

## DB Assignment 2

Christina DiMaggio

September 26, 2024

### 1. Average Price of Food at Each Restaurant

#### Query Results:

The screenshot shows the MySQL Workbench interface. The SQL editor contains a query to calculate the average price of food at each restaurant. The query uses an inner join between the restaurants, serves, and foods tables. The results are displayed in a table with two columns: restaurant\_name and avg\_food\_price.

```
10
11 -- 1. Average Price of Foods at Each Restaurant
12 -- restaurants to join with food and serves
13 -- chain broken if there is no link with primary keys, everything has to be connected
14 -- serves has foreign keys
15 -- need names of the restaurants
16 -- need the price of the food
17 -- need to know what food is served in restaurants
18
19 select restaurants.name as restaurant_name, avg(foods.price) as avg_food_price -- alias for restaurants.name and avg(foods.price)
20 from restaurants -- universal set
21 inner join serves on restaurants.restID = serves.restID
22 inner join foods on serves.foodID = foods.foodID
23 group by restaurant_name;
24
25
```

restaurant_name	avg_food_price
La Trattoria	13.5
Sushi Haven	12
Taco Town	9.5
Bistro Paris	13.5
Thai Delight	12
Indian Spice	13.5

The bottom of the screenshot shows the Action Output pane with a list of actions and their results. The actions include using examples, selecting restaurants, and selecting chefs. The results show the number of rows affected and the duration of each action.

What does query do? How does it solve the problem?

This query can INNER join restaurants with serves and food by utilizing their primary keys and or foreign keys. Inner joins are for exactly which columns we need and are using. Comments on SQL queries also better define what codes represent.

The query nicely lays out the name of the restaurant on the left and its respective average price for food on the right of the table.

## 2. Maximum Food Price at Each Restaurant

### Query Results:

The screenshot shows the MySQL Workbench interface. The SQL editor contains a query to find the maximum food price at each restaurant. The query is as follows:

```
-- 2. Maximum Food Price at Each Restaurant
-- restaurants to join with food and serves
-- chain broken if there is no link with primary keys, everything has to be connected
-- serves has foreign keys
-- need names of the restaurants
-- need the maximum price of the food
-- need to know what food is served in restaurants

select restaurants.name as restaurant_name, max(foods.price) as max_food_price -- alias for restaurants.name and max(foods.price)
from restaurants -- universal set
inner join serves on restaurants.restID = serves.restID
inner join foods on serves.foodID = foods.foodID
group by restaurant_name; -- way to summarize value and findings
-- looking for aggregate needs grouping
```

The results are displayed in a table with the following data:

restaurant_name	max_food_price
La Trattoria	15
Sushi Haven	14
Taco Town	11
Bistro Paris	18
Thai Delight	13
Indien Spice	15

The bottom panel shows the Action Output with the following messages:

#	Time	Action	Message	Duration / Fetch
1	22:30:44	use examples	0 row(s) affected	0.000 sec
2	22:30:44	select restaurants name as restaurant_name, avg(foods.price) as avg_food_price -- alias for restaurants.name and ...	6 row(s) returned	0.000 sec / 0.000 sec
3	22:30:44	select restaurants name as restaurant_name, max(foods.price) as max_food_price -- alias for restaurants.name and ...	6 row(s) returned	0.000 sec / 0.000 sec
4	22:30:44	select restaurants name as restaurants_name, count(foods.type) as food_types -- alias for restaurants.name and co...	6 row(s) returned	0.000 sec / 0.000 sec
5	22:30:44	select chefs.name as chef_name, avg(foods.price) as average_price -- alias for chefs.name and avg(foods.price) fr...	6 row(s) returned	0.000 sec / 0.000 sec

**What does query do? How does it solve the problem?**

This query can INNER join restaurants with serves and food by utilizing their primary keys and or foreign keys. Inner joins are for exactly which columns we need and are using. Comments on SQL queries also better define what codes represent.

The query nicely lays out the restaurant's name on the left and the maximum amount of money you could spend on one food item sold there on the right.

### 3. Count of Different Food Types Served at Each Restaurant

#### Query Results:

The screenshot shows the MySQL Workbench interface. The SQL editor contains a query that counts the number of different food types served at each restaurant. The query is as follows:

```
-- 3. Count of Different Food Types Served at Each Restaurant
-- restaurants to join with food and serves
-- chain broken if there is no link with primary keys, everything has to be connected
-- serves has foreign keys
-- need names of the restaurants
-- need types of the food (COUNT, how many?)
-- need to know what food is served in restaurants

54: select restaurants.name as restaurants_name, count(foods.type) as food_types -- alias for restaurants.name and count(foods.type)
55: from restaurants -- universal set
56: inner join serves on restaurants.restID = serves.restID
57: inner join foods on serves.foodID = foods.foodID
58: group by restaurants_name; -- way to summarize value and findings
-- looking for aggregate needs grouping
```

The results are displayed in a table with two columns: **restaurants\_name** and **food\_types**. The data is as follows:

restaurants_name	food_types
La Trattoria	2
Sushi Haven	2
Taco Town	2
Bistro Paris	2
Thai Delight	2
Indian Spice	2

The bottom of the screenshot shows the 'Action Output' pane, which displays the execution of the query. The output shows that the query was executed successfully, returning 6 rows of data.

**What does query do? How does it solve the problem?**

This query can INNER join restaurants with serves and food by utilizing their primary keys and or foreign keys. Inner joins are for exactly which columns we need and are using. Comments on SQL queries also better define what codes represent.

The query nicely lays out the name of the restaurant on the left and how many types of food they serve on the right of the table.

#### 4. Average Price of Foods Served by Each Chef

##### Query Results:

The screenshot displays the MySQL Workbench interface. The SQL editor contains a query to calculate the average price of foods served by each chef. The query uses multiple INNER JOINs to connect the chefs, works, restaurants, serves, and foods tables. The results are shown in a table with two columns: chef\_name and average\_price.

```
61
62 -- 4. Average Price of Foods Served by Each Chef
63 -- chefs to join with works, restaurants, serves, foods
64 -- chain broken if there is no link with primary keys, everything has to be connected
65 -- serves has foreign keys
66 -- need names of the chefs and what restaurant they work at
67 -- need to know types of the food served at the restaurants
68 -- need to be able to calculate average price of foods
69 select chefs.name as chef_name, avg(foods.price) as average_price
70 from chefs
71 inner join works on chefs.chefID = works.chefID
72 inner join restaurants on works.restID = restaurants.restID
73 inner join serves on restaurants.restID = serves.restID
74 inner join foods on serves.foodID = foods.foodID
75 group by chef_name;
76
```

chef_name	average_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 bottom panel shows the Action Output with a list of executed queries and their durations.

Time	Action	Message	Duration / Fetch
22:30:44	use examples	0 row(s) affected	0.000 sec
22:30:44	select restaurants name as restaurant_name, avg(foods price) as avg_food_price -- alias for restaurants name and ...	6 row(s) returned	0.000 sec / 0.000 sec
22:30:44	select restaurants name as restaurant_name, max(foods price) as max_food_price -- alias for restaurants name and ...	6 row(s) returned	0.000 sec / 0.000 sec
22:30:44	select restaurants name as restaurants_name, count(foods type) as food_types -- alias for restaurants name and co...	6 row(s) returned	0.000 sec / 0.000 sec
22:30:44	select chefs name as chef_name, avg(foods price) as average_price -- alias for chefs name and avg(foods price) fr...	6 row(s) returned	0.000 sec / 0.000 sec

**What does query do? How does it solve the problem?**

**This query can INNER join chefs with works, restaurants, serves, and food by utilizing their primary keys and or foreign keys. Inner joins are for exactly which columns we need and are using. Comments on SQL queries also better define what codes represent.**

**The query nicely lays out the name of the chef on the left and the respective average price for the food they serve on the right of the table.**

## 5. Find the Restaurant with the Highest Average Food Price

### Query Results:

The screenshot shows the MySQL Workbench interface. The main editor displays a SQL query with detailed comments explaining each step. The query uses an INNER JOIN to connect the restaurants, serves, and foods tables, then groups the results by restaurant name and orders them by average food price in descending order, limiting the output to three rows.

```
-- 5. Find the Restaurant with the Highest Average Food Price
-- restaurants to join with food and serves
-- chain broken if there is no link with primary keys, everything has to be connected
-- serves has foreign keys
-- need names of the restaurants
-- need the price of the food
-- need to know what food is served in restaurants

select restaurants.name as restaurant_name, avg(foods.price) as avg_food_price
from restaurants
inner join serves on restaurants.restID = serves.restID
inner join foods on serves.foodID = foods.foodID
group by restaurants.name
order by avg_food_price desc -- running up to this line you run at this line you will see three are tied for highest average food price
-- this is a special case where we can limit to three
-- from the line 'order...' it is known that La Trattoria, Bistro Paris, and Indian Spice
-- all have the highest average food price
-- thus why we limit to 3
-- only showing these restaurants in a tied for highest average food prices.

limit 3;
```

The results pane shows the following data:

restaurant_name	avg_food_price
La Trattoria	13.5
Bistro Paris	13.5
Indian Spice	13.5

The bottom pane shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
1	22:30:44	Use examples	0 row(s) affected	0.000 sec
2	22:30:44	select restaurants name as restaurant_name, avg(foods price) as avg_food_price -- alias for restaurants name and ...	6 row(s) returned	0.000 sec / 0.000 sec
3	22:30:44	select restaurants name as restaurant_name, max(foods price) as max_food_price -- alias for restaurants name and ...	6 row(s) returned	0.000 sec / 0.000 sec
4	22:30:44	select restaurants name as restaurant_name, count(foods type) as food_types -- alias for restaurants name and co...	6 row(s) returned	0.000 sec / 0.000 sec
5	22:30:44	select chefs name as chef_name, avg(foods price) as average_price -- alias for chefs name and avg(foods price) fr...	6 row(s) returned	0.000 sec / 0.000 sec

### What does query do? How does it solve the problem?

This query can INNER join restaurants with servers and food by utilizing their primary keys and or foreign keys. Inner joins are for exactly which columns we need and are using. Comments on SQL queries also better define what codes represent.

When utilizing the limit function (see comment) query nicely lays out the name of the three restaurants on the left showing they have the highest average food prices at their restaurant on the right of the table.



6. Extra Credit: Which chef has the highest average price of the foods served at the restaurants where they work?

Query Results:

The screenshot shows the MySQL Workbench interface. The SQL editor contains a query to find the chef with the highest average food price. The query uses INNER JOINs to connect chefs, works, restaurants, and foods tables. The results are displayed in a table with columns: restaurant\_name, chef\_name, and average\_price. The output shows 12 rows of data, with the highest average price being 13.5, shared by Jane Smith and John Doe at La Trattoria.

```
96 -- 6. Extra Credit: Which chef has the highest average price of the foods served at the restaurants where they work?
97 -- chefs to join with works, restaurants, serves, foods
98 -- chain broken if there is no link with primary keys, everything has to be connected
99 -- serves has foreign keys
100 -- need names of the chefs and what restaurant they work at
101 -- need to know types of the food served at the restaurants
102 -- need to be able to calculate average price of foods
103
104 select restaurants.name as restaurant_name, chefs.name as chef_name, avg(foods.price) as average_price -- alias for restaurants.name, chefs.name and avg(foods.price)
105 from chefs -- universal set
106 inner join works on chefs.chefID = works.chefID
107 inner join restaurants on works.restID = restaurants.restID
108 inner join serves on restaurants.restID = serves.restID
109 inner join foods on serves.foodID = foods.foodID
110 group by chef_name, restaurant_name -- way to summarize value and findings
111 -- looking for aggregate needs grouping
112 order by average_price desc;
```

restaurant_name	chef_name	average_price
La Trattoria	Jane Smith	13.5
La Trattoria	John Doe	13.5
Bistro Paris	Alice Johnson	13.5
Bistro Paris	Robert Brown	13.5
Indian Spice	Emily Davis	13.5
Indian Spice	Michael Wilson	13.5
Sushi Haven	Robert Brown	12
Sushi Haven	Jane Smith	12
Thai Delight	Michael Wilson	12
Thai Delight	Emily Davis	12
Taco Town	John Doe	9.5
Taco Town	Alice Johnson	9.5

What does query do? How does it solve the problem?

This query can INNER join from chefs to relate works, restaurants, servers and food by utilizing their primary keys and or foreign keys. Inner joins are for exactly which columns we need and are using. Comments on SQL queries also better define what codes represent.

The query nicely lays out the names of the restaurants on the far left and their respective chefs in the middle. The right column shows the average of how much the food costs that each chef cooks in the restaurant.