DB Assignment 2
Katrina Cwiertniewicz
9/26/2024

Data type datetime field type for servescsv./date_sold was found using MySQl Data Types (Documentation).

1. Average Price of Foods at Each Restaurant

This query selects the columns' restaurant name and food prices while aggregating food prices for their average. It joins foods to restaurants through the foreign keys included with serves. The query then groups by restaurant names and orders by average food price ascending.

```
41    -- Average Price of Foods at Each Restaurant
42 •  select restaurants.name as "Restaurant Name", avg(foods.price) as "Average Food Price"
43    from foods join serves on foods.foodID = serves.foodID
44              join restaurants on serves.restID = restaurants.restID
45    group by restaurants.name
46    order by avg(foods.price);
47
```

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

Restaurant Name	Average Food Price
Taco Town	9.5
Sushi Haven	12
Thai Delight	12
La Trattoria	13.5
Bistro Paris	13.5
Indian Spice	13.5

2. Maximum Food Price at Each Restaurant

This query selects the columns' restaurant name and food prices while aggregating food prices for their maximum. It joins foods to restaurants through the foreign keys included with serves. The query then groups by restaurant names and orders by maximum food price descending.

```
49    -- Maximum Food Price at Each Restaurant
50 •  select restaurants.name as "Restaurant Name", max(foods.price) as "Maximum Food Price"
51    from foods join serves on foods.foodID = serves.foodID
52              join restaurants on serves.restID = restaurants.restID
53    group by restaurants.name
54    order by max(foods.price) desc;
55
56
```

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

Restaurant Name	Maximum Food Price
Bistro Paris	18
La Trattoria	15
Indian Spice	15
Sushi Haven	14
Thai Delight	13
Taco Town	11

3. Count of Different Food Types Served at Each Restaurant

This query selects the column's restaurant name and food types while aggregating food types for its count. It joins foods to restaurants through the foreign keys included with serves. The query then groups by restaurant names and orders by restaurant names alphabetically.

```
57    -- Count of Different Food Types Served at Each Restaurant
58 •  select restaurants.name as "Restaurant Name", count(foods.type) as "Types of Food"
59    from foods join serves on foods.foodID = serves.foodID
60              join restaurants on serves.restID = restaurants.restID
61    group by restaurants.name
62    order by restaurants.name;
```

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

Restaurant Name	Types of Food
Bistro Paris	2
Indian Spice	2
La Trattoria	2
Sushi Haven	2
Taco Town	2
Thai Delight	2

4. Average Price of Foods Served by each Chef

This query selects the columns' chef's name and food prices while aggregating food prices for its average. It joins foods to chefs through the foreign keys included with serves and works. The query then groups by chef names and orders by average food price ascending.

```
65    -- Average Price of Foods Served by Each Chef
66 •  select chefs.name as "Chef Name", avg(foods.price) as "Average Food Price"
67    from foods join serves on foods.foodID = serves.foodID
68              join restaurants on serves.restID = restaurants.restID
69              join works on restaurants.restID = works.restID
70              join chefs on works.restID = chefs.chefID
71    group by chefs.name
72    order by avg(foods.price);
73
74
```

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

Chef Name	Average Food Price
Alice Johnson	9.5
Jane Smith	12
Emily Davis	12
John Doe	13.5
Robert Brown	13.5
Michael Wilson	13.5

5. Find the Restaurant with the Highest Average Food Price

This query selects the columns' restaurant name and food prices while aggregating food prices for their average. It joins foods to restaurants through the foreign keys included with serves. The query then groups by restaurant names and uses having to subquery for the highest average restaurants by using >= all. This is found by selecting average food prices and joining foods to restaurants and group by restaurant names again.

```
74    -- Find the Restaurant with the Highest Average Food Price
75    select restaurants.name as "Restaurant Name", avg(foods.price) as "Average Food Price"
76    from foods join serves on foods.foodID = serves.foodID
77              join restaurants on serves.restID = restaurants.restID
78    group by restaurants.name
79    having avg(foods.price) >= all (
80        select avg(foods.price)
81        from foods
82        join serves on foods.foodID = serves.foodID
83        join restaurants on serves.restID = restaurants.restID
84        group by restaurants.name
85    );
```

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

Restaurant Name	Average Food Price
La Trattoria	13.5
Bistro Paris	13.5
Indian Spice	13.5

6. Determine which chef has the highest average price of 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.

This query selects the columns' chef's name and food prices while aggregating food prices for its average and restaurant names. It joins chefs to foods through the foreign keys included with serves and works. The query then groups by chef and restaurant names and orders by average food price descending. For this problem, I was able to find the chef with the highest average price at each restaurant but not the overall average of restaurants combined for each chef.

```
75    -- Determine which chef has the highest average price of the foods served at the restaurant where they work.
76    -- Include the chef's name, the average food price, and the names of the restaurants where the chef works.
77    -- Sort the results by the average food price in descending order.
78
79    select chefs.name as "Chef Name", avg(foods.price) as "Average Food Price", restaurants.name as "Resturant where they Work"
80    from chefs join works on chefs.chefID = works.chefID
81              join restaurants on works.restID = restaurants.restID
82              join serves on restaurants.restID = serves.restID
83              join foods on serves.foodID = foods.foodID
84    group by chefs.name, restaurants.name
85    order by avg(foods.price) desc;
```

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

Chef Name	Average Food Price	Resturant where they Work
John Doe	13.5	La Trattoria
Jane Smith	13.5	La Trattoria
Robert Brown	13.5	Bistro Paris
Alice Johnson	13.5	Bistro Paris
Michael Wilson	13.5	Indian Spice
Emily Davis	13.5	Indian Spice
Jane Smith	12	Sushi Haven
Robert Brown	12	Sushi Haven
Emily Davis	12	Thai Delight
Michael Wilson	12	Thai Delight
Alice Johnson	9.5	Taco Town
John Doe	9.5	Taco Town