Fall 2017 Comp 533
Assignment 3
Due: October 29, 2017 at 11:45 PM

The goal of this assignment is to actually construct the database we designed in Assignment 2. As a result, you will become an expert on our application and our business.

What to turn in

You must turn in a .sql file on Canvas.

Grading

Building and loading the tables is worth 10 points, total.

Each query is worth 10 points. Points will be assigned for each query based, approximately, on the following guidelines:

- 0 points: Query not attempted, query does not give any results, or it does not compile
- 5 points: Query compiles, runs and is most of the way towards a correct answer
- 8 points: The query and answer it produces are almost correct, but there is a slight or subtle bug in the query
- 10 points: The query is correct and gives the right answer

Each short answer is worth 2.5 points.

The survey at the end is worth 5 points.

What's In and Out of Scope

This is intended to be a SQL query assignment. Therefore, you must write queries in SQL (not stored procedures or functions). You may use VIEWs as needed and you may use standard built-in MySQL functions (e.g. ROUND, IF or CASE statements). If you're not sure if something is allowed, ask!

Academic Honesty

The following level of collaboration is allowed on this assignment:

You may discuss the assignment with your classmates at a high level. Any issues getting MySQL running is totally fine. What is not allowed is direct examination of anyone else's SQL code (on a computer, email, whiteboard, etc.) or allowing anyone else to see your SQL code.

You may use the search engine of your choice to look up the syntax for SQL commands, but may not use it to find answers to queries.

You MAY post and discuss query results with your classmates.

1 Create Tables

Create the following tables for the food truck. Provide your SQL code. Be sure to specify `NULL` or `NOT NULL`, any appropriate default values, primary and foreign keys, and appropriate constraints.

1.1 Product

Attribute Name	Attribute Type
productCode	alphanumeric
productName	alphanumeric

1.1.1 Constraints

1. Product codes must be unique.
2. Product names must be unique.
3. Product code is the primary key.

1.2 ComponentCategory

Attribute Name	Attribute Type
compCatId	AUTO_INCREMENT integer
compCatName	alphanumeric

1.2.1 Constraints

1. componentCategory ids must be unique.
2. componentCategory names must be unique.
3. componentCategory id is the primary key.
4. componentCategory name may NOT be undefined.

1.3 Component

Attribute Name	Attribute Type
compId	AUTO_INCREMENT integer
compName	alphanumeric
compCatId	reference to componentCategory/compCatId

1.3.1 Constraints

1. Component ids must be unique.
2. ~~Component names must be unique.~~
3. compId is the primary key.

4. ~~Component category may be undefined.~~

1.4 Unit

Attribute Name	Attribute Type
unitId	AUTO_INCREMENT integer
unitName	alphanumeric

1.4.1 Constraints

1. unitId is the primary key.
2. Unit names and ids must be unique.

1.5 RecipeItem

Attribute Name	Attribute Type
recipeItemId	AUTO_INCREMENT integer
productCode	reference to product/productCode
compCatId	reference to componentCategory/compCatId
qty	real number
unitId	reference to unit/unitId
compId	reference to component/compId

1.5.1 Constraints

1. recipeItemId is the primary key.
2. Each combination of productCode, compCatId, and compId must be unique.

1.6 Location

Attribute Name	Attribute Type
locationId	AUTO_INCREMENT integer

Attribute Name	Attribute Type
locationName	alphanumeric
locationAddress	alphanumeric
locationCity	alphanumeric
locationState	alphanumeric
locationZip	alphanumeric

1.6.1 Constraints

1. locationId is the primary key.

1.7 Menu

Attribute Name	Attribute Type
menuId	integer
menuName	alphanumeric

1.7.1 Constraints

1. menuId is the primary key.
2. MenuName must be unique.

1.8 MenuItem

Attribute Name	Attribute Type
menuItemId	AUTO_INCREMENT integer
menuId	reference to menuId
productCode	reference to product/productCode
price	money

1.8.1 Constraints

1. menuItemId id is the primary key.

2. A productCode may only be included up to one time on a menu.

1.9 Event

Attribute Name	Attribute Type
eventId	AUTO_INCREMENT integer
eventName	alphanumeric
eventStart	datetime
eventEnd	datetime
locationId	reference to location/locationId
menuId	reference to menu/menuId

1.9.1 Constraints

1. eventId is the primary key.

1.10 Employee

Attribute Name	Attribute Type
employeeId	AUTO_INCREMENT integer
firstName	alphanumeric
lastName	alphanumeric
DOB	date
phone	alphanumeric

We are only going to store one phone number for each employee (change from Assignment 2)

1.10.1 Constraints

1. employeeId is the primary key.

1.11 EventAssignment

Attribute Name	Attribute Type
eventAssignmentId	AUTO_INCREMENT integer
eventId	reference to event/eventId
employeeId	reference to employee/employeeId

1.11.1 Constraints

1. eventAssignmentId is the primary key.

Minor change from the ERD assignment - we will only record one phone number for each employee.

1.12 Ticket

Attribute Name	Attribute Type
ticketId	AUTO_INCREMENT integer
eventId	reference to event/eventId
ticketTime	datetime
soldBy	reference to employee/employeeId
numProducts	number of products on the ticket

1.12.1 Constraints

1. ticketId is the primary key.

1.13 ProductSold

Attribute Name	Attribute Type
productSoldId	AUTO_INCREMENT integer
productCode	reference to product/productCode
ticketId	reference to ticket/ticketId

1.13.1 Constraints

1. productSoldId is the primary key.

1.14 ItemSold

Attribute Name	Attribute Type
itemSoldId	AUTO_INCREMENT integer
productSoldId	alphanumeric
compId	reference to component/compId
qty	real number
unitId	reference to unit/unitId

1.14.1 Constraints

1. itemSoldId is the primary key.

2 Load Data

Load the data needed for the assignment. The files are located on the class Canvas site:
https://canvas.rice.edu/courses/3600/files/folder/HW3
and are also listed out in the assignment description. Download the files and then load the contents into your tables.

If you accidentally load the data more than once, or run into some other problem, you can rerun these steps after dropping the tables and recreating and loading them. If you have to repeat the process, either DROP or TRUNCATE the table(s) and restart. If you truncate the table, use the follow command to reset the auto number sequence:

`ALTER TABLE` tablename `AUTO_INCREMENT` = 1;

3 Task 3 - Queries

Answer all of the questions below by writing and executing SQL queries. The queries must contain ONLY the answer to the question (no extra rows or columns). You many only use SQL to answer the questions.

Some of these queries are complex. You may use VIEWS in your implementation.

Note: The recipes for the database have a basis in reality. However, the tickets generated and the products and items sold per ticket were synthetically generated. As a result, some of the query results based on tickets, ticket prices and products sold may not match your vision of reality. For example, the most popular ice cream topping sold may be surprising.

3.0.1 Query 1

How many products are there? Return an integer count of unique product codes in a column named 'numProducts'.

3.0.2 Query 2

What is the recipe for a 'turtle sundae'? Return the component category name, the component name, the quantity, and the unit name. Order by recipeItemId.

3.0.3 Query 3

Show the number of cones (of any size, including waffle cones) sold at each event that took place at Valhalla. List each event and the total number of cones sold. Order by the event start date/time.

3.0.4 Query 4

Compute the cost of each ticket.

Add an attribute named 'totalPrice" to the ticket table and populate it. The attribute should be able to hold US currency values.

To check your results, return the ticketId, numProducts and totalPrice for tickets 170 and 1089. Sort by ticketId.

3.0.5 Query 5a

We want to maximize sales and work the most valuable events. To do that, we need to know which events generate the most money per hour.

List the top 10 events by sales per hour. List the eventId, eventName, locationName and price per hour, rounded to the nearest penny. List in order from highest price per hour to lowest.

To determine the number of minutes between 2 timestamps, you may use: TIME_TO_SEC(TIMEDIFF(endTime,startTime))/60

3.0.6 Query 5b

Now restrict the query to the colleges. Which are the top 2 colleges for sales per hour? Show the eventId, locationName and price per hour, rounded to the nearest penny. Sort by price per hour.

3.0.7 Query 6a

In order to stock inventory, it's important to know the most popular items.

Which 3 toppings are ordered most frequently on ice cream cones, dishes and waffle cones (including extra toppings)? You may use product codes in your query.

List the component name and quantity. Sort by quantity, descending.

3.0.8 Query 6b

Similarly, we want to stop stocking items that do not get sold. Which components are not used?

List the component id, name and category. Sory by component id.

3.0.9 Query 7a

The food truck's budget is pretty small. Consequently, there is no money available to hire a web developer to create a product builder front-end. Instead, we will write SQL code to add elements to the database.

A long standing Rice Undergraduate tradition is 'Beer Debates." This year there is a new twist: serving beer floats.

Write the SQL code to add a new event 'Beer Debate", held at 'Willy's Pub". Willy's Pub's address is 'RMC Basement". The Beer debate will be held on November 9, 2017, from 7 - 11 PM. The event will have a new, restricted menu, named 'Beer event", with menuId 5. The only products that will be sold are Beer (product code 'be'), existing drinks, and beer floats (product code 'bd' for beer debate). Beer will cost $5 and a beer float $7. Regular drinks will cost $1.

Beer and beer floats fall into the category 'alcoholic beverage".

A float consists of 12 ounces of beer, 5 ounces of ice cream and is sold with a tall napkin and a long spoon. It is served in a 20 ounce cup.

When adding the recipe items, you may only use product and component names and categories, not ids for the newly added items. You may use compId, compCatId, unitId, etc. for existing items.

3.0.10 Query 7b

Write a query to show the menu items on the new menu. For each item, list the menu name, productCode, productName and price. Sort by product code.

3.0.11 Query 7c

Run your query from #2 to show the recipe for a beer float.

3.0.12 Query 8a

In hindsight, 'Beer event' is too general of a menu name. Change the menu name to 'Beer Debate Menu'. Access the existing record by the menu name, not the id. Show your SQL statement that executes this change.

3.0.13 Query 8b

Rerun your query from 7b, but showing menu items for menus named 'Beer event' and 'Beer Debate Menu'.

3.1 Short answer questions (2.5 points each)

3.1.1 Short answer 1

What alternatives did you consider for string lengths and data types in your tables? How did you decide which to use?

3.1.2 Short answer 2

What different numeric types did you use? Why did you choose those?

3.2 Survey (5 points)

It took me approximately N hours to complete this assignment, where N is:

My favorite flavor of ice cream is: