DB Assignment 2

9/30/25

1. Average Price of Foods at Each Restaurant

```
-- salesInfo Abstraction

create view salesInfo as
select r.name as rest_name, price,
   f.name as food_name, f.type as food_type, f.foodID, r.restID
from serves s
join foods f using (foodID)
join restaurants  r using (restID);

-- This section joins tables foods, serves, and restaurants as a view
-- for use in following queries


  -- Question #1 --

  select avg(price) as avg_price, rest_name
  from salesInfo
  group by rest_name
  order by avg_price desc;

  -- This query generates a table of average food prices per restaurant grouped
  -- by their name in a descending order using the view salesInfo
```

This problem aims to create a table showing the average price of food that is sold at every restaurant. I started solving this problem by making a view of the joined tables, serves, foods, and restaurants, so I wouldn't have to repeat the joins for every query after. My following query generates the table using the view, grouped by the restaurant name, and ordered so that the most expensive are at the top of the list.

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

2. Maximum Food Price at Each Restaurant

```
-- Question #2 --

select max(price) as max_price, rest_name
from salesInfo
group by rest_name
order by max_price desc;

-- This query generates a table of the highest pricing of food prices per restaurant grouped
-- by restaurant name in a descending order using the view salesInfo
```

This query aims to create a table of the highest prices of foods grouped by the names of the restaurants using the max aggregate function. This query also uses the salesInfo view to avoid having to join tables and orders the table so that the highest price is at the top.

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

3. Count of Different Food Types Served at Each Restaurant

```
-- Question #3 --

select  count(distinct food_type) as food_types, rest_name
from salesInfo
group by rest_name;

-- This query generates the total number of different food types per restraurant grouped by their name,
-- using the distinct keyword to not count any repeats, from the salesInfo view
```

This query is intended to create a table of the number of different food types at each restaurant and that restaurant's name. This problem requires the use of the keyword distinct, so it only detects unique types and also uses salesInfo.

	food_types	rest_name
▶	1	Bistro Paris
	1	Indian Spice
	1	La Trattoria
	2	Sushi Haven
	1	Taco Town
	1	Thai Delight

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

```
-- Question #4 --

select avg(price) as avg_price, name
from salesInfo
join works using (restID)
join chefs using (chefID)
group by name
order by avg_price desc;

-- This query generates the average price of food sold per chef grouped by their name
-- by joining salesInfo with works and chefs and ordered descending
```

This query is intended to create a table of the average price of foods that are made by each chef and their names. To compare prices and chefs, I had to join tables works, chefs, and salesInfo. The output table was ordered in descending order, so the highest average prices are on top.

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

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

```
-- Question #5 --

select avg(price) as avg_price, rest_name
from salesInfo
group by rest_name
having avg_price >= all
   (select avg (price) from salesInfo group by rest_name);

   -- This query generates the highest average price of food per restaurant,
   -- grouped their name and prints multiple if they're tied for the highest,
   -- using a subquery
```

This query was made to find the restaurant with the highest average price. The table uses salesInfo and uses the keyword having and a subquery to make sure that if the highest value is shared by multiple restaurants, they are all output.

avg_price	rest_name
13.5	La Trattoria
13.5	Bistro Paris
13.5	Indian Spice

6. Extra Credit

```
-- Question #6 --

select avg(price) as avg_price, name, rest_name
from salesInfo
join works on works.restID = salesInfo.restID
join chefs using (chefID)
group by name, rest_name
order by avg_price desc;

-- This query generates the average price of the food each chef serves for each restaurant ,
-- joining works and chefs, grouped by the chefs and restaurants name and order in a descending order
```

This query is intended to generate a table that shows the average price of foods for each chef at each restaurant and the chefs' and restaurants' names. This query also joins works, chefs, and salesInfo. The resulting table is order in descending order so the most expensive average is on top.

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