https://github.com/dotlogon/ETL-demo **The Ultimate PDI Workflow Build Guide (Step-by-Step, UI Focused)**

This transformation is built around three parallel streams: POS Sales (GBP), Online Sales (EUR), and Market Rates (Lookup), which are then combined and processed.  
  
**Sample Workflow developed on Pentaho**  

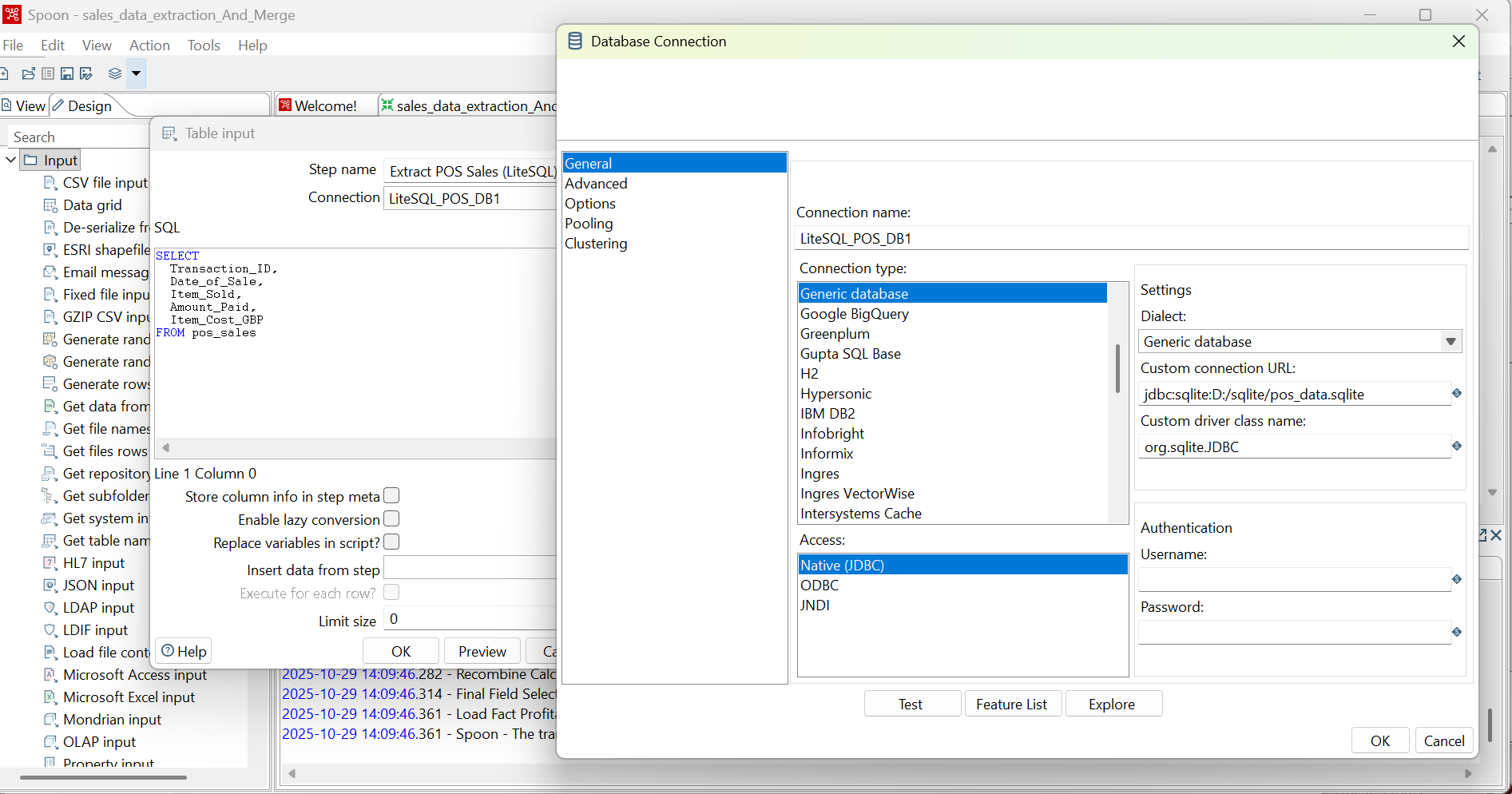

a. Database connections (sqlite)

|  |
| --- |
| <https://sqlite.org/download.html> [sqlite-tools-win-x64-3500400.zip](https://sqlite.org/2025/sqlite-tools-win-x64-3500400.zip)  D:\sqlite> .\sqlite3.exe D:\sqlite\pos\_data.sqlite  sqlite> .databases  sqlite> CREATE TABLE IF NOT EXISTS pos\_sales (  Transaction\_ID TEXT PRIMARY KEY,  StoreID INTEGER,  Date\_of\_Sale TEXT, -- SQLite often uses TEXT for dates  Item\_Sold TEXT,  Amount\_Paid REAL,  Item\_Cost\_GBP REAL  );  sqlite> .tables  sqlite> INSERT INTO pos\_sales (Transaction\_ID, StoreID, Date\_of\_Sale, Item\_Sold, Amount\_Paid, Item\_Cost\_GBP)  VALUES ('POS2001', 12, '2025-10-25', 'Item X', 75.00, 40.00);  sqlite> INSERT INTO pos\_sales (Transaction\_ID, StoreID, Date\_of\_Sale, Item\_Sold, Amount\_Paid, Item\_Cost\_GBP)  VALUES ('POS2002', 15, '2025-10-26', 'Item Y', 200.00, 110.00);  sqlite> INSERT INTO pos\_sales (Transaction\_ID, StoreID, Date\_of\_Sale, Item\_Sold, Amount\_Paid, Item\_Cost\_GBP)  VALUES ('POS2003', 12, '2025-10-27', 'Item X', 75.00, 40.00);  sqlite> SELECT \* FROM pos\_sales LIMIT 5;  sqlite> .quit |

Install Pentaho  
<https://pentaho.com/pentaho-developer-edition/#communityProducts>  
Enter any details and then you get download list, select below one and download.

pdi-ce-10.2.0.0-222.zip  
  
D:\pentaho > Spoon.bat

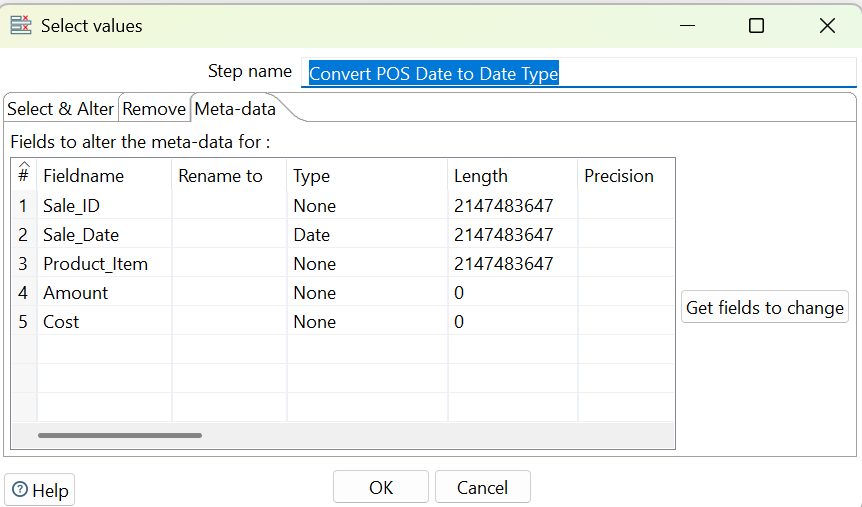
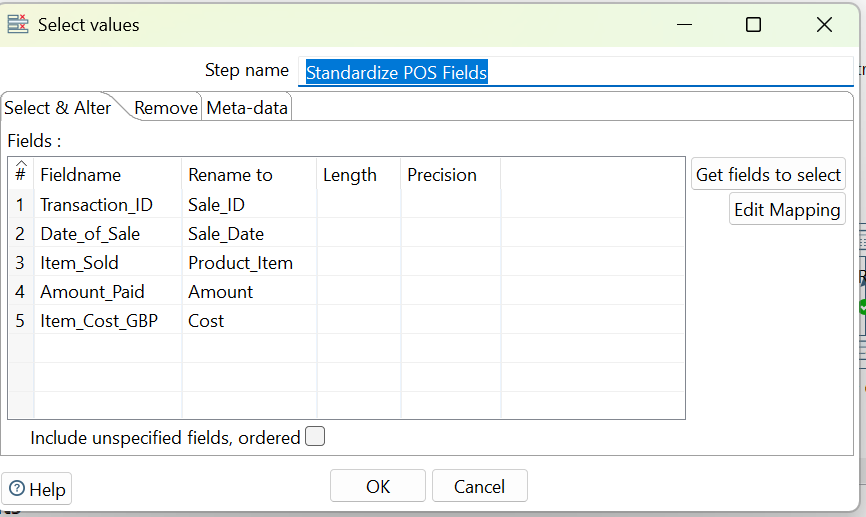
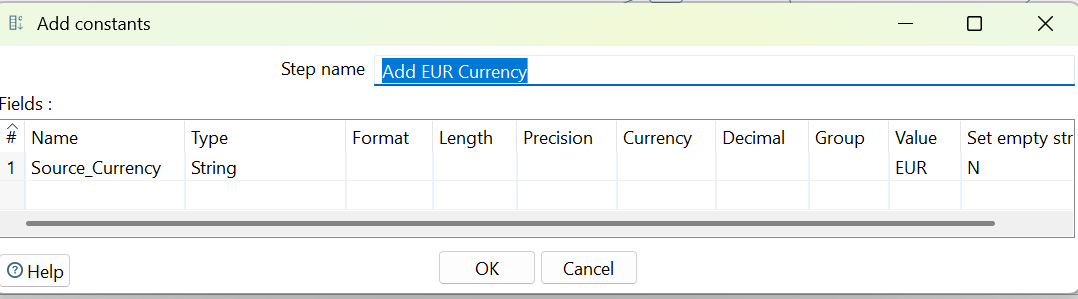
In Pentaho UI , follow this.  
============== Pentaho UI Workflow / Transformation creation steps ================  
  
File 🡪 New 🡪 Transformations (Save the transformation immediately)  
Design 🡪 Input 🡪 Table input 🡪 New



Connection type : Generic database  
Custom connection URL: jdbc:sqlite:D:/sqlite/pos\_data.sqlite  
Custum driver class name : org.sqlite.JDBC  
Click test to test connection

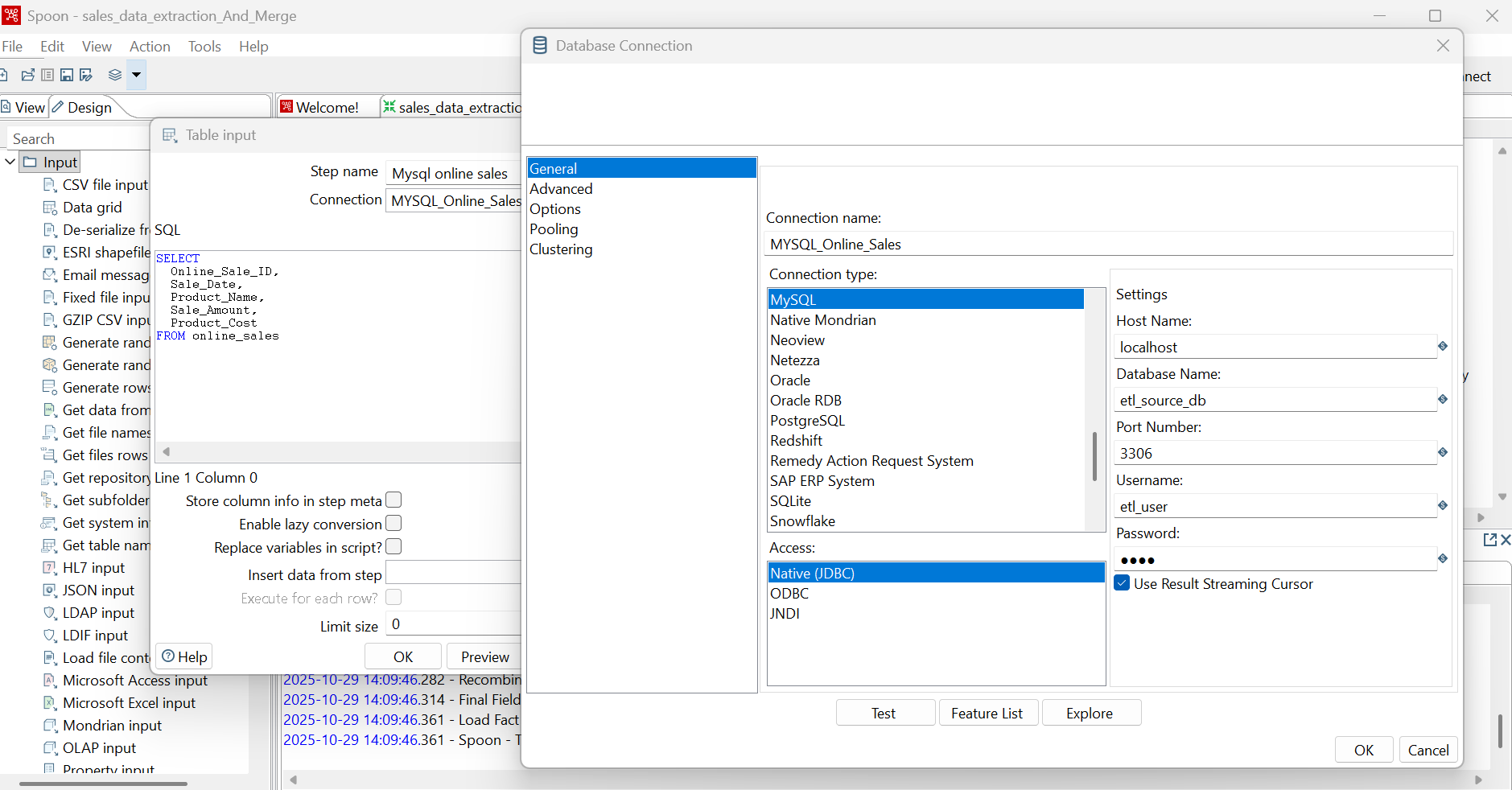
**1. Phase 1: Extraction & Unification**

**A. POS Sales Stream (SQLite/GBP)**

1. **Extract POS Sales (LiteSQL) (Table Input)**
   * **Double-click** the step.
   * Select your **SQLite database connection**.
   * Enter your SELECT query (e.g., SELECT Sale\_ID, Sale\_Date, Product\_Item, Amount, Cost FROM pos\_sales;).
2. **Convert POS Date to Date Type (Select Values)**
   * **Meta-data Tab:** Click **Get fields to change**.
   * Find the **Sale\_Date** field.
   * Change the **Type** column value to **Date**. *This is critical for joining.*
3. **Standardize POS Fields (Select Values)**
   * **Select & Alter Tab:** Click **Get fields**.
   * Ensure the fields are present in the order they will be merged (e.g., Sale\_ID, Sale\_Date, Product\_Item, Amount, Cost).  
     
4. **Add GBP Currency (Add Constants)**
   * **Add Field:**
     + **Name:** Source\_Currency
     + **Type:** String
     + **Value:** GBP  
       

**B. Online Sales Stream (MySQL/EUR)**Important : Copy mysql-connector-j-9.5.0.jar to   
**D:\pentaho\lib\** **mysql-connector-j-9.5.0.jar**

|  |
| --- |
| **mysql details https://dev.mysql.com/downloads/installer/ (mysql-installer-community-8.0.44.0.msi)**  Database etl\_source\_db  user : etl\_user password : abcd  mysql shell --> from windows programs (after installing mysql )  MySQL JS > \connect root@localhost <no password set>  MySQL localhost:33060+ ssl JS > \sql  SELECT host, user, authentication\_string FROM mysql.user WHERE user = 'etl\_user';  MySQL JS > \connect root@localhost  <no password set>  MySQL localhost:33060+ ssl JS > \sql MySQL localhost:33060+ ssl SQL > show databases; MySQL localhost:33060+ ssl SQL > CREATE DATABASE etl\_source\_db; MySQL localhost:33060+ ssl SQL > use etl\_source\_db MySQL localhost:33060+ ssl SQL > show tables; MySQL localhost:33060+ ssl SQL > CREATE USER 'etl\_user'@'localhost' IDENTIFIED BY 'abcd'; MySQL localhost:33060+ ssl SQL > GRANT SELECT, INSERT, UPDATE, DELETE, CREATE ON etl\_source\_db.\* TO 'etl\_user'@'localhost'; MySQL localhost:33060+ ssl SQL > FLUSH PRIVILEGES;  MySQL localhost:33060+ ssl SQL > CREATE TABLE test\_online\_sales (  Online\_Sale\_ID VARCHAR(10) PRIMARY KEY,  Customer\_ID VARCHAR(10),  Sale\_Date DATE,  Product\_Name VARCHAR(50),  Sale\_Amount DECIMAL(10, 2), -- In EUR  Product\_Cost DECIMAL(10, 2) -- In EUR  );  **Populating Sample Data**  -- Insert Command 1 (Sale on 2025-10-25)  INSERT INTO online\_sales (Online\_Sale\_ID, Customer\_ID, Sale\_Date, Product\_Name, Sale\_Amount, Product\_Cost)  VALUES ('ON1001', 'CUST001', '2025-10-25', 'Widget A', 150.00, 80.00);  -- Insert Command 2 (Sale on 2025-10-25)  INSERT INTO online\_sales (Online\_Sale\_ID, Customer\_ID, Sale\_Date, Product\_Name, Sale\_Amount, Product\_Cost)  VALUES ('ON1002', 'CUST002', '2025-10-25', 'Gadget B', 50.00, 20.00);  -- Insert Command 3 (Sale on 2025-10-26)  INSERT INTO online\_sales (Online\_Sale\_ID, Customer\_ID, Sale\_Date, Product\_Name, Sale\_Amount, Product\_Cost)  VALUES ('ON1003', 'CUST003', '2025-10-26', 'Widget A', 150.00, 80.00);  **Create final FACT table.**  CREATE TABLE IF NOT EXISTS fact\_profitability (Profit\_Key INT PRIMARY KEY AUTO\_INCREMENT,Sale\_ID VARCHAR(20),Sale\_Date DATE,Source\_Currency VARCHAR(10), Product\_Item VARCHAR(100),Sale\_Amount\_USD DECIMAL(10, 2),Profit\_USD DECIMAL(10, 2),Currency\_Rate\_Used DECIMAL(5, 4) ); |

Design 🡪 Input 🡪 Table input 🡪 New ****

Connection type  
MySql  
Host  
localhost  
Access:  
Native(JDBC)

1. **Mysql online sales (Table Input)**
   * **Double-click** the step.
   * Select your **MySQL database connection**.
   * Enter your SELECT query (e.g., SELECT Sale\_ID, Sale\_Date, Product\_Item, Amount, Cost FROM online\_sales;).
2. **Standardize Online Fields (Select Values)**
   * Use the **Select & Alter Tab** to ensure the field names, types, and order exactly match the POS stream's fields (Step 3).
3. **Add EUR Currency (Add Constants)**
   * **Add Field:**
     + **Name:** Source\_Currency
     + **Type:** String
     + **Value:** EUR
4. **Combine Sales Data (Append Streams)**
   * Connect the output of **Add GBP Currency** (First Stream) and **Add EUR Currency** (Second Stream) to this step.
   * No configuration is needed if the field metadata matches.

**2. Phase 2: Enrichment & Cleansing (The Fixes)**

**A. Market Rates Cleansing Path (The Lookup Stream)**

1. **Load Market Rates (CSV) (CSV File Input)**
   * **Content Tab:** Specify the file path for daily\_rates.txt.
   * **Fields Tab:** Click **Get Fields**.
   * **CRITICAL FIX:** For the EUR\_to\_USD and GBP\_to\_USD fields, manually change the **Type** column value to **String**. *This prevents the step from failing on the "Error" text.*
2. **Convert Rates to String (Select Values)**
   * **Meta-data Tab:** Confirm the GBP\_to\_USD field's Type is **String**.
3. **Cleanse GBP Rate (Replace in string)**
   * **CRITICAL FIX:** Configure the step to replace the error text:
     + **In stream field:** GBP\_to\_USD
     + **Out stream field:** GBP\_to\_USD
     + **use RegEx:** Y
     + **Search:** ^.\*Error.\*$ (This matches "Error" regardless of hidden spaces or characters).
     + **Replace with:** 1.20 (A valid numeric string for the fallback rate).
4. **Convert Rates to Number (Select Values)**
   * **CRITICAL FIX:** **Meta-data Tab:**
     + Find EUR\_to\_USD and change **Type** to **Number**.
     + Find GBP\_to\_USD and change **Type** to **Number**.
   * *This completes the lookup data preparation.*

**B. Order Control and Lookup**

1. **Wait for Rates Load (Block this step until steps finish)**
   * **Draw Hop:** Draw a hop from **Combine Sales Data** to this step (This is the **Main Stream**).
   * **Configuration:** Click **Get steps**.
     + Ensure the **only** step listed is **Convert Rates to Number**. *This forces the lookup data to finish loading before the sales data proceeds.*
2. **Lookup Exchange Rate (Stream Lookup)**
   * **Draw Hops (Order is Critical!):**
     + **Hop 1 (Lookup Stream):** From **Convert Rates to Number** to **Lookup Exchange Rate**. (Draw this first).
     + **Hop 2 (Main Stream):** From **Wait for Rates Load** to **Lookup Exchange Rate**.
   * **Configuration:**
     + **Lookup step:** Select **Convert Rates to Number**.
     + **Key(s) to look up the value(s):**
       - **Field (Main Stream):** Sale\_Date
       - **LookupField (Lookup Stream):** market\_date
     + **Specify the fields to retrieve:** Click **Get lookup fields** and select **EUR\_to\_USD** and **GBP\_to\_USD**.

**3. Phase 3: Transformation & Load**

1. **Filter by Currency (Filter Rows)**
   * **Condition:** Source\_Currency = EUR (Select the **(String)** option).
   * **True Destination:** **Calc EUR to USD**
   * **False Destination:** **Calc GBP to USD**
2. **Calc EUR to USD / Calc GBP to USD (Calculator)**
   * **New field:** Sale\_Amount\_USD
   * **Calculation:** A \* B
   * **Field A:** Amount
   * **Field B (EUR):** EUR\_to\_USD
   * **Field B (GBP):** GBP\_to\_USD
3. **Recombine Calculated Data (Append Streams)**
   * Merge the outputs of the two Calc... steps.
4. **Final Field Selection (Select Values)**
   * **Select & Alter Tab:** **Rename:** Find **Source\_Currency** and set **Rename to** as **Source\_System** (if that's the name in your MySQL table).
   * **Remove Tab:** Click **Get fields to remove** and select the now-redundant fields: EUR\_to\_USD and GBP\_to\_USD.
5. **Load Fact Profitability (Table Output)**
   * **Configuration:** Select your **MySQL database connection** and the target table, fact\_profitability.
   * **Database Fields Tab:** **CRITICAL FINAL FIX:** Click **Get fields**. **Manually verify** that the column names in the **Table field** column exactly match the names in the **Stream field** column (e.g., ensure Source\_System in the table maps to the renamed field from Step 18).