###### **Views**

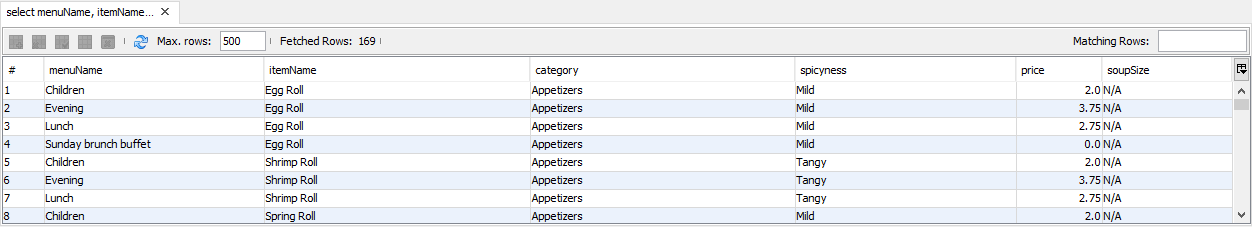
1. MenuItem\_v – For each menu item, give it’s spiciness, and all of the different costs for that item. If a given item is not on a particular menu, then report “N/A” for that particular item for that particular menu. Also, if an item only appears as a single serving portion, put in “N/A” into the report for the gallon, … prices.

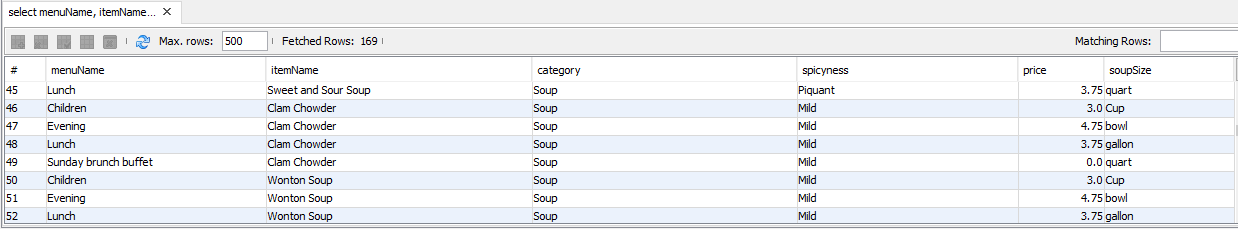
create VIEW MenuItem\_V as

select menuName, itemName,category, spicyness, price, soupSize

from Menuitem left join Orderselect

on Menuitem.itemID = Orderselect.itemID;



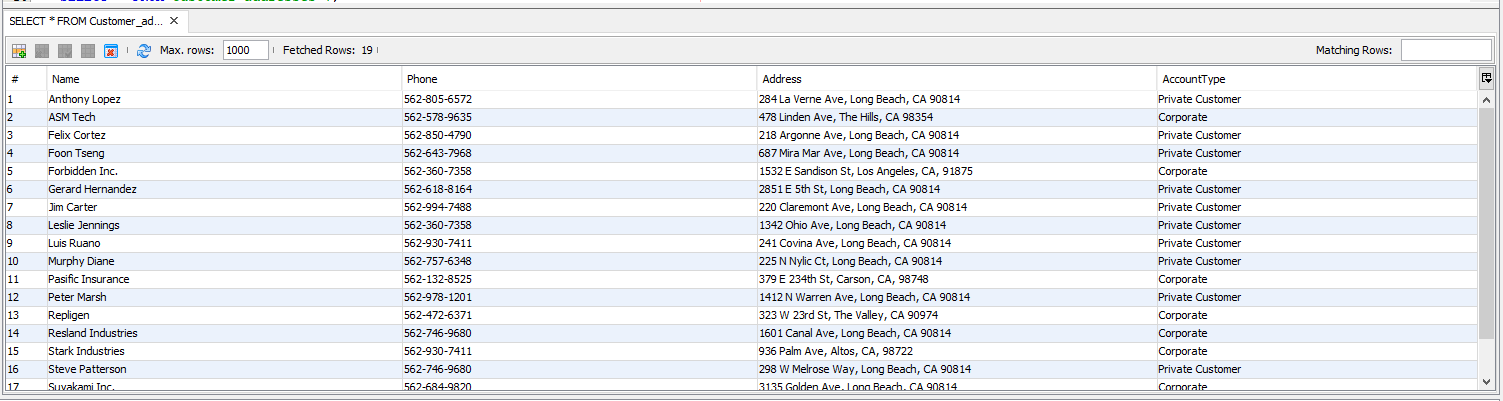


1. Customer\_addresses\_v – for each customer, indicate whether they are an individual or a corporate account, and display all of the information that we are managing for that customer.

CREATE VIEW Customer\_addresses\_v AS

SELECT customerName AS Name, customerPhone AS Phone, SnailMail AS Address, 'Private Customer' AS AccountType FROM Customers NATURAL JOIN Privatecustomer

UNION SELECT corporationName AS Name, customerPhone AS Phone, OfficeAddress AS Address, 'Corporate' AS AccountType FROM Customers NATURAL JOIN Corporate ORDER BY name;



1. . Sous\_mentor\_v – reports all of the mentor/mentee relationships at Miming’s, sorted by the name of the mentor, then the name of the mentee. Show the skill that the mentorship passes, as well as the start date.

CREATE VIEW Sous\_mentor\_v AS

SELECT concat(E.FirstName, E.LastName ) AS menteeName, concat(M.FirstName, M.LastName ) AS mentorName, S.itemName AS SkillPassed, S.startDate

From (SELECT \* FROM Employee NATURAL JOIN Souschef WHERE Employee.empID = Souschef.empID) AS E LEFT OUTER JOIN

((SELECT \* FROM Mentoring NATURAL JOIN Menuitem WHERE Mentoring.itemID = Menuitem.itemID) AS S

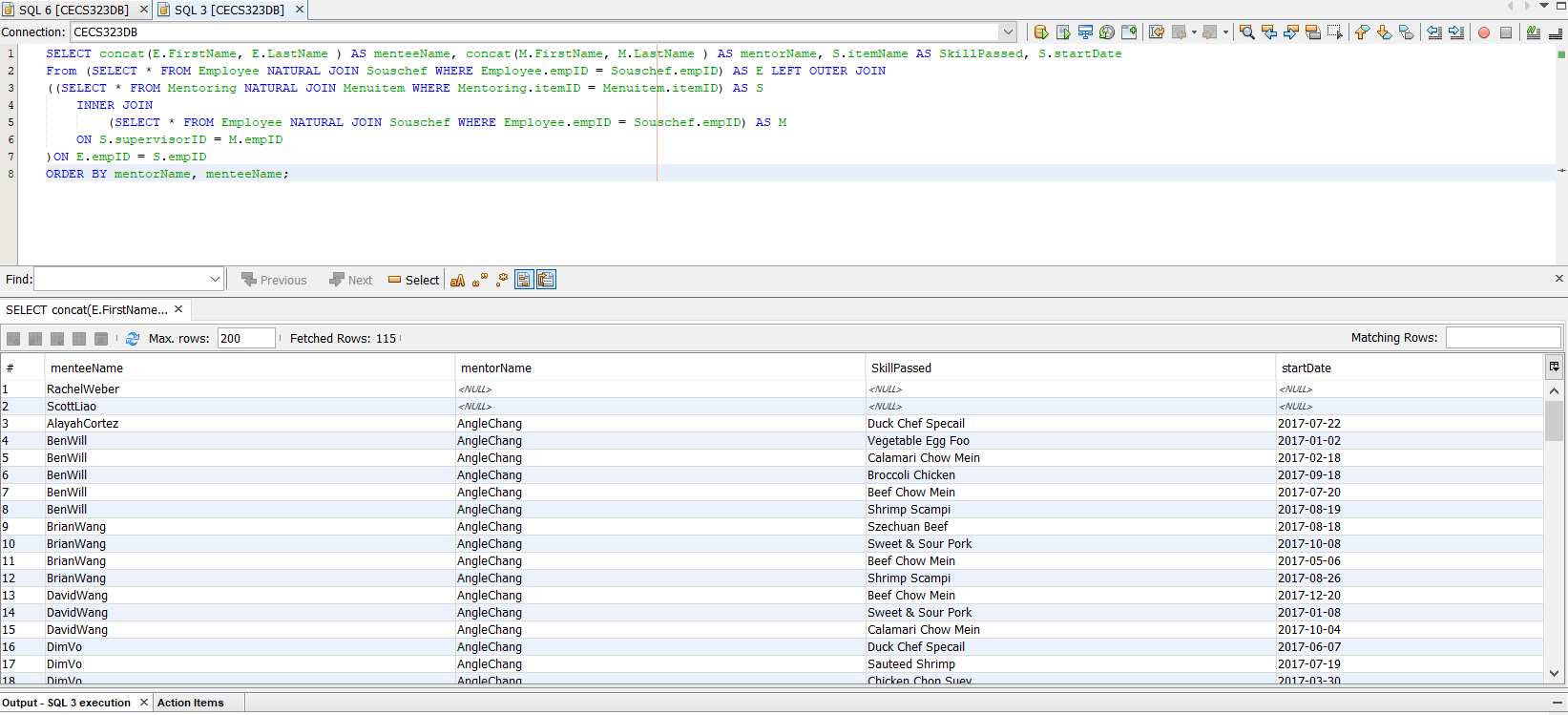
INNER JOIN

(SELECT \* FROM Employee NATURAL JOIN Souschef WHERE Employee.empID = Souschef.empID) AS M

ON S.supervisorID = M.empID

)ON E.empID = S.empID

ORDER BY mentorName, menteeName;



1. Customer\_Sales\_v – On a year by year basis, show how much each customer has spent at Miming’s.

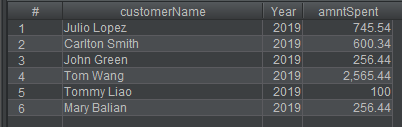
CREATE VIEW Customer\_Sales\_v AS

select customerName,year(orderDate) as "Year", amntSpent from OrderJunction

natural join Account left join Customers using(customerID)

group by customerName

order by count(orderDate) desc;



1. List each customer and the total $ amount of their orders for the past year, in order of the value of customer orders, from highest to the lowest.

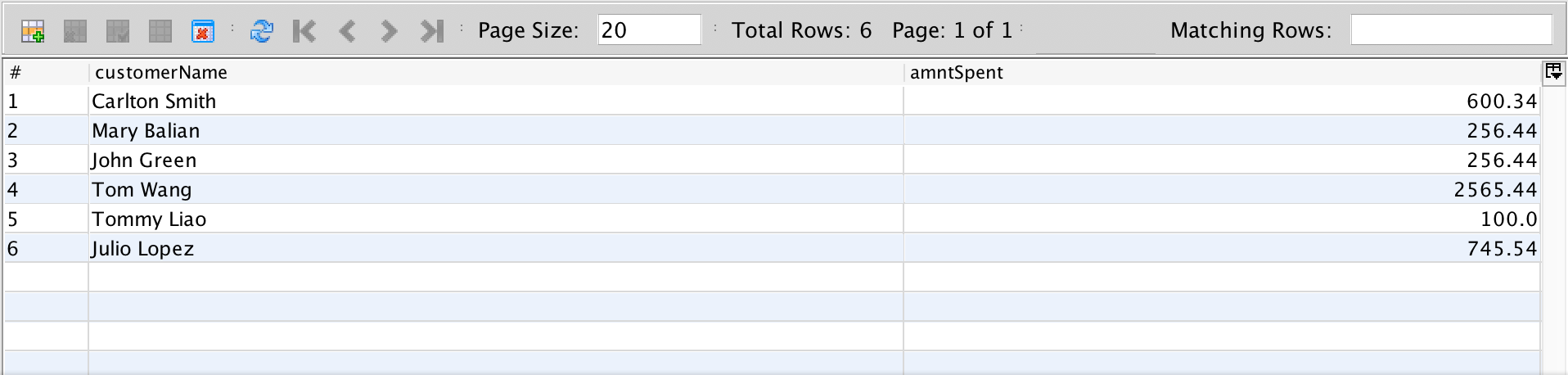
Create View CustomerValuev as

select customerName, amntSpent

from Customers

inner join Account on

Customers.customerID = Account.customerID;



###### **Queries**

1. List the customers. For each customer, indicate which category he or she fall into, and his or her contact information. If you have more than one independent categorization of customers, please indicate which category the customer falls into for all of the categorizations.

SELECT customerName AS "Name", customerPhone AS "Phone", SnailMail AS "Address",

'Private Customer' AS "AccountType" FROM Customers NATURAL JOIN Privatecustomer

UNION

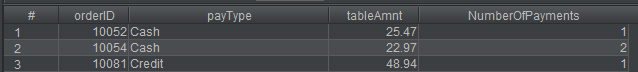
SELECT corporationName AS "Name", customerPhone AS "Phone", OfficeAddress AS "Address",

'Corporate' AS "AccountType" FROM Customers NATURAL JOIN Corporate ORDER BY Name;



1. For each order, list the total cost to the customer for that order. Sometimes, a group of customers will come in and split the order up into multiple checks. Treat that as though you had two separate parties at the same table.

select \* , count(orderID) as NumberOfPayments from Payments group by orderID;



1. List the top three customers in terms of their net spending for the past two years, and the total that they have spent in that period.

SELECT customerName AS "Name", SUM(quantity \* price) AS "Total Spent"

FROM Customers c INNER JOIN OrderJunction oj ON c.customerID = oj.customerID

INNER JOIN

Orders on oj.orderID = Orders.orderID

INNER JOIN

Orderdetail ord on Orders.orderID = ord.orderID

INNER JOIN

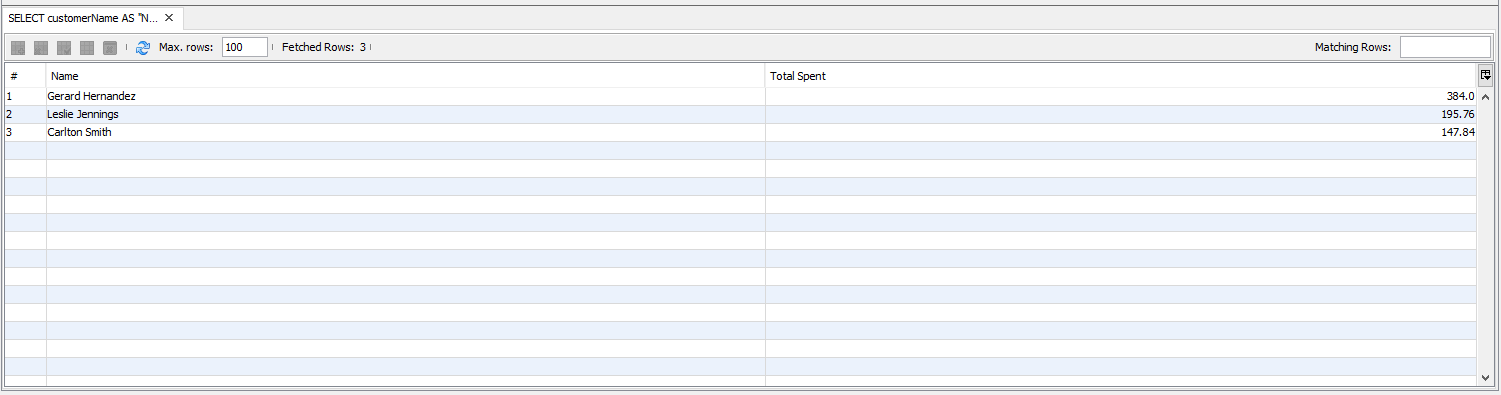
Orderselect os on ord.itemID = os.itemID

WHERE YEAR(orderDate) <2020 AND YEAR(orderDate)>2017

GROUP BY customerName

ORDER BY SUM(quantity \* price) DESC

LIMIT 3;



1. Find all of the sous chefs who have three or more menu items that they can prepare. For each sous chef, list their name, the number of menu items that they can prepare, and each of the menu items. You can use group\_concat to get all of a given sous chef’s data on one row, or print out one row per mechanic per skill.

SELECT FirstName, LastName, NumLearnedDishes, itemName FROM Employee

INNER JOIN Chef ON Employee.empID = Chef.empID

INNER JOIN Souschef ON Chef.empID = Souschef.empID

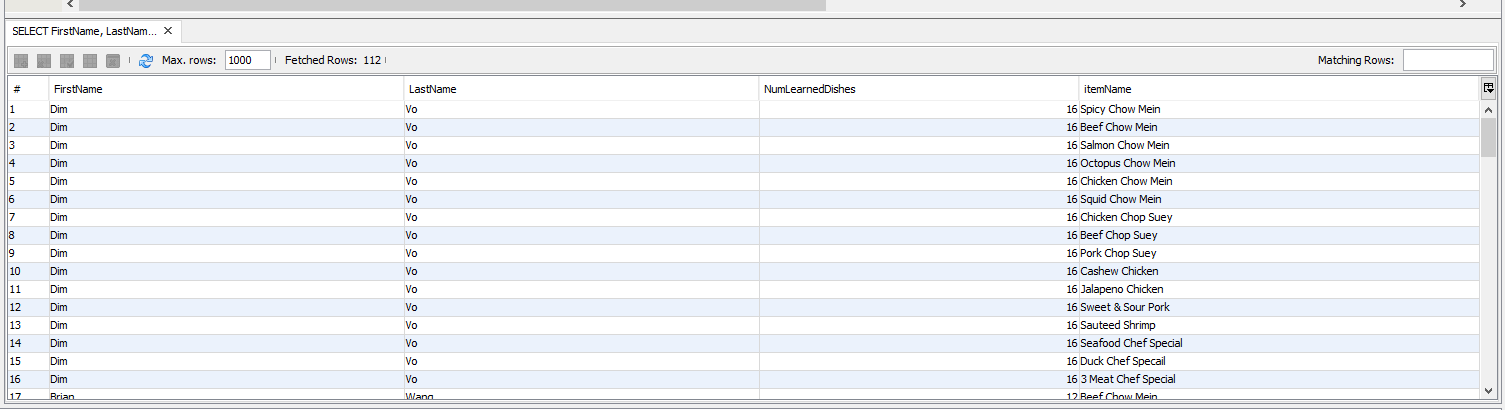
INNER JOIN Mentoring ON Souschef.empID = Mentoring.empID

INNER JOIN Menuitem ON Mentoring.itemID = Menuitem.itemID

WHERE Souschef.NumLearnedDishes >= 3

ORDER BY NumLearnedDishes DESC

LIMIT 1000;



1. Find all of the sous chefs who have three or more menu items in common.
   * 1. Please give the name of each of the two sous chefs sharing three or more menu items.
     2. Please make sure that any given pair of sous chefs only shows up once.

select s1.firstName as "Chef 1", s2.firstName as "Chef 2", COUNT(itemID) as "Shared items"

from Employee as s1, Employee as s2

natural join Chef

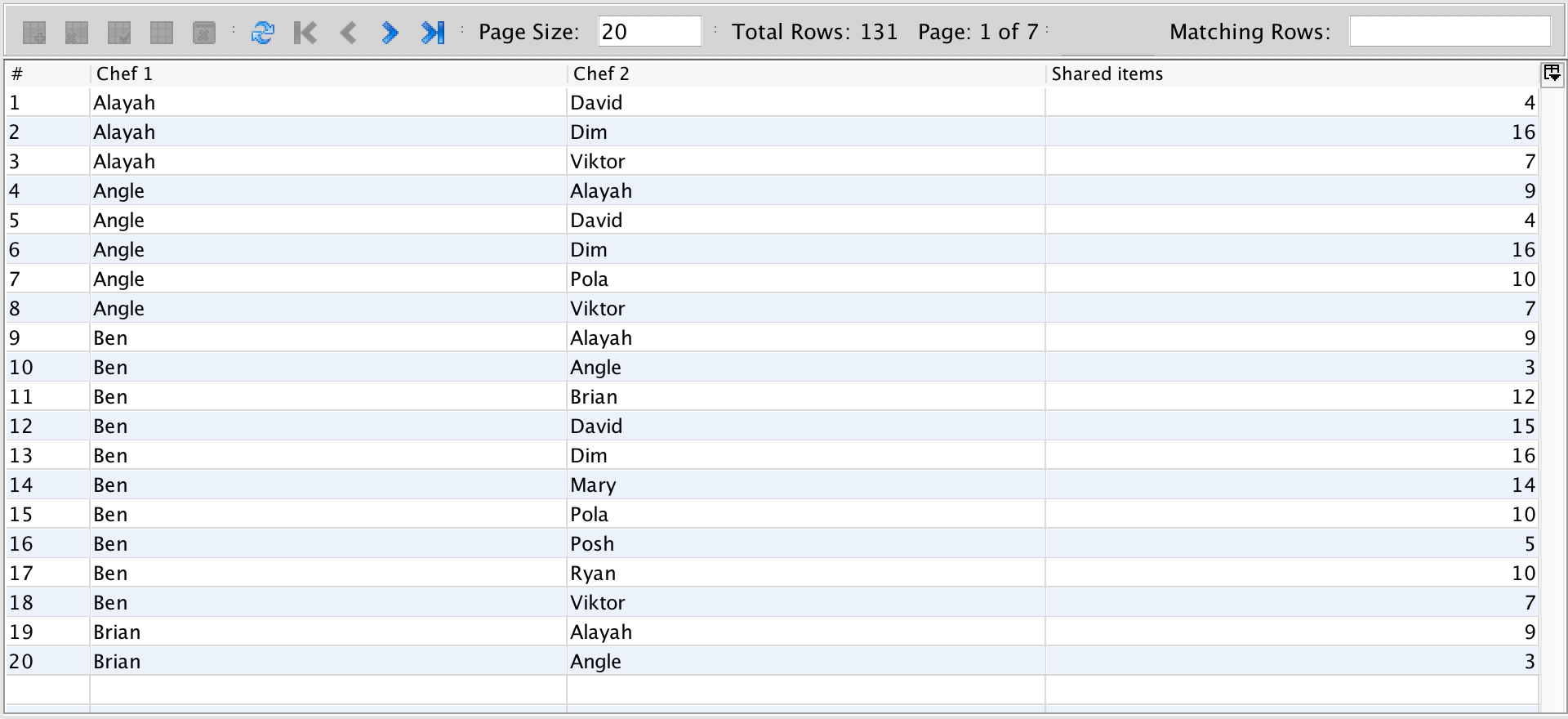
natural join Souschef

natural join Mentoring

where s1.empID != s2.empID and s1.empID < s2.empID

group by s1.firstName, s2.firstName

having COUNT(itemID) >= 3



1. Find the three menu items most often ordered from the Children’s menu and order them from most frequently ordered to least frequently ordered.

select itemName, COUNT(orderID) as "Ordered" from Menuitem

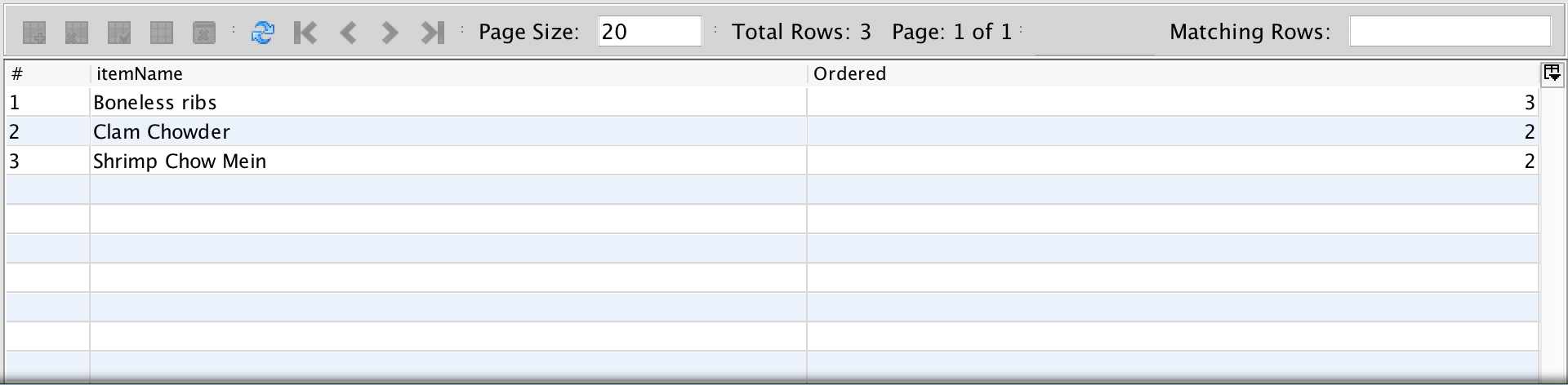
inner join Orderselect on Menuitem.itemID = Orderselect.itemID

inner join Orderdetail on Orderselect.itemID = Orderdetail.itemID

where menuName = 'Children'

group by itemName

order by COUNT(orderID) DESC limit 3;



1. List all of the menu items, the shift in which the menu item was ordered, and the sous chef on duty at the time, when the sous chef was not an expert in that menu item.

SELECT itemName, CrewType AS "Shift", Employee.FirstName AS "Sous Chef First Name", Employee.LastName AS "Sous Chef Last Name" FROM Employee

INNER JOIN Chef ON Employee.empID = Chef.empID

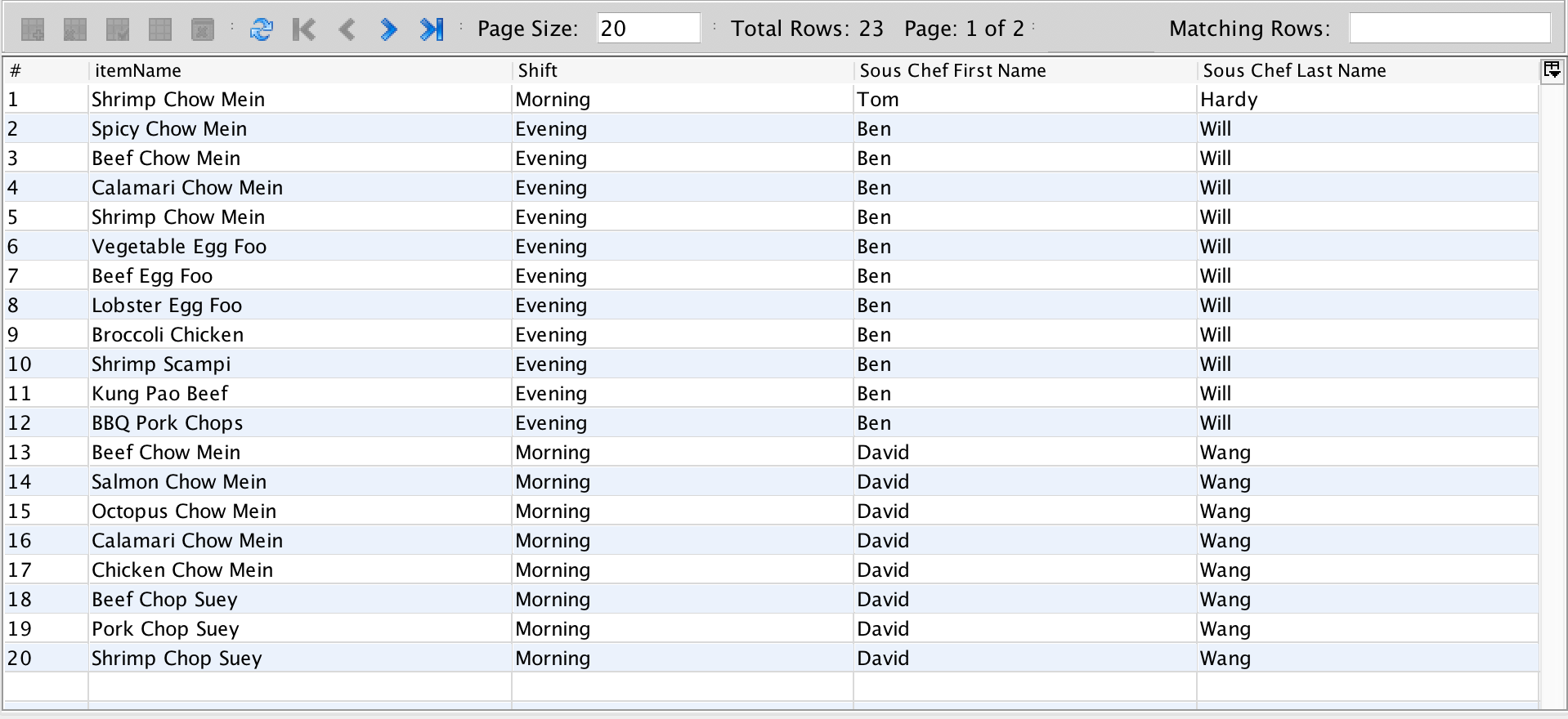
INNER JOIN Souschef ON Chef.empID = Souschef.empID

INNER JOIN Mentoring ON Souschef.empID = Mentoring.empID

INNER JOIN Menuitem ON Mentoring.itemID = Menuitem.itemID

INNER JOIN Schedule ON Employee.empID = Schedule.empID

INNER JOIN WorkCrew ON Schedule.empID = WorkCrew.empID

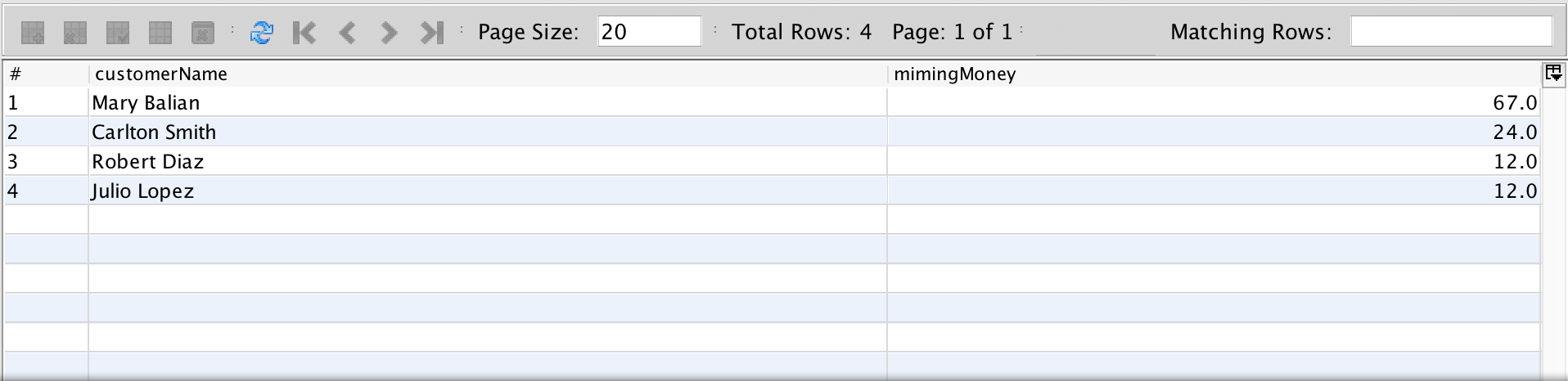


1. List the customers, sorted by the amount of Miming’s Money that they have, from largest to smallest.

select customerName, mimingMoney from Account

inner join Customers on Account.customerID = Customers.customerID

order by mimingMoney desc;



1. .List the customers and the total that they have spent at Miming’s ever, in descending order by the amount that they have spent.

select customerName, sum(quantity\*price) as "Total Spent"

from Customers c inner join OrderJunction oj

on c.customerID = oj.customerID

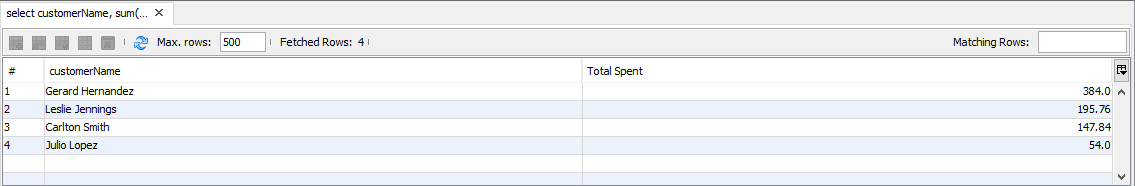
inner join Orders on oj.orderID = Orders.orderID inner join

Orderdetail ord on Orders.orderID = ord.orderID

inner join Orderselect os on ord.itemID = os.itemID

group by customerName

order by sum(quantity\*price) desc;



1. Report on the customers at Miming’s by the number of times that they come in by month, and order the report from most frequent to the least frequent

select customerName,month(orderDate) as "Month" ,year(orderDate) as "Year" ,count(orderDate) as "Number of Visits"

from OrderJunction left join Customers using(customerID)

group by customerName

order by count(orderDate) desc;



1. List the three customers who have spent the most at Miming’s over the past year. Order by the amount that they spent, from largest to smallest.

select customerName, orderDate as "Date Ordered",sum(quantity\*price) as "Total Spent"

from Customers c inner join OrderJunction oj

on c.customerID = oj.customerID

inner join Orders on oj.orderID = Orders.orderID inner join

Orderdetail ord on Orders.orderID = ord.orderID

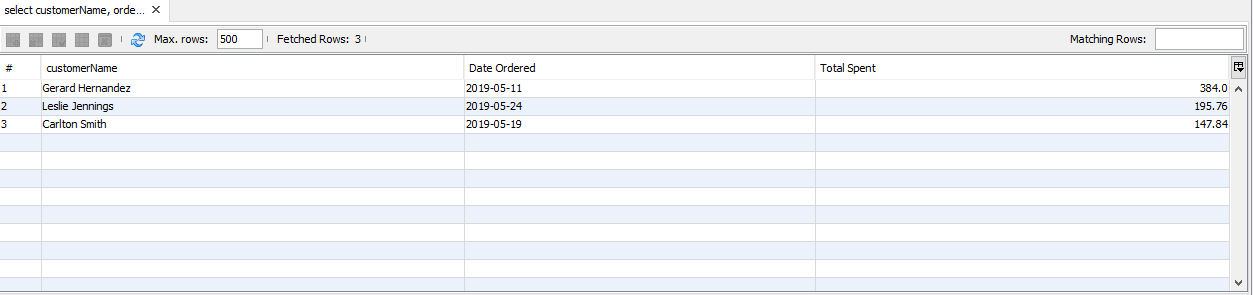
inner join Orderselect os on ord.itemID = os.itemID

where year(orderDate) = 2019

group by customerName

order by sum(quantity\*price) desc

limit 3;



1. List the five menu items that have generated the most revenue for Miming’s over the past year.

select itemID, itemName, sum(quantity\*price) from Menuitem

inner join Orderselect using(itemID)

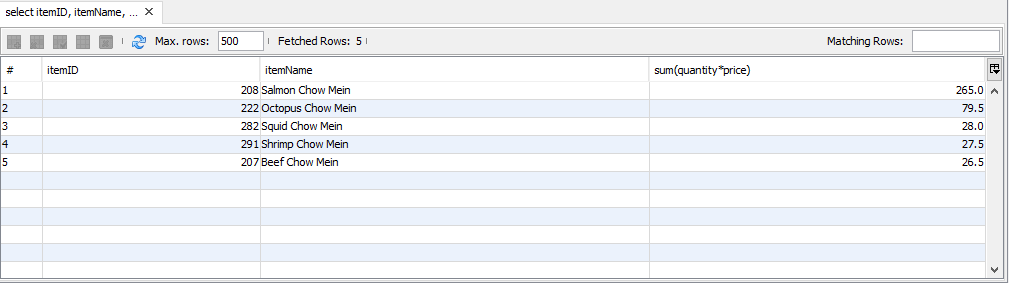
inner join Orderdetail using (itemID)

inner join Orders using(orderID)

group by itemID,itemName

order by sum(price) desc

limit 5;

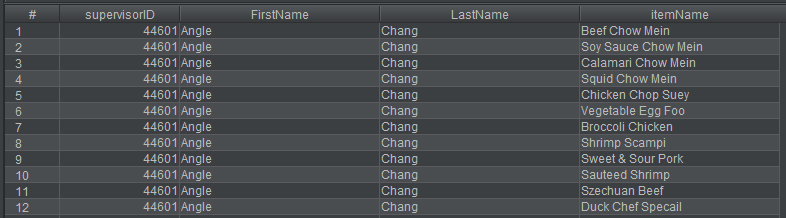


1. Find the sous chef who is mentoring the most other sous chef. List the menu items that the sous chef is passing along to the other sous chefs.

Select Distinct supervisorID, FirstName, LastName, itemName from Mentoring NATURAL JOIN Menuitem join Employee on Employee.empID = Mentoring.supervisorID

where supervisorID = (Select supervisorID from Mentoring group by supervisorID having count(supervisorID) =

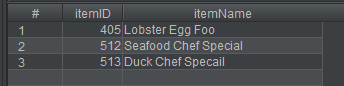
(SELECT MAX(y.amount) FROM (SELECT COUNT(supervisorID) AS amount from Mentoring group by supervisorID) y) ) ;



1. Find the three menu items that have the fewest sous chefs skilled in those menu items.

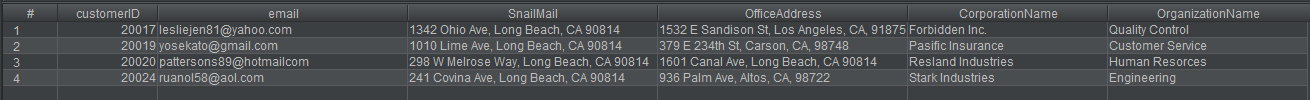
SELECT distinct itemID, itemName from Mentoring NATURAL JOIN Menuitem where itemID IN (Select itemID from Mentoring group by itemID having count(itemID) =

(SELECT MIN(y.skilled) FROM (SELECT COUNT(\*) AS skilled FROM Mentoring GROUP BY itemID) y)) limit 3;



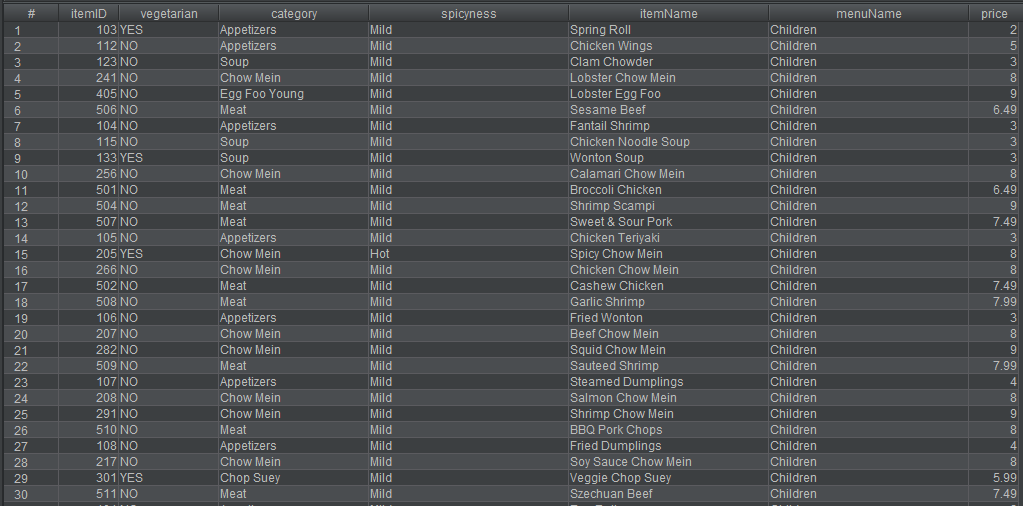
1. List all of the customers who eat at Miming’s on their own as well as ordering for their corporation.

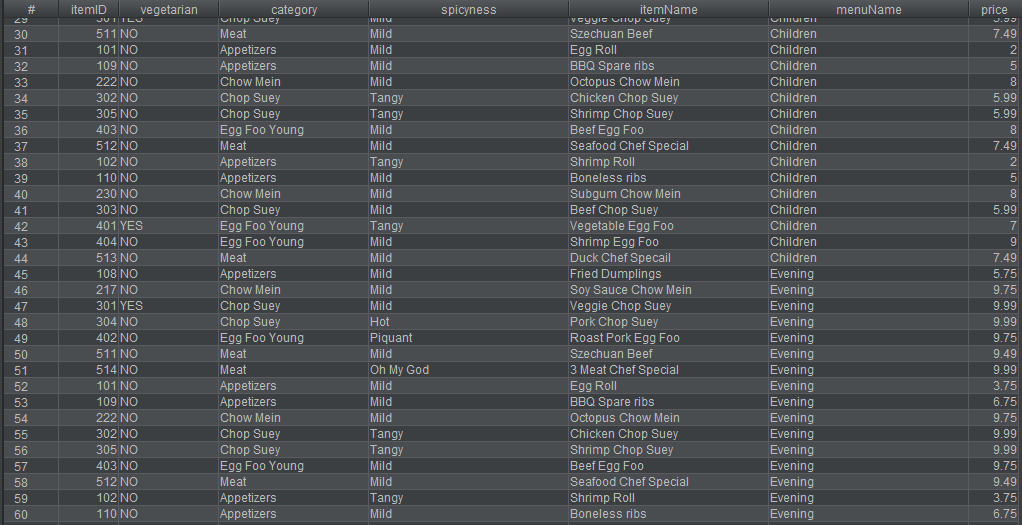
SELECT \* FROM Privatecustomer natural join Corporate WHERE customerID IN (SELECT customerID FROM Corporate);

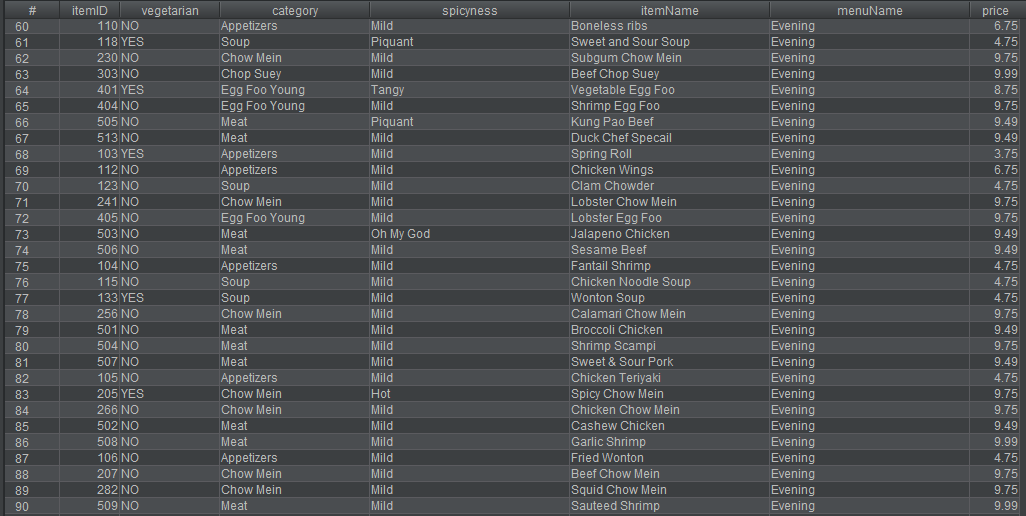


1. List the contents and prices of each of the menus.

SELECT \* FROM Menuitem NATURAL JOIN Orderselect ORDER BY menuName;













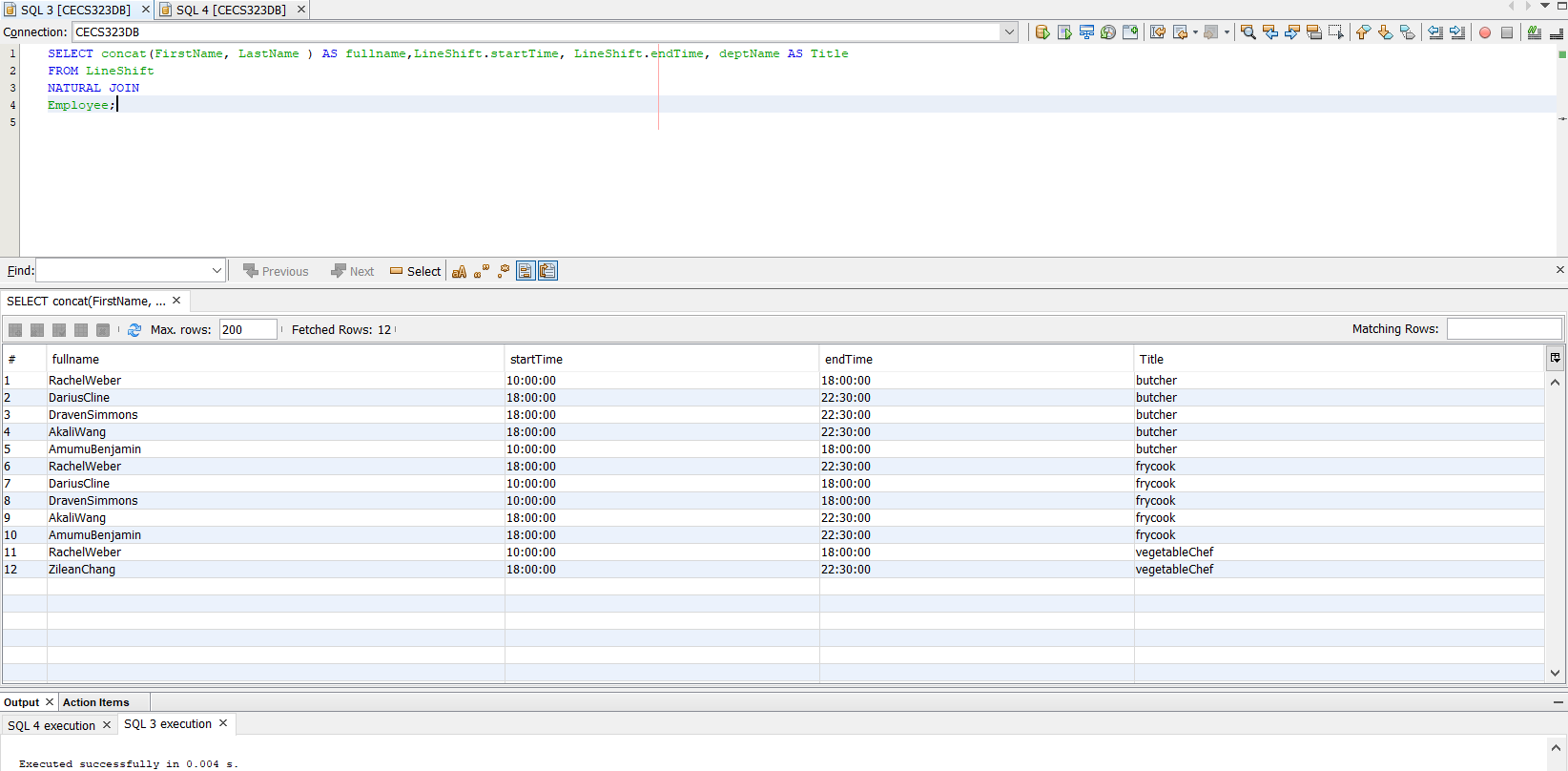
1. List each of the different shifts, who works during that shift and what their job is for that shift

SELECT concat(FirstName, LastName ) AS fullname,LineShift.startTime, LineShift.endTime, deptName AS Title

FROM LineShift

NATURAL JOIN

Employee;



1. The maximum number of seats a table cannot exceed 8

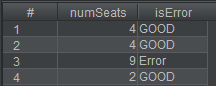
SELECT numSeats,

CASE WHEN numSeats > 8 THEN 'Error'

ELSE 'GOOD'

END AS isError

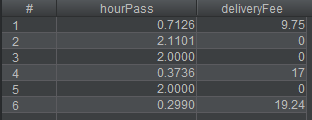
FROM Restauranttable;



1. If delivery>1.5 hour, order is free

SELECT (Delivery.timedelivered - Orders.orderTime)/10000 AS hourPass, Delivery.deliveryFee

From Delivery NATURAL JOIN Orders;



1. List all the vegetarian menu

SELECT itemName AS vegetarian\_menu From Menuitem WHERE vegetarian = 'YES';

