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

For example:

<code>select * from //return</code>	leads to a new line
<code>property; //return</code>	causes the statement to be executed;

6. Exit MySQL with `exit; //return`.

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 in your program account.
- 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"`.

Extra Credit Assignment

For those of you that would like to do some DB programming, you may develop either a stand-alone software package or a website that interacts with your database, using whatever programming language or web technology that you wish. This will be graded in a one-on-one demo with the instructor and/or one of the TAs. Your program or website should be able to do as much as possible of 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.