Shawarma Food Truck Company

*Final Project*

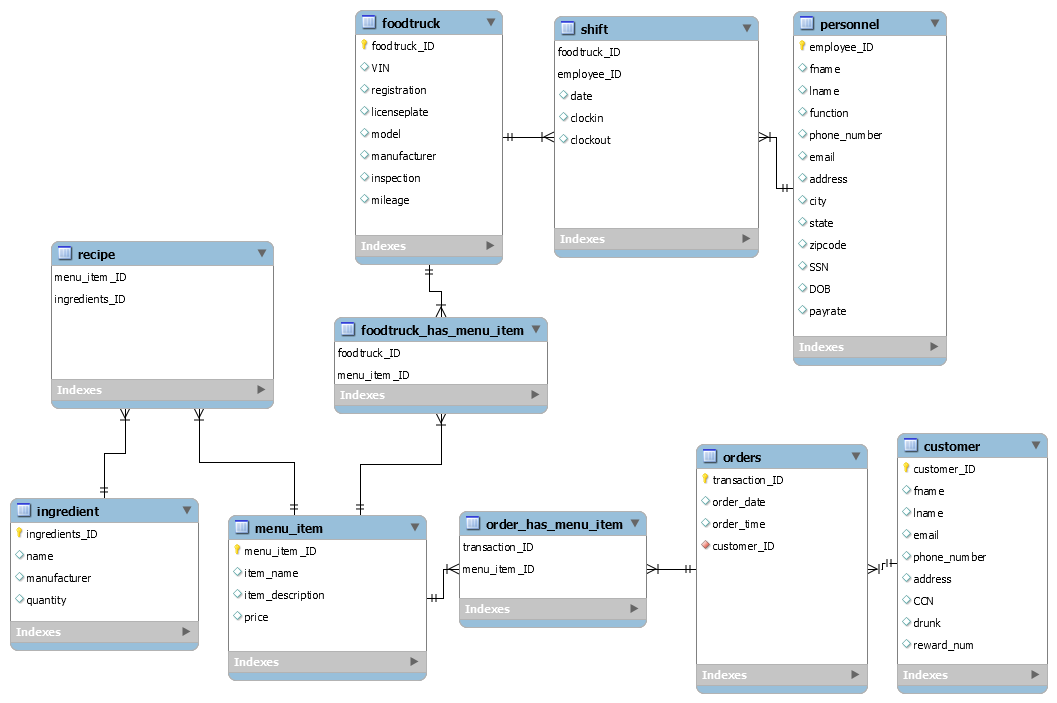
Group 1: Hull Collins, Killian Green, Duncan Mackay, John Schmitt

Professor Larson

BUS 315-02

20 November 2020

**I. Data Model**



### In our creation of the shawarma food truck database, we had to be very strategic in how we created our data models as well as tables. Our data model begins with “foodtruck” and has a many-to-many relationship with shift as well as foodtruck\_has\_menu\_item via an associative table. We thought that it made the most sense to connect foodtruck and personnel to shift. We want to be able to connect with foodtruck and personnel to shift because an employee’s shift is dependent on both the employee and the food truck. We created a many-to-many relationship between menu\_item and foodtruck. These two relationships function together––as a food truck serves the menu items––which is why we decided to connect these tables. Furthermore, we created a many-to-many relationship between menu\_item, ingredient and recipe, which connects the ingredients within our inventory and which ingredients are required in each menu item. We also created a many-to-many relationship with menu\_item and orders because an order can have multiple menu items. We felt that this design choice made the most sense because so as to link the menu\_item\_id with transaction\_id. Moreover there is a one-to-many relationship between customer and orders, as one customer can have many orders (or no orders at all). The connection of customer\_ID and transaction\_ID helps the food truck know which customers are ordering which items. This will enable us, in turn, to be able to measure the menu\_item\_id, ingredients, and recipe that are being sold most, which aids us in ordering food for inventory.

**II. Data Dictionary**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| customer |  |  |  |  |  |
| *Element or value display name* | *Description* | *Data Type* | *Character Length* | *Acceptable Values* | *Null Value* |
| fname | Customer provided first name | varchar | 45 | N/A | n |
| lname | Customer provided last name | varchar | 45 | N/A | n |
| email | Customer contact information, desired email address | varchar | 45 | N/A | n |
| phone\_number | Customer contact information, desired phone number | varchar | 45 | 0.0-99999 | n |
| address | Customer contact information, current residence of address (for mailing purposes) | varchar | 45 | N/A | n |
| CCN | Customer payment information, credit card number | varchar | 45 | 0.0-99999 | n |
| drunk | State of inebriation (for fun) | enum | N/A | ('Y','N') | n |
| reward\_num | Unique reward number for each customer; not every customer served will sign up for rewards and receive a reward number | varchar | 45 | 0.0-99999 | n |
| \*customer\_ID | Primary key; customer\_ID source database for each customer | int | 11 | 0.0-40 | n |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ingredient |  |  |  |  |  |
| *Element or value display name* | *Description* | *Data Type* | *Character Length* | *Acceptable Values* | *Null Values* |
| \*ingredients\_ID | Primary key; unique ingredients\_ID for each ingredient | int | 11 | 0.0-53 | n |
| name | Common name for each ingredient (i.e. ‘red onion’) | varchar | 45 | N/A | n |
| manufacturer | Food provider for respective ingredient | varchar | 45 | N/A | n |
| quantity | Number of each ingredient currently held within the inventory | varchar | 45 | 0.0-500 | n |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| personnel |  |  |  |  |  |
| *Element or value display name* | *Description* | *Data Type* | *Character Length* | *Acceptable Values* | *Null Values* |
| \*employee\_ID | Primary key; unique employee\_ID for all personnel in database | int | 11 | 0.0-53 | n |
| fname | Legal first name of employee | varchar | 45 | n/a | n |
| lname | Legal last name of employees | varchar | 45 | n/a | n |
| function | Employee role on food truck | enum | N/A | (Chef, Sous Chef, Driver) | n |
| phone\_number | Employee contact information, phone number | int | 10 | n/a | n |
| email | Employee contact information, email address | varchar | 45 | n/a | n |
| address | Current residence address of employee, billing and employment purposes | varchar | 45 | n/a | n |
| city | city in which employee residence address is located | varchar | 45 | n/a | n |
| state | state in which employee residence address is located | varchar | 45 | n/a | n |
| zipcode | zipcode in which employee residence address is located | int | 5 | n/a | n |
| SSN | Employee social security number, employment purposes | int | 9 | n/a | n |
| DOB | Employee date of birth | int | 8 | n/a | n |
| payrate | Current respective employee wage | int | 4 | 0.0-50.00 | n |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| foodtruck |  |  |  |  |  |
| *Element or value display name* | *Description* | *Data Type* | *Character Length* | *Acceptable Values* | *Null Values* |
| \*foodtruck\_ID | Primary key; foodtruck\_ID source each truck in database | int | 11 | 0.0-20 | N |
| VIN | VIN––vehicle identification number, source all food trucks in database | varchar | 45 | n/a | N |
| registration | Registration: expiration date source all food trucks in database | varchar | 45 | n/a | N |
| licenseplate | Licenseplate:renewal date source all food trucks in database | varchar | 45 | n/a | N |
| model | model type: year made source all food trucks in database | varchar | 45 | n/a | N |
| manufacturer | Food truck vehicle manufacturer (Mercedes) source all food trucks in database | varchar | 45 | n/a | N |
| inspection | Inspection: inspection status source all food trucks in database | varchar | 45 | n/a | N |
| mileage | Mileage: total miles driven with specific truck | varchar | 45 | n/a | N |

**III. Queries**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 | Q10 |
| Subquery |  |  |  |  |  |  | x |  |  | x |
| OUTER JOIN |  |  | x |  |  | x |  |  |  |  |
| GROUP BY |  | x |  | x | x |  | x | x |  |  |
| HAVING |  | x |  |  |  |  |  |  |  |  |
| ORDER BY |  |  |  | x | x |  |  | x | x | x |
| IN/NOT IN |  |  |  |  |  |  |  |  |  | x |
| Aggregate Function(s) |  | x |  | x | x |  | x | x |  |  |
| REGEXP | x | x |  |  |  |  |  |  | x |  |
| Date function(s) |  | x |  |  | x |  |  |  |  |  |
| IS NULL |  |  | x |  |  | x |  |  |  |  |

1. On 8/13/2020 the company may have accidentally provided all Mercedes model food trucks with expired beef. In order to understand how many customers were affected, which food trucks, if any, were given the tainted beef?

SELECT menu\_item\_ID, foodtruck\_ID, foodtruck.manufacturer

FROM shift

JOIN foodtruck USING(foodtruck\_ID)

JOIN foodtruck\_has\_menu\_item USING(foodtruck\_ID)

JOIN menu\_item USING(menu\_item\_ID)

JOIN recipe USING(menu\_item\_ID)

JOIN ingredient USING(ingredients\_ID)

WHERE ingredients\_ID = "beef"

AND foodtruck.manufacturer REGEXP "Mercedes"

AND shift.date = "2020-08-13";



*(In this case, no Mercedes trucks were given the meat, as none of the trucks had shifts on that day. The company lucked out since no customers were served tainted meat.)*

1. Since the inception of the company, customers have been allowed to enter into a promotional event that randomly selects a winning customer(s). This month (September 2020), two customers will win the raffle. However, before entering the names into a random number generator, the customers must be filtered to only provide customers who visited the Mercedez food trucks, as they were the only ones participating in the promotional feature. Secondly, only the customers who placed three or more orders are eligible to receive the prize, in an attempt to bring return customers (not new business).

SELECT customer.customer\_ID,

customer.fname,

customer.lname,

MONTHNAME(orders.order\_date) "Month"

FROM customer JOIN orders USING(customer\_ID)

JOIN order\_has\_menu\_item USING(transaction\_ID)

JOIN menu\_item USING(menu\_item\_id)

JOIN foodtruck\_has\_menu\_item USING(menu\_item\_ID)

JOIN foodtruck USING(foodtruck\_ID)

WHERE foodtruck.manufacturer REGEXP "Mercedez"

AND MONTHNAME(orders.order\_date) REGEXP "September"

GROUP BY customer\_ID

HAVING COUNT(transaction\_ID) >= 3;

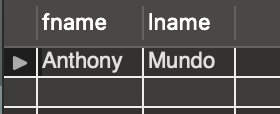


1. Some employees of the company are complaining they are not being scheduled for shifts. Investigate this claim and report any employees who have not been scheduled for a shift but have been hired.

SELECT fname, lname

FROM personnel LEFT JOIN shift USING(employee\_ID)

WHERE shift.employee\_ID IS NULL;



1. Report the number of employees working in each truck. The company wants to ensure each truck is paired with a similar number of crew maits to ensure one truck is not receiving more resources than any other. Organize the results in ascending order. Report all the food trucks’ staff counts so the company can compare the staffing of all 5 trucks to identify all understaffed trucks.

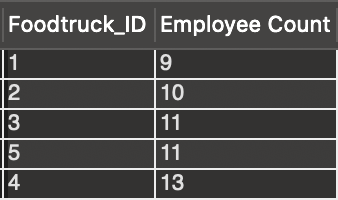
SELECT shift.Foodtruck\_ID,

COUNT(shift.employee\_ID) "Employee Count"

FROM shift

GROUP BY shift.foodtruck\_ID

ORDER BY COUNT(shift.employee\_ID);



1. The company is considering switching suppliers of red onions, a relatively minor item because the company usually uses white onions. Regardless, this means that all of their items with red onions will not be offered in the interim period. The company wants to know how many red onion items have been sold by each month, since the company would be without red onions for one month and would like to anticipate the loss of sales. Make sure to order the month by DESC order by number of transactions to illustrate a trend in the data, if any.

SELECT MONTH(order.order\_date) "Month",

COUNT(transaction\_ID) "Red Onion Transactions made during the Month"

FROM ingredient

JOIN recipe USING(ingredients\_ID)

JOIN menu\_item USING(menu\_item\_ID)

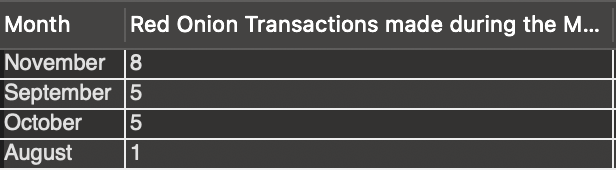
JOIN order\_has\_menu\_item USING(menu\_item\_ID)

JOIN order USING(transaction\_ID)

WHERE ingredient.name = "Red Onion"

GROUP BY MONTH(order.order\_date)

ORDER BY COUNT(transaction\_ID) DESC;



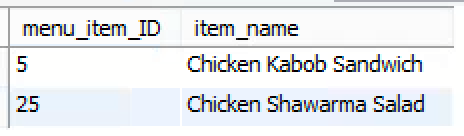
1. With 37 items, the company is looking to downsize its menu to reduce costs, overhead, and the amount of prep time its employees have to do each day. You definitely don't want to remove items that are popular and selling well, so find out which menu items, if any, have not been ordered since the company began operations?

SELECT menu\_item.menu\_item\_ID, item\_name

FROM menu\_item LEFT JOIN order\_has\_menu\_item

USING(menu\_item\_ID)

WHERE order\_has\_menu\_item.menu\_item\_ID IS NULL;



1. The company knows from email surveys that customers love the Kafta Kabob Plate.While this item is popular among recurring customers, the company also knows that it is not the most ordered item on the menu. Which menu items have been ordered equally or more times than the Kafta Kabob Plate?

SELECT menu\_item\_ID, item\_name

FROM menu\_item

JOIN order\_has\_menu\_item USING(menu\_item\_ID)

GROUP BY menu\_item\_ID HAVING

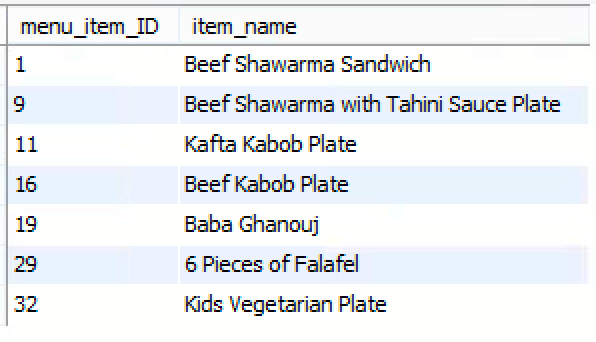
COUNT(menu\_item\_ID) >=

(SELECT COUNT(order\_has\_menu\_item.menu\_item\_ID)

FROM menu\_item

JOIN order\_has\_menu\_item USING(menu\_item\_ID)

WHERE item\_name = 'Kafta Kabob Plate');



1. Despite the company only being open for a few months, you have noticed stellar work from your employees and are looking to give raises. You approach your manager, who explains that, in an effort to fuel competition between the trucks, a raise is typically given to all employees of the truck with the most items sold. He then asks for your recommendation on which truck deserves a raise based on this criteria.

SELECT COUNT(\*), foodtruck\_ID

FROM foodtruck JOIN foodtruck\_has\_menu\_item USING(foodtruck\_ID)

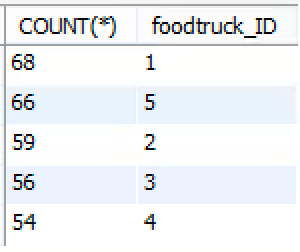
JOIN menu\_item USING(menu\_item\_ID)

JOIN order\_has\_menu\_item USING(menu\_item\_ID)

JOIN orders USING(transaction\_ID)

GROUP BY foodtruck\_ID

ORDER BY COUNT(\*) DESC;



1. It’s August 27, 2020, and you just received an email from Tom Collins, one of your chefs.He informs you that he unfortunately has tested positive for COVID-19. Customer safety and customer loyalty are the company’s two highest priorities, so you decide to reach out to all customers served by Tom to inform them of potential exposure. List the names, emails, and phone numbers of all customers who visited the truck when Tom was working over the past two weeks.

SELECT DISTINCT customer.fname, customer.lname, customer.email, customer.phone\_number

FROM customer JOIN orders USING(customer\_ID)

JOIN order\_has\_menu\_item USING(transaction\_ID)

JOIN menu\_item USING(menu\_item\_ID)

JOIN foodtruck\_has\_menu\_item USING(menu\_item\_ID)

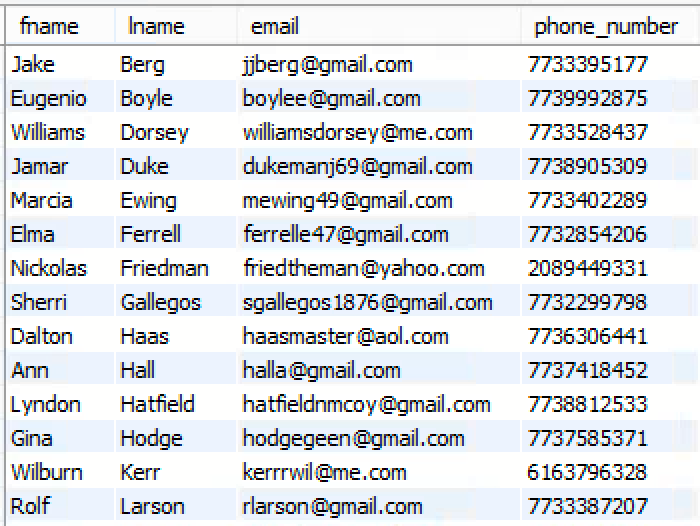
JOIN foodtruck USING(foodtruck\_ID)

JOIN shift USING(foodtruck\_ID)

JOIN personnel USING(employee\_ID)

WHERE personnel.fname REGEXP("Tom") AND personnel.lname REGEXP("Collins")

ORDER BY customer.lname;



*(results truncated)*

1. The company is looking to send out personalized emails to loyal customers, offering them a discount on their favorite items using their reward number. Report the customer’s name alphabetically by last name as well as their reward number and email. (Since we launched a reward signup option on our website, you cannot assume that every customer in the database has already placed an order. Only report customers who have placed at least one order.)

SELECT DISTINCT lname, fname, reward\_num, email

FROM customer JOIN orders USING(customer\_ID)

JOIN order\_has\_menu\_item USING(transaction\_ID)

WHERE customer\_ID IN (SELECT customer\_ID FROM orders)

ORDER BY lname;



*(results truncated)*