CSCI 585 – Database Systems – Spring 2012
Professor Farid Parvini
Homework Assignment 2
Due: Feb 29, 2012 @ 03:30 PM PDT

The previous assignment examined conceptual schema design in EER and the conversion of EER to a relational schema. In this assignment, you will continue on to the next step of realizing a practical database implementation of a given relational schema, and writing and running queries on it.

Preliminaries: Installing and Using MySQL Community Server

The current Generally Available release of MySQL is version **5.5.20**.
Download: http://www.mysql.com/downloads/mysql/5.5.html

Detailed installation instructions can be found in the reference manual, which also provides a tutorial, documentation of client and administrative programs, data types, SQL statement syntax, etc.
Reference Manual: http://dev.mysql.com/doc/refman/5.5/en/

You will be using the `mysql` command-line client in this assignment to execute SQL DDL and DML statements to create, populate and query your database. You will be required to submit these statements in respective `*.sql` files. Use of MySQL workbench and/or other GUI tools is *not recommended, not required, and will not be supported*.

Problem

Domain Specification

Consider a database used by a car rental firm to keep track of various aspects of their business. The system maintains records for each of their branches and employees; including information about which branch each employee works at, as well as the manager of each branch.

The car rental firm owns many different vehicles that it offers for rental, but groups them into a few vehicle classes (e.g. economy, standard, premium, etc.); the class of vehicle determines how much it would cost to rent a vehicle per day. Each vehicle is always based at a single branch (assume you cannot rent a car in one place and then return it somewhere else).

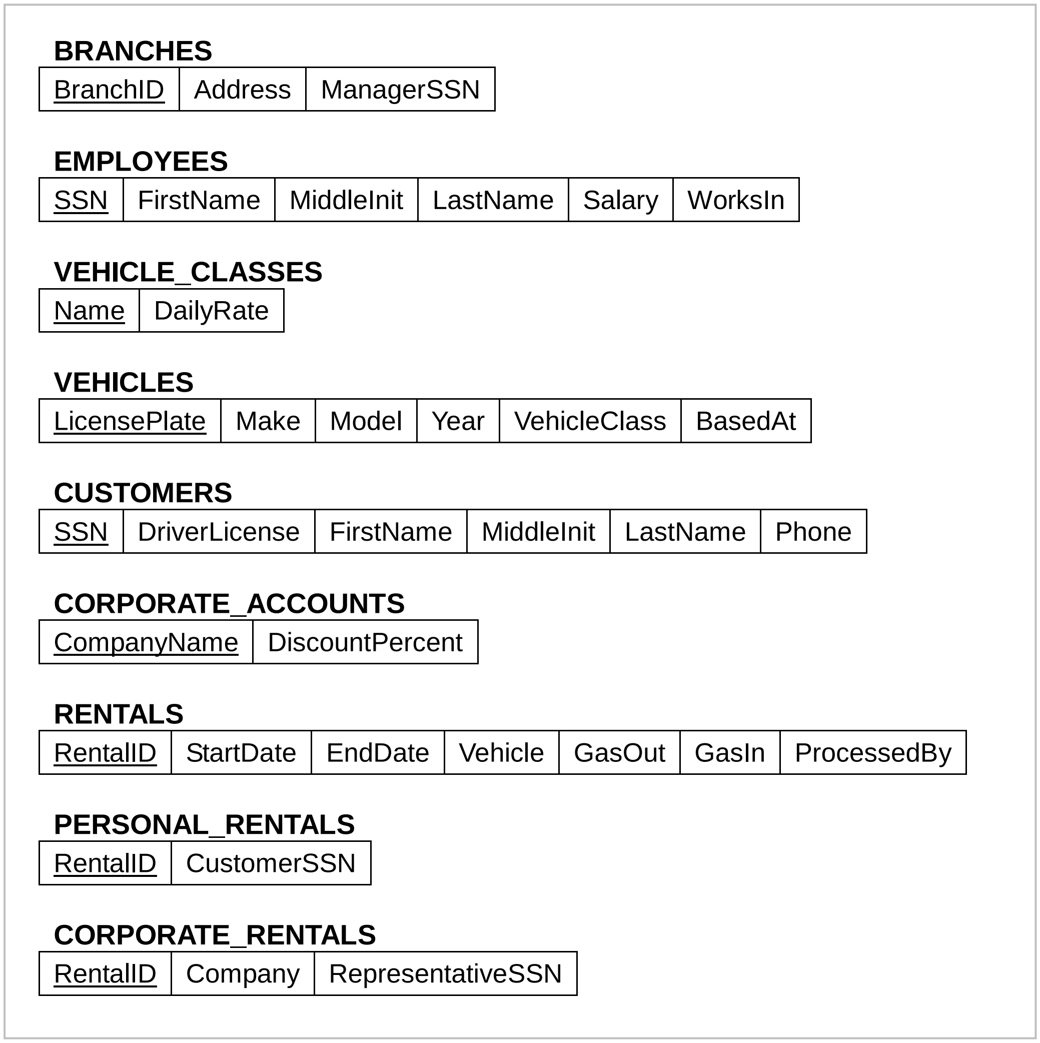
The system assigns a unique rental ID to keep track of each time one of the vehicles is rented out. It also notes the start and end dates of the rental, the amount of gas in the vehicle before it is loaned out and after it is returned, as well as the employee who processed the rental transaction.

When a vehicle is rented by an individual for personal use, the system simply keeps track of pertinent customer information. However, to increase business, the car rental firm also has contracts with various companies and organizations to provide discounted rates when renting vehicles for corporate use. The application thus needs to maintain a record of the agreed discount rate for each such company; records for corporate rentals also need to keep track of the renting company in addition to information about the person picking up the vehicle (the company representative).

Step 1: Creating the database (20 points)

The relational schema for the application is given in Fig. 1. Primary keys have been underlined. Foreign keys have been omitted, but you are expected to identify and implement them accordingly.

Prepare and submit a `createdb.sql` file containing SQL statements that will create a database `CarRental` containing the given tables. Primary keys and foreign keys must be implemented. Choose the most appropriate data type and size for each field.[1]



BRANCHES

BranchID	Address	ManagerSSN

EMPLOYEES

SSN	FirstName	MiddleInit	LastName	Salary	WorksIn

VEHICLE_CLASSES

Name	DailyRate

VEHICLES

LicensePlate	Make	Model	Year	VehicleClass	BasedAt

CUSTOMERS

SSN	DriverLicense	FirstName	MiddleInit	LastName	Phone

CORPORATE_ACCOUNTS

CompanyName	DiscountPercent

RENTALS

RentalID	StartDate	EndDate	Vehicle	GasOut	GasIn	ProcessedBy

PERSONAL_RENTALS

RentalID	CustomerSSN

CORPORATE_RENTALS

RentalID	Company	RepresentativeSSN

Fig. 1: Schema for `CarRental`

Step 2: Populating the database (9 points)

Prepare and submit a `populate.sql` file that populates the database with the data provided in the accompanying CSV files using `INSERT` statements. **Read the FAQ** for suggestions on how to do this efficiently. You **must not** use `LOAD_DATA_INFILE` or other direct import methods – you will be penalized if you do so.

[1] Hint: examine the data provided.

Step 3: Queries on the database (70 points)

Write the following queries in SQL and run them on your database. Prepare and submit them as separate files `q1.sql` to `q8.sql`. If two or more SQL statements are needed for a single question, they should be written after each other in one file. Depending on the data, your query might not return any results but that does not necessarily mean that your query is wrong.

Do not create or use views. Storing intermediate query results is not an appropriate use of views; you should be able to obtain the required results using nested queries.

Q1. Find all instances where a rented vehicle was returned with at least 3 additional gallons of gas or more in the tank than there originally was at the start of the rental. List the vehicle's license plate and start and end dates of each such rental instance.

Q2. Give the full name of each employee that earns more than the manager of the branch that they work in. The full name should be combined into a single field according to the format "John H. Smith" if there is a middle initial, or “Jane Doe” if not.

Q3. List the phone numbers of customers that have only made corporate rentals, and have never made any personal rentals.

Q4. List the three most popular vehicle make and model based on number of rentals.

Q5. List the IDs and addresses of all branches for which the total salary of all employees in the branch is less than $12000. List the branches with least total salary first.

Q6. List the full name and drivers’ license of all customers who have ever rented a Mitsubishi vehicle.

Q7. List all employees sorted by the number of rentals they processed in the month of Nov 2011 (according to rental start date). List the employees with most rentals processed first. Give the employee's full name, salary, and number of rentals processed.

Extra Credit

Q8. List all of the companies with corporate accounts along with the total cost over all corporate rentals by each company. Assume that rental costs include both start and end date, e.g. Feb 1 - Feb 3 is 3 day rental. When a rental is returned with less gas than at the start, a surcharge of $4 per gallon is imposed. No refund given for extra gas, however. The corporate discount percentage applies only to daily rental rate; there is no discount for the gas surcharge. The cost of each rental is rounded to the nearest cent after discounts are applied.

Step 4: Cleaning up (1 point)

Prepare and submit a `dropdb.sql` file that drops your entire database, including all types and tables that are created by `createdb.sql`.

FAQ

Q: Can I use a different version of MySQL or a different RDBMS?

A: For consistency, all submissions will be graded using MySQL 5.5.20 as the common platform. This means graders will take your submitted SQL statements and execute them in 5.5. We do not guarantee compatibility with other versions; your submission must work in 5.5.20 during grading.

Q: What are these *.CSV files, and how do I open them?

A: Please see: http://en.wikipedia.org/wiki/Comma-separated_values

Q: What is an `*.sql` file? How do I make one? How do I run it?

A: An `*.sql` file is a text file containing SQL commands. You can use any plain text editor of your choosing to create and edit `*.sql` files. To run the files using the `mysql` command line client, see http://dev.mysql.com/doc/refman/5.5/en/batch-commands.html. Note that this is the same procedure that will be used to run your submitted files according during grading.

Of course, you can also use the `mysql` command line client interactively and keep track of your SQL statements and queries separately in a text file, then reorganize your solutions into the `*.sql` files for final submission.

Q: Can I use file names that are different than the ones specified?

A: You **must not** alter the name, case, spacing or extension of any of the specified files. The grading processes uses these exact names and altering them will cause some files not to be graded.

Q: That is a lot of `INSERT` statements; isn't this going to take forever to finish typing up?

A: Here are a couple of suggestions for generating `populate.sql` without excessive labor:

- Write a script to parse the data files and output the required SQL statements.
- Open a copy of the data files in a spreadsheet, add the required SQL around the data, and copy out into a plain text file. You only need to type SQL for a couple of rows per file, and drag-to-copy for the rest of the table. Remember to check for exceptions!

You can then run `populate.sql` as described earlier instead of typing everything manually.

Q: I am having trouble creating foreign keys! / I cannot insert some data because of foreign key errors!

A: You must create the referenced table before you can create the foreign key that references it. There are no circular references in the schema so you should be able to create all the tables in some order that satisfies this. Also note that the column types must be exactly the same.

Likewise, if you are trying to insert some data that includes a reference to another table, you must have populated the other table first, otherwise you would be violating a referential integrity constraint by trying to reference something that does not exist.

Submission Instructions

- This assignment is to be submitted ***electronically*** on USC Blackboard.
  A how-to movie is available at:
  http://www.usc.edu/its/blackboard/support/bb9/assets/movies/bb91/submittingassignment/index.html
  - You should have up to 11 SQL files: `createdb.sql`, `populate.sql`, up to eight query files `q1.sql` to `q8.sql`, and `dropdb.sql`. Failure to adhere to filenames will incur a 5 point penalty. Make sure the file extensions are correct! (e.g. *not* `*.sql.txt`)
  - Also prepare a `readme.txt` containing your name, id and email address. You may also include comments and note any assumptions you made in the file (optional).
  - Compress your `readme.txt` and SQL files into a single zip archive and name it `lastname_firstname_hw2.zip`. Use only standard zip format. Do **not** use other formats such as zipx, rar, ace, etc. Improper filename or format will incur a 5 point penalty.
  - Make sure that you have attached the file the when submitting. Failure to do so will be treated as non-submission. Late policy applies until correctly submitted.
  - Successful submission will be indicated in the assignment's submission history. We advise that you verify the timestamp, download and double check your zip file for good measure.
  - It is ok to modify and re-send your file. We will automatically assume the submission with the latest timestamp as your final answer for grading. **Caution**: if you had submitted earlier but choose to amend your submission *after* the deadline has passed, it will be treated as a late submission, with penalties as specified below.
- This assignment is electronically by the due date and time shown above.
- There is a 20% deduction per day for late submissions. Assignments more than 5 days late will not be accepted.

Discussion Board and Student Collaboration and Academic Integrity Policy

A discussion board for this assignment has been set up on blackboard. Please use it as your main resource to post any questions or requests for clarification related to the assignment. The TAs will participate and answer questions on the board – **do not ask homework-specific questions by email**.

The discussion boards are moderated, so your posts will not show up until after they have been approved – please do not re-post the same message. Also, **start your homework early**. Although the discussion board will remain open until the assignment deadline, the TAs cannot guarantee that last minute questions posted less than 24 hours before the deadline will be answered.

You may discuss general strategies to be used on the assignment, but please refrain from revealing and/or hinting at any answers. In this course we encourage you to study together, but do remember that all work submitted for the class **is to be done individually** and within the realm of the USC Academic Integrity Guidelines. Violations of the Student Conduct Code will be filed with the Office of Student Conduct, and appropriate **sanctions will be given**. If you have questions about what is allowed, please discuss it with the instructor.