DB Assignment 2

Erin Zahner

17 September 2024

Part II

Problem 1: Average Price of Food at Each Restaurant

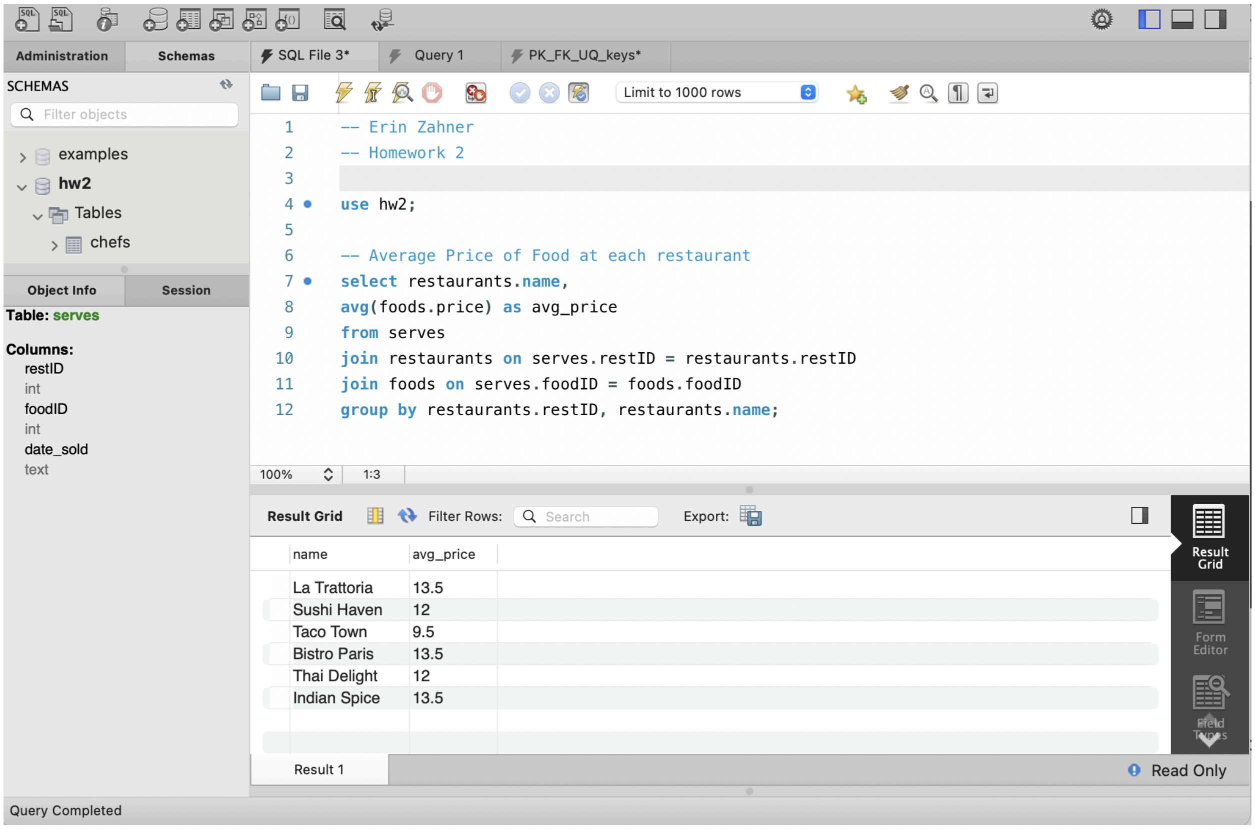


Table: serves

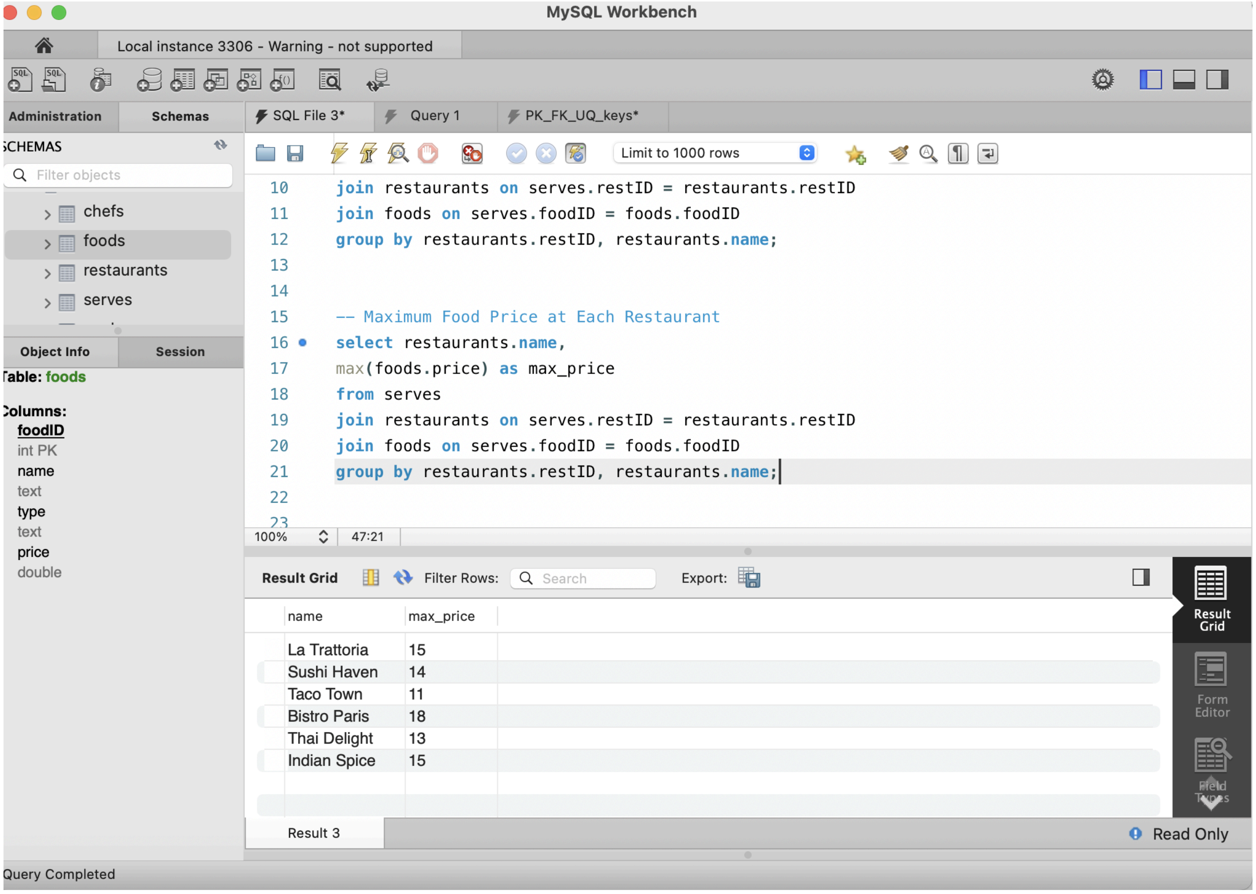
Columns:
restID
int
foodID
int
date_sold
text

```
-- Erin Zahner
-- Homework 2

use hw2;

-- Average Price of Food at each restaurant
select restaurants.name,
avg(foods.price) as avg_price
from serves
join restaurants on serves.restID = restaurants.restID
join foods on serves.foodID = foods.foodID
group by restaurants.restID, restaurants.name;
```

name	avg_price
La Trattoria	13.5
Sushi Haven	12
Taco Town	9.5
Bistro Paris	13.5
Thai Delight	12
Indian Spice	13.5

In this screenshot, you can see both the query statement used to solve this problem, as well as its output. To produce the average price of food at each restaurant, I first selected the restaurant name and the average of the food price from the “serves” table (as seen in the first three lines above). Then, in order to do the operation, I performed joins on the tables, joining the restIDs and foodIDs from serves to the IDs in their respective tables. Finally, I grouped them by the restaurant ID and name. In the result, you can see that the average price of food is listed for each restaurant.

Problem 2: Maximum Food Price at Each Restaurant

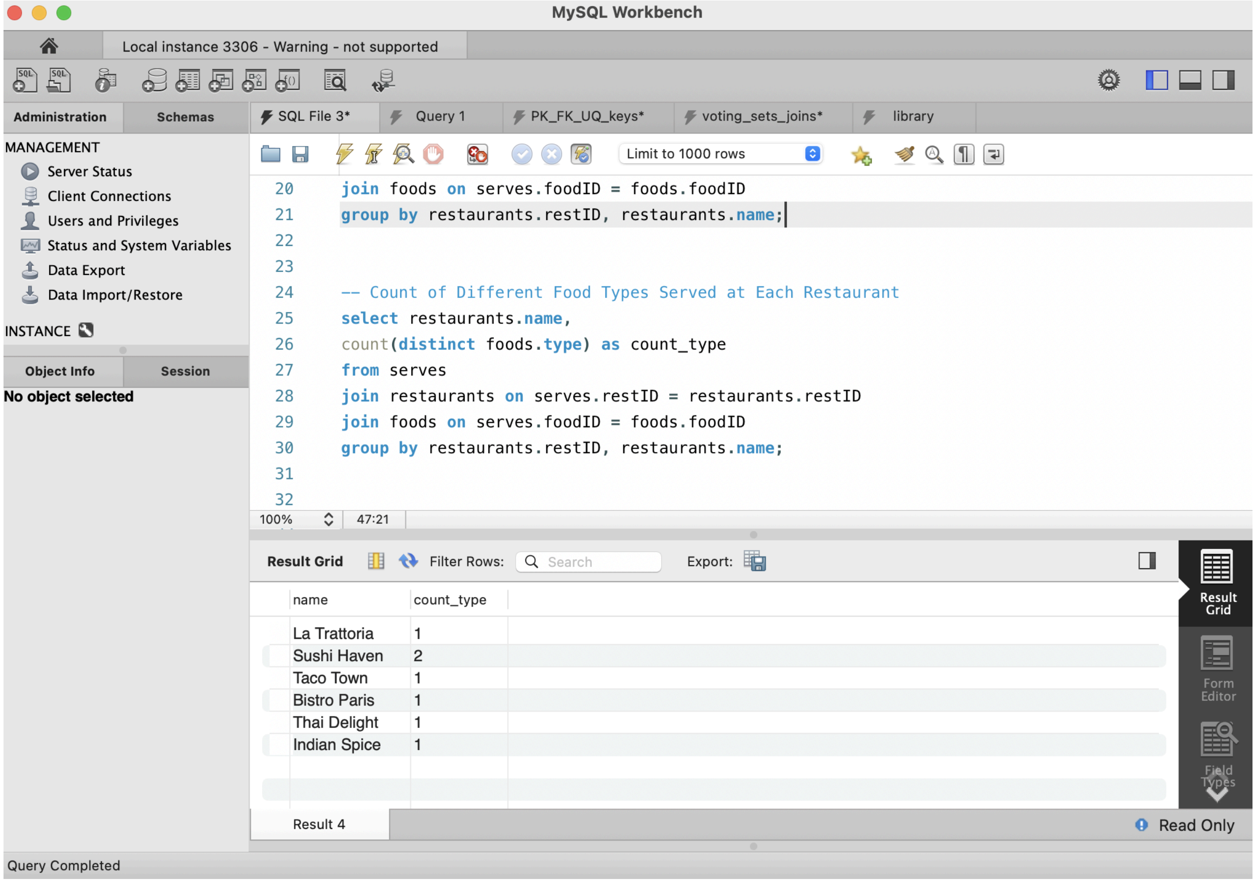


```
join restaurants on serves.restID = restaurants.restID
join foods on serves.foodID = foods.foodID
group by restaurants.restID, restaurants.name;


-- Maximum Food Price at Each Restaurant
select restaurants.name,
max(foods.price) as max_price
from serves
join restaurants on serves.restID = restaurants.restID
join foods on serves.foodID = foods.foodID
group by restaurants.restID, restaurants.name;
```

name	max_price
La Trattoria	15
Sushi Haven	14
Taco Town	11
Bistro Paris	18
Thai Delight	13
Indian Spice	15

This problem was very similar to the last one, only needing to change the aggregate function from avg to max. Other than that, all the lines were the same. By change avg to max, the result shows the maximum price of food at each restaurant.

Problem 3: Count of Different Food Types Served at Each Restaurant

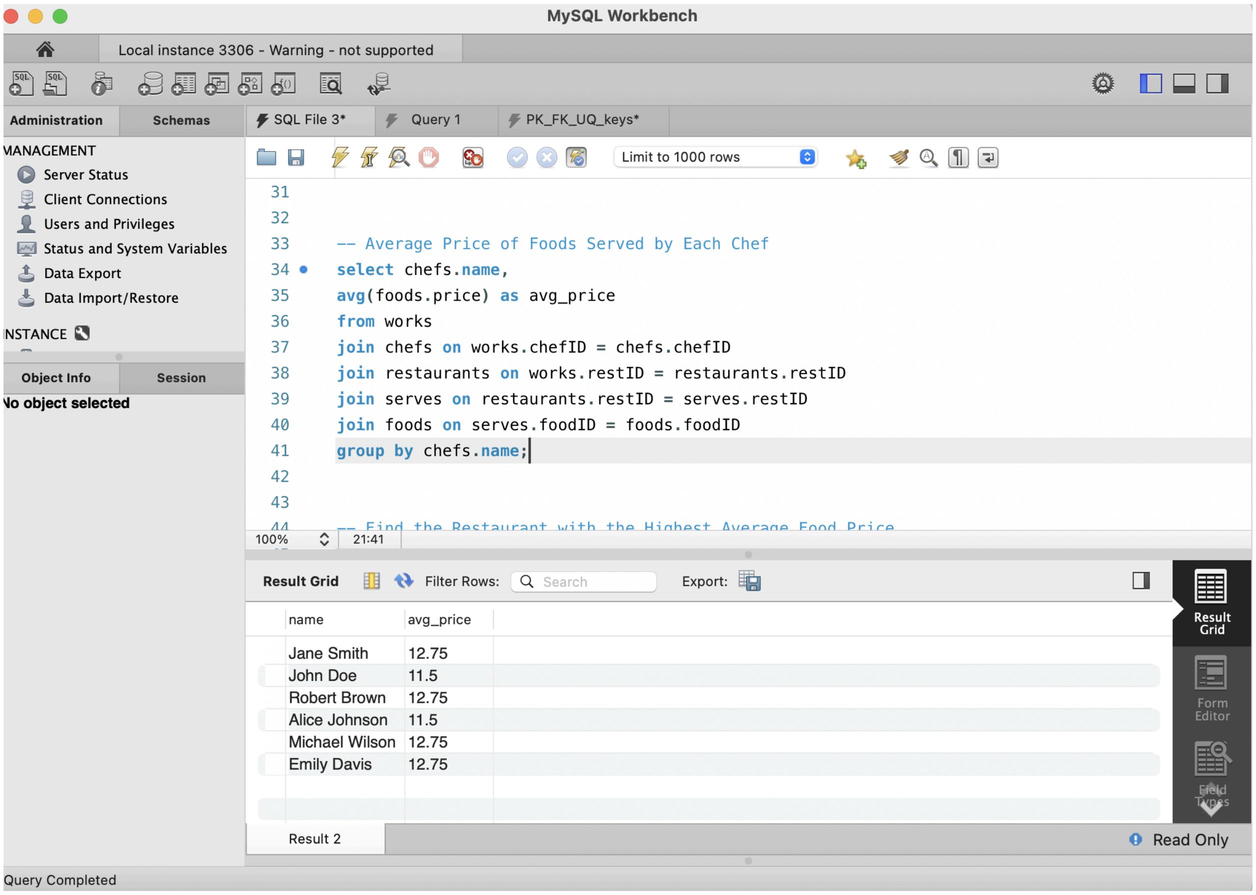


```sql
join foods on serves.foodID = foods.foodID
group by restaurants.restID, restaurants.name;


-- Count of Different Food Types Served at Each Restaurant
select restaurants.name,
count(distinct foods.type) as count_type
from serves
join restaurants on serves.restID = restaurants.restID
join foods on serves.foodID = foods.foodID
group by restaurants.restID, restaurants.name;
```

name	count_type
La Trattoria	1
Sushi Haven	2
Taco Town	1
Bistro Paris	1
Thai Delight	1
Indian Spice	1

Again, this problem is very similar to the last two, so most of the lines remained the same. Here, we only needed to change the second line to use the aggregate function count() on the column foods.type. However, it is important that we include “distinct” while counting so each food type is only counted once. By doing this, the result shows the number of different foods served at each restaurant.

Problem 4: Average Price of Foods Served by Each Chef

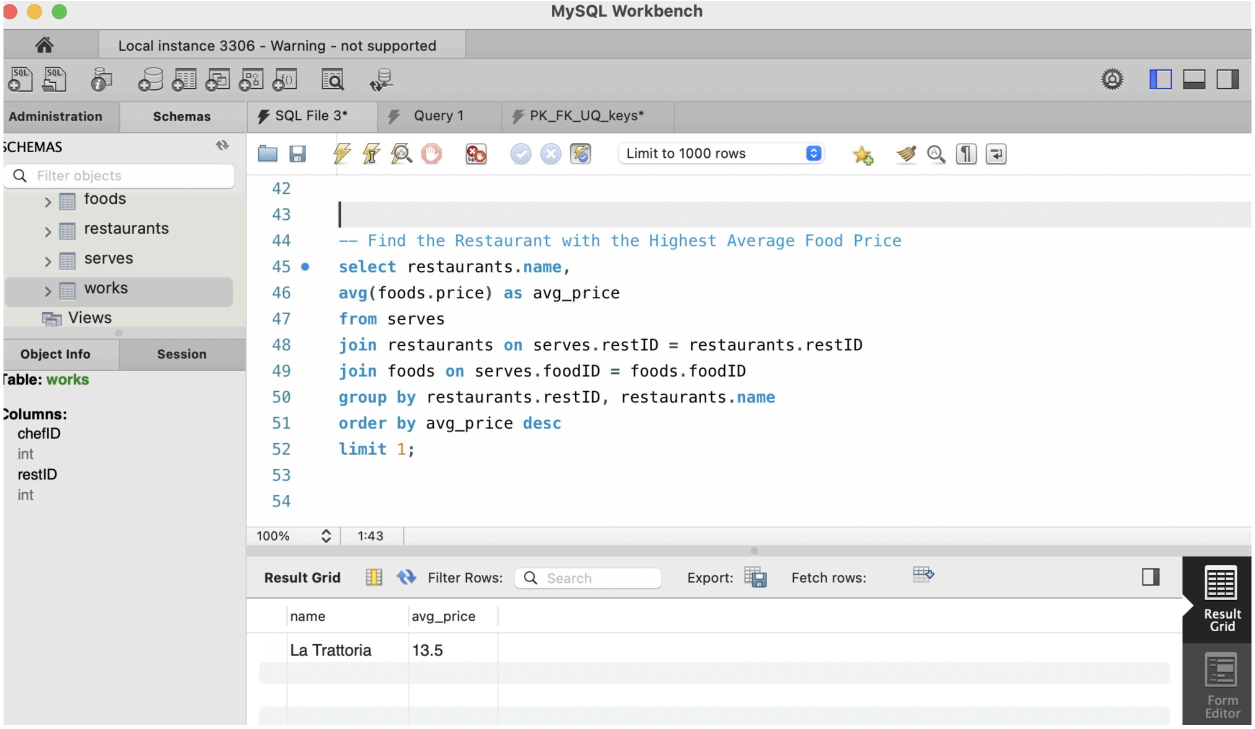


```
-- Average Price of Foods Served by Each Chef
select chefs.name,
avg(foods.price) as avg_price
from works
join chefs on works.chefID = chefs.chefID
join restaurants on works.restID = restaurants.restID
join serves on restaurants.restID = serves.restID
join foods on serves.foodID = foods.foodID
group by chefs.name;


-- Find the Restaurant with the Highest Average Food Price
```

name	avg_price
Jane Smith	12.75
John Doe	11.5
Robert Brown	12.75
Alice Johnson	11.5
Michael Wilson	12.75
Emily Davis	12.75

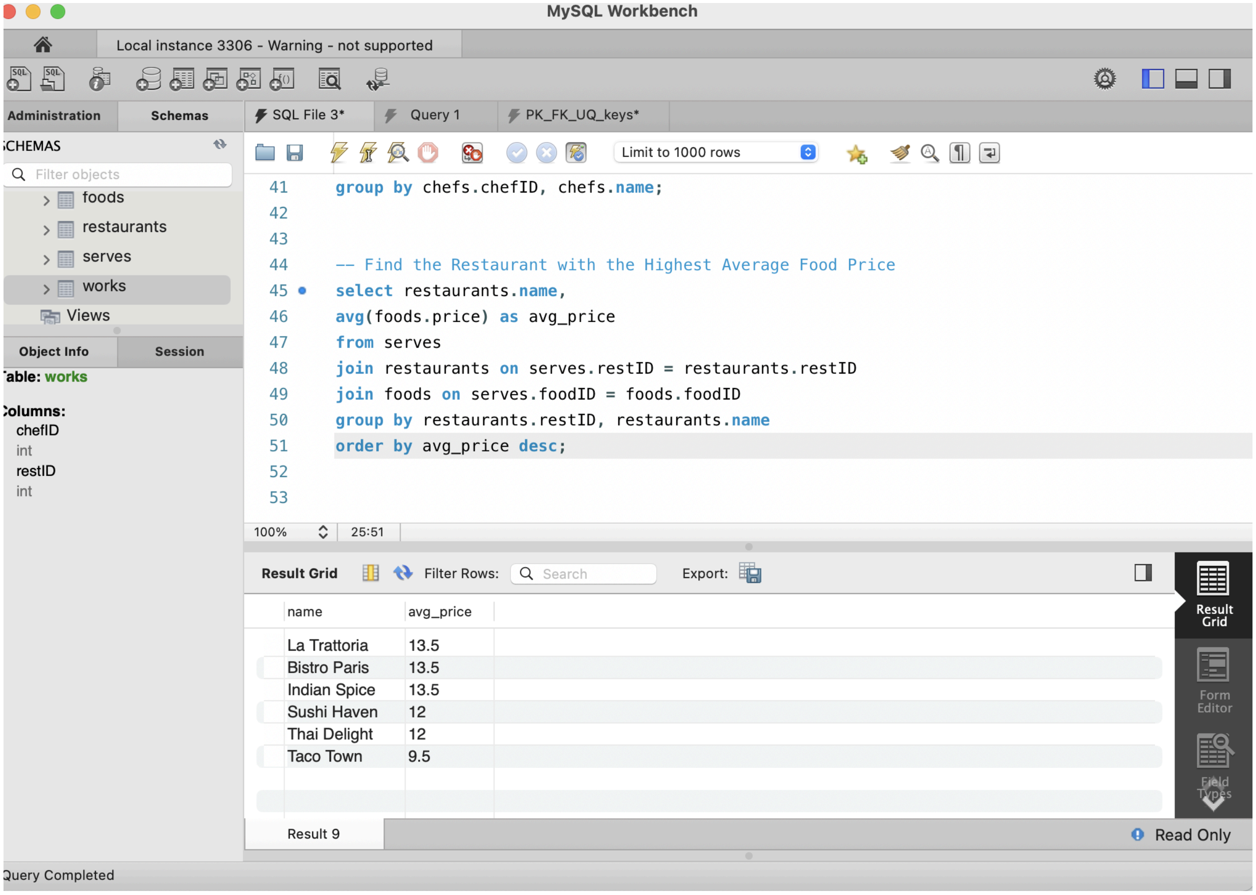
This problem got a bit more complicated. First, we have to select the names of the chefs and the average food price from works. Since there are no tables that directly connect chefs with foods, we have to do a lot of joins in order to use them together. To do this, we have to join the chefID from works to the chefID in chefs, the restID from works with the restID from restaurants and the restID from serves, and the foodID from serves with the foodID from foods. Finally, we group by chefs.name, and get a result that shows the average price of food served by each chef.

Problem 5: Find the Restaurant with the Highest Average Food Price



```
-- Find the Restaurant with the Highest Average Food Price
select restaurants.name,
avg(foods.price) as avg_price
from serves
join restaurants on serves.restID = restaurants.restID
join foods on serves.foodID = foods.foodID
group by restaurants.restID, restaurants.name
order by avg_price desc
limit 1;
```

name	avg_price
La Trattoria	13.5

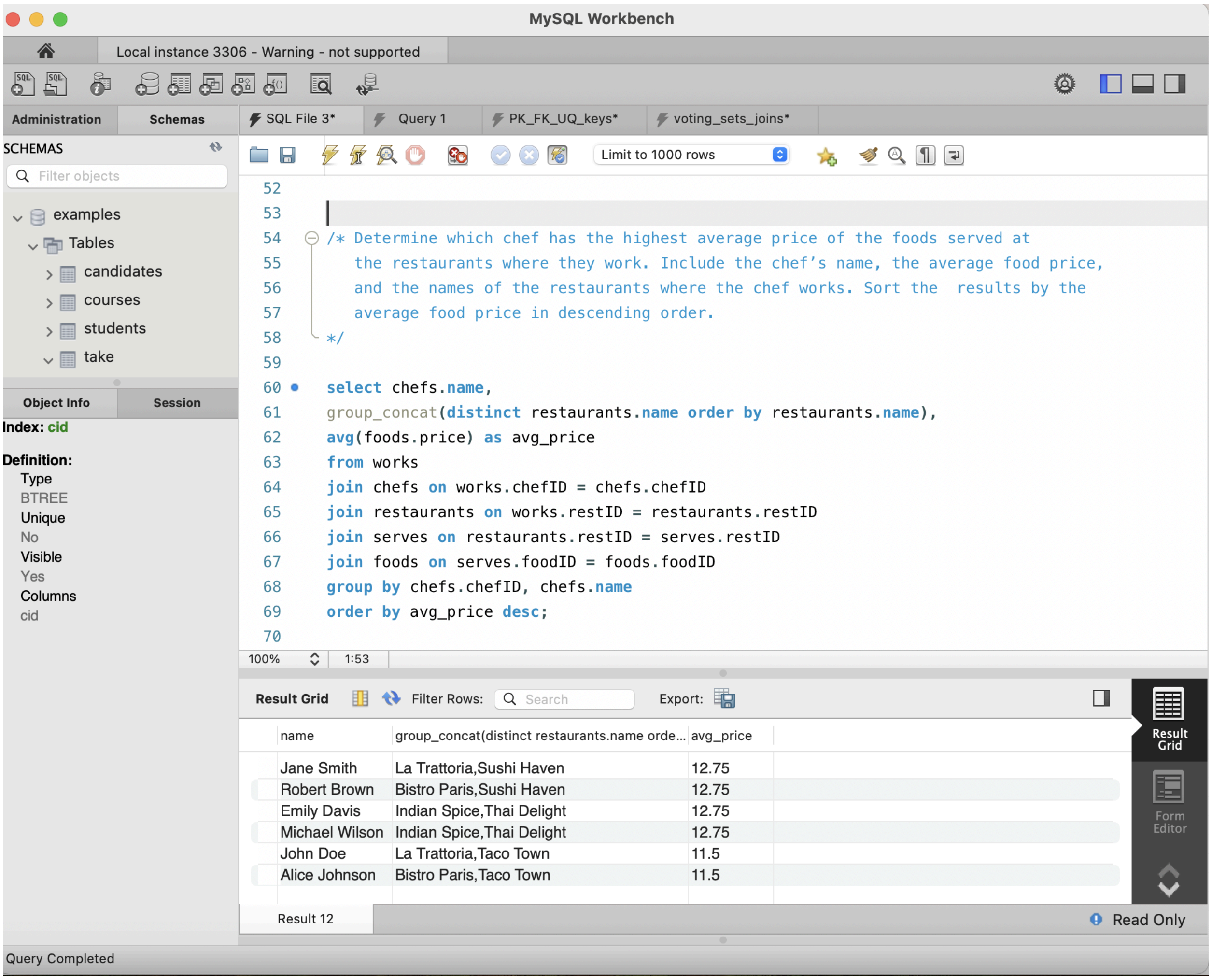


```
group by chefs.chefID, chefs.name;


-- Find the Restaurant with the Highest Average Food Price
select restaurants.name,
avg(foods.price) as avg_price
from serves
join restaurants on serves.restID = restaurants.restID
join foods on serves.foodID = foods.foodID
group by restaurants.restID, restaurants.name
order by avg_price desc;
```

name	avg_price
La Trattoria	13.5
Bistro Paris	13.5
Indian Spice	13.5
Sushi Haven	12
Thai Delight	12
Taco Town	9.5

This problem was almost the exact same as the first problem. The only thing we had to do instead was put the average price of foods in descending order, as can be seen in the last line above. Normally, when looking for the highest price, we want to limit the results to 1 in descending order, as seen in the first screenshot. However, for this problem, multiple restaurants had the highest average price of $13.50. As a result, I thought it was best to show all the results, as seen in the second screenshot.

Extra Credit: Determine which chef has the highest average price of the foods served at the restaurants where they work. Include the chef’s name, the average food price, and the names of the restaurants where the chef works. Sort the results by the average food price in descending order.



```
/* Determine which chef has the highest average price of the foods served at
   the restaurants where they work. Include the chef’s name, the average food price,
   and the names of the restaurants where the chef works. Sort the  results by the
   average food price in descending order.
*/

select chefs.name,
group_concat(distinct restaurants.name order by restaurants.name),
avg(foods.price) as avg_price
from works
join chefs on works.chefID = chefs.chefID
join restaurants on works.restID = restaurants.restID
join serves on restaurants.restID = serves.restID
join foods on serves.foodID = foods.foodID
group by chefs.chefID, chefs.name
order by avg_price desc;
```

name	group_concat(distinct restaurants.name orde...	avg_price
Jane Smith	La Trattoria,Sushi Haven	12.75
Robert Brown	Bistro Paris,Sushi Haven	12.75
Emily Davis	Indian Spice,Thai Delight	12.75
Michael Wilson	Indian Spice,Thai Delight	12.75
John Doe	La Trattoria,Taco Town	11.5
Alice Johnson	Bistro Paris,Taco Town	11.5

This extra credit problem was very similar to Question 4, but slightly more complicated. For this problem, I had to figure out how to incorporate the restaurants each chef works at. To do this, I did group_concat() (as seen in the second line). This allowed me to list the restaurant names, while making sure that each restaurant is only named once (distinct). After that, the rest was pretty much the same, with the addition of ordering average price in descending order. Due to all of the chefs having very similarly priced food served, We cannot determine one chef as serving the highest average food price.