The Urban Fence

Amaya Alisca, Deandra Martin, Alex Mayer, Cam Reeves

CMS 375 – Database Design & Dev

Dr. Elva

April 22, 2021

Table of Contents

Overview of Universe of Discourse

A new start up restaurant, The Urban Fence, is scaling fast, leaving old methods in the dust. There is no system to display orders, food items, or payment method electronically. To keep up with the fast-paced business, we have decided to formulate a database where all transactions can be tracked and organized efficiently. We seek to store information about orders, chefs, our customers, food items we provide, and payment methods. The database will also be able to quickly query and answer questions such as:

- What are the available dishes?
- What is the price of spaghetti?
- What payment methods are offered?
- What is the chef's name?
- What payment method did ORD16 use?

Rationale

We decided to create this database for our friends at The Urban Fence to optimize their restaurant and allow them to compensate for their scaling business. With use of our database, the restaurant can access, update, and delete any information within the operations of the restaurant at an expedited pace. This advanced database will allow the management of Urban Fence to guide their attention to more important details of the business and let the database tale charge of the time invasive tasks of querying data.

Requirements Analysis

Structural Assertions

- An order must have at least one food item, an order total, and a unique order ID.

- A customer has a unique customer ID, order ID, name, and contact.

- A chef has a chef id, name, shift, and contact.

- A food item has a name, price, and ingredients.

- A payment has a payment ID, order ID, date, payment type, and amount.

- A coupon has a promocode and a fixed discount amount.

- A credit card’s type and currency are stored in the database.

Action Assertions

- Payment will only be accepted in the form of cash, credit, or coupon.

- Chefs have at least one shift to be employed at the restaurant.

- A customer cannot use more than one form of payment to fulfill an order.

- A customer can place multiple orders, but an order is place by exactly one customer.

- A payment can only fulfill one order, but a customer can make multiple payments.

- An order can have multiple food items and food items can belong to many orders.

- A chef can cook multiple food items but only one food item can be cooked by one chef.

- When an order is placed, the quantity of each food item is also recorded.

Derived Statements

- Order total is derived from the total price of food items and each quantity ordered.

- Payment amount is calculated from the order total minus a discount amount if the customer is using a coupon.

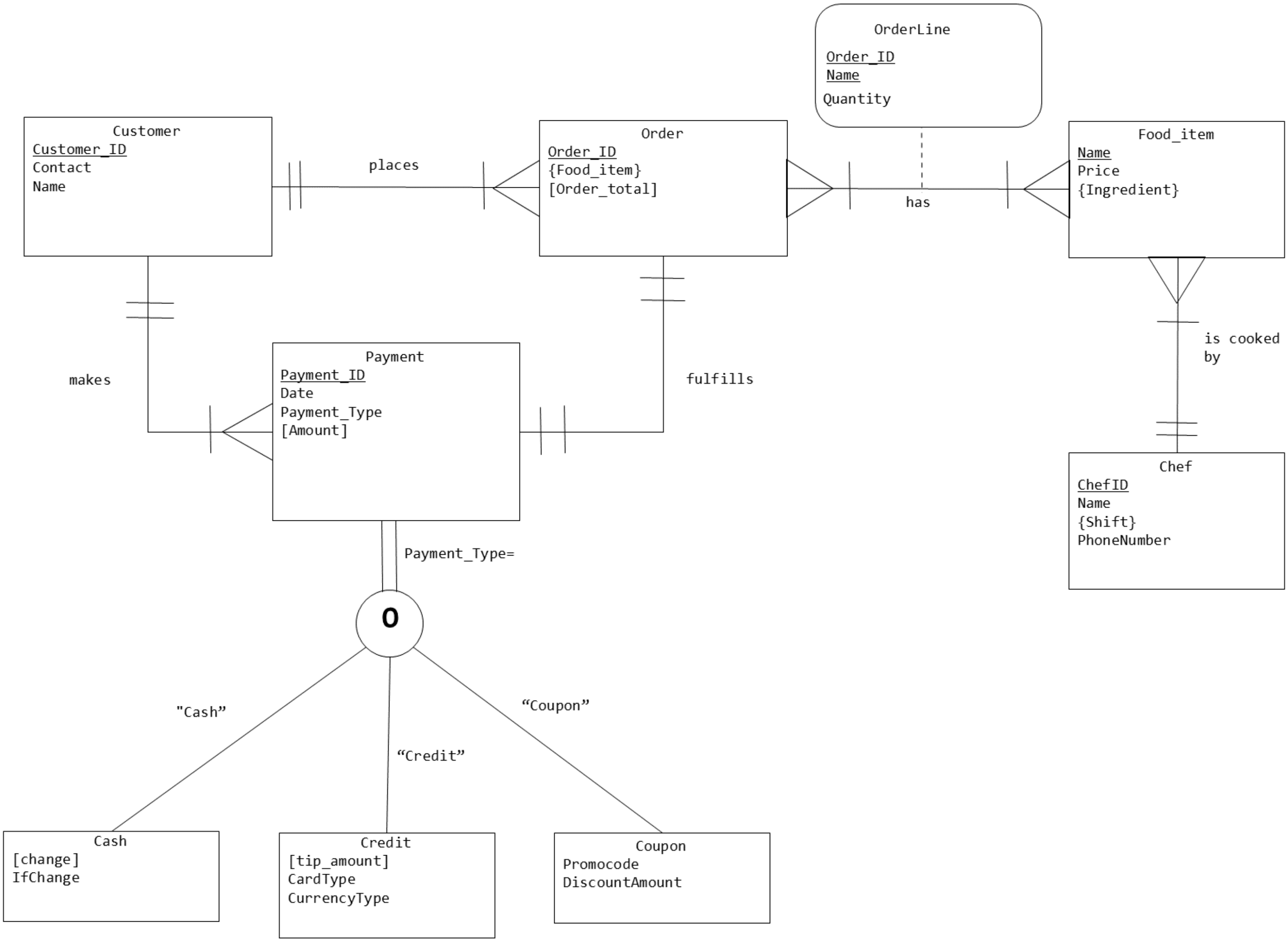
- Change is calculated from the amount of cash remaining after a payment is made.

- Using a credit card, customers have the option of adding a tip amount to their total if they wish.

Conceptual Design

In our EER diagram, the main entity types include Customer, Order, Food Item, Chef, and Payment. Our diagram has three one-to-many relationships, a single one-to-one and many-to-many relationship and an associative entity called OrderLine. Payment has a total specialization and overlap constraint with three sub-types, Cash, Credit, and Coupon.

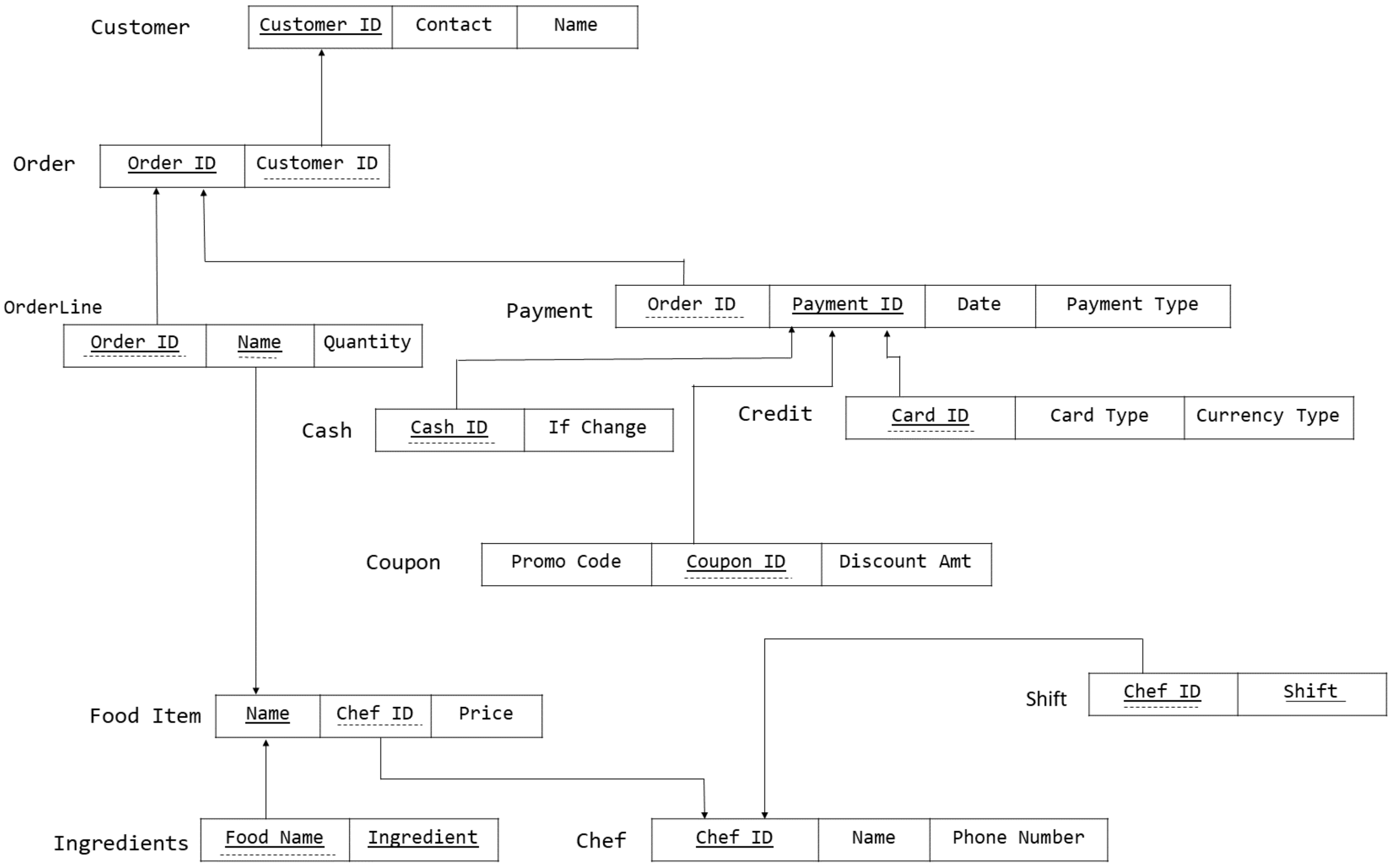
THE URBAN FENCE EERD



Logical Design

From our EERD, we came up with a set of schemas that would be represented as tables in our database. There are more schemas than entities from our EERD because we had to account for multivalued attributes like Ingredients, Shift and Food Item, and the many to many relationship between Order and Food Item represented by OrderLine.

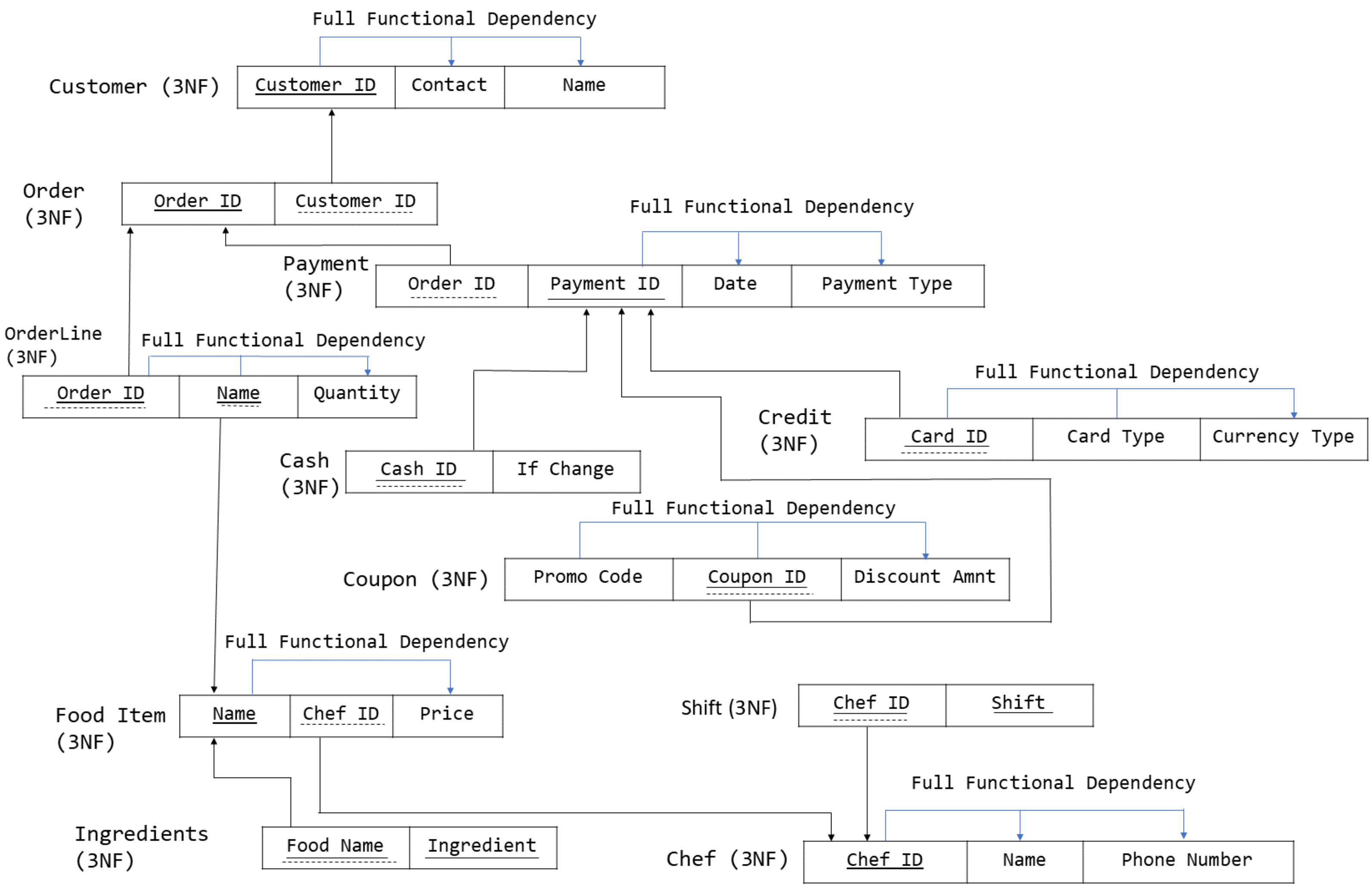
THE URBAN FENCE TABLE SCHEMAS



Schema Refinement

Our schema refinement phase went rather smoothly. We did not have any partial or transitive functional dependencies in our schema design, so all our schemas were in fully normalized and ready to be coded in MySQL.

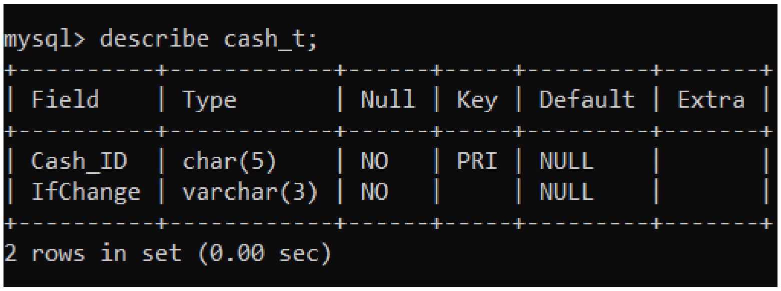
THE URBAN FENCE SCHEMA REFINEMENT



Physical Design

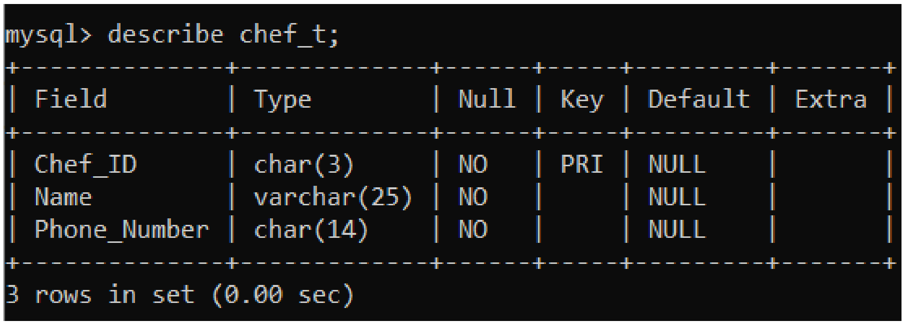
Schemas of the relations in our database:

1. Cash_T schema



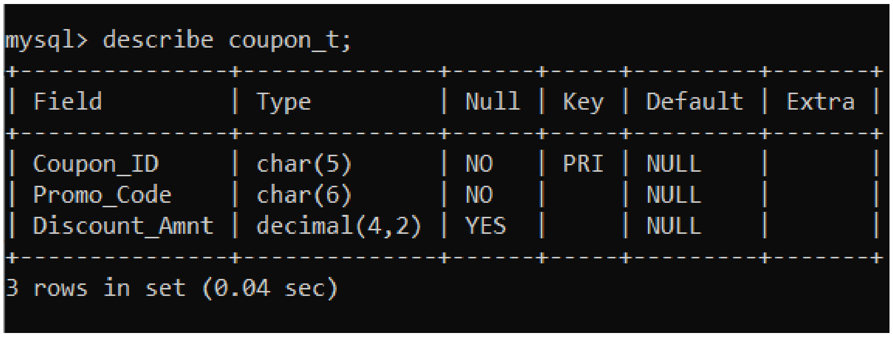
```
mysql> describe cash_t;
+----------+------------+------+-----+---------+-------+
| Field    | Type       | Null | Key | Default | Extra |
+----------+------------+------+-----+---------+-------+
| Cash_ID  | char(5)    | NO   | PRI | NULL    |       |
| IfChange | varchar(3) | NO   |     | NULL    |       |
+----------+------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

2. Chef_T schema



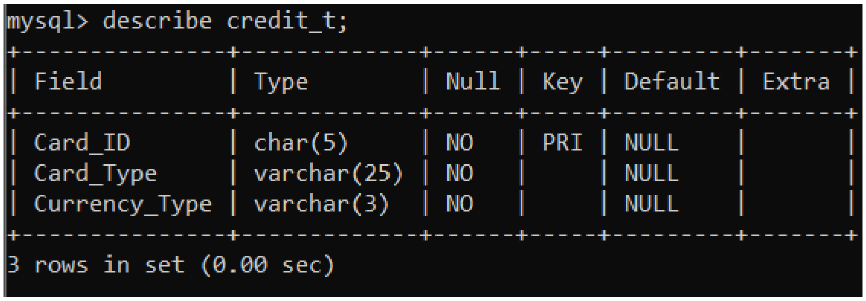
```
mysql> describe chef_t;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| Chef_ID      | char(3)     | NO   | PRI | NULL    |       |
| Name         | varchar(25) | NO   |     | NULL    |       |
| Phone_Number | char(14)    | NO   |     | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

3. Coupon_T schema



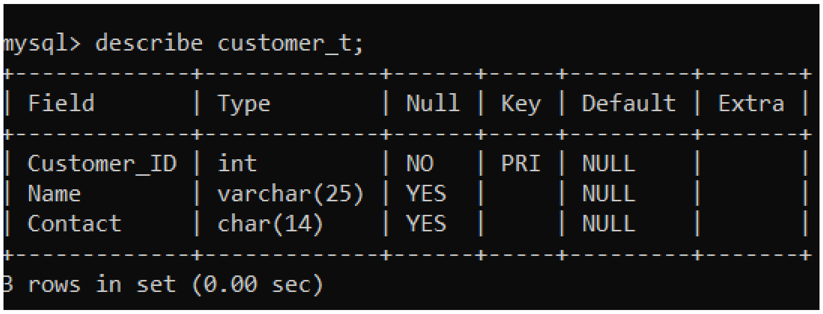
```
mysql> describe coupon_t;
+---------------+--------------+------+-----+---------+-------+
| Field         | Type         | Null | Key | Default | Extra |
+---------------+--------------+------+-----+---------+-------+
| Coupon_ID     | char(5)      | NO   | PRI | NULL    |       |
| Promo_Code    | char(6)      | NO   |     | NULL    |       |
| Discount_Amnt | decimal(4,2) | YES  |     | NULL    |       |
+---------------+--------------+------+-----+---------+-------+
3 rows in set (0.04 sec)
```

4. Credit_T schema



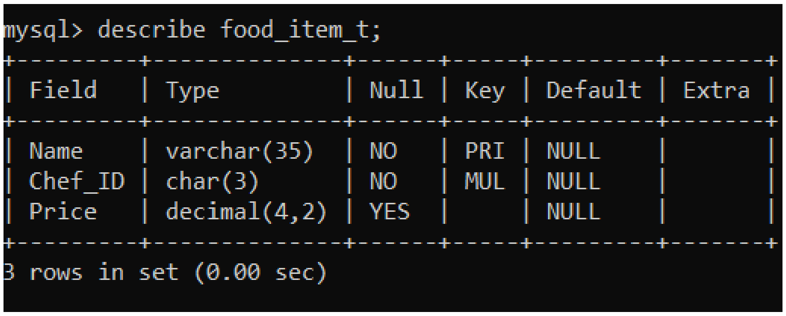
```
mysql> describe credit_t;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| Card_ID       | char(5)     | NO   | PRI | NULL    |       |
| Card_Type     | varchar(25) | NO   |     | NULL    |       |
| Currency_Type | varchar(3)  | NO   |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

5. Customer_T schema



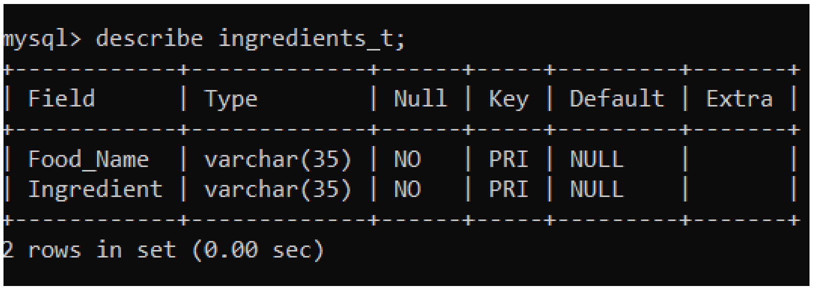
```
mysql> describe customer_t;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| Customer_ID | int         | NO   | PRI | NULL    |       |
| Name        | varchar(25) | YES  |     | NULL    |       |
| Contact     | char(14)    | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

6. Food_Item_T schema



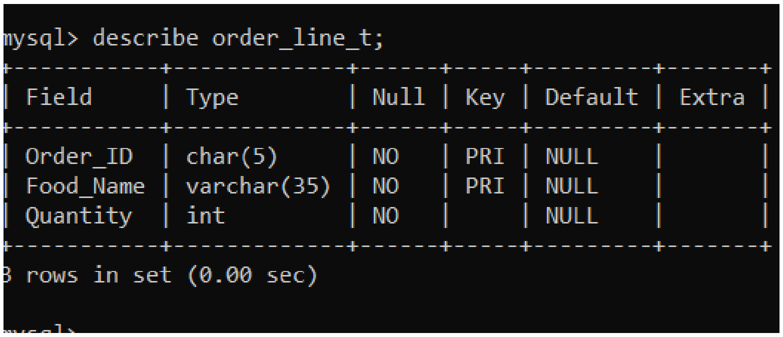
```
mysql> describe food_item_t;
+---------+--------------+------+-----+---------+-------+
| Field   | Type         | Null | Key | Default | Extra |
+---------+--------------+------+-----+---------+-------+
| Name    | varchar(35)  | NO   | PRI | NULL    |       |
| Chef_ID | char(3)      | NO   | MUL | NULL    |       |
| Price   | decimal(4,2) | YES  |     | NULL    |       |
+---------+--------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

7. Ingredients_T schema



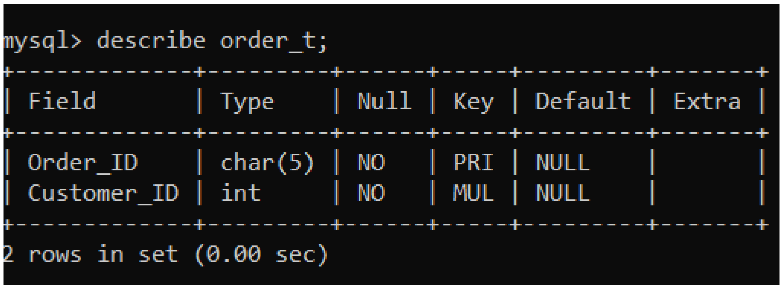
```
mysql> describe ingredients_t;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| Food_Name  | varchar(35) | NO   | PRI | NULL    |       |
| Ingredient | varchar(35) | NO   | PRI | NULL    |       |
+------------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

8. Order_Line_T schema



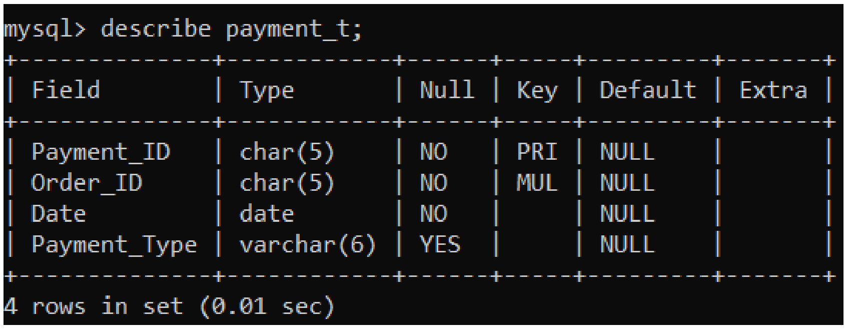
```
mysql> describe order_line_t;
+-----------+-------------+------+-----+---------+-------+
| Field     | Type        | Null | Key | Default | Extra |
+-----------+-------------+------+-----+---------+-------+
| Order_ID  | char(5)     | NO   | PRI | NULL    |       |
| Food_Name | varchar(35) | NO   | PRI | NULL    |       |
| Quantity  | int         | NO   |     | NULL    |       |
+-----------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

9. Order_T schema



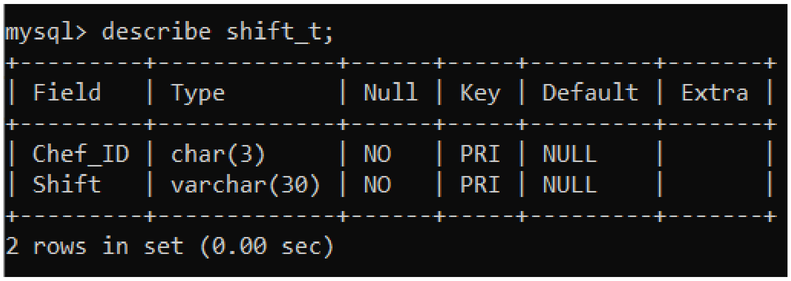
```
mysql> describe order_t;
+-------------+---------+------+-----+---------+-------+
| Field       | Type    | Null | Key | Default | Extra |
+-------------+---------+------+-----+---------+-------+
| Order_ID    | char(5) | NO   | PRI | NULL    |       |
| Customer_ID | int     | NO   | MUL | NULL    |       |
+-------------+---------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

10. Payment_T schema



```
mysql> describe payment_t;
+--------------+------------+------+-----+---------+-------+
| Field        | Type       | Null | Key | Default | Extra |
+--------------+------------+------+-----+---------+-------+
| Payment_ID   | char(5)    | NO   | PRI | NULL    |       |
| Order_ID     | char(5)    | NO   | MUL | NULL    |       |
| Date         | date       | NO   |     | NULL    |       |
| Payment_Type | varchar(6) | YES  |     | NULL    |       |
+--------------+------------+------+-----+---------+-------+
4 rows in set (0.01 sec)
```

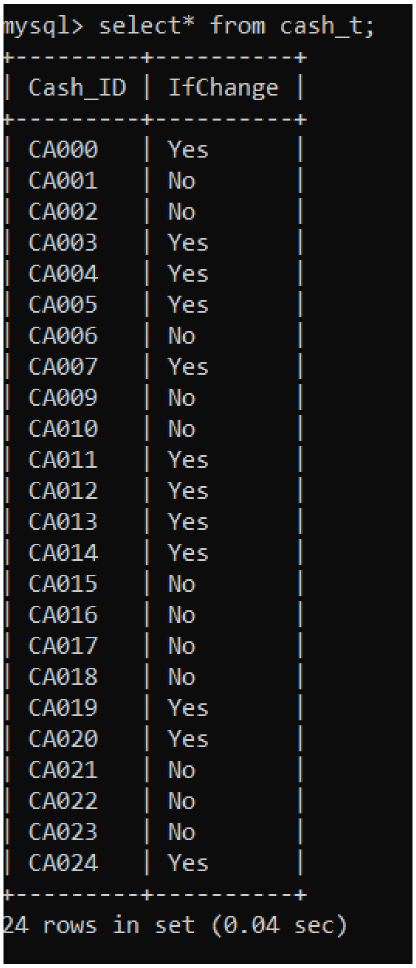
11. Shift_T schema



```
mysql> describe shift_t;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| Chef_ID | char(3)     | NO   | PRI | NULL    |       |
| Shift   | varchar(30) | NO   | PRI | NULL    |       |
+---------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

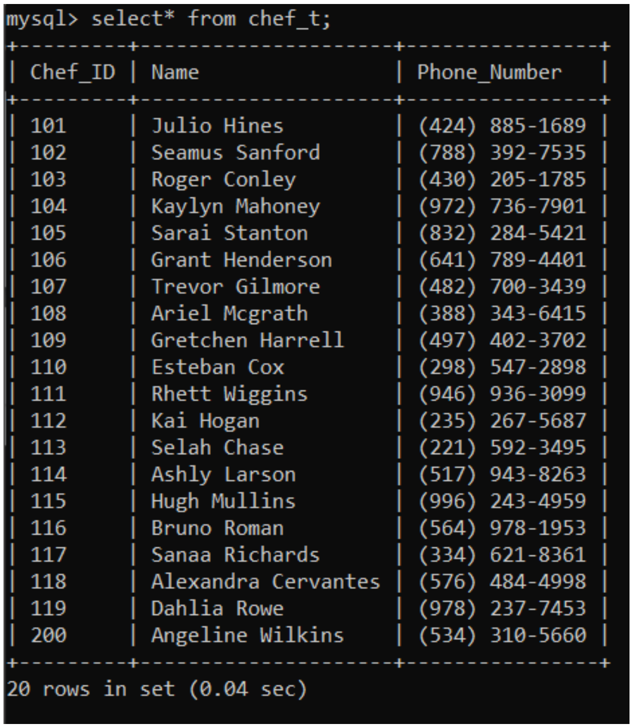
Population of the database:

Cash_T



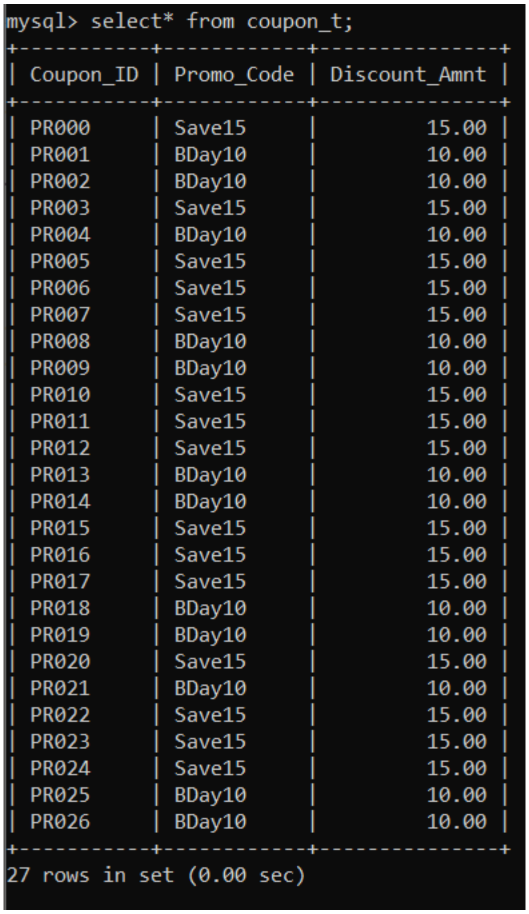
```
mysql> select* from cash_t;
+---------+----------+
| Cash_ID | IfChange |
+---------+----------+
| CA000   | Yes      |
| CA001   | No       |
| CA002   | No       |
| CA003   | Yes      |
| CA004   | Yes      |
| CA005   | Yes      |
| CA006   | No       |
| CA007   | Yes      |
| CA009   | No       |
| CA010   | No       |
| CA011   | Yes      |
| CA012   | Yes      |
| CA013   | Yes      |
| CA014   | Yes      |
| CA015   | No       |
| CA016   | No       |
| CA017   | No       |
| CA018   | No       |
| CA019   | Yes      |
| CA020   | Yes      |
| CA021   | No       |
| CA022   | No       |
| CA023   | No       |
| CA024   | Yes      |
+---------+----------+
24 rows in set (0.04 sec)
```

Chef_T



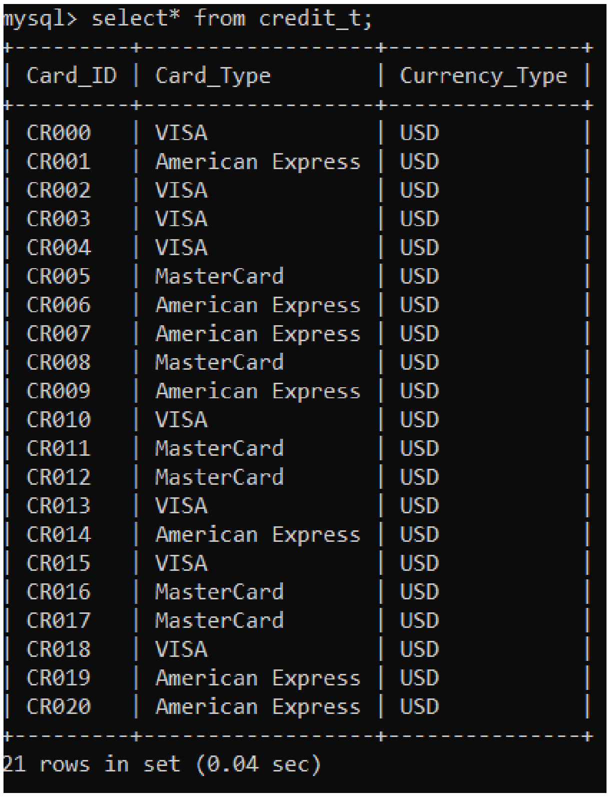
```
mysql> select* from chef_t;
+---------+---------------------+----------------+
| Chef_ID | Name                | Phone_Number   |
+---------+---------------------+----------------+
| 101     | Julio Hines         | (424) 885-1689 |
| 102     | Seamus Sanford      | (788) 392-7535 |
| 103     | Roger Conley        | (430) 205-1785 |
| 104     | Kaylyn Mahoney      | (972) 736-7901 |
| 105     | Sarai Stanton       | (832) 284-5421 |
| 106     | Grant Henderson     | (641) 789-4401 |
| 107     | Trevor Gilmore      | (482) 700-3439 |
| 108     | Ariel Mcgrath       | (388) 343-6415 |
| 109     | Gretchen Harrell    | (497) 402-3702 |
| 110     | Esteban Cox         | (298) 547-2898 |
| 111     | Rhett Wiggins       | (946) 936-3099 |
| 112     | Kai Hogan           | (235) 267-5687 |
| 113     | Selah Chase         | (221) 592-3495 |
| 114     | Ashly Larson        | (517) 943-8263 |
| 115     | Hugh Mullins        | (996) 243-4959 |
| 116     | Bruno Roman         | (564) 978-1953 |
| 117     | Sanaa Richards      | (334) 621-8361 |
| 118     | Alexandra Cervantes | (576) 484-4998 |
| 119     | Dahlia Rowe         | (978) 237-7453 |
| 200     | Angeline Wilkins    | (534) 310-5660 |
+---------+---------------------+----------------+
20 rows in set (0.04 sec)
```

Coupon_T



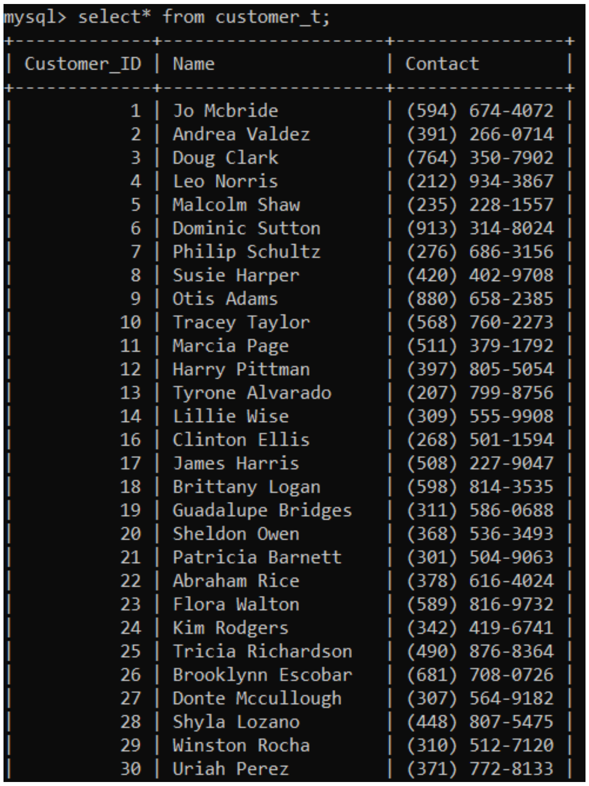
```
mysql> select* from coupon_t;
+-----------+------------+---------------+
| Coupon_ID | Promo_Code | Discount_Amnt |
+-----------+------------+---------------+
| PR000     | Save15     |         15.00 |
| PR001     | BDay10     |         10.00 |
| PR002     | BDay10     |         10.00 |
| PR003     | Save15     |         15.00 |
| PR004     | BDay10     |         10.00 |
| PR005     | Save15     |         15.00 |
| PR006     | Save15     |         15.00 |
| PR007     | Save15     |         15.00 |
| PR008     | BDay10     |         10.00 |
| PR009     | BDay10     |         10.00 |
| PR010     | Save15     |         15.00 |
| PR011     | Save15     |         15.00 |
| PR012     | Save15     |         15.00 |
| PR013     | BDay10     |         10.00 |
| PR014     | BDay10     |         10.00 |
| PR015     | Save15     |         15.00 |
| PR016     | Save15     |         15.00 |
| PR017     | Save15     |         15.00 |
| PR018     | BDay10     |         10.00 |
| PR019     | BDay10     |         10.00 |
| PR020     | Save15     |         15.00 |
| PR021     | BDay10     |         10.00 |
| PR022     | Save15     |         15.00 |
| PR023     | Save15     |         15.00 |
| PR024     | Save15     |         15.00 |
| PR025     | BDay10     |         10.00 |
| PR026     | BDay10     |         10.00 |
+-----------+------------+---------------+
27 rows in set (0.00 sec)
```

Credit_T



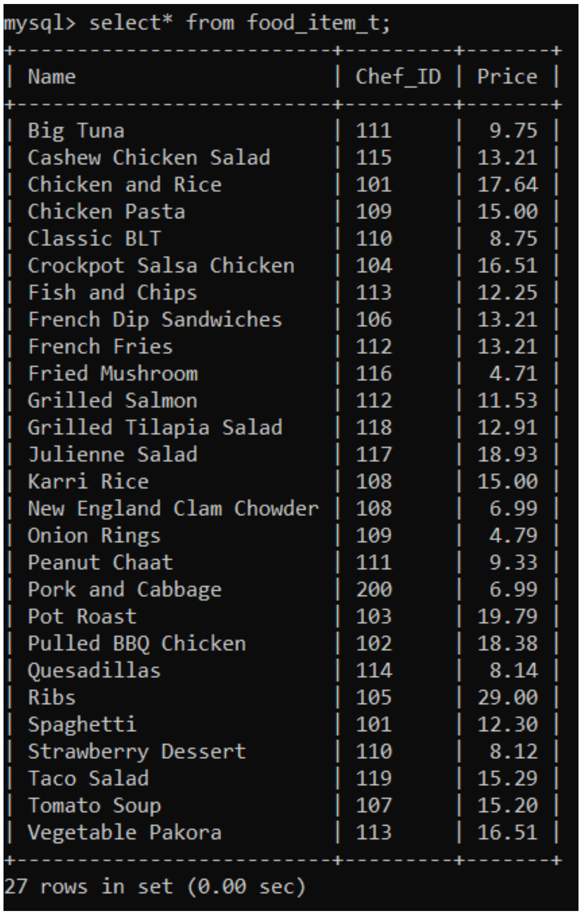
```
mysql> select* from credit_t;
+---------+------------------+---------------+
| Card_ID | Card_Type        | Currency_Type |
+---------+------------------+---------------+
| CR000   | VISA             | USD           |
| CR001   | American Express | USD           |
| CR002   | VISA             | USD           |
| CR003   | VISA             | USD           |
| CR004   | VISA             | USD           |
| CR005   | MasterCard       | USD           |
| CR006   | American Express | USD           |
| CR007   | American Express | USD           |
| CR008   | MasterCard       | USD           |
| CR009   | American Express | USD           |
| CR010   | VISA             | USD           |
| CR011   | MasterCard       | USD           |
| CR012   | MasterCard       | USD           |
| CR013   | VISA             | USD           |
| CR014   | American Express | USD           |
| CR015   | VISA             | USD           |
| CR016   | MasterCard       | USD           |
| CR017   | MasterCard       | USD           |
| CR018   | VISA             | USD           |
| CR019   | American Express | USD           |
| CR020   | American Express | USD           |
+---------+------------------+---------------+
21 rows in set (0.04 sec)
```

Customer_T (cropped results)



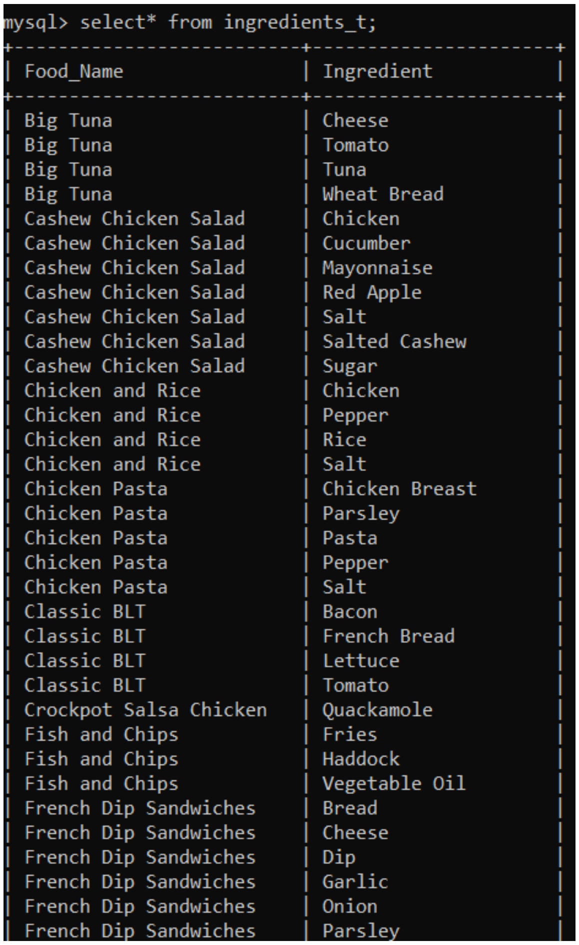
```
mysql> select* from customer_t;
+-------------+--------------------+---------------+
| Customer_ID | Name               | Contact       |
+-------------+--------------------+---------------+
|           1 | Jo Mcbride         | (594) 674-4072 |
|           2 | Andrea Valdez      | (391) 266-0714 |
|           3 | Doug Clark         | (764) 350-7902 |
|           4 | Leo Norris         | (212) 934-3867 |
|           5 | Malcolm Shaw       | (235) 228-1557 |
|           6 | Dominic Sutton     | (913) 314-8024 |
|           7 | Philip Schultz     | (276) 686-3156 |
|           8 | Susie Harper       | (420) 402-9708 |
|           9 | Otis Adams         | (880) 658-2385 |
|          10 | Tracey Taylor      | (568) 760-2273 |
|          11 | Marcia Page        | (511) 379-1792 |
|          12 | Harry Pittman      | (397) 805-5054 |
|          13 | Tyrone Alvarado    | (207) 799-8756 |
|          14 | Lillie Wise        | (309) 555-9908 |
|          16 | Clinton Ellis      | (268) 501-1594 |
|          17 | James Harris       | (508) 227-9047 |
|          18 | Brittany Logan     | (598) 814-3535 |
|          19 | Guadalupe Bridges  | (311) 586-0688 |
|          20 | Sheldon Owen       | (368) 536-3493 |
|          21 | Patricia Barnett   | (301) 504-9063 |
|          22 | Abraham Rice       | (378) 616-4024 |
|          23 | Flora Walton       | (589) 816-9732 |
|          24 | Kim Rodgers        | (342) 419-6741 |
|          25 | Tricia Richardson  | (490) 876-8364 |
|          26 | Brooklynn Escobar  | (681) 708-0726 |
|          27 | Donte Mccullough   | (307) 564-9182 |
|          28 | Shyla Lozano       | (448) 807-5475 |
|          29 | Winston Rocha      | (310) 512-7120 |
|          30 | Uriah Perez        | (371) 772-8133 |
```

Food_Item_T (cropped results)



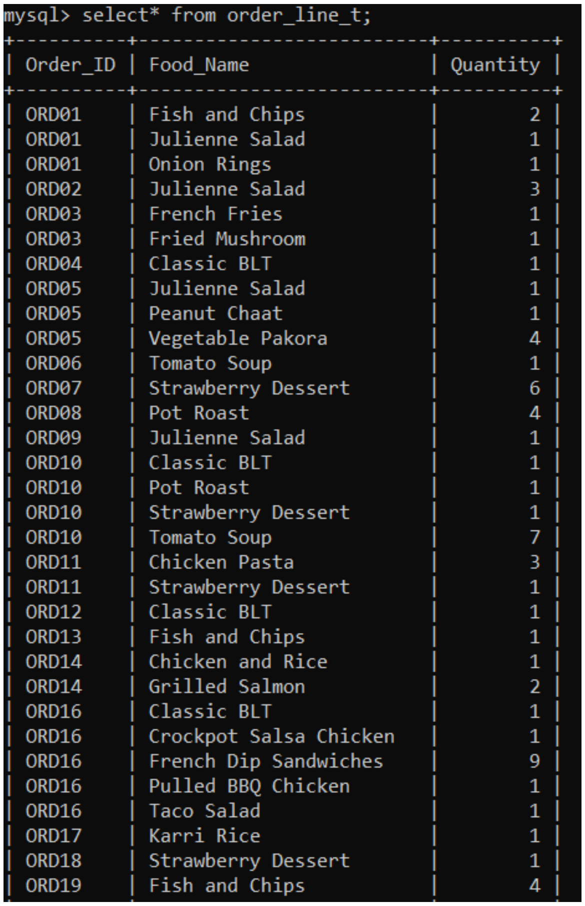
```
mysql> select* from food_item_t;
+-------------------------+---------+-------+
| Name                    | Chef_ID | Price |
+-------------------------+---------+-------+
| Big Tuna                | 111     |  9.75 |
| Cashew Chicken Salad    | 115     | 13.21 |
| Chicken and Rice        | 101     | 17.64 |
| Chicken Pasta           | 109     | 15.00 |
| Classic BLT             | 110     |  8.75 |
| Crockpot Salsa Chicken  | 104     | 16.51 |
| Fish and Chips          | 113     | 12.25 |
| French Dip Sandwiches   | 106     | 13.21 |
| French Fries            | 112     | 13.21 |
| Fried Mushroom          | 116     |  4.71 |
| Grilled Salmon          | 112     | 11.53 |
| Grilled Tilapia Salad   | 118     | 12.91 |
| Julienne Salad          | 117     | 18.93 |
| Karri Rice              | 108     | 15.00 |
| New England Clam Chowder | 108    |  6.99 |
| Onion Rings             | 109     |  4.79 |
| Peanut Chaat            | 111     |  9.33 |
| Pork and Cabbage        | 200     |  6.99 |
| Pot Roast               | 103     | 19.79 |
| Pulled BBQ Chicken      | 102     | 18.38 |
| Quesadillas             | 114     |  8.14 |
| Ribs                    | 105     | 29.00 |
| Spaghetti               | 101     | 12.30 |
| Strawberry Dessert      | 110     |  8.12 |
| Taco Salad              | 119     | 15.29 |
| Tomato Soup             | 107     | 15.20 |
| Vegetable Pakora        | 113     | 16.51 |
+-------------------------+---------+-------+
27 rows in set (0.00 sec)
```

Ingredients_T (cropped results)



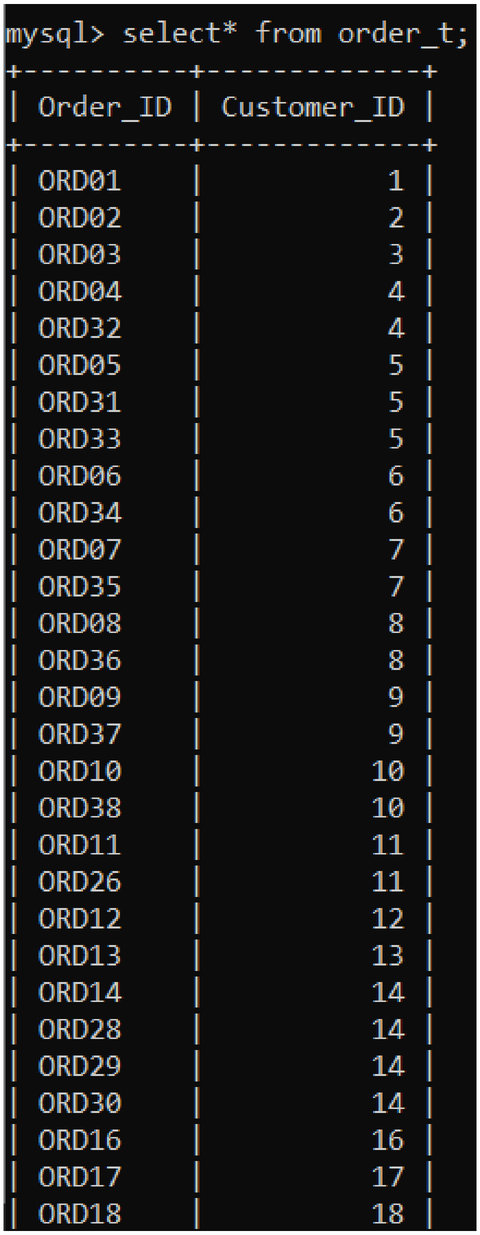
```
mysql> select* from ingredients_t;
+---------------------------+-----------------------+
| Food_Name                 | Ingredient            |
+---------------------------+-----------------------+
| Big Tuna                  | Cheese                |
| Big Tuna                  | Tomato                |
| Big Tuna                  | Tuna                  |
| Big Tuna                  | Wheat Bread           |
| Cashew Chicken Salad      | Chicken               |
| Cashew Chicken Salad      | Cucumber              |
| Cashew Chicken Salad      | Mayonnaise            |
| Cashew Chicken Salad      | Red Apple             |
| Cashew Chicken Salad      | Salt                  |
| Cashew Chicken Salad      | Salted Cashew         |
| Cashew Chicken Salad      | Sugar                 |
| Chicken and Rice          | Chicken               |
| Chicken and Rice          | Pepper                |
| Chicken and Rice          | Rice                  |
| Chicken and Rice          | Salt                  |
| Chicken Pasta             | Chicken Breast        |
| Chicken Pasta             | Parsley               |
| Chicken Pasta             | Pasta                 |
| Chicken Pasta             | Pepper                |
| Chicken Pasta             | Salt                  |
| Classic BLT               | Bacon                 |
| Classic BLT               | French Bread          |
| Classic BLT               | Lettuce               |
| Classic BLT               | Tomato                |
| Crockpot Salsa Chicken    | Quackamole            |
| Fish and Chips            | Fries                 |
| Fish and Chips            | Haddock               |
| Fish and Chips            | Vegetable Oil         |
| French Dip Sandwiches     | Bread                 |
| French Dip Sandwiches     | Cheese                |
| French Dip Sandwiches     | Dip                   |
| French Dip Sandwiches     | Garlic                |
| French Dip Sandwiches     | Onion                 |
| French Dip Sandwiches     | Parsley               |
```

Order_Line_T (cropped results)



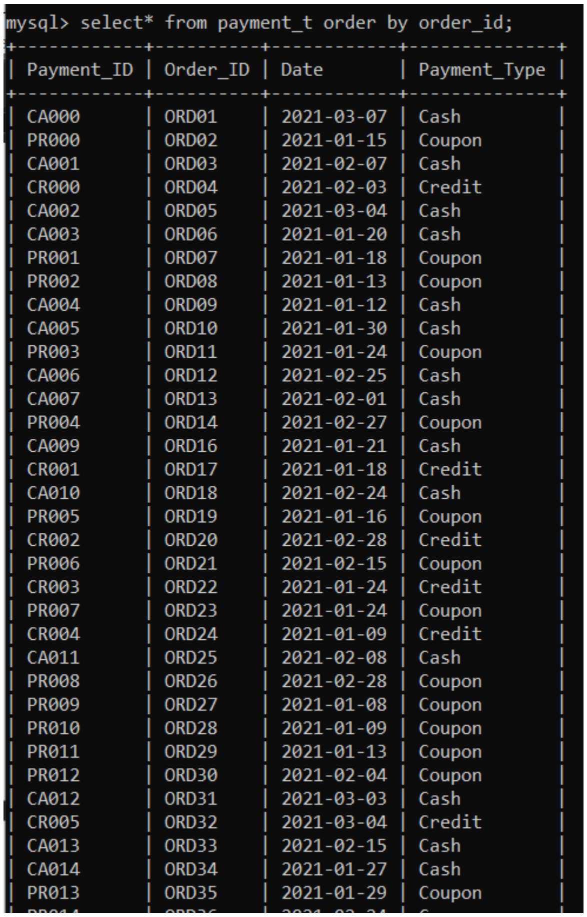
```
mysql> select* from order_line_t;
+----------+---------------------------+----------+
| Order_ID | Food_Name                 | Quantity |
+----------+---------------------------+----------+
| ORD01    | Fish and Chips            |        2 |
| ORD01    | Julienne Salad            |        1 |
| ORD01    | Onion Rings               |        1 |
| ORD02    | Julienne Salad            |        3 |
| ORD03    | French Fries              |        1 |
| ORD03    | Fried Mushroom            |        1 |
| ORD04    | Classic BLT               |        1 |
| ORD05    | Julienne Salad            |        1 |
| ORD05    | Peanut Chaat              |        1 |
| ORD05    | Vegetable Pakora          |        4 |
| ORD06    | Tomato Soup               |        1 |
| ORD07    | Strawberry Dessert        |        6 |
| ORD08    | Pot Roast                 |        4 |
| ORD09    | Julienne Salad            |        1 |
| ORD10    | Classic BLT               |        1 |
| ORD10    | Pot Roast                 |        1 |
| ORD10    | Strawberry Dessert        |        1 |
| ORD10    | Tomato Soup               |        7 |
| ORD11    | Chicken Pasta             |        3 |
| ORD11    | Strawberry Dessert        |        1 |
| ORD12    | Classic BLT               |        1 |
| ORD13    | Fish and Chips            |        1 |
| ORD14    | Chicken and Rice          |        1 |
| ORD14    | Grilled Salmon            |        2 |
| ORD16    | Classic BLT               |        1 |
| ORD16    | Crockpot Salsa Chicken    |        1 |
| ORD16    | French Dip Sandwiches     |        9 |
| ORD16    | Pulled BBQ Chicken        |        1 |
| ORD16    | Taco Salad                |        1 |
| ORD17    | Karri Rice                |        1 |
| ORD18    | Strawberry Dessert        |        1 |
| ORD19    | Fish and Chips            |        4 |
```

Order_T (cropped results)



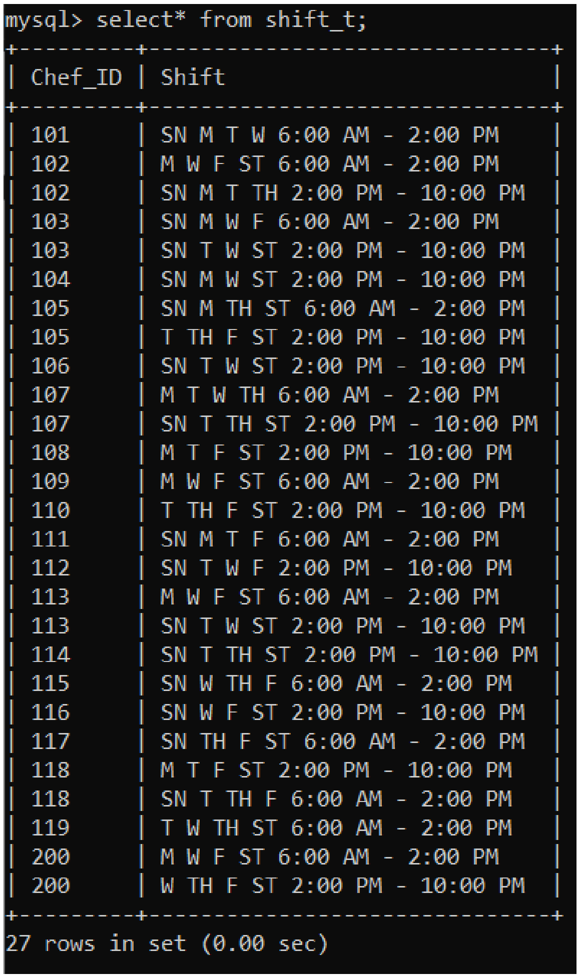
```
mysql> select* from order_t;
+----------+-------------+
| Order_ID | Customer_ID |
+----------+-------------+
| ORD01    |           1 |
| ORD02    |           2 |
| ORD03    |           3 |
| ORD04    |           4 |
| ORD32    |           4 |
| ORD05    |           5 |
| ORD31    |           5 |
| ORD33    |           5 |
| ORD06    |           6 |
| ORD34    |           6 |
| ORD07    |           7 |
| ORD35    |           7 |
| ORD08    |           8 |
| ORD36    |           8 |
| ORD09    |           9 |
| ORD37    |           9 |
| ORD10    |          10 |
| ORD38    |          10 |
| ORD11    |          11 |
| ORD26    |          11 |
| ORD12    |          12 |
| ORD13    |          13 |
| ORD14    |          14 |
| ORD28    |          14 |
| ORD29    |          14 |
| ORD30    |          14 |
| ORD16    |          16 |
| ORD17    |          17 |
| ORD18    |          18 |
```

Payment_T (cropped results)



```
mysql> select* from payment_t order by order_id;
+------------+----------+------------+--------------+
| Payment_ID | Order_ID | Date       | Payment_Type |
+------------+----------+------------+--------------+
| CA000      | ORD01    | 2021-03-07 | Cash         |
| PR000      | ORD02    | 2021-01-15 | Coupon       |
| CA001      | ORD03    | 2021-02-07 | Cash         |
| CR000      | ORD04    | 2021-02-03 | Credit       |
| CA002      | ORD05    | 2021-03-04 | Cash         |
| CA003      | ORD06    | 2021-01-20 | Cash         |
| PR001      | ORD07    | 2021-01-18 | Coupon       |
| PR002      | ORD08    | 2021-01-13 | Coupon       |
| CA004      | ORD09    | 2021-01-12 | Cash         |
| CA005      | ORD10    | 2021-01-30 | Cash         |
| PR003      | ORD11    | 2021-01-24 | Coupon       |
| CA006      | ORD12    | 2021-02-25 | Cash         |
| CA007      | ORD13    | 2021-02-01 | Cash         |
| PR004      | ORD14    | 2021-02-27 | Coupon       |
| CA009      | ORD16    | 2021-01-21 | Cash         |
| CR001      | ORD17    | 2021-01-18 | Credit       |
| CA010      | ORD18    | 2021-02-24 | Cash         |
| PR005      | ORD19    | 2021-01-16 | Coupon       |
| CR002      | ORD20    | 2021-02-28 | Credit       |
| PR006      | ORD21    | 2021-02-15 | Coupon       |
| CR003      | ORD22    | 2021-01-24 | Credit       |
| PR007      | ORD23    | 2021-01-24 | Coupon       |
| CR004      | ORD24    | 2021-01-09 | Credit       |
| CA011      | ORD25    | 2021-02-08 | Cash         |
| PR008      | ORD26    | 2021-02-28 | Coupon       |
| PR009      | ORD27    | 2021-01-08 | Coupon       |
| PR010      | ORD28    | 2021-01-09 | Coupon       |
| PR011      | ORD29    | 2021-01-13 | Coupon       |
| PR012      | ORD30    | 2021-02-04 | Coupon       |
| CA012      | ORD31    | 2021-03-03 | Cash         |
| CR005      | ORD32    | 2021-03-04 | Credit       |
| CA013      | ORD33    | 2021-02-15 | Cash         |
| CA014      | ORD34    | 2021-01-27 | Cash         |
| PR013      | ORD35    | 2021-01-29 | Coupon       |
```

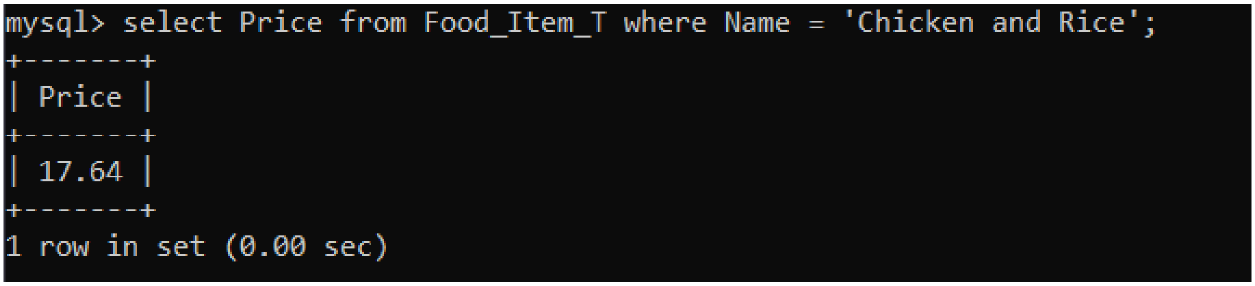
Shift_T



```
mysql> select* from shift_t;
+---------+-------------------------------+
| Chef_ID | Shift                         |
+---------+-------------------------------+
| 101     | SN M T W 6:00 AM - 2:00 PM    |
| 102     | M W F ST 6:00 AM - 2:00 PM    |
| 102     | SN M T TH 2:00 PM - 10:00 PM  |
| 103     | SN M W F 6:00 AM - 2:00 PM    |
| 103     | SN T W ST 2:00 PM - 10:00 PM  |
| 104     | SN M W ST 2:00 PM - 10:00 PM  |
| 105     | SN M TH ST 6:00 AM - 2:00 PM  |
| 105     | T TH F ST 2:00 PM - 10:00 PM  |
| 106     | SN T W ST 2:00 PM - 10:00 PM  |
| 107     | M T W TH 6:00 AM - 2:00 PM    |
| 107     | SN T TH ST 2:00 PM - 10:00 PM |
| 108     | M T F ST 2:00 PM - 10:00 PM   |
| 109     | M W F ST 6:00 AM - 2:00 PM    |
| 110     | T TH F ST 2:00 PM - 10:00 PM  |
| 111     | SN M T F 6:00 AM - 2:00 PM    |
| 112     | SN T W F 2:00 PM - 10:00 PM   |
| 113     | M W F ST 6:00 AM - 2:00 PM    |
| 113     | SN T W ST 2:00 PM - 10:00 PM  |
| 114     | SN T TH ST 2:00 PM - 10:00 PM |
| 115     | SN W TH F 6:00 AM - 2:00 PM   |
| 116     | SN W F ST 2:00 PM - 10:00 PM  |
| 117     | SN TH F ST 6:00 AM - 2:00 PM  |
| 118     | M T F ST 2:00 PM - 10:00 PM   |
| 118     | SN T TH F 6:00 AM - 2:00 PM   |
| 119     | T W TH ST 6:00 AM - 2:00 PM   |
| 200     | M W F ST 6:00 AM - 2:00 PM    |
| 200     | W TH F ST 2:00 PM - 10:00 PM  |
+---------+-------------------------------+
27 rows in set (0.00 sec)
```

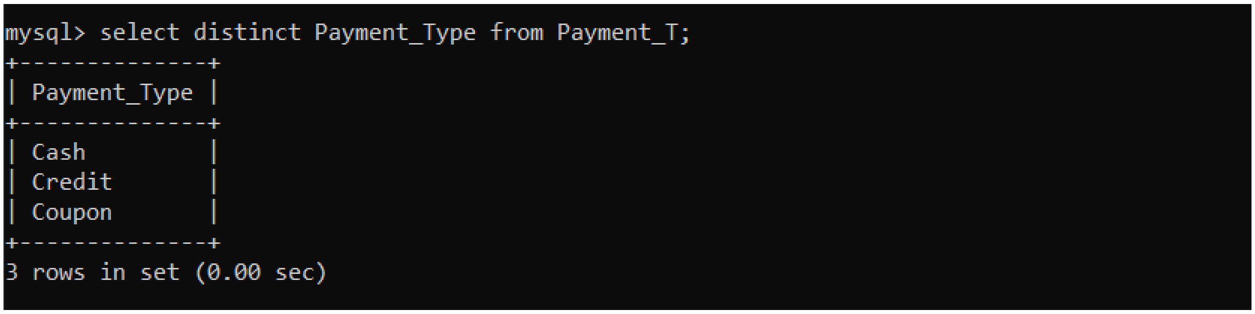
Execution of queries and other DML commands:

1. Find the price of chicken and rice.



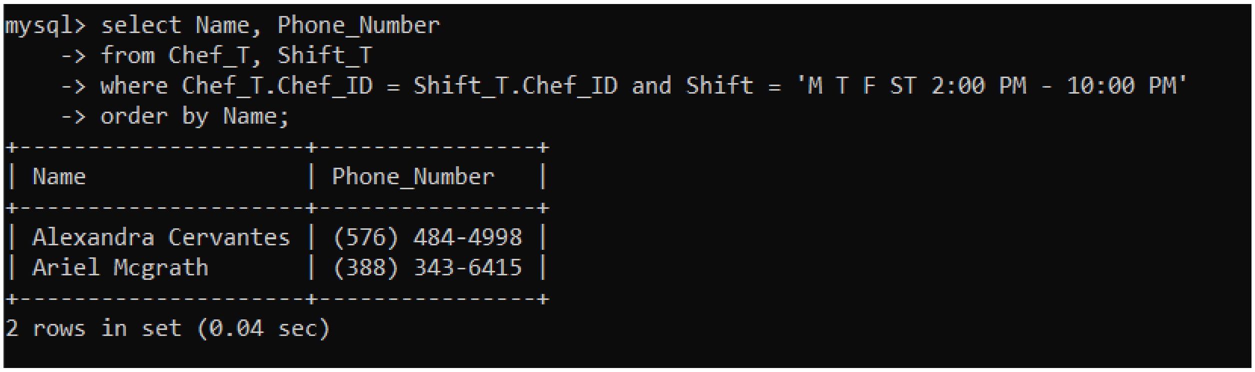
```
mysql> select Price from Food_Item_T where Name = 'Chicken and Rice';
+-------+
| Price |
+-------+
| 17.64 |
+-------+
1 row in set (0.00 sec)
```

2. Print a list of all payment methods accepted.



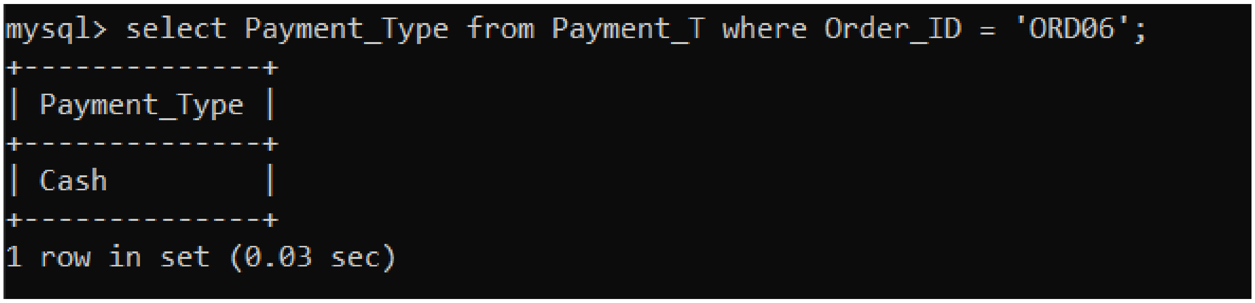
```
mysql> select distinct Payment_Type from Payment_T;
+--------------+
| Payment_Type |
+--------------+
| Cash         |
| Credit       |
| Coupon       |
+--------------+
3 rows in set (0.00 sec)
```

3. List in order the name and phone number of chef(s) who worked on the M T F ST 2:00 PM - 10:00 PM shift.



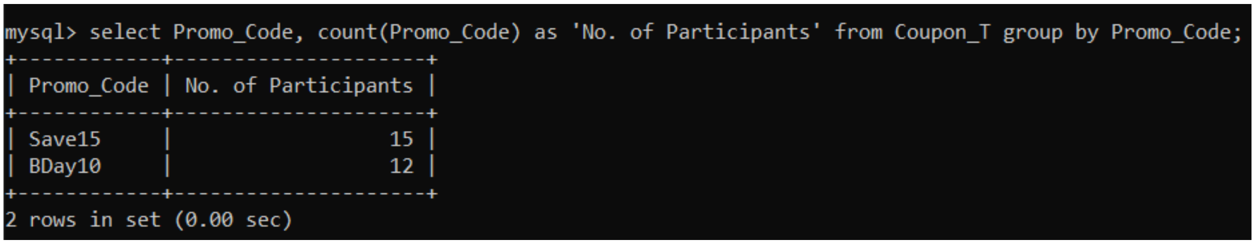
```
mysql> select Name, Phone_Number
    -> from Chef_T, Shift_T
    -> where Chef_T.Chef_ID = Shift_T.Chef_ID and Shift = 'M T F ST 2:00 PM - 10:00 PM'
    -> order by Name;
+---------------------+----------------+
| Name                | Phone_Number   |
+---------------------+----------------+
| Alexandra Cervantes | (576) 484-4998 |
| Ariel Mcgrath       | (388) 343-6415 |
+---------------------+----------------+
2 rows in set (0.04 sec)
```

4. Get the payment method used for order ORD06.



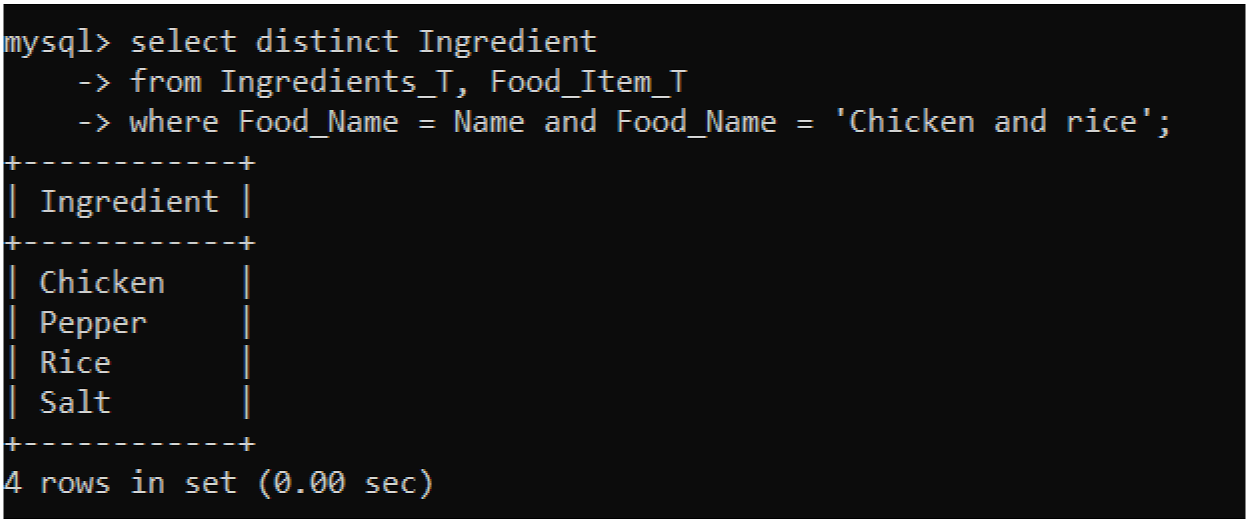
```
mysql> select Payment_Type from Payment_T where Order_ID = 'ORD06';
+--------------+
| Payment_Type |
+--------------+
| Cash         |
+--------------+
1 row in set (0.03 sec)
```

5. Find out how many people participated in each promotion.



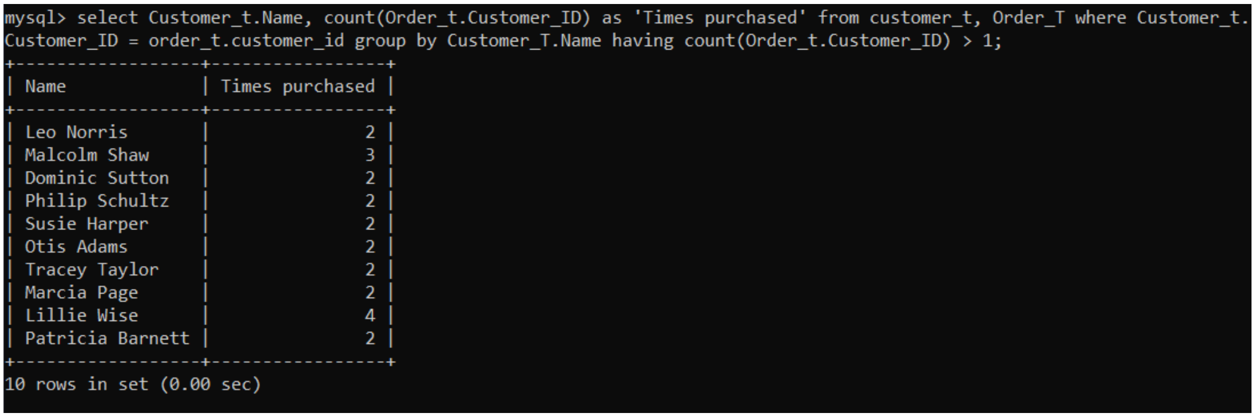
```
mysql> select Promo_Code, count(Promo_Code) as 'No. of Participants' from Coupon_T group by Promo_Code;
+------------+---------------------+
| Promo_Code | No. of Participants |
+------------+---------------------+
| Save15     |                  15 |
| BDay10     |                  12 |
+------------+---------------------+
2 rows in set (0.00 sec)
```

6. What are the ingredients in Chicken and Rice?



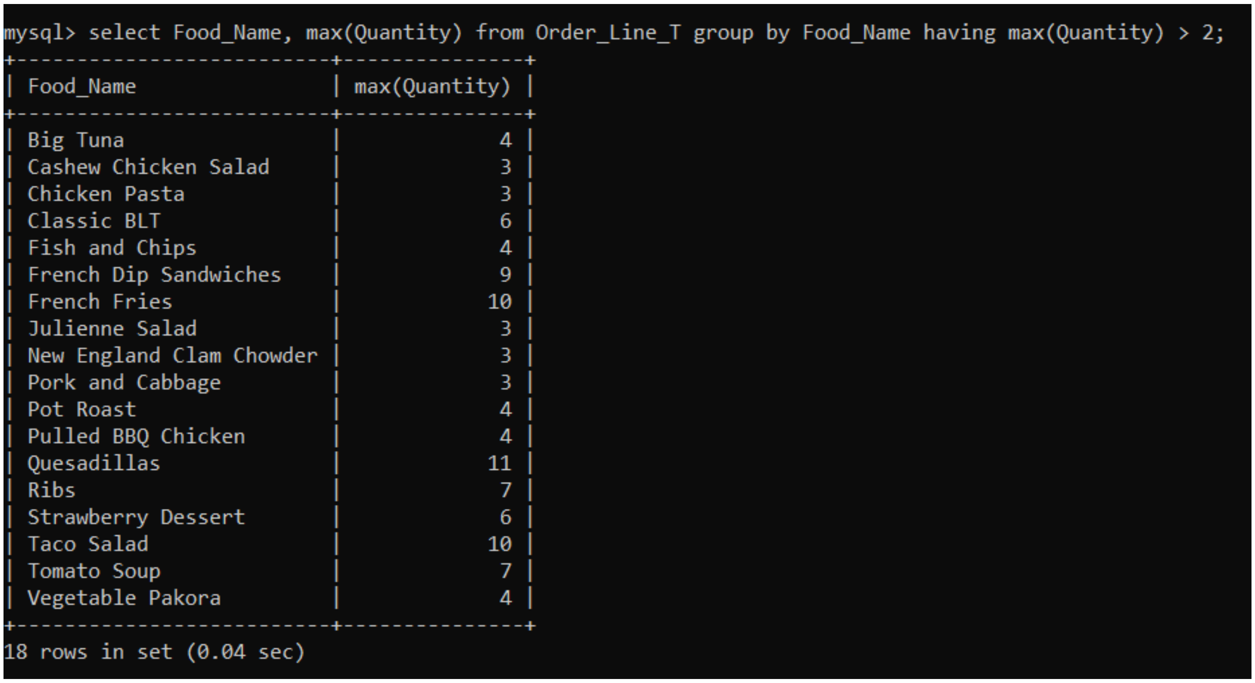
```
mysql> select distinct Ingredient
    -> from Ingredients_T, Food_Item_T
    -> where Food_Name = Name and Food_Name = 'Chicken and rice';
+------------+
| Ingredient |
+------------+
| Chicken    |
| Pepper     |
| Rice       |
| Salt       |
+------------+
4 rows in set (0.00 sec)
```

7. Get which customers purchased from the restaurant more than once.



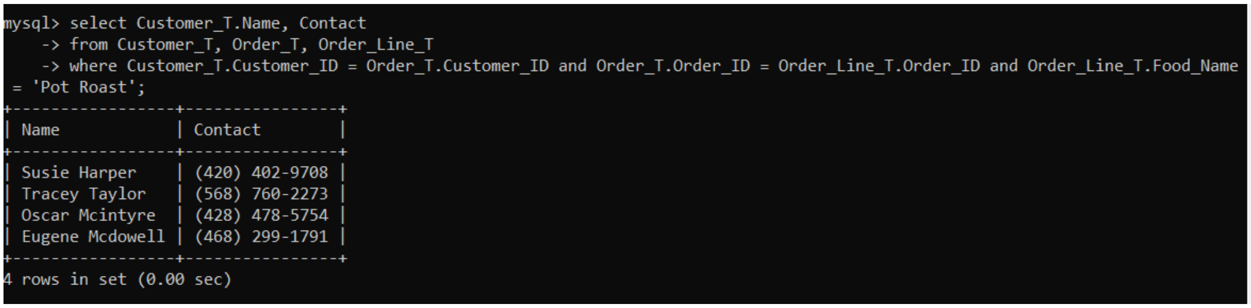
```
mysql> select Customer_t.Name, count(Order_t.Customer_ID) as 'Times purchased' from customer_t, Order_T where Customer_t.
Customer_ID = order_t.customer_id group by Customer_T.Name having count(Order_t.Customer_ID) > 1;
+------------------+-----------------+
| Name             | Times purchased |
+------------------+-----------------+
| Leo Norris       |               2 |
| Malcolm Shaw     |               3 |
| Dominic Sutton   |               2 |
| Philip Schultz   |               2 |
| Susie Harper     |               2 |
| Otis Adams       |               2 |
| Tracey Taylor    |               2 |
| Marcia Page      |               2 |
| Lillie Wise      |               4 |
| Patricia Barnett |               2 |
+------------------+-----------------+
10 rows in set (0.00 sec)
```

8. Find food items ordered in quantities over 2.



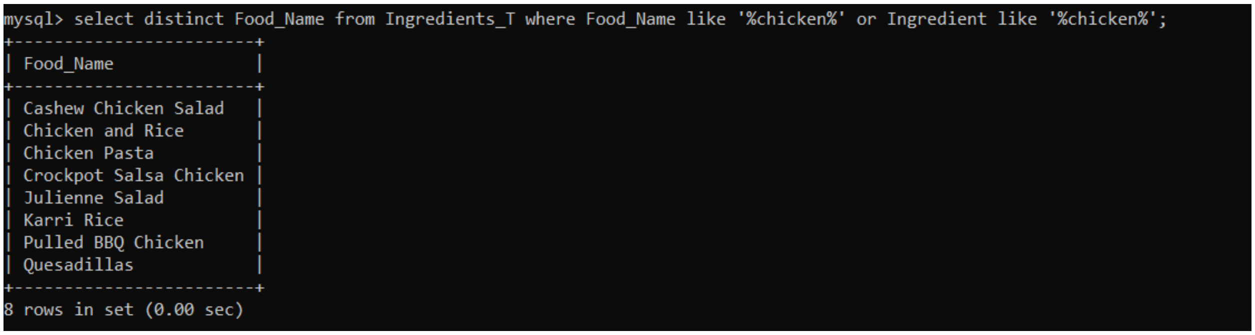
```
mysql> select Food_Name, max(Quantity) from Order_Line_T group by Food_Name having max(Quantity) > 2;
+--------------------------+---------------+
| Food_Name                | max(Quantity) |
+--------------------------+---------------+
| Big Tuna                 |             4 |
| Cashew Chicken Salad     |             3 |
| Chicken Pasta            |             3 |
| Classic BLT              |             6 |
| Fish and Chips           |             4 |
| French Dip Sandwiches    |             9 |
| French Fries             |            10 |
| Julienne Salad           |             3 |
| New England Clam Chowder |             3 |
| Pork and Cabbage         |             3 |
| Pot Roast                |             4 |
| Pulled BBQ Chicken       |             4 |
| Quesadillas              |            11 |
| Ribs                     |             7 |
| Strawberry Dessert       |             6 |
| Taco Salad               |            10 |
| Tomato Soup              |             7 |
| Vegetable Pakora         |             4 |
+--------------------------+---------------+
18 rows in set (0.04 sec)
```

9. Get the name and contact for all customers who ordered the Pot Roast.



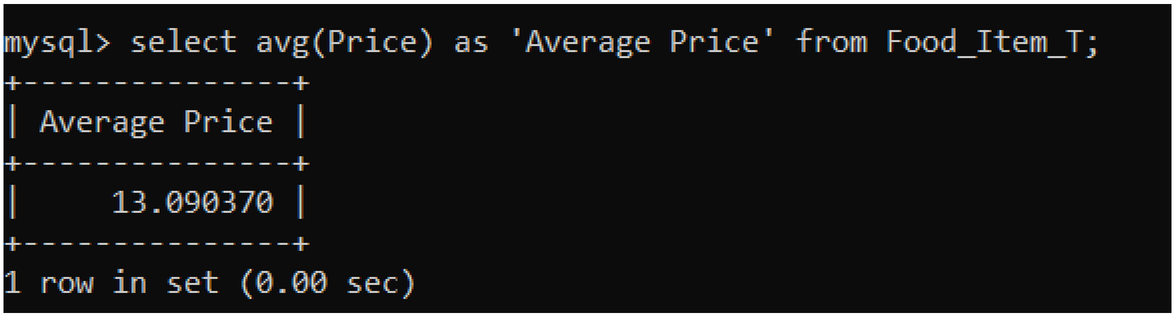
```
mysql> select Customer_T.Name, Contact
    -> from Customer_T, Order_T, Order_Line_T
    -> where Customer_T.Customer_ID = Order_T.Customer_ID and Order_T.Order_ID = Order_Line_T.Order_ID and Order_Line_T.Food_Name
 = 'Pot Roast';
+-----------------+----------------+
| Name            | Contact        |
+-----------------+----------------+
| Susie Harper    | (420) 402-9708 |
| Tracey Taylor   | (568) 760-2273 |
| Oscar Mcintyre  | (428) 478-5754 |
| Eugene Mcdowell | (468) 299-1791 |
+-----------------+----------------+
4 rows in set (0.00 sec)
```

10. Get a list of all the meals made with chicken.



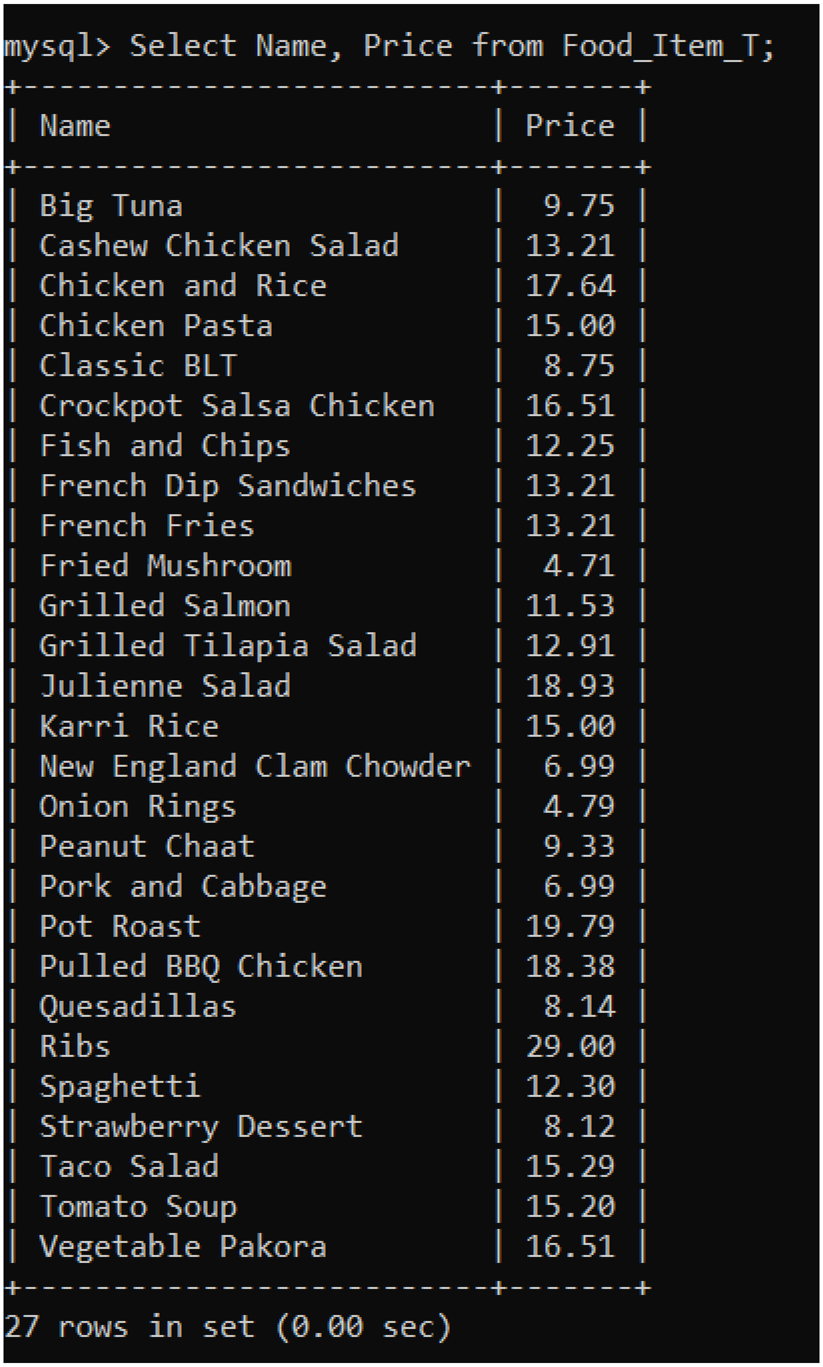
```
mysql> select distinct Food_Name from Ingredients_T where Food_Name like '%chicken%' or Ingredient like '%chicken%';
+-----------------------+
| Food_Name             |
+-----------------------+
| Cashew Chicken Salad  |
| Chicken and Rice      |
| Chicken Pasta         |
| Crockpot Salsa Chicken |
| Julienne Salad        |
| Karri Rice            |
| Pulled BBQ Chicken    |
| Quesadillas           |
+-----------------------+
8 rows in set (0.00 sec)
```

11. What is the average price of food items?



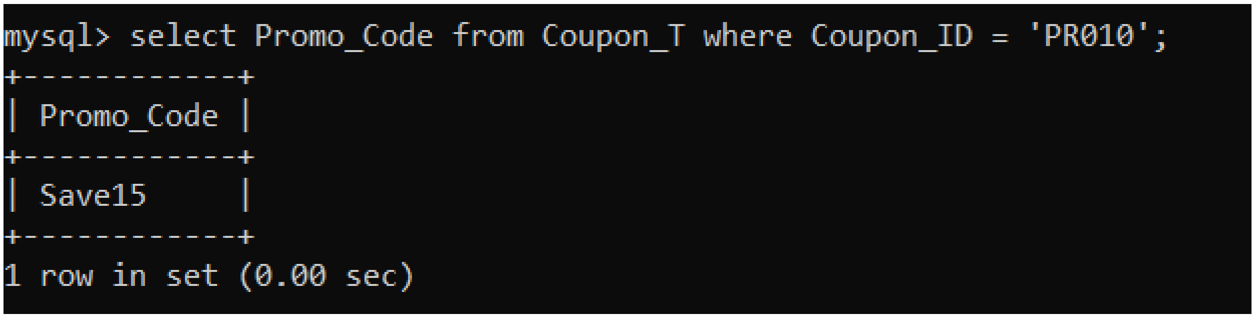
```
mysql> select avg(Price) as 'Average Price' from Food_Item_T;
+---------------+
| Average Price |
+---------------+
|     13.090370 |
+---------------+
1 row in set (0.00 sec)
```

12. Get the available dishes and their prices.



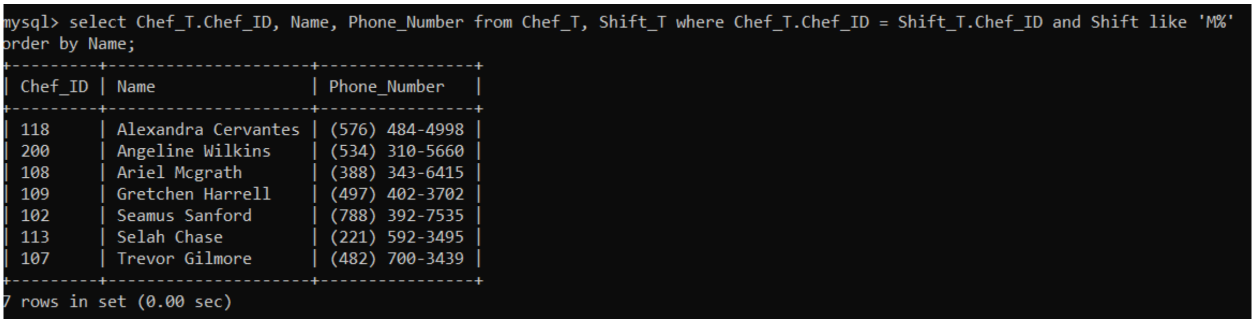
```
mysql> Select Name, Price from Food_Item_T;
+--------------------------+-------+
| Name                     | Price |
+--------------------------+-------+
| Big Tuna                 |  9.75 |
| Cashew Chicken Salad     | 13.21 |
| Chicken and Rice         | 17.64 |
| Chicken Pasta            | 15.00 |
| Classic BLT              |  8.75 |
| Crockpot Salsa Chicken   | 16.51 |
| Fish and Chips           | 12.25 |
| French Dip Sandwiches    | 13.21 |
| French Fries             | 13.21 |
| Fried Mushroom           |  4.71 |
| Grilled Salmon           | 11.53 |
| Grilled Tilapia Salad    | 12.91 |
| Julienne Salad           | 18.93 |
| Karri Rice               | 15.00 |
| New England Clam Chowder |  6.99 |
| Onion Rings              |  4.79 |
| Peanut Chaat             |  9.33 |
| Pork and Cabbage         |  6.99 |
| Pot Roast                | 19.79 |
| Pulled BBQ Chicken       | 18.38 |
| Quesadillas              |  8.14 |
| Ribs                     | 29.00 |
| Spaghetti                | 12.30 |
| Strawberry Dessert       |  8.12 |
| Taco Salad               | 15.29 |
| Tomato Soup              | 15.20 |
| Vegetable Pakora         | 16.51 |
+--------------------------+-------+
27 rows in set (0.00 sec)
```

13. Get the promocode of coupon PR010.



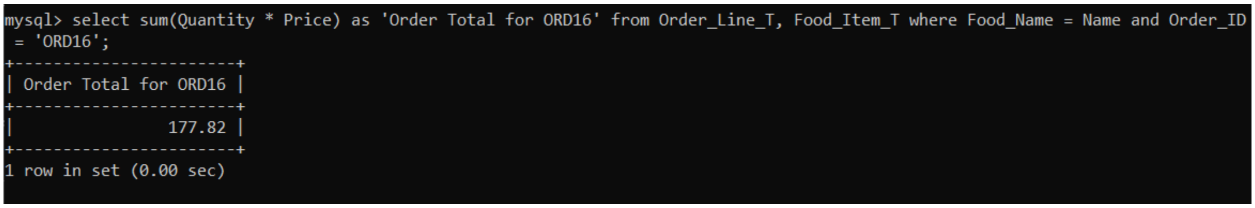
```
mysql> select Promo_Code from Coupon_T where Coupon_ID = 'PR010';
+------------+
| Promo_Code |
+------------+
| Save15     |
+------------+
1 row in set (0.00 sec)
```

14. Get a list of all chefs who start working on Monday by order of name.



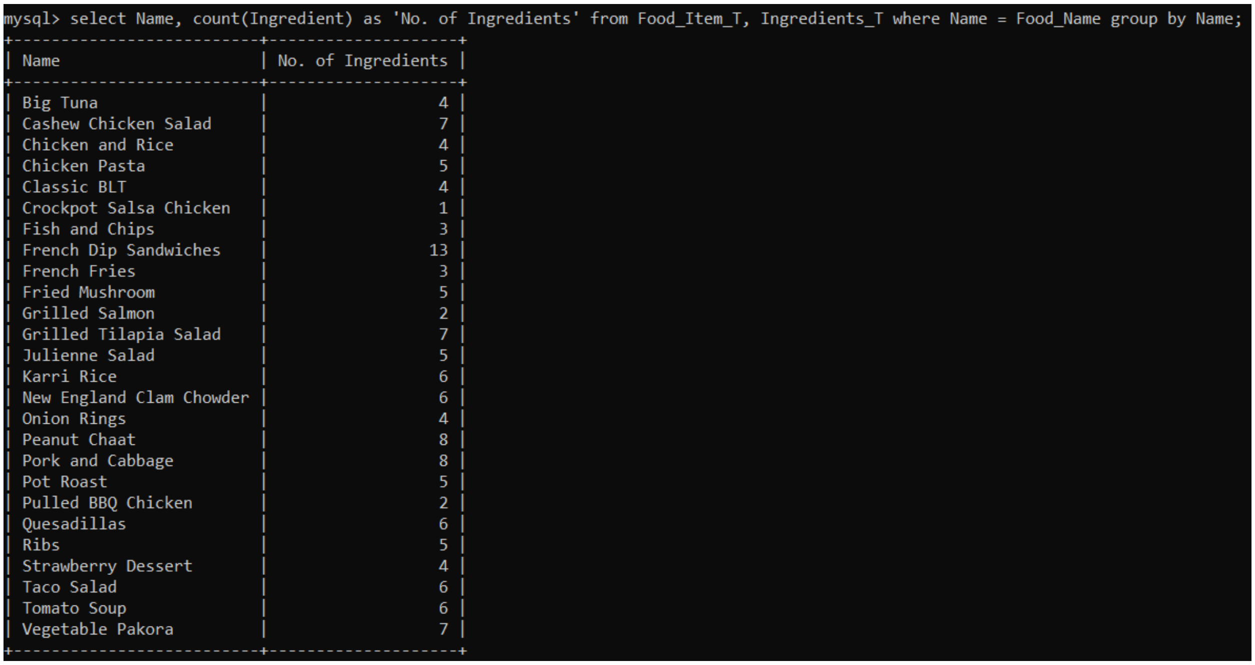
```
mysql> select Chef_T.Chef_ID, Name, Phone_Number from Chef_T, Shift_T where Chef_T.Chef_ID = Shift_T.Chef_ID and Shift like 'M%'
order by Name;
+---------+--------------------+----------------+
| Chef_ID | Name               | Phone_Number   |
+---------+--------------------+----------------+
| 118     | Alexandra Cervantes | (576) 484-4998 |
| 200     | Angeline Wilkins   | (534) 310-5660 |
| 108     | Ariel Mcgrath      | (388) 343-6415 |
| 109     | Gretchen Harrell   | (497) 402-3702 |
| 102     | Seamus Sanford     | (788) 392-7535 |
| 113     | Selah Chase        | (221) 592-3495 |
| 107     | Trevor Gilmore     | (482) 700-3439 |
+---------+--------------------+----------------+
7 rows in set (0.00 sec)
```

15. Get the order total for ORD16.



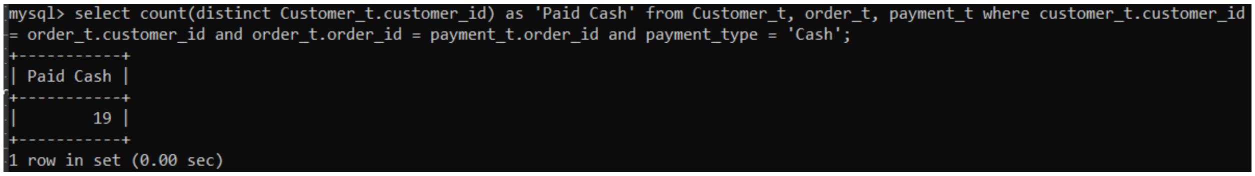
```
mysql> select sum(Quantity * Price) as 'Order Total for ORD16' from Order_Line_T, Food_Item_T where Food_Name = Name and Order_ID
 = 'ORD16';
+-----------------------+
| Order Total for ORD16 |
+-----------------------+
|                177.82 |
+-----------------------+
1 row in set (0.00 sec)
```

16. How many ingredients are in each food item?



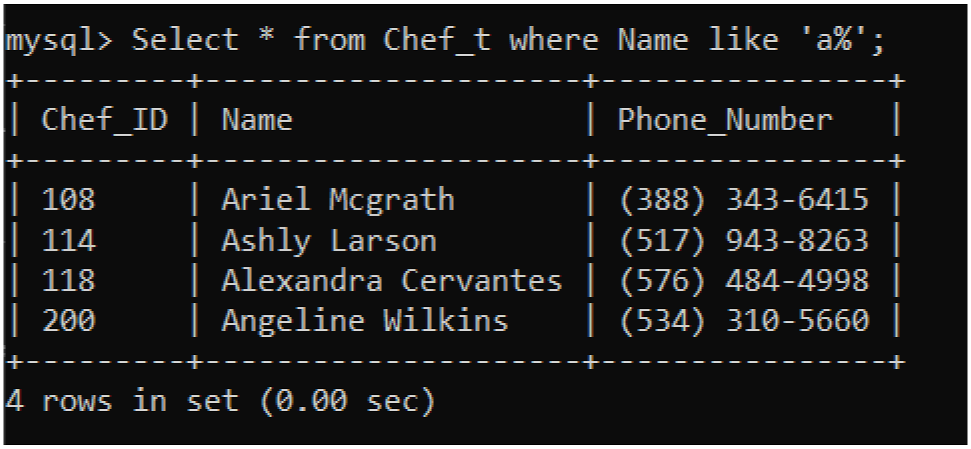
```
mysql> select Name, count(Ingredient) as 'No. of Ingredients' from Food_Item_T, Ingredients_T where Name = Food_Name group by Name;
+--------------------------+--------------------+
| Name                     | No. of Ingredients |
+--------------------------+--------------------+
| Big Tuna                 |                  4 |
| Cashew Chicken Salad     |                  7 |
| Chicken and Rice         |                  4 |
| Chicken Pasta            |                  5 |
| Classic BLT              |                  4 |
| Crockpot Salsa Chicken   |                  1 |
| Fish and Chips           |                  3 |
| French Dip Sandwiches    |                 13 |
| French Fries             |                  3 |
| Fried Mushroom           |                  5 |
| Grilled Salmon           |                  2 |
| Grilled Tilapia Salad    |                  7 |
| Julienne Salad           |                  5 |
| Karri Rice               |                  6 |
| New England Clam Chowder |                  6 |
| Onion Rings              |                  4 |
| Peanut Chaat             |                  8 |
| Pork and Cabbage         |                  8 |
| Pot Roast                |                  5 |
| Pulled BBQ Chicken       |                  2 |
| Quesadillas              |                  6 |
| Ribs                     |                  5 |
| Strawberry Dessert       |                  4 |
| Taco Salad               |                  6 |
| Tomato Soup              |                  6 |
| Vegetable Pakora         |                  7 |
+--------------------------+--------------------+
```

17. How many customers paid in cash?



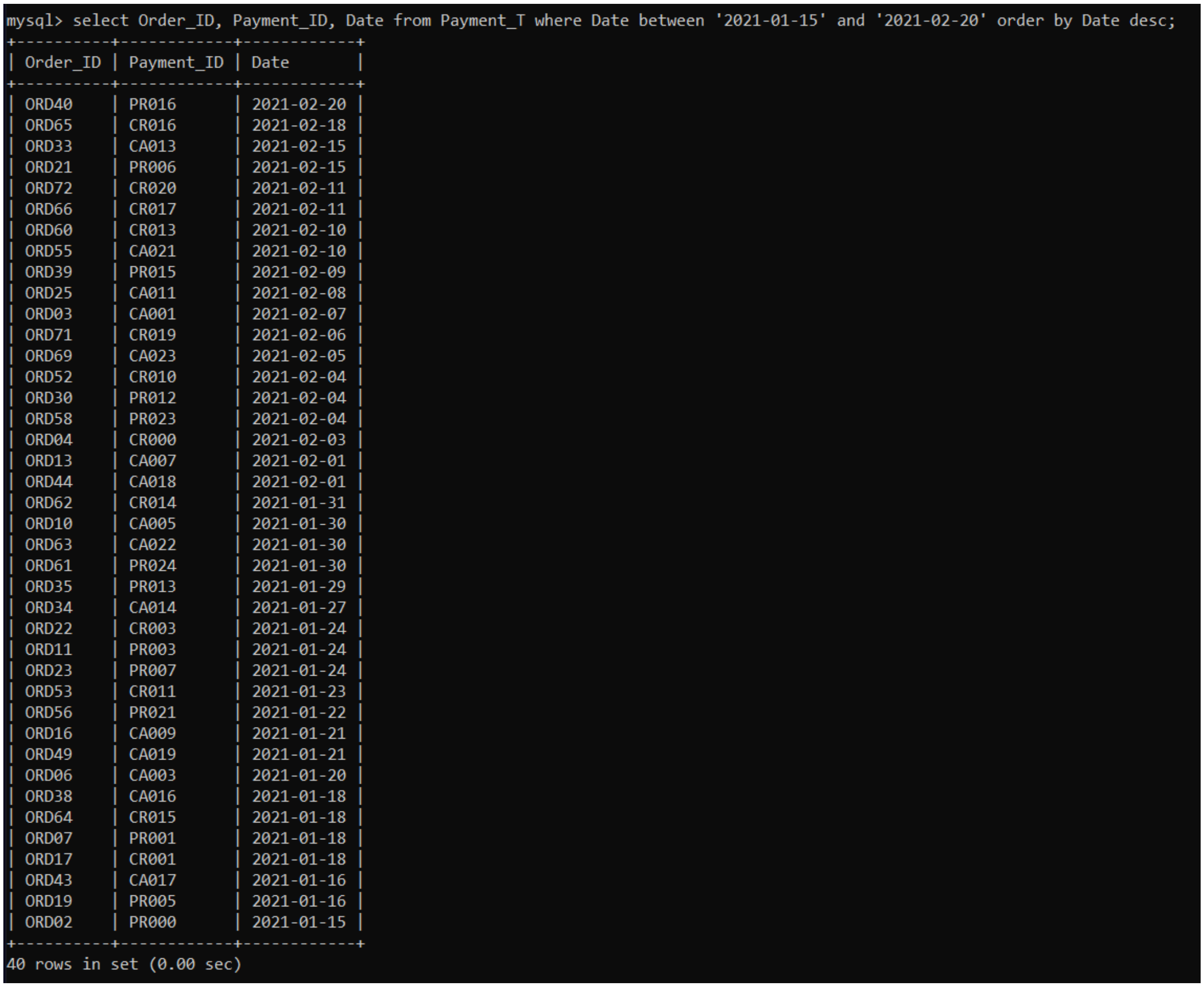
```
mysql> select count(distinct Customer_t.customer_id) as 'Paid Cash' from Customer_t, order_t, payment_t where customer_t.customer_id
= order_t.customer_id and order_t.order_id = payment_t.order_id and payment_type = 'Cash';
+-----------+
| Paid Cash |
+-----------+
|        19 |
+-----------+
1 row in set (0.00 sec)
```

18. Display all chefs with their names beginning with A.



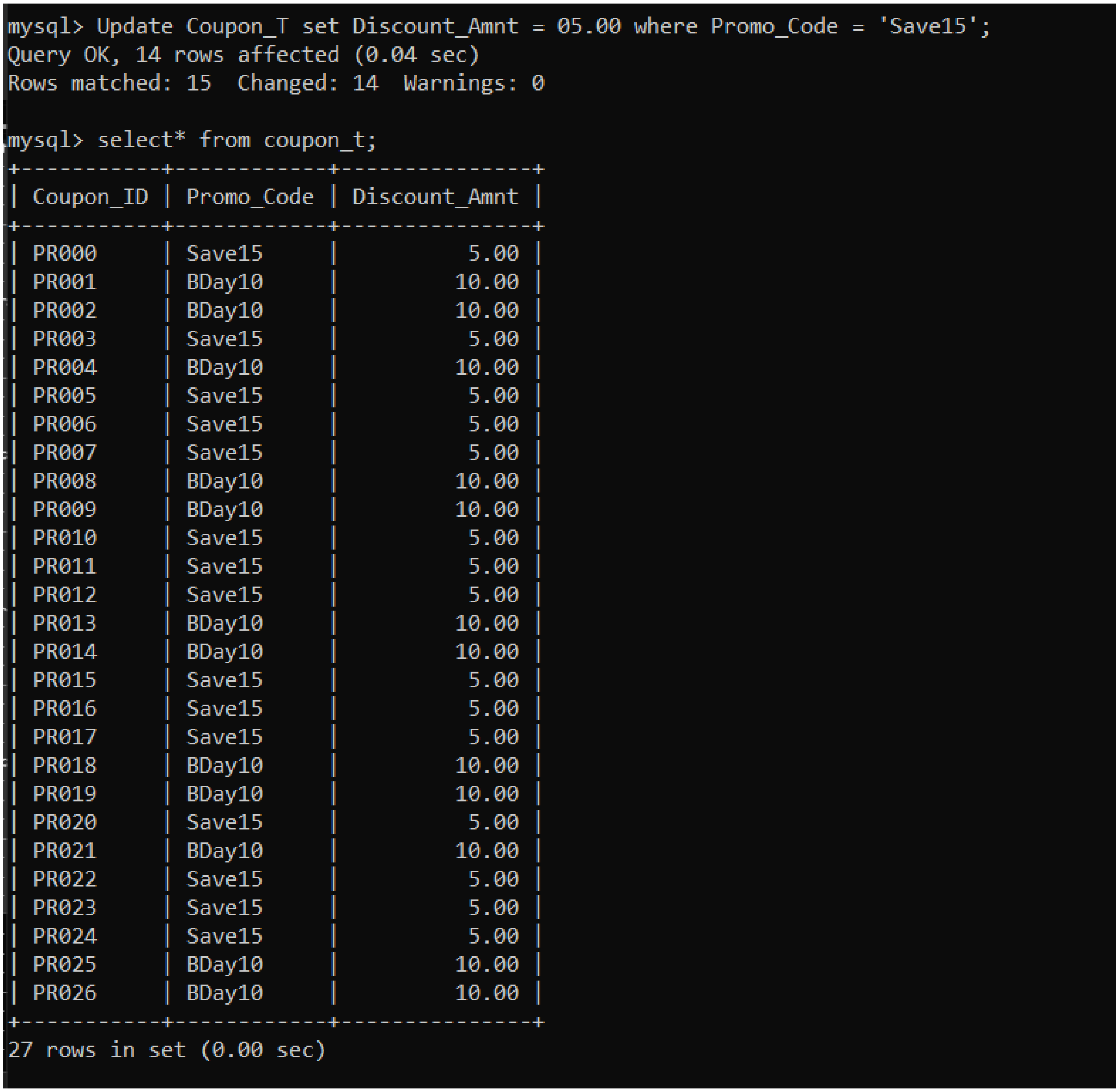
```
mysql> Select * from Chef_t where Name like 'a%';
+---------+----------------------+----------------+
| Chef_ID | Name                 | Phone_Number   |
+---------+----------------------+----------------+
| 108     | Ariel Mcgrath        | (388) 343-6415 |
| 114     | Ashly Larson         | (517) 943-8263 |
| 118     | Alexandra Cervantes  | (576) 484-4998 |
| 200     | Angeline Wilkins     | (534) 310-5660 |
+---------+----------------------+----------------+
4 rows in set (0.00 sec)
```

19. Get list of order ids and payment types where payment was made between 2021-01-15 and 2021-02-20. List in desc order by date.



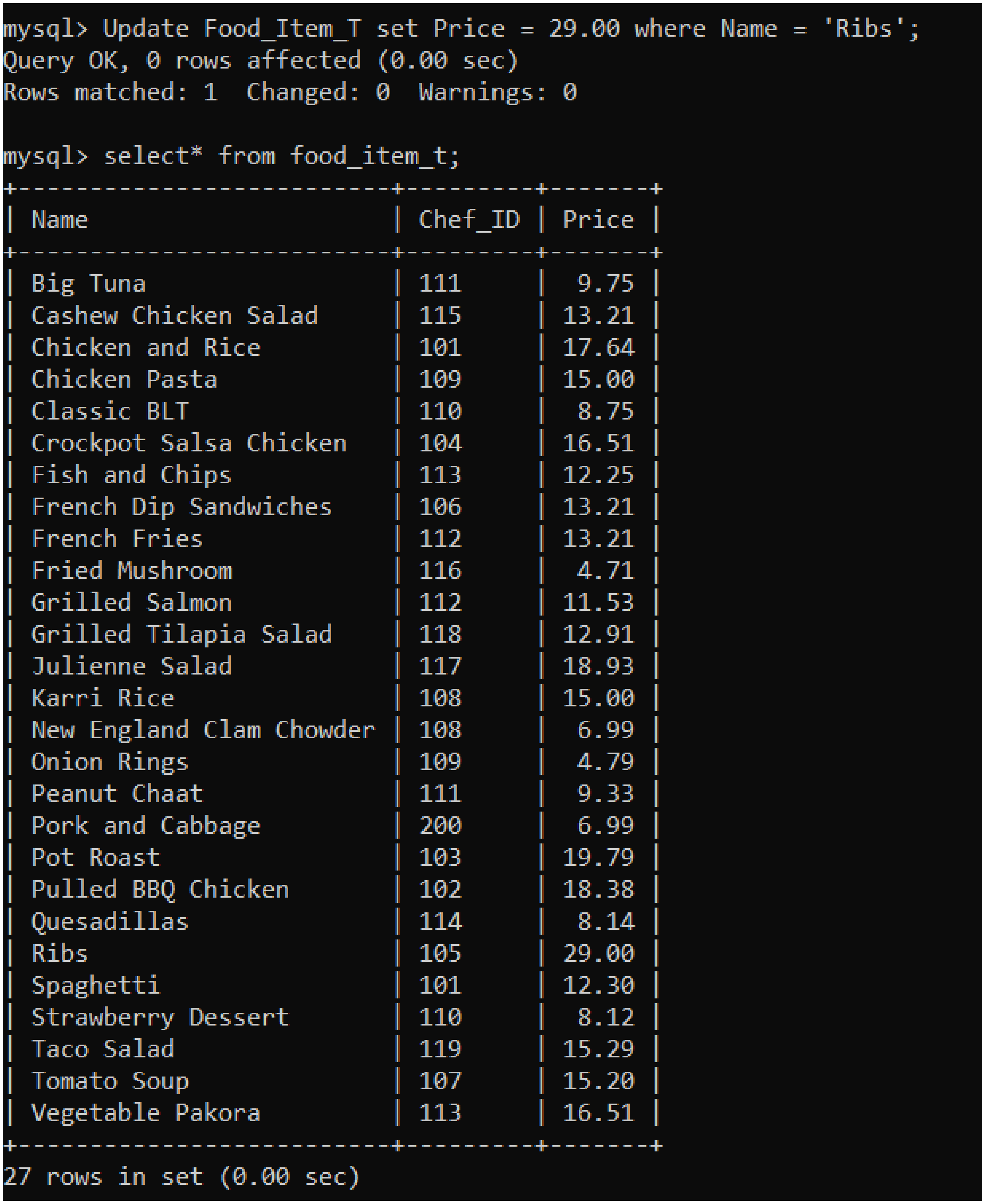
```
mysql> select Order_ID, Payment_ID, Date from Payment_T where Date between '2021-01-15' and '2021-02-20' order by Date desc;
+----------+------------+------------+
| Order_ID | Payment_ID | Date       |
+----------+------------+------------+
| ORD40    | PR016      | 2021-02-20 |
| ORD65    | CR016      | 2021-02-18 |
| ORD33    | CA013      | 2021-02-15 |
| ORD21    | PR006      | 2021-02-15 |
| ORD72    | CR020      | 2021-02-11 |
| ORD66    | CR017      | 2021-02-11 |
| ORD60    | CR013      | 2021-02-10 |
| ORD55    | CA021      | 2021-02-10 |
| ORD39    | PR015      | 2021-02-09 |
| ORD25    | CA011      | 2021-02-08 |
| ORD03    | CA001      | 2021-02-07 |
| ORD71    | CR019      | 2021-02-06 |
| ORD69    | CA023      | 2021-02-05 |
| ORD52    | CR010      | 2021-02-04 |
| ORD30    | PR012      | 2021-02-04 |
| ORD58    | PR023      | 2021-02-04 |
| ORD04    | CR000      | 2021-02-03 |
| ORD13    | CA007      | 2021-02-01 |
| ORD44    | CA018      | 2021-02-01 |
| ORD62    | CR014      | 2021-01-31 |
| ORD10    | CA005      | 2021-01-30 |
| ORD63    | CA022      | 2021-01-30 |
| ORD61    | PR024      | 2021-01-30 |
| ORD35    | PR013      | 2021-01-29 |
| ORD34    | CA014      | 2021-01-27 |
| ORD22    | CR003      | 2021-01-24 |
| ORD11    | PR003      | 2021-01-24 |
| ORD23    | PR007      | 2021-01-24 |
| ORD53    | CR011      | 2021-01-23 |
| ORD56    | PR021      | 2021-01-22 |
| ORD16    | CA009      | 2021-01-21 |
| ORD49    | CA019      | 2021-01-21 |
| ORD06    | CA003      | 2021-01-20 |
| ORD38    | CA016      | 2021-01-18 |
| ORD64    | CR015      | 2021-01-18 |
| ORD07    | PR001      | 2021-01-18 |
| ORD17    | CR001      | 2021-01-18 |
| ORD43    | CA017      | 2021-01-16 |
| ORD19    | PR005      | 2021-01-16 |
| ORD02    | PR000      | 2021-01-15 |
+----------+------------+------------+
40 rows in set (0.00 sec)
```

20. Change the discount amount of promocode Save15 to $5.00.



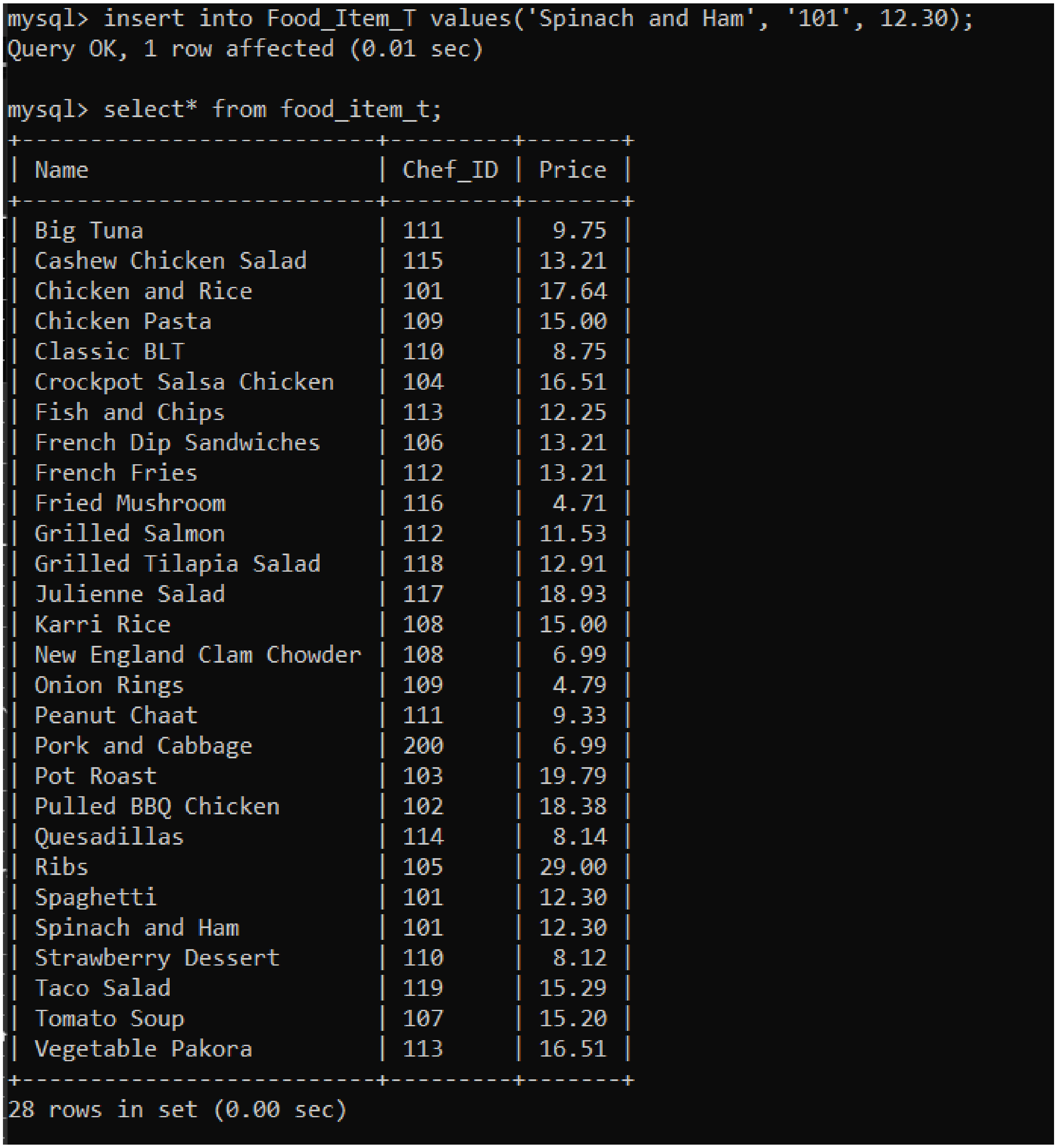
```
mysql> Update Coupon_T set Discount_Amnt = 05.00 where Promo_Code = 'Save15';
Query OK, 14 rows affected (0.04 sec)
Rows matched: 15  Changed: 14  Warnings: 0

mysql> select* from coupon_t;
+-----------+------------+---------------+
| Coupon_ID | Promo_Code | Discount_Amnt |
+-----------+------------+---------------+
| PR000     | Save15     |          5.00 |
| PR001     | BDay10     |         10.00 |
| PR002     | BDay10     |         10.00 |
| PR003     | Save15     |          5.00 |
| PR004     | BDay10     |         10.00 |
| PR005     | Save15     |          5.00 |
| PR006     | Save15     |          5.00 |
| PR007     | Save15     |          5.00 |
| PR008     | BDay10     |         10.00 |
| PR009     | BDay10     |         10.00 |
| PR010     | Save15     |          5.00 |
| PR011     | Save15     |          5.00 |
| PR012     | Save15     |          5.00 |
| PR013     | BDay10     |         10.00 |
| PR014     | BDay10     |         10.00 |
| PR015     | Save15     |          5.00 |
| PR016     | Save15     |          5.00 |
| PR017     | Save15     |          5.00 |
| PR018     | BDay10     |         10.00 |
| PR019     | BDay10     |         10.00 |
| PR020     | Save15     |          5.00 |
| PR021     | BDay10     |         10.00 |
| PR022     | Save15     |          5.00 |
| PR023     | Save15     |          5.00 |
| PR024     | Save15     |          5.00 |
| PR025     | BDay10     |         10.00 |
| PR026     | BDay10     |         10.00 |
+-----------+------------+---------------+
27 rows in set (0.00 sec)
```

21. Update the price of food item ribs to $29.00.



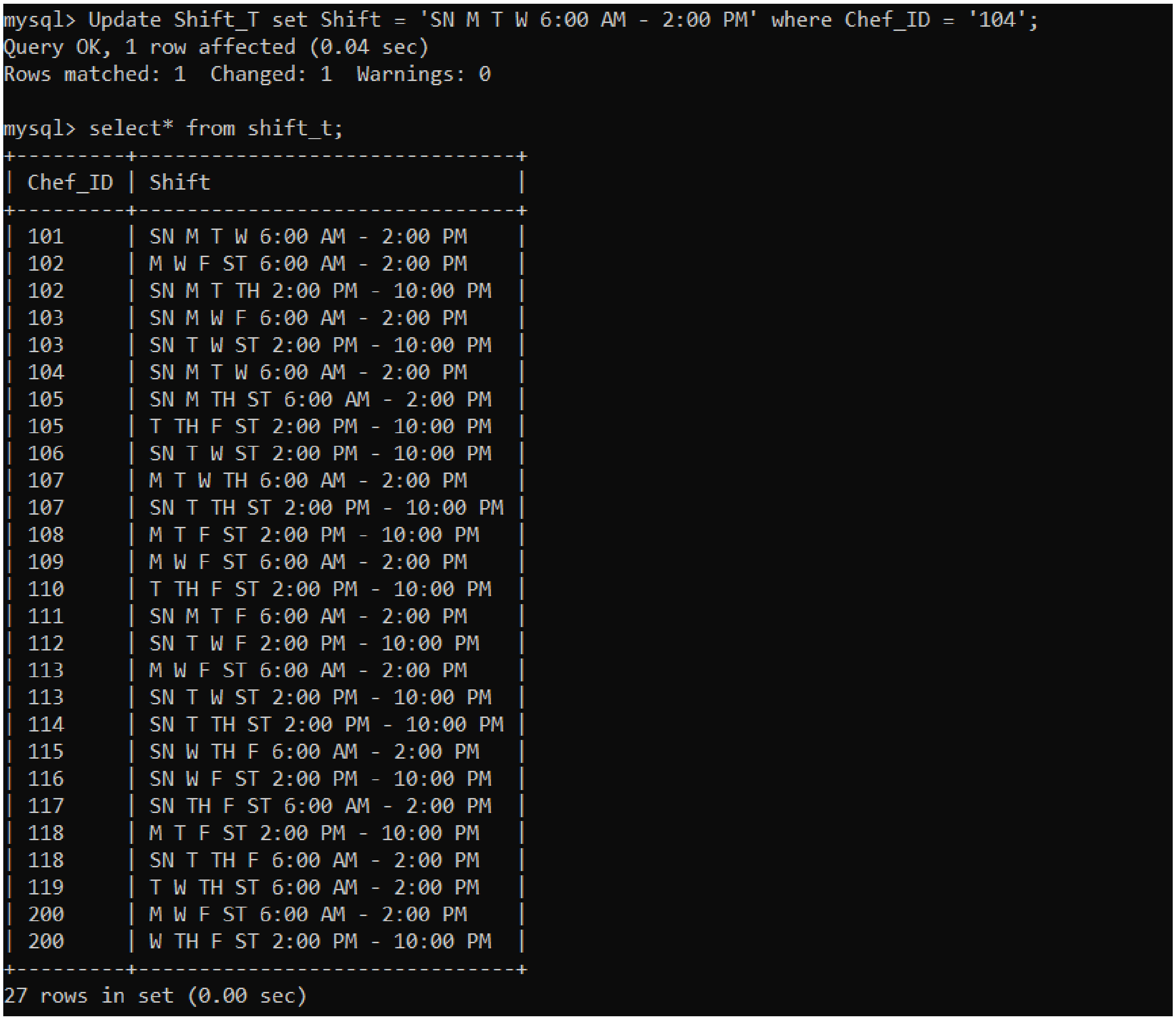
```
mysql> Update Food_Item_T set Price = 29.00 where Name = 'Ribs';
Query OK, 0 rows affected (0.00 sec)
Rows matched: 1  Changed: 0  Warnings: 0

mysql> select* from food_item_t;
+---------------------------+---------+-------+
| Name                      | Chef_ID | Price |
+---------------------------+---------+-------+
| Big Tuna                  | 111     |  9.75 |
| Cashew Chicken Salad      | 115     | 13.21 |
| Chicken and Rice          | 101     | 17.64 |
| Chicken Pasta             | 109     | 15.00 |
| Classic BLT               | 110     |  8.75 |
| Crockpot Salsa Chicken    | 104     | 16.51 |
| Fish and Chips            | 113     | 12.25 |
| French Dip Sandwiches     | 106     | 13.21 |
| French Fries              | 112     | 13.21 |
| Fried Mushroom            | 116     |  4.71 |
| Grilled Salmon            | 112     | 11.53 |
| Grilled Tilapia Salad     | 118     | 12.91 |
| Julienne Salad            | 117     | 18.93 |
| Karri Rice                | 108     | 15.00 |
| New England Clam Chowder  | 108     |  6.99 |
| Onion Rings               | 109     |  4.79 |
| Peanut Chaat              | 111     |  9.33 |
| Pork and Cabbage          | 200     |  6.99 |
| Pot Roast                 | 103     | 19.79 |
| Pulled BBQ Chicken        | 102     | 18.38 |
| Quesadillas               | 114     |  8.14 |
| Ribs                      | 105     | 29.00 |
| Spaghetti                 | 101     | 12.30 |
| Strawberry Dessert        | 110     |  8.12 |
| Taco Salad                | 119     | 15.29 |
| Tomato Soup               | 107     | 15.20 |
| Vegetable Pakora          | 113     | 16.51 |
+---------------------------+---------+-------+
27 rows in set (0.00 sec)
```

22. Add a new item Spinach and Ham to the available dishes made by chef 101 and costs $12.30.



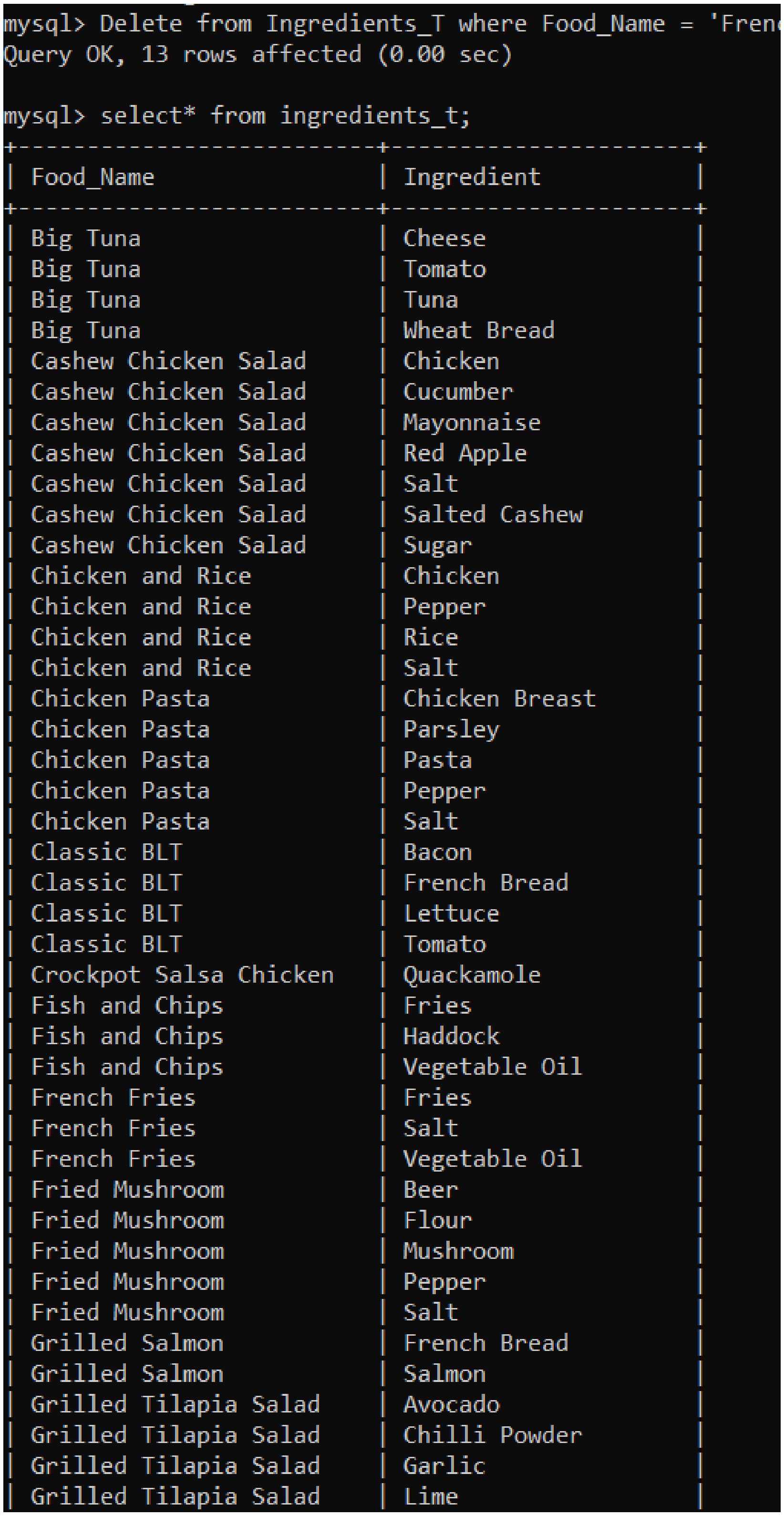
```
mysql> insert into Food_Item_T values('Spinach and Ham', '101', 12.30);
Query OK, 1 row affected (0.01 sec)

mysql> select* from food_item_t;
+--------------------------+---------+-------+
| Name                     | Chef_ID | Price |
+--------------------------+---------+-------+
| Big Tuna                 | 111     |  9.75 |
| Cashew Chicken Salad     | 115     | 13.21 |
| Chicken and Rice         | 101     | 17.64 |
| Chicken Pasta            | 109     | 15.00 |
| Classic BLT              | 110     |  8.75 |
| Crockpot Salsa Chicken   | 104     | 16.51 |
| Fish and Chips           | 113     | 12.25 |
| French Dip Sandwiches    | 106     | 13.21 |
| French Fries             | 112     | 13.21 |
| Fried Mushroom           | 116     |  4.71 |
| Grilled Salmon           | 112     | 11.53 |
| Grilled Tilapia Salad    | 118     | 12.91 |
| Julienne Salad           | 117     | 18.93 |
| Karri Rice               | 108     | 15.00 |
| New England Clam Chowder | 108     |  6.99 |
| Onion Rings              | 109     |  4.79 |
| Peanut Chaat             | 111     |  9.33 |
| Pork and Cabbage         | 200     |  6.99 |
| Pot Roast                | 103     | 19.79 |
| Pulled BBQ Chicken       | 102     | 18.38 |
| Quesadillas              | 114     |  8.14 |
| Ribs                     | 105     | 29.00 |
| Spaghetti                | 101     | 12.30 |
| Spinach and Ham          | 101     | 12.30 |
| Strawberry Dessert       | 110     |  8.12 |
| Taco Salad               | 119     | 15.29 |
| Tomato Soup              | 107     | 15.20 |
| Vegetable Pakora         | 113     | 16.51 |
+--------------------------+---------+-------+
28 rows in set (0.00 sec)
```

23. Change chef 104's shift to SN M T W 6:00 AM - 2:00 PM.

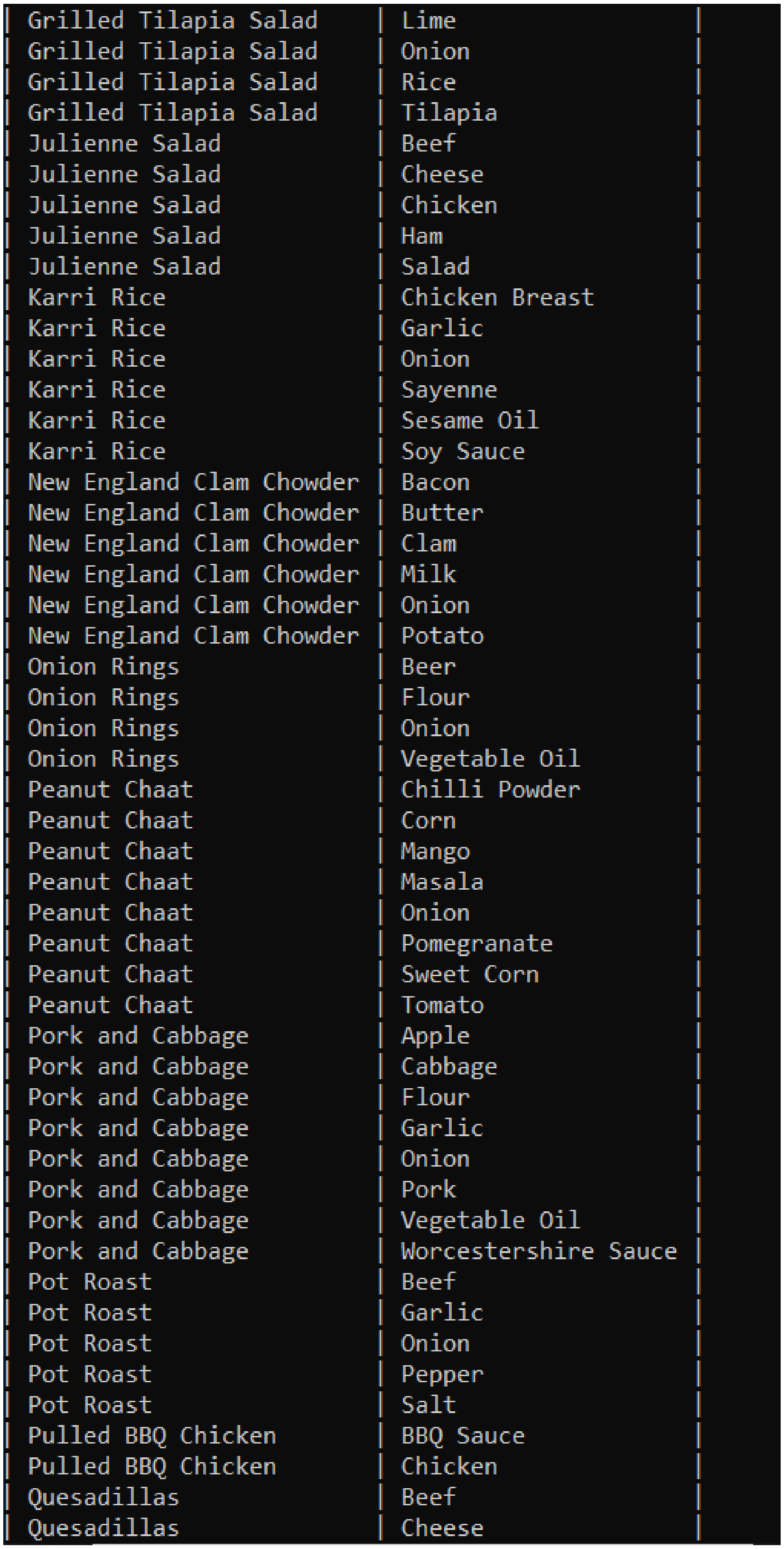


```
mysql> Update Shift_T set Shift = 'SN M T W 6:00 AM - 2:00 PM' where Chef_ID = '104';
Query OK, 1 row affected (0.04 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select* from shift_t;
+---------+-------------------------------+
| Chef_ID | Shift                         |
+---------+-------------------------------+
| 101     | SN M T W 6:00 AM - 2:00 PM    |
| 102     | M W F ST 6:00 AM - 2:00 PM    |
| 102     | SN M T TH 2:00 PM - 10:00 PM  |
| 103     | SN M W F 6:00 AM - 2:00 PM    |
| 103     | SN T W ST 2:00 PM - 10:00 PM  |
| 104     | SN M T W 6:00 AM - 2:00 PM    |
| 105     | SN M TH ST 6:00 AM - 2:00 PM  |
| 105     | T TH F ST 2:00 PM - 10:00 PM  |
| 106     | SN T W ST 2:00 PM - 10:00 PM  |
| 107     | M T W TH 6:00 AM - 2:00 PM    |
| 107     | SN T TH ST 2:00 PM - 10:00 PM |
| 108     | M T F ST 2:00 PM - 10:00 PM   |
| 109     | M W F ST 6:00 AM - 2:00 PM    |
| 110     | T TH F ST 2:00 PM - 10:00 PM  |
| 111     | SN M T F 6:00 AM - 2:00 PM    |
| 112     | SN T W F 2:00 PM - 10:00 PM   |
| 113     | M W F ST 6:00 AM - 2:00 PM    |
| 113     | SN T W ST 2:00 PM - 10:00 PM  |
| 114     | SN T TH ST 2:00 PM - 10:00 PM |
| 115     | SN W TH F 6:00 AM - 2:00 PM   |
| 116     | SN W F ST 2:00 PM - 10:00 PM  |
| 117     | SN TH F ST 6:00 AM - 2:00 PM  |
| 118     | M T F ST 2:00 PM - 10:00 PM   |
| 118     | SN T TH F 6:00 AM - 2:00 PM   |
| 119     | T W TH ST 6:00 AM - 2:00 PM   |
| 200     | M W F ST 6:00 AM - 2:00 PM    |
| 200     | W TH F ST 2:00 PM - 10:00 PM  |
+---------+-------------------------------+
27 rows in set (0.00 sec)
```

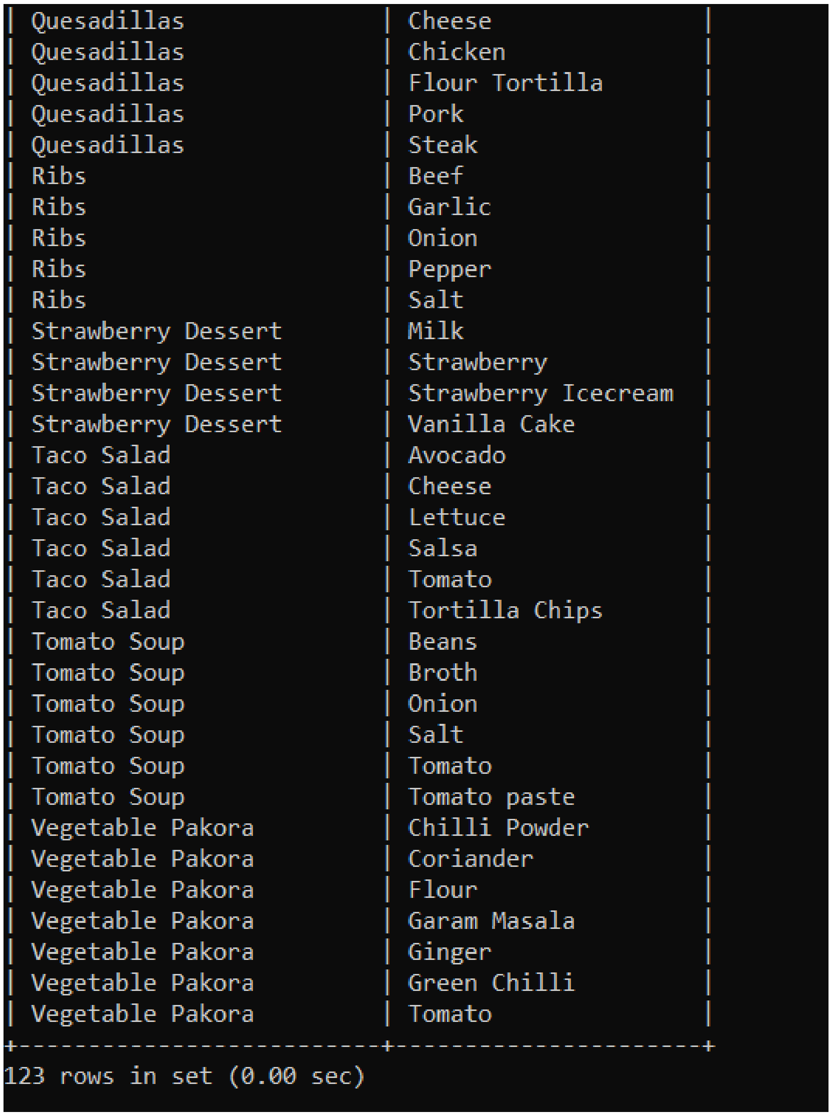
24. Delete French Dip Sandwiches from the ingredients table.



```
mysql> Delete from Ingredients_T where Food_Name = 'Fren
Query OK, 13 rows affected (0.00 sec)

mysql> select* from ingredients_t;
+--------------------------+----------------------+
| Food_Name                | Ingredient           |
+--------------------------+----------------------+
| Big Tuna                 | Cheese               |
| Big Tuna                 | Tomato               |
| Big Tuna                 | Tuna                 |
| Big Tuna                 | Wheat Bread          |
| Cashew Chicken Salad     | Chicken              |
| Cashew Chicken Salad     | Cucumber             |
| Cashew Chicken Salad     | Mayonnaise           |
| Cashew Chicken Salad     | Red Apple            |
| Cashew Chicken Salad     | Salt                 |
| Cashew Chicken Salad     | Salted Cashew        |
| Cashew Chicken Salad     | Sugar                |
| Chicken and Rice         | Chicken              |
| Chicken and Rice         | Pepper               |
| Chicken and Rice         | Rice                 |
| Chicken and Rice         | Salt                 |
| Chicken Pasta            | Chicken Breast       |
| Chicken Pasta            | Parsley              |
| Chicken Pasta            | Pasta                |
| Chicken Pasta            | Pepper               |
| Chicken Pasta            | Salt                 |
| Classic BLT              | Bacon                |
| Classic BLT              | French Bread         |
| Classic BLT              | Lettuce              |
| Classic BLT              | Tomato               |
| Crockpot Salsa Chicken   | Quackamole           |
| Fish and Chips           | Fries                |
| Fish and Chips           | Haddock              |
| Fish and Chips           | Vegetable Oil        |
| French Fries             | Fries                |
| French Fries             | Salt                 |
| French Fries             | Vegetable Oil        |
| Fried Mushroom           | Beer                 |
| Fried Mushroom           | Flour                |
| Fried Mushroom           | Mushroom             |
| Fried Mushroom           | Pepper               |
| Fried Mushroom           | Salt                 |
| Grilled Salmon           | French Bread         |
| Grilled Salmon           | Salmon               |
| Grilled Tilapia Salad    | Avocado              |
| Grilled Tilapia Salad    | Chilli Powder        |
| Grilled Tilapia Salad    | Garlic               |
| Grilled Tilapia Salad    | Lime                 |
```



```
| Grilled Tilapia Salad    | Lime                 |
| Grilled Tilapia Salad    | Onion                |
| Grilled Tilapia Salad    | Rice                 |
| Grilled Tilapia Salad    | Tilapia              |
| Julienne Salad           | Beef                 |
| Julienne Salad           | Cheese               |
| Julienne Salad           | Chicken              |
| Julienne Salad           | Ham                  |
| Julienne Salad           | Salad                |
| Karri Rice               | Chicken Breast       |
| Karri Rice               | Garlic               |
| Karri Rice               | Onion                |
| Karri Rice               | Sayenne              |
| Karri Rice               | Sesame Oil           |
| Karri Rice               | Soy Sauce            |
| New England Clam Chowder | Bacon                |
| New England Clam Chowder | Butter               |
| New England Clam Chowder | Clam                 |
| New England Clam Chowder | Milk                 |
| New England Clam Chowder | Onion                |
| New England Clam Chowder | Potato               |
| Onion Rings              | Beer                 |
| Onion Rings              | Flour                |
| Onion Rings              | Onion                |
| Onion Rings              | Vegetable Oil        |
| Peanut Chaat             | Chilli Powder        |
| Peanut Chaat             | Corn                 |
| Peanut Chaat             | Mango                |
| Peanut Chaat             | Masala               |
| Peanut Chaat             | Onion                |
| Peanut Chaat             | Pomegranate          |
| Peanut Chaat             | Sweet Corn           |
| Peanut Chaat             | Tomato               |
| Pork and Cabbage         | Apple                |
| Pork and Cabbage         | Cabbage              |
| Pork and Cabbage         | Flour                |
| Pork and Cabbage         | Garlic               |
| Pork and Cabbage         | Onion                |
| Pork and Cabbage         | Pork                 |
| Pork and Cabbage         | Vegetable Oil        |
| Pork and Cabbage         | Worcestershire Sauce |
| Pot Roast                | Beef                 |
| Pot Roast                | Garlic               |
| Pot Roast                | Onion                |
| Pot Roast                | Pepper               |
| Pot Roast                | Salt                 |
| Pulled BBQ Chicken       | BBQ Sauce            |
| Pulled BBQ Chicken       | Chicken              |
| Quesadillas              | Beef                 |
| Quesadillas              | Cheese               |
```



```
| Quesadillas              | Cheese               |
| Quesadillas              | Chicken              |
| Quesadillas              | Flour Tortilla       |
| Quesadillas              | Pork                 |
| Quesadillas              | Steak                |
| Ribs                     | Beef                 |
| Ribs                     | Garlic               |
| Ribs                     | Onion                |
| Ribs                     | Pepper               |
| Ribs                     | Salt                 |
| Strawberry Dessert       | Milk                 |
| Strawberry Dessert       | Strawberry           |
| Strawberry Dessert       | Strawberry Icecream  |
| Strawberry Dessert       | Vanilla Cake         |
| Taco Salad               | Avocado              |
| Taco Salad               | Cheese               |
| Taco Salad               | Lettuce              |
| Taco Salad               | Salsa                |
| Taco Salad               | Tomato               |
| Taco Salad               | Tortilla Chips       |
| Tomato Soup              | Beans                |
| Tomato Soup              | Broth                |
| Tomato Soup              | Onion                |
| Tomato Soup              | Salt                 |
| Tomato Soup              | Tomato               |
| Tomato Soup              | Tomato paste         |
| Vegetable Pakora         | Chilli Powder        |
| Vegetable Pakora         | Coriander            |
| Vegetable Pakora         | Flour                |
| Vegetable Pakora         | Garam Masala         |
| Vegetable Pakora         | Ginger               |
| Vegetable Pakora         | Green Chilli         |
| Vegetable Pakora         | Tomato               |
+--------------------------+----------------------+
123 rows in set (0.00 sec)
```

Restaurant_Def.sql

```
-- Group 2: The Urban Fence Restaurant Database
-- Group Members: Amaya Alisca, Deandra Martin, Alex Mayer, Cam Reeves
-- CMS 375
-- Due Date: 4/22/2021

-- TABLE DEFINITIONS --
DROP database IF EXISTS Restaurant;
CREATE database Restaurant;
USE Restaurant;

drop table if exists Customer_T;
-- Customer Table definition --
create table Customer_T (
   Customer_ID                    int not null,
   Name                           varchar(25),
   Contact                        char(14),
         constraint Customer_pk primary key (Customer_ID));

-------------------------------------------------------------


drop table if exists Chef_T;
-- Chef Table definition --
create table Chef_T (
   Chef_ID                        char(3) not null,
   Name                           varchar(25) not null,
   Phone_Number       char(14) not null,
   constraint Chef_pk primary key (Chef_ID));
```

```
----------------------------------------------------------------

drop table if exists Order_T;
-- Order table definition --
create table Order_T (
    Order_ID                char(5) not null,
    Customer_ID                         int not null,
    constraint Order_pk primary key (Order_ID),
    constraint Order_fk foreign key (Customer_ID) references Customer_T(Customer_ID));

----------------------------------------------------------------

drop table if exists Shift_T;
-- Shift table definition --
create table Shift_T (
    Chef_ID                             char(3) not null,
    Shift                   varchar(30) not null,
    constraint Shift_pk primary key (Chef_ID, Shift),
    constraint Shift_fk foreign key (Chef_ID) references Chef_T(Chef_ID));

----------------------------------------------------------------

drop table if exists Food_Item_T;
-- Food Item table definition --
create table Food_Item_T (
    Name                                varchar(35) not null,
    Chef_ID                             char(3) not null,
    Price                               Decimal(4,2),
    constraint Food_Item_pk primary key (Name),
    constraint Food_Item_fk1 foreign key (Chef_ID) references Chef_T(Chef_ID));
```

```
------------------------------------------------------------------

drop table if exists Ingredients_T;
-- Ingredients table definition --
create table Ingredients_T (
   Food_Name          varchar(35) not null,
   Ingredient         varchar(35) not null,
   constraint Ingredients_pk primary key (Food_Name, Ingredient),
   constraint Ingredients_fk foreign key (Food_Name) references Food_Item_T(Name));

------------------------------------------------------------------

drop table if exists Order_Line_T;
-- OrderLine table definition --
create table Order_Line_T (
   Order_ID           char(5) not null,
   Food_Name          varchar(35) not null,
   Quantity           int not null,
   constraint Order_Line_pk primary key (Order_ID, Food_Name),
   constraint Order_Line_fk1 foreign key (Order_ID) references Order_T(Order_ID),
   constraint Order_Line_fk2 foreign key (Food_Name) references Food_Item_T(Name));

------------------------------------------------------------------

drop table if exists Payment_T;
-- Payment table definition --
create table Payment_T (
   Payment_ID         char(5) not null,
   Order_ID           char(5) not null,
   Date                    date not null,
   Payment_Type       varchar(6),
                        check(Payment_Type in ("Cash", "Credit", "Coupon")), -- these are the only
accepted payment methods
   constraint Payment_pk primary key (Payment_ID),
   constraint Payment_fk foreign key (Order_ID) references Order_T(Order_ID));

--------------------------------------------------------------

drop table if exists Cash_T;
-- Sub class Cash table definition --
create table Cash_T (
   Cash_ID                 char(5) not null,
   IfChange                varchar(3) not null,
   constraint Cash_pk primary key (Cash_ID),
   constraint Cash_fk foreign key (Cash_ID) references Payment_T(Payment_ID));

--------------------------------------------------------------

drop table if exists Credit_T;
-- Sub class Credit table definition --
create table Credit_T (
   Card_ID                 char(5) not null,
   Card_Type              varchar(25) not null,
   Currency_Type       varchar(3) not null,
   constraint Credit_pk primary key (Card_ID),
   constraint Credit_fk foreign key (Card_ID) references Payment_T(Payment_ID));

--------------------------------------------------------------
```

```
drop table if exists Coupon_T;
-- Sub class Coupon table definition --
create table Coupon_T (
    Coupon_ID           char(5) not null,
    Promo_Code          char(6) not null,
    Discount_Amnt       Decimal(4,2),
    constraint Coupon_pk primary key (Coupon_ID),
    constraint Coupon_fk foreign key (Coupon_ID) references Payment_T(Payment_ID));


commit;
```

Restaurant_Pop.sql

```
-- Group 2: The Urban Fence Restaurant Database
-- Group Members: Amaya Alisca, Deandra Martin, Alex Mayer, Cam Reeves
-- CMS 375
-- Due Date: 4/22/2021

-- TABLE POPULATION --
Use Restaurant;

drop table if exists Customer_T;
-- Populating Customer table --
insert into Customer_T values(0001, 'Jo Mcbride', '(594) 674-4072');
insert into Customer_T values(0002, 'Andrea Valdez', '(391) 266-0714');
insert into Customer_T values(0003, 'Doug Clark', '(764) 350-7902');
insert into Customer_T values(0004, 'Leo Norris', '(212) 934-3867');
insert into Customer_T values(0005, 'Malcolm Shaw', '(235) 228-1557');
insert into Customer_T values(0006, 'Dominic Sutton', '(913) 314-8024');
insert into Customer_T values(0007, 'Philip Schultz', '(276) 686-3156');
insert into Customer_T values(0008, 'Susie Harper', '(420) 402-9708');
insert into Customer_T values(0009, 'Otis Adams', '(880) 658-2385');
insert into Customer_T values(0010, 'Tracey Taylor', '(568) 760-2273');
insert into Customer_T values(0011, 'Marcia Page', '(511) 379-1792');
insert into Customer_T values(0012, 'Harry Pittman', '(397) 805-5054');
insert into Customer_T values(0013, 'Tyrone Alvarado', '(207) 799-8756');
insert into Customer_T values(0014, 'Lillie Wise', '(309) 555-9908');
insert into Customer_T values(0015, 'Timmy Mendoza' '(833) 931-0546');
insert into Customer_T values(0016, 'Clinton Ellis', '(268) 501-1594');
insert into Customer_T values(0017, 'James Harris', '(508) 227-9047');
insert into Customer_T values(0018, 'Brittany Logan', '(598) 814-3535');
insert into Customer_T values(0019, 'Guadalupe Bridges', '(311) 586-0688');
insert into Customer_T values(0020, 'Sheldon Owen',         '(368) 536-3493');
insert into Customer_T values(0021, 'Patricia Barnett',       '(301) 504-9063');
insert into Customer_T values(0022, 'Abraham Rice',          '(378) 616-4024');
insert into Customer_T values(0023, 'Flora Walton','(589) 816-9732');
insert into Customer_T values(0024, 'Kim Rodgers','(342) 419-6741');
insert into Customer_T values(0025, 'Tricia Richardson', '(490) 876-8364');
insert into Customer_T values(0026, 'Brooklynn Escobar', '(681) 708-0726');
insert into Customer_T values(0027, 'Donte Mccullough',   '(307) 564-9182');
insert into Customer_T values(0028, 'Shyla Lozano', '(448) 807-5475');
insert into Customer_T values(0029, 'Winston Rocha', '(310) 512-7120');
insert into Customer_T values(0030, 'Uriah Perez',   '(371) 772-8133');
insert into Customer_T values(0031, 'Brian Gardner', '(753) 913-1717');
insert into Customer_T values(0032, 'Estrella Lamb', '(453) 642-1047');
insert into Customer_T values(0033, 'Oscar Mcintyre', '(428) 478-5754');
insert into Customer_T values(0034, 'Blake Golden', '(817) 282-1535');
insert into Customer_T values(0035, 'Kameron Rojas', '(733) 719-7630');
insert into Customer_T values(0036, 'Annalise Greer', '(965) 569-6794');
insert into Customer_T values(0037, 'Jabari Barajas', '(940) 747-5689');
insert into Customer_T values(0038, 'Dario Logan', '(318) 999-2526');
insert into Customer_T values(0039, 'Rory Wood', '(591) 937-1303');
insert into Customer_T values(0040, 'Guadalupe Armstrong', '(334) 931-7416');
insert into Customer_T values(0041, 'Preston Guerrero',      '(844) 816-0474');
insert into Customer_T values(0042, 'Josue Pierce',  '(279) 891-9093');
insert into Customer_T values(0043, 'Kailee Hendrix', '(790) 451-1773');
insert into Customer_T values(0044, 'Haylee Bonilla', '(605) 707-4514');
insert into Customer_T values(0045, 'Eugene Mcdowell', '(468) 299-1791');
insert into Customer_T values(0046, 'Kennedi Mckenzie', '(269) 669-2740');
insert into Customer_T values(0047, 'Erika Schmidt', '(737) 845-1759');
insert into Customer_T values(0048, 'Anastasia Khan', '(824) 299-7559');
insert into Customer_T values(0049, 'Holden Gilbert', '(642) 966-7041');
insert into Customer_T values(0050, 'Gianna Lee', '(885) 241-3331');
insert into Customer_T values(0051, 'Mareli Wilkerson',     '(255) 626-0572');
insert into Customer_T values(0052, 'Danika Sanford',       '(996) 514-3002');
insert into Customer_T values(0053, 'Eugene Allen',         '(225) 507-3888');
insert into Customer_T values(0054, 'Delaney Leach', '(227) 593-0856');
insert into Customer_T values(0055, 'Jase Cardenas', '(379) 408-7586');
insert into Customer_T values(0056, 'Natalie Shelton', '(331) 753-8566');
insert into Customer_T values(0057, 'Madalynn Rocha', '(403) 228-5924');
insert into Customer_T values(0058, 'Joaquin Rose','(608) 233-7758');
insert into Customer_T values(0059, 'Kaylie Harding', '(848) 887-2106');
insert into Customer_T values(0060, 'Anika Galloway', '(441) 910-1203');

---------------------------------------------------------------


drop table if exists Chef_T;
-- Populating Chef table --
insert into Chef_T values('101', 'Julio Hines', '(424) 885-1689');
insert into Chef_T values('102', 'Seamus Sanford', '(788) 392-7535');
insert into Chef_T values('103', 'Roger Conley', '(430) 205-1785');
insert into Chef_T values('104', 'Kaylyn Mahoney', '(972) 736-7901');
insert into Chef_T values('105', 'Sarai Stanton', '(832) 284-5421');
insert into Chef_T values('106', 'Grant Henderson', '(641) 789-4401');
insert into Chef_T values('107', 'Trevor Gilmore', '(482) 700-3439');
insert into Chef_T values('108', 'Ariel Mcgrath', '(388) 343-6415');
insert into Chef_T values('109', 'Gretchen Harrell', '(497) 402-3702');
insert into Chef_T values('110', 'Esteban Cox', '(298) 547-2898');
insert into Chef_T values('111', 'Rhett Wiggins', '(946) 936-3099');
insert into Chef_T values('112', 'Kai Hogan', '(235) 267-5687');
insert into Chef_T values('113', 'Selah Chase', '(221) 592-3495');
insert into Chef_T values('114', 'Ashly Larson', '(517) 943-8263');
insert into Chef_T values('115', 'Hugh Mullins', '(996) 243-4959');
insert into Chef_T values('116', 'Bruno Roman', '(564) 978-1953');
insert into Chef_T values('117', 'Sanaa Richards', '(334) 621-8361');
insert into Chef_T values('118', 'Alexandra Cervantes', '(576) 484-4998');
insert into Chef_T values('119', 'Dahlia Rowe', '(978) 237-7453');
insert into Chef_T values('200', 'Angeline Wilkins', '(534) 310-5660');

---------------------------------------------------------------

drop table if exists Order_T;
-- Populating Order table --
insert into Order_T values('ORD01', 0001);
insert into Order_T values('ORD02', 0002);
insert into Order_T values('ORD03', 0003);
insert into Order_T values('ORD04', 0004);
insert into Order_T values('ORD05', 0005);
insert into Order_T values('ORD06', 0006);
insert into Order_T values('ORD07', 0007);
insert into Order_T values('ORD08', 0008);
insert into Order_T values('ORD09', 0009);
insert into Order_T values('ORD10', 0010);
insert into Order_T values('ORD11', 0011);
insert into Order_T values('ORD12', 0012);
insert into Order_T values('ORD13', 0013);
insert into Order_T values('ORD14', 0014);
insert into Order_T values('ORD15', 0015);
insert into Order_T values('ORD16', 0016);
insert into Order_T values('ORD17', 0017);
insert into Order_T values('ORD18', 0018);
insert into Order_T values('ORD19', 0019);
insert into Order_T values('ORD20', 0020);
insert into Order_T values('ORD21', 0021);
insert into Order_T values('ORD22', 0022);
insert into Order_T values('ORD23', 0023);
insert into Order_T values('ORD24', 0024);
insert into Order_T values('ORD25', 0025);
insert into Order_T values('ORD26', 0011);
insert into Order_T values('ORD27', 0021);
insert into Order_T values('ORD28', 0014);
insert into Order_T values('ORD29', 0014);
insert into Order_T values('ORD30', 0014);
insert into Order_T values('ORD31', 0005);
insert into Order_T values('ORD32', 0004);
insert into Order_T values('ORD33', 0005);
insert into Order_T values('ORD34', 0006);
insert into Order_T values('ORD35', 0007);
insert into Order_T values('ORD36', 0008);
insert into Order_T values('ORD37', 0009);
insert into Order_T values('ORD38', 0010);
insert into Order_T values('ORD39', 0026);
insert into Order_T values('ORD40', 0027);
insert into Order_T values('ORD41', 0028);
insert into Order_T values('ORD42', 0029);
insert into Order_T values('ORD43', 0030);
insert into Order_T values('ORD44', 0031);
insert into Order_T values('ORD45', 0032);
insert into Order_T values('ORD46', 0033);
insert into Order_T values('ORD47', 0034);
insert into Order_T values('ORD48', 0035);
insert into Order_T values('ORD49', 0036);
insert into Order_T values('ORD50', 0037);
insert into Order_T values('ORD51', 0038);
insert into Order_T values('ORD52', 0039);
insert into Order_T values('ORD53', 0040);
insert into Order_T values('ORD54', 0041);
insert into Order_T values('ORD55', 0042);
insert into Order_T values('ORD56', 0043);
insert into Order_T values('ORD57', 0044);
insert into Order_T values('ORD58', 0045);
insert into Order_T values('ORD59', 0046);
insert into Order_T values('ORD60', 0047);
insert into Order_T values('ORD61', 0048);
insert into Order_T values('ORD62', 0049);
insert into Order_T values('ORD63', 0050);
insert into Order_T values('ORD64', 0051);
insert into Order_T values('ORD65', 0052);
insert into Order_T values('ORD66', 0053);
insert into Order_T values('ORD67', 0054);
insert into Order_T values('ORD68', 0055);
insert into Order_T values('ORD69', 0056);
insert into Order_T values('ORD70', 0057);
insert into Order_T values('ORD71', 0058);
insert into Order_T values('ORD72', 0059);
insert into Order_T values('ORD73', 0060);

------------------------------------------------------------------------

drop table if exists Shift_T;
-- Populating Shift table --
```

```
insert into Shift_T values('101', 'SN M T W 6:00 AM - 2:00 PM');
insert into Shift_T values('102', 'SN M T TH 2:00 PM - 10:00 PM');
insert into Shift_T values('102', 'M W F ST 6:00 AM - 2:00 PM');
insert into Shift_T values('103', 'SN T W ST 2:00 PM - 10:00 PM');
insert into Shift_T values('103', 'SN M W F 6:00 AM - 2:00 PM');
insert into Shift_T values('104', 'SN M W ST 2:00 PM - 10:00 PM');
insert into Shift_T values('105', 'SN M TH ST 6:00 AM - 2:00 PM');
insert into Shift_T values('105', 'T TH F ST 2:00 PM - 10:00 PM');
insert into Shift_T values('106', 'SN T W ST 2:00 PM - 10:00 PM');
insert into Shift_T values('107', 'SN T TH ST 2:00 PM - 10:00 PM');
insert into Shift_T values('107', 'M T W TH 6:00 AM - 2:00 PM');
insert into Shift_T values('108', 'M T F ST 2:00 PM - 10:00 PM');
insert into Shift_T values('109', 'M W F ST 6:00 AM - 2:00 PM');
insert into Shift_T values('110', 'T TH F ST 2:00 PM - 10:00 PM');
insert into Shift_T values('111', 'SN M T F 6:00 AM - 2:00 PM');
insert into Shift_T values('112', 'SN T W F 2:00 PM - 10:00 PM');
insert into Shift_T values('113', 'SN T W ST 2:00 PM - 10:00 PM');
insert into Shift_T values('113', 'M W F ST 6:00 AM - 2:00 PM');
insert into Shift_T values('114', 'SN T TH ST 2:00 PM - 10:00 PM');
insert into Shift_T values('115', 'SN W TH F 6:00 AM - 2:00 PM');
insert into Shift_T values('116', 'SN W F ST 2:00 PM - 10:00 PM');
insert into Shift_T values('117', 'SN TH F ST 6:00 AM - 2:00 PM');
insert into Shift_T values('118', 'M T F ST 2:00 PM - 10:00 PM');
insert into Shift_T values('118', 'SN T TH F 6:00 AM - 2:00 PM');
insert into Shift_T values('119', 'T W TH ST 6:00 AM - 2:00 PM');
insert into Shift_T values('200', 'M W F ST 6:00 AM - 2:00 PM');
insert into Shift_T values('200', 'W TH F ST 2:00 PM - 10:00 PM');
```

```
drop table if exists Food_Item_T;
-- Populating Food Item table --
insert into Food_Item_T values('Chicken and Rice', '101', 17.64);
insert into Food_Item_T values('Pulled BBQ Chicken', '102', 18.38);
insert into Food_Item_T values('Pot Roast', '103', 19.79);
insert into Food_Item_T values('Crockpot Salsa Chicken', '104', 16.51);
insert into Food_Item_T values('Ribs', '105', 20.83);
insert into Food_Item_T values('French Dip Sandwiches', '106', 13.21);
insert into Food_Item_T values('Tomato Soup', '107', 15.20);
insert into Food_Item_T values('Karri Rice', '108', 15.00);
insert into Food_Item_T values('Chicken Pasta', '109', 15.00);
insert into Food_Item_T values('Strawberry Dessert', '110', 08.12);
insert into Food_Item_T values('Peanut Chaat', '111', 09.33);
insert into Food_Item_T values('French Fries', '112', 13.21);
insert into Food_Item_T values('Vegetable Pakora', '113', 16.51);
insert into Food_Item_T values('Quesadillas', '114', 08.14);
insert into Food_Item_T values('Cashew Chicken Salad', '115', 13.21);
insert into Food_Item_T values('Fried Mushroom', '116', 04.71);
insert into Food_Item_T values('Julienne Salad', '117', 18.93);
insert into Food_Item_T values('Grilled Tilapia Salad', '118', 12.91);
insert into Food_Item_T values('Taco Salad', '119', 15.29);
insert into Food_Item_T values('Pork and Cabbage', '200', 06.99);
insert into Food_Item_T values('New England Clam Chowder', '108', 06.99);
insert into Food_Item_T values('Onion Rings', '109', 04.79);
insert into Food_Item_T values('Classic BLT', '110', 08.75);
insert into Food_Item_T values('Big Tuna', '111', 09.75);
insert into Food_Item_T values('Grilled Salmon', '112', 11.53);
insert into Food_Item_T values('Fish and Chips', '113', 12.25);


------------------------------------------------------------------
```

```
drop table if exists Ingredients_T;
-- Populating Ingredients table --
insert into Ingredients_T values('Chicken and Rice', 'Chicken');
insert into Ingredients_T values('Chicken and Rice', 'Rice');
insert into Ingredients_T values('Chicken and Rice', 'Salt');
insert into Ingredients_T values('Chicken and Rice', 'Pepper');
insert into Ingredients_T values('Pulled BBQ Chicken', 'Chicken');
insert into Ingredients_T values('Pulled BBQ Chicken', 'BBQ Sauce');
insert into Ingredients_T values('Crockpot Salsa Chicken', 'Quackamole');
insert into Ingredients_T values('French Dip Sandwiches', 'Bread');
insert into Ingredients_T values('French Dip Sandwiches', 'Dip');
insert into Ingredients_T values('French Dip Sandwiches', 'Vinegar');
insert into Ingredients_T values('French Dip Sandwiches', 'Rum');
insert into Ingredients_T values('French Dip Sandwiches', 'Steak');
insert into Ingredients_T values('French Dip Sandwiches', 'Parsley');
insert into Ingredients_T values('French Dip Sandwiches', 'Cheese');
insert into Ingredients_T values('French Dip Sandwiches', 'Onion');
insert into Ingredients_T values('French Dip Sandwiches', 'Garlic');
insert into Ingredients_T values('French Dip Sandwiches', 'Rosemary');
insert into Ingredients_T values('French Dip Sandwiches', 'Worcestershire Sauce');
insert into Ingredients_T values('French Dip Sandwiches', 'Salt');
insert into Ingredients_T values('French Dip Sandwiches', 'Pepper');
insert into Ingredients_T values('Ribs', 'Salt');
insert into Ingredients_T values('Ribs', 'Pepper');
insert into Ingredients_T values('Ribs', 'Onion');
insert into Ingredients_T values('Ribs', 'Garlic');
insert into Ingredients_T values('Ribs', 'Beef');
insert into Ingredients_T values('Pot Roast', 'Salt');
insert into Ingredients_T values('Pot Roast', 'Pepper');
insert into Ingredients_T values('Pot Roast', 'Onion');
insert into Ingredients_T values('Pot Roast', 'Garlic');
insert into Ingredients_T values('Pot Roast', 'Beef');
insert into Ingredients_T values('Karri Rice', 'Onion');
insert into Ingredients_T values('Karri Rice', 'Garlic');
insert into Ingredients_T values('Karri Rice', 'Sesame Oil');
insert into Ingredients_T values('Karri Rice', 'Sayenne');
insert into Ingredients_T values('Karri Rice', 'Chicken Breast');
insert into Ingredients_T values('Karri Rice', 'Soy Sauce');
insert into Ingredients_T values('Tomato Soup', 'Onion');
insert into Ingredients_T values('Tomato Soup', 'Salt');
insert into Ingredients_T values('Tomato Soup', 'Broth');
insert into Ingredients_T values('Tomato Soup', 'Tomato');
insert into Ingredients_T values('Tomato Soup', 'Tomato paste');
insert into Ingredients_T values('Tomato Soup', 'Beans');
insert into Ingredients_T values('Chicken Pasta', 'Chicken Breast');
insert into Ingredients_T values('Chicken Pasta', 'Pasta');
insert into Ingredients_T values('Chicken Pasta', 'Parsley');
insert into Ingredients_T values('Chicken Pasta', 'Salt');
insert into Ingredients_T values('Chicken Pasta', 'Pepper');
insert into Ingredients_T values('Strawberry Dessert', 'Strawberry');
insert into Ingredients_T values('Strawberry Dessert', 'Vanilla Cake');
insert into Ingredients_T values('Strawberry Dessert', 'Strawberry Icecream');
insert into Ingredients_T values('Strawberry Dessert', 'Milk');
insert into Ingredients_T values('Peanut Chaat', 'Chilli Powder');
insert into Ingredients_T values('Peanut Chaat', 'Masala');
insert into Ingredients_T values('Peanut Chaat', 'Onion');
insert into Ingredients_T values('Peanut Chaat', 'Tomato');
insert into Ingredients_T values('Peanut Chaat', 'Corn');
insert into Ingredients_T values('Peanut Chaat', 'Mango');
insert into Ingredients_T values('Peanut Chaat', 'Pomegranate');
insert into Ingredients_T values('Peanut Chaat', 'Sweet Corn');
insert into Ingredients_T values('French Fries', 'Fries');
insert into Ingredients_T values('French Fries', 'Salt');
insert into Ingredients_T values('French Fries', 'Vegetable Oil');
insert into Ingredients_T values('Vegetable Pakora', 'Green Chilli');
insert into Ingredients_T values('Vegetable Pakora', 'Chilli Powder');
insert into Ingredients_T values('Vegetable Pakora', 'Garam Masala');
insert into Ingredients_T values('Vegetable Pakora', 'Coriander');
insert into Ingredients_T values('Vegetable Pakora', 'Flour');
insert into Ingredients_T values('Vegetable Pakora', 'Tomato');
insert into Ingredients_T values('Vegetable Pakora', 'Ginger');
insert into Ingredients_T values('Quesadillas', 'Flour Tortilla');
insert into Ingredients_T values('Quesadillas', 'Cheese');
insert into Ingredients_T values('Quesadillas', 'Chicken');
insert into Ingredients_T values('Quesadillas', 'Steak');
insert into Ingredients_T values('Quesadillas', 'Beef');
insert into Ingredients_T values('Quesadillas', 'Pork');
insert into Ingredients_T values('Cashew Chicken Salad', 'Chicken');
insert into Ingredients_T values('Cashew Chicken Salad', 'Salted Cashew');
insert into Ingredients_T values('Cashew Chicken Salad', 'Red Apple');
insert into Ingredients_T values('Cashew Chicken Salad', 'Cucumber');
insert into Ingredients_T values('Cashew Chicken Salad', 'Mayonnaise');
insert into Ingredients_T values('Cashew Chicken Salad', 'Sugar');
insert into Ingredients_T values('Cashew Chicken Salad', 'Salt');
insert into Ingredients_T values('Fried Mushroom', 'Beer');
insert into Ingredients_T values('Fried Mushroom', 'Mushroom');
insert into Ingredients_T values('Fried Mushroom', 'Flour');
insert into Ingredients_T values('Fried Mushroom', 'Salt');
insert into Ingredients_T values('Fried Mushroom', 'Pepper');
insert into Ingredients_T values('Julienne Salad', 'Salad');
insert into Ingredients_T values('Julienne Salad', 'Chicken');
insert into Ingredients_T values('Julienne Salad', 'Cheese');
insert into Ingredients_T values('Julienne Salad', 'Ham');
insert into Ingredients_T values('Julienne Salad', 'Beef');
insert into Ingredients_T values('Grilled Tilapia Salad', 'Tilapia');
insert into Ingredients_T values('Grilled Tilapia Salad', 'Chilli Powder');
insert into Ingredients_T values('Grilled Tilapia Salad', 'Lime');
insert into Ingredients_T values('Grilled Tilapia Salad', 'Onion');
insert into Ingredients_T values('Grilled Tilapia Salad', 'Rice');
insert into Ingredients_T values('Grilled Tilapia Salad', 'Garlic');
insert into Ingredients_T values('Grilled Tilapia Salad', 'Avocado');
insert into Ingredients_T values('Taco Salad', 'Lettuce');
insert into Ingredients_T values('Taco Salad', 'Tomato');
insert into Ingredients_T values('Taco Salad', 'Cheese');
insert into Ingredients_T values('Taco Salad', 'Avocado');
insert into Ingredients_T values('Taco Salad', 'Salsa');
insert into Ingredients_T values('Taco Salad', 'Tortilla Chips');
insert into Ingredients_T values('Pork and Cabbage', 'Vegetable Oil');
insert into Ingredients_T values('Pork and Cabbage', 'Pork');
insert into Ingredients_T values('Pork and Cabbage', 'Cabbage');
insert into Ingredients_T values('Pork and Cabbage', 'Apple');
insert into Ingredients_T values('Pork and Cabbage', 'Onion');
insert into Ingredients_T values('Pork and Cabbage', 'Garlic');
insert into Ingredients_T values('Pork and Cabbage', 'Worcestershire Sauce');
insert into Ingredients_T values('Pork and Cabbage', 'Flour');
insert into Ingredients_T values('New England Clam Chowder', 'Bacon');
insert into Ingredients_T values('New England Clam Chowder', 'Onion');
insert into Ingredients_T values('New England Clam Chowder', 'Potato');
insert into Ingredients_T values('New England Clam Chowder', 'Milk');
insert into Ingredients_T values('New England Clam Chowder', 'Butter');
insert into Ingredients_T values('New England Clam Chowder', 'Clam');
insert into Ingredients_T values('Onion Rings', 'Onion');
insert into Ingredients_T values('Onion Rings', 'Beer');
insert into Ingredients_T values('Onion Rings', 'Flour');
insert into Ingredients_T values('Onion Rings', 'Vegetable Oil');
insert into Ingredients_T values('Classic BLT', 'Bacon');
insert into Ingredients_T values('Classic BLT', 'Lettuce');
insert into Ingredients_T values('Classic BLT', 'Tomato');
insert into Ingredients_T values('Classic BLT', 'French Bread');
insert into Ingredients_T values('Big Tuna', 'Tuna');
insert into Ingredients_T values('Big Tuna', 'Tomato');
insert into Ingredients_T values('Big Tuna', 'Wheat Bread');
insert into Ingredients_T values('Big Tuna', 'Cheese');
insert into Ingredients_T values('Grilled Salmon', 'Salmon');
insert into Ingredients_T values('Grilled Salmon', 'French Bread');
insert into Ingredients_T values('Fish and Chips', 'Haddock');
insert into Ingredients_T values('Fish and Chips', 'Fries');
insert into Ingredients_T values('Fish and Chips', 'Vegetable Oil');

---------------------------------------------------------------

drop table if exists Order_Line_T;
-- Populating OrderLine table --
insert into Order_Line_T values('ORD01', 'Fish and Chips', 2);
insert into Order_Line_T values('ORD01', 'Onion Rings', 1);
insert into Order_Line_T values('ORD01', 'Julienne Salad', 1);
insert into Order_Line_T values('ORD02', 'Julienne Salad', 3);
insert into Order_Line_T values('ORD03', 'Fried Mushroom', 1);
insert into Order_Line_T values('ORD03', 'French Fries', 1);
insert into Order_Line_T values('ORD04', 'Classic BLT', 1);
insert into Order_Line_T values('ORD05', 'Vegetable Pakora', 4);
insert into Order_Line_T values('ORD05', 'Peanut Chaat', 1);
insert into Order_Line_T values('ORD05', 'Julienne Salad', 1);
insert into Order_Line_T values('ORD06', 'Tomato Soup', 1);
insert into Order_Line_T values('ORD07', 'Strawberry Dessert', 6);
insert into Order_Line_T values('ORD08', 'Pot Roast', 4);
insert into Order_Line_T values('ORD09', 'Julienne Salad', 1);
insert into Order_Line_T values('ORD10', 'Tomato Soup', 7);
insert into Order_Line_T values('ORD10', 'Strawberry Dessert', 1);
insert into Order_Line_T values('ORD10', 'Pot Roast', 1);
insert into Order_Line_T values('ORD10', 'Classic BLT', 1);
insert into Order_Line_T values('ORD11', 'Strawberry Dessert', 1);
insert into Order_Line_T values('ORD11', 'Chicken Pasta', 3);
insert into Order_Line_T values('ORD12', 'Classic BLT', 1);
insert into Order_Line_T values('ORD13', 'Fish and Chips', 1);
insert into Order_Line_T values('ORD14', 'Grilled Salmon', 2);
insert into Order_Line_T values('ORD14', 'Chicken and Rice', 1);
insert into Order_Line_T values('ORD15', 'French Fries', 1);
insert into Order_Line_T values('ORD15', 'French Dip Sandwiches', 1);
insert into Order_Line_T values('ORD15', 'Pork and Cabbage', 6);
insert into Order_Line_T values('ORD16', 'Classic BLT', 1);
insert into Order_Line_T values('ORD16', 'Taco Salad', 1);
insert into Order_Line_T values('ORD16', 'Crockpot Salsa Chicken', 1);
insert into Order_Line_T values('ORD16', 'French Dip Sandwiches', 9);
insert into Order_Line_T values('ORD16', 'Pulled BBQ Chicken', 1);
insert into Order_Line_T values('ORD17', 'Karri Rice', 1);
insert into Order_Line_T values('ORD18', 'Strawberry Dessert', 1);
insert into Order_Line_T values('ORD19', 'Fish and Chips', 4);
insert into Order_Line_T values('ORD20', 'Chicken Pasta', 1);
insert into Order_Line_T values('ORD21', 'Strawberry Dessert', 1);
insert into Order_Line_T values('ORD22', 'Fried Mushroom', 1);
insert into Order_Line_T values('ORD22', 'Grilled Tilapia Salad', 1);
insert into Order_Line_T values('ORD23', 'Classic BLT', 6);
insert into Order_Line_T values('ORD24', 'Chicken Pasta', 1);
insert into Order_Line_T values('ORD25', 'Fried Mushroom', 1);
insert into Order_Line_T values('ORD25', 'Pulled BBQ CHicken', 4);
insert into Order_Line_T values('ORD25', 'Crockpot Salsa Chicken', 1);
insert into Order_Line_T values('ORD26', 'New England Clam Chowder', 2);
insert into Order_Line_T values('ORD26', 'Vegetable Pakora', 1);
insert into Order_Line_T values('ORD27', 'New England Clam Chowder', 1);
insert into Order_Line_T values('ORD28', 'Big Tuna', 1);
insert into Order_Line_T values('ORD29', 'Pork and Cabbage', 1);
insert into Order_Line_T values('ORD30', 'Pork and Cabbage', 3);
insert into Order_Line_T values('ORD30', 'Chicken Pasta', 2);
insert into Order_Line_T values('ORD30', 'Grilled Tilapia Salad', 1);
insert into Order_Line_T values('ORD31', 'Tomato Soup', 1);
insert into Order_Line_T values('ORD32', 'Grilled Salmon', 1);
insert into Order_Line_T values('ORD33', 'Taco Salad', 1);
insert into Order_Line_T values('ORD33', 'New England Clam Chowder', 1);
insert into Order_Line_T values('ORD34', 'Quesadillas', 7);
insert into Order_Line_T values('ORD35', 'Karri Rice', 1);
insert into Order_Line_T values('ORD36', 'New England Clam Chowder');
insert into Order_Line_T values('ORD36', 'Peanut Chaat');
insert into Order_Line_T values('ORD37', 'Quesadillas', 5);
insert into Order_Line_T values('ORD38', 'Peanut Chaat');
insert into Order_Line_T values('ORD39', 'Big Tuna', 1);
insert into Order_Line_T values('ORD39', 'Chicken and Rice', 1);
insert into Order_Line_T values('ORD39', 'Pulled BBQ Chicken', 1);
insert into Order_Line_T values('ORD40', 'Peanut Chaat', 1);
insert into Order_Line_T values('ORD40', 'Quesadillas', 1);
insert into Order_Line_T values('ORD41', 'Karri Rice', 1);
insert into Order_Line_T values('ORD42', 'Pork and Cabbage', 1);
insert into Order_Line_T values('ORD43', 'Karri Rice', 1);
insert into Order_Line_T values('ORD44', 'Peanut Chaat', 1);
insert into Order_Line_T values('ORD45', 'Quesadillas', 2);
insert into Order_Line_T values('ORD45', 'Karri Rice', 1);
insert into Order_Line_T values('ORD45', 'Chicken and Rice', 1);
insert into Order_Line_T values('ORD45', 'Pulled BBQ Chicken', 1);
insert into Order_Line_T values('ORD46', 'Pot Roast', 1);
insert into Order_Line_T values('ORD46', 'Pulled BBQ Chicken', 1);
insert into Order_Line_T values('ORD47', 'Taco Salad', 10);
insert into Order_Line_T values('ORD48', 'Ribs', 1);
insert into Order_Line_T values('ORD49', 'Pork and Cabbage', 3);
insert into Order_Line_T values('ORD49', 'French Fries', 1);
insert into Order_Line_T values('ORD49', 'Quesadillas', 1);
insert into Order_Line_T values('ORD49', 'Grilled Salmon', 1);
insert into Order_Line_T values('ORD49', 'Karri Rice', 1);
insert into Order_Line_T values('ORD50', 'Fish and Chips', 4);
insert into Order_Line_T values('ORD51', 'Taco Salad', 5);
insert into Order_Line_T values('ORD52', 'Ribs', 5);
insert into Order_Line_T values('ORD53', 'French Fries', 1);
insert into Order_Line_T values('ORD54', 'Strawberry Dessert', 1);
insert into Order_Line_T values('ORD54', 'New England Clam Chowder', 1);
insert into Order_Line_T values('ORD55', 'Ribs', 1);
insert into Order_Line_T values('ORD56', 'Ribs', 7);
insert into Order_Line_T values('ORD57', 'Big Tuna', 1);
insert into Order_Line_T values('ORD58', 'Pot Roast', 1);
insert into Order_Line_T values('ORD59', 'New England Clam Chowder', 2 );
insert into Order_Line_T values('ORD59', 'Karri Rice', 1);
insert into Order_Line_T values('ORD60', 'Tomato Soup', 2);
insert into Order_Line_T values('ORD61', 'Big Tuna', 1);
insert into Order_Line_T values('ORD61', 'French Dip Sandwiches', 1);
insert into Order_Line_T values('ORD62', 'French Fries', 10);
insert into Order_Line_T values('ORD63', 'Peanut Chaat', 1);
insert into Order_Line_T values('ORD63', 'Cashew Chicken Salad', 1);
insert into Order_Line_T values('ORD64', 'Cashew Chicken Salad', 3);
insert into Order_Line_T values('ORD65', 'New England Clam Chowder', 1);
insert into Order_Line_T values('ORD65', 'French Fries', 1);
insert into Order_Line_T values('ORD65', 'Big Tuna', 1);
insert into Order_Line_T values('ORD66', 'New England Clam Chowder', 3);
insert into Order_Line_T values('ORD67', 'Vegetable Pakora', 1);
insert into Order_Line_T values('ORD68', 'French Fries', 1);
insert into Order_Line_T values('ORD68', 'Pork and Cabbage', 1);
insert into Order_Line_T values('ORD68', 'Karri Rice', 1);
insert into Order_Line_T values('ORD69', 'Crockpot Salsa Chicken', 2);
insert into Order_Line_T values('ORD69', 'Cashew Chicken Salad', 1);
insert into Order_Line_T values('ORD70', 'Vegetable Pakora', 1);
insert into Order_Line_T values('ORD71', 'Taco Salad', 1);
insert into Order_Line_T values('ORD71', 'Big Tuna', 4);
insert into Order_Line_T values('ORD71', 'Pork and Cabbage', 1);
insert into Order_Line_T values('ORD72', 'Quesadillas', 11);
insert into Order_Line_T values('ORD72', 'Grilled Salmon');
insert into Order_Line_T values('ORD73', 'Taco Salad', 8);
insert into Order_Line_T values('ORD73', 'Julienne Salad', 3);

----------------------------------------------------------------

drop table if exists Payment_T;
-- Populating Payment table --
```

```
insert into Payment_T values('CA000', 'ORD01', '2021-03-07', 'Cash');
insert into Payment_T values('CA001', 'ORD03', '2021-02-07', 'Cash');
insert into Payment_T values('CA002', 'ORD05', '2021-03-04', 'Cash');
insert into Payment_T values('CA003', 'ORD06', '2021-01-20', 'Cash');
insert into Payment_T values('CA004', 'ORD09', '2021-01-12', 'Cash');
insert into Payment_T values('CA005', 'ORD10', '2021-01-30', 'Cash');
insert into Payment_T values('CA006', 'ORD12', '2021-02-25', 'Cash');
insert into Payment_T values('CA007', 'ORD13', '2021-02-01', 'Cash');
insert into Payment_T values('CA008', 'ORD15', '2021-02-12', 'Cash');
insert into Payment_T values('CA009', 'ORD16', '2021-01-21', 'Cash');
insert into Payment_T values('CA010', 'ORD18', '2021-02-24', 'Cash');
insert into Payment_T values('CA011', 'ORD25', '2021-02-08', 'Cash');
insert into Payment_T values('CA012', 'ORD31', '2021-03-03', 'Cash');
insert into Payment_T values('CA013', 'ORD33', '2021-02-15', 'Cash');
insert into Payment_T values('CA014', 'ORD34', '2021-01-27', 'Cash');
insert into Payment_T values('CA016', 'ORD38', '2021-01-18', 'Cash');
insert into Payment_T values('CA015', 'ORD37', '2021-03-01', 'Cash');
insert into Payment_T values('CA017', 'ORD43', '2021-01-16', 'Cash');
insert into Payment_T values('CA018', 'ORD44', '2021-02-01', 'Cash');
insert into Payment_T values('CA019', 'ORD49', '2021-01-21', 'Cash');
insert into Payment_T values('CA020', 'ORD51', '2021-01-14', 'Cash');
insert into Payment_T values('CA021', 'ORD55', '2021-02-10', 'Cash');
insert into Payment_T values('CA022', 'ORD63', '2021-01-30', 'Cash');
insert into Payment_T values('CA023', 'ORD69', '2021-02-05', 'Cash');
insert into Payment_T values('CA024', 'ORD73', '2021-01-05', 'Cash');
insert into Payment_T values('CR000', 'ORD04', '2021-02-03', 'Credit');
insert into Payment_T values('CR001', 'ORD17', '2021-01-18', 'Credit');
insert into Payment_T values('CR002', 'ORD20', '2021-02-28', 'Credit');
insert into Payment_T values('CR003', 'ORD22', '2021-01-24', 'Credit');
insert into Payment_T values('CR004', 'ORD24', '2021-01-09', 'Credit');
insert into Payment_T values('CR005', 'ORD32', '2021-03-04', 'Credit');
insert into Payment_T values('CR006', 'ORD41', '2021-02-27', 'Credit');
insert into Payment_T values('CR007', 'ORD45', '2021-02-26', 'Credit');
insert into Payment_T values('CR008', 'ORD46', '2021-01-05', 'Credit');
insert into Payment_T values('CR009', 'ORD47', '2021-02-28', 'Credit');
insert into Payment_T values('CR010', 'ORD52', '2021-02-04', 'Credit');
insert into Payment_T values('CR011', 'ORD53', '2021-01-23', 'Credit');
insert into Payment_T values('CR012', 'ORD59', '2021-02-24', 'Credit');
insert into Payment_T values('CR013', 'ORD60', '2021-02-10', 'Credit');
insert into Payment_T values('CR014', 'ORD62', '2021-01-31', 'Credit');
insert into Payment_T values('CR015', 'ORD64', '2021-01-18', 'Credit');
insert into Payment_T values('CR016', 'ORD65', '2021-02-18', 'Credit');
insert into Payment_T values('CR017', 'ORD66', '2021-02-11', 'Credit');
insert into Payment_T values('CR018', 'ORD68', '2021-01-10', 'Credit');
insert into Payment_T values('CR019', 'ORD71', '2021-02-06', 'Credit');
insert into Payment_T values('CR020', 'ORD72', '2021-02-11', 'Credit');
insert into Payment_T values('PR000', 'ORD02', '2021-01-15', 'Coupon');
insert into Payment_T values('PR001', 'ORD07', '2021-01-18', 'Coupon');
insert into Payment_T values('PR002', 'ORD08', '2021-01-13', 'Coupon');
insert into Payment_T values('PR003', 'ORD11', '2021-01-24', 'Coupon');
insert into Payment_T values('PR004', 'ORD14', '2021-02-27', 'Coupon');
insert into Payment_T values('PR005', 'ORD19', '2021-01-16', 'Coupon');
insert into Payment_T values('PR006', 'ORD21', '2021-02-15', 'Coupon');
insert into Payment_T values('PR007', 'ORD23', '2021-01-24', 'Coupon');
insert into Payment_T values('PR008', 'ORD26', '2021-02-28', 'Coupon');
insert into Payment_T values('PR009', 'ORD27', '2021-01-08', 'Coupon');
insert into Payment_T values('PR010', 'ORD28', '2021-01-09', 'Coupon');
insert into Payment_T values('PR011', 'ORD29', '2021-01-13', 'Coupon');
insert into Payment_T values('PR012', 'ORD30', '2021-02-04', 'Coupon');
insert into Payment_T values('PR013', 'ORD35', '2021-01-29', 'Coupon');
insert into Payment_T values('PR014', 'ORD36', '2021-02-24', 'Coupon');
insert into Payment_T values('PR015', 'ORD39', '2021-02-09', 'Coupon');
insert into Payment_T values('PR016', 'ORD40', '2021-02-20', 'Coupon');
insert into Payment_T values('PR017', 'ORD42', '2021-01-12', 'Coupon');
insert into Payment_T values('PR018', 'ORD48', '2021-01-01', 'Coupon');
insert into Payment_T values('PR019', 'ORD50', '2021-01-06', 'Coupon');
insert into Payment_T values('PR020', 'ORD54', '2021-01-12', 'Coupon');
insert into Payment_T values('PR021', 'ORD56', '2021-01-22', 'Coupon');
insert into Payment_T values('PR022', 'ORD57', '2021-01-06', 'Coupon');
insert into Payment_T values('PR023', 'ORD58', '2021-02-04', 'Coupon');
insert into Payment_T values('PR024', 'ORD61', '2021-01-30', 'Coupon');
insert into Payment_T values('PR025', 'ORD67', '2021-02-24', 'Coupon');
insert into Payment_T values('PR026', 'ORD70', '2021-03-07', 'Coupon');

----------------------------------------------------------------


drop table if exists Cash_T;
-- Populating Cash table --
insert into Cash_T values('CA000', 'Yes');
insert into Cash_T values('CA001', 'No');
insert into Cash_T values('CA002', 'No');
insert into Cash_T values('CA003', 'Yes');
insert into Cash_T values('CA004', 'Yes');
insert into Cash_T values('CA005', 'Yes');
insert into Cash_T values('CA006', 'No');
insert into Cash_T values('CA007', 'Yes');
insert into Cash_T values('CA008', 'Yes');
insert into Cash_T values('CA009', 'No');
insert into Cash_T values('CA010', 'No');
insert into Cash_T values('CA011', 'Yes');
insert into Cash_T values('CA012', 'Yes');
insert into Cash_T values('CA013', 'Yes');
insert into Cash_T values('CA014', 'Yes');
insert into Cash_T values('CA015', 'No');
insert into Cash_T values('CA016', 'No');
insert into Cash_T values('CA017', 'No');
insert into Cash_T values('CA018', 'No');
insert into Cash_T values('CA019', 'Yes');
insert into Cash_T values('CA020', 'Yes');
insert into Cash_T values('CA021', 'No');
insert into Cash_T values('CA022', 'No');
insert into Cash_T values('CA023', 'No');
insert into Cash_T values('CA024', 'Yes');

------------------------------------------------------------------------------

drop table if exists Credit_T;
-- Populating Credit table --
insert into Credit_T values('CR000', 'VISA', 'USD');
insert into Credit_T values('CR001', 'American Express', 'USD');
insert into Credit_T values('CR002', 'VISA', 'USD');
insert into Credit_T values('CR003', 'VISA', 'USD');
insert into Credit_T values('CR004', 'VISA', 'USD');
insert into Credit_T values('CR005', 'MasterCard', 'USD');
insert into Credit_T values('CR006', 'American Express', 'USD');
insert into Credit_T values('CR007', 'American Express', 'USD');
insert into Credit_T values('CR008', 'MasterCard', 'USD');
insert into Credit_T values('CR009', 'American Express', 'USD');
insert into Credit_T values('CR010', 'VISA', 'USD');
insert into Credit_T values('CR011', 'MasterCard', 'USD');
insert into Credit_T values('CR012', 'MasterCard', 'USD');
insert into Credit_T values('CR013', 'VISA', 'USD');
insert into Credit_T values('CR014', 'American Express', 'USD');
insert into Credit_T values('CR015', 'VISA', 'USD');
insert into Credit_T values('CR016', 'MasterCard', 'USD');
insert into Credit_T values('CR017', 'MasterCard', 'USD');
insert into Credit_T values('CR018', 'VISA', 'USD');
insert into Credit_T values('CR019', 'American Express', 'USD');
insert into Credit_T values('CR020', 'American Express', 'USD');

---------------------------------------------------------------


drop table if exists Coupon_T;
-- Populating Coupon table --
insert into Coupon_T values('PR000', 'Save15', 15.00);
insert into Coupon_T values('PR001', 'BDay10', 10.00);
insert into Coupon_T values('PR002', 'BDay10', 10.00);
insert into Coupon_T values('PR003', 'Save15', 15.00);
insert into Coupon_T values('PR004', 'BDay10', 10.00);
insert into Coupon_T values('PR005', 'Save15', 15.00);
insert into Coupon_T values('PR006', 'Save15', 15.00);
insert into Coupon_T values('PR007', 'Save15', 15.00);
insert into Coupon_T values('PR008', 'BDay10', 10.00);
insert into Coupon_T values('PR009', 'BDay10', 10.00);
insert into Coupon_T values('PR010', 'Save15', 15.00);
insert into Coupon_T values('PR011', 'Save15', 15.00);
insert into Coupon_T values('PR012', 'Save15', 15.00);
insert into Coupon_T values('PR013', 'BDay10', 10.00);
insert into Coupon_T values('PR014', 'BDay10', 10.00);
insert into Coupon_T values('PR015', 'Save15', 15.00);
insert into Coupon_T values('PR016', 'Save15', 15.00);
insert into Coupon_T values('PR017', 'Save15', 15.00);
insert into Coupon_T values('PR018', 'BDay10', 10.00);
insert into Coupon_T values('PR019', 'BDay10', 10.00);
insert into Coupon_T values('PR020', 'Save15', 15.00);
insert into Coupon_T values('PR021', 'BDay10', 10.00);
insert into Coupon_T values('PR022', 'Save15', 15.00);
insert into Coupon_T values('PR023', 'Save15', 15.00);
insert into Coupon_T values('PR024', 'Save15', 15.00);
insert into Coupon_T values('PR025', 'BDay10', 10.00);
insert into Coupon_T values('PR026', 'BDay10', 10.00);
```

Restaurant_Que.sql

```
-- Group 2: The Urban Fence Restaurant Database
-- Group Members: Amaya Alisca, Deandra Martin, Alex Mayer, Cam Reeves
-- CMS 375
-- Due Date: 4/22/2021

-- QUERIES --
-- to find the price of chicken and rice
select Price from Food_Item_T where Name = 'Chicken and Rice';

-- to find all payment methods accepted
select distinct Payment_Type from Payment_T;

-- list in order the names and phone number of chef(s) who worked on the M T F ST 2:00 PM -
10:00 PM shift
select Name, Phone_Number from Chef_T, Shift_T where Chef_T.Chef_ID = Shift_T.Chef_ID
and Shift = 'M T F ST 2:00 PM - 10:00 PM' order by Name;

-- to find out the payment method used for order ODR06
select Payment_Type from Payment_T where Order_ID = 'ORD06';

-- to find out how many people participate in each promotion
select Promo_Code, count(Promo_Code) as 'No. of Participants' from Coupon_T group by
Promo_Code;

-- list the ingredients for a Chicken and rice
select distinct Ingredient from Ingredients_T, Food_Item_T where Food_Name = Name and
Food_Name = 'Chicken and rice';
```

```
-- to find out which customers purchased from the restaurant more than once

select Customer_t.Name, count(Order_t.Customer_ID) as 'Times purchased' from customer_t,
Order_T where Customer_t.Customer_ID = order_t.customer_id group by Customer_T.Name
having count(Order_t.Customer_ID) > 1;
```

```
-- find the food items ordered in quantities over 2

select Food_Name, max(Quantity) from Order_Line_T group by Food_Name having
max(Quantity) > 2;
```

```
-- get name and contact for all customers who ordered the pot roast

select Customer_T.Name, Contact from Customer_T, Order_T, Order_Line_T where
Customer_T.Customer_ID = Order_T.Customer_ID and Order_T.Order_ID =
Order_Line_T.Order_ID and Order_Line_T.Food_Name = 'Pot Roast';
```

```
-- get a list of meals made with chicken

select distinct Food_Name from Ingredients_T where Food_Name like '%chicken%' or
Ingredient like '%chicken%';
```

```
-- get the available dishes and their prices

Select Name, Price from Food_Item_T;
```

```
-- get the average price of food items on menu

select avg(Price) as 'Average Price' from Food_Item_T;
```

```
-- get the promocode of coupon PR010

select Promo_Code from Coupon_T where Coupon_ID = 'PR010';
```

```
-- get all chefs who start working on Monday, list alphabetically

select Chef_T.Chef_ID, Name, Phone_Number from Chef_T, Shift_T where Chef_T.Chef_ID =
Shift_T.Chef_ID and Shift like 'M%' order by Name;
```

```
-- get order total of ORD16

select sum(Quantity * Price) as 'Order Total for ORD16' from Order_Line_T, Food_Item_T
where Food_Name = Name and Order_ID = 'ORD16';


-- find out how many ingredients are in each food item

select Name, count(Ingredient) as 'No. of Ingredients' from Food_Item_T, Ingredients_T where
Name = Food_Name group by Name;


-- how many customers paid with cash

select count(distinct Customer_t.customer_id) as 'Paid Cash' from Customer_t, order_t,
payment_t where customer_t.customer_id = order_t.customer_id

and order_t.order_id = payment_t.order_id and payment_type = 'cash';


-- Display all chefs with the letter s in them

Select * from Chef_t where Name like 'a%';


-- get list of order ids and payment types where payment was made between 2021-01-15 and
2021-02-20

-- list in desc order

select Order_ID, Payment_ID, Date from Payment_T where Date between '2021-01-15' and
'2021-02-20' order by Date desc;

--------------------------------------------------


-- OTHER DML COMMANDS --


-- to change the discount amount of promocode Save15

Update Coupon_T set Discount_Amnt = 05.00 where Promo_Code = 'Save15';


-- update the price of food item ribs

Update Food_Item_T set Price = 29.00 where Name = 'Ribs';
```

```
-- add a new item to the available dishes
insert into Food_Item_T values('Spinach and Ham', '101', 12.30);

-- change chef 104's shift
Update Shift_T set Shift = 'SN M T W 6:00 AM - 2:00 PM' where Chef_ID = '104';

-- delete French Dip Sandwiches from the ingredients table
Delete from Ingredients_T where Food_Name = 'French Dip Sandwiches';
```