

SQL Relational Database Project

Food Bucket List

<Introduction>

The **Food Bucket List** database is a platform designed to help users track, review, and discover food places. It aims to bring people together to share food experiences and recommendations.

<Team members' Names & Database Location on Akira>

Anita Hu

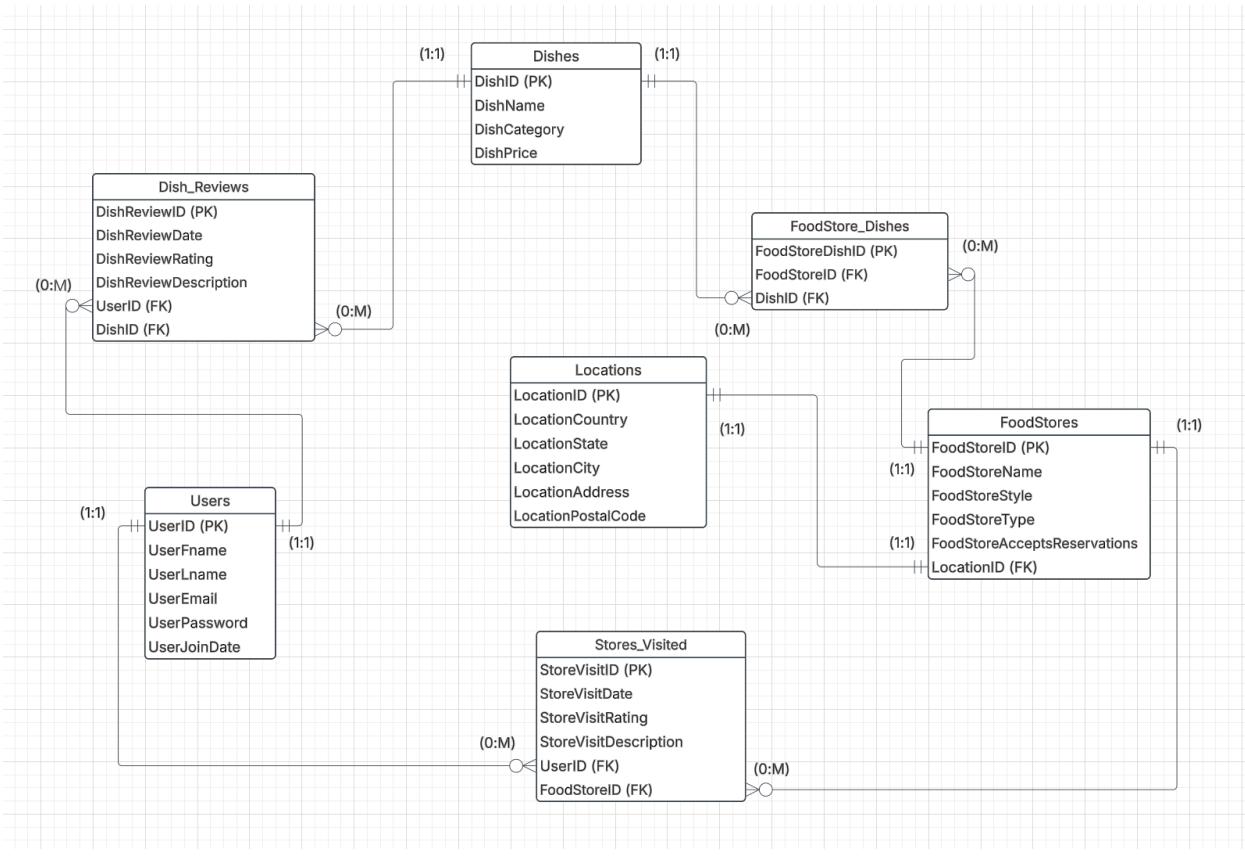
<Mission statement>

The **Food Bucket List** database allows users to track, review, and discover food places, share recommendations, and provide feedback to restaurants, fostering a community-driven platform to help others find great places to eat.

<Mission Objectives>

1. **Maintain a list of food stores** including their names, locations, and types, accessible to all users.
 2. **Track food places** users have visited and those they still want to try.
 3. **Store users' ratings** for each food store and the dishes they've tried.
 4. **Enable sorting and filtering** by store type, rating, and location for all users.
 5. **Record users' favorite and recommended** dishes or items at each store.
 6. **Allow users to share their lists** of food places, ratings, and recommendations with others.
 7. **Allow food stores to view** users' feedback and ratings for improvement.
-

<Conceptual Design – ERD>



<Logical design - Data dictionary table>

Table	Field Name	Description	Key Type	Data Type	Length	Null Support	Index	Default Value	Value Range
Users	UserID AUTO_INCREMENT	Unique identifier for the user	PK	INT		✗	✓		
	UserFname	First name of the user		VARCHAR	50	✗	✗		
	UserLname	Last name of the user		VARCHAR	50	✗	✗		
	UserEmail UNIQUE	Email of the user		VARCHAR	100	✗	✓		
	UserPassword	Password for the user		VARCHAR	255	✗	✗		
Dishes	DishID AUTO_INCREMENT	Unique identifier for the dish	PK	INT		✗	✓		
	DishName	Name of the dish		VARCHAR	100	✗	✗		
	DishCategory	Category of the dish		ENUM		✗	✓		Appetizer, Main Course, Dessert, Beverage, Soup, Snack, Other

	DishPrice	Price of the dish		DECIMAL	(5,2)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	0.00	
Locations	LocationID AUTO_INCREMENT	Unique identifier for the location	PK	INT		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
	LocationCountry	Country of the location		VARCHAR	100	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
	LocationState	State/Province of the location		VARCHAR	100	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
	LocationCity	LocationCity		VARCHAR	100	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
	LocationAddress	Address of the location		VARCHAR	255	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
	LocationPostalCode	Postal code of the location		VARCHAR	20	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
FoodStores	FoodStoreID AUTO_INCREMENT	FoodStoreID	PK	INT		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
	FoodStoreName	Name of the food store		VARCHAR	100	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
	FoodStoreStyle	Style of the food store		SET		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		French, Italian, Japanese, Chinese, American, Mexican, Korean, Indian, Thai, Vegan, Middle Eastern, Mediterranean, Other
	FoodStoreType	Type of the food store		SET		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		Restaurant , Cafe, Beverage Shop, Fast Food, Buffet, Takeout, Fine Dining, Casual Dining, Food Truck, Bakery, Bar, Ice Cream, Pizzeria, Sushi Bar, Fast Casual, Pub, Diner, Other
	FoodStoreAcceptsReservations	Whether the food store accepts reservations		BOOLEAN		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	FALSE	TRUE (Accepts reservations), FALSE (Does not accept reservations)

	LocationID	Foreign key to the Locations table	FK	INT		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Stores_Visited	StoreVisitID AUTO_INCREMENT	Unique identifier for the store visit	PK	INT		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
	StoreVisitDate	Date when the store was visited		DATE		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
	StoreVisitRating	Rating for the store visit		INT		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	3	1-5
	StoreVisitDescription	Description of the visit		TEXT		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
	UserID	Foreign key to the Users table	FK	INT		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
	FoodStoreID	Foreign key to the FoodStores table	FK	INT		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
Dish_Reviews	DishReviewID AUTO_INCREMENT	Unique identifier for the review	PK	INT		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
	DishReviewDate	Date when the review was made		DATE		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
	DishReviewRating	Rating for the dish		INT		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	3	1-5
	DishReviewDescription	Text of the review		TEXT		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
	UserID	Foreign key referencing the Users table	FK	INT		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
	DishID	Foreign key referencing the Dishes table	FK	INT		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
FoodStore_Dis hes	FoodStoreDishID	Unique identifier for the combination of food store and dish	PK	INT		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
	FoodStoreID	Foreign key referencing the FoodStores table	FK	INT		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
	DishID	Foreign key referencing the Dishes table	FK	INT		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

< Logical design - business rules >

1. User Email Uniqueness Rule

Rule Information	
Statement	Each user's email must be unique.
Constraint	The UserEmail field must be unique and cannot contain duplicate values.

Type	Database Oriented
Category	Field Specific
Test on	Insert
Structures Affected	
Field Names	UserEmail
Table Names	Users
Field Elements Affected	
Physical Elements	<input checked="" type="checkbox"/> Data Type <input checked="" type="checkbox"/> Length <input checked="" type="checkbox"/> Character Support
Logical Elements	<input checked="" type="checkbox"/> Key Type <input checked="" type="checkbox"/> Null Support <input checked="" type="checkbox"/> Uniqueness <input type="checkbox"/> Range of Values <input type="checkbox"/> Key Structure <input type="checkbox"/> Values Entered By <input type="checkbox"/> Edit Rule <input type="checkbox"/> Default Value
Relationship Characteristics Affected	
<input type="checkbox"/> Deletion Rule	
<input type="checkbox"/> Type of Participation	
<input type="checkbox"/> Degree of Participation	
Action Taken	
If violated, insertion is blocked, and an error is returned indicating a duplicate email address. If enforced, the new user is successfully added with a unique email.	

2. Every Visit Must Link to Valid User and Food Store

Rule Information	
Statement	Every visit record must link to valid users and food stores.
Constraint	The UserID and FoodStoreID must both exist in their respective tables as foreign keys.
Type	Database Oriented
Category	Relationship Specific
Test on	Insert
Structures Affected	
Field Names	UserID, FoodStoreID
Table Names	Stores_Visited
Field Elements Affected	
Physical Elements	<input checked="" type="checkbox"/> Data Type <input checked="" type="checkbox"/> Length <input type="checkbox"/> Character Support
Logical Elements	<input checked="" type="checkbox"/> Key Type

	<input checked="" type="checkbox"/> Null Support <input type="checkbox"/> Uniqueness <input type="checkbox"/> Range of Values <input type="checkbox"/> Key Structure <input type="checkbox"/> Values Entered By <input type="checkbox"/> Edit Rule <input type="checkbox"/> Default Value
Relationship Characteristics Affected	
<input type="checkbox"/> Deletion Rule	
<input checked="" type="checkbox"/> Type of Participation	
<input checked="" type="checkbox"/> Degree of Participation	
Action Taken	
If violated, an error is triggered, and the visit record cannot be inserted.	
If enforced, the visit record is added successfully with the correct user and food store relationships.	

3. Default Dish Price

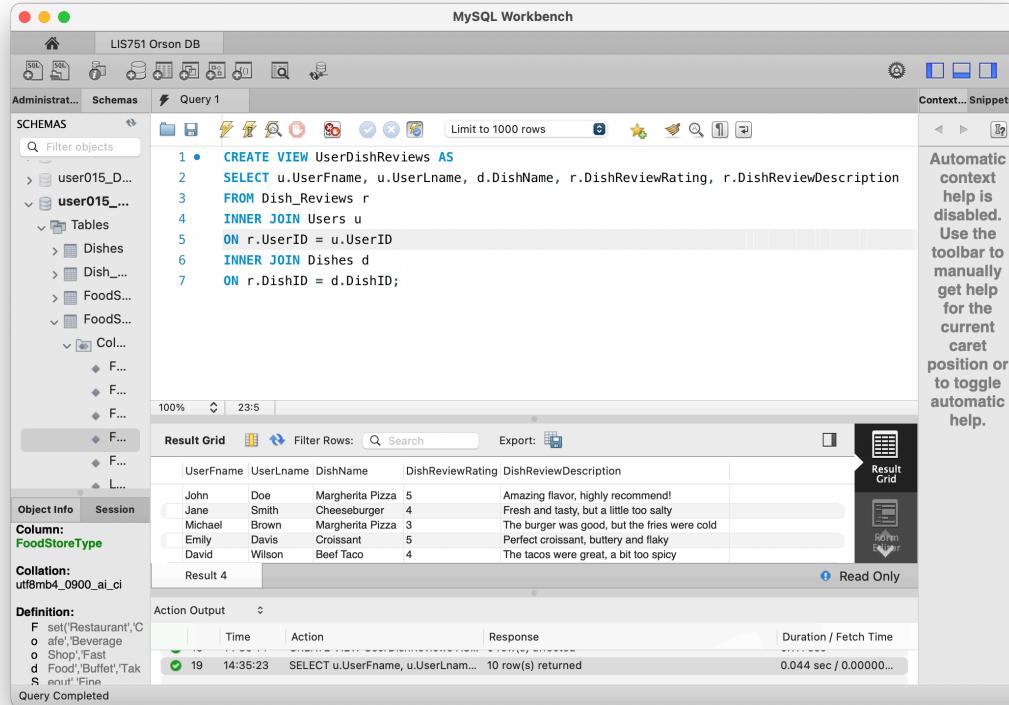
Rule Information	
Statement	If no dish price is provided, the price will default to 0.00.
Constraint	DishPrice should default to 0.00 if not provided.
Type	Database Oriented
Category	Field Specific
Test on	Insert
Structures Affected	
Field Names	DishPrice
Table Names	Dishes
Field Elements Affected	
Physical Elements	<input checked="" type="checkbox"/> Data Type <input checked="" type="checkbox"/> Length <input type="checkbox"/> Character Support
Logical Elements	<input type="checkbox"/> Key Type <input checked="" type="checkbox"/> Null Support <input type="checkbox"/> Uniqueness <input checked="" type="checkbox"/> Range of Values <input type="checkbox"/> Key Structure <input type="checkbox"/> Values Entered By <input type="checkbox"/> Edit Rule <input checked="" type="checkbox"/> Default Value
Relationship Characteristics Affected	
<input type="checkbox"/> Deletion Rule	
<input type="checkbox"/> Type of Participation	
<input type="checkbox"/> Degree of Participation	
Action Taken	

If violated (no price is given), the DishPrice is automatically set to 0.00.
If enforced, the dish record is inserted with the correct price.

<Queries>

1. Display all users and the dishes they reviewed

(joins the **Users**, **Dish_Reviews**, and **Dishes** tables)



The screenshot shows the MySQL Workbench interface with a query editor window. The query is:

```

1 • CREATE VIEW UserDishReviews AS
2   SELECT u.UserName, u.UserLname, d.DishName, r.DishReviewRating, r.DishReviewDescription
3   FROM Dish_Reviews r
4   INNER JOIN Users u
5   ON r.UserID = u.UserID
6   INNER JOIN Dishes d
7   ON r.DishID = d.DishID;

```

The results grid displays the following data:

User	Name	Dish	Rating	Description
John	Doe	Margherita Pizza	5	Amazing flavor, highly recommend!
Jane	Smith	Cheeseburger	4	Fresh and tasty, but a little too salty
Michael	Brown	Margherita Pizza	3	The burger was good, but the fries were cold
Emily	Davis	Croissant	5	Perfect croissant, buttery and flaky
David	Wilson	Beef Taco	4	The tacos were great, a bit too spicy

Below the results, the status bar shows "Result 4" and "Read Only". The bottom pane shows the execution log:

Action	Time	Action	Response	Duration / Fetch Time
19	14:35:23	SELECT u.UserName, u.UserLnam...	10 row(s) returned	0.044 sec / 0.00000...

2. Display food stores and their location details

(joins the **FoodStores** and **Locations** tables)

MySQL Workbench

```

1 • CREATE VIEW FoodStoreLocation AS
2   SELECT f.FoodStoreName, f.FoodStoreStyle, f.FoodStoreType, l.LocationCity, l.LocationState
3   FROM FoodStores f
4   JOIN Locations l ON f.LocationID = l.LocationID;

```

Result Grid

FoodStoreName	FoodStoreStyle	FoodStoreType	LocationCity	LocationState	LocationAddress
Pizza Place	Italian	Restaurant	Los Angeles	California	123 Sunset Blvd
Sushi World	Japanese	Restaurant	San Francisco	California	456 Market St
Burger King	American	Fast Food	New York City	New York	789 Broadway St
Café Mocha	French	Cafe	Austin	Texas	101 Congress Ave
Taco Bell	Mexican	Fast Food	Toronto	Ontario	202 King St

Result 5

Action Output

	Time	Action	Response	Duration / Fetch Time
1	14:40:29	CREATE VIEW FoodStoreLocation...	0 row(s) affected	0.170 sec
2	14:40:37	SELECT f.FoodStoreName, f.FoodS...	10 row(s) returned	0.052 sec / 0.000009...

Query Completed

3. Display all dishes along with their average review rating (uses Dishes, Dish_Reviews)

MySQL Workbench

```

1 • CREATE VIEW DishAverageRating AS
2   SELECT d.DishName, AVG(r.DishReviewRating) AS AvgRating
3   FROM Dishes d
4   JOIN Dish_Reviews r ON d.DishID = r.DishID
5   GROUP BY d.DishID;
6

```

Result Grid

DishName	AvgRating
Margarita Pizza	4.0000
California Roll	5.0000
Cheeseburger	4.0000
Croissant	5.0000
Beef Taco	4.0000

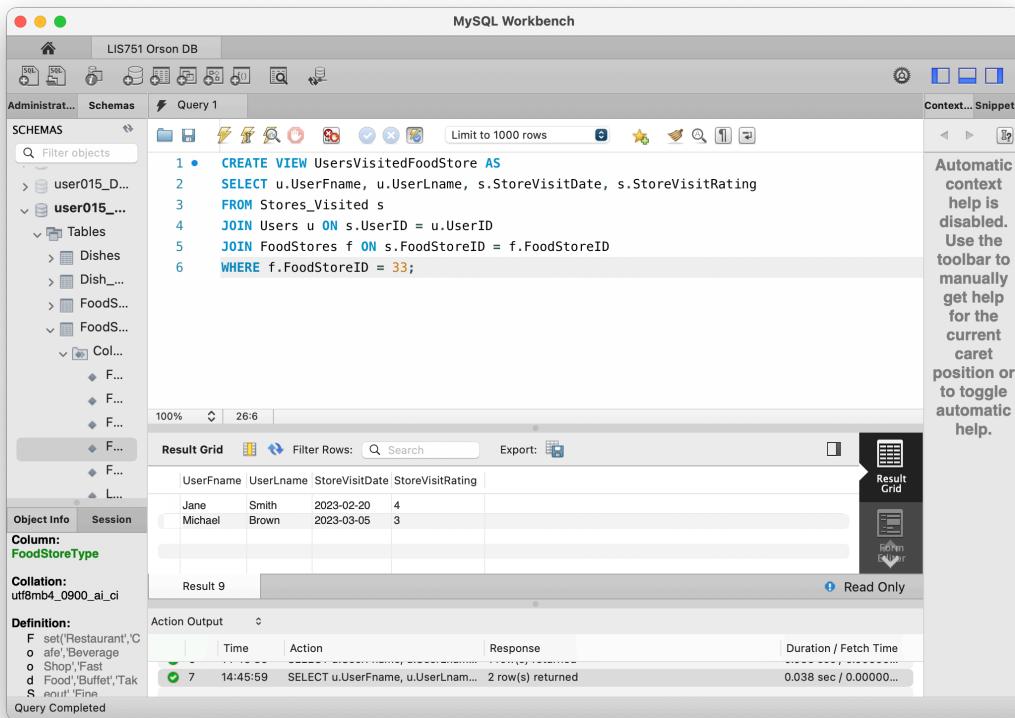
Result 7

Action Output

	Time	Action	Response	Duration / Fetch Time
5	14:44:20	SELECT d.DishName, AVG(r.DishRe... 8 row(s) returned		0.039 sec / 0.00000...

Query Completed

4. Display all users who have visited a specific food store (joins **Users**, **Stores_Visited**, and **FoodStores**)



The screenshot shows the MySQL Workbench interface with the title bar "MySQL Workbench" and the database "LIS751 Orson DB". The left sidebar shows the schema "user015_D..." with tables "Dishes", "FoodS...", and "FoodS...". The main area displays a query results grid for a view named "UsersVisitedFoodStore". The SQL code is:

```
1 • CREATE VIEW UsersVisitedFoodStore AS
2 SELECT u.UserFname, u.UserLname, s.StoreVisitDate, s.StoreVisitRating
3 FROM Stores_Visited s
4 JOIN Users u ON s.UserID = u.UserID
5 JOIN Foodstores f ON s.FoodStoreID = f.FoodStoreID
6 WHERE f.FoodStoreID = 33;
```

The results grid shows two rows:

UserName	UserLname	StoreVisitDate	StoreVisitRating
Jane	Smith	2023-02-20	4
Michael	Brown	2023-03-05	3

Below the grid, the status bar shows "Result: 9" and "Read Only". The bottom pane shows the action output with one row:

Action	Time	Response	Duration / Fetch Time
SELECT u.UserFname, u.UserLnam...	14:45:59	2 row(s) returned	0.038 sec / 0.00000...

A note on the right side of the interface says: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."

5. Display food stores and their dishes (joins **FoodStores**, **FoodStore_Dishes**, and **Dishes**)

MySQL Workbench

```

1 • CREATE VIEW FoodStoreDishes AS
2   SELECT f.FoodStoreName, d.DishName
3   FROM FoodStores f
4   JOIN FoodStore_Dishes fsd ON f.FoodStoreID = fsd.FoodStoreID
5   JOIN Dishes d ON fsd.DishID = d.DishID;
6

```

Result Grid

FoodStoreName	DishName
Pizza Place	Margherita Pizza
Sushi World	California Roll
Burger King	Cheeseburger
Burger King	Beef Taco
Café Mocha	Croissant

Result 11

Action Output

Time	Action	Response	Duration / Fetch Time
11	14:52:28	SELECT f.FoodStoreName, d.DishN... 10 row(s) returned	0.036 sec / 0.00000...

Query Completed

6. Display the average rating for each food store based on store visits (joins **FoodStores** and **Stores_Visited**)

MySQL Workbench

```

1 • CREATE VIEW FoodStoreAvgVisitRating AS
2   SELECT f.FoodStoreName, AVG(s.StoreVisitRating) AS AvgVisitRating
3   FROM FoodStores f
4   JOIN Stores_Visited s ON f.FoodStoreID = s.FoodStoreID
5   GROUP BY f.FoodStoreID;
6

```

Result Grid

FoodStoreName	AvgVisitRating
Pizza Place	5.0000
Sushi World	5.0000
Burger King	3.5000
Café Mocha	4.0000
Pasta Paradise	4.5000

Result 12

Action Output

Time	Action	Response	Duration / Fetch Time
13	14:54:02	SELECT f.FoodStoreName, AVG(s.S... 8 row(s) returned	0.052 sec / 0.000013...

Query Completed

7. Display the count of reviews for each dish (joins Dishes and Dish_Reviews)

The screenshot shows the MySQL Workbench interface with the following details:

- Schemas:** LIS751 Orson DB
- Query Editor:** Query 1 (CREATE VIEW command)
- Result Grid:** Displays the results of the query, showing the count of reviews for each dish.
- Action Output:** Shows the execution log with the query and its duration.
- Note:** A note on the right side states: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."

```
1 • CREATE VIEW DishReviewCount AS
2   SELECT d.DishName, COUNT(r.DishReviewID) AS ReviewCount
3   FROM Dishes d
4   JOIN Dish_Reviews r ON d.DishID = r.DishID
5   GROUP BY d.DishID;
6
```

DishName	ReviewCount
Margherita Pizza	2
California Roll	1
Cheeseburger	1
Croissant	1
Beef Taco	2

Action Output

Time	Action	Response	Duration / Fetch Time
15 14:56:02	SELECT d.DishName, COUNT(r.DishReviewID) AS ReviewCount FROM Dishes d JOIN Dish_Reviews r ON d.DishID = r.DishID; GROUP BY d.DishID;	8 row(s) returned	0.055 sec / 0.000006...