

# CECS 323 Term Project

## Spring 2019

### Project Description

You are designing a database for Miming's Chinese Cuisine restaurant, a small family-owned restaurant that specializes in authentic Chinese food for parties large and small.

### Miming's Money

We also encourage repeat customers. Each time that a customer places an order with us, we put Miming's Money put into an "account" with us that they can apply to their next order. For every \$10 of money that they spend (discounting the Miming's Money that they spend), they get a \$1 of Miming's Money deposited into their Miming's account. We only deal in whole numbers, so an order for \$137.42, for instance, would give that customer \$13 of Miming's Money.

### Customer Information

Customers can remain anonymous if they pay in cash. However, if they give us some basic information about them, we can award them Miming's Money to reward their loyalty. In addition, we will be sure to let them know of specials that we have from time to time. What we would like to know about each **private** customer is:

- Name
- E-mail address
- Snail-mail address

If they represent a **corporation** (for instance, they are buying for a corporate function) –

- The name of the corporation
- The name of the organization within the corporation (say Sales, Fabrication, Engineering, ...)
- The address of their office
- The contact information for someone that we can advise them of specials

### Staff

We treat our staff like family, because most of them are. Each employee at Miming's has a specific title and function:

- Chef – prepares the food
  - Head Chef
    - Designs new recipes and oversees the rest of the chefs. We keep track of which of our recipes were developed by a given head chef
    - We must have one and only one of our head chefs in the kitchen whenever Miming's is open.
  - Sous Chef

- Works with the head chef, but tends to be more “hands on”. There is always at least one sous chef on duty during any shift.
  - Each sous chef is expert in preparing a set of menu items. A sous chef who wishes to learn how to prepare a given menu item must enter into a mentoring relationship with one of the other sous chefs who is an expert in that menu item. When they do, Miming’s records the start date, the menu item, and the end date of the mentorship.
- Line Cook
  - Works in a particular department (butcher, fry cook, grill chef, pantry chef, pastry chef, roast chef, sauté chef (in charge of the sauces and gravies) and vegetable chef (this is the one who earns the highest celery).
  - During a given shift, more than one line cook could staff any one of these stations, and a given line cook might be in charge of more than one station. Typically, during a time of day that is usually slow, we will assign one line cook to several stations so that we get the best value for their time.
- All of our cooks are full time employees and work on salary.
- Maître d’ – welcomes guests, gets them seated, and evenly distributes the patrons across the wait staff who are on duty.
  - Miming’s keeps track of which booth/table(s) that the maître d’ assigns each party.
  - All of our Maître d’s have a fixed, hourly rate.
- The wait staff – takes the orders, brings the food and drink and helps the customers with any questions that they might have.
  - During a given shift, each wait staff member covers some collection of tables. That wait staff team member has those tables for their entire shift. No other wait staff member serves any of those tables during a given shift.
  - The wait staff must turn in their tips at the end of each meal. Miming’s then gives that money back to the wait staff member as part of their paycheck.
  - The wait staff also has a fixed hourly rate that they receive in addition to their tips.
- Dishwasher – fixed hourly rate.
- Manager – Full-time salaried
  - Each shift needs one and only one manager.

## **Scheduling of staff**

To keep things simple, the employees work shifts.

Each day has just two shifts: morning and evening. A given employee always works a full shift, no matter what their function is. An employee always works the same shift.

Each shift is assigned a specific work crew. Each work crew is composed of one head chef, one to three sous chefs, three to eight line cooks, one Maitre d', one dishwasher, one manager and two to five wait staff according to the number of customers anticipated for a given shift.

Many of our employees are part-time, so we pay them by the hour.

Others are on salary and receive a weekly set rate.

Only the salaried employees receive health care benefits. It's not that we do not care about our employees, it is just that we care a lot more about our customers.

## **Menu Items**

- Each menu item has a spiciness associated with it. The spiciness values are:
  - Mild
  - Tangy
  - Piquant
  - Hot
  - Oh My God
- Each menu item falls into one or more of several categories<sup>1</sup>:
  - Appetizer
  - Soup
  - Meat Entrees
    - Chef Special
    - Pork
    - Chicken
    - Beef
    - Seafood
    - Vegetables
  - Chow Mein
  - Egg Foo Young
  - Chop Suey
- Chow Mein, Egg Foo Young and Chop Suey all come in each of the types of meat listed above under "Meat Entrees".

## **Pricing**

Miming's has four menus:

---

<sup>1</sup> borrowed from <https://www.goldenphoenixinla.com/> for these.

- Evening
- Lunch
- Sunday brunch buffet
- Children's

A given menu item can appear on any of the three menus, but it must appear on no less than one of them. The same menu item showing up on the **evening** menu will cost the customer more money than the same menu item on the **lunch** menu. We reduce the portion size for the lunch menu, so we can offer them for less. A menu item can only come in one size for one price on a given menu. No options as to size.

The All You Can Eat Sensational Suicide Sunday Brunch has no prices for the individual items at all, since each customer pays a fixed rate to partake of our superlatively sumptuous buffet.

As you would guess, the portions on the Children's menu are even smaller than they are on the Lunch menu. None of our really spicy items go into the children's menu.

Miming's prices some of our menu items, such as our soups, by volume as well as more traditional serving options. Those menu items will have a price for a cup, bowl, pint, quart, and gallon sized serving.

## **Orders**

An order can come in via the web, the phone, or "eat in". Miming's records the type of payment used on each order.

Customers must pay with a credit card for their order when they place the order on the web.

Customers pay for their phone order when they pick it up. The phone order and eat in orders can be paid for by cash, debit card, or credit card.

Miming's does not take checks. Evil customers have taken advantage of Miming's with bad checks too often in the past to allow customers to use them as a form of payment.

For the "to go" orders, we give the customer a time when they can come and pick it up. We also record exactly when the order was ready so that we can be sure to have the food hot and ready, no matter how much they order.

## **Additional Business Rules**

You will add five business rules to the above business rules. You will have to provide some means to enforce these business rules in your database, either by means of a database constraint such as referential integrity, uniqueness constraint, not null constraint or the like. The business rule needs to be something that will show up in the model of your design.

None of your additional business rules can contradict any of the business rules provided in the project definition.

## **Denormalization**

Denormalization is a conscious, deliberate change of a design from 3<sup>rd</sup> normal form to some lower normal form in order to meet some particular objective. You will describe the denormalization used in part one of the project. If you did any denormalization in your design, please include a separate paragraph(s) stating what you did and why. If you did not do any denormalization, state that and why.

Please be sure that your UML model depicts a 3<sup>rd</sup> Normal Form design. The changes to your design to reflect the denormalization that you have selected will appear in the relation scheme diagram.

For the purposes of this project, a denormalization must introduce redundancy of some sort into the physical structure of the dataset. For instance, merging a child table and its parent together and creating a subkey in the resulting table, or creating a multi-valued attribute while maintaining a junction table to represent those values as well would also introduce redundancy into the structure.

I will also consider any deliberate breaking of the design rules to enforce the model. For example, you may need to model a subclass even though it doesn't have any attributes of its own but it will enforce a constraint if you do.

## **Output**

Output of the database must support the following features. You do not need to develop "pretty" printed or on-screen reports. You will run the views/queries in MySQL Workbench in the lab.

## **Views**

1. `Menuitem_v` – For each menu item, give it's spiciness, and all of the different costs for that item. If a given item is not on a particular menu, then report "N/A" for that particular item for that particular menu. Also, if an item only appears as a single serving portion, put in "N/A" into the report for the gallon, ... prices.
2. `Customer_addresses_v` – for each customer, indicate whether they are an individual or a corporate account, and display all of the information that we are managing for that customer.
3. `Sous_mentor_v` – reports all of the mentor/mentee relationships at Miming's, sorted by the name of the mentor, then the name of the mentee. Show the skill that the mentorship passes, as well as the start date.
4. `Customer_Sales_v` – On a year by year basis, show how much each customer has spent at Miming's.

5. Customer\_Value\_v – List each customer and the total \$ amount of their orders for the past year, in order of the value of customer orders, from highest to the lowest.

Please perform a select \* from each view and put the results of that select into a file called view\_output.docx, or .pdf. Put the SQL for the view in the same file with the output.

## Queries

In all cases, remember:

- Each query is a **single** SQL statement.
- **Never** return just the ID of a given thing in your queries, always do any necessary joins so that you can display a proper name.
- I will dock points for using literals in your queries. For instance, use the now() function to get the current date when asked to find visits within the past year, do not use a literal and put in the due date of the assignment for the current date.
- Be sure that the sample data that you insert into your tables is adequate to return **some** data from each of these queries:
  1. List the customers. For each customer, indicate which category he or she fall into, and his or her contact information. If you have more than one independent categorization of customers, please indicate which category the customer falls into for all of the categorizations.
  2. For each order, list the total cost to the customer for that order. Sometimes, a group of customers will come in and split the order up into multiple checks. Treat that as though you had two separate parties at the same table.
  3. List the top three customers in terms of their net spending for the past two years, and the total that they have spent in that period.
  4. Find all of the sous chefs who have three or more menu items that they can prepare. For each sous chef, list their name, the number of menu items that they can prepare, and each of the menu items. You can use group\_concat to get all of a given sous chef's data on one row, or print out one row per mechanic per skill.
  5. Find all of the sous chefs who have three or more menu items in common.
    - i. Please give the name of each of the two sous chefs sharing three or more menu items.
    - ii. Please make sure that any given pair of sous chefs only shows up once.
  6. Find the three menu items most often ordered from the Children's menu and order them from most frequently ordered to least frequently ordered.
  7. List all of the menu items, the shift in which the menu item was ordered, and the sous chef on duty at the time, when the sous chef was not an expert in that menu item.

8. List the customers, sorted by the amount of Miming's Money that they have, from largest to smallest.
9. List the customers and the total that they have spent at Miming's ever, in descending order by the amount that they have spent.
10. Report on the customers at Miming's by the number of times that they come in by month, and order the report from most frequent to the least frequent.
11. List the three customers who have spent the most at Miming's over the past year. Order by the amount that they spent, from largest to smallest.
12. List the five menu items that have generated the most revenue for Miming's over the past year.
13. Find the sous chef who is mentoring the most other sous chef. List the menu items that the sous chef is passing along to the other sous chefs.
14. Find the three menu items that have the fewest sous chefs skilled in those menu items.
15. List all of the customers who eat at Miming's on their own as well as ordering for their corporation.
16. List the contents and prices of each of the menus.
17. List each of the different shifts, who works during that shift and what their job is for that shift.
18. Three additional queries that demonstrate the five additional business rules. Feel free to create additional views to support these queries if you so desire.

### **Deliverables**

Your work will be done in three parts with one submission of each part for each team. See the class schedule for due dates.

Parts one and two will be turned in on paper. Part three will be submitted through dropbox.

The first part, design (conceptual), will consist of:

- Your five additional business rules
- Class diagram.
- English description of all classes.
- English description of all associations.
- If you did any denormalization in your design, please include a separate paragraph(s) stating what you did and why. If you did not do any denormalization, state that and why.

The second part (logical), will include:

- A revised design, based on feedback from the first part. This includes class diagrams and English descriptions of classes and associations.

- The relation scheme, based on the design.

The third part(physical), will include:

- A revised design, based on feedback from the second part. This includes class diagrams and English descriptions of classes and associations. .
- The relation scheme, based on the design and feedback from the second part.
- English description of all attributes.
- DDL used to create all the tables and the DML used to insert the data.
- Sample Output from the Queries – **Be sure to include sample output** in with your queries. The sample output could just be text, or screen captures. Call this file query\_output.docx or .pdf. Please include the SQL for the query immediately before the query output. This makes it immensely easier for me to review the query output

## Teams

This project will be done in teams of four people, whom you may self-select.

## Lessons Learned by Your Predecessors

- Unless you plan your time carefully, and stay on top of the deliverables, you **will** run out of time.
- Make sure that your relation scheme diagram is neat and easy to read early on. This will make everything much easier throughout the project.
- The data manipulation language (both the inserts and the selects) will take an immense amount of time if you are not careful. To keep that under control:
  - Read the queries in this rubric first, and plan your UML model around how you will answer the queries.
  - Use the MySQL workbench to interactively populate the data, and then export that data as insert statements.
  - Split the sample data up between at least two of the team members. This takes a long time to do because of all of the referential integrity constraints.
- Get the DDL done as soon as you are reasonably sure that the UML is solid. Start this at least two weeks ahead of the due date.
- Have someone in the team who has not done a given task, such as the queries, do a QA check before you move forward.
- Make use of a tool such as Google Drive to manage all of the files for the project.
- Use a group-messaging tool to keep in touch with each other.
- Start using the MySQL environment as soon it is issued so that you get familiar with it.
- Within the team, be very clear who has what responsibility. Do not change things without going through the person who is responsible for that part of the project.
- Write out your create table statements in the proper order so that the referential integrity constraints are a part of the create table statement or immediately afterwards.



- When you get feedback from me (your friendly professor) be sure to share it with the whole team.
- Get feedback early. Face to face is always best, but I take E-mail, as you doubtless know by now.
- Make a “to do” list of the project deliverables, and track % completion on all of them from the very beginning.
- Read this rubric several times. There are no “throw away” statements here!

Above all, have fun!

### **Configuration Management**

In the business world, there is nothing more deadly to a project than sloppy or spotty configuration management. I will leave it up to the team how they manage the configuration of the various deliverables for this project, but please be certain that the team as a whole is 100% certain that what is going into the project drop box is what the team wants to turn in. I am not going to be sympathetic to the team that tells me the day after a given phase is due that the wrong version of something got turned in by mistake, or that one of the team members turned something in before the rest of the team was entirely ready.

### **Technology**

Complete this project in MySQL. Nothing else will meet the requirements. Each of you will get an individual MySQL account. Each team gets a group MySQL account as well. All of the team members of a given team will be able to login using their individual account, and then jointly access the team database within MySQL. For that reason, it's important that everyone on the team make sure that you can access the campus MySQL server both through BeachNet+ as well as from home so that you don't all have to be here on campus to work together on this. You will use your individual account for development work. Once you are confident that your work is ready to share, you will run those scripts in the group database so that the rest of the team can benefit from your work. You will **not** be able to grant access to any objects in your personal database to the rest of your team.