**Data Migration and SCD Implementation in Azure SQL Database**

**Trainee Name:** Nachiket Prajapati

**Batch:** C-19

### **1. Introduction**

I recently wrapped up a project focused on migrating data from an on-premises database to Azure SQL. During the process, I set up five tables and applied Slowly Changing Dimensions (SCD) Types 1 and 2 logic to two of them, ensuring data integrity as it moved to the cloud. To keep everything clear and well-documented, I’ve included screenshots of each step along the way.

### **2. Project Goals**

* Migrated data seamlessly from an on-premises database to ADLS Gen 2.
* Implemented Slowly Changing Dimensions (SCD) logic to load data from ADLS Gen 2 into Azure SQL Database.
* Conducted thorough validation to ensure accurate data migration and SCD logic in the Azure SQL Database.

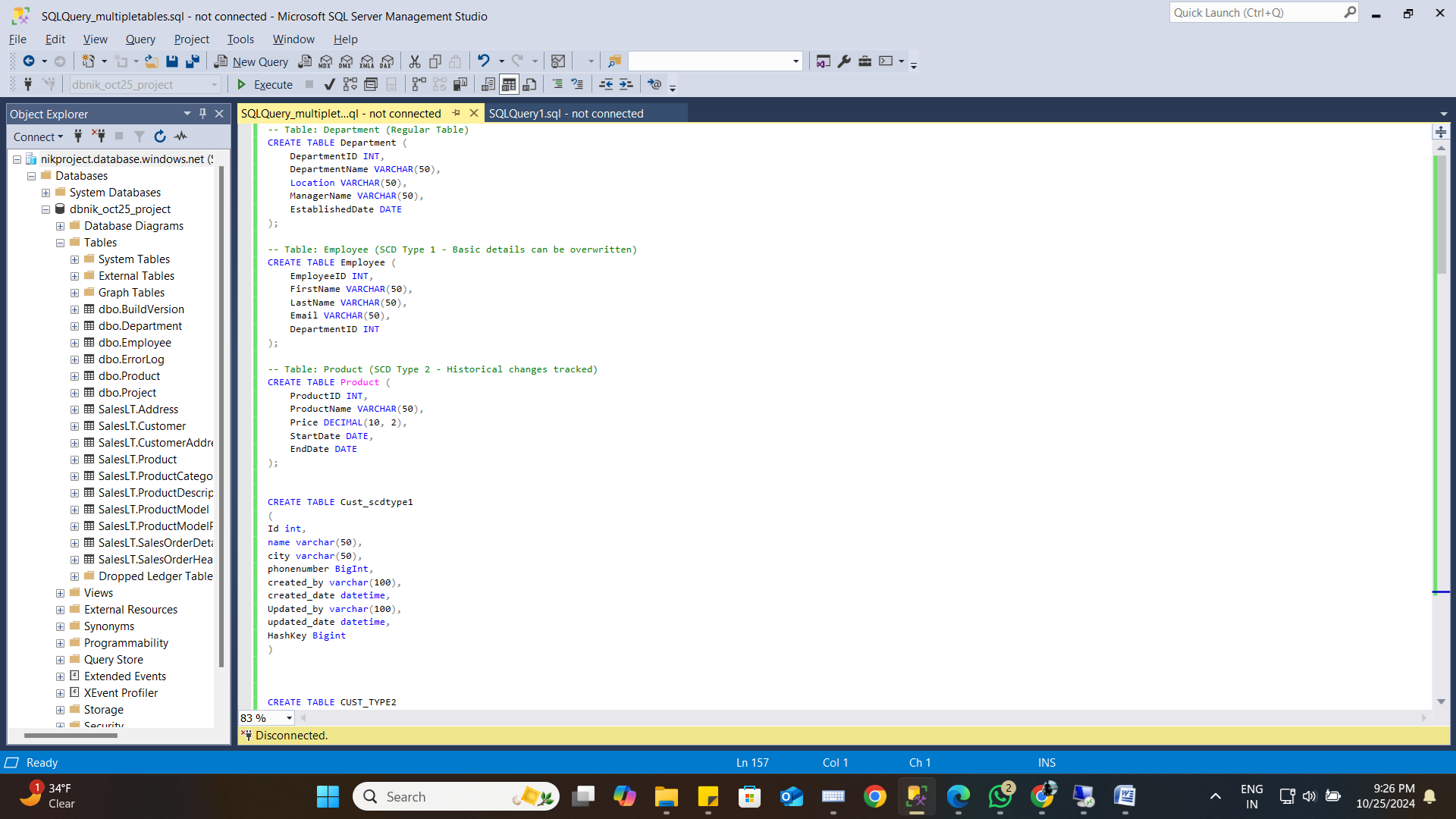
### **3. Prerequisites**

* Checked that I had proper access to the on-premises database, ADLS Gen 2, and Azure SQL databases, making sure I could connect without issues.
* Set up Azure Synapse Analytics to handle data migration and manage transformations.
* Gained practical experience with SQL, Slowly Changing Dimensions (SCD), and creating data flows in Synapse, building my skills throughout the project.

### **4. Steps to Follow**

#### **4.1 Table Creation**

* The **Department** table is implemented with SCD Type 0 logic, performing a full load without tracking changes.
* The **Employee** table uses SCD Type 1 logic, where basic details such as name and department can be overwritten as updates occur.
* The **Product** table is implemented with SCD Type 2 logic, allowing historical changes to be tracked by adding a new row whenever a record changes, while also capturing the start and end dates.
* The **Cust\_scdtype1** table applies SCD Type 1 logic for simple updates that overwrite existing data.
* The **Cust\_TYPE2** table is configured with SCD Type 2 logic to maintain historical records by adding new rows for each change, ensuring that past data remains accessible.

Tables:- 

#### **4.2 Data Migration to ADLS Gen 2**

* Customer
* CustomerType1
* CustomerType2
* Orders
* product

#### **4.3 Data Transformation and SCD Logic in Synapse**

* In Synapse, I set up data flows that apply Slowly Changing Dimension (SCD) logic, using customized transformations to track historical data and manage record updates effectively. I documented the configuration process with screenshots, which show each transformation step applied before the data was loaded into the Azure SQL Database. These transformations ensure the data's accuracy and historical integrity throughout the pipeline.

#### **4.4 Data Loading to Azure SQL Database**

* After applying SCD transformations, I loaded the final transformed data into the Azure SQL Database, ensuring that each table was stored persistently. Screenshots document the data loading process for verification, capturing each step in detail.

### **5. Validation**

I validated the data migration and SCD logic implementation by running relevant queries, confirming the integrity of the data in the Azure SQL Database. Screenshots of the validation queries and results are also included.

### **6. Deliverables**

* **SQL Table Creation Scripts**: Developed scripts for setting up five specific tables in Azure SQL Database to meet project requirements.
* **Data Migration Pipeline Configuration**: Configured an Azure Data Factory (ADF) pipeline, detailing the source, destination, and transformation processes for seamless data transfer.
* **SCD Logic Implementation**: Implemented SCD Type 0, Type 1, and Type 2 logic, with SQL scripts and proof of each type’s functionality in preserving historical and updated data accurately.
* **Final Data Verification in Azure SQL Database**: Ensured that all required data was successfully migrated and loaded into Azure SQL Database, aligning with the project’s objectives.
* **Comprehensive Documentation with Screenshots**: Included screenshots for each phase of the project, from initial table setup to final validation, capturing the complete process for reference.

### **7. Output Format**

For each step, the following outputs should be submitted:

#### **7.1 Table Creation**

#### **sql script_1.png**

#### **sql script_2.png**

#### **sql script_3.png**

#### **7.2 Data Migration**

#### **pipeline succeeded till incremental tables.png**

#### **7.3 SCD Type 1 Implementation**

#### **data flow for scd type 1.png**

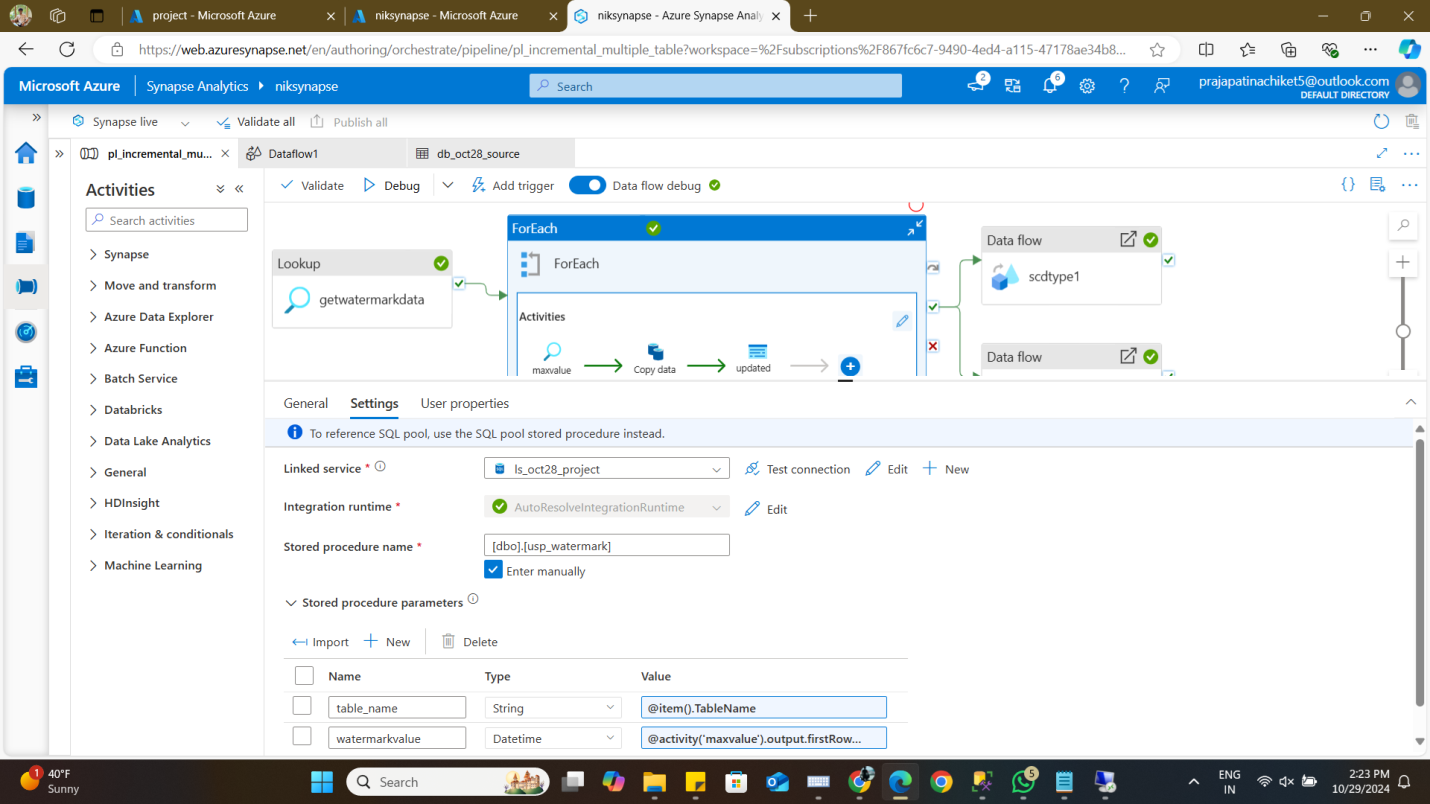
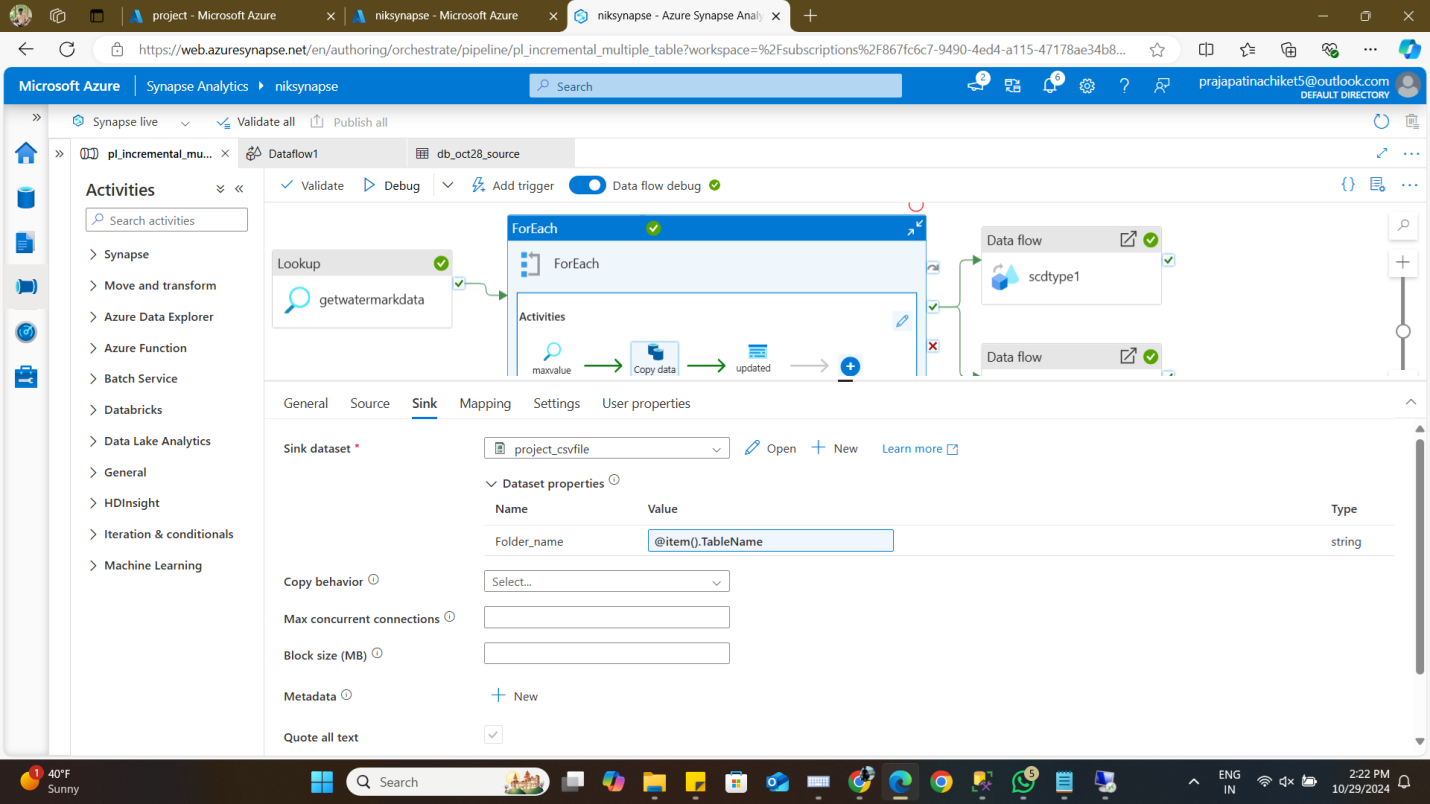
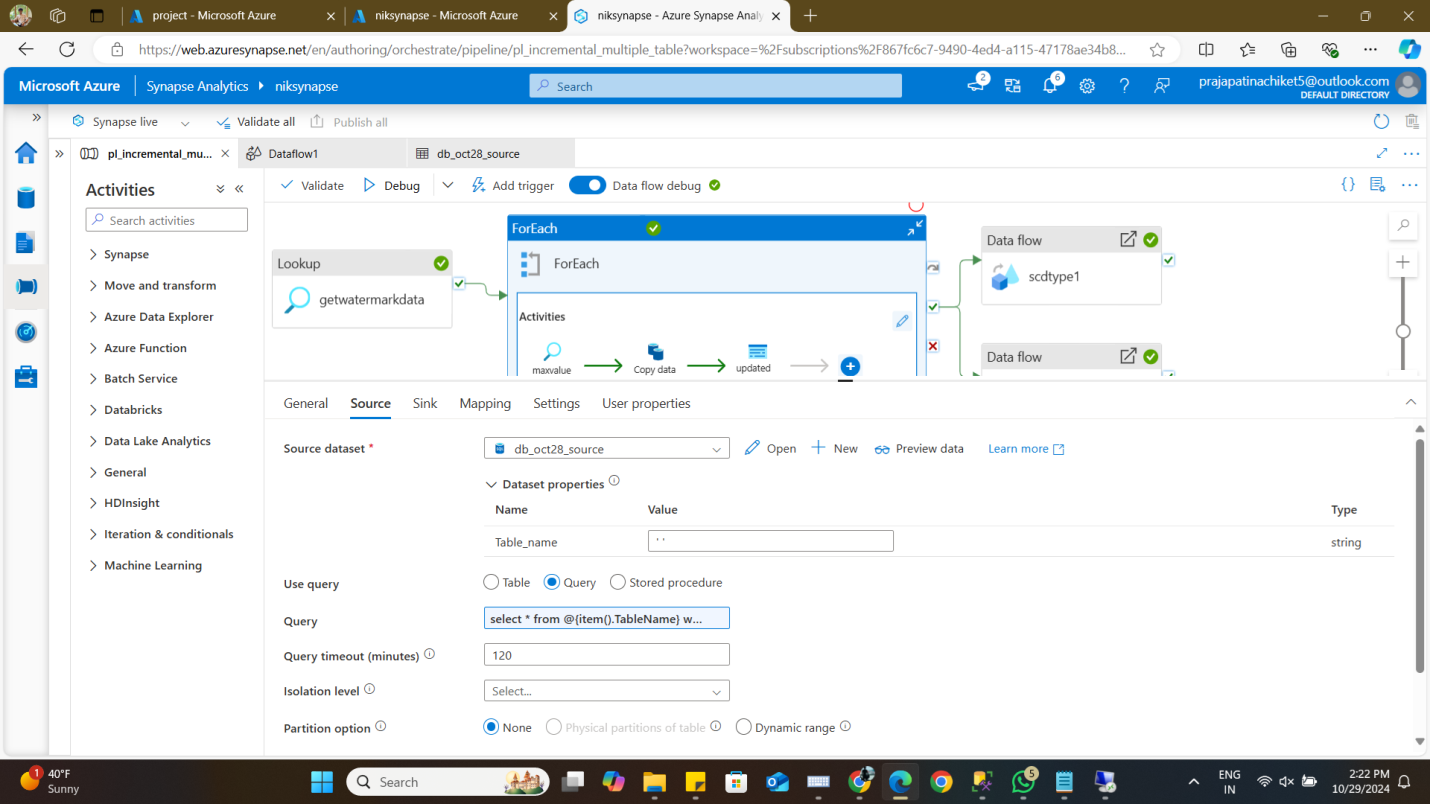
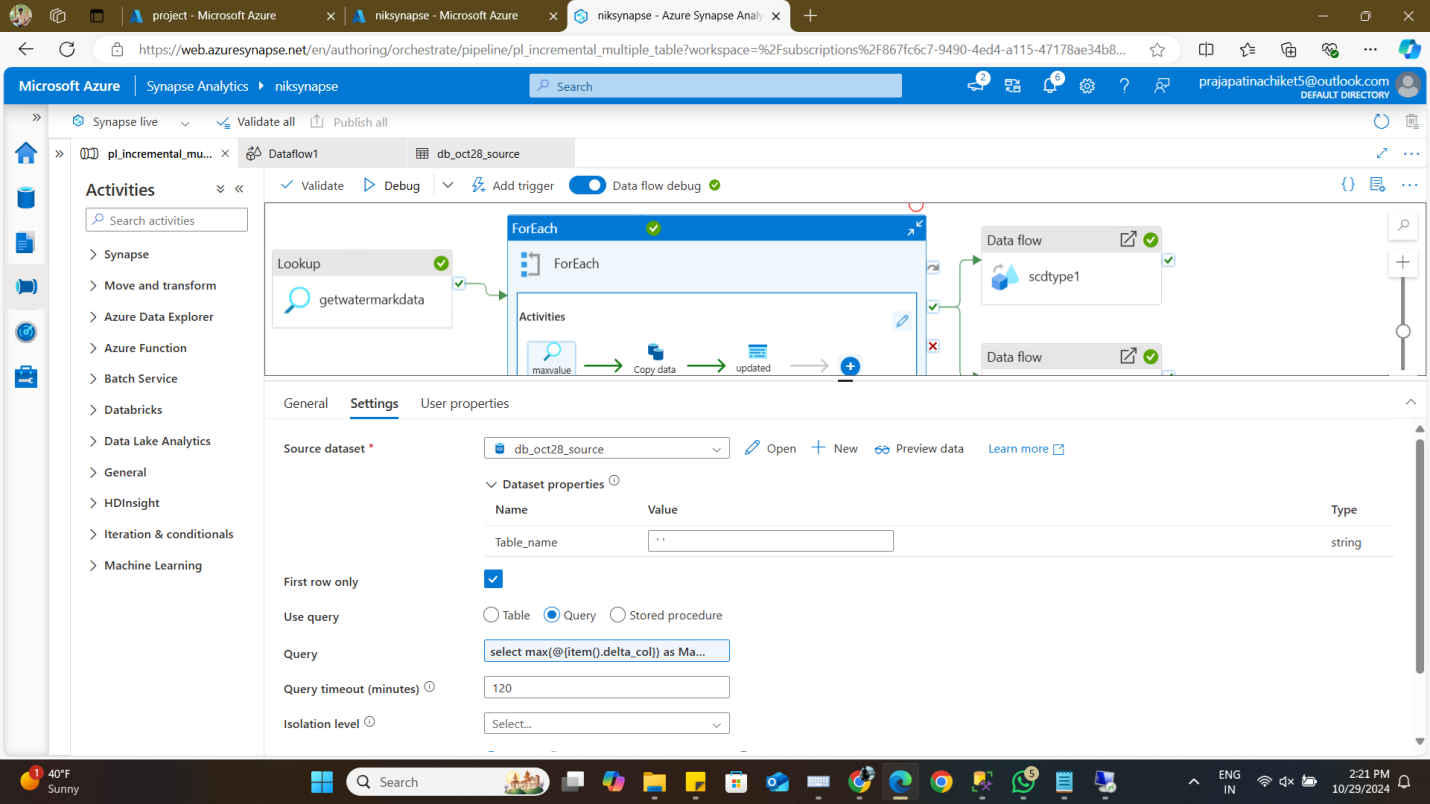
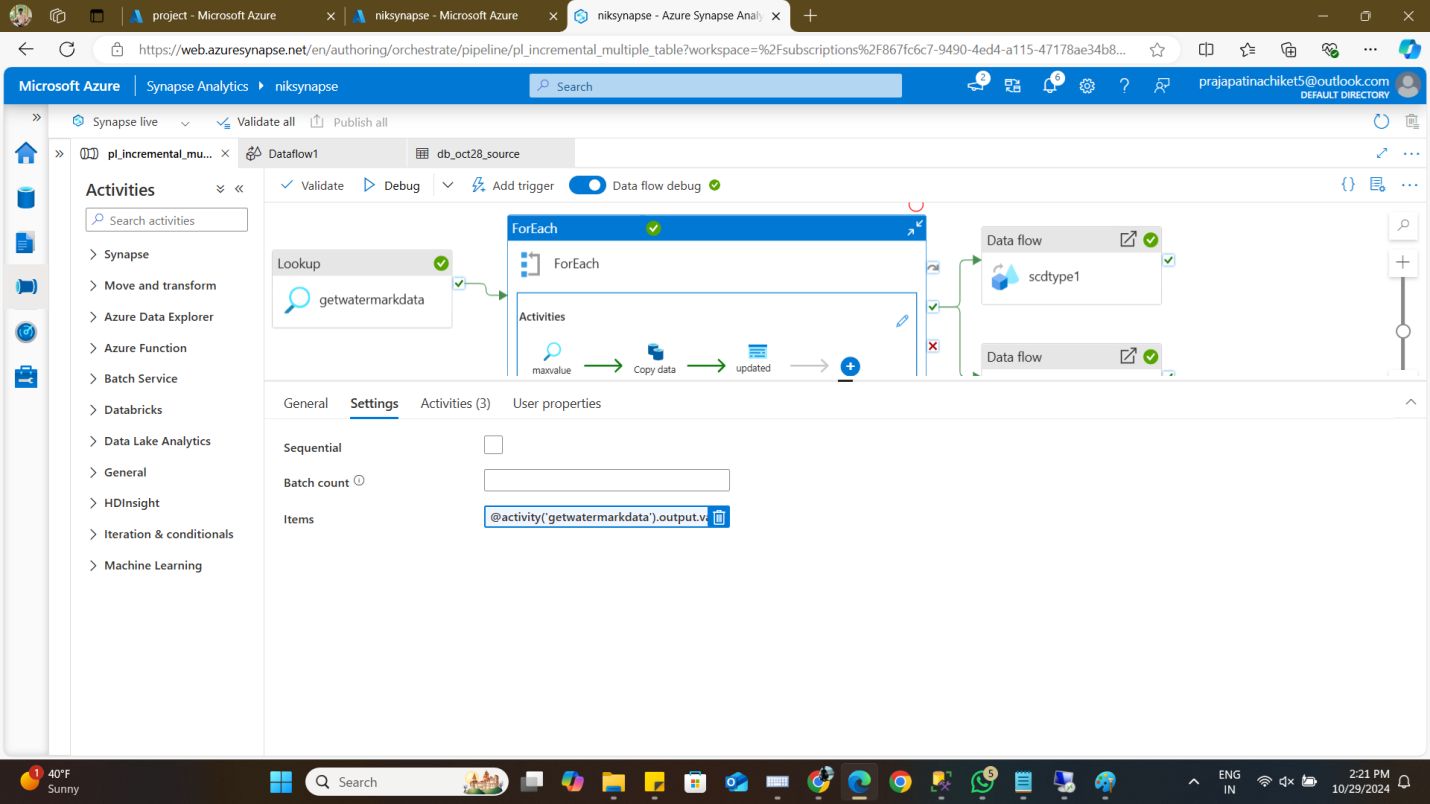
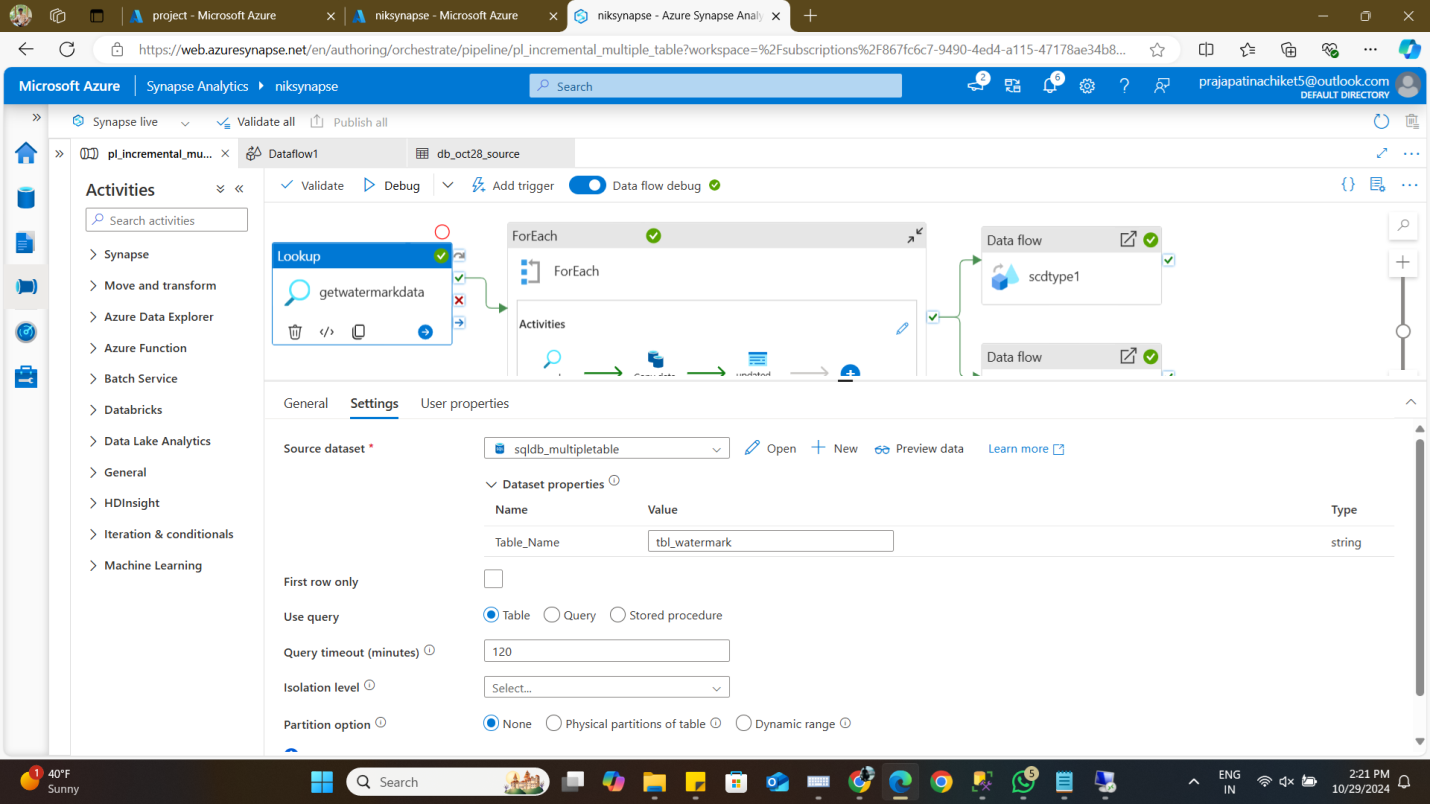
#### **7.4 SCD Type 2 Implementation**

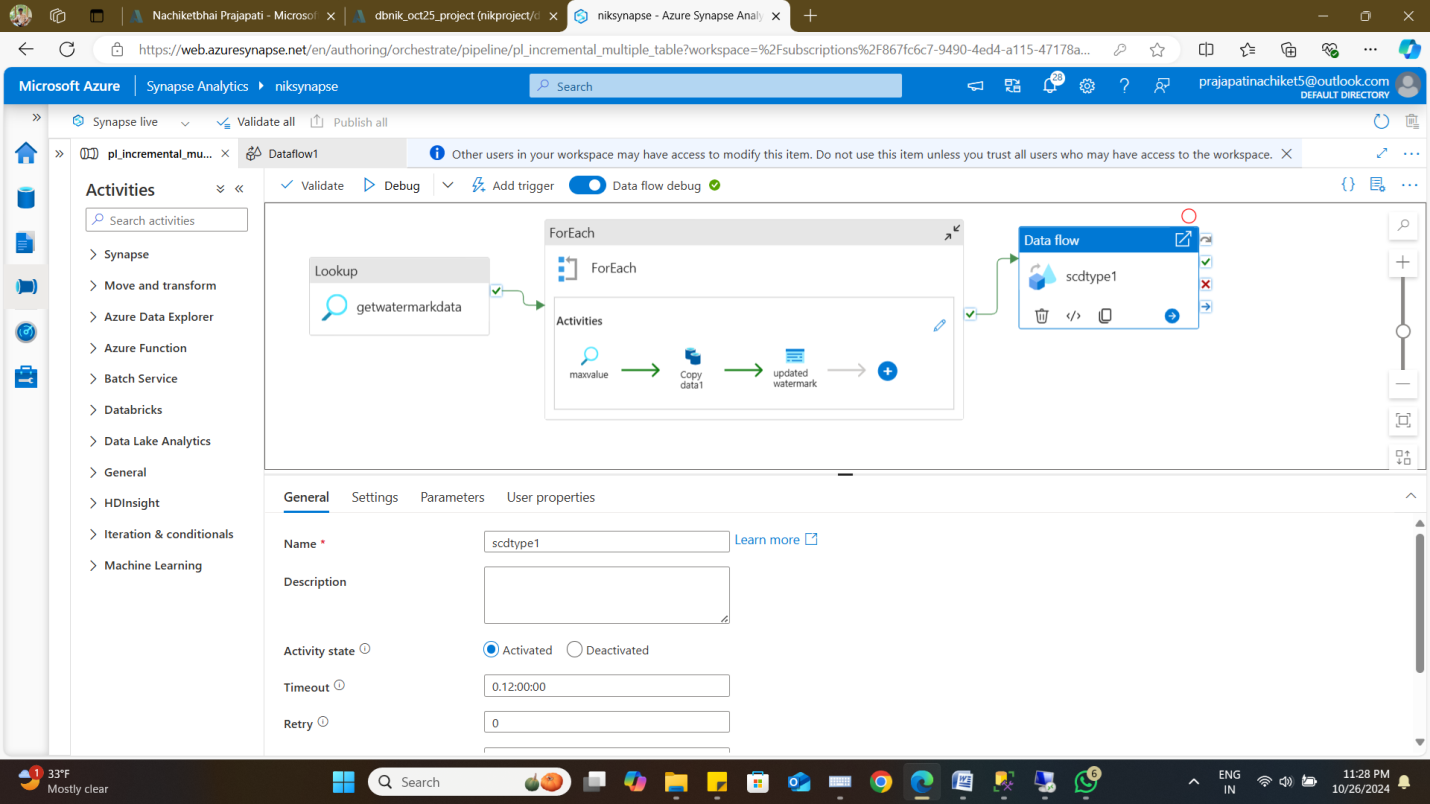
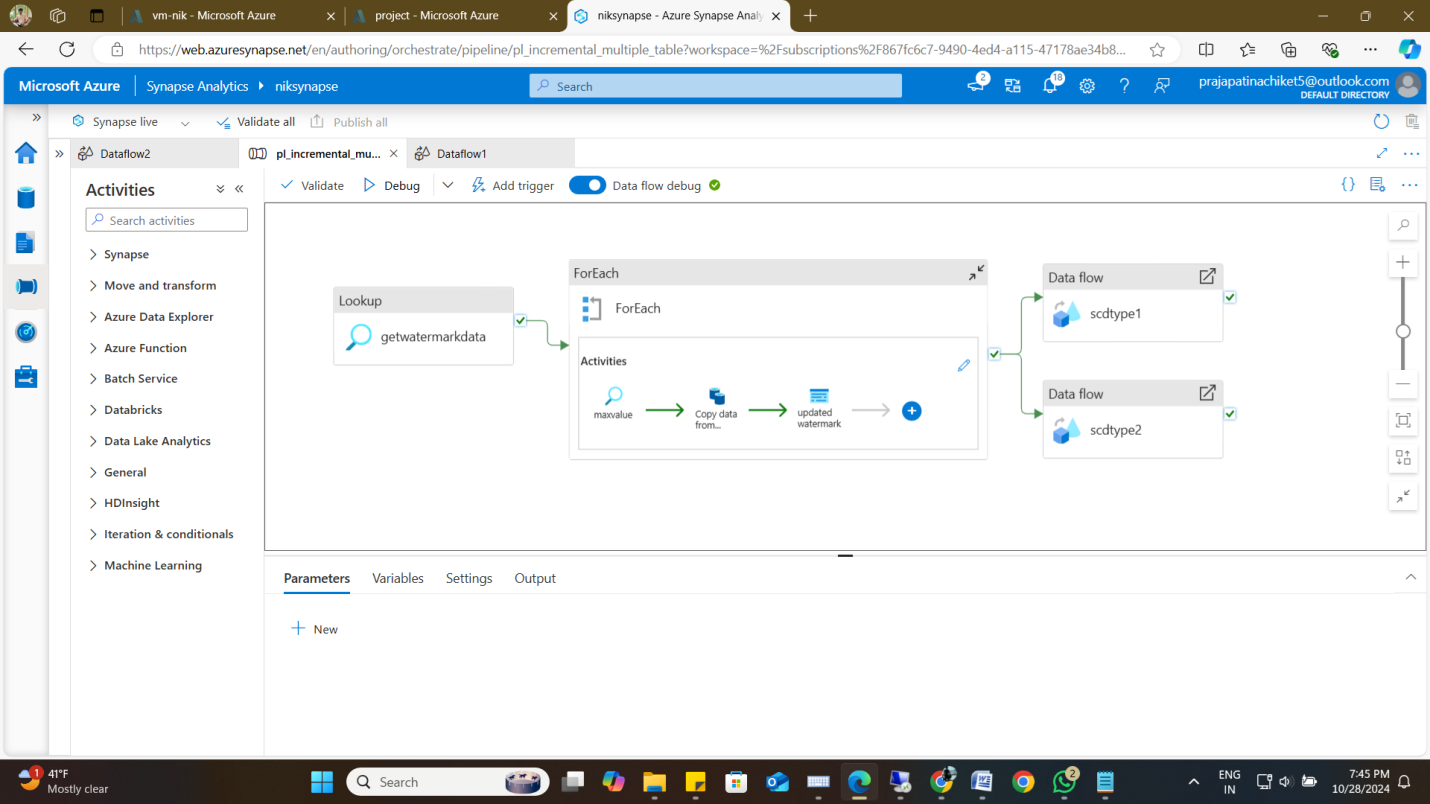
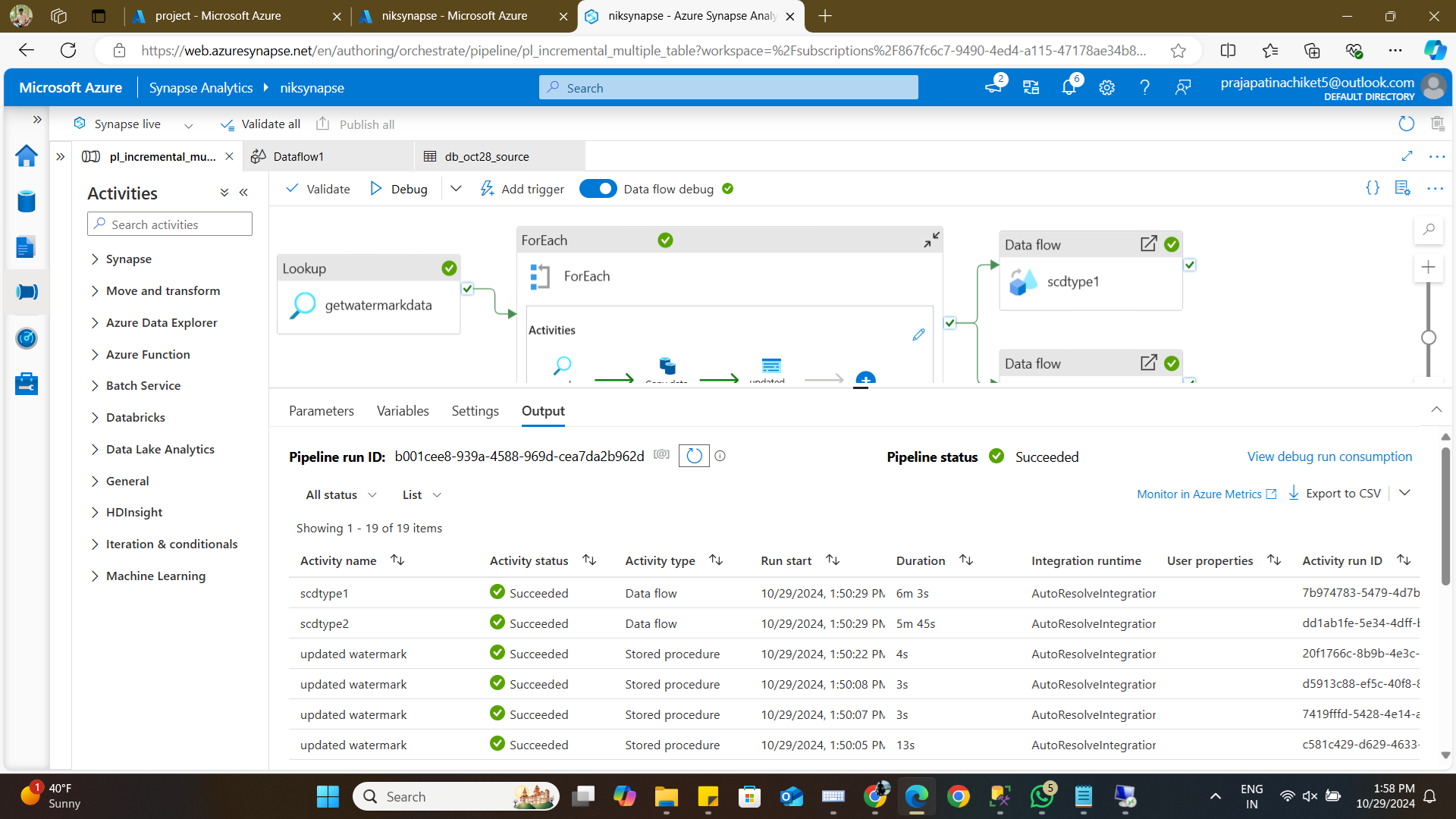
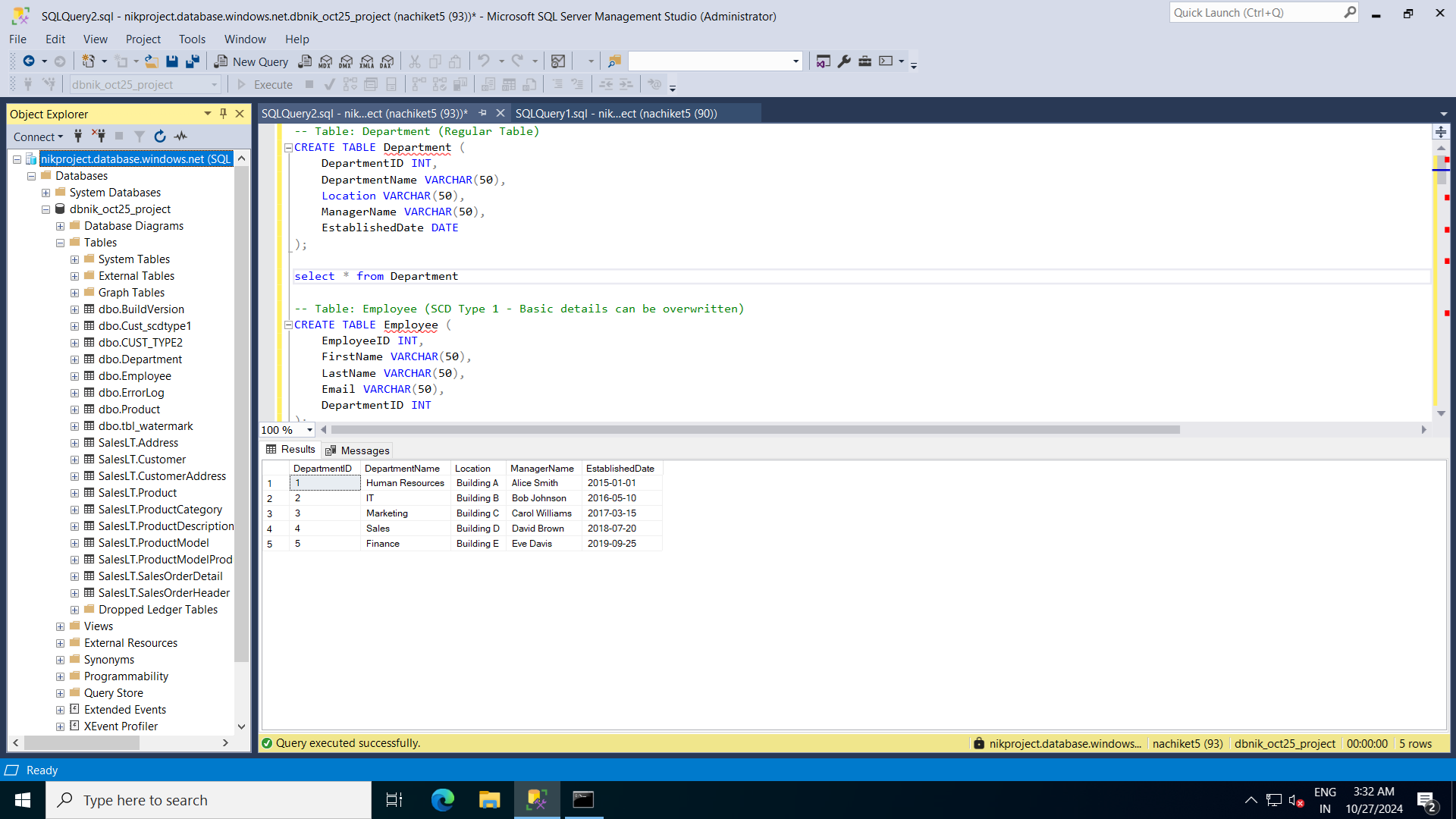
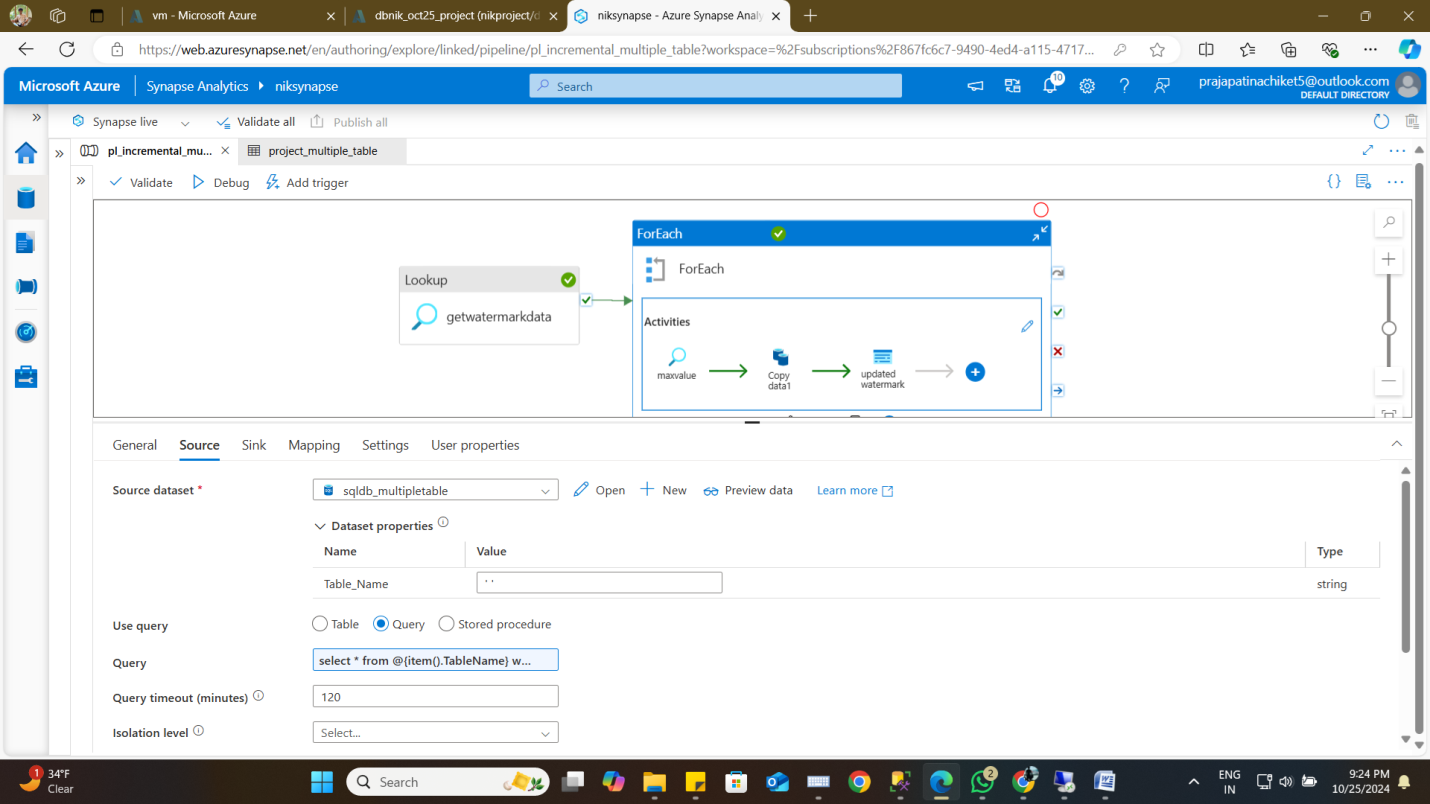
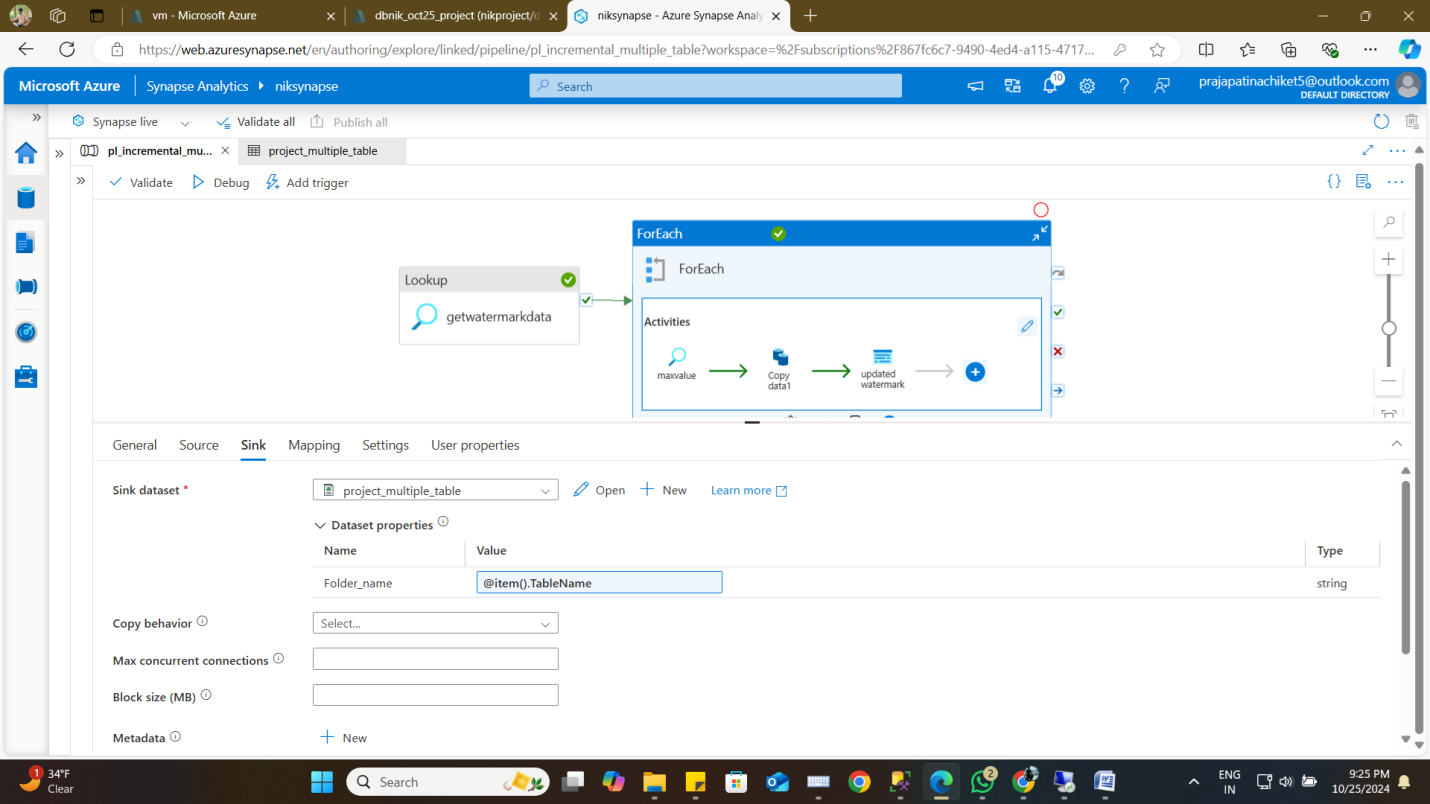
#### **data flow for scd type 2.png**

#### **7.5 Data Loading**

#### **after incremental load filed created.pngafter incremental load folder created.png**

#### **7.6 Validation**





### **8. Conclusion**

Once the project was finalized, I carefully examined each component to verify that data migration, SCD implementations, and data loading were accurately carried out. Step-by-step screenshots were included to document and confirm the success and precision of each phase.