

## DB Assignment 2

Erin Zahner

17 September 2024

### Part II

#### Problem 1: Average Price of Food at Each Restaurant

The screenshot shows a SQL IDE interface with a query editor and a results grid. The query editor contains the following SQL code:

```
1 -- Erin Zahner
2 -- Homework 2
3
4 • use hw2;
5
6 -- Average Price of Food at each restaurant
7 • select restaurants.name,
8   avg(foods.price) as avg_price
9   from serves
10  join restaurants on serves.restID = restaurants.restID
11  join foods on serves.foodID = foods.foodID
12  group by restaurants.restID, restaurants.name;
```

The results grid displays the output of the query, showing the average price of food at each restaurant. The columns are 'name' and 'avg\_price'.

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

The interface also shows a sidebar with a tree view of the database schema, including 'examples', 'hw2', 'Tables', and 'chefs'. The 'Object Info' tab is selected, showing details for the 'serves' table, including columns 'restID', 'foodID', and 'date\_sold'.

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

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' panel with a tree view containing 'chefs', 'foods', 'restaurants', and 'serves'. The 'Object Info' tab is selected, showing details for the 'foods' table: 'foodID' (int PK), 'name' (text), 'type' (text), and 'price' (double). The main query editor contains the following SQL code:

```
10 join restaurants on serves.restID = restaurants.restID
11 join foods on serves.foodID = foods.foodID
12 group by restaurants.restID, restaurants.name;
13
14
15 -- Maximum Food Price at Each Restaurant
16 • select restaurants.name,
17    max(foods.price) as max_price
18 from serves
19 join restaurants on serves.restID = restaurants.restID
20 join foods on serves.foodID = foods.foodID
21 group by restaurants.restID, restaurants.name;
22
23
```

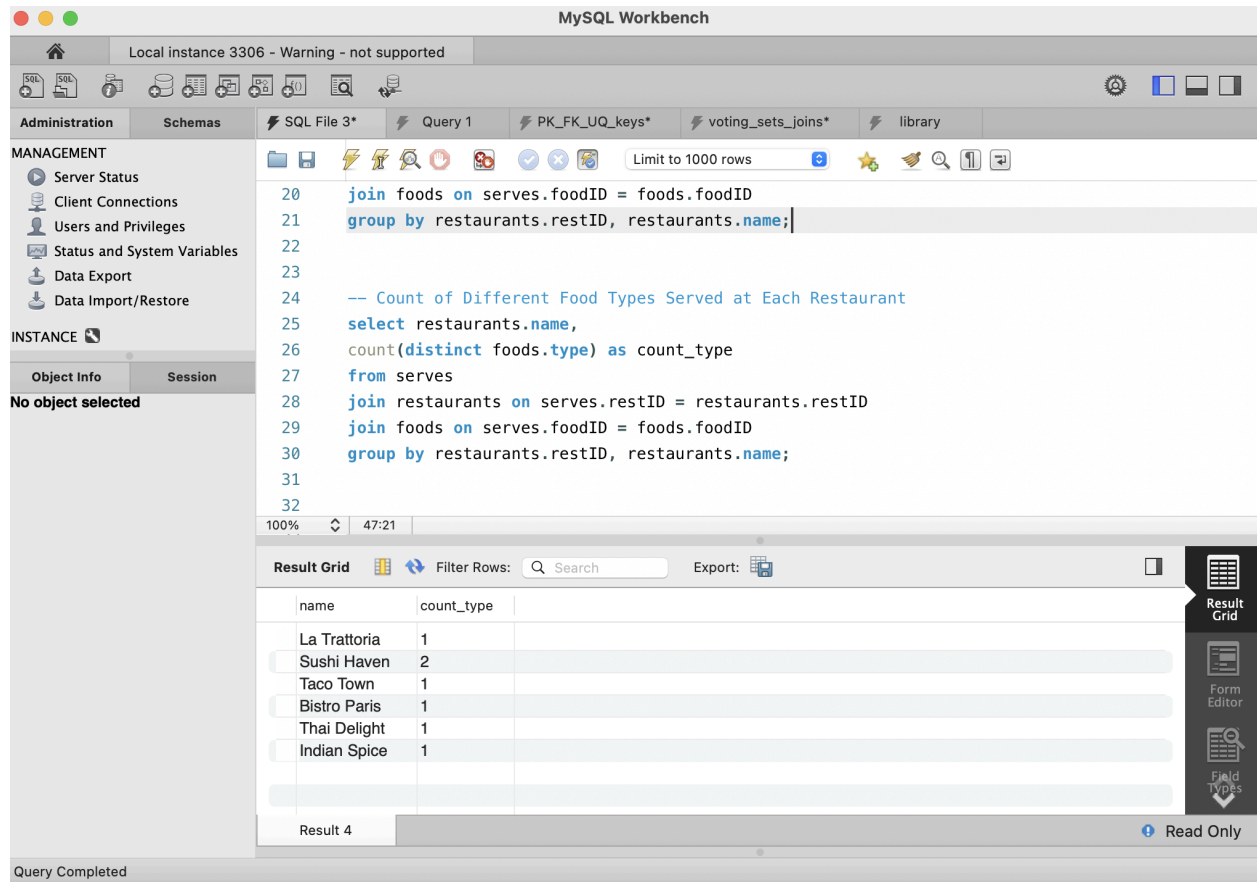
The 'Result Grid' at the bottom displays the results of the query. It shows a table with two columns: 'name' and 'max\_price'. The data rows are:

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

The status bar at the bottom indicates 'Query Completed'.

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



The screenshot shows the MySQL Workbench interface. The SQL editor contains the following query:

```
20 join foods on serves.foodID = foods.foodID
21 group by restaurants.restID, restaurants.name;
22
23
24 -- Count of Different Food Types Served at Each Restaurant
25 select restaurants.name,
26 count(distinct foods.type) as count_type
27 from serves
28 join restaurants on serves.restID = restaurants.restID
29 join foods on serves.foodID = foods.foodID
30 group by restaurants.restID, restaurants.name;
31
32
```

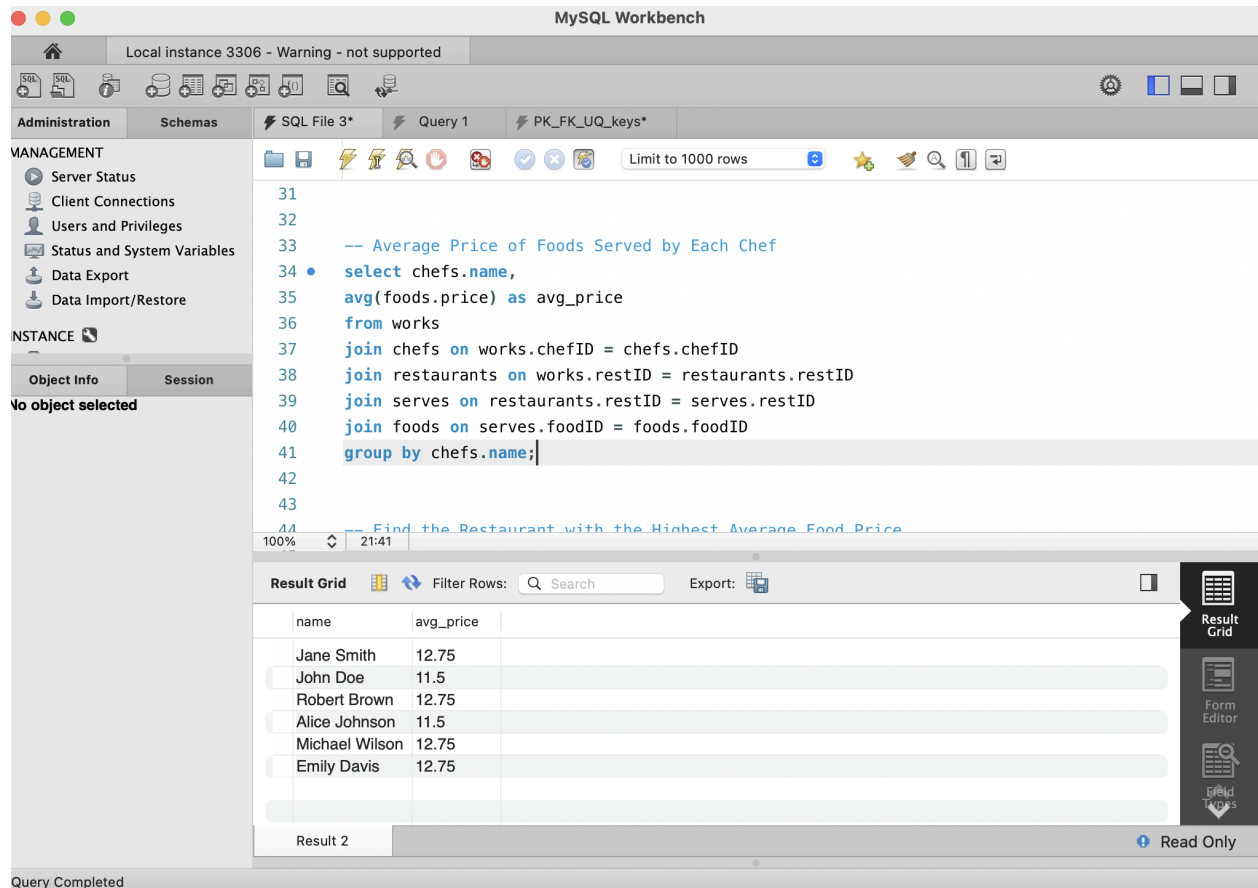
The query results are displayed in the Result Grid below the editor:

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

The status bar at the bottom indicates "Query Completed".

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



The screenshot shows the MySQL Workbench interface. The left sidebar contains 'MANAGEMENT' and 'INSTANCE' sections. The main window displays a SQL query in the 'Query 1' tab. The query is as follows:

```
31
32
33 -- Average Price of Foods Served by Each Chef
34 • select chefs.name,
35    avg(foods.price) as avg_price
36  from works
37  join chefs on works.chefID = chefs.chefID
38  join restaurants on works.restID = restaurants.restID
39  join serves on restaurants.restID = serves.restID
40  join foods on serves.foodID = foods.foodID
41  group by chefs.name;
42
43
44 -- Find the Restaurant with the Highest Average Food Price
```

Below the query, the 'Result Grid' is visible, showing the results of the query. The grid has two columns: 'name' and 'avg\_price'. The results are as follows:

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

The status bar at the bottom indicates 'Query Completed'.

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

MySQL Workbench

Local instance 3306 - Warning - not supported

Administration Schemas SQL File 3\* Query 1 PK\_FK\_UQ\_keys\*

SCHEMAS

Filter objects

- foods
- restaurants
- serves
- works
- Views

Object Info Session

Table: works

Columns:

- chefID int
- restID int

```
42
43
44 -- Find the Restaurant with the Highest Average Food Price
45 • select restaurants.name,
46    avg(foods.price) as avg_price
47 from serves
48 join restaurants on serves.restID = restaurants.restID
49 join foods on serves.foodID = foods.foodID
50 group by restaurants.restID, restaurants.name
51 order by avg_price desc
52 limit 1;
53
54
```

100% 1:43

Result Grid Filter Rows: Search Export: Fetch rows:

name	avg_price
La Trattoria	13.5

Result Grid Form Editor

MySQL Workbench

Local instance 3306 - Warning - not supported

Administration Schemas SQL File 3\* Query 1 PK\_FK\_UQ\_keys\*

SCHEMAS

Filter objects

- foods
- restaurants
- serves
- works
- Views

Object Info Session

Table: works

Columns:

- chefID int
- restID int

```
41 group by chefs.chefID, chefs.name;
42
43
44 -- Find the Restaurant with the Highest Average Food Price
45 • select restaurants.name,
46    avg(foods.price) as avg_price
47 from serves
48 join restaurants on serves.restID = restaurants.restID
49 join foods on serves.foodID = foods.foodID
50 group by restaurants.restID, restaurants.name
51 order by avg_price desc;
52
53
```

100% 25:51

Result Grid Filter Rows: Search Export:

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

Result 9 Read Only

Query Completed

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.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'examples' expanded, showing 'Tables' like 'candidates', 'courses', 'students', and 'take'. The 'Object Info' tab is selected, showing details for the 'cid' index. The main editor displays a SQL query (lines 52-70) that determines the chef with the highest average food price. The query uses a subquery to calculate the average price for each chef and then joins it with the 'works' table to get the chef's name and the restaurant names. The results are sorted by the average price in descending order.

```
52
53
54 /* Determine which chef has the highest average price of the foods served at
55    the restaurants where they work. Include the chef's name, the average food price,
56    and the names of the restaurants where the chef works. Sort the results by the
57    average food price in descending order.
58 */
59
60 • select chefs.name,
61    group_concat(distinct restaurants.name order by restaurants.name),
62    avg(foods.price) as avg_price
63 from works
64 join chefs on works.chefID = chefs.chefID
65 join restaurants on works.restID = restaurants.restID
66 join serves on restaurants.restID = serves.restID
67 join foods on serves.foodID = foods.foodID
68 group by chefs.chefID, chefs.name
69 order by avg_price desc;
70
```

The 'Result Grid' at the bottom shows the results of the query. It displays 6 rows of data, with the first 5 rows having an average price of 12.75 and the last row having an average price of 11.5. The results are sorted by the average price in descending order.

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

Result 12

Query Completed

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.