SPSS PRACTICAL

Name:	Ganesh Agrahari
Class:	BCA DS 33
Roll No.:	1230258176
Submitted To:	Mr. Robin Tyagi

INDEX

SPSS Practical 8

This practical includes:

- Objective and Introduction
- Theoretical Background
- Methodology and Procedure
- Data Analysis
- Results and Interpretation
- Screenshots and Output
- Conclusion
- References

Practical: 8

Definition:

You work for a telecommunications firm where you have to cleanse and enrich a dataset that stores demographic and churn data on the company's customers. Using the transformed dataset

Outcomes/Learning:

- Learned to derive new calculated and conditional fields using **CLEM expressions**.
- Understood how to perform **data cleansing and enrichment** for improved data quality.
- Gained experience in **string, date, and numeric field manipulation** using Derive Nodes.
- Prepared a complete and reliable dataset for **churn analysis and predictive modelling**.

Required Tool:

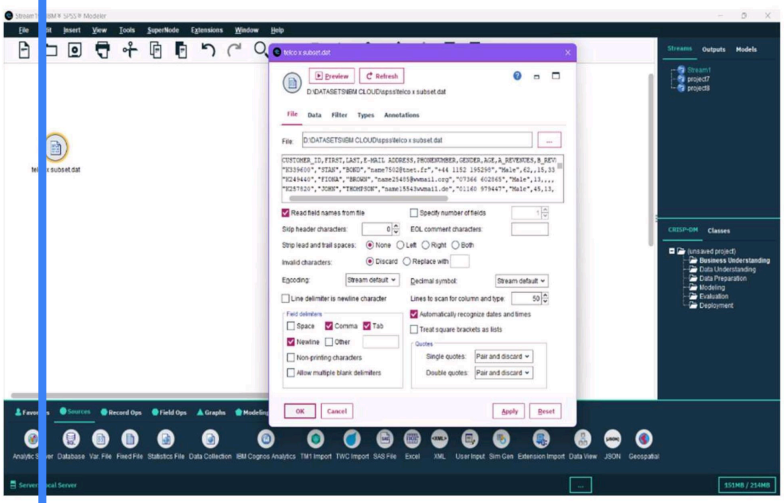
IBM SPSS MODELER

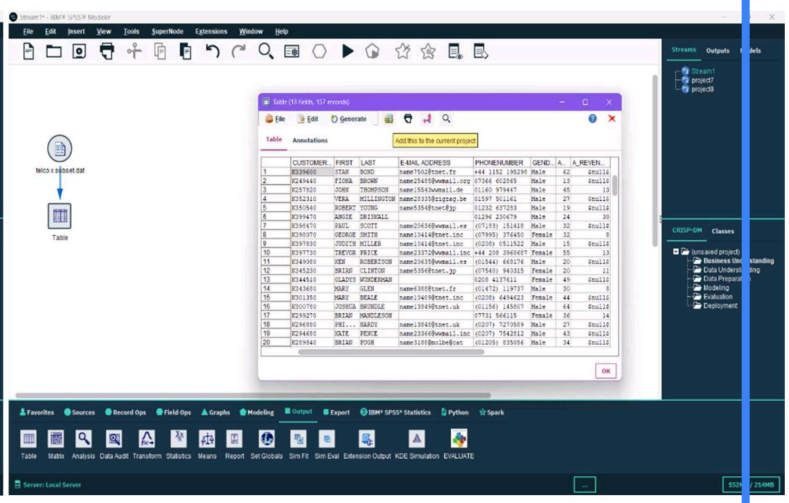
Working:

In this practical, the goal is to **cleanse, transform, and enrich** a customer dataset to prepare it for analysis and modelling.
The dataset (*telco x subset.dat*) is imported using a **Var. File Node**, and several **Derive Nodes** are applied to create new calculated and conditional fields using **CLEM (Control Language for Expression Manipulation)**.
New fields such as *MONTHS_CUSTOMER*, *DOMAIN NAME*, *MEAN_REVENUES*, and *CHURN* are generated to improve data consistency, fill missing information, and enhance analytical readiness.
The main nodes used are:
- **Var. File Node** – to import the dataset.
- **Derive Node** – to create calculated, conditional, and flag fields using CLEM expressions.
- **Table Node** – to verify and display transformed data.

Step 1: Importing the Dataset

- Open IBM SPSS Modeler → create a new stream.
- Add a **Var. File Node** from the Sources tab → browse and select *telco x subset.dat*.
- Click **Apply** → **OK** to load the dataset.
- Attach a **Table Node** → click **Run** to view the imported records.



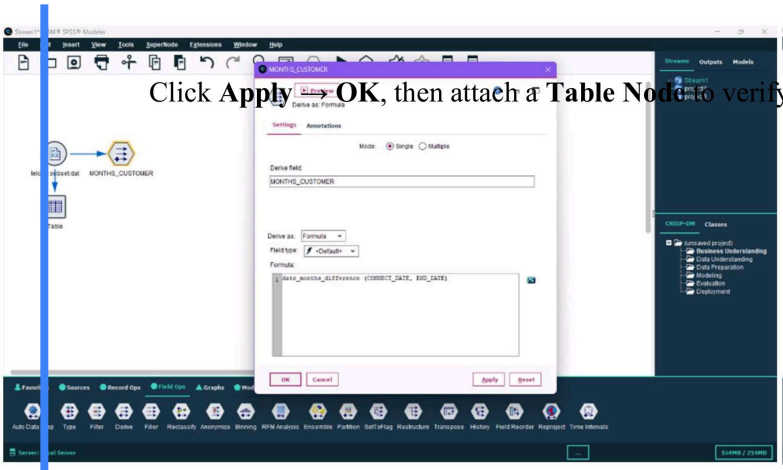


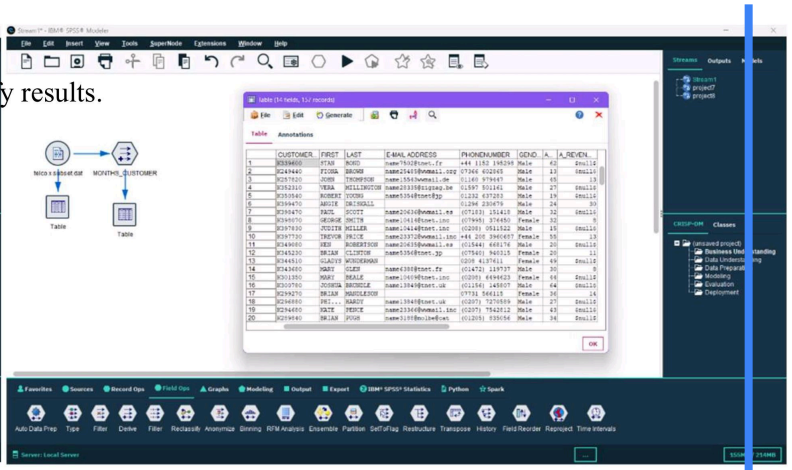
Step 2: Creating the MONTHS_CUSTOMER Field

Attach a **Derive Node** to the Var. File Node.
Configure as follows:

- Rename the field to **MONTHS_CUSTOMER**.
- Set **Derive As = Formula** and **Field Type = Default**.
- Enter the expression:
- date_months_difference(CONNECT_DATE, END_DATE)

Click **Apply → OK**, then attach a **Table Node** to verify results.





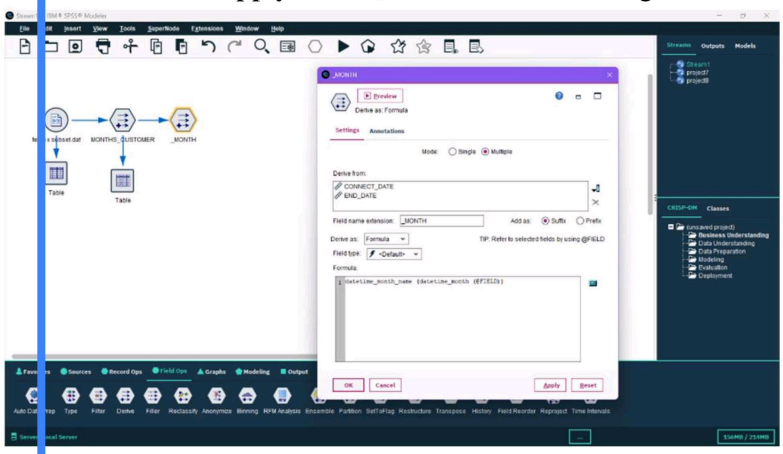
Step 3: Creating the _MONTHS Field

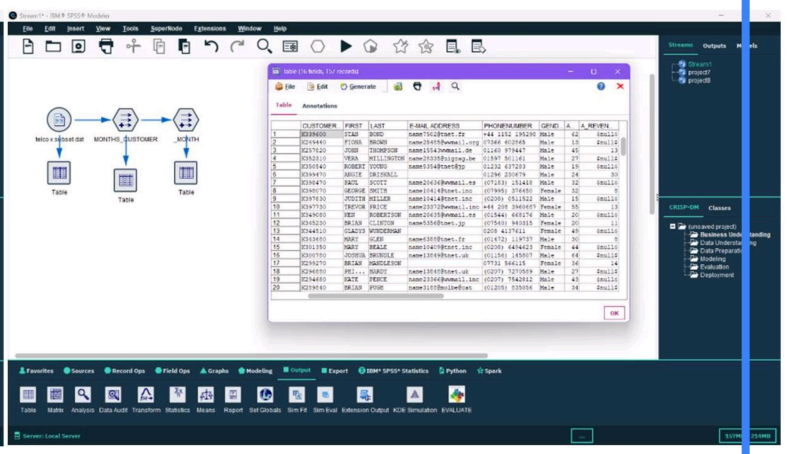
Connect another **Derive Node** to the MONTHS_CUSTOMER Node.

Configure as follows:

- Rename the field to **_MONTHS**.
- Set **Mode = Multiple**, **Derive As = Formula**, **Field Type = Default**.
- Enter the expression:
- datetime_month_name(datetime_month(@FIELD))

Click **Apply → OK**, then view results using a **Table Node**.





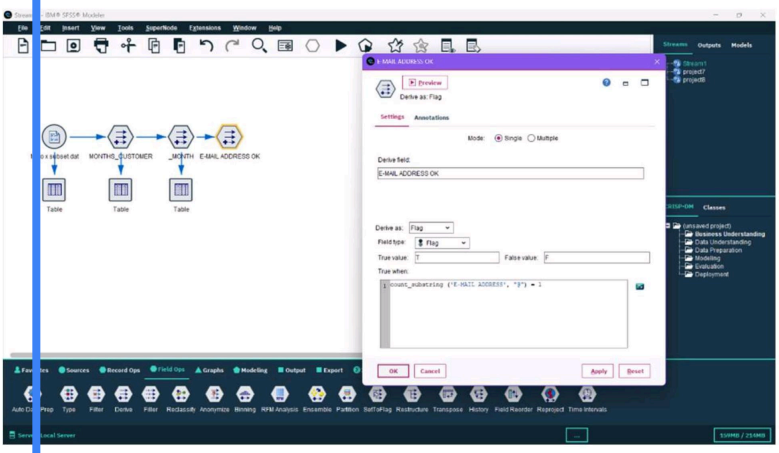
Step 4: Creating the E-MAIL ADDRESS OK Field

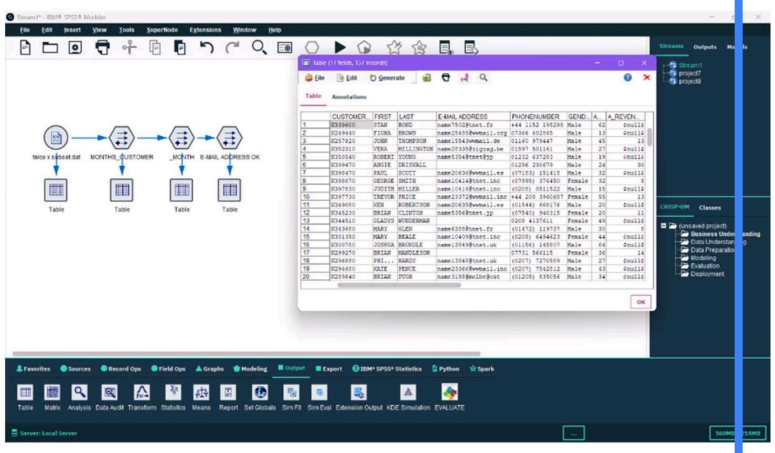
Attach a new **Derive Node** to the _MONTHS Node.

Configure as follows:

- Rename it to **E-MAIL ADDRESS OK**.
- Set **Mode = Single**, **Derive As = Flag**, **Field Type = Flag**.
- Enter the expression:
- count_substring('E-MAIL ADDRESS', "@") = 1

Click **Apply → OK**, and check results with a **Table Node**.



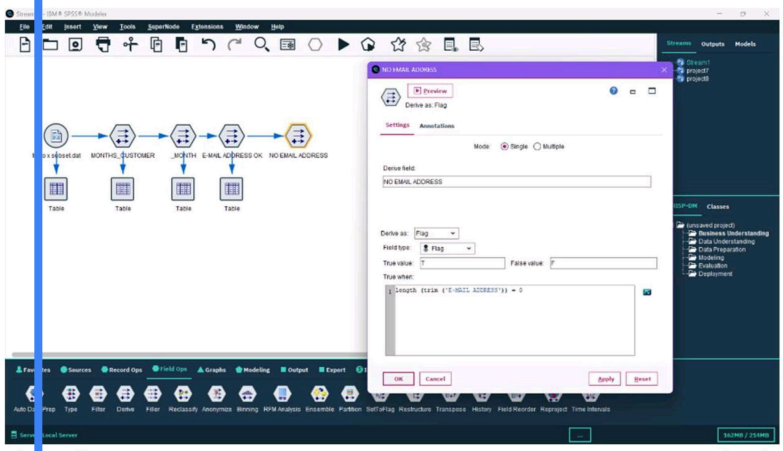


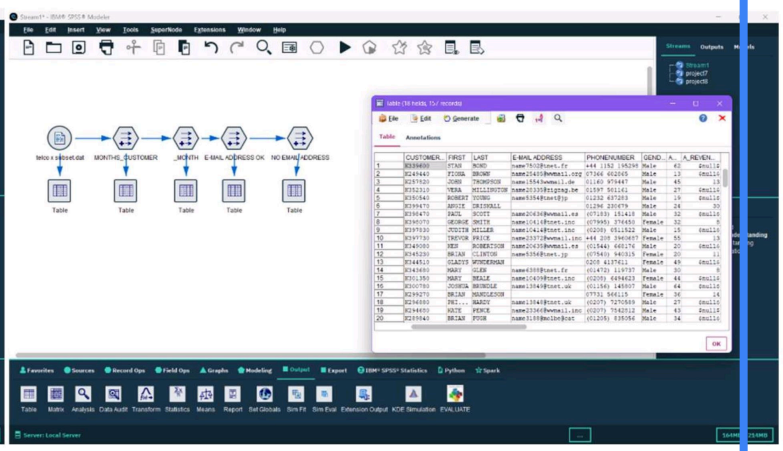
Step 5: Creating the NO EMAIL ADDRESS Field

Connect another **Derive Node** to the E-MAIL ADDRESS OK Node.
Configure as follows:

- Rename to **NO EMAIL ADDRESS**.
- Set **Mode = Single**, **Derive As = Flag**, **Field Type = Flag**.
- Enter the expression:
- length(trim('E-MAIL ADDRESS')) = 0

Click **Apply → OK**, and view results with a **Table Node**.



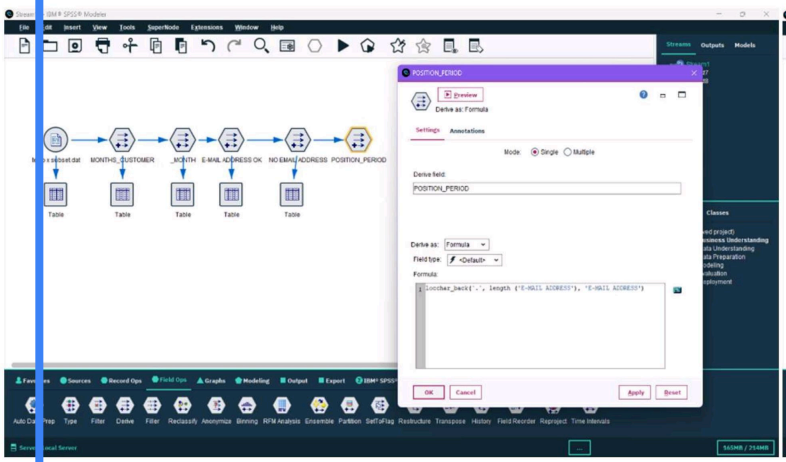


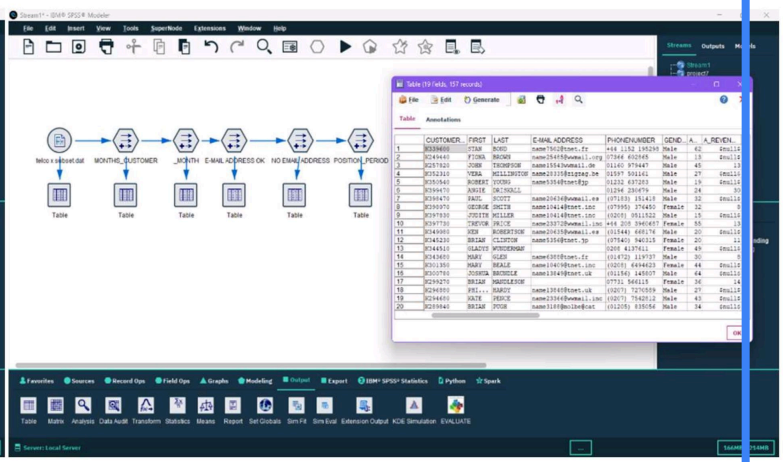
Step 6: Creating the POSITION_PERIOD Field

Attach a **Derive Node** to the NO EMAIL ADDRESS Node.
Configure as follows:

- Rename to **POSITION_PERIOD**.
- Set **Mode = Single**, **Derive As = Formula**, **Field Type = Default**.
- Enter the formula:
- locchar_back('.', length('E-MAIL ADDRESS'), 'E-MAIL ADDRESS')

Click **Apply → OK**, and check the output using a **Table Node**.



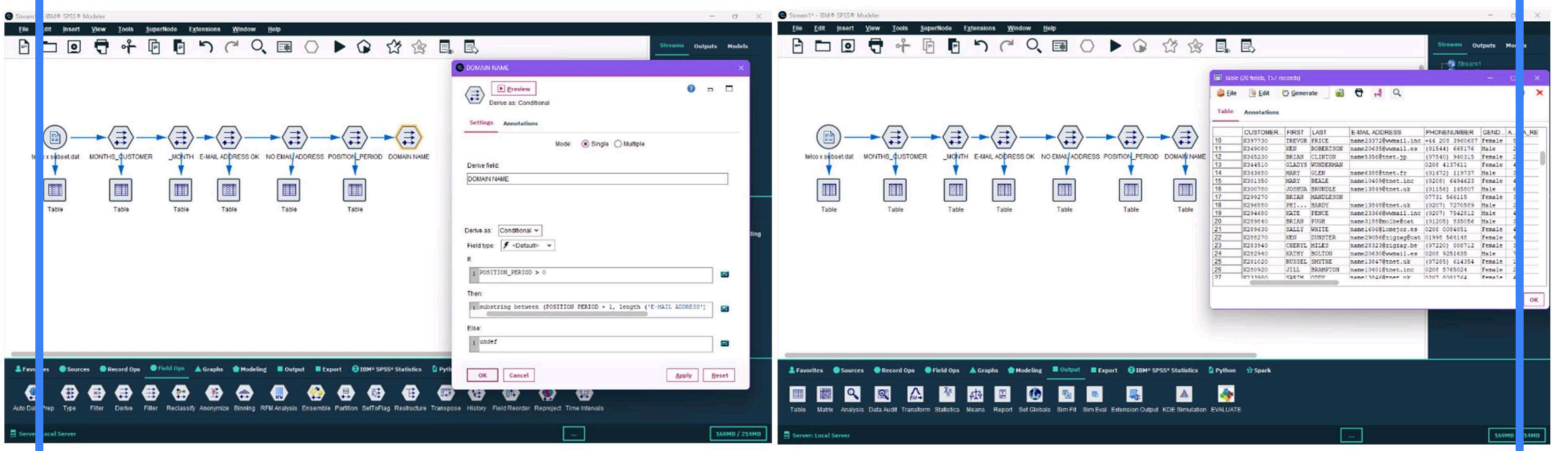


Step 7: Creating the DOMAIN NAME Field

Add another **Derive Node** connected to the POSITION_PERIOD Node.
Configure as follows:

- Rename to **DOMAIN NAME**.
- Set **Mode = Single**, **Derive As = Conditional**, **Field Type = Default**.
- Enter the expression:
- If POSITION_PERIOD > 0 Then substring_between(POSITION_PERIOD + 1, length('E-MAIL ADDRESS')) Else undef

Click **Apply → OK**, and validate results using a **Table Node**.



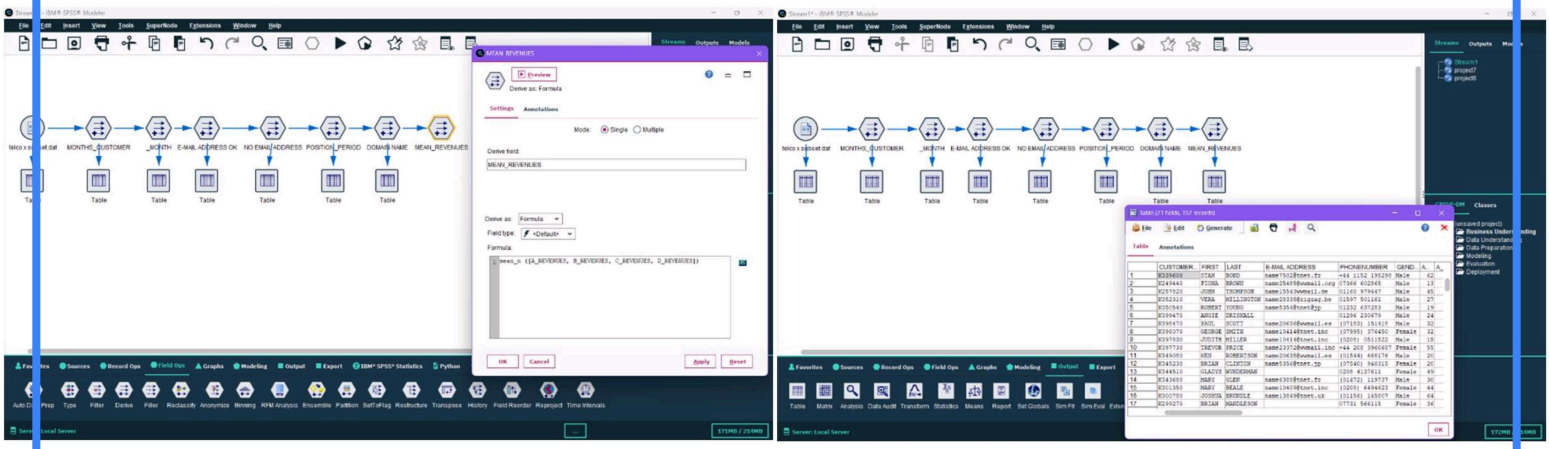
Step 8: Creating the MEAN_REVENUES Field

Attach a **Derive Node** to the DOMAIN NAME Node.

Configure as follows:

- Rename to **MEAN_REVENUES**.
- Set **Mode = Single**, **Derive As = Formula**, **Field Type = Default**.
- Enter the expression:
- mean_n([A_REVENUES, B_REVENUES, C_REVENUES, D_REVENUES])

Click **Apply → OK**, and use a **Table Node** to verify results.



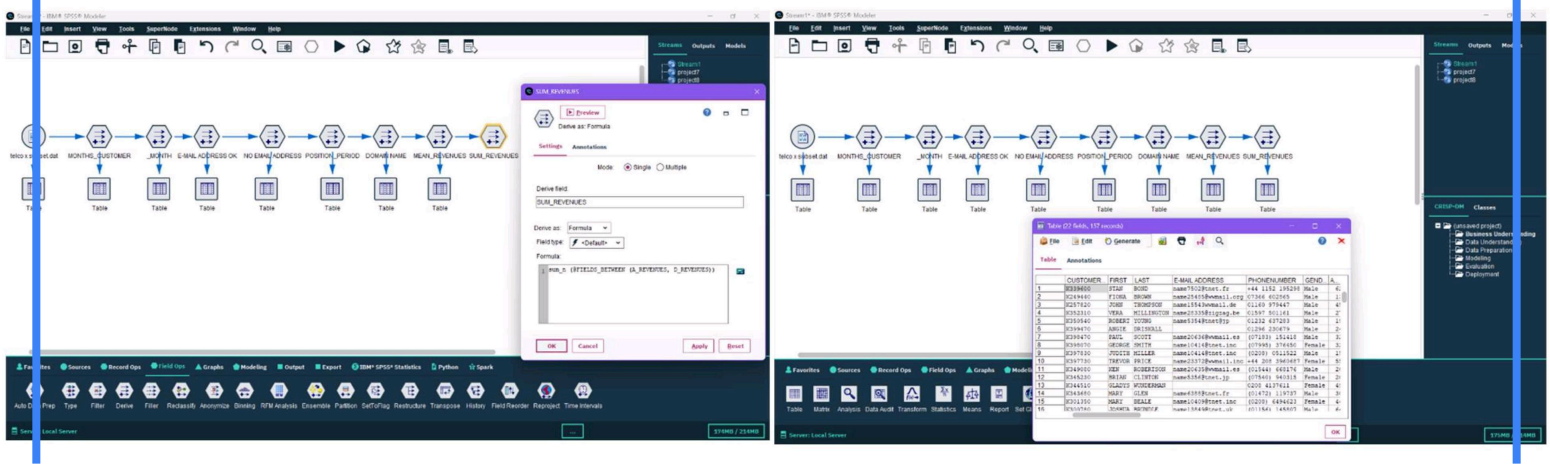
Step 9: Creating the SUM_REVENUES Field

Connect another **Derive Node** to the MEAN_REVENUES Node.

Configure as follows:

- Rename to **SUM_REVENUES**.
- Set **Mode = Single**, **Derive As = Formula**, **Field Type = Default**.
- Enter the expression:
- sum_n(@FIELDS_BETWEEN(A_REVENUES, D_REVENUES))

Click **Apply → OK**, and check the output using a **Table Node**.



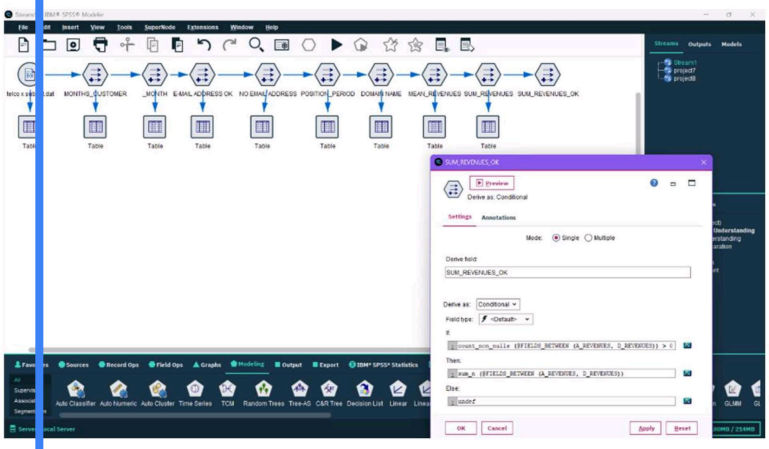
Step 10: Creating the SUM_REVENUES_OK Field

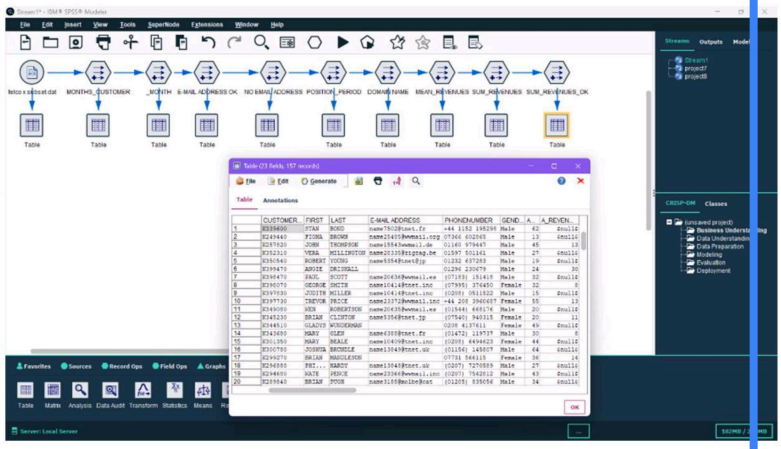
Attach a **Derive Node** to the SUM_REVENUES Node.

Configure as follows:

- Rename to **SUM_REVENUES_OK**.
- Set **Mode = Single**, **Derive As = Conditional**, **Field Type = Default**.
- Enter the expression:
- If count_non_nulls(@FIELDS_BETWEEN(A_REVENUES, D_REVENUES)) > 0 Then sum_n(@FIELDS_BETWEEN(A_REVENUES, D_REVENUES)) Else undef

Click **Apply → OK**, and view results using a **Table Node**.





Step 11: Creating the CHURN Field

Connect a **Derive Node** to the SUM_REVENUES_OK Node.

Configure as follows:

- Rename to **CHURN**.
- Set **Mode = Single**, **Derive As = Flag**, **Field Type = Flag**.
- Enter the formula:
- not(@NULL(END_DATE))

Click **Apply → OK**, then attach a **Table Node** to view the final results.