Cairo University, Faculty of Computers and Artificial

FACULTY OF COMPUTERS AND ARTIFICIAL INTELLIGENCE, CAIRO UNIVERSITY

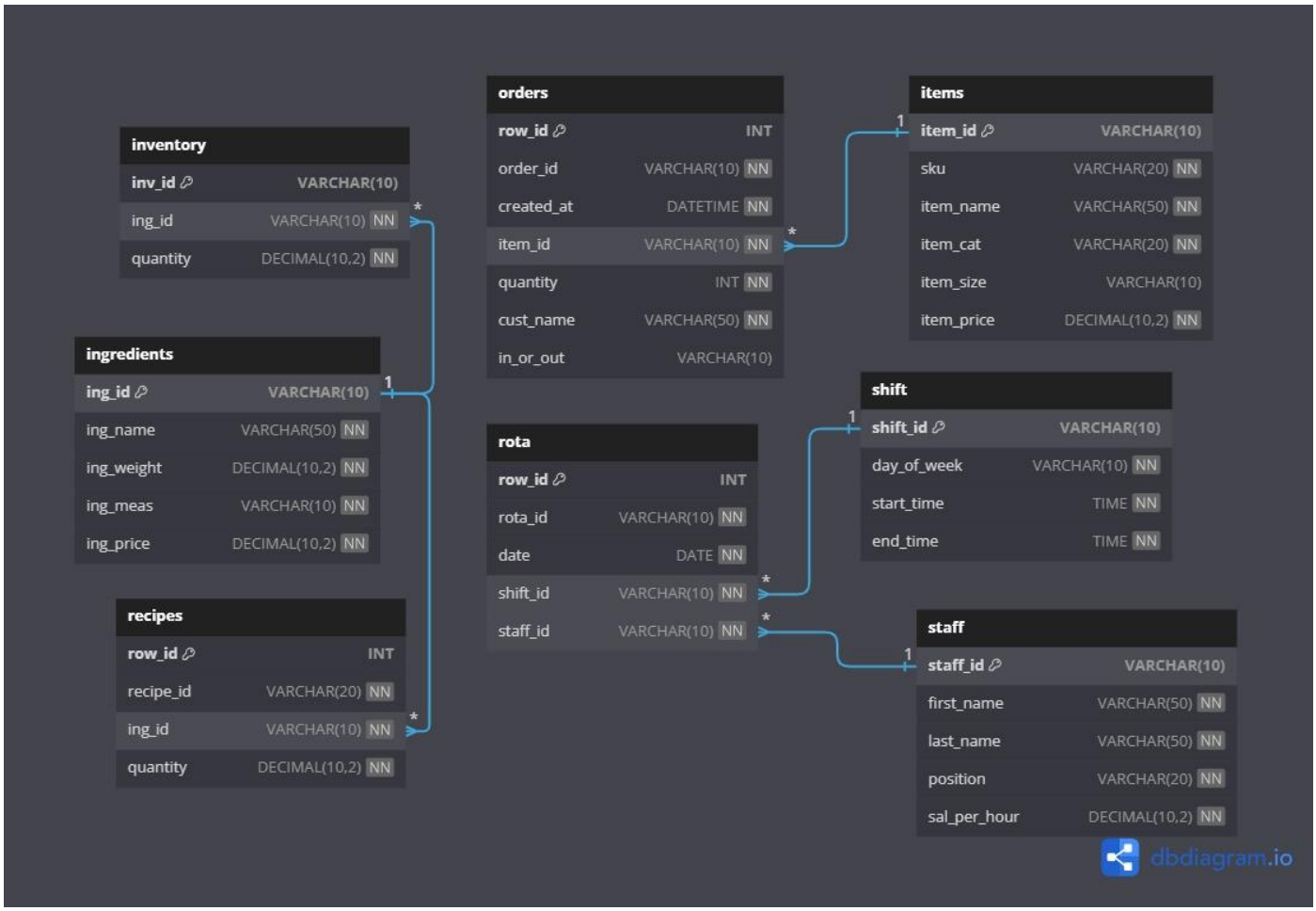
Data Warehouse(Spring 2025)

CoffeeShop DataWarehouse Project

Team Members:

Name	ID
Menna Hassan Helmy	20220345
Nada Ibrahim Mohamed	20220358
Ahmed Hegazy Ahmed	20220015

1. Physical model of the source system



2. The business processes that we will model.

- **Sales Tracking** – capturing item sales per customer and order.
- **Staff Cost Management** – recording staff working hours and related costs.
- **Recipe Cost Analysis** – analyzing the cost and profitability of recipes used in menu items.
- **Customer Visit Analysis** – tracking monthly visits and spending by customers

3. Fact Tables:

1. Fact_Sales

- **Type**: Transaction fact table
- **Measures**:
  - item_quantity (additive)
  - sales_amount (additive)
- **Grain**: One record per item sold in an order on a specific date for a specific customer and order type (captures the quantity and dollar amount for each item sold together)
- **Dimensions**:
  - Date (date_id)
  - Customer (cust_id)
  - Order Type (order_type_id)

2. Fact_Staff_Cost

- **Type**: Periodic Snapshot fact table
- **Measures**:
  - worked_hours (additive)
  - sal_per_hour (non-additive)
  - cost (additive)
- **Granularity**: One record per staff per shift per date, capturing salary and hours worked (records each work session's hours and resulting labor cost for every staff–shift occurrence)
- **Dimensions**:
  - Staff (staff_id)
  - Shift (shift_id)
  - Date (date_id)

3. Fact_Recipe

- **Type**: Transactional fact table
- **Measures**:
  - item_price (additive)
  - total_cost (additive)
  - profit (additive)
- **Granularity**: One record per item-recipe combination, capturing cost and profit. (logs the cost and profit contribution of each ingredient used in a given recipe.)
- **Dimensions**:
  - Recipe (recipe_id)
  - Item (item_id)

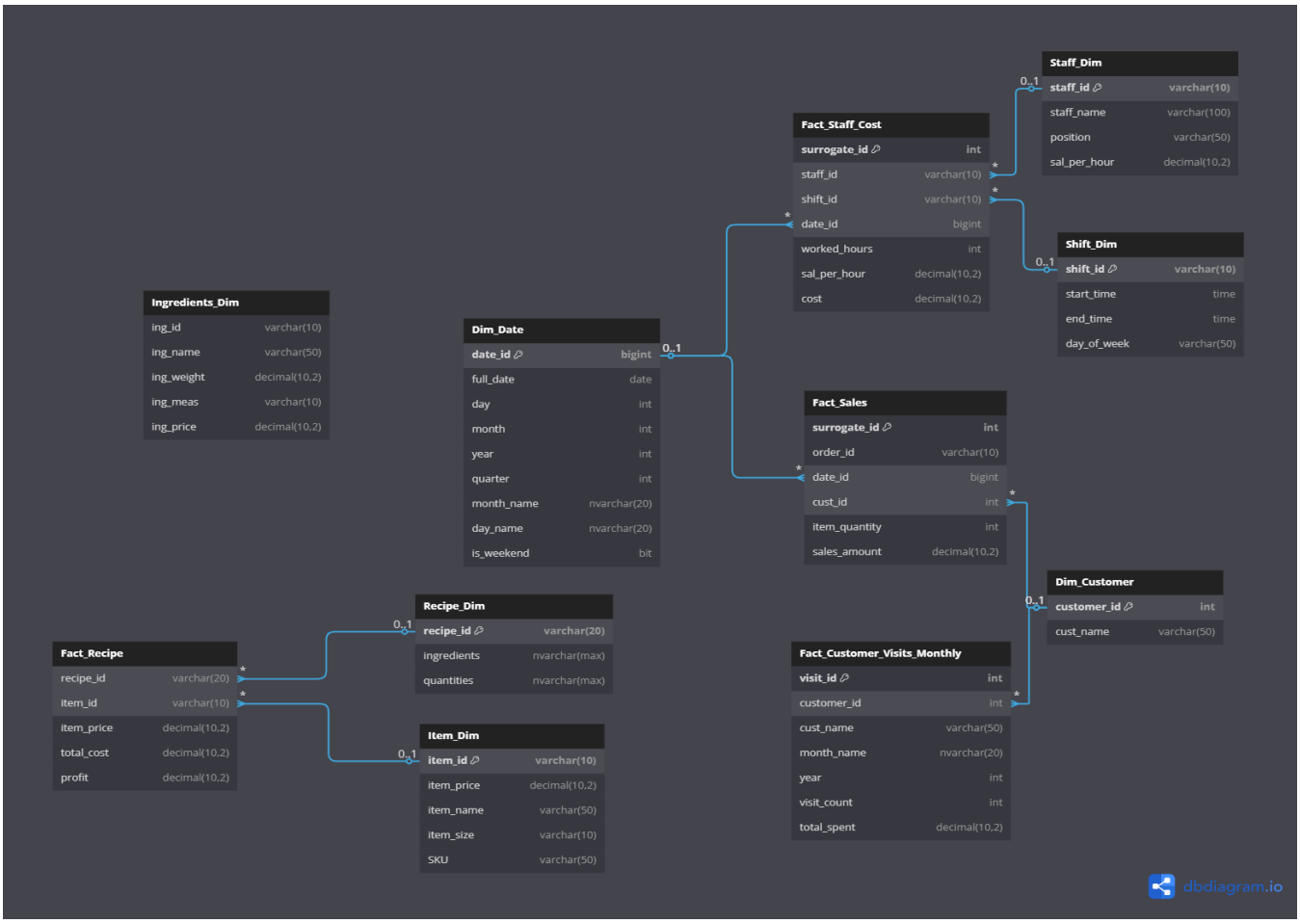
4. Fact_Customer_Visits_Monthly

- **Type**: Periodic snapshot fact table
- **Measures**:
  - visit_count (semi-additive)
  - total_spent (semi-additive)
- **Granularity**: One record per customer per month per year, summarizing visits and total spending. (One record per item-recipe combination, capturing cost and profit.)
- **Dimensions**:
  - Customer (customer_id)

4. Dimension Tables

1. Recipe_Dim
    - **Type**: Slowly Changing (type 1)
2. Ingredients_Dim
    - **Type**: Static
3. Item_Dim
    - **Type**: Slowly Changing (type 1), Conformed
4. Staff_Dim
    - **Type**: Slowly Changing (type 2)
5. Shift_Dim
    - **Type**: Static
6. Dim_Date
    - **Type**: Conformed, Role Playing dimension
7. Dim_Customer
    - **Type**: Conformed, Slowly Changing (type 2)

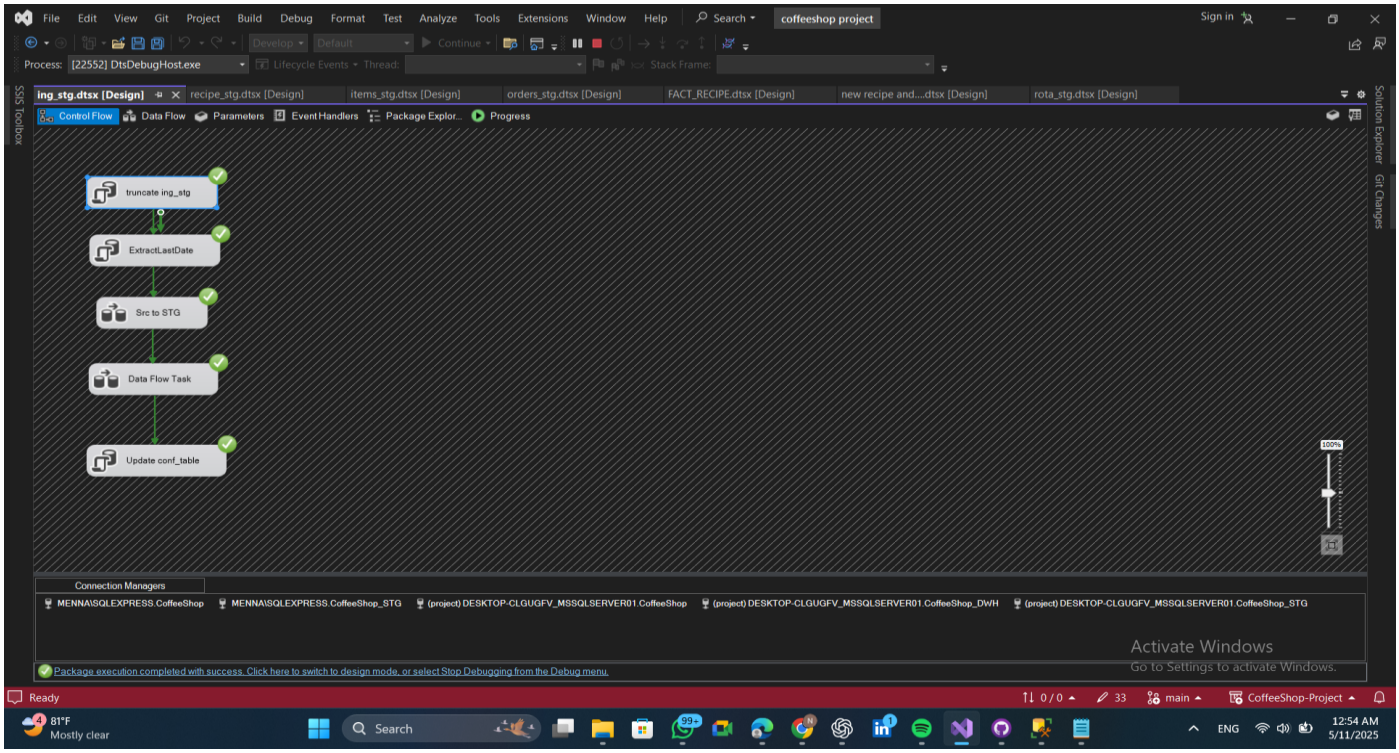
5. Physical Model of The Schema



SCHEMA

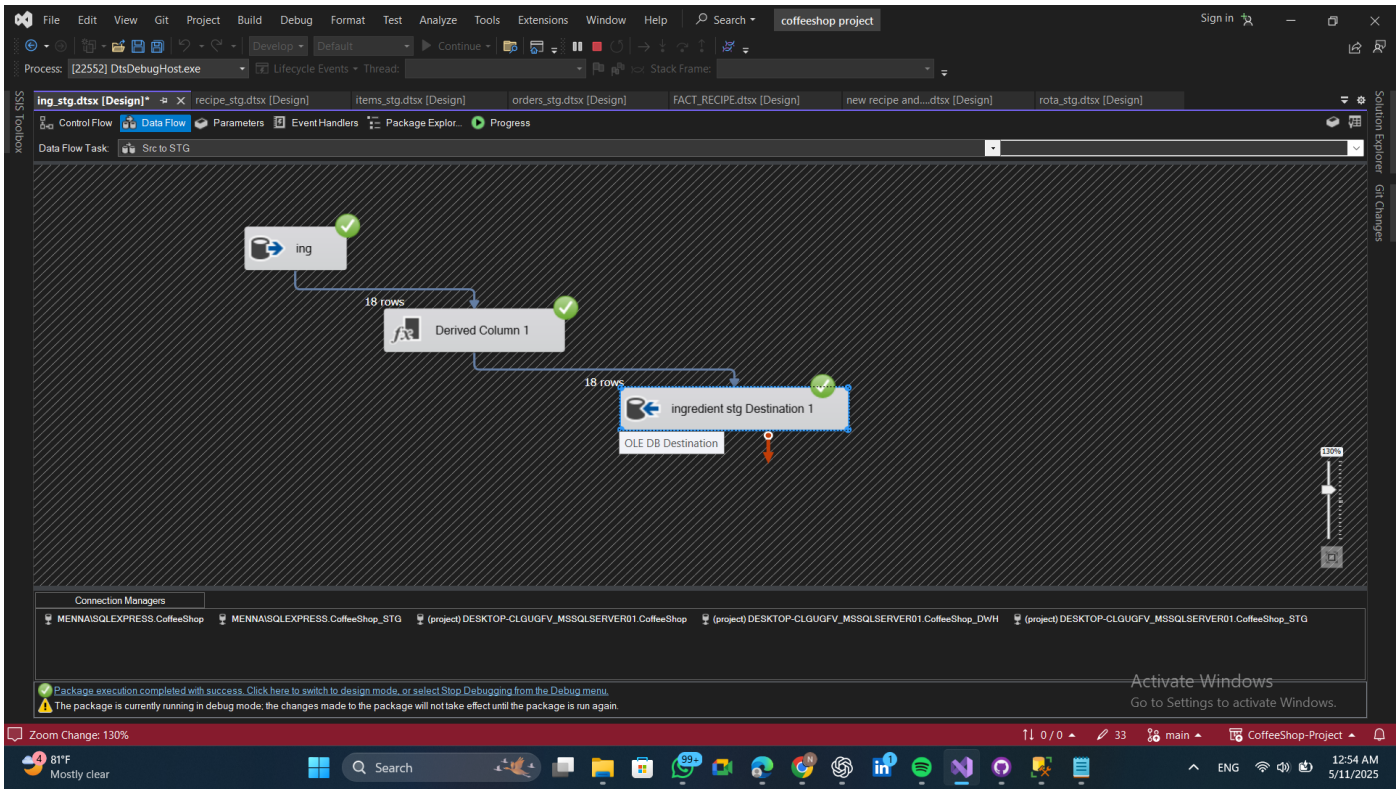
6. SSIS Control and Data Flow

- **Ingredients** control flow → Delete all data from the staging table to prepare for new inserts or updates, and update the last extract date in the configuration table.

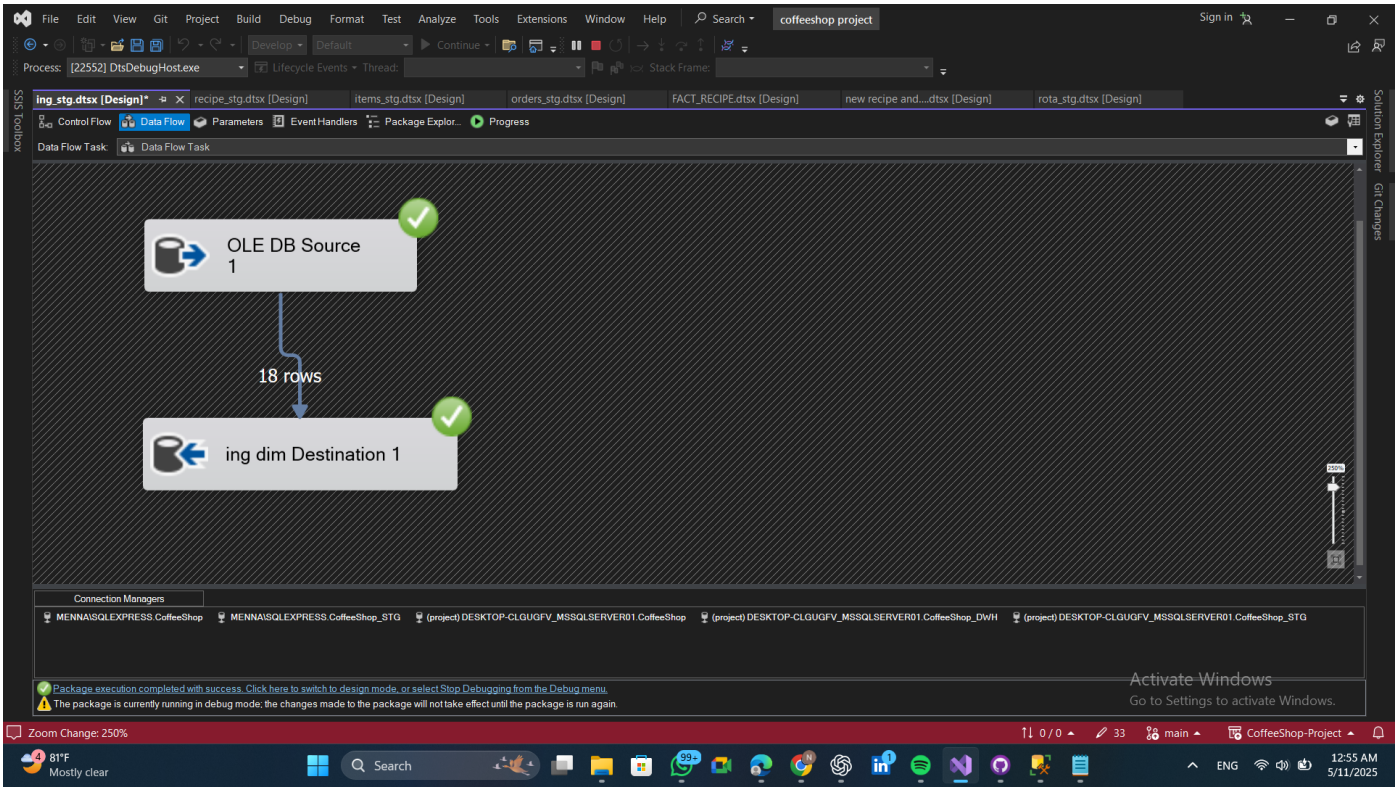


- **Ingredients** data flow

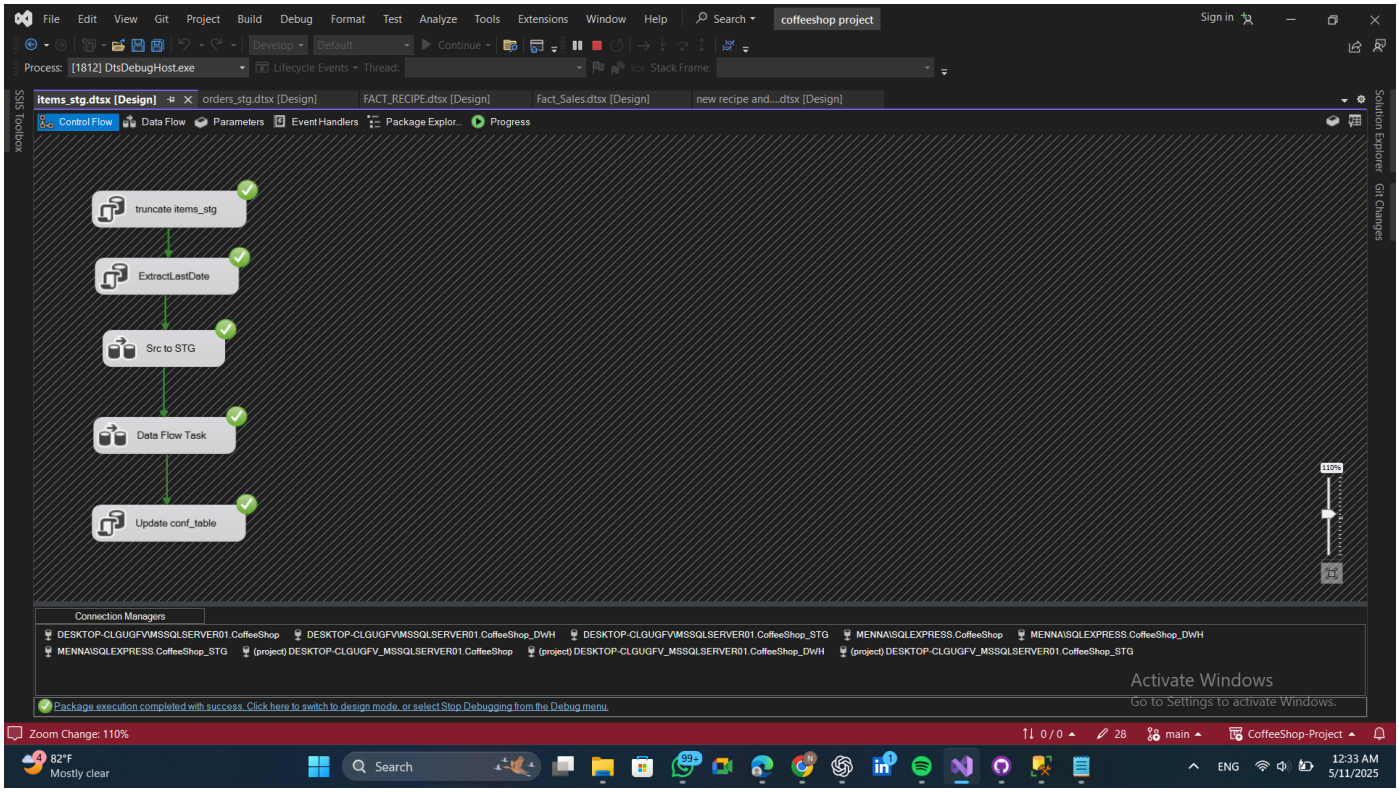
➔ Transfer data from the source database into the ingredient table in the staging area.



➔ Load data from the ingredients table in the staging area into the dimension table in the data warehouse

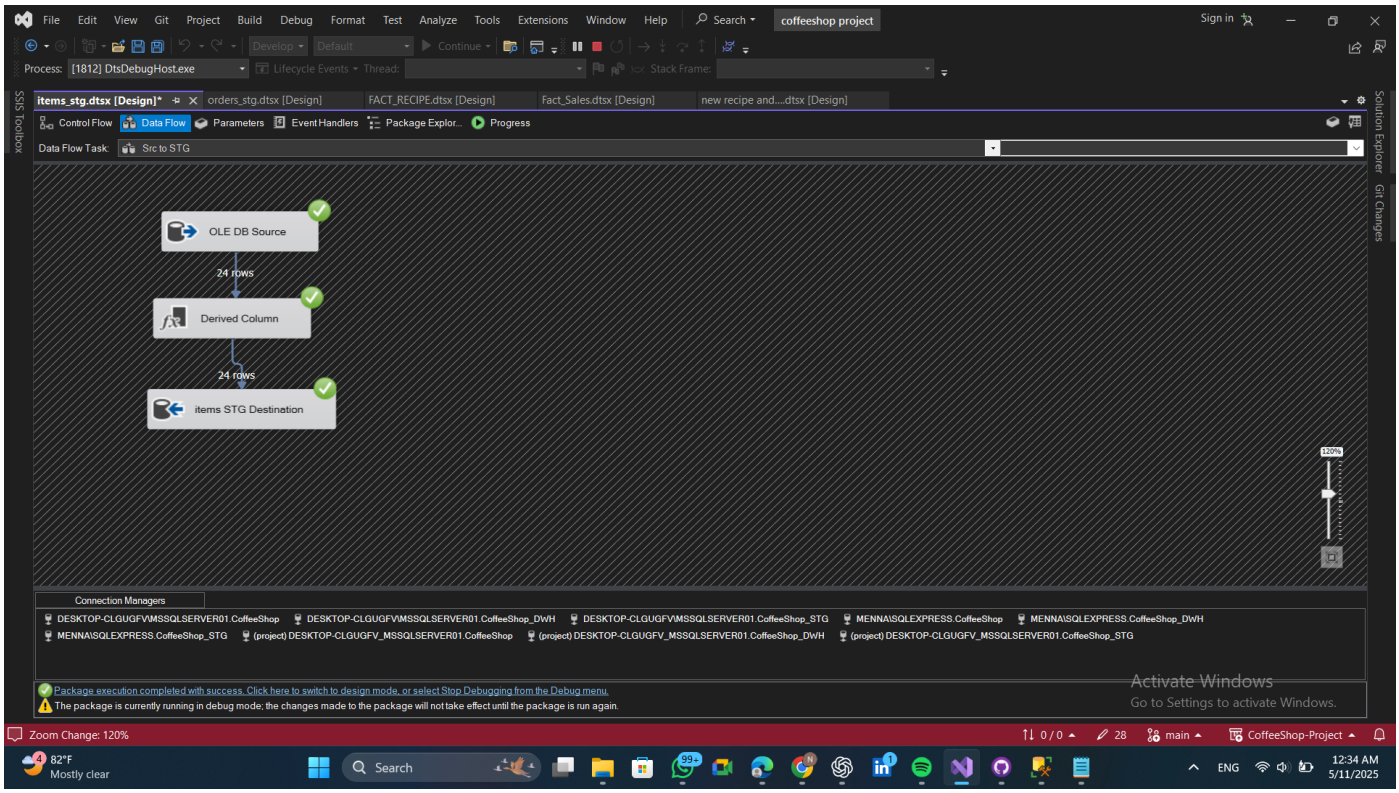


- **Items control flow** → Delete all data from the staging table to prepare for new inserts or updates, and update the last extract date in the configuration table.

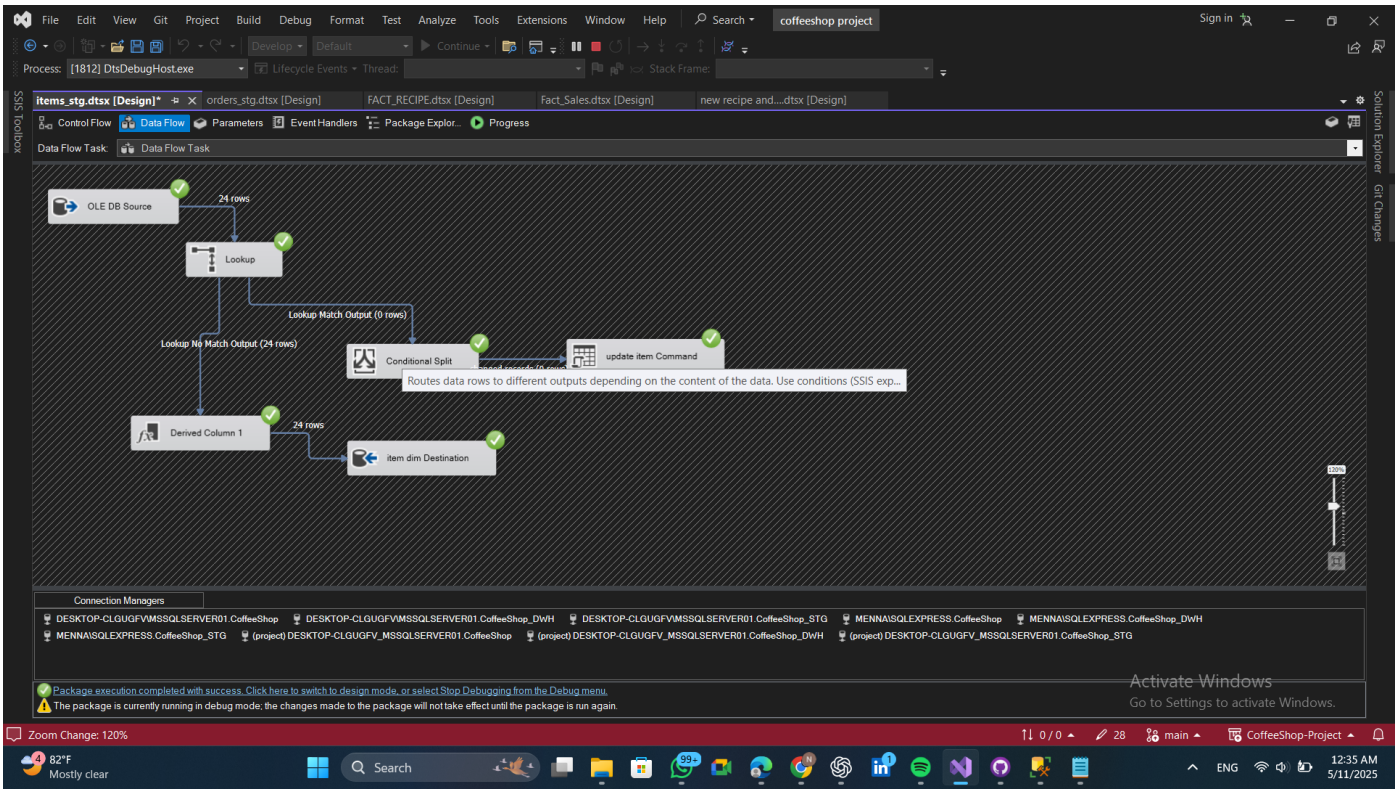


- **Items data flow**

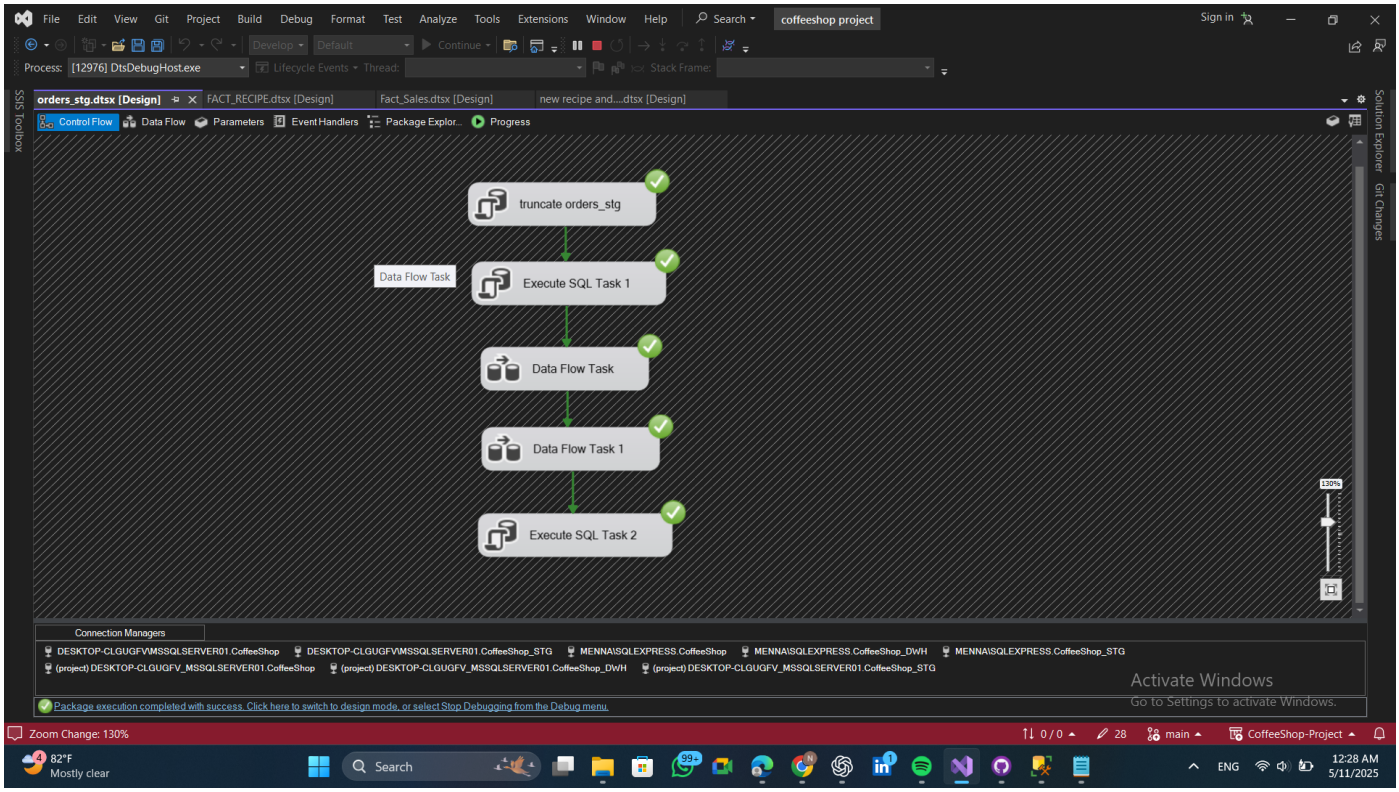
➔ Extract data from the source and load it into the staging area.



➔ Retrieve data from the staging area, perform a lookup to determine if the data is new or updated, and then insert or update the items dimension table accordingly

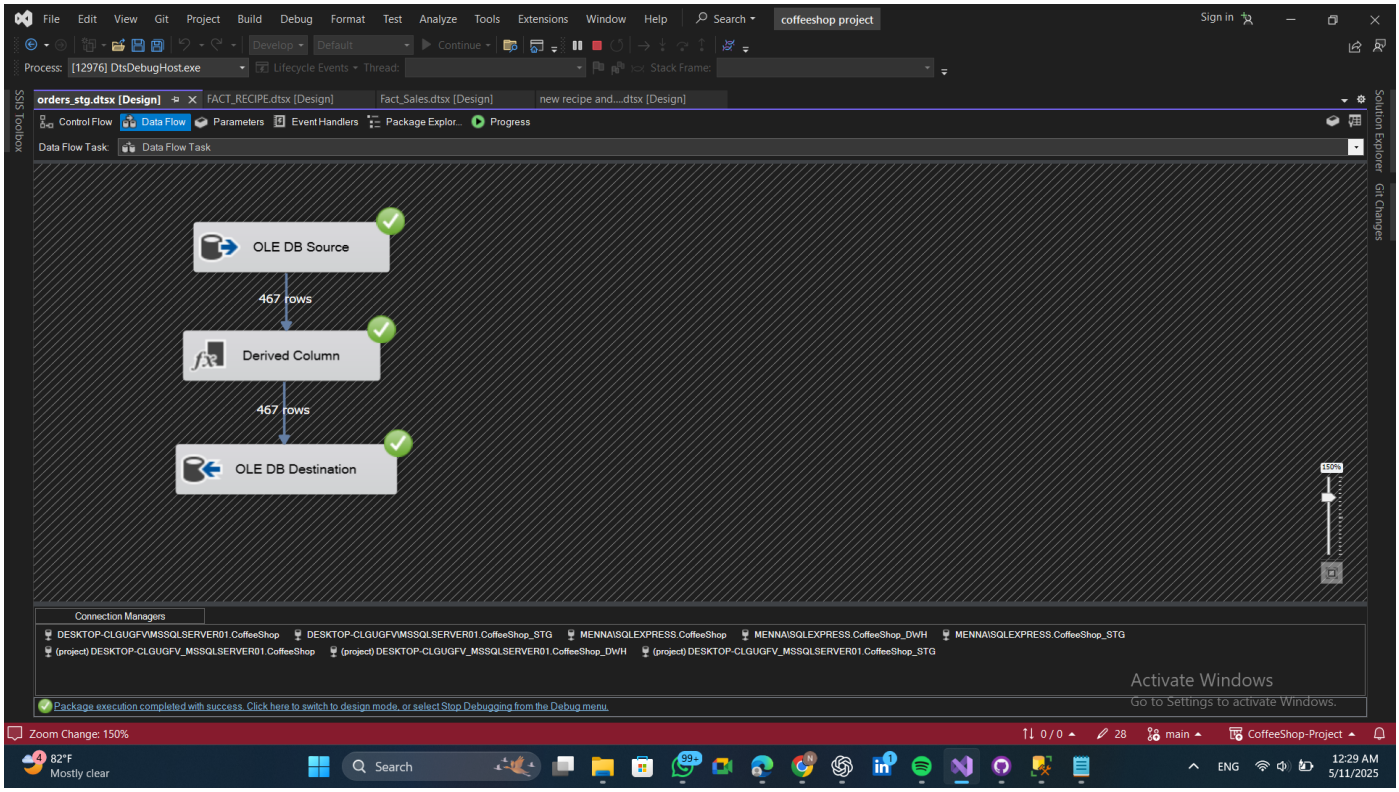


- **Orders control flow** → Delete all data from the staging table to prepare for new inserts or updates, and update the last extract date in the configuration table.

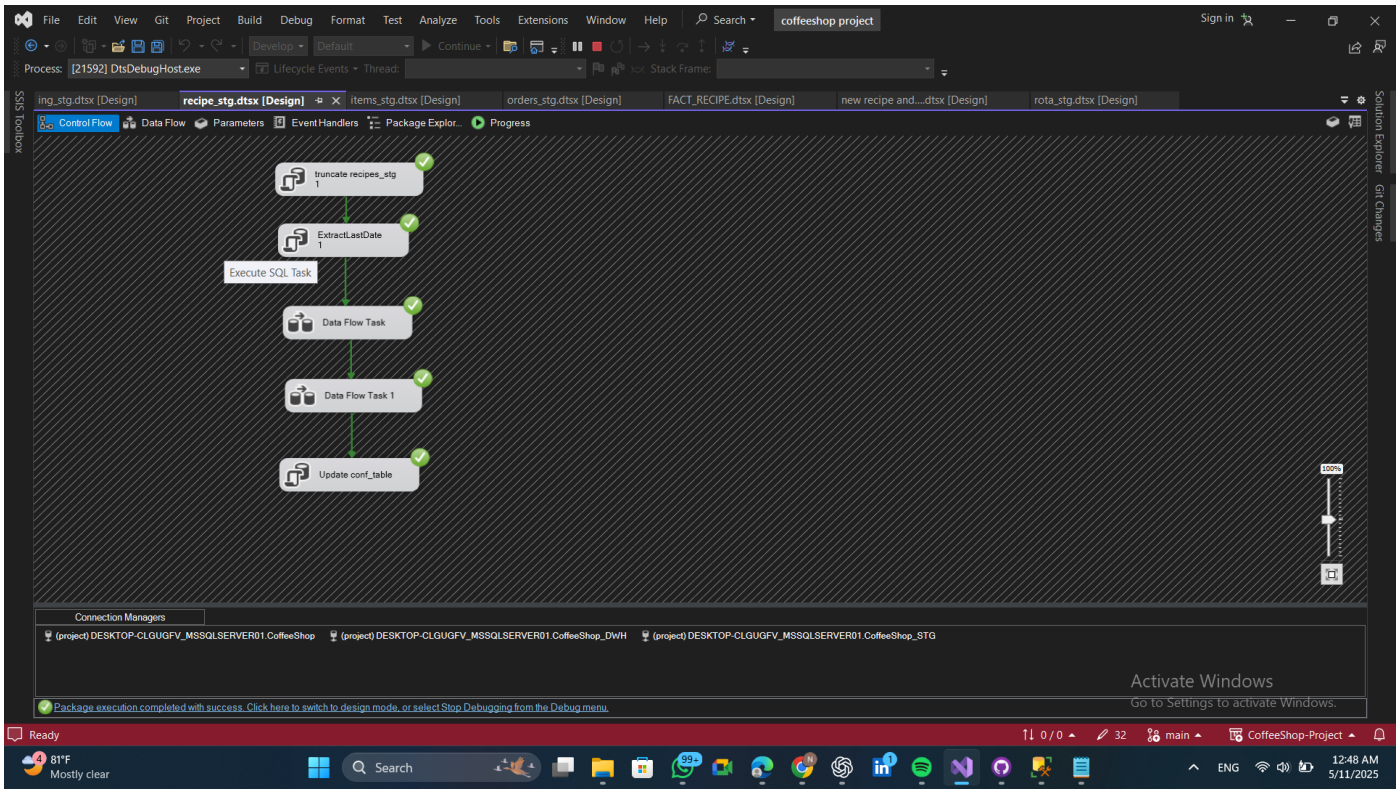


- **Orders data flow (no dimension for order table)**

  ➔ Extract data from the source and load it into the staging area.

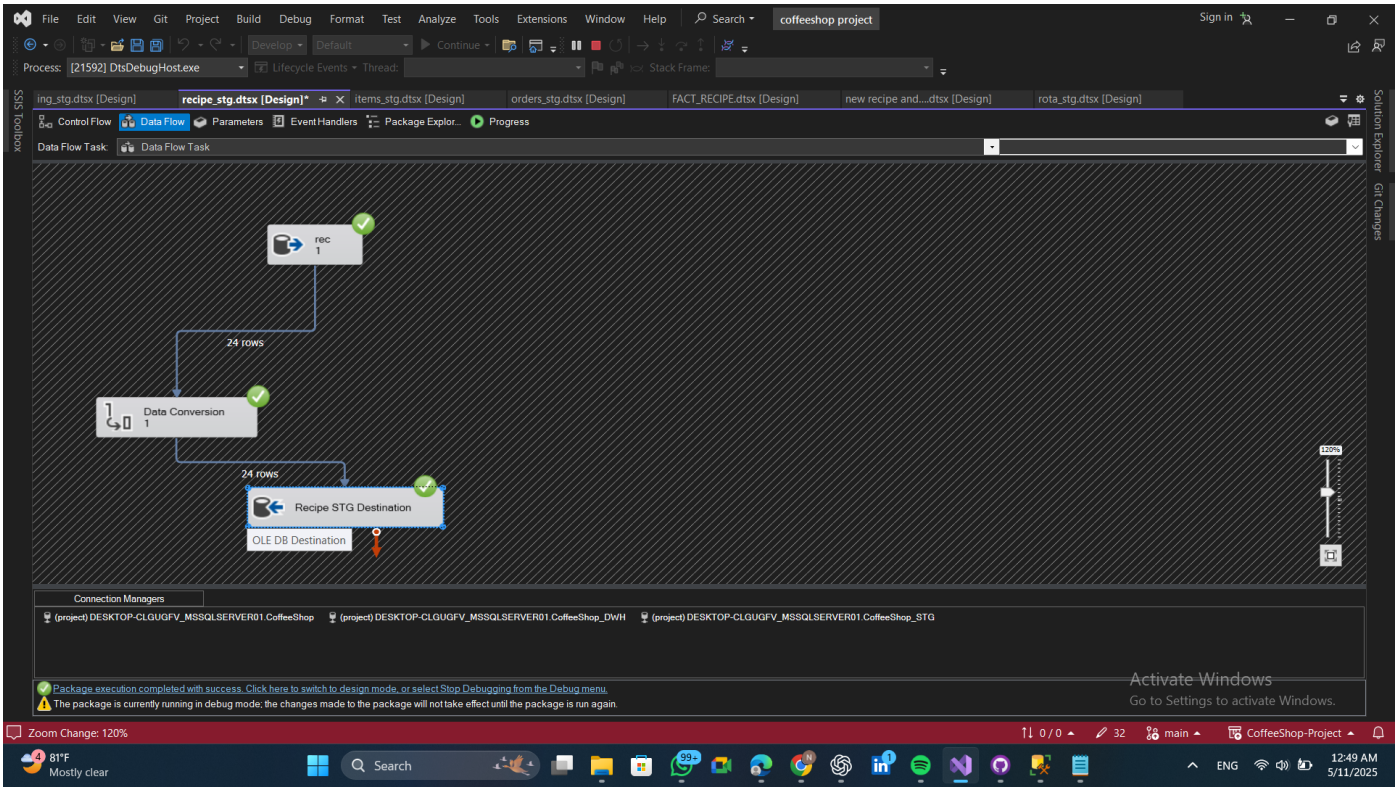


- **Recipe control flow** → Delete all data from the staging table to prepare for new inserts or updates, and update the last extract date in the configuration table.

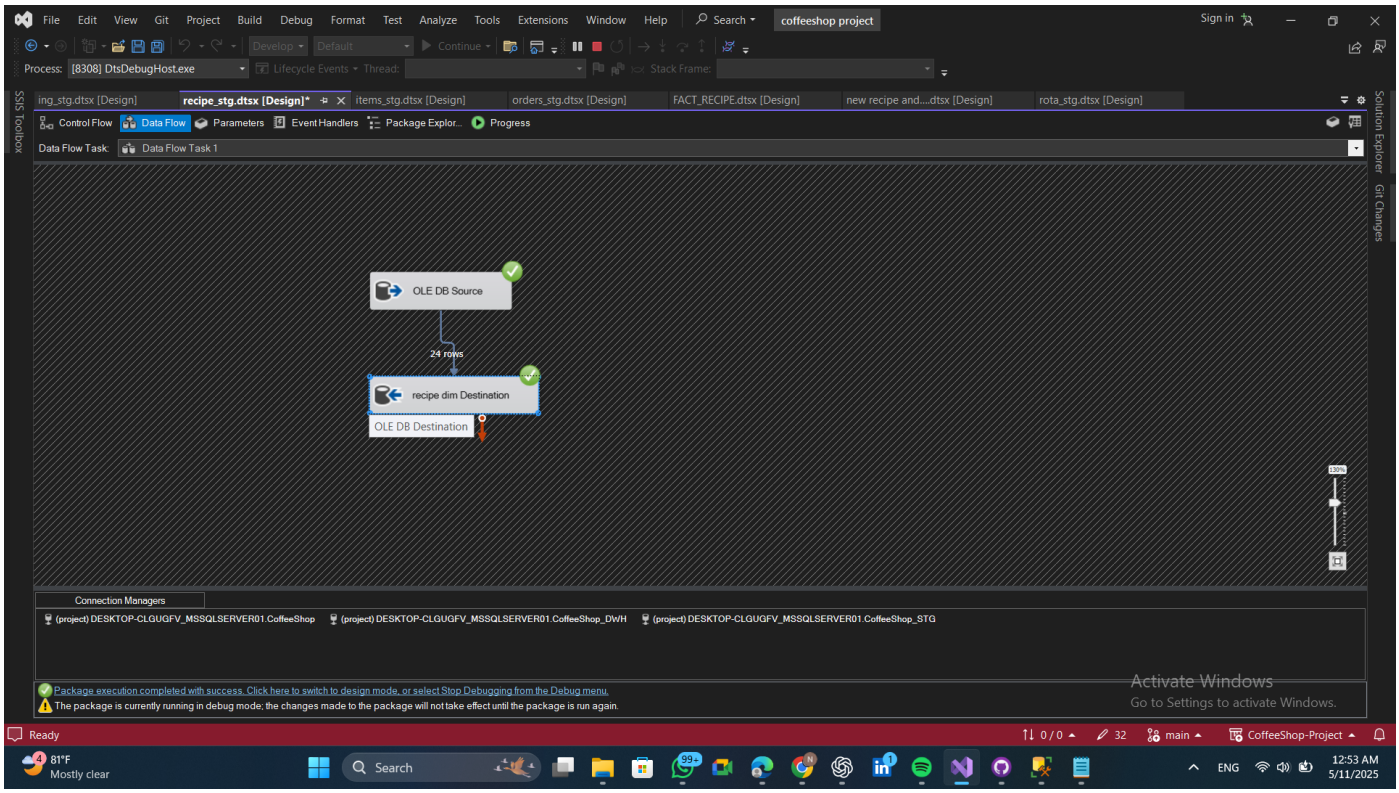


- **Recipe data flow**

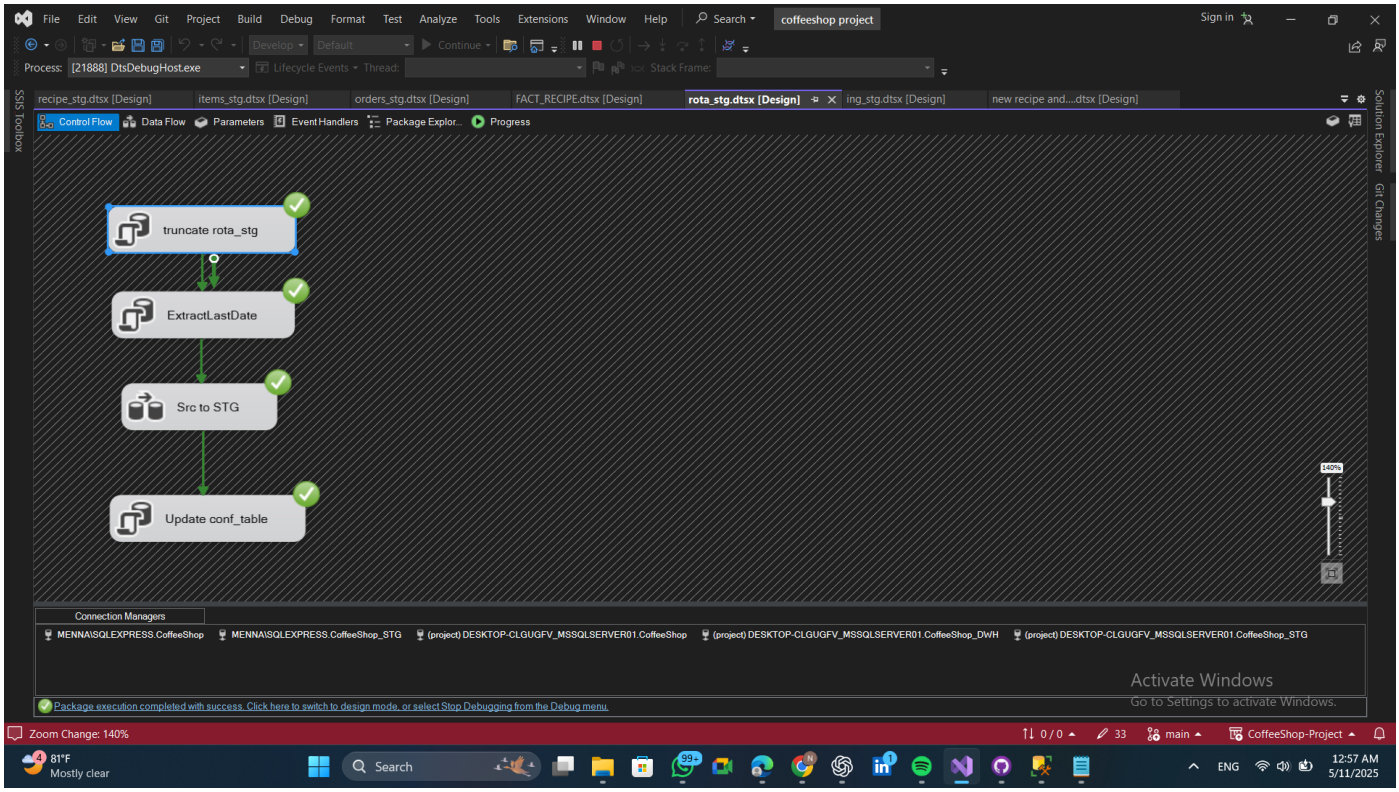
  ➔ Extract data from the source and load it into the staging area. (collect ingrediennds and quantities for the recipe in 1 record)



➔ Load data from the staging area into the dimension table.

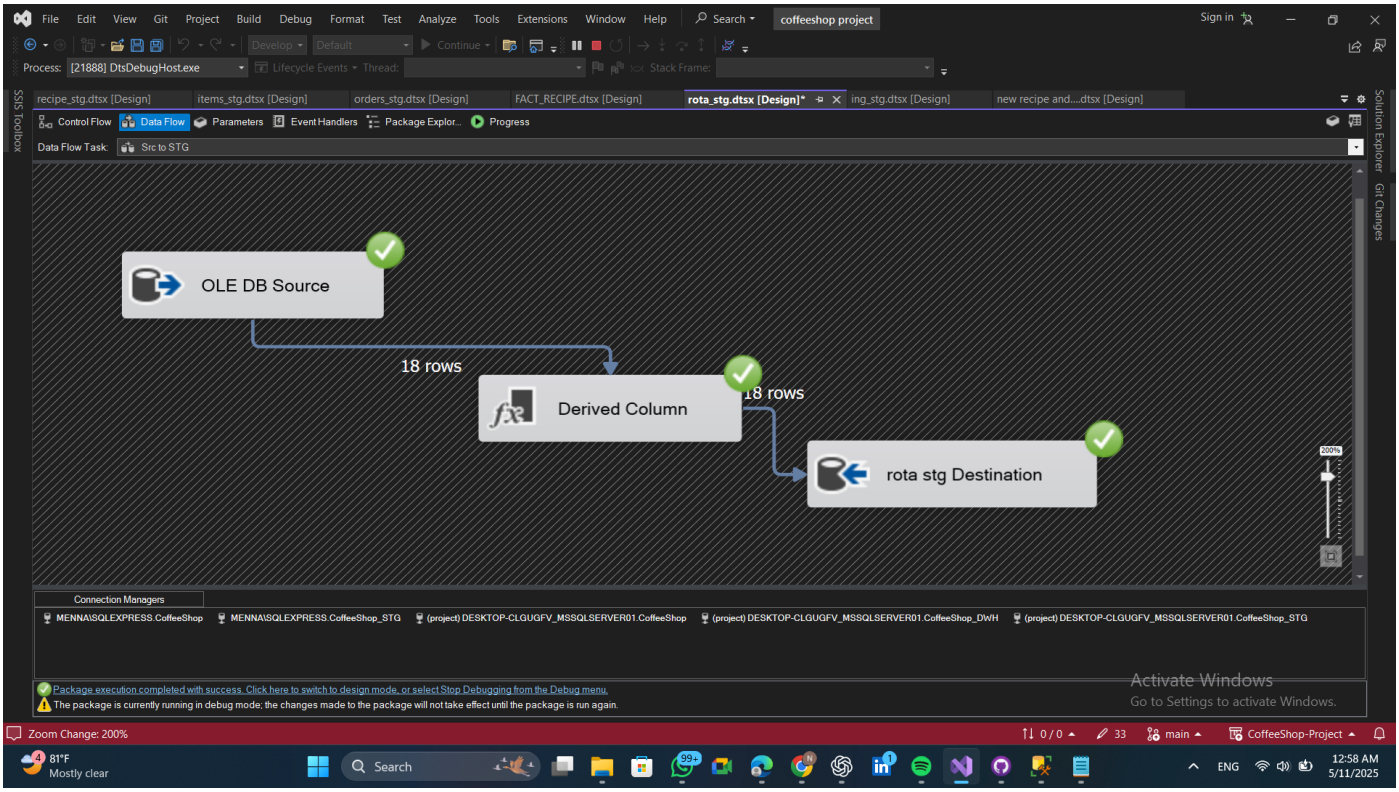


- **Rota control flow** → Delete all data from the staging table to prepare for new inserts or updates, and update the last extract date in the configuration table.

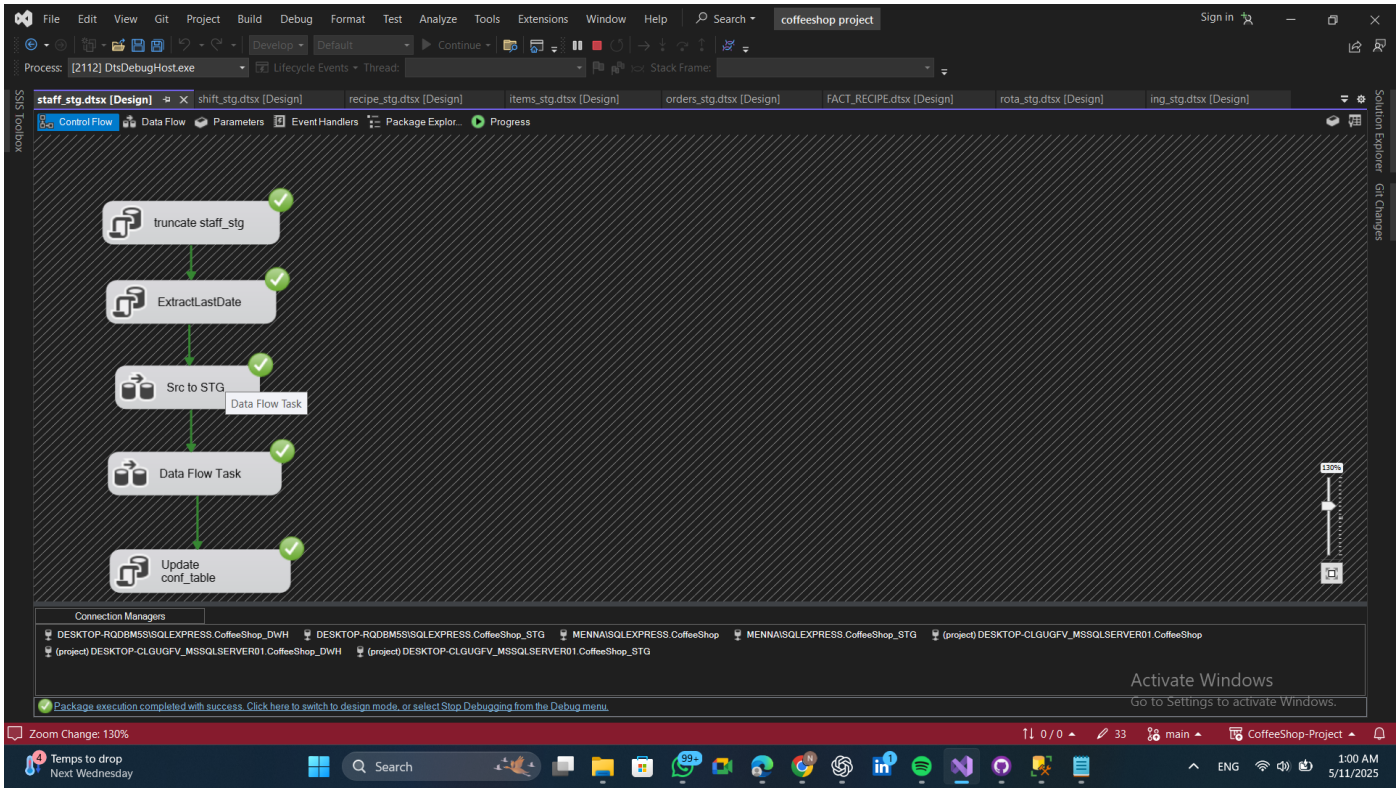


- **Rota data flow**

  ➔ Extract data from the source and load it into the staging area.

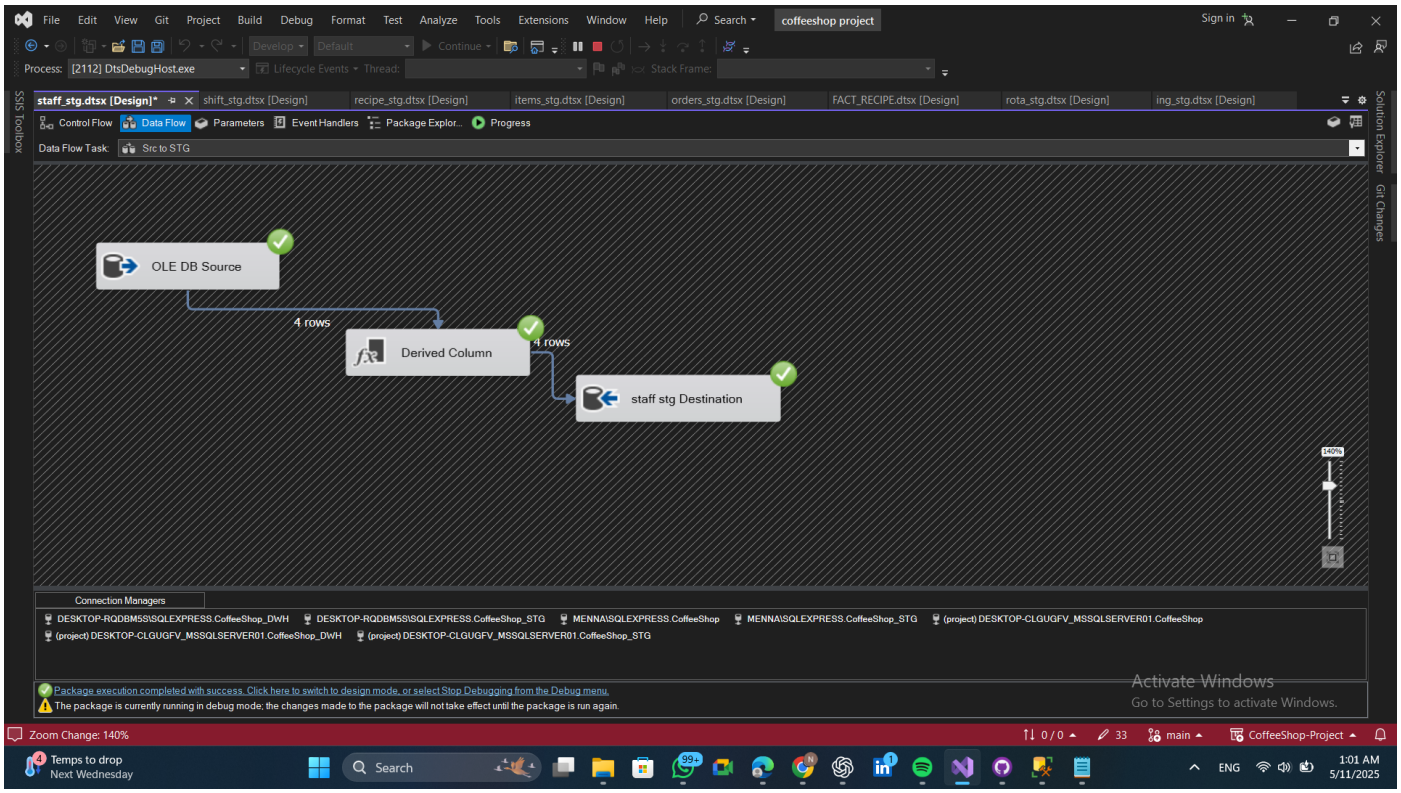


- **Staff control flow** → Delete all data from the staging table to prepare for new inserts or updates, and update the last extract date in the configuration table.

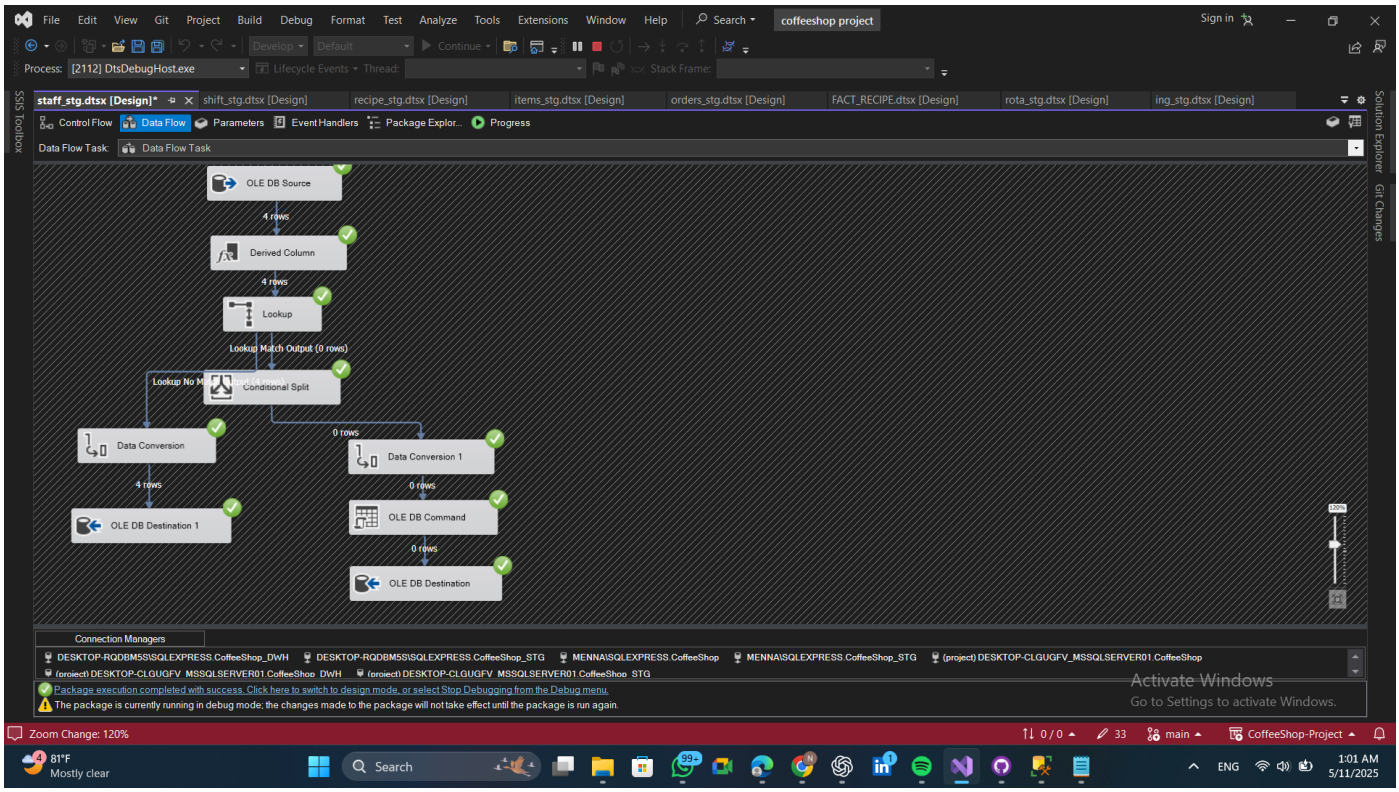


- **Staff data flow**

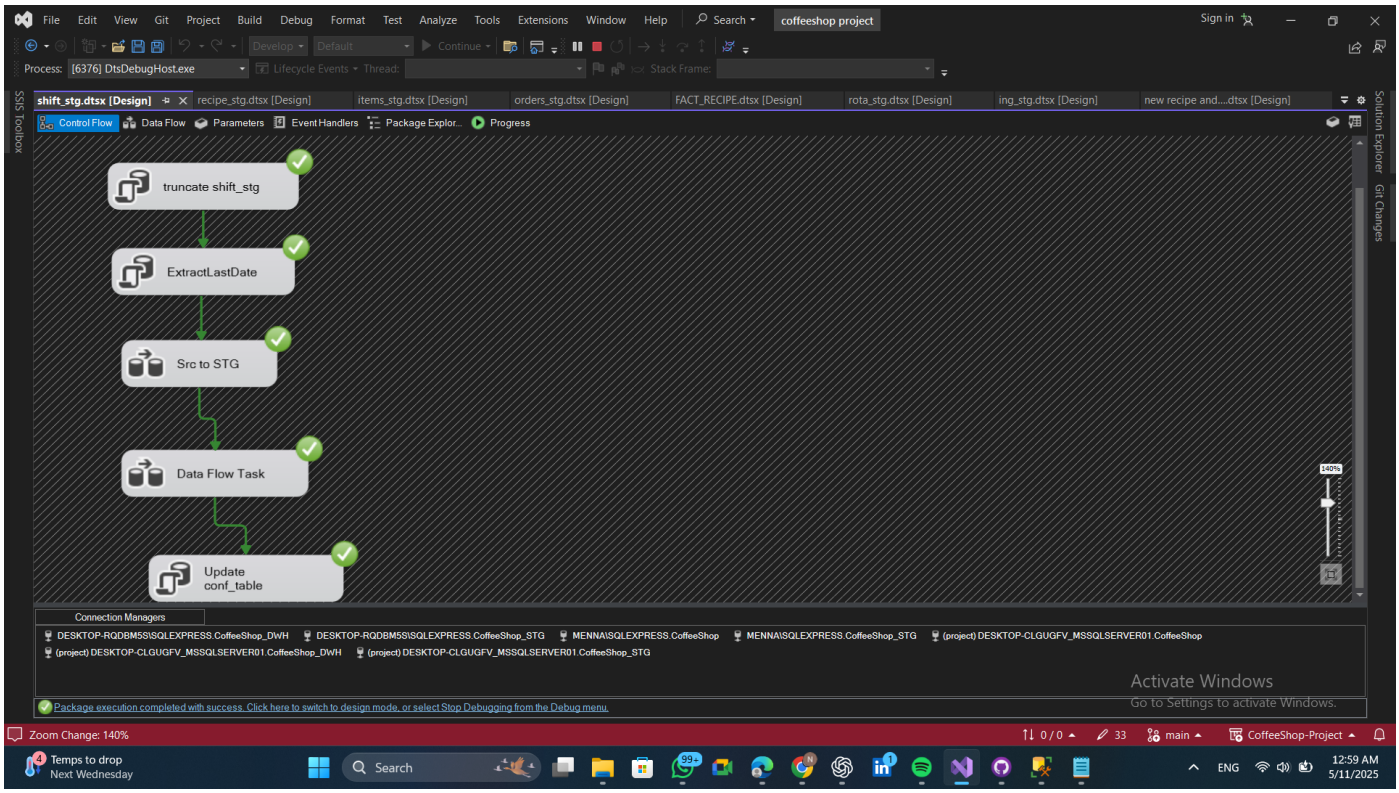
➔ Extract data from the source and load it into the staging area.



➔ Retrieve data from the staging area, perform a lookup to compare new and old staff members, and then load the data into the dimension table

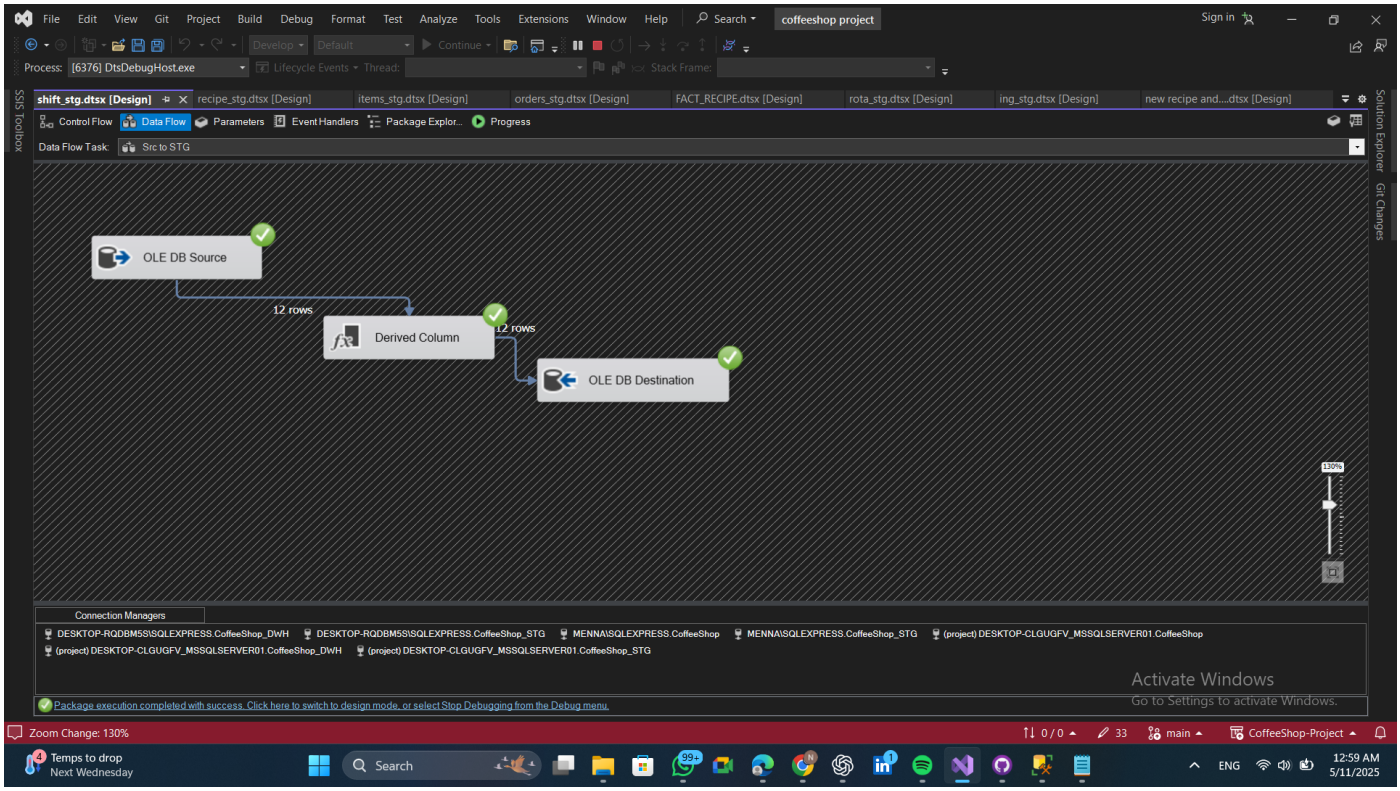


- **Shift control flow** → Delete all data from the staging table to prepare for new inserts or updates, and update the last extract date in the configuration table.

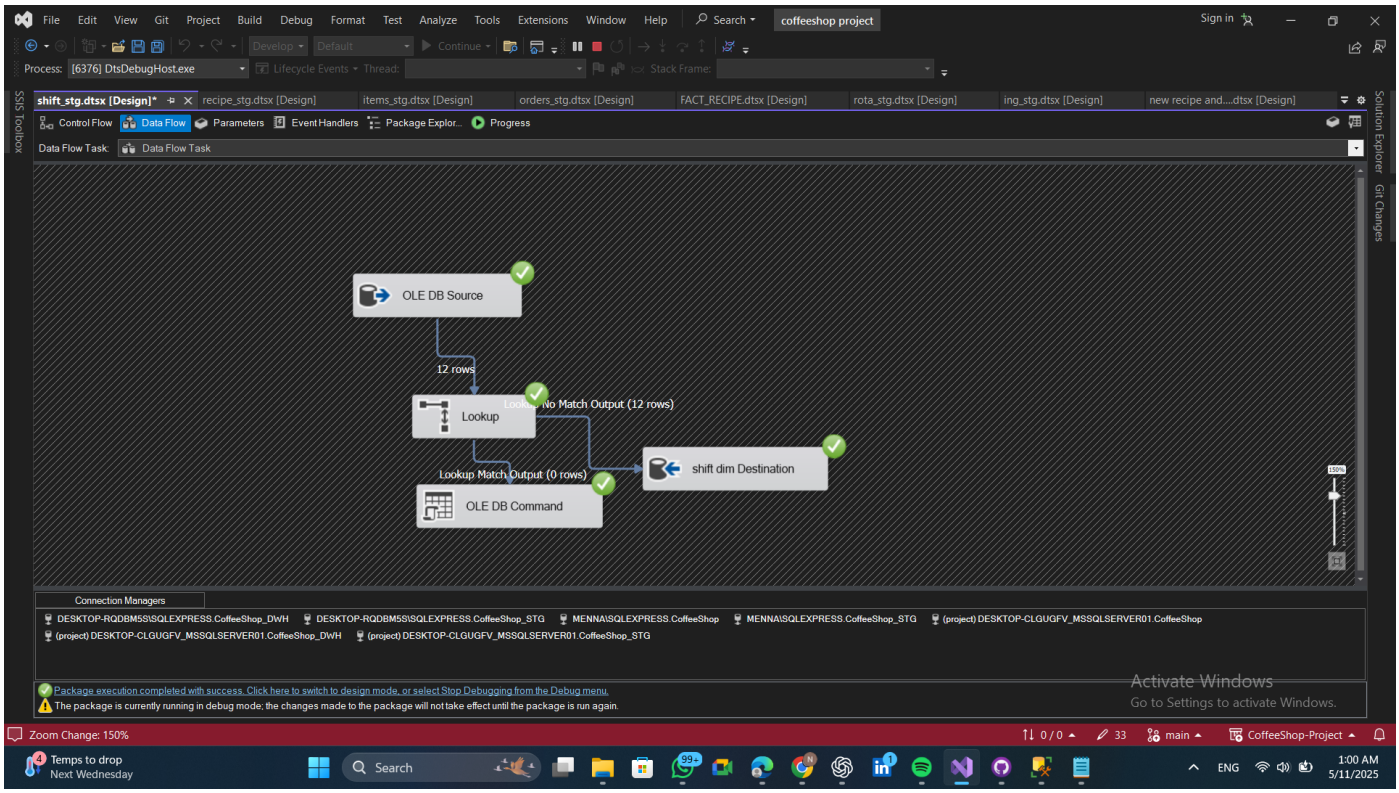


- **Shift data flow**

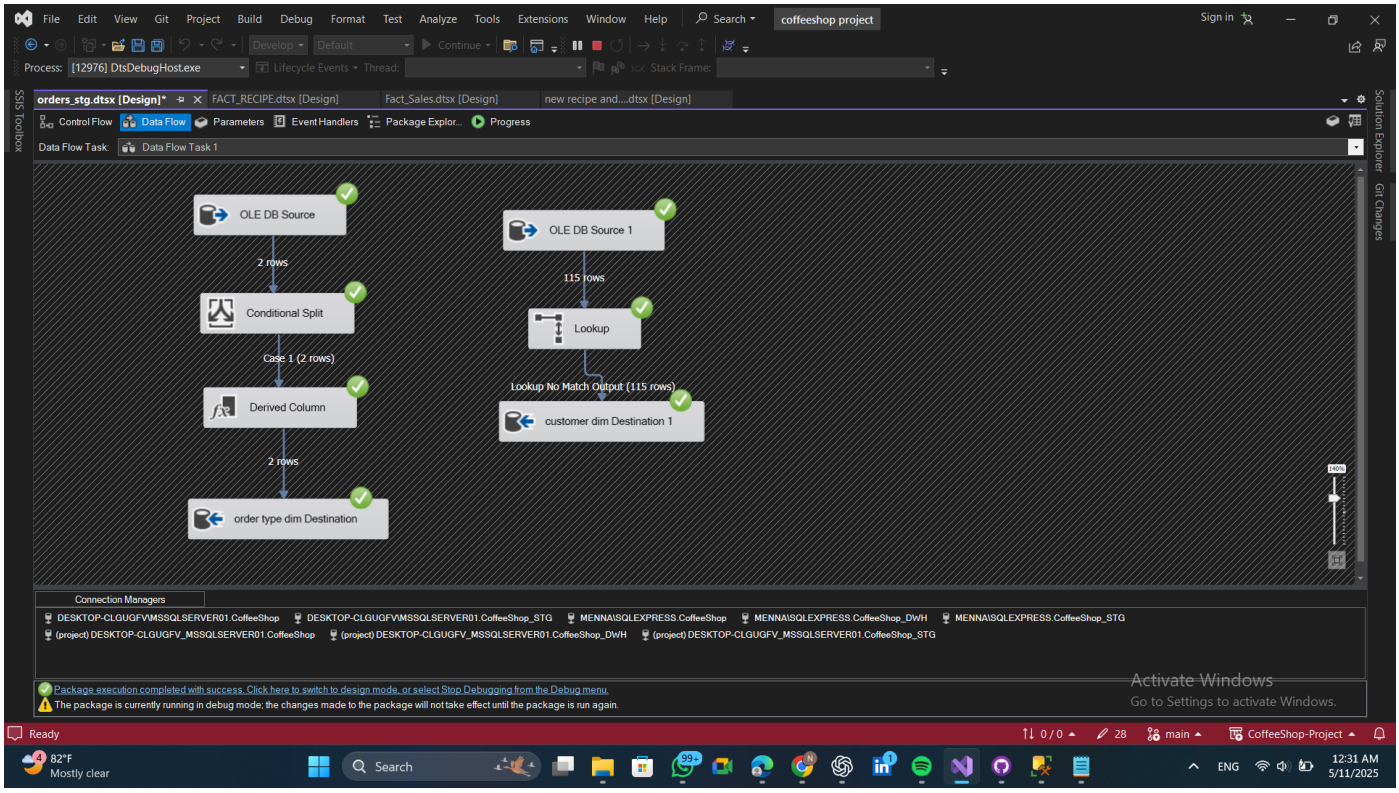
  ➔ Extract data from the source and load it into the staging area.



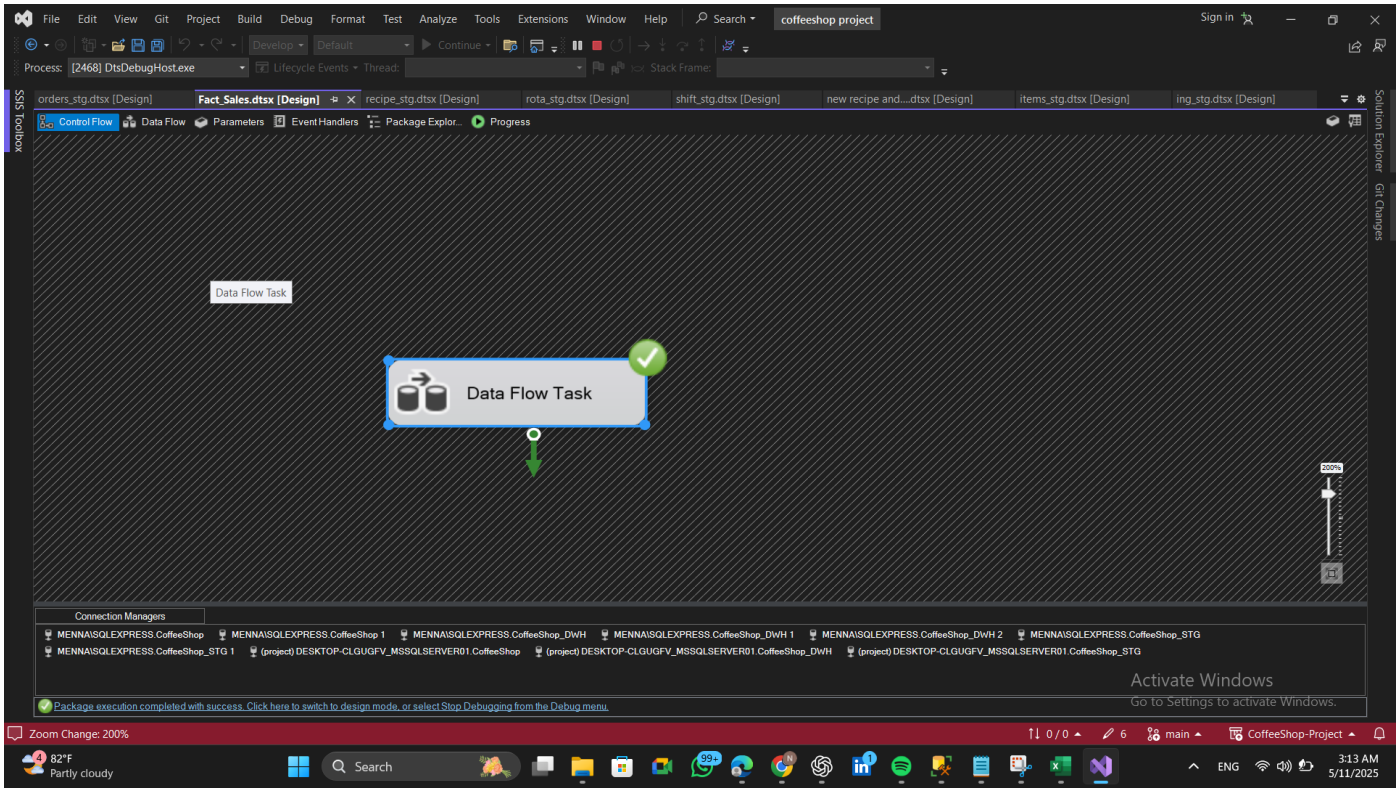
➔ Load data from the staging area into the dimension table.



- **Customer Dim and Order type data flow ➔ Retrieved from staging area of order**

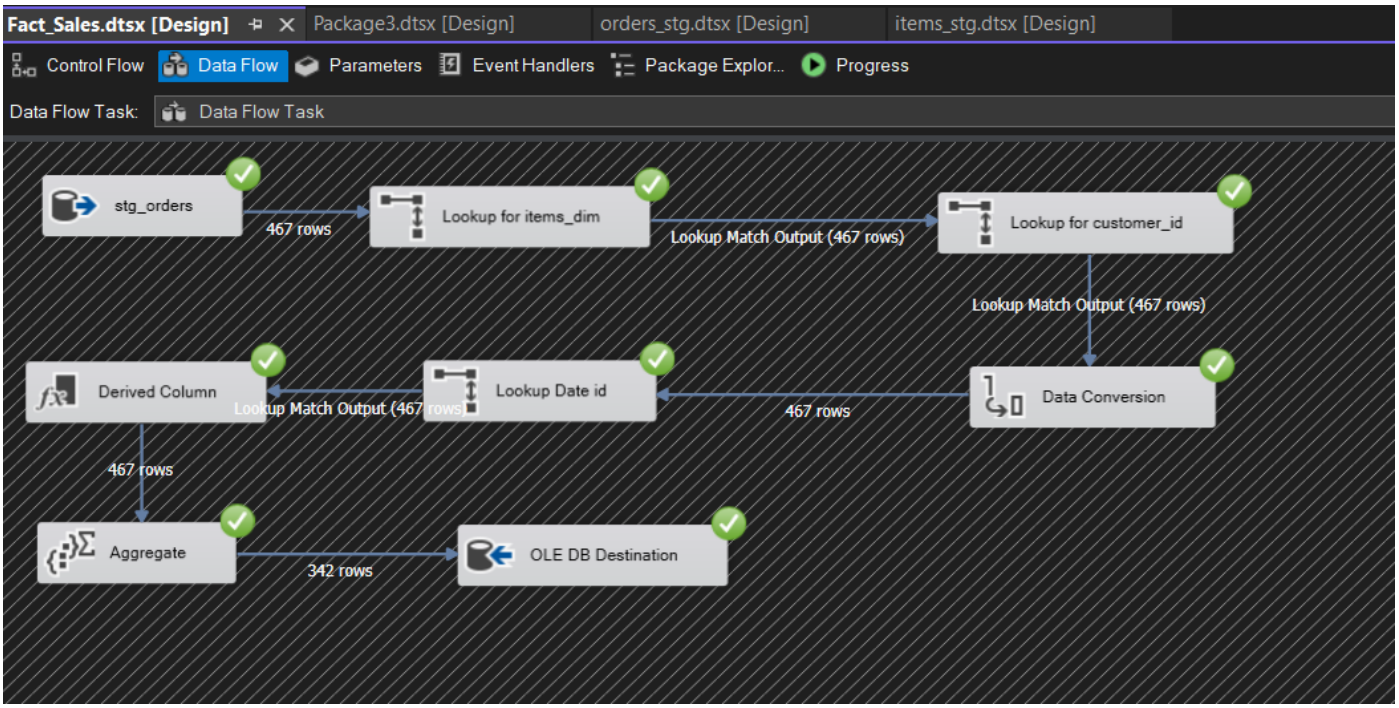


- **Fact Sales control flow**

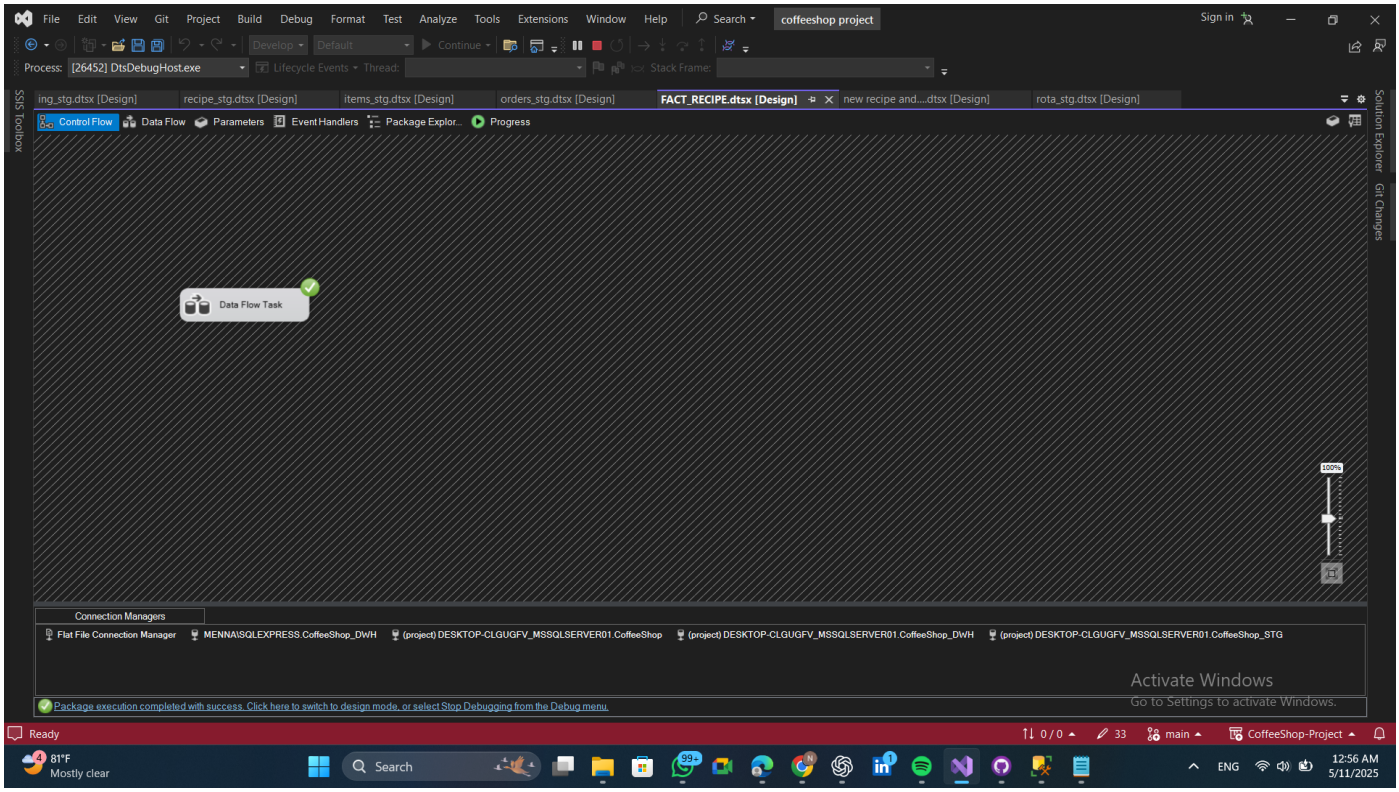


- **Fact Sales data flow**

➔ Load sales data by performing lookups with item, customer, and date dimensions, then aggregate and insert the data into the fact table.

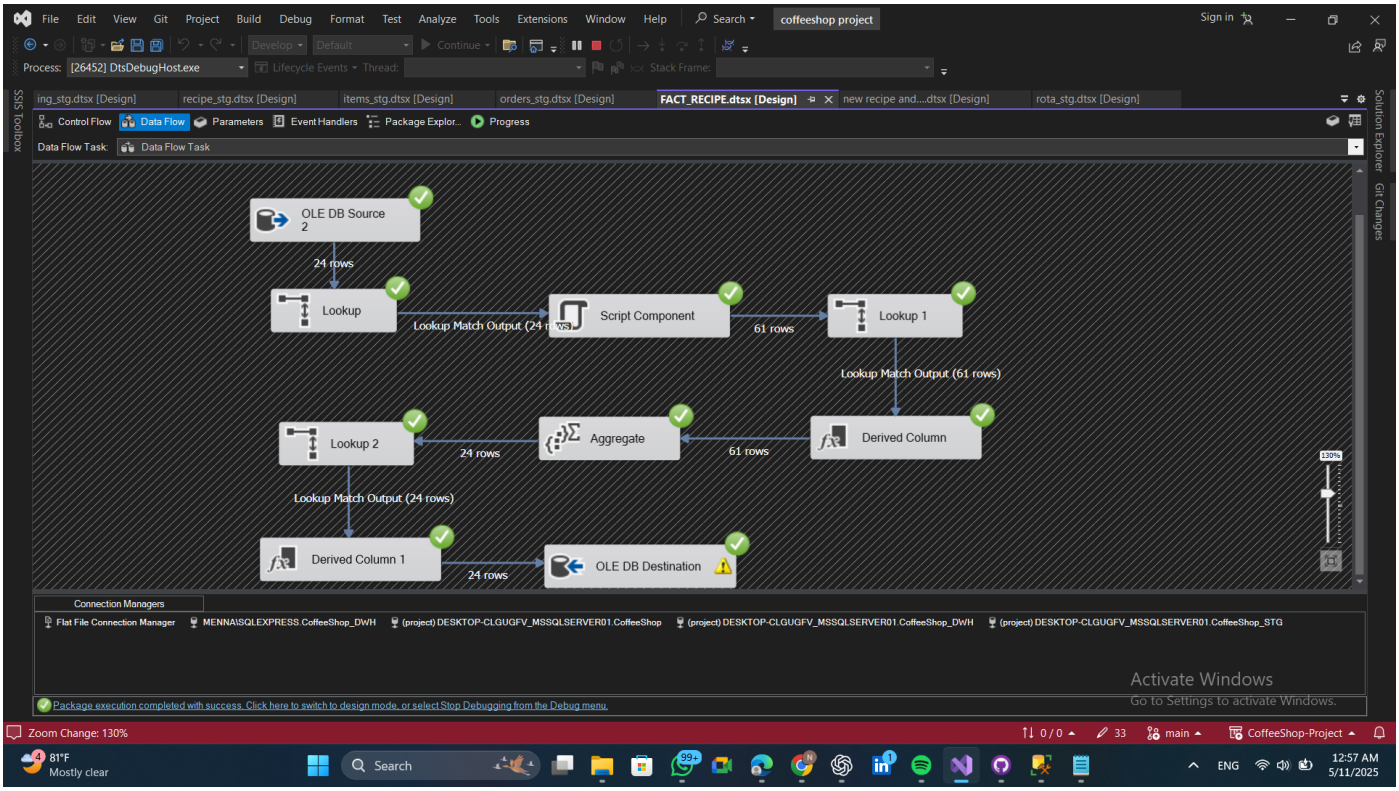


- **Fact Recipe control flow**

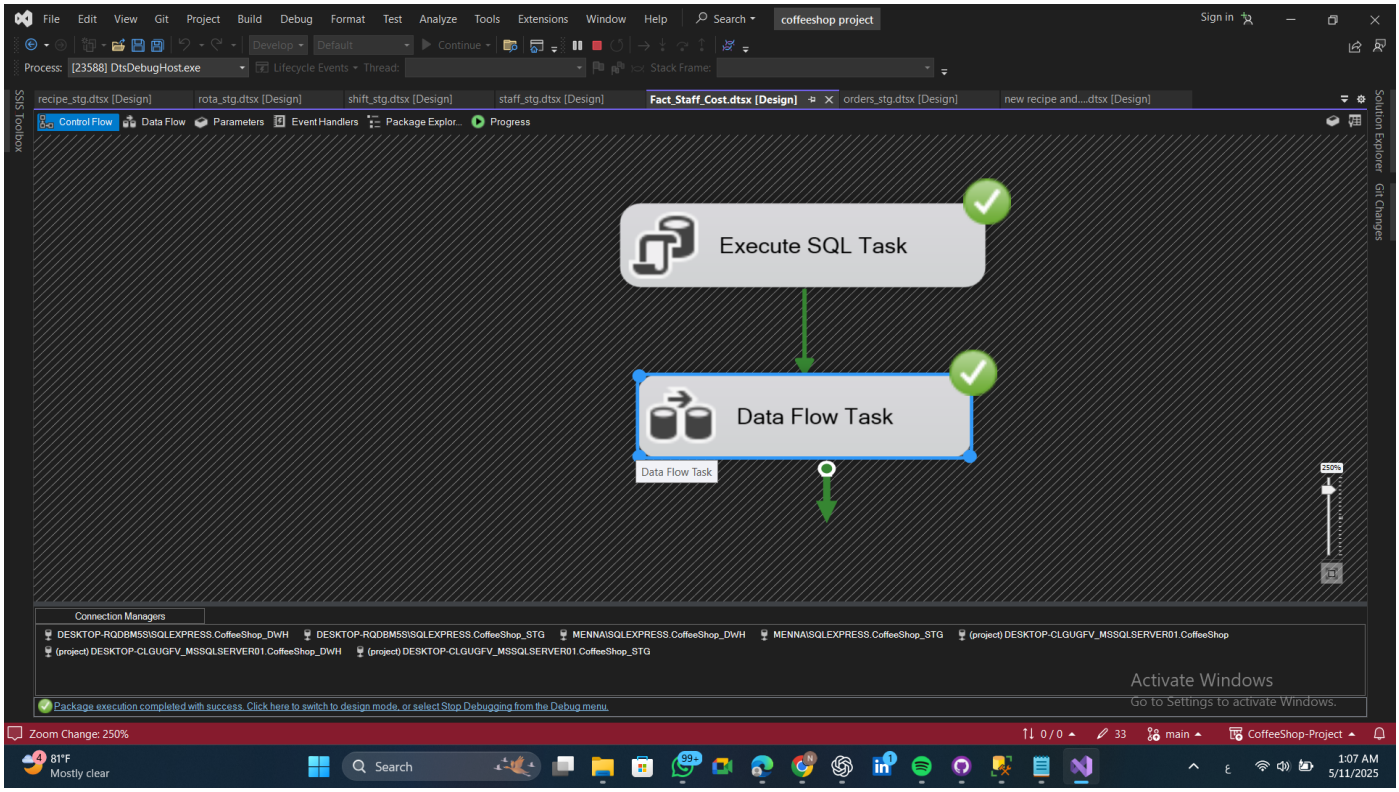


- **Fact Recipe data flow**

➔ Retrieve data from the recipe dimension, add a script component to fetch the ingredients, calculate the cost and profit (item price - cost), and then process the results

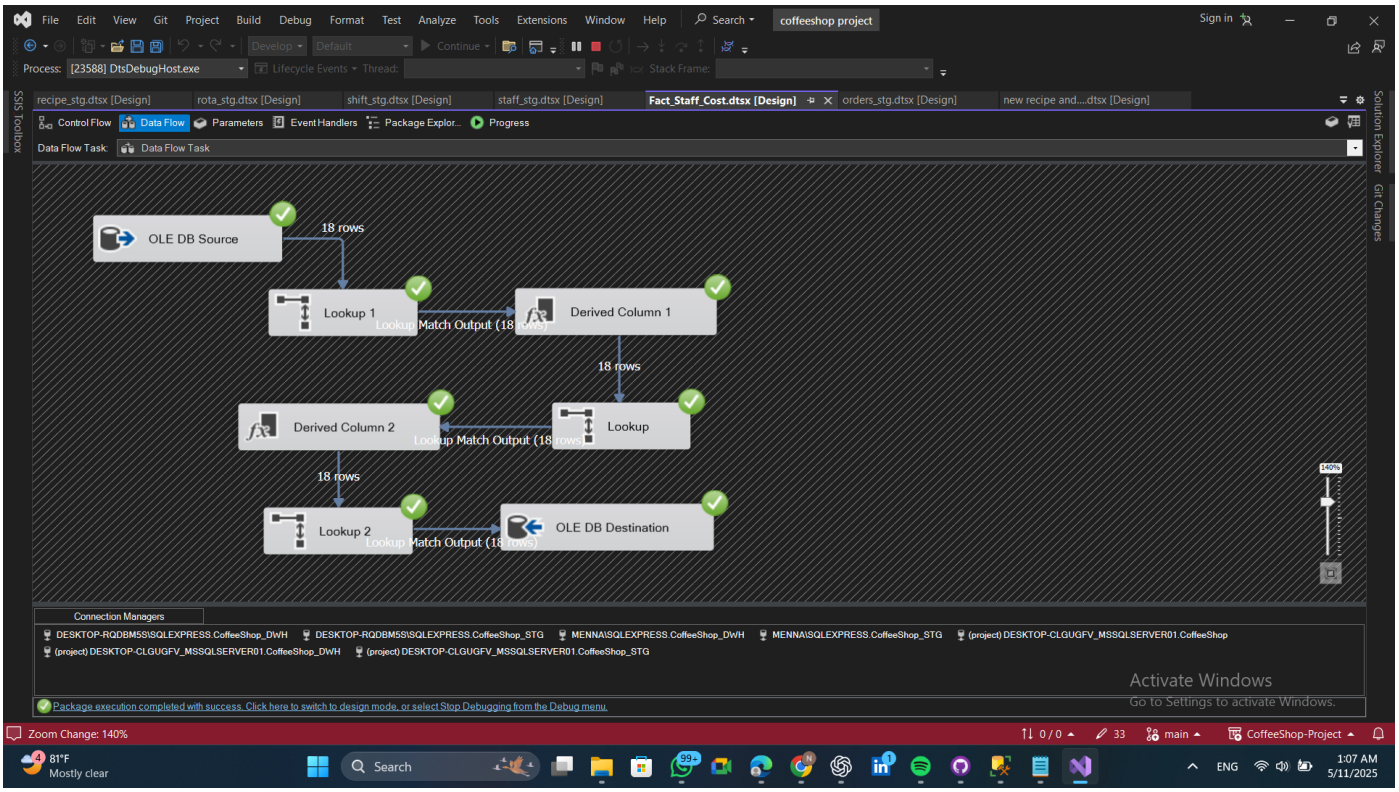


- **Fact Staff Cost control flow**

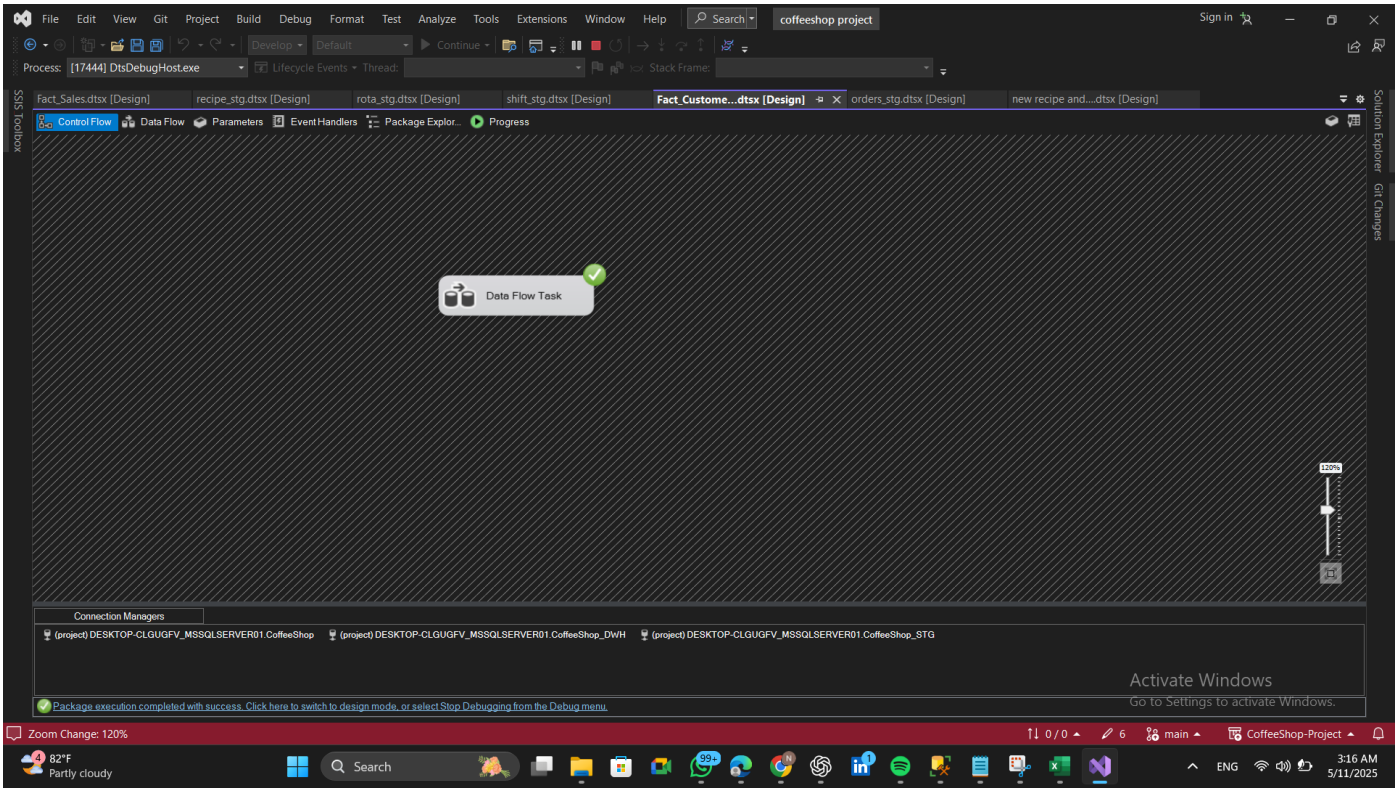


- **Fact Staff Cost data flow**

➔ Load staff cost data, add role and shift information through lookups, transform the data, and then save it to the destination

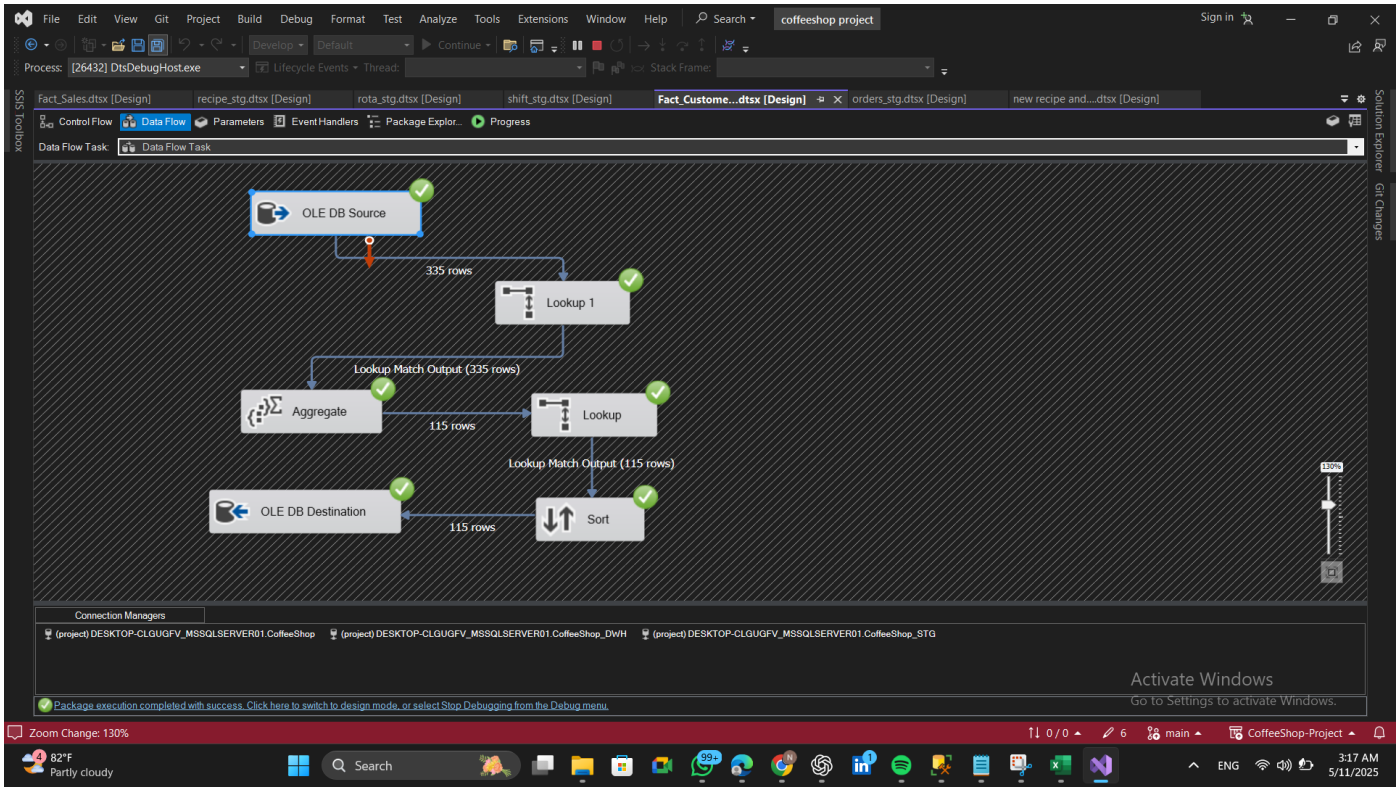


- **Fact Customer Visits Monthly control flow**



- **Fact Customer Visits Monthly data flow**

➔ Retrieve data from the fact sales, perform a lookup on the month name, aggregate by customer, month, and year, then lookup the customer name to retrieve all related data



7. Analytical Queries

```sql
-- Count of All Orders
SELECT COUNT(*) AS Total_Orders FROM Fact_Sales;

-- Sales by Order Type
```

190 %

Results | Messages

	Total_Orders
1	342

```sql
-------------------------------------
-----FACT_CUSTOMER_VISITS_MONTHLY---
-- Total Visits
SELECT SUM(visit_count) AS Total_Visits FROM Fact_Customer_Visits_Monthly;
```

190 %

Results | Messages

	Total_Visits
1	342

```sql
-- Most Expensive Recipe (By Cost)
SELECT TOP 1 recipe_id, total_cost
FROM Fact_Recipe
ORDER BY total_cost DESC;
```

100 %

Results | Messages

	recipe_id	total_cost
1	HDR-HCH-LG	3.06

```sql
-- Staff with Highest Cost in a Single Month
SELECT TOP 1 s.staff_id, d.MonthName, d.Year, SUM(fsc.cost) AS Monthly_Cost
FROM Fact_Staff_Cost fsc
JOIN Staff_Dim s ON fsc.staff_id = s.staff_id
JOIN Dim_Date d ON fsc.date_id = d.Date_ID
GROUP BY s.staff_id, s.staff_name, d.MonthName, d.Year
ORDER BY Monthly_Cost DESC;
```

100 %

Results | Messages

	staff_id	MonthName	Year	Monthly_Cost
1	ST002	February	2024	260.00

```sql
-----------------------------------
------------ FACT_SALES -------------
-- Total Sales Amount
SELECT SUM(sales_amount) AS Total_Sales FROM Fact_Sales;
```

190 %

Results | Messages

	Total_Sales
1	1864.05

```sql
-- Staff Efficiency: Cost per Hour
SELECT staff_id, SUM(cost)/NULLIF(SUM(worked_hours), 0) AS Cost_Per_Hour
FROM Fact_Staff_Cost
GROUP BY staff_id;
```

100 %

Results | Messages

	staff_id	Cost_Per_Hour
1	ST001	10.000000
2	ST002	10.000000
3	ST003	10.000000
4	ST004	10.000000

```sql
-- Average Monthly Visit Count Per Customer
SELECT customer_id, cust_name,
       AVG(visit_count) AS Avg_Monthly_Visits
FROM Fact_Customer_Visits_Monthly
GROUP BY customer_id, cust_name;
```

190 %

Results | Messages

	customer_id	cust_name	Avg_Monthly_Visits
1	1	Alex	19
2	2	Alice	1
3	3	Amelia	2
4	4	Amy	3
5	5	Anna	2
6	6	Avery	3
7	7	Blake	1
8	8	Bob	2
9	9	Brad	3
10	10	Brooke	12
11	11	Cameron	11
12	12	Cara	1
13	13	Carol	1
14	14	Casey	16
15	15	Charles	1
16	16	Charlie	2
17	17	Chloe	1
18	18	Chris	10
19	19	Cindy	2
20	20	Dakota	1

Query executed successfully. MENNA\SQLEXPRESS

```sql
-- Total Cost of Ingredients Used
SELECT SUM(total_cost) AS Total_Recipe_Cost FROM Fact_Recipe;
```

100 %

Results | Messages

	Total_Recipe_Cost
1	33.99

```sql
-- Profit Margin per Item
SELECT item_id,
       SUM(profit) / NULLIF(SUM(item_price * 1.0), 0) AS Profit_Margin
FROM Fact_Recipe
GROUP BY item_id;

-------------------------------------
```

100 %

Results | Messages

	item_id	Profit_Margin
1	It001	0.614492
2	It002	0.512000
3	It003	0.614492
4	It004	0.512000
5	It005	0.482539
6	It006	0.666666
7	It007	0.589130
8	It008	0.986046
9	It009	0.650000
10	It010	0.589130
11	It011	0.688888
12	It012	0.597872
13	It013	0.476190
14	It014	0.332608
15	It015	0.614492
16	It016	0.512000
17	It017	0.650000
18	It018	0.589130
19	It019	0.941538
20	It020	0.923943
21	It021	0.895522
22	It022	0.904000
23	It023	0.621428
24	It024	0.760000

```sql
-- Profit by Item
SELECT item_id, SUM(profit) AS Profit_Per_Item
FROM Fact_Recipe
GROUP BY item_id;
```

100 %

Results | Messages

	item_id	Profit_Per_Item
1	It001	2.12
2	It002	1.92
3	It003	2.12
4	It004	1.92
5	It005	1.52
6	It006	2.80
7	It007	2.71
8	It008	2.12
9	It009	2.60
10	It010	2.71
11	It011	3.10
12	It012	2.81
13	It013	2.00
14	It014	1.53
15	It015	2.12
16	It016	1.92
17	It017	2.60
18	It018	2.71
19	It019	3.06
20	It020	3.28
21	It021	3.00
22	It022	3.39
23	It023	3.48
24	It024	4.18

```sql
-- Top 5 Customers by Spending
SELECT customer_id, cust_name,
       SUM(total_spent) AS Total_Spent
FROM Fact_Customer_Visits_Monthly
GROUP BY customer_id, cust_name
ORDER BY Total_Spent DESC
OFFSET 0 ROWS FETCH NEXT 5 ROWS ONLY;
```

190 %

Results | Messages

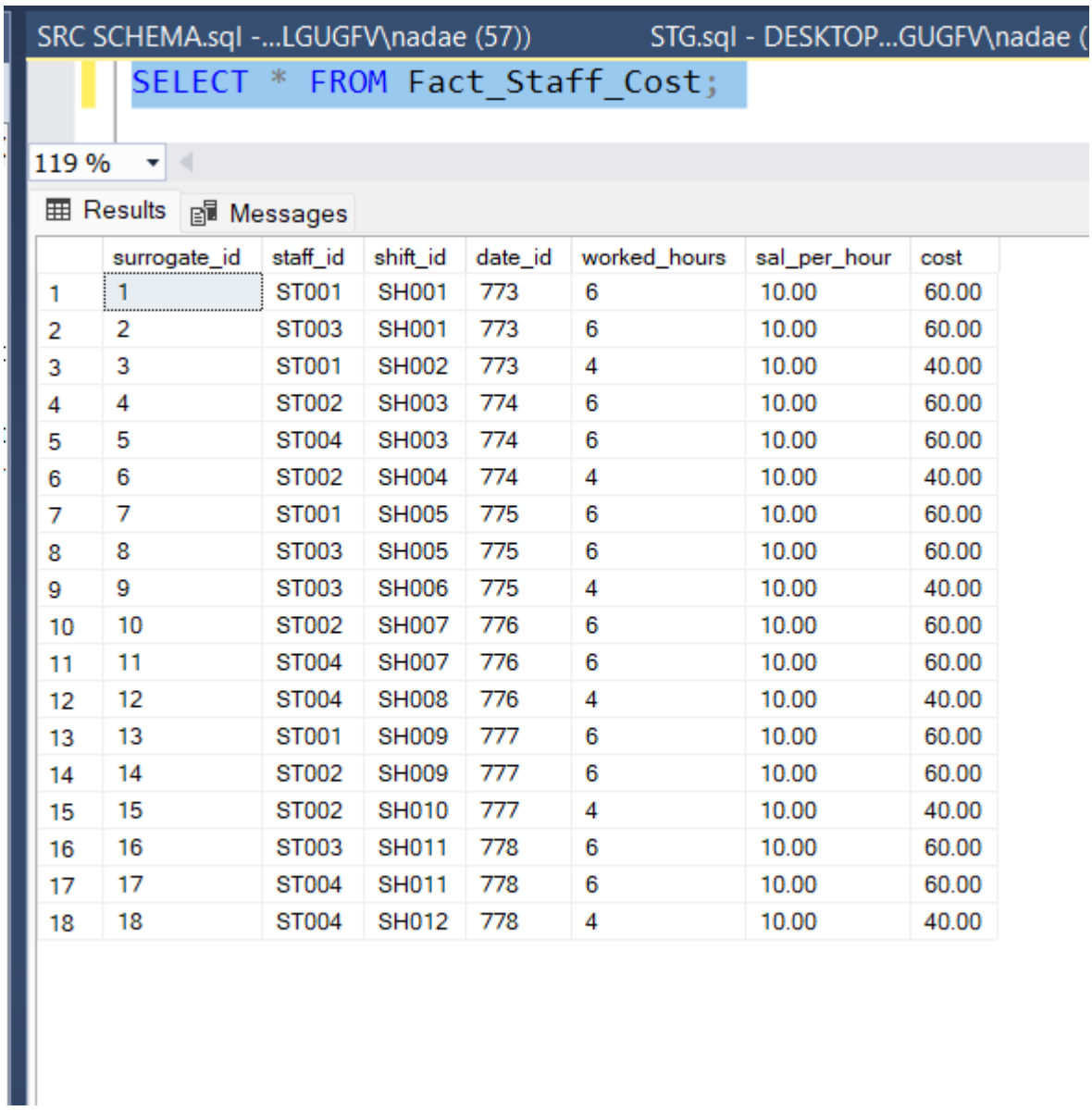
	customer_id	cust_name	Total_Spent
1	94	Riley	94.45
2	1	Alex	89.60
3	91	Quinn	88.45
4	98	Taylor	81.95
5	58	Jordan	79.25

```sql
-------------------------------------------------------
-------MULTIPLE TABLES
-- Profit Per Staff Hour (Sales / Worked Hours)
SELECT fsc.staff_id,
       ROUND(SUM(fs.sales_amount) / NULLIF(SUM(fsc.worked_hours), 0), 2) AS Sales_Per_Hour
FROM Fact_Staff_Cost fsc
JOIN Fact_Sales fs ON fsc.date_id = fs.date_id
GROUP BY fsc.staff_id;
```

100 %

Results | Messages

	staff_id	Sales_Per_Hour
1	ST003	0.850000
2	ST002	0.750000
3	ST001	0.740000
4	ST004	1.000000



	surrogate_id	staff_id	shift_id	date_id	worked_hours	sal_per_hour	cost
1	1	ST001	SH001	773	6	10.00	60.00
2	2	ST003	SH001	773	6	10.00	60.00
3	3	ST001	SH002	773	4	10.00	40.00
4	4	ST002	SH003	774	6	10.00	60.00
5	5	ST004	SH003	774	6	10.00	60.00
6	6	ST002	SH004	774	4	10.00	40.00
7	7	ST001	SH005	775	6	10.00	60.00
8	8	ST003	SH005	775	6	10.00	60.00
9	9	ST003	SH006	775	4	10.00	40.00
10	10	ST002	SH007	776	6	10.00	60.00
11	11	ST004	SH007	776	6	10.00	60.00
12	12	ST004	SH008	776	4	10.00	40.00
13	13	ST001	SH009	777	6	10.00	60.00
14	14	ST002	SH009	777	6	10.00	60.00
15	15	ST002	SH010	777	4	10.00	40.00
16	16	ST003	SH011	778	6	10.00	60.00
17	17	ST004	SH011	778	6	10.00	60.00
18	18	ST004	SH012	778	4	10.00	40.00

```
SELECT * FROM Fact_Recipe;
```

119 %

Results | Messages

	recipe_id	item_id	item_price	total_cost	profit
1	HDR-CRM-MD	It006	4.20	1.39	2.80
2	HDR-CRM-LG	It007	4.60	1.88	2.71
3	HDR-MOC-MD	It009	4.00	1.39	2.60
4	HDR-MOC-LG	It010	4.60	1.88	2.71
5	HDR-FLT	It005	3.15	1.62	1.52
6	HDR-LAT-MD	It003	3.45	1.32	2.12
7	HDR-LAT-LG	It004	3.75	1.83	1.92
8	CDR-LMN-MD	It021	3.35	0.34	3.00
9	CDR-LMN-LG	It022	3.75	0.36	3.39
10	CDR-ICT-MD	It019	3.25	0.19	3.06
11	CDR-ICT-LG	It020	3.55	0.26	3.28
12	SNK-SHC	It023	5.60	2.11	3.48
13	HDR-HCH-MD	It013	4.20	2.19	2.00
14	HDR-HCH-LG	It014	4.60	3.06	1.53
15	CDR-CMO-MD	It017	4.00	1.39	2.60
16	CDR-CMO-LG	It018	4.60	1.88	2.71
17	HDR-ESP	It008	2.15	0.02	2.12
18	HDR-CAP-MD	It001	3.45	1.32	2.12
19	HDR-CAP-LG	It002	3.75	1.83	1.92
20	HDR-WMO-...	It011	4.50	1.39	3.10
21	HDR-WMO-LG	It012	4.70	1.88	2.81
22	CDR-CCF-MD	It015	3.45	1.32	2.12
23	CDR-CCF-LG	It016	3.75	1.83	1.92
24	SNK-SSM	It024	5.50	1.31	4.18

```sql
-- Loyalty Score: Total Visits * Avg Spend
SELECT customer_id, cust_name,
       SUM(visit_count) * AVG(total_spent / NULLIF(visit_count, 0)) AS Loyalty_Score
FROM Fact_Customer_Visits_Monthly
GROUP BY customer_id, cust_name
ORDER BY Loyalty_Score DESC;
```

173 %

Results | Messages

	customer_id	cust_name	Loyalty_Score
1	94	Riley	94.450000
2	1	Alex	89.600000
3	91	Quinn	88.450000
4	98	Taylor	81.950000
5	58	Jordan	79.250000
6	54	Jamie	76.750000
7	76	Morgan	73.300000
8	14	Casey	71.000000
9	89	Peyton	59.800000
10	11	Cameron	57.500000
11	10	Brooke	55.700000
12	95	Sam	50.150000
13	18	Chris	46.950000
14	108	Wendy	31.150000
15	50	Iris	27.400000
16	52	Jack	23.400000
17	23	Derek	22.450000
18	44	Harper	22.050000
19	110	Xavier	21.850000
20	86	Pat	21.750000

```sql
-- Total Hours Worked Per Staff
SELECT staff_id, SUM(worked_hours) AS Total_Hours
FROM Fact_Staff_Cost
GROUP BY staff_id;
```

100 %

Results | Messages

	staff_id	Total_Hours
1	ST001	22
2	ST002	26
3	ST003	22
4	ST004	26

```sql
-- Average Salary Per Hour Across All Staff
SELECT AVG(sal_per_hour) AS Avg_Salary_Per_Hour FROM Fact_Staff_Cost;
```

100 %

Results | Messages

	Avg_Salary_Per_Hour
1	10.000000

```sql
-- Average Sales Amount Per Order
SELECT AVG(sales_amount) AS Avg_Sales_Amount FROM Fact_Sales;
```

190 %

Results | Messages

	Avg_Sales_Amount
1	5.450438

```sql
-- Staff Cost Per Month
SELECT d.MonthName, d.Year, SUM(fsc.cost) AS Monthly_Staff_Cost
FROM Fact_Staff_Cost fsc
JOIN Dim_Date d ON fsc.date_id = d.Date_ID
GROUP BY d.MonthName, d.Year
ORDER BY d.Year, d.MonthName;
```

100 %

Results | Messages

	MonthName	Year	Monthly_Staff_Cost
1	February	2024	960.00

```sql
-- Count of Shifts Worked
SELECT COUNT(*) AS Total_Shifts FROM Fact_Staff_Cost;
```

100 %

Results | Messages

	Total_Shifts
1	18

```sql
-- Average Item Price
SELECT AVG(item_price) AS Avg_Item_Price FROM Fact_Recipe;
```

100 %

Results | Messages

	Avg_Item_Price
1	3.995833

```sql
------------ FACT_RECIPE--------------
-- Total Profit from All Recipes
SELECT SUM(profit) AS Total_Profit FROM Fact_Recipe;
```

100 %

Results | Messages

	Total_Profit
1	61.72

```sql
-----------------------------------------
------------ FACT_STAFF_COST-------------
-- Total Staff Cost
SELECT SUM(cost) AS Total_Staff_Cost FROM Fact_Staff_Cost;
```

100 %

Results | Messages

	Total_Staff_Cost
1	960.00

```sql
-- Visits Per Month
SELECT month_name, year, SUM(visit_count) AS Monthly_Visits
FROM Fact_Customer_Visits_Monthly
GROUP BY month_name, year
ORDER BY year, month_name;
```

190 %

Results | Messages

	month_name	year	Monthly_Visits
1	February	2024	342

```sql
-- Monthly Sales Trend
SELECT d.MonthName, d.Year, SUM(fs.sales_amount) AS Monthly_Sales
FROM Fact_Sales fs
JOIN Dim_Date d ON fs.date_id = d.Date_ID
GROUP BY d.MonthName, d.Year
ORDER BY d.Year, d.MonthName;
```

190 %

Results | Messages

	MonthName	Year	Monthly_Sales
1	February	2024	1864.05

```
SELECT * FROM Fact_Recipe;
SELECT * FROM Fact_Customer_Visits_Monthly;
```

190 %

Results | Messages

	visit_id	customer_id	cust_name	month_name	year	visit_count	total_spent
1	1	61	Keith	February	2024	1	3.15
2	2	87	Paul	February	2024	1	3.15
3	3	36	Georgia	February	2024	1	3.15
4	4	40	Grace	February	2024	1	3.25
5	5	25	Elvis	February	2024	1	3.45
6	6	81	Oli	February	2024	1	3.45
7	7	33	Frank	February	2024	1	3.45
8	8	67	Lana	February	2024	1	3.45
9	9	97	Susie	February	2024	1	3.45
10	10	12	Cara	February	2024	1	3.45
11	11	73	Mason	February	2024	1	3.45
12	12	96	Steve	February	2024	1	3.55
13	13	42	Hannah	February	2024	1	3.75
14	14	64	Kip	February	2024	1	3.75
15	15	26	Emerson	February	2024	1	3.75
16	16	29	Faisal	February	2024	1	3.75
17	17	38	Gigi	February	2024	1	3.75
18	18	88	Paula	February	2024	1	3.75
19	19	30	Finley	February	2024	1	3.75
20	20	43	Haris	February	2024	1	3.75
21	21	102	Ulla	February	2024	1	3.75
22	22	84	Pam	February	2024	1	4.00
23	23	59	Kara	February	2024	1	4.00
24	24	49	Ida	February	2024	1	4.00
25	25	27	Evan	February	2024	1	4.00
26	26	56	Jenna	February	2024	1	4.00
27	27	85	Parker	February	2024	1	4.20
28	28	13	Carol	February	2024	1	4.20

Query executed successfully. MENNA\SQLEXPRESS

```
-- Total Revenue from Visits
SELECT SUM(total_spent) AS Total_Revenue FROM Fact_Customer_Visits_Monthly;
```

190 %

Results | Messages

	Total_Revenue
1	1864.05

```sql
-- Staff Cost vs Worked Hours Correlation
SELECT staff_id,
       SUM(cost) AS Total_Cost,
       SUM(worked_hours) AS Total_Hours,
       ROUND(SUM(cost)*1.0 / NULLIF(SUM(worked_hours), 0), 2) AS Effective_Rate
FROM Fact_Staff_Cost
GROUP BY staff_id
ORDER BY Effective_Rate DESC;
```

100 %

Results | Messages

	staff_id	Total_Cost	Total_Hours	Effective_Rate
1	ST001	220.00	22	10.000000
2	ST002	260.00	26	10.000000
3	ST003	220.00	22	10.000000
4	ST004	260.00	26	10.000000

```sql
-- Visit Frequency & Spending Trend per Year
SELECT year,
       AVG(visit_count) AS Avg_Visits,
       AVG(total_spent) AS Avg_Spent
FROM Fact_Customer_Visits_Monthly
GROUP BY year
ORDER BY year;
```

190 %

Results | Messages

	year	Avg_Visits	Avg_Spent
1	2024	2	16.209130

```sql
-- Top 5 Most Profitable Recipes
SELECT recipe_id, SUM(profit) AS Total_Profit
FROM Fact_Recipe
GROUP BY recipe_id
ORDER BY Total_Profit DESC
OFFSET 0 ROWS FETCH NEXT 5 ROWS ONLY;
```

100 %

Results | Messages

	recipe_id	Total_Profit
1	SNK-SSM	4.18
2	SNK-SHC	3.48
3	CDR-LMN-LG	3.39
4	CDR-ICT-LG	3.28
5	HDR-WMO-MD	3.10

```sql
-- Top 5 Customers by Total Sales
SELECT c.customer_id, c.cust_name, SUM(fs.sales_amount) AS Total_Spent
FROM Fact_Sales fs
JOIN Dim_Customer c ON fs.cust_id = c.customer_id
GROUP BY c.customer_id, c.cust_name
ORDER BY Total_Spent DESC
OFFSET 0 ROWS FETCH NEXT 5 ROWS ONLY;
```

190 %

Results | Messages

	customer_id	cust_name	Total_Spent
1	94	Riley	94.45
2	1	Alex	89.60
3	91	Quinn	88.45
4	98	Taylor	81.95
5	58	Jordan	79.25

```sql
-- Average Spending per Customer
SELECT customer_id, cust_name,
       AVG(total_spent) AS Avg_Spending
FROM Fact_Customer_Visits_Monthly
GROUP BY customer_id, cust_name;
```

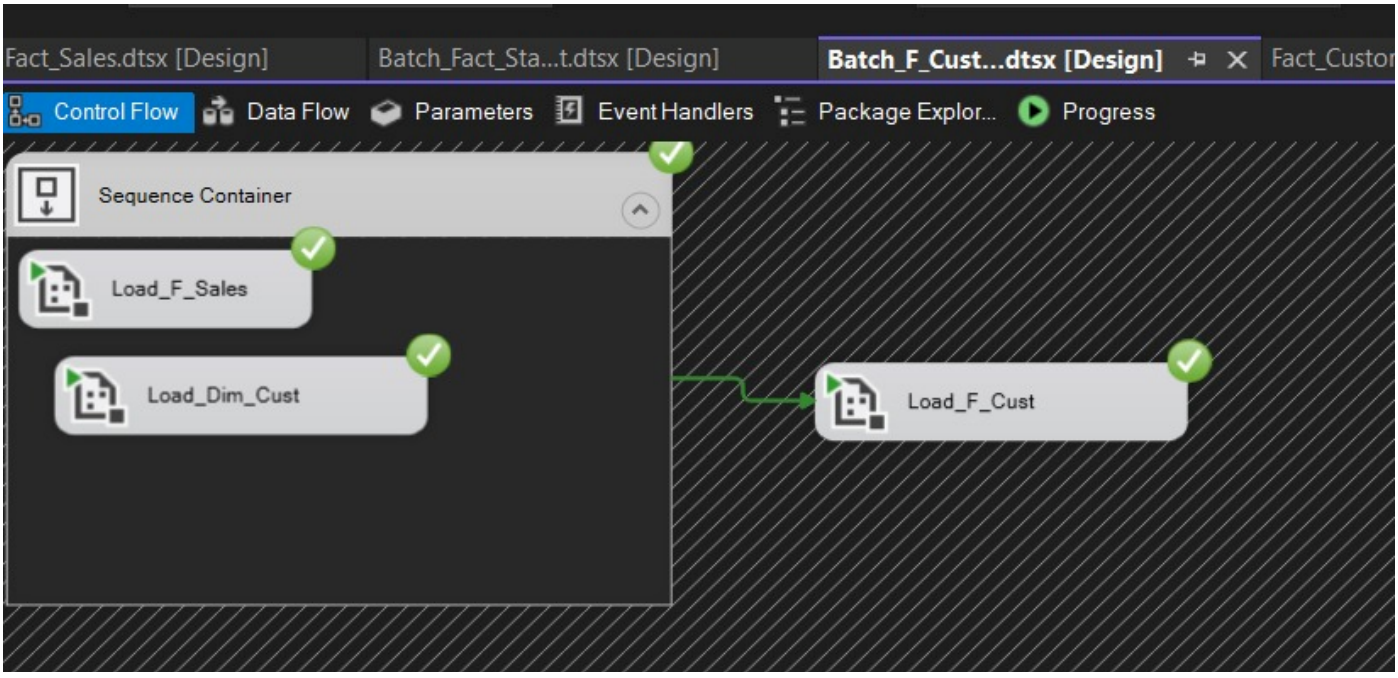
190 %

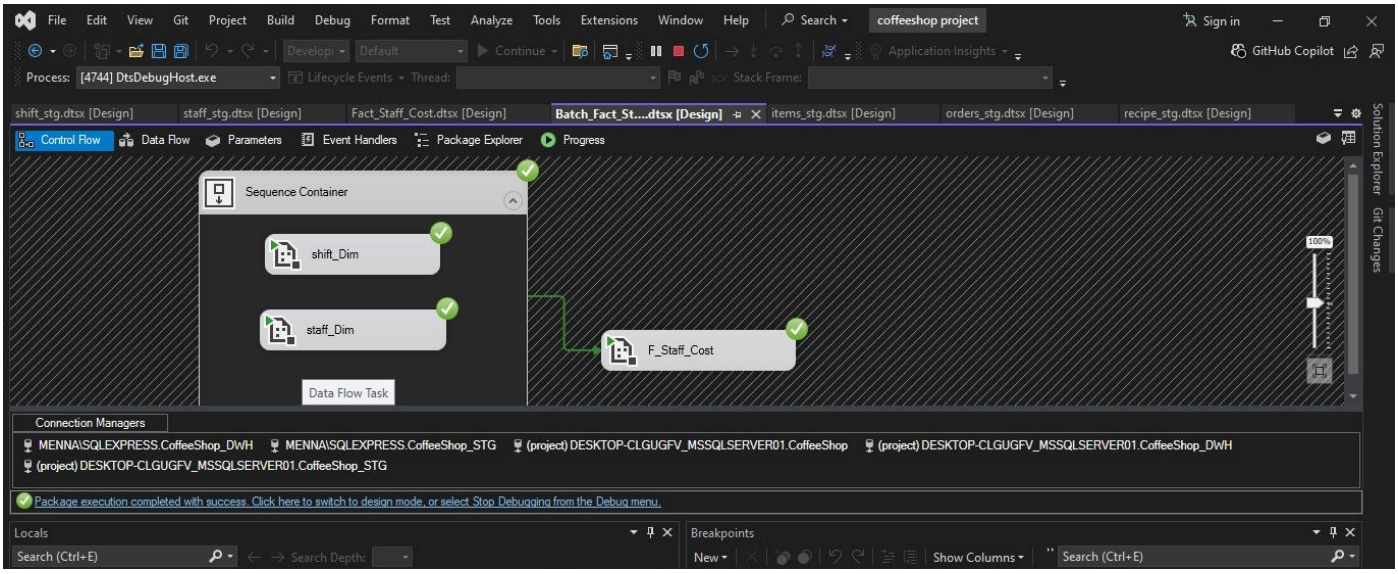
Results | Messages

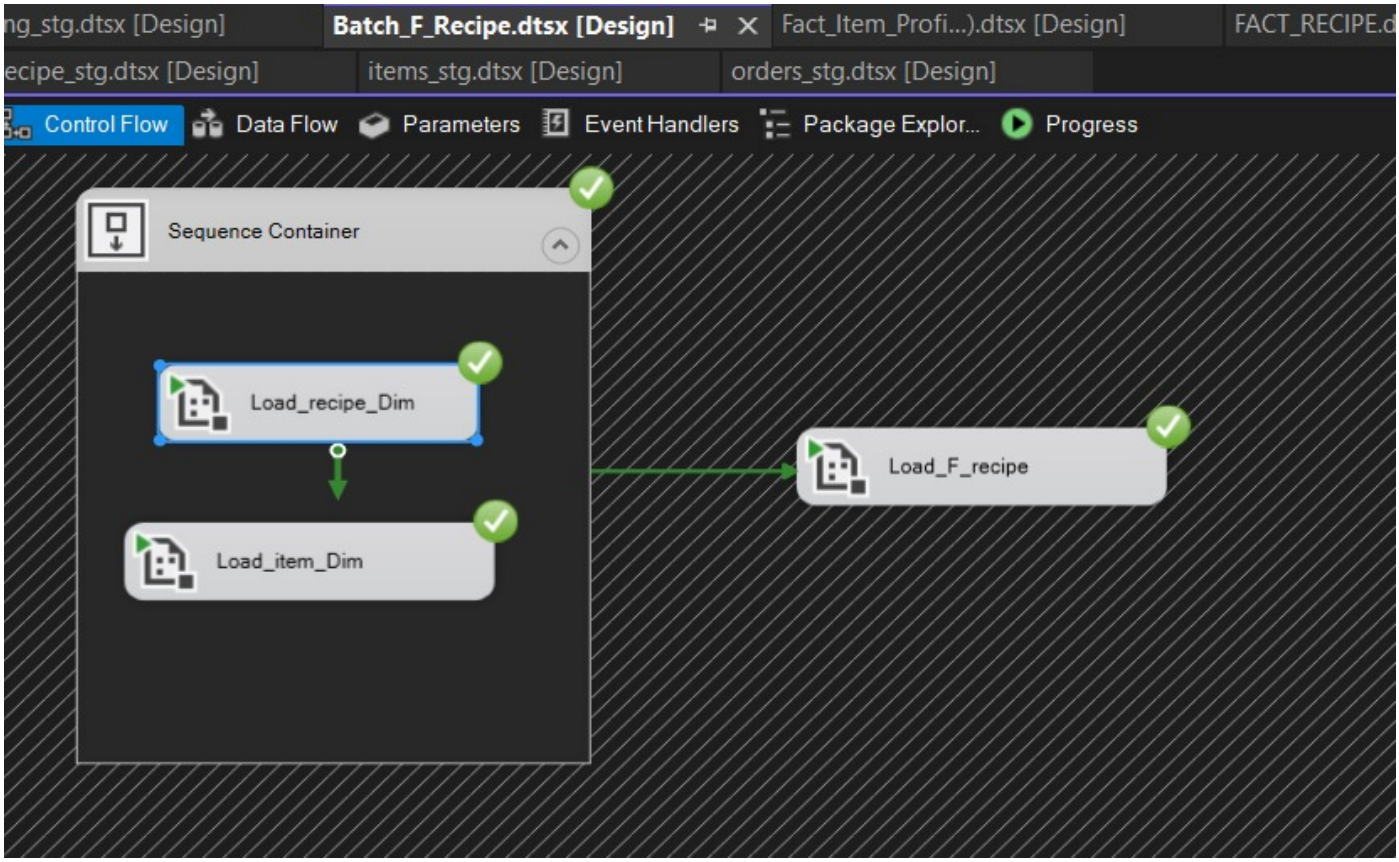
	customer_id	cust_name	Avg_Spending
1	1	Alex	89.600000
2	2	Alice	4.700000
3	3	Amelia	18.150000
4	4	Amy	17.350000
5	5	Anna	7.200000
6	6	Avery	19.700000
7	7	Blake	4.600000
8	8	Bob	10.700000
9	9	Brad	14.600000
10	10	Brooke	55.700000
11	11	Cameron	57.500000
12	12	Cara	3.450000
13	13	Carol	4.200000
14	14	Casey	71.000000
15	15	Charles	9.200000
16	16	Charlie	11.100000
17	17	Chloe	8.400000
18	18	Chris	46.950000
19	19	Cindy	7.550000
20	20	Dakota	8.450000

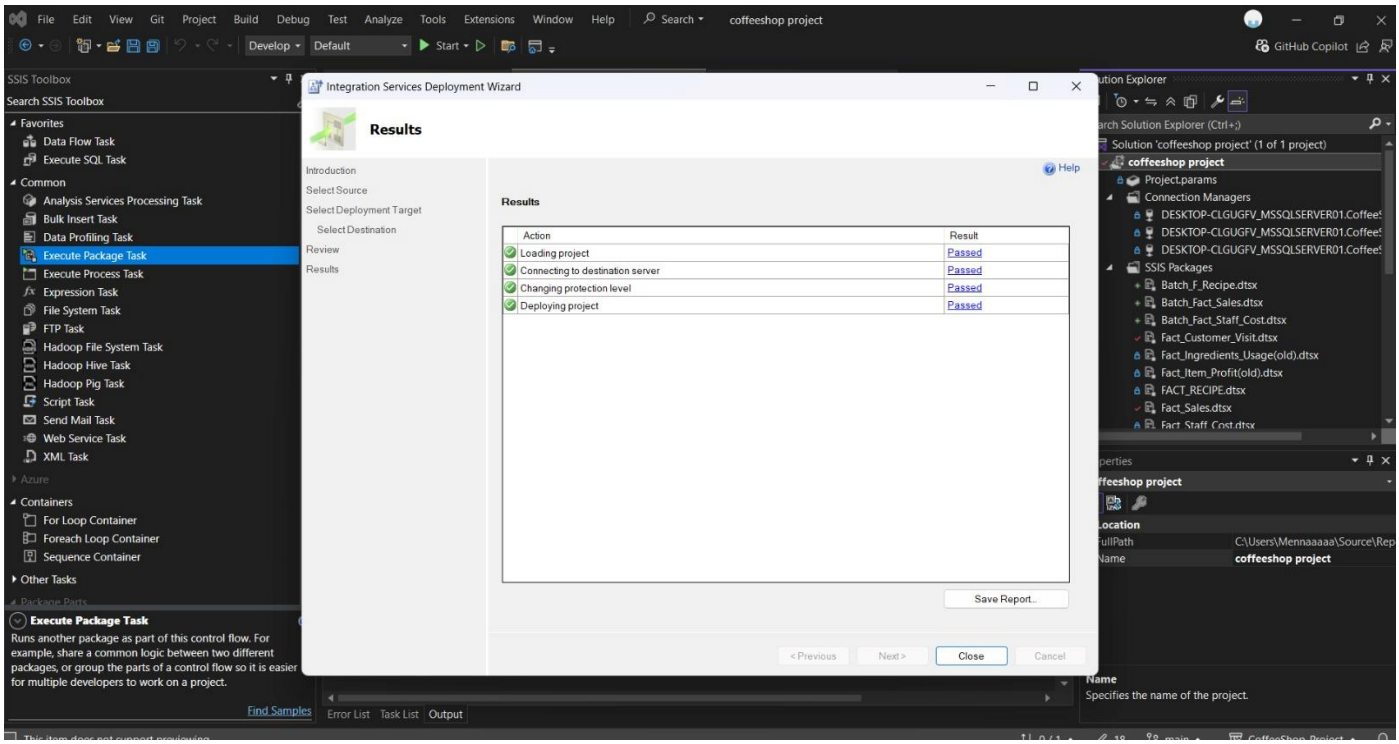
Query executed successfully.

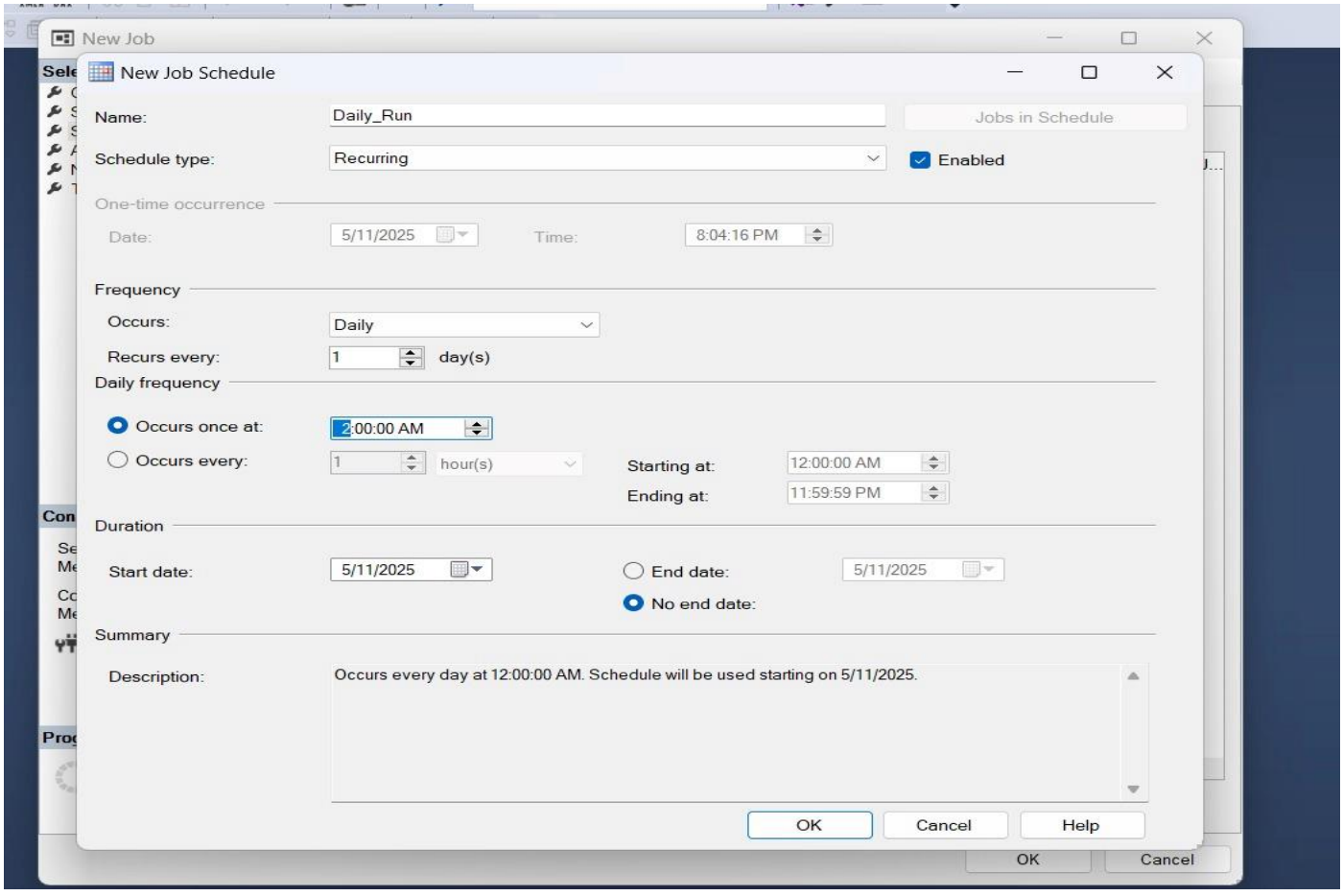
8. Screenshots of the deployed packages in SSIS with their schedule.

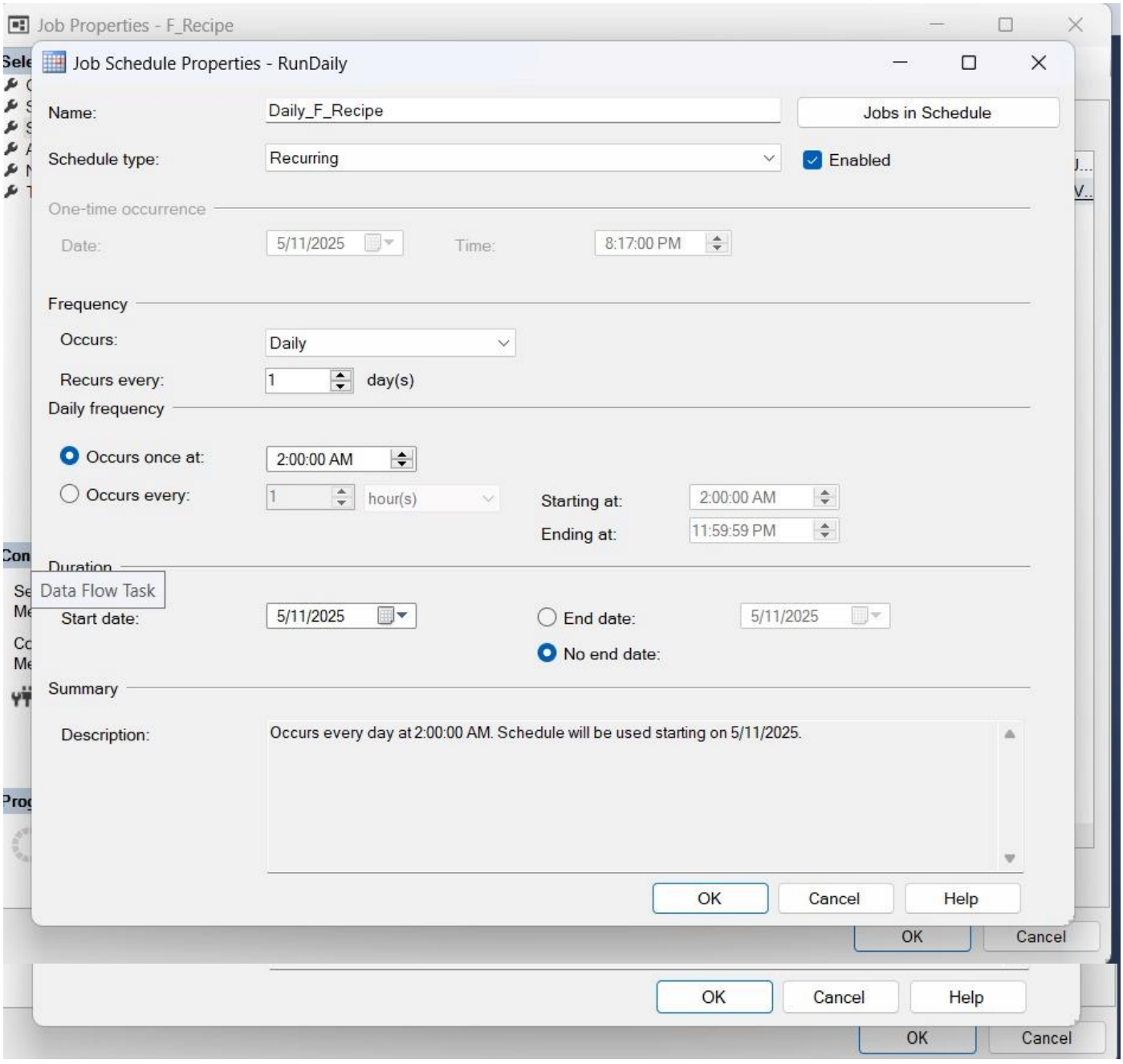


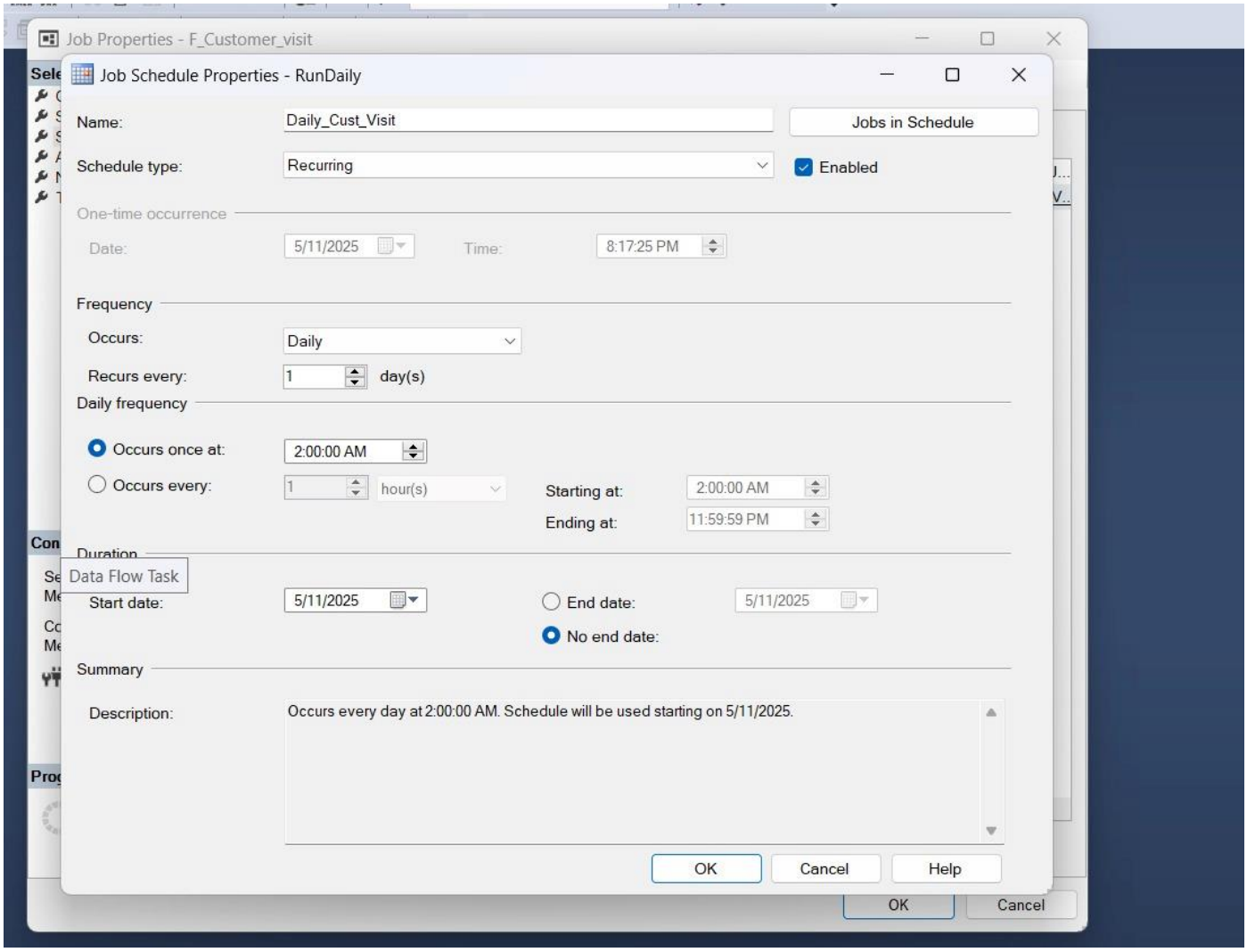

File Edit View Git Project Build Debug Format Test Analyze Tools Extensions Window Help
coffeeshop project
Sign in
GitHub Copilot
Process: [4744] DtsDebugHost.exe
shift_stg.dtsx [Design]
staff_stg.dtsx [Design]
Fact_Staff_Cost.dtsx [Design]
Batch_Fact_St....dtsx [Design]
items_stg.dtsx [Design]
orders_stg.dtsx [Design]
recipe_stg.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explorer
Progress
Sequence Container
shift_Dim
staff_Dim
Data Flow Task
F_Staff_Cost
Solution Explorer
Git Changes
Connection Managers
MENNA\SQLEXPRESS.CoffeeShop_DWH
MENNA\SQLEXPRESS.CoffeeShop_STG
(project) DESKTOP-CLGUGFV_MSSQLSERVER01.CoffeeShop
(project) DESKTOP-CLGUGFV_MSSQLSERVER01.CoffeeShop_DWH
(project) DESKTOP-CLGUGFV_MSSQLSERVER01.CoffeeShop_STG
Package execution completed with success. Click here to switch to design mode, or select Stop Debugging from the Debug menu.
Locals
Search (Ctrl+E)
Breakpoints
New
Show Columns
Search (Ctrl+E)

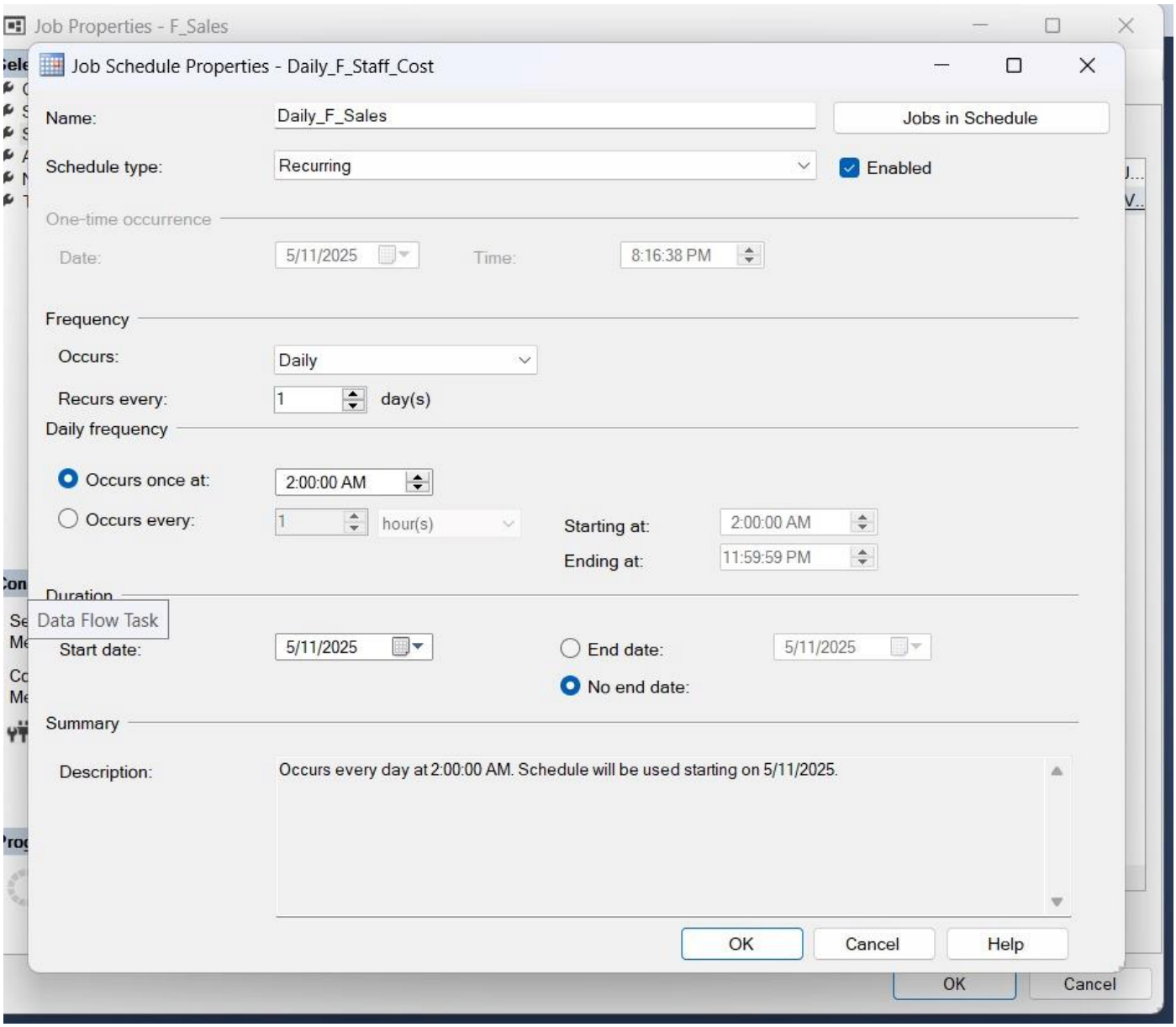

ng_stg.dtsx [Design]
Batch_F_Recipe.dtsx [Design]
Fact_Item_Profi...).dtsx [Design]
FACT_RECIPE.d
ecipe_stg.dtsx [Design]
items_stg.dtsx [Design]
orders_stg.dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explor...
Progress
Sequence Container
Load_recipe_Dim
Load_item_Dim
Load_F_recipe


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
coffeeshop project
Integration Services Deployment Wizard
Results
Introduction
Select Source
Select Deployment Target
Select Destination
Review
Results
Action
Result
Loading project
Passed
Connecting to destination server
Passed
Changing protection level
Passed
Deploying project
Passed
Save Report...
< Previous
Next >
Close
Cancel
SSIS Toolbox
Search SSIS Toolbox
Favorites
Data Flow Task
Execute SQL Task
Common
Analysis Services Processing Task
Bulk Insert Task
Data Profiling Task
Execute Package Task
Execute Process Task
Expression Task
File System Task
FTP Task
Hadoop File System Task
Hadoop Hive Task
Hadoop Pig Task
Script Task
Send Mail Task
Web Service Task
XML Task
Azure
Containers
For Loop Container
Foreach Loop Container
Sequence Container
Other Tasks
Execute Package Task
Runs another package as part of this control flow. For example, share a common logic between two different packages, or group the parts of a control flow so it is easier for multiple developers to work on a project.
Find Samples
Error List
Task List
Output
Solution Explorer
Solution 'coffeeshop project' (1 of 1 project)
coffeeshop project
Project.params
Connection Managers
SSIS Packages
Batch_F_Recipe.dtsx
Batch_Fact_Sales.dtsx
Batch_Fact_Staff_Cost.dtsx
Fact_Customer_Visit.dtsx
Fact_Ingredients_Usage(old).dtsx
Fact_Item_Profit(old).dtsx
FACT_RECIPE.dtsx
Fact_Sales.dtsx
Name
Specifies the name of the project.

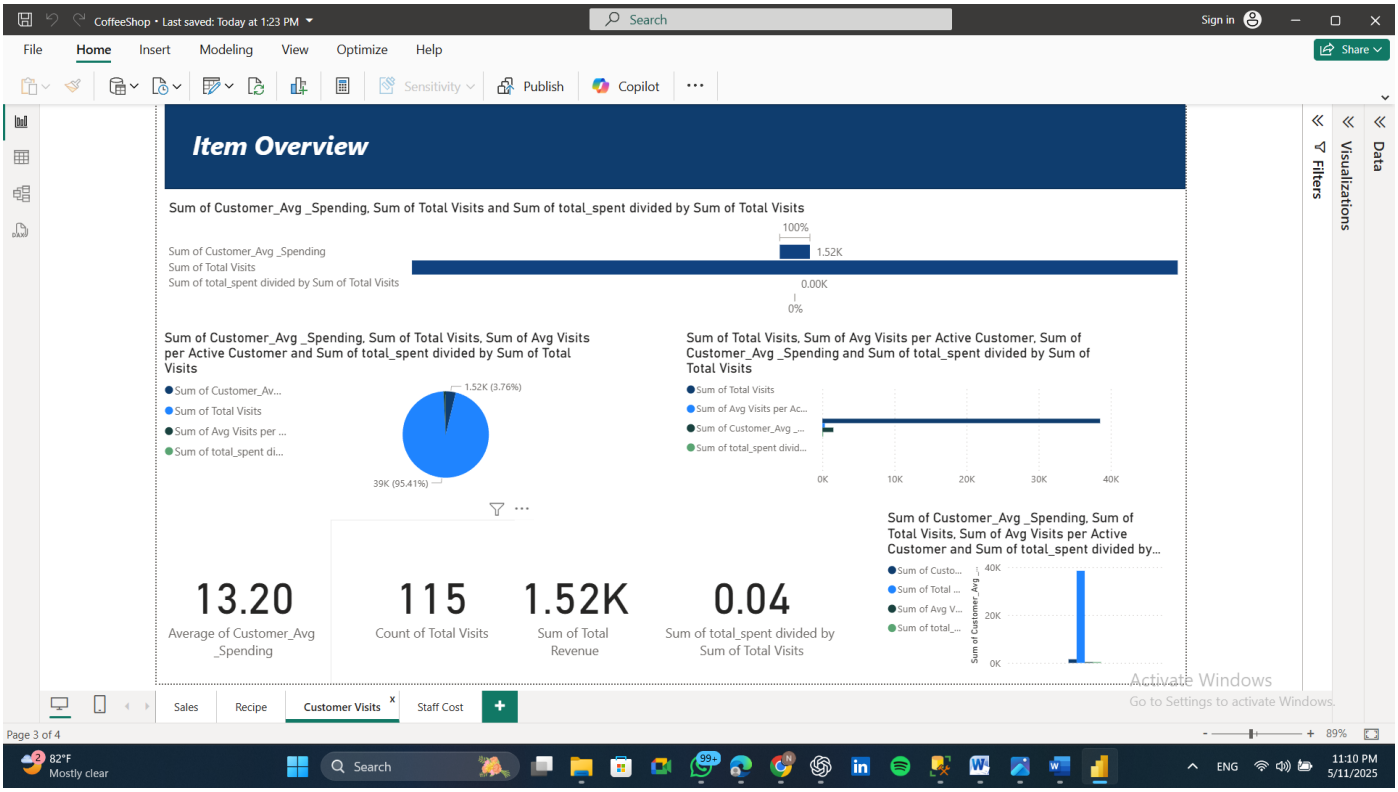

New Job
New Job Schedule
Name:
Daily_Run
Jobs in Schedule
Schedule type:
Recurring
Enabled
One-time occurrence
Date:
5/11/2025
Time:
8:04:16 PM
Frequency
Occurs:
Daily
Recurs every:
1
day(s)
Daily frequency
Occurs once at:
2:00:00 AM
Occurs every:
1
hour(s)
Starting at:
12:00:00 AM
Ending at:
11:59:59 PM
Duration
Start date:
5/11/2025
End date:
5/11/2025
No end date:
Summary
Description:
Occurs every day at 12:00:00 AM. Schedule will be used starting on 5/11/2025.
OK
Cancel
Help
OK
Cancel

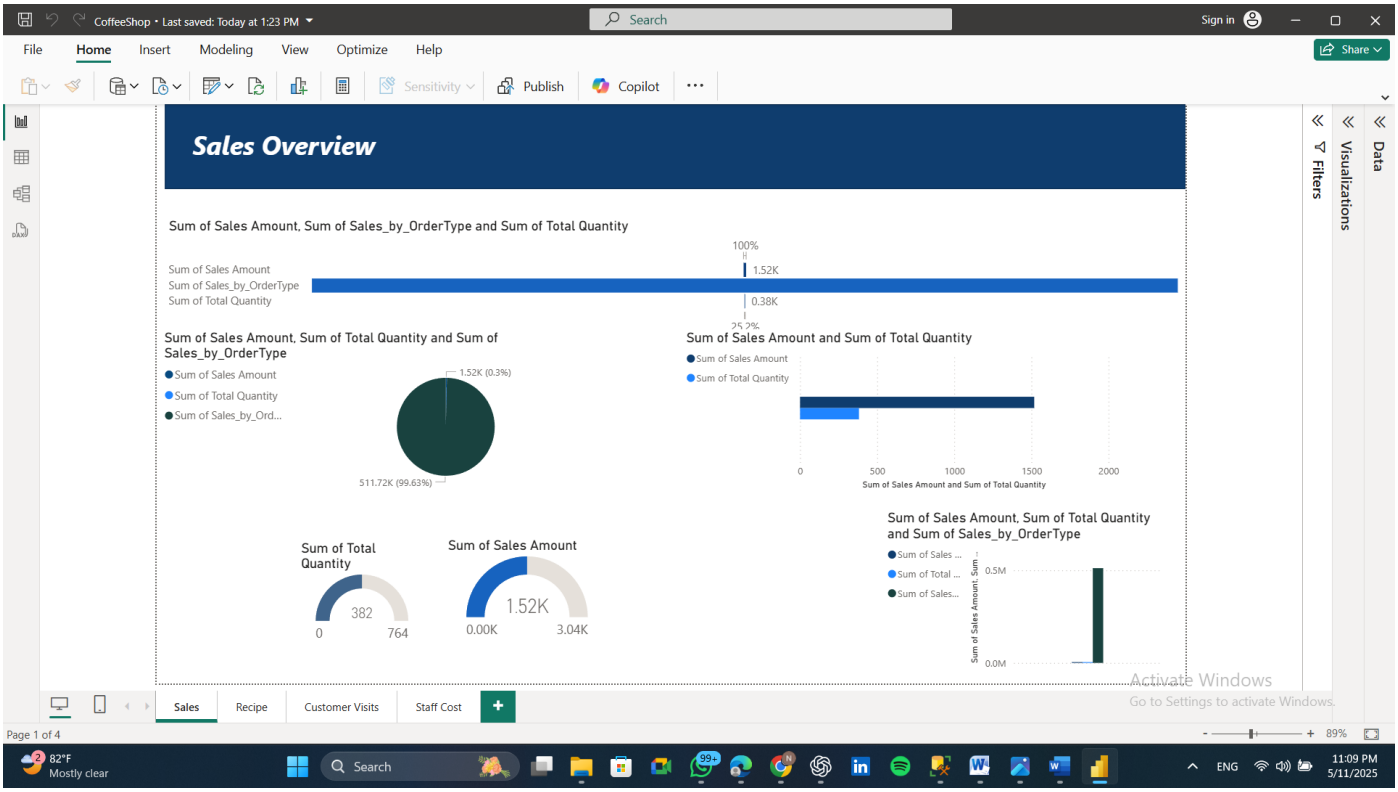

Job Properties - F_Recipe
Job Schedule Properties - RunDaily
Name:
Daily_F_Recipe
Jobs in Schedule
Schedule type:
Recurring
Enabled
One-time occurrence
Date:
5/11/2025
Time:
8:17:00 PM
Frequency
Occurs:
Daily
Recurs every:
1
day(s)
Daily frequency
Occurs once at:
2:00:00 AM
Occurs every:
1
hour(s)
Starting at:
2:00:00 AM
Ending at:
11:59:59 PM
Duration
Start date:
5/11/2025
End date:
5/11/2025
No end date:
Summary
Description:
Occurs every day at 2:00:00 AM. Schedule will be used starting on 5/11/2025.
OK
Cancel
Help
Data Flow Task
OK
Cancel
OK
Cancel
Help
OK
Cancel

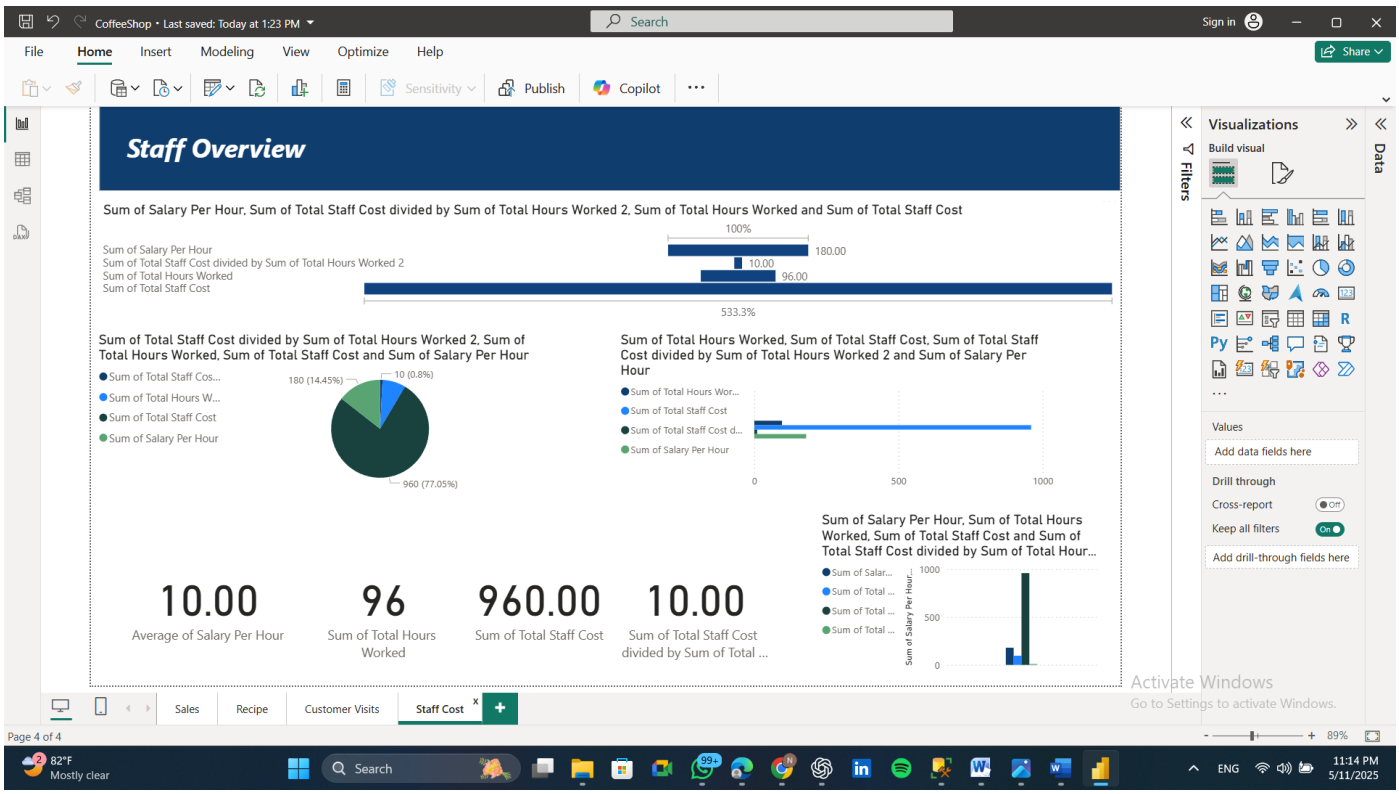

Job Properties - F_Customer_visit
Job Schedule Properties - RunDaily
Name:
Daily_Cust_Visit
Jobs in Schedule
Schedule type:
Recurring
Enabled
One-time occurrence
Date:
5/11/2025
Time:
8:17:25 PM
Frequency
Occurs:
Daily
Recurs every:
1
day(s)
Daily frequency
Occurs once at:
2:00:00 AM
Occurs every:
1
hour(s)
Starting at:
2:00:00 AM
Ending at:
11:59:59 PM
Duration
Start date:
5/11/2025
End date:
5/11/2025
No end date:
Summary
Description:
Occurs every day at 2:00:00 AM. Schedule will be used starting on 5/11/2025.
OK
Cancel
Help
Data Flow Task
OK
Cancel

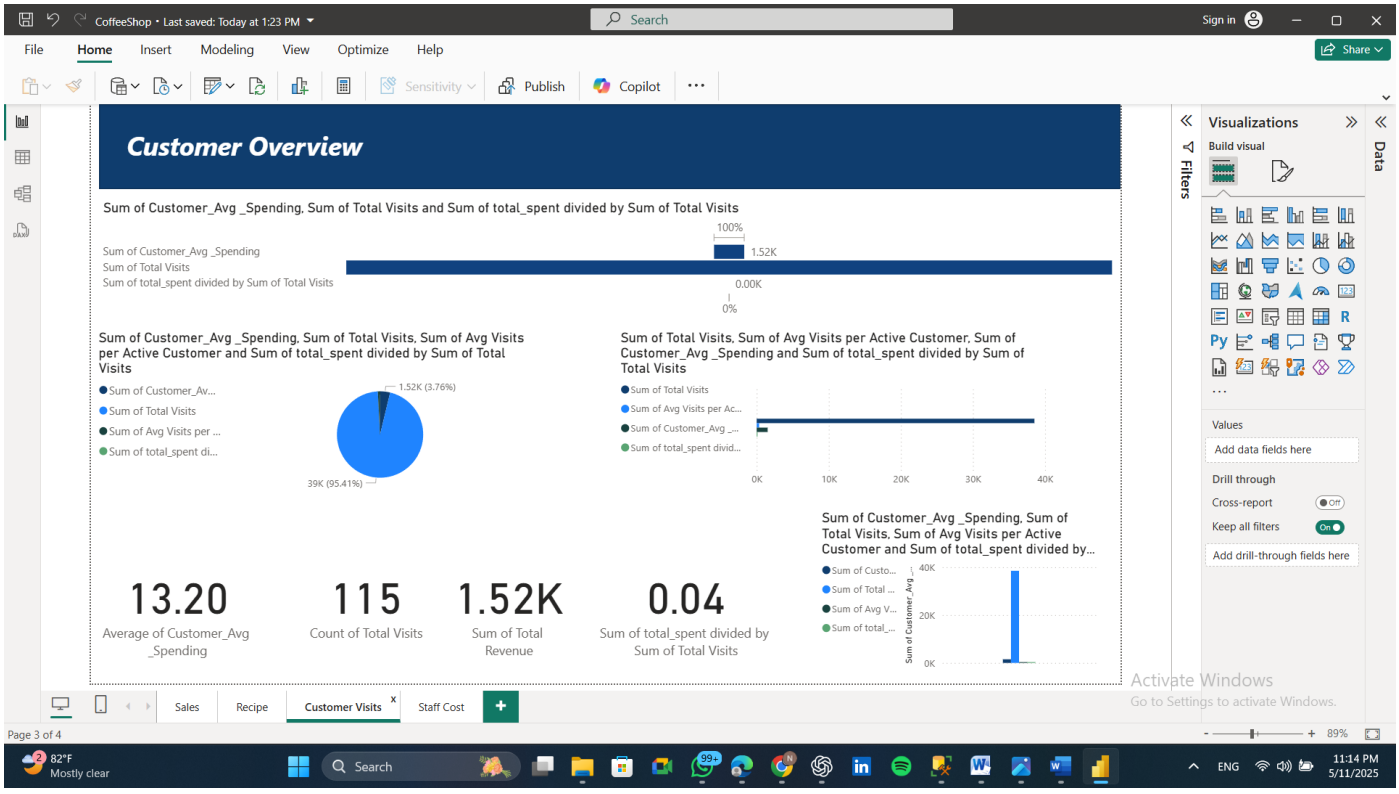

Job Properties - F_Sales
Job Schedule Properties - Daily_F_Staff_Cost
Name:
Daily_F_Sales
Jobs in Schedule
Schedule type:
Recurring
Enabled
One-time occurrence
Date:
5/11/2025
Time:
8:16:38 PM
Frequency
Occurs:
Daily
Recurs every:
1
day(s)
Daily frequency
Occurs once at:
2:00:00 AM
Occurs every:
1
hour(s)
Starting at:
2:00:00 AM
Ending at:
11:59:59 PM
Duration
Start date:
5/11/2025
End date:
5/11/2025
No end date:
Summary
Description:
Occurs every day at 2:00:00 AM. Schedule will be used starting on 5/11/2025.
OK
Cancel
Help
Data Flow Task
OK
Cancel

9. Data Visualization Using PowerBI.






CoffeeShop • Last saved: Today at 1:23 PM
Search
Sign in
File
Home
Insert
Modeling
View
Optimize
Help
Share
Publish
Copilot
Staff Overview
Sum of Salary Per Hour, Sum of Total Staff Cost divided by Sum of Total Hours Worked 2, Sum of Total Hours Worked and Sum of Total Staff Cost
Sum of Salary Per Hour
Sum of Total Staff Cost divided by Sum of Total Hours Worked 2
Sum of Total Hours Worked
Sum of Total Staff Cost
100%
180.00
10.00
96.00
533.3%
Sum of Total Staff Cost divided by Sum of Total Hours Worked 2, Sum of Total Hours Worked, Sum of Total Staff Cost and Sum of Salary Per Hour
180 (14.45%)
10 (0.8%)
960 (77.05%)
Sum of Total Hours Worked, Sum of Total Staff Cost, Sum of Total Staff Cost divided by Sum of Total Hours Worked 2 and Sum of Salary Per Hour
0
500
1000
Sum of Salary Per Hour, Sum of Total Hours Worked, Sum of Total Staff Cost and Sum of Total Staff Cost divided by Sum of Total Hour...
10.00
Average of Salary Per Hour
96
Sum of Total Hours Worked
960.00
Sum of Total Staff Cost
10.00
Sum of Total Staff Cost divided by Sum of Total ...
Visualizations
Build visual
Values
Add data fields here
Drill through
Cross-report
Keep all filters
Add drill-through fields here
Filters
Data
Sales
Recipe
Customer Visits
Staff Cost
Page 4 of 4
89%
82°F
Mostly clear
ENG
11:14 PM
5/11/2025
Activate Windows
Go to Settings to activate Windows.


CoffeeShop • Last saved: Today at 1:23 PM
Search
Sign in
File
Home
Insert
Modeling
View
Optimize
Help
Share
Publish
Copilot
Customer Overview
Sum of Customer_Avg _Spending, Sum of Total Visits and Sum of total_spent divided by Sum of Total Visits
Sum of Customer_Avg _Spending
Sum of Total Visits
Sum of total_spent divided by Sum of Total Visits
100%
1.52K
0.00K
0%
Sum of Customer_Avg _Spending, Sum of Total Visits, Sum of Avg Visits per Active Customer and Sum of total_spent divided by Sum of Total Visits
1.52K (3.76%)
39K (95.41%)
Sum of Total Visits, Sum of Avg Visits per Active Customer, Sum of Customer_Avg _Spending and Sum of total_spent divided by Sum of Total Visits
0K
10K
20K
30K
40K
Sum of Customer_Avg _Spending, Sum of Total Visits, Sum of Avg Visits per Active Customer and Sum of total_spent divided by...
13.20
Average of Customer_Avg _Spending
115
Count of Total Visits
1.52K
Sum of Total Revenue
0.04
Sum of total_spent divided by Sum of Total Visits
Visualizations
Build visual
Values
Add data fields here
Drill through
Cross-report
Keep all filters
Add drill-through fields here
Filters
Data
Sales
Recipe
Customer Visits
Staff Cost
Page 3 of 4
89%
82°F
Mostly clear
ENG
11:14 PM
5/11/2025
Activate Windows
Go to Settings to activate Windows.