Project 5

Implementing a Database

Written By

Mr.	Suppakorn	Pojsomphong	6588077
Miss	Nuttida	Wiphaalongkot	6588144
Miss	Ananya	Sangtunchai	6588191

A Report Submitted in Partial Fulfillment of

the Requirements for

ITCS413 Database Design

Faculty of Information and Communication Technology

Semester 2/2024

Table of Contents

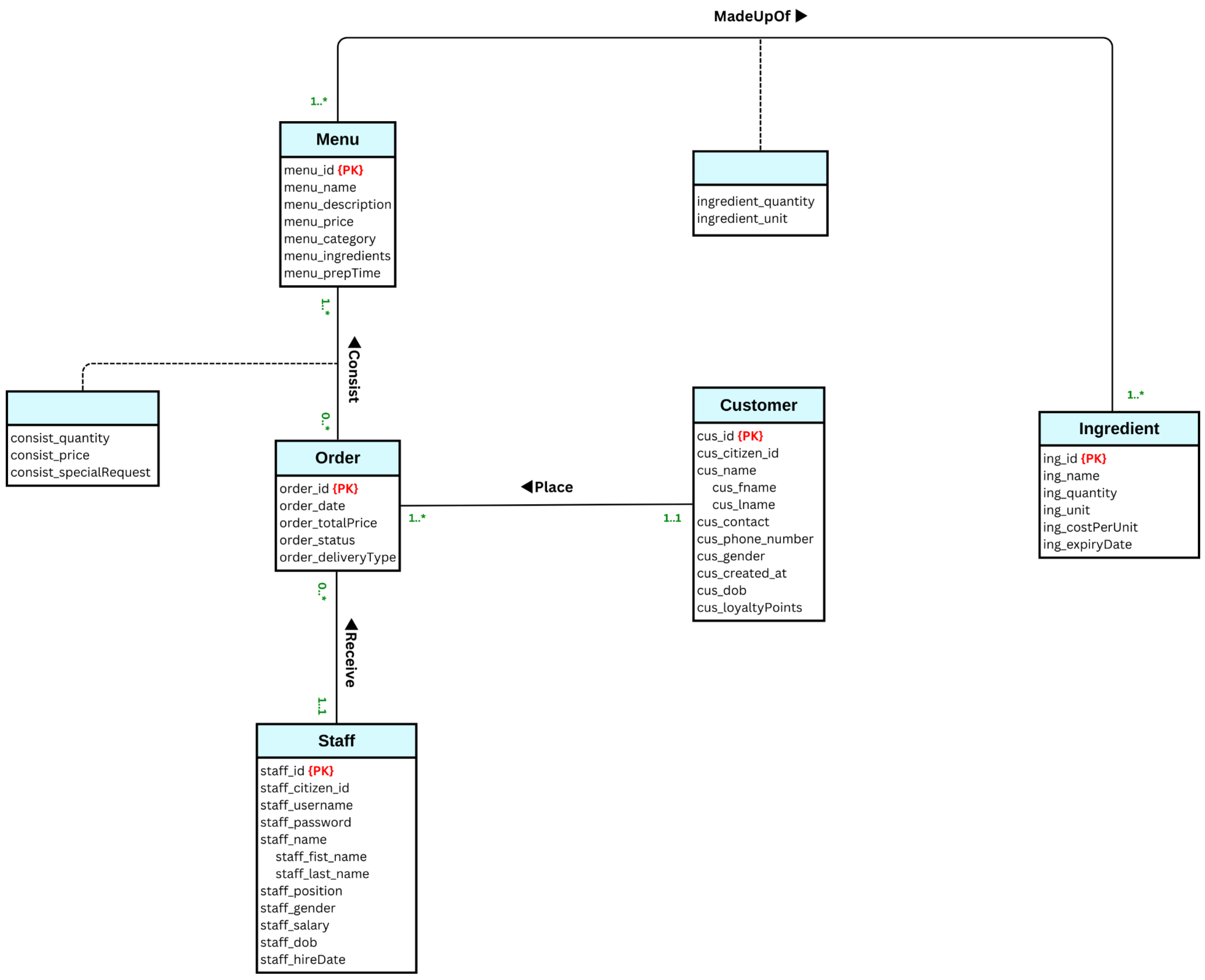
1. Database Application Requirements

User / Entity	Data Entry	Data Update/Deletion	Data Queries
Customer	Add a new customer (e.g., Register a new customer, Mr. Ronaldo, with citizen ID 123456789)	● Update a customer's contact information (e.g., Change the phone number of Mr. Ronaldo) ● Delete a customer record (e.g., Remove a customer who has not ordered in a year)	● Retrieve a list of all customers who have placed an order in the last 30 days ● Search for a customer by name ● List all male customers
Staff	Add a new staff member (e.g., Hire a new chef, Mr. Smith, for the restaurant)	● Update a staff member's position (e.g., Promote a waiter to manager) ● Remove a staff record (e.g., Delete a staff member who resigned)	● List all staff members with the position "Chef" ● Find all male staff members ● Retrieve all staff hired in the last 6 months
Admin	Register a new admin	● Update admin contact details ● Remove an admin account	● Retrieve admin details based on email ● List all active admins
Ingredient	Add a new ingredient to stock	● Update stock quantity for an ingredient ● Delete an expired ingredient	● List all ingredients with less than 5 units in stock ● Retrieve ingredients supplied by "FreshFarm ● Retrieve the ingredients needed for specific menu items ● Retrieve which ingredients are used in which menu items and in what quantity
Menu	Add a new dish to the menu	● Update dish price ● Remove a discontinued dish	● List all menu items in the "Burger" category ● Find menu items below 100 Baht.

Order	Place a new order	• Update order status to "Completed" • Cancel an order	• Retrieve orders placed by a specific customer • Find all pending orders • Count how many orders each staff member has handled in a specific date range
Feedback	Add a new feedback entry	• Update customer rating • Delete inappropriate feedback	• Retrieve feedback for a specific menu item • Find all feedback with a rating of 1 or 2 stars
Promotion	Add a new promotional offer	• Update discount percentage • Remove expired promotions	• List all active promotions • Find promotions with a discount of 20% or more

2. Final Conceptual Database Model (ER Diagram)

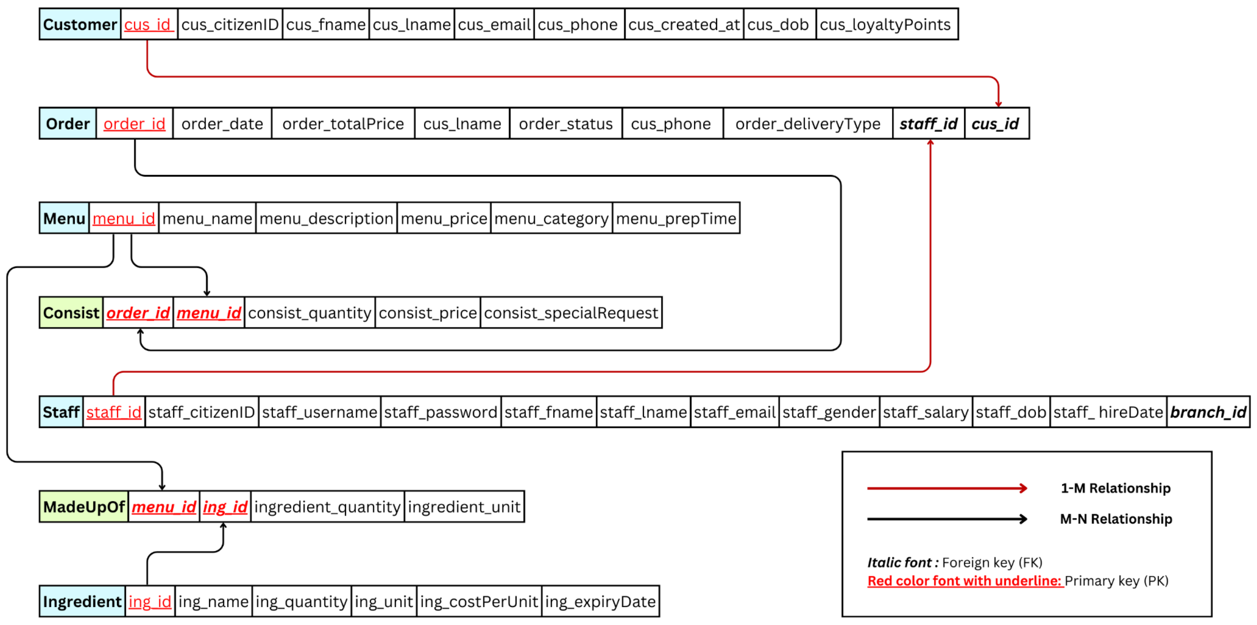
- Highlight parts related to selected transactions



https://www.canva.com/design/DAGhHnQ5ZTo/PqykRRQULkOmhp4q7SFDwA/edit?utm_content=DAGhHnQ5ZTo&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

3. Final Logical Database Model (Relational Schema)

- Highlight parts related to selected transactions



Customer | cus_id | cus_citizenID | cus_fname | cus_lname | cus_email | cus_phone | cus_created_at | cus_dob | cus_loyaltyPoints
Order | order_id | order_date | order_totalPrice | cus_lname | order_status | cus_phone | order_deliveryType | staff_id | cus_id
Menu | menu_id | menu_name | menu_description | menu_price | menu_category | menu_prepTime
Consist | order_id | menu_id | consist_quantity | consist_price | consist_specialRequest
Staff | staff_id | staff_citizenID | staff_username | staff_password | staff_fname | staff_lname | staff_email | staff_gender | staff_salary | staff_dob | staff_ hireDate | branch_id
MadeUpOf | menu_id | ing_id | ingredient_quantity | ingredient_unit
Ingredient | ing_id | ing_name | ing_quantity | ing_unit | ing_costPerUnit | ing_expiryDate
1-M Relationship
M-N Relationship
Italic font : Foreign key (FK)
Red color font with underline: Primary key (PK)

https://www.canva.com/design/DAGhHnQ5ZTo/PqykRRQULkOmhp4q7SFDwA/edit?utm_content=DAGhHnQ5ZTo&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton

4. SQL Commands & Table Specs

SQL queries for selected transactions

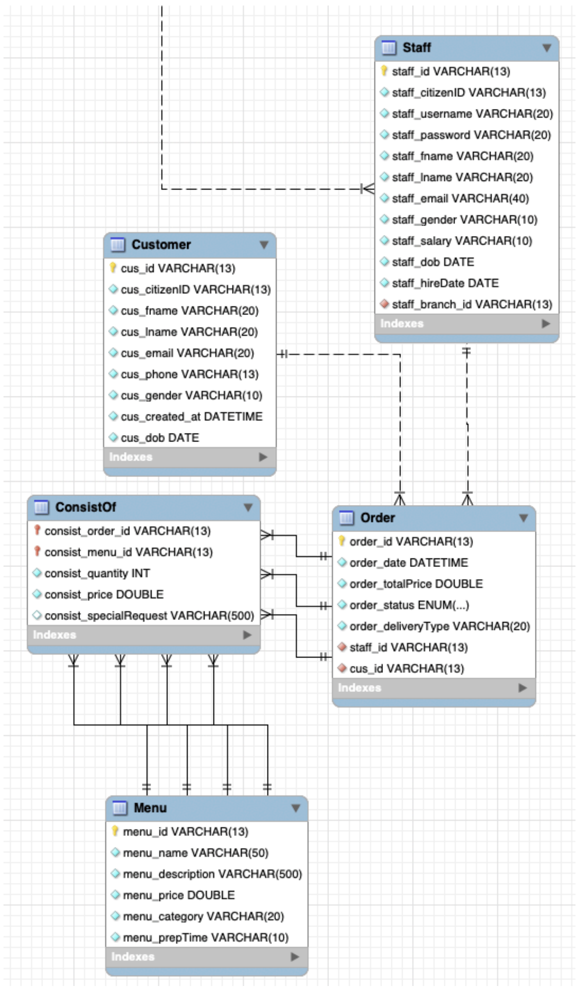
- **Related SQL:** CREATE TABLE, INSERT, etc.

Describe the table specs:

- Number of records
- Record size
- Any constraints/indexes before tuning

Key Transactions (Queries):

1. Order Processing Query

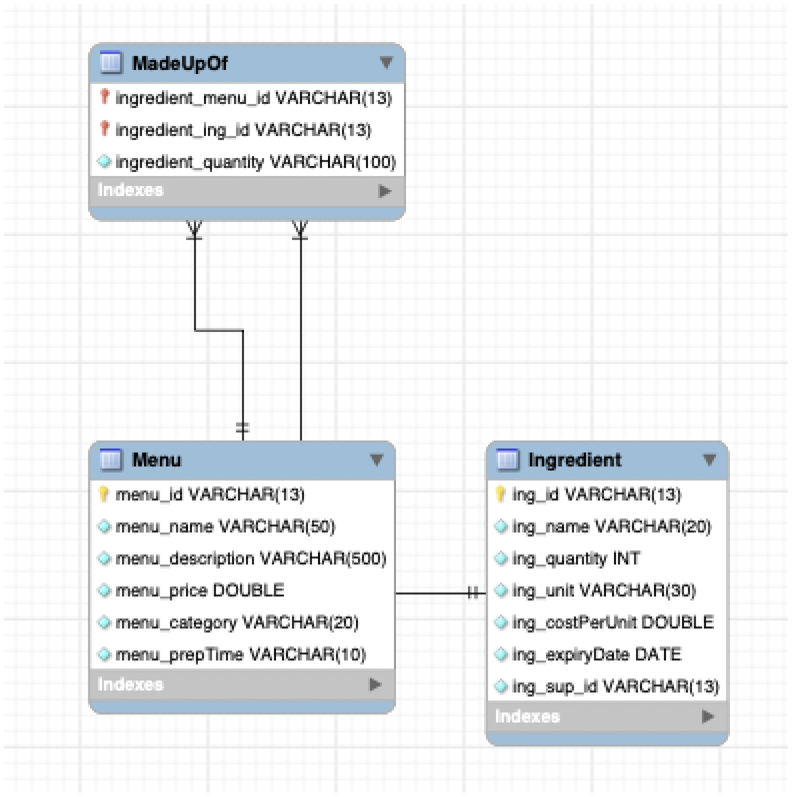


- **Purpose:** To retrieve all the items in a particular order and calculate the total price.
- **Calculate:**
  `order_id VARCHAR(13)`: 13 bytes,
  `order_date DATETIME`: 8 bytes,
  `order_totalPrice DOUBLE`: 8 bytes,
  `cus_fname VARCHAR(20)`: 20 bytes,
  `cus_lname VARCHAR(20)`: 20 bytes,
  `staff_fname VARCHAR(20)`: 20 bytes,
  `staff_lname VARCHAR(20`: 20 bytes,
  `menu_name VARCHAR(50)`: 50 bytes,
  `consist_quantity INT`: 4 bytes

  Total size of each record: 13+8+8+20+20+20+20+50+4 = 163 bytes
  - Before Improvement :
    Total size of the query result = Number of records scanned * Record size of each record
    = 596*163
    = 97,148 bytes
  - After Improvement: The query result size did not change after the improvement, so the record size of query 1 result both before and after the improvement stands at 97,148 bytes
- **Tables Involved:** Order, ConsistOf, Menu, Customer, Staff

```
SELECT o.order_id, o.order_date, o.order_totalPrice, c.cus_fname,
       c.cus_lname, s.staff_fname, s.staff_lname, m.menu_name,
       co.consist_quantity
FROM Order o
JOIN ConsistOf co ON o.order_id = co.consist_order_id
JOIN Menu m ON co.consist_menu_id = m.menu_id
JOIN Customer c ON o.cus_id = c.cus_id
JOIN Staff s ON o.staff_id = s.staff_id
WHERE o.order_status = 'Pending';
```

2. Inventory Tracking Query

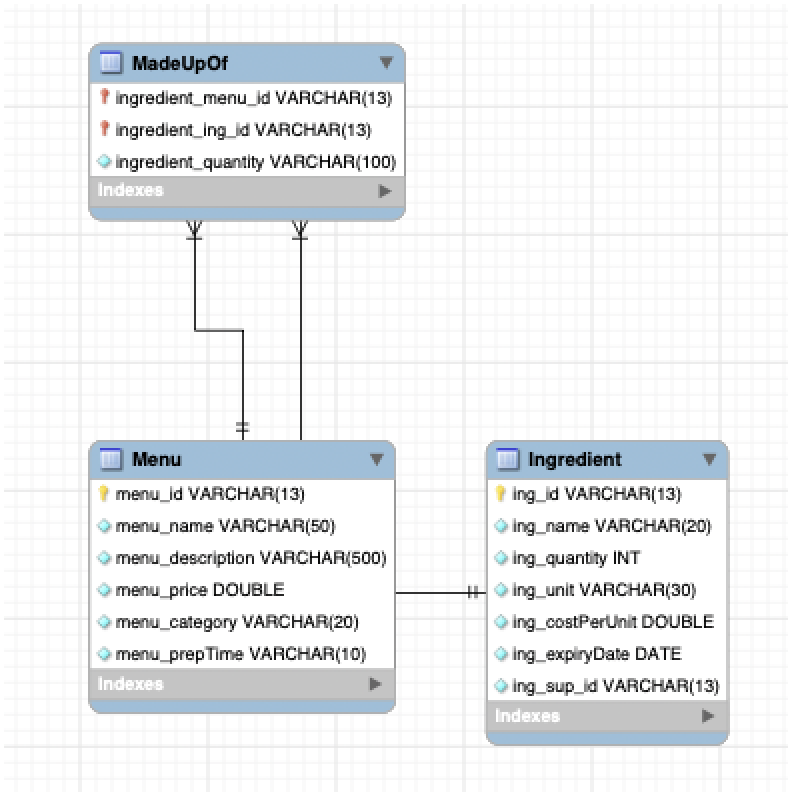


- **Purpose:** To track the ingredients needed for specific menu items, which helps with stock control and inventory management.
- **Calculate:**
  `menu_name VARCHAR(50)`: 50 bytes,
  `ing_name VARCHAR(20)`: 20 bytes,
  `ingredient_quantity VARCHAR(100)`: 100 bytes,
  `ing_quantity INT`: 4 bytes
- Total size of each record: 50+20+100+4 = 174 bytes
  - Before Improvement :
    Total size of the query result = Number of records scanned * Record size of each record
    = 5*174
    = 870 bytes
  - After Improvement: The query result size did not change after the improvement, so the record size of query 2 result both before and after the improvement stands at 870 bytes

- **Tables Involved:** MadeUpOf, Ingredient, Menu

```
SELECT m.menu_name, i.ing_name,
       mu.ingredient_quantity, i.ing_quantity
FROM MadeUpOf mu
JOIN Ingredient i ON mu.ingredient_ing_id = i.ing_id
JOIN Menu m ON mu.ingredient_menu_id = m.menu_id
WHERE m.menu_id = 'BK0045';
```

3. Track Inventory Ingredients in Menus

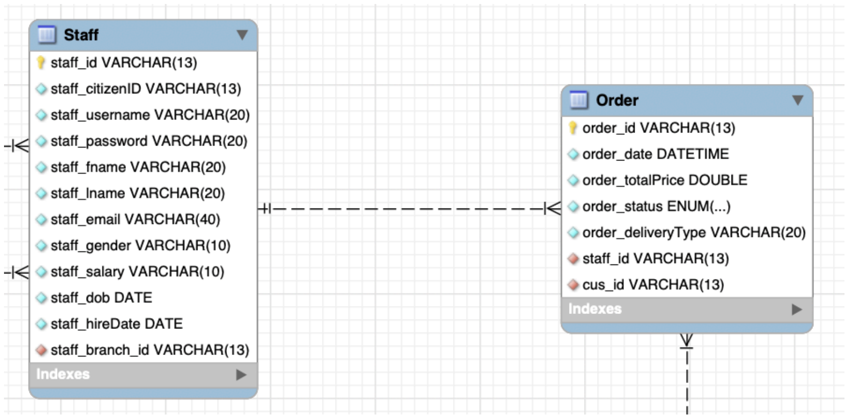


- **Purpose:** To view which ingredients are used in which menu items and in what quantity.
- **Calculate:**
  `ing_name VARCHAR(20)`: 20 bytes,
  `ing_quantity INT`: 4 bytes,
  `menu_name VARCHAR(50)`: 50 bytes,
  `ingredient_quantity VARCHAR(100)`: 100 bytes
  Total size of each record: 20+4+50+100 = 174 bytes
  - Before Improvement :
    Total size of the query result = Number of records scanned * Record size of each record
    = 631*174
    = 109,794 bytes

- After Improvement: The query result size did not change after the improvement, so the record size of query 3 result both before and after the improvement stands at 109,794 bytes
- **Tables Involved:** MadeUpOf, Ingredient, Menu

```
SELECT i.ing_name, i.ing_quantity,
       m.menu_name, mo.ingredient_quantity
FROM Ingredient i
JOIN MadeUpOf mo ON i.ing_id = mo.ingredient_ing_id
JOIN Menu m ON mo.ingredient_menu_id = m.menu_id
WHERE i.ing_quantity > 0;
```

4. Monitor Active Staff Order Assignments



- **Purpose:** To count how many orders each staff member has handled in a specific date range.
- **Calculate:**

```
staff_id VARCHAR(13),
staff_fname VARCHAR(20),
staff_lname VARCHAR(20),
order_id VARCHAR(13)
```

- Total size of each record: 13+20+20+13 = 66 bytes
  - Before Improvement :
    Total size of the query result = Number of records scanned * Record size of each record
    = 10*66
    = 660 bytes
  - After Improvement: The query result size did not change after the improvement, so the record size of query 4 result both before and after the improvement stands at 660 bytes
- **Tables Involved:** Order, Staff

```
SELECT s.staff_id, s.staff_fname, s.staff_lname, COUNT(o.order_id) AS
total_orders
FROM Staff s
JOIN [Order] o ON s.staff_id = o.staff_id
WHERE o.order_date BETWEEN '2024-11-20' AND '2025-02-21'
GROUP BY s.staff_id, s.staff_fname, s.staff_lname
ORDER BY total_orders DESC;
```

5. Performance Before Optimization

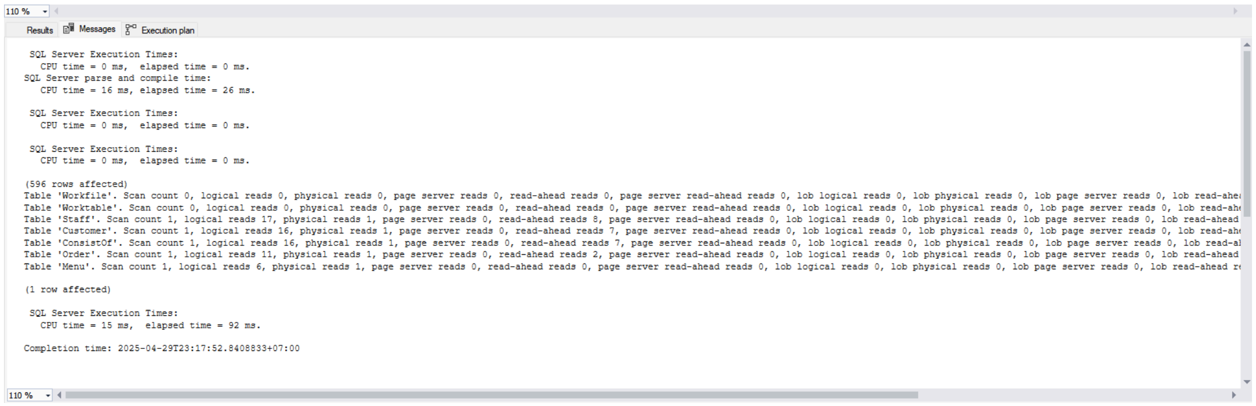
1. Analyze Order Processing Query

Transaction Analysis Form			
			April 30, 2025
Transaction:	To retrieve all the items in a particular order and calculate the total price.		
Volume:	Average		
	Peak		

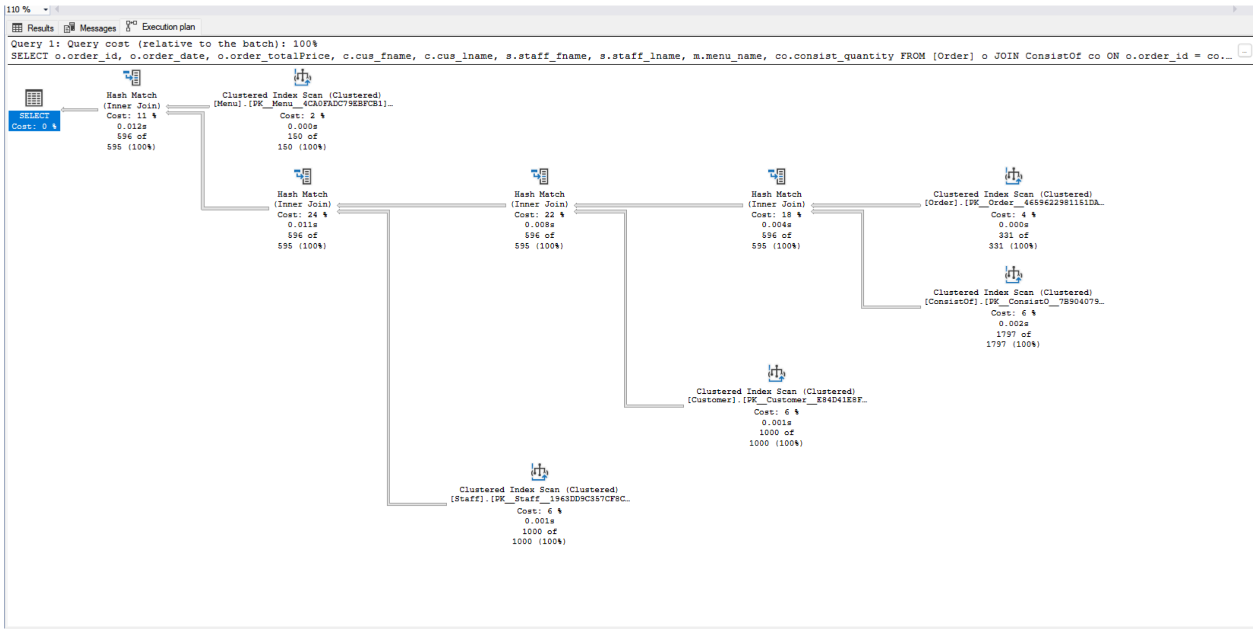
```
SELECT o.order_id, o.order_date,
o.order_totalPrice, c.cus_fname,
              c.cus_lname, s.staff_fname,
       s.staff_lname, m.menu_name,
              co.consist_quantity
FROM Order o
JOIN ConsistOf co ON o.order_id =
co.consist_order_id
JOIN Menu m ON co.consist_menu_id = m.menu_id
JOIN Customer c ON o.cus_id = c.cus_id
JOIN Staff s ON o.staff_id = s.staff_id
WHERE o.order_status = 'Pending';
```

Predicate:	`o.order_status = 'Pending';`
Join Attributes:	`order_id, menu_id, cus_id, staff_id`
Ordering attributes:	-
Grouping attributes:	-
Built-in functions:	-
Attributes updated	-

Access	Entity	Type of Access	No. of References	
			Per Transaction	Peek Per Hour
1	Order	R	1	500
2	ConsistOf	R	1	500
3	Menu	R	1	500
4	Customer	R	1	500
5	Staff	R	1	500



```
 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.
SQL Server parse and compile time:
   CPU time = 16 ms, elapsed time = 26 ms.

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.

(596 rows affected)
Table 'Workfile'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahea
Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahe
Table 'Staff'. Scan count 1, logical reads 17, physical reads 1, page server reads 0, read-ahead reads 8, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahead
Table 'Customer'. Scan count 1, logical reads 16, physical reads 1, page server reads 0, read-ahead reads 7, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahe
Table 'ConsistOf'. Scan count 1, logical reads 16, physical reads 1, page server reads 0, read-ahead reads 7, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-al
Table 'Order'. Scan count 1, logical reads 11, physical reads 1, page server reads 0, read-ahead reads 2, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahead
Table 'Menu'. Scan count 1, logical reads 6, physical reads 1, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahead re

(1 row affected)

 SQL Server Execution Times:
   CPU time = 15 ms,  elapsed time = 92 ms.

Completion time: 2025-04-29T23:17:52.8408833+07:00
```



```
Query 1: Query cost (relative to the batch): 100%
SELECT o.order_id, o.order_date, o.order_totalPrice, c.cus_fname, c.cus_lname, s.staff_fname, s.staff_lname, m.menu_name, co.consist_quantity FROM [Order] o JOIN ConsistOf co ON o.order_id = co.…
```

```
SELECT
Cost: 0 %

Hash Match
(Inner Join)
Cost: 11 %
0.012s
596 of
595 (100%)

Clustered Index Scan (Clustered)
[Menu].[PK__Menu__4CA0FADC79EBFCB1]…
Cost: 2 %
0.000s
150 of
150 (100%)

Hash Match
(Inner Join)
Cost: 24 %
0.011s
596 of
595 (100%)

Hash Match
(Inner Join)
Cost: 22 %
0.008s
596 of
595 (100%)

Hash Match
(Inner Join)
Cost: 18 %
0.004s
596 of
595 (100%)

Clustered Index Scan (Clustered)
[Order].[PK__Order__4659622981151DA…
Cost: 4 %
0.000s
331 of
331 (100%)

Clustered Index Scan (Clustered)
[ConsistOf].[PK__ConsistO__7B904079…
Cost: 6 %
0.002s
1797 of
1797 (100%)

Clustered Index Scan (Clustered)
[Customer].[PK__Customer__E84D41E8F…
Cost: 6 %
0.001s
1000 of
1000 (100%)

Clustered Index Scan (Clustered)
[Staff].[PK__Staff__1963DD9C357CF8C…
Cost: 6 %
0.001s
1000 of
1000 (100%)
```

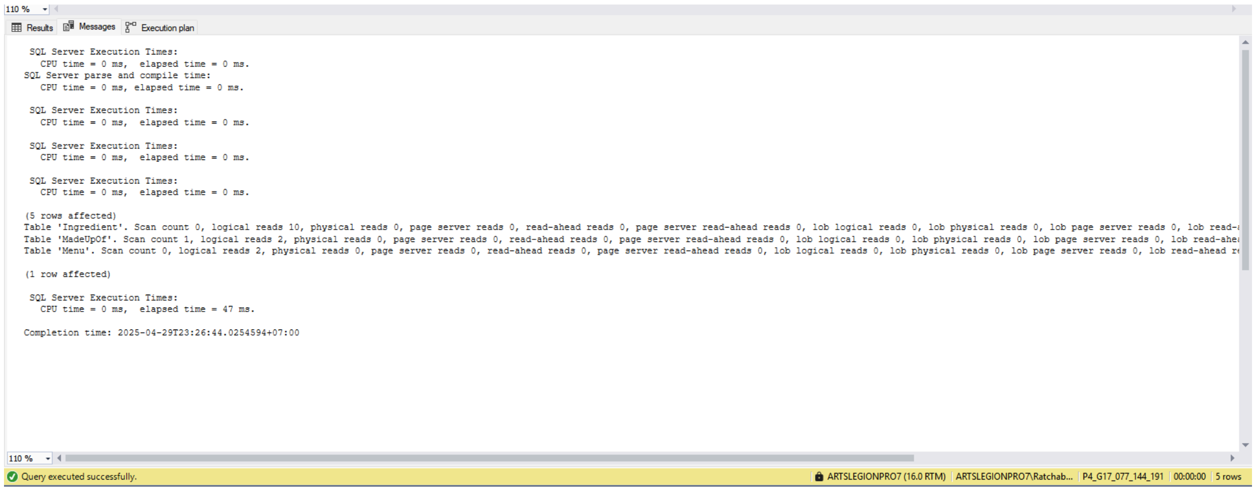
2. Inventory Tracking Query

Transaction Analysis Form			
			April 30, 2025
Transaction:	To track the ingredients needed for specific menu items, which helps with stock control and inventory management.		
Volume:	Average		
	Peak		

```
SELECT m.menu_name, i.ing_name,
             mu.ingredient_quantity,
            i.ing_quantity
FROM MadeUpOf mu
JOIN Ingredient i ON mu.ingredient_ing_id =
i.ing_id
JOIN Menu m ON mu.ingredient_menu_id =
m.menu_id
WHERE m.menu_id = 'BK0045';
```

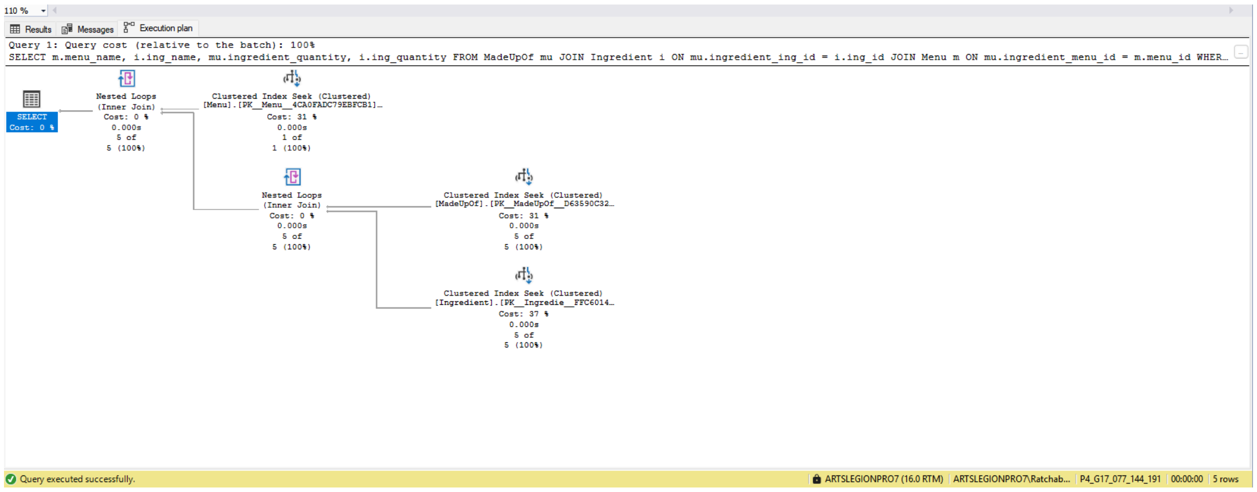
Predicate:	`m.menu_id = 'BK0045'`
Join Attributes:	`ing_id, menu_id`
Ordering attributes:	-
Grouping attributes:	-
Built-in functions:	-
Attributes updated	-

Access	Entity	Type of Access	No. of References	
			Per Transaction	Peek Per Hour
1	`MadeUpOf`	R	1	500
2	`Ingredient`	R	1	500
3	`Menu`	R	1	500



```
 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.
SQL Server parse and compile time:
   CPU time = 0 ms, elapsed time = 0 ms.

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.

(5 rows affected)
Table 'Ingredient'. Scan count 0, logical reads 10, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-
Table 'MadeUpOf'. Scan count 1, logical reads 2, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahe
Table 'Menu'. Scan count 0, logical reads 2, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahead r

(1 row affected)

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 47 ms.

Completion time: 2025-04-29T23:26:44.0254594+07:00
```

110 %
Query executed successfully.
ARTSLEGIONPRO7 (16.0 RTM) | ARTSLEGIONPRO7\Ratchab... | P4_G17_077_144_191 | 00:00:00 | 5 rows


110 %
Results
Messages
Execution plan

```
Query 1: Query cost (relative to the batch): 100%
SELECT m.menu_name, i.ing_name, mu.ingredient_quantity, i.ing_quantity FROM MadeUpOf mu JOIN Ingredient i ON mu.ingredient_ing_id = i.ing_id JOIN Menu m ON mu.ingredient_menu_id = m.menu_id WHER…
```

SELECT
Cost: 0 %
Nested Loops
(Inner Join)
Cost: 0 %
0.000s
5 of
5 (100%)
Clustered Index Seek (Clustered)
[Menu].[PK__Menu__4CA0FADC79EBFCB1]…
Cost: 31 %
0.000s
1 of
1 (100%)
Nested Loops
(Inner Join)
Cost: 0 %
0.000s
5 of
5 (100%)
Clustered Index Seek (Clustered)
[MadeUpOf].[PK__MadeUpOf__D63590C32…
Cost: 31 %
0.000s
5 of
5 (100%)
Clustered Index Seek (Clustered)
[Ingredient].[PK__Ingredie__FFC6014…
Cost: 37 %
0.000s
5 of
5 (100%)
Query executed successfully.
ARTSLEGIONPRO7 (16.0 RTM) | ARTSLEGIONPRO7\Ratchab... | P4_G17_077_144_191 | 00:00:00 | 5 rows

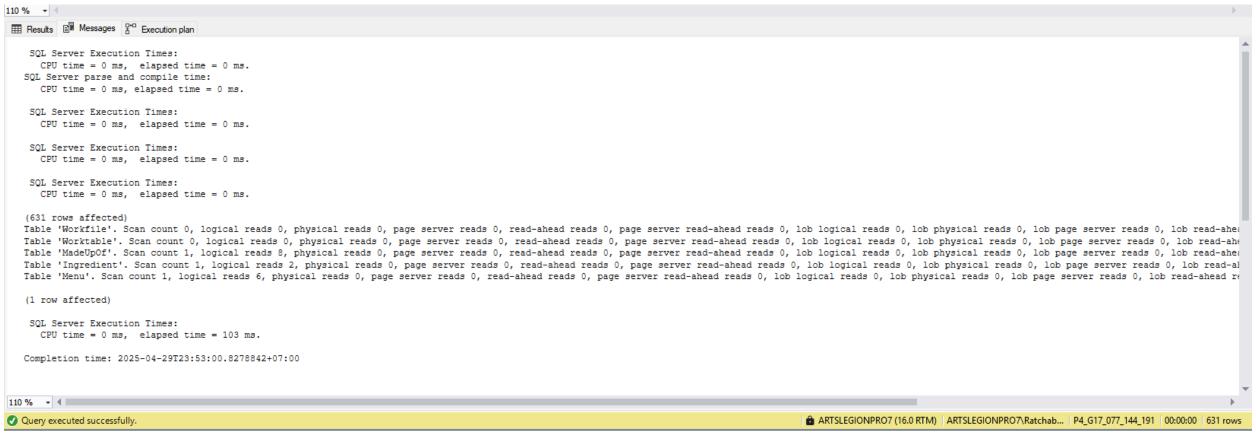
3. Track Inventory Ingredients in Menus

Transaction Analysis Form		
		April 30, 2025
Transaction:	To view which ingredients are used in which menu items and in what quantity.	
Volume:	Average	
	Peak	

```
SELECT i.ing_name, i.ing_quantity,
       m.menu_name, mo.ingredient_quantity
FROM Ingredient i
JOIN MadeUpOf mo ON i.ing_id =
mo.ingredient_ing_id
JOIN Menu m ON mo.ingredient_menu_id =
m.menu_id
WHERE i.ing_quantity > 0;
```

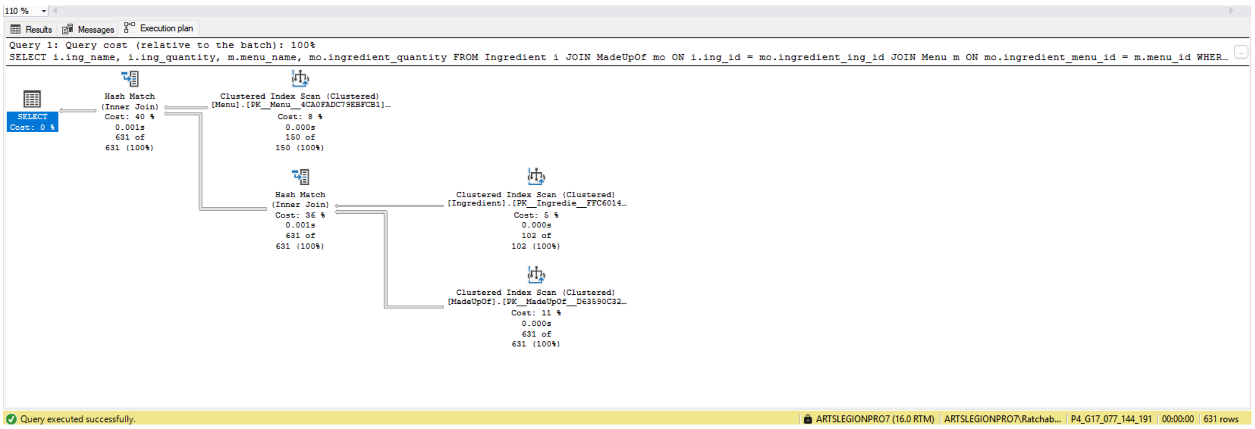
Predicate:	`ing_quantity > 0`
Join Attributes:	`ing_id, menu_id`
Ordering attributes:	-
Grouping attributes:	-
Built-in functions:	-
Attributes updated	-

Access	Entity	Type of Access	No. of References	
			Per Transaction	Peek Per Hour
1	`MadeUpOf`	R	1	500
2	`Ingredient`	R	1	500
3	`Menu`	R	1	500



```
 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.
SQL Server parse and compile time:
   CPU time = 0 ms, elapsed time = 0 ms.

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.

(631 rows affected)
Table 'Workfile'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahea
Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahe
Table 'MadeUpOf'. Scan count 1, logical reads 8, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahea
Table 'Ingredient'. Scan count 1, logical reads 2, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ah
Table 'Menu'. Scan count 1, logical reads 6, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahead re

(1 row affected)

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 103 ms.

Completion time: 2025-04-29T23:53:00.8278842+07:00
```

Query executed successfully. | ARTSLEGIONPRO7 (16.0 RTM) | ARTSLEGIONPRO7\Ratchab... | P4_G17_077_144_191 | 00:00:00 | 631 rows



```
Query 1: Query cost (relative to the batch): 100%
SELECT i.ing_name, i.ing_quantity, m.menu_name, mo.ingredient_quantity FROM Ingredient i JOIN MadeUpOf mo ON i.ing_id = mo.ingredient_ing_id JOIN Menu m ON mo.ingredient_menu_id = m.menu_id WHER…
```

Query executed successfully. | ARTSLEGIONPRO7 (16.0 RTM) | ARTSLEGIONPRO7\Ratchab... | P4_G17_077_144_191 | 00:00:00 | 631 rows

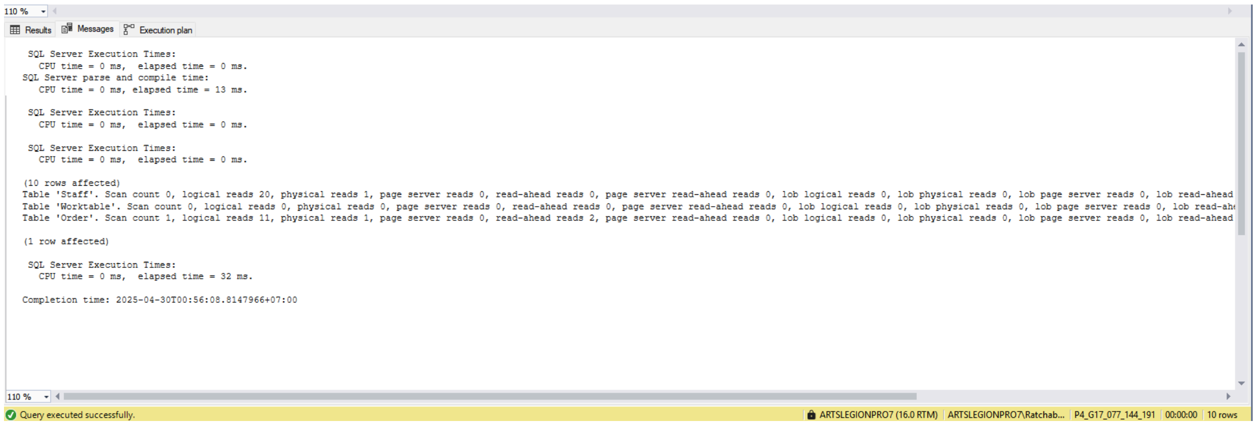
4. Monitor Active Staff Order Assignments

Transaction Analysis Form		
		April 30, 2025
Transaction:	To count how many orders each staff member has handled in a specific date range.	
Volume:	Average	
	Peak	

`SELECT s.staff_id, s.staff_fname, s.staff_lname, COUNT(o.order_id) AS total_orders`
`FROM Staff s`
`JOIN [Order] o ON s.staff_id = o.staff_id`
`WHERE o.order_date BETWEEN '2024-11-20' AND '2025-02-21'`
`GROUP BY s.staff_id, s.staff_fname, s.staff_lname`
`ORDER BY total_orders DESC;`
	Predicate:	`o.order_date BETWEEN '2024-11-20' AND '2025-02-21'`

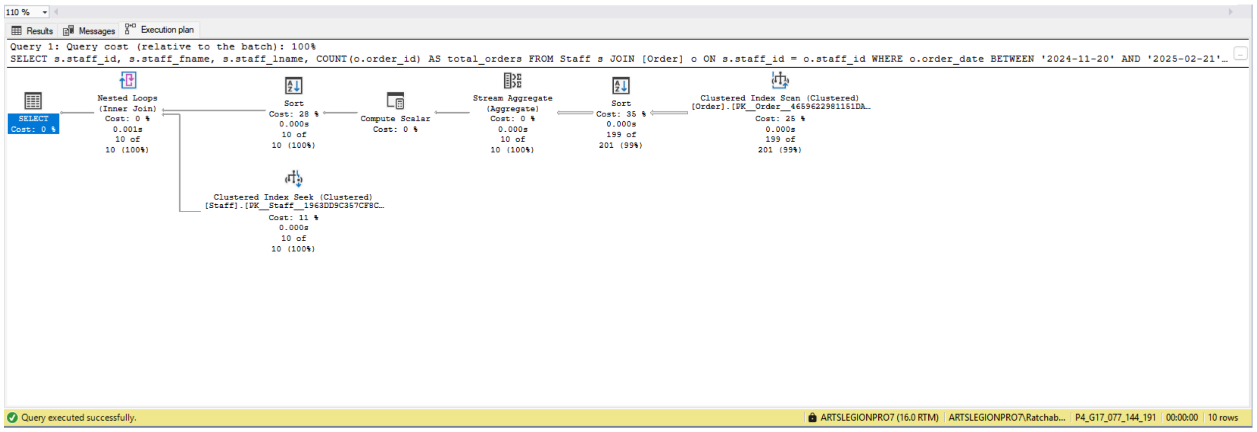
	Join Attributes:	`staff_id`
	Ordering attributes:	`total_orders`
	Grouping attributes:	`s.staff_id, s.staff_fname, s.staff_lname`
	Built-in functions:	-
	Attributes updated	-

Access	Entity	Type of Access	No. of References	
			Per Transaction	Peek Per Hour
1	Staff	R	1	500
2	Order	R	1	500



```
 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.
SQL Server parse and compile time:
   CPU time = 0 ms, elapsed time = 13 ms.

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.

(10 rows affected)
Table 'Staff'. Scan count 0, logical reads 20, physical reads 1, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahead
Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahe
Table 'Order'. Scan count 1, logical reads 11, physical reads 1, page server reads 0, read-ahead reads 2, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahead

(1 row affected)

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 32 ms.

Completion time: 2025-04-30T00:56:08.8147966+07:00
```

Query executed successfully. ARTSLEGIONPRO7 (16.0 RTM) | ARTSLEGIONPRO7\Ratchab... | P4_G17_077_144_191 | 00:00:00 | 10 rows



```
Query 1: Query cost (relative to the batch): 100%
SELECT s.staff_id, s.staff_fname, s.staff_lname, COUNT(o.order_id) AS total_orders FROM Staff s JOIN [Order] o ON s.staff_id = o.staff_id WHERE o.order_date BETWEEN '2024-11-20' AND '2025-02-21'…
```

Query executed successfully. ARTSLEGIONPRO7 (16.0 RTM) | ARTSLEGIONPRO7\Ratchab... | P4_G17_077_144_191 | 00:00:00 | 10 rows

6.Performance After Optimization

Apply Indexing technique

1. Order Processing Query:

```
CREATE NONCLUSTERED INDEX idx_order_id
ON [Order](order_id);
CREATE NONCLUSTERED INDEX idx_consist_order_id
ON ConsistOf(consist_order_id);
```

2. Inventory Tracking Query:

```
CREATE NONCLUSTERED INDEX idx_ingredient_menu_id
ON MadeUpOf(ingredient_menu_id);
CREATE NONCLUSTERED INDEX idx_ingredient_id
ON Ingredient(ing_id);
```

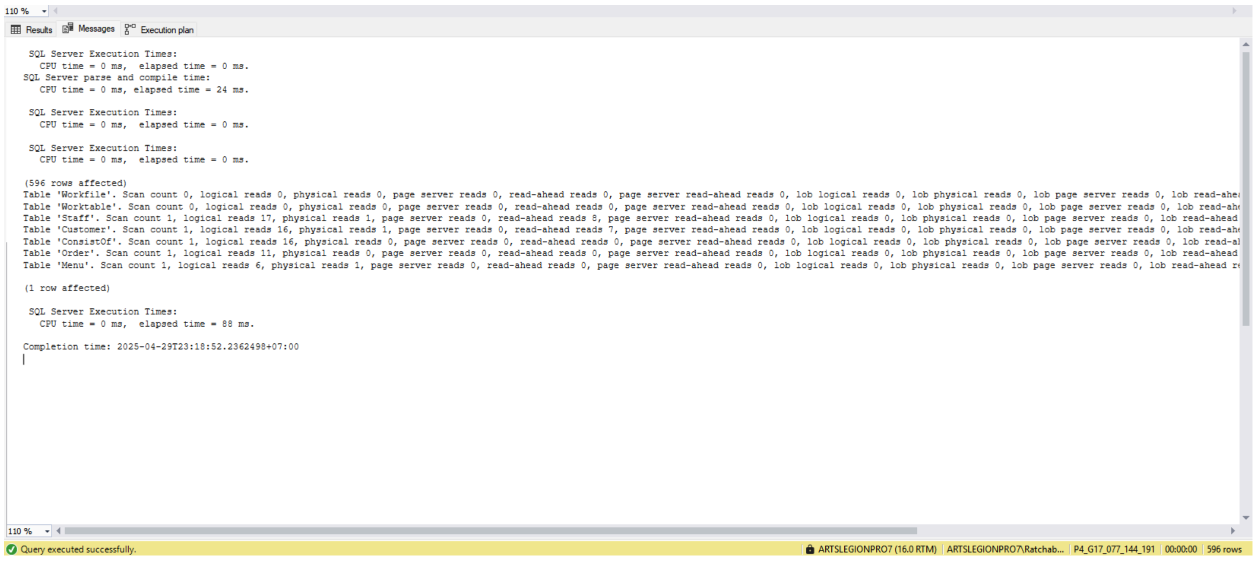
3. Track Inventory Ingredients in Menus

```
CREATE INDEX idx_ingredient_stock_level
ON Ingredient (ing_quantity);
CREATE INDEX idx_madeupof_ingredient_menu
ON MadeUpOf (ingredient_ing_id, ingredient_menu_id);
```

4. Monitor Active Staff Order Assignments

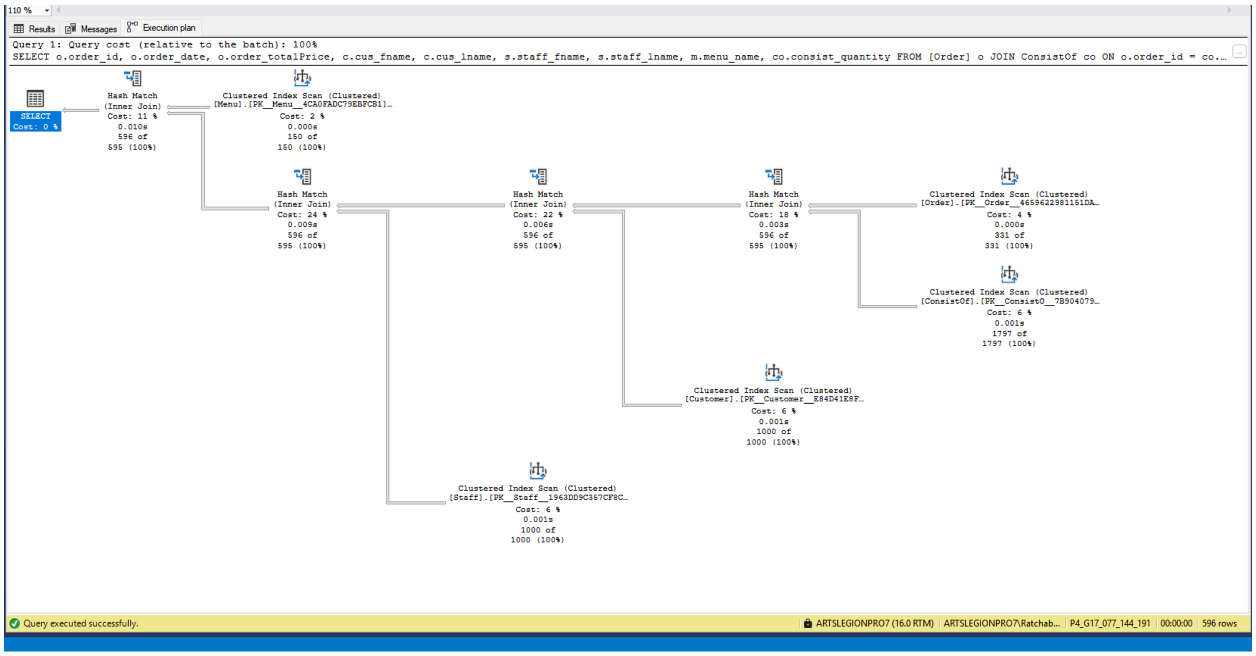
```
CREATE INDEX idx_order_staff_date
ON [Order](staff_id, order_date);
CREATE INDEX idx_staff_id
ON Staff(staff_id);
```

1. Analyze Order Processing Query



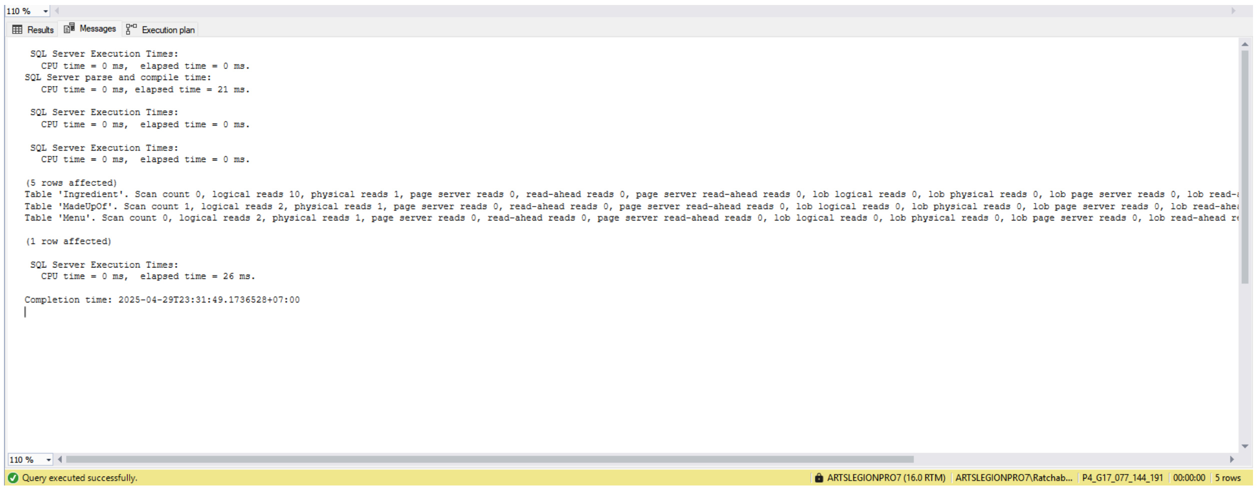
```
 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.
SQL Server parse and compile time:
   CPU time = 0 ms, elapsed time = 24 ms.

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.

(596 rows affected)
Table 'Workfile'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahea
Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahe
Table 'Staff'. Scan count 1, logical reads 17, physical reads 1, page server reads 0, read-ahead reads 8, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahead
Table 'Customer'. Scan count 1, logical reads 16, physical reads 1, page server reads 0, read-ahead reads 7, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahe
Table 'ConsistOf'. Scan count 1, logical reads 16, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ah
Table 'Order'. Scan count 1, logical reads 11, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahead
Table 'Menu'. Scan count 1, logical reads 6, physical reads 1, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahead re

(1 row affected)

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 88 ms.

Completion time: 2025-04-29T23:18:52.2362498+07:00
```

110 %
Results
Messages
Execution plan
110 %
Query executed successfully.
ARTSLEGIONPRO7 (16.0 RTM)
ARTSLEGIONPRO7\Ratchab...
P4_G17_077_144_191
00:00:00
596 rows



```
Query 1: Query cost (relative to the batch): 100%
SELECT o.order_id, o.order_date, o.order_totalPrice, c.cus_fname, c.cus_lname, s.staff_fname, s.staff_lname, m.menu_name, co.consist_quantity FROM [Order] o JOIN ConsistOf co ON o.order_id = co.…
```

2. Inventory Tracking Query



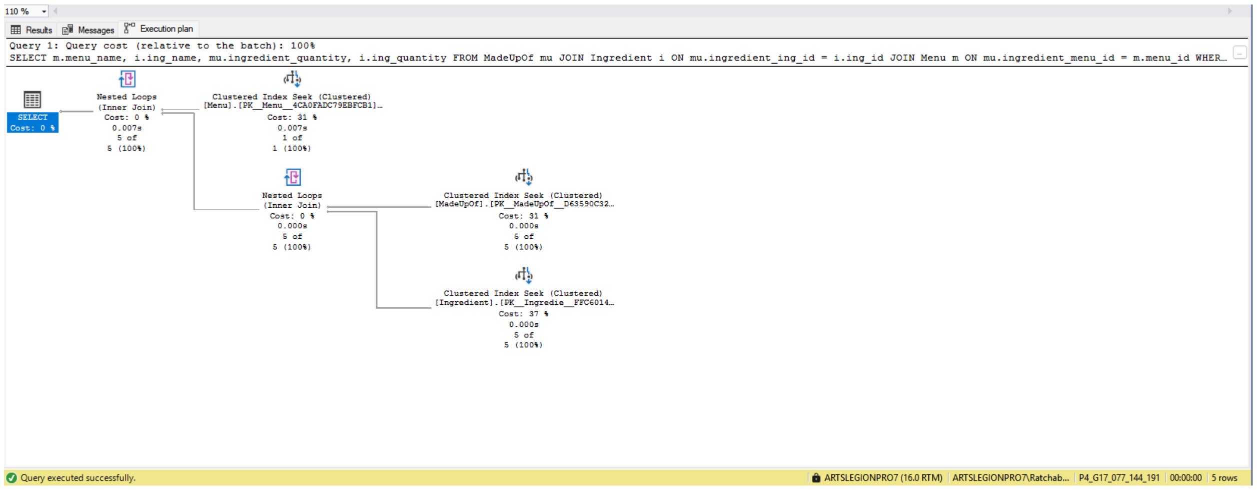
110 %

Results | Messages | Execution plan

```
 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.
SQL Server parse and compile time:
   CPU time = 0 ms, elapsed time = 21 ms.

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.

(5 rows affected)
Table 'Ingredient'. Scan count 0, logical reads 10, physical reads 1, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-a
Table 'MadeUpOf'. Scan count 1, logical reads 2, physical reads 1, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahea
Table 'Menu'. Scan count 0, logical reads 2, physical reads 1, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahead re

(1 row affected)

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 26 ms.

Completion time: 2025-04-29T23:31:49.1736528+07:00
```

110 %

Query executed successfully. | ARTSLEGIONPRO7 (16.0 RTM) | ARTSLEGIONPRO7\Ratchab... | P4_G17_077_144_191 | 00:00:00 | 5 rows



110 %

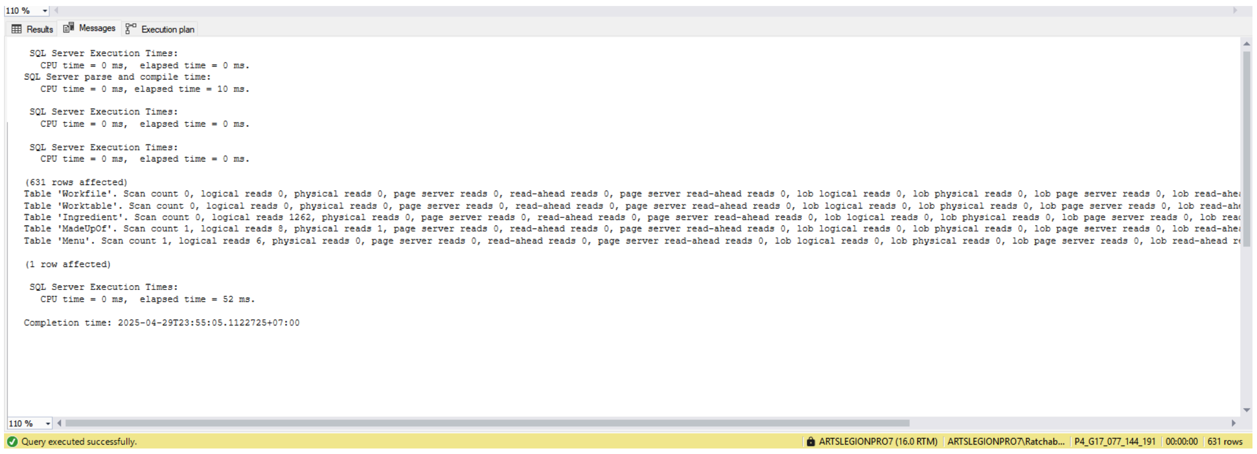
Results | Messages | Execution plan

```
Query 1: Query cost (relative to the batch): 100%
SELECT m.menu_name, i.ing_name, mu.ingredient_quantity, i.ing_quantity FROM MadeUpOf mu JOIN Ingredient i ON mu.ingredient_ing_id = i.ing_id JOIN Menu m ON mu.ingredient_menu_id = m.menu_id WHER...
```

SELECT
Cost: 0 %
Nested Loops
(Inner Join)
Cost: 0 %
0.007s
5 of
5 (100%)
Clustered Index Seek (Clustered)
[Menu].[PK__Menu__4CA0FADC79EBFCB1]...
Cost: 31 %
0.007s
1 of
1 (100%)
Nested Loops
(Inner Join)
Cost: 0 %
0.000s
5 of
5 (100%)
Clustered Index Seek (Clustered)
[MadeUpOf].[PK__MadeUpOf__D63590C32...
Cost: 31 %
0.000s
5 of
5 (100%)
Clustered Index Seek (Clustered)
[Ingredient].[PK__Ingredie__FFC6014...
Cost: 37 %
0.000s
5 of
5 (100%)

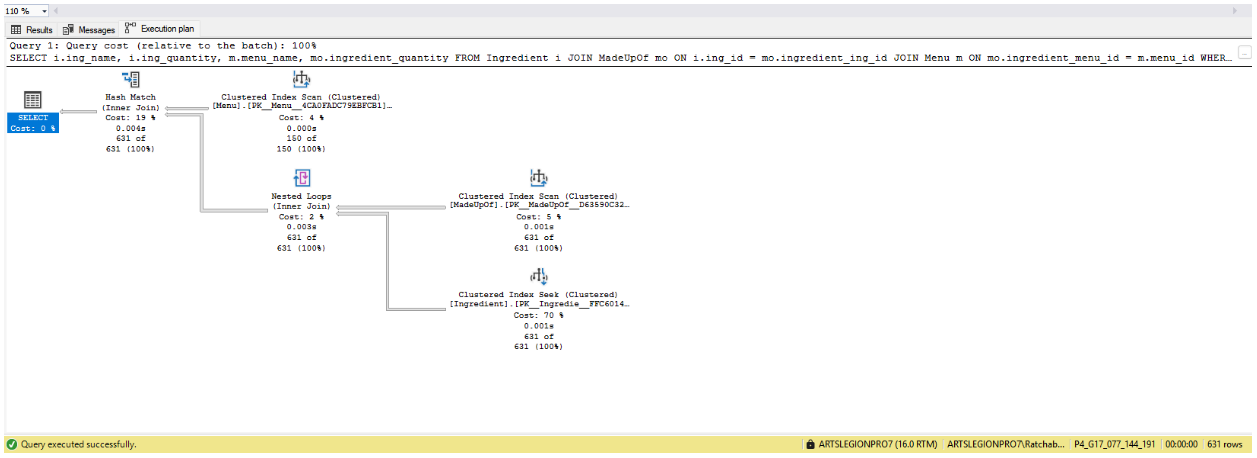
Query executed successfully. | ARTSLEGIONPRO7 (16.0 RTM) | ARTSLEGIONPRO7\Ratchab... | P4_G17_077_144_191 | 00:00:00 | 5 rows

3. Track Inventory Ingredients in Menus



```
 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.
SQL Server parse and compile time:
   CPU time = 0 ms, elapsed time = 10 ms.

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.

(631 rows affected)
Table 'Workfile'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahe
Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ah
Table 'Ingredient'. Scan count 0, logical reads 1262, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob rea
Table 'MadeUpOf'. Scan count 1, logical reads 8, physical reads 1, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahe
Table 'Menu'. Scan count 1, logical reads 6, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahead r

(1 row affected)

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 52 ms.

Completion time: 2025-04-29T23:55:05.1122725+07:00
```

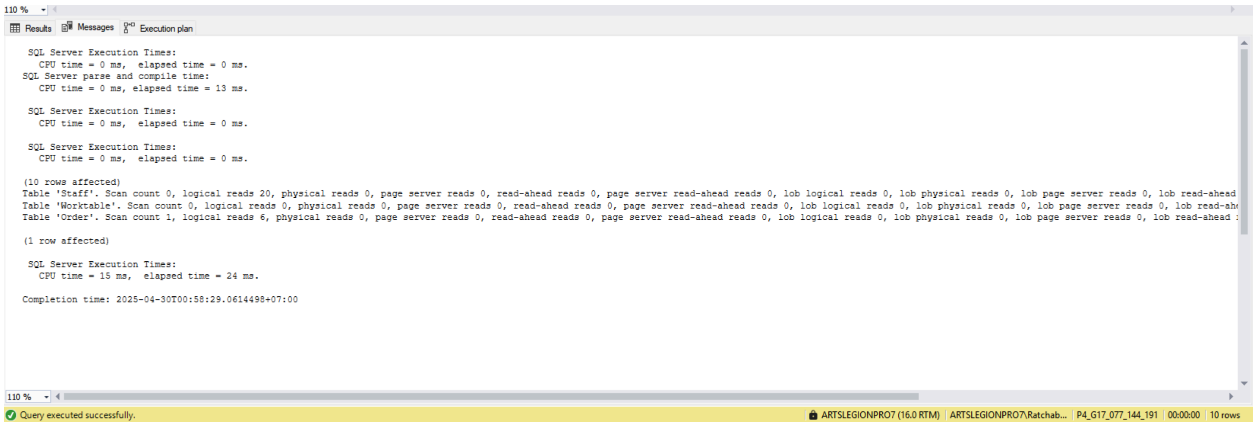
110 %
Results
Messages
Execution plan
Query executed successfully.
ARTSLEGIONPRO7 (16.0 RTM) | ARTSLEGIONPRO7\Ratchab... | P4_G17_077_144_191 | 00:00:00 | 631 rows



```
Query 1: Query cost (relative to the batch): 100%
SELECT i.ing_name, i.ing_quantity, m.menu_name, mo.ingredient_quantity FROM Ingredient i JOIN MadeUpOf mo ON i.ing_id = mo.ingredient_ing_id JOIN Menu m ON mo.ingredient_menu_id = m.menu_id WHER…
```

110 %
Results
Messages
Execution plan
SELECT
Cost: 0 %
Hash Match
(Inner Join)
Cost: 19 %
0.004s
631 of
631 (100%)
Clustered Index Scan (Clustered)
[Menu].[PK__Menu__4CA0FADC79EBFCB1]…
Cost: 4 %
0.000s
150 of
150 (100%)
Nested Loops
(Inner Join)
Cost: 2 %
0.003s
631 of
631 (100%)
Clustered Index Scan (Clustered)
[MadeUpOf].[PK__MadeUpOf__D63590C32…
Cost: 5 %
0.001s
631 of
631 (100%)
Clustered Index Seek (Clustered)
[Ingredient].[PK__Ingredie__FFC6014…
Cost: 70 %
0.001s
631 of
631 (100%)
Query executed successfully.
ARTSLEGIONPRO7 (16.0 RTM) | ARTSLEGIONPRO7\Ratchab... | P4_G17_077_144_191 | 00:00:00 | 631 rows

4. Monitor Active Staff Order Assignments



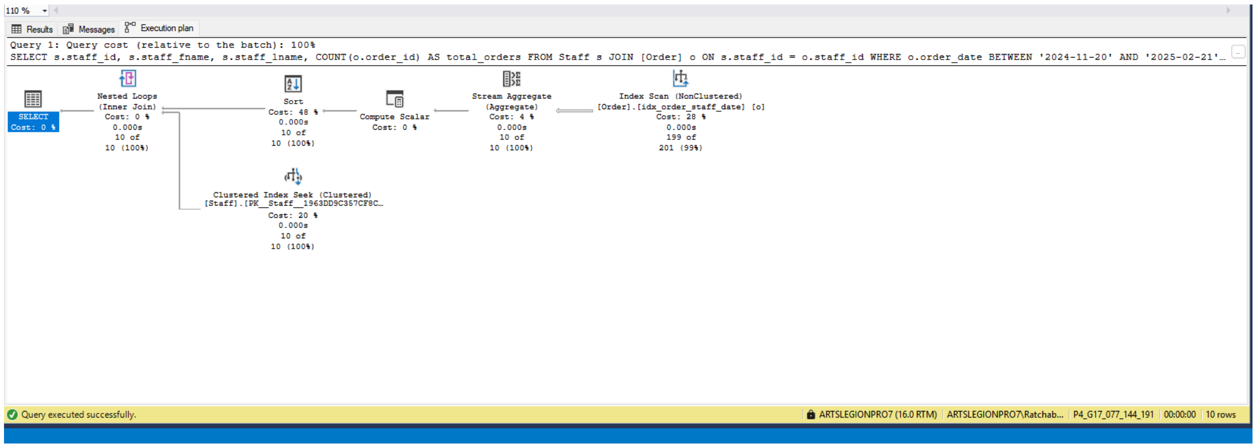
110 %

Results | Messages | Execution plan

```
 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.
SQL Server parse and compile time:
   CPU time = 0 ms, elapsed time = 13 ms.

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.

 SQL Server Execution Times:
   CPU time = 0 ms,  elapsed time = 0 ms.

(10 rows affected)
Table 'Staff'. Scan count 0, logical reads 20, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahead
Table 'Worktable'. Scan count 0, logical reads 0, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahe
Table 'Order'. Scan count 1, logical reads 6, physical reads 0, page server reads 0, read-ahead reads 0, page server read-ahead reads 0, lob logical reads 0, lob physical reads 0, lob page server reads 0, lob read-ahead

(1 row affected)

 SQL Server Execution Times:
   CPU time = 15 ms,  elapsed time = 24 ms.

Completion time: 2025-04-30T00:58:29.0614498+07:00
```

110 %

Query executed successfully. | ARTSLEGIONPRO7 (16.0 RTM) | ARTSLEGIONPRO7\Ratchab... | P4_G17_077_144_191 | 00:00:00 | 10 rows



110 %

Results | Messages | Execution plan

```
Query 1: Query cost (relative to the batch): 100%
SELECT s.staff_id, s.staff_fname, s.staff_lname, COUNT(o.order_id) AS total_orders FROM Staff s JOIN [Order] o ON s.staff_id = o.staff_id WHERE o.order_date BETWEEN '2024-11-20' AND '2025-02-21'…
```

Query executed successfully. | ARTSLEGIONPRO7 (16.0 RTM) | ARTSLEGIONPRO7\Ratchab... | P4_G17_077_144_191 | 00:00:00 | 10 rows

7. Results & Discussion

7.1 Performance Comparison Table

Transaction	Execution Time (Before)	Execution Time (After)	Improvement
Order Processing Query	92 ms	88 ms	↓ ~ 4.35%
Inventory by Menu	42 ms	26 ms	↓ ~ 38.01%
Ingredients in Menus	103 ms	52 ms	↓ ~ 49.51%
Monitor Active Staff Order Assignments	32 ms	24 ms	↓ 75%

7.2 Discussion of the results

Based on the Before and After Transaction Analysis, we can clearly see that the execution time of all four key queries significantly decreased after applying indexing techniques to the database schema. Indexing improves the efficiency of data retrieval by allowing the database system to quickly locate the needed records without scanning the entire table.

In conclusion, the indexing techniques applied across the relevant attributes drastically reduced the system's workload during execution. This optimization is critical in a real-time restaurant management system where fast response time supports better service delivery and operational efficiency.