Title Page

- **Title:** DB Assignment 2
- **Due Date:** 30 September 2025

SQL Section

1. Average Price of Foods at Each Restaurant

```
3    -- NUMBER 1
4    -- Selecting the restaurant and their average price
5 •  select r.name as Restaurant,
6        avg(f.Price) as AvgPrice
7    from serves s
8    inner join restaurants r
9        on s.restID = r.restID
10   inner join foods f
11       on s.foodID = f.foodID
12   group by r.name;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Restaurant	AvgPrice
La Trattoria	13.5
Sushi Haven	12
Taco Town	9.5
Bistro Paris	13.5
Thai Delight	12
Indian Spice	13.5

Explanation: In this query, I joined the serves, restaurants, and foods tables together. To calculate the average food price, grouping by restaurant name was necessary. The restaurant name and the average food price are returned in the output.

2. Maximum Food Price at Each Restaurant

```
16    -- NUMBER 2
17    -- Selecting the restaurant and their highest price
18 •  select r.name as Restaurant,
19        max(f.price) as MaxPrice
20    from serves s
21    inner join restaurants r
22        on s.restID = r.restID
23    inner join foods f
24        on s.foodID = f.foodID
25    group by r.name;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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

Explanation: In this query, I joined the serves, restaurants, and foods tables together once again. To calculate the highest price at each restaurant, grouping by restaurant name was necessary again. The restaurant name and the maximum food price are returned in the output.

3. Count of Different Food Types Served at Each Restaurant

```
28    -- NUMBER 3
29    -- Selecting the count of different food types at each restaurant
30 •  select r.name as Restaurant,
31        count(distinct(f.name)) as NumberItems,
32        f.type as FoodType
33    from serves s
34    inner join restaurants r
35        on s.restID = r.restID
36    inner join foods f
37        on s.foodID = f.foodID
38    group by r.name, f.type;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Restaurant	NumberItems	FoodType
Bistro Paris	2	French
Indian Spice	2	Indian
La Trattoria	2	Italian
Sushi Haven	1	Rice
Sushi Haven	1	Seafood
Taco Town	2	Mexican
Thai Delight	2	Thai

Explanation: The serves, restaurants, and foods tables were joined together to answer this question. To calculate the number of different food types at each restaurant, grouping by restaurant name as well as food type was necessary. The restaurant name, the count of distinct food items, and the food types were selected to be returned in the output.

4. Average Price of Foods Served by Each Chef

```
41    -- NUMBER 4
42    -- Selecting the average food price for each chef
43 •  select c.name as Chef,
44        avg(f.price) as AvgPrice
45    from serves s
46    inner join restaurants r
47        on s.restID = r.restID
48    inner join foods f
49        on s.foodID = f.foodID
50    inner join works w
51        on w.restID = r.restID
52    inner join chefs c
53        on c.chefID = w.chefID
54    group by c.name;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Chef	AvgPrice
John Doe	11.5
Jane Smith	12.75
Robert Brown	12.75
Alice Johnson	11.5
Emily Davis	12.75
Michael Wilson	12.75

Explanation: The serves, restaurants, foods, works, and chefs tables were joined together to answer this question due to the schema design. To calculate the average food price of each chef, grouping by chef name was necessary. The chef name and their average food prices were selected to be returned in the output.

5. Find the Restaurant with the Highest Average Food Price

```
58      -- NUMBER 5
59      -- Finding the restaurant with the highest average food price
60      select r.name as Restaurant,
61             avg(f.price) as AvgPrice
62      from serves s
63      inner join restaurants r
64          on s.restID = r.restID
65      inner join foods f
66          on s.foodID = f.foodID
67      group by r.name
68      having avg(f.price) >= all
69          (
70              -- Sub Query
71              -- returns list of all avg prices grouped by restaurants
72              -- used as a refernce for the Having clause in the main query
73              select avg(f.price) as AvgPrice
74              from serves s
75              inner join restaurants r
76                  on s.restID = r.restID
77              inner join foods f
78                  on s.foodID = f.foodID
79              group by r.name
80      );
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Restaurant	AvgPrice
La Trattoria	13.5
Bistro Paris	13.5
Indian Spice	13.5

Explanation: The serves, restaurants, and foods tables were joined together to answer this question. To find the restaurant with the highest average food price, I first needed to calculate the average food price of all restaurants by using a subquery. This allowed the main query to use the 'Having' clause to filter for only the restaurants with the highest of these average food prices. This one was done by using the '>= all ()' statement. The restaurant names and the average food price were selected to be returned in the output.

6. **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.

```
91  select c.name as ChefName,
92         avg(f.price) as AvgPrice,
93        GROUP_CONCAT(DISTINCT r.name) AS Restaurants -- used online resources to understand this
94  from serves s
95  inner join restaurants r on s.restID = r.restID
96  inner join foods f on s.foodID = f.foodID
97  inner join works w on w.restID = r.restID
98  inner join chefs c on c.chefID = w.chefID
99  group by c.name
100 having avg(f.price) >= all
101     (
102         -- Sub Query
103         -- returns list of all avg prices for each chef
104         -- used as a refernce for the Having clause in the main query
105         select avg(f.price) as AvgPrice
106         from serves s
107         inner join restaurants r on s.restID = r.restID
108         inner join foods f on s.foodID = f.foodID
109         inner join works w on w.restID = r.restID
110         inner join chefs c on c.chefID = w.chefID
111         group by c.name
112 )
113 order by avg(f.price) desc;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

ChefName	AvgPrice	Restaurants
Emily Davis	12.75	Indian Spice,Thai Delight
Jane Smith	12.75	La Trattoria,Sushi Haven
Michael Wilson	12.75	Indian Spice,Thai Delight
Robert Brown	12.75	Bistro Paris,Sushi Haven

Explanation: This question confused me because it seems to state that there is only one chef with a higher average food price than all others, but there are multiple chefs that are tied for this position. I proceeded by returning results for each of these chefs.

My approach was somewhat similar to #5 in the sense that I used a subquery to filter for the highest numbers within a subquery that returns a column of average food prices. However, in this case, my focus was on the chef's average food prices instead of at the restaurant level. By using this subquery to filter for only the chefs with the highest average food price, the specific columns were selected, and the restaurant names were concatenated for a cleaner output.