



**Sri Lanka Institute of Information Technology**

**IT3021 - Data warehousing and  
Business Intelligence**

**Assignment 01**

**IT22118936**

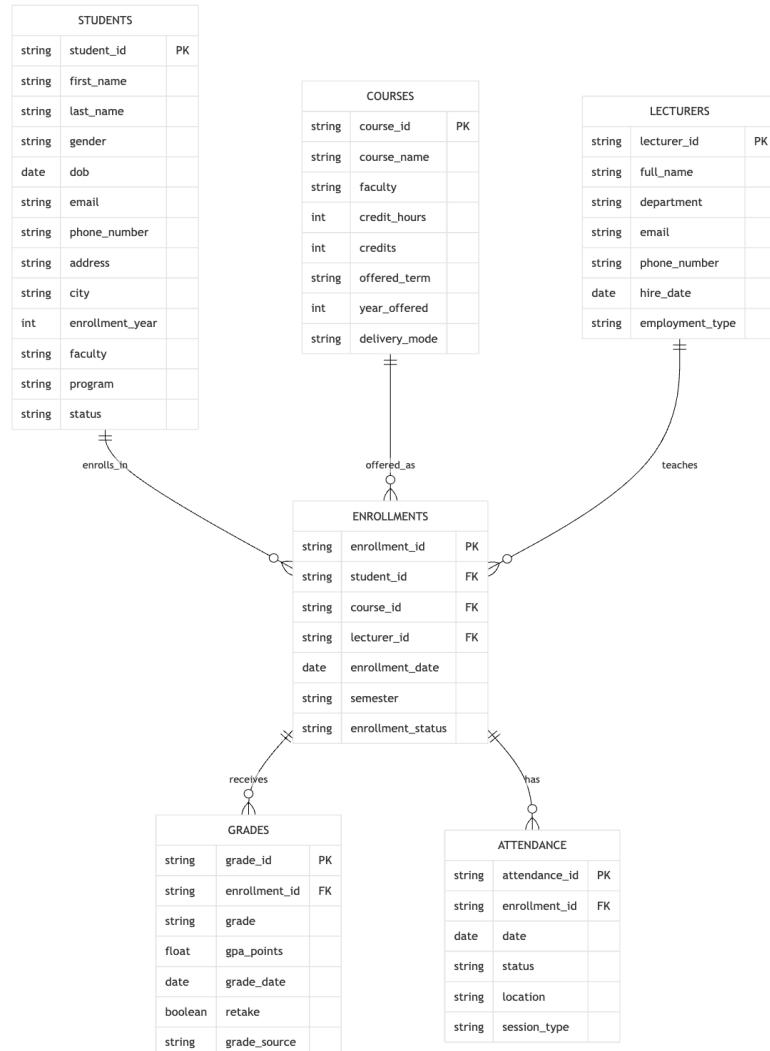
**K.A.E.H.N.Samarasinghe**

## Step 1: Data set selection

### Dataset

The dataset represents a **university academic performance system** used by *TechVerse University*. It tracks student enrollments, attendance, grades, lecturers, and course data. This data is gathered to perform analytical tasks like:

- Tracking student performance by term, course, and lecturer
- Identifying dropout patterns.
- Analyzing attendance vs GPA.
- Visualizing academic KPIs.

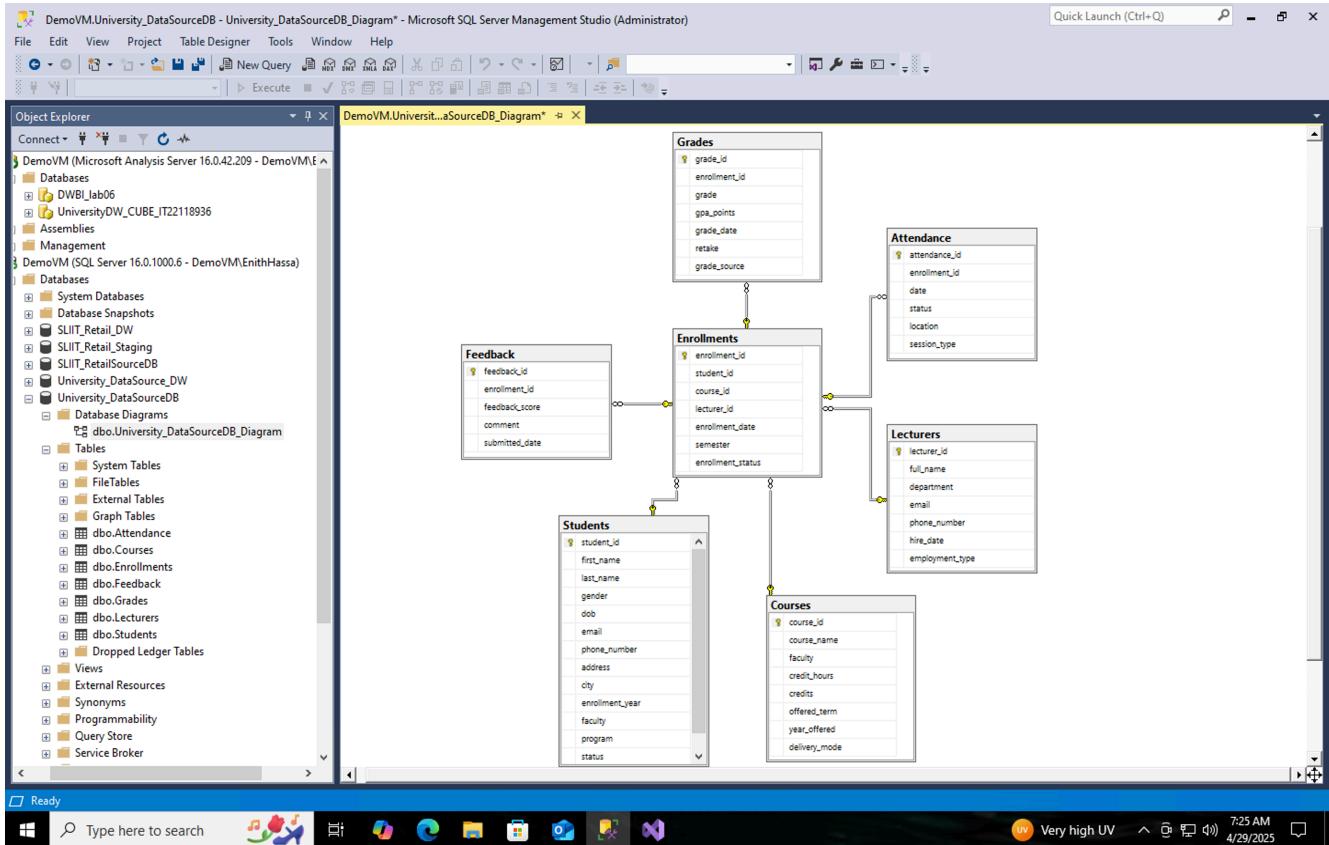


The dataset includes over 3000 records generated through synthetic data and was uploaded into SQL Server Management Studio (SSMS).

[View Dataset From Here](#)

## Step 2: Preparation of data sources

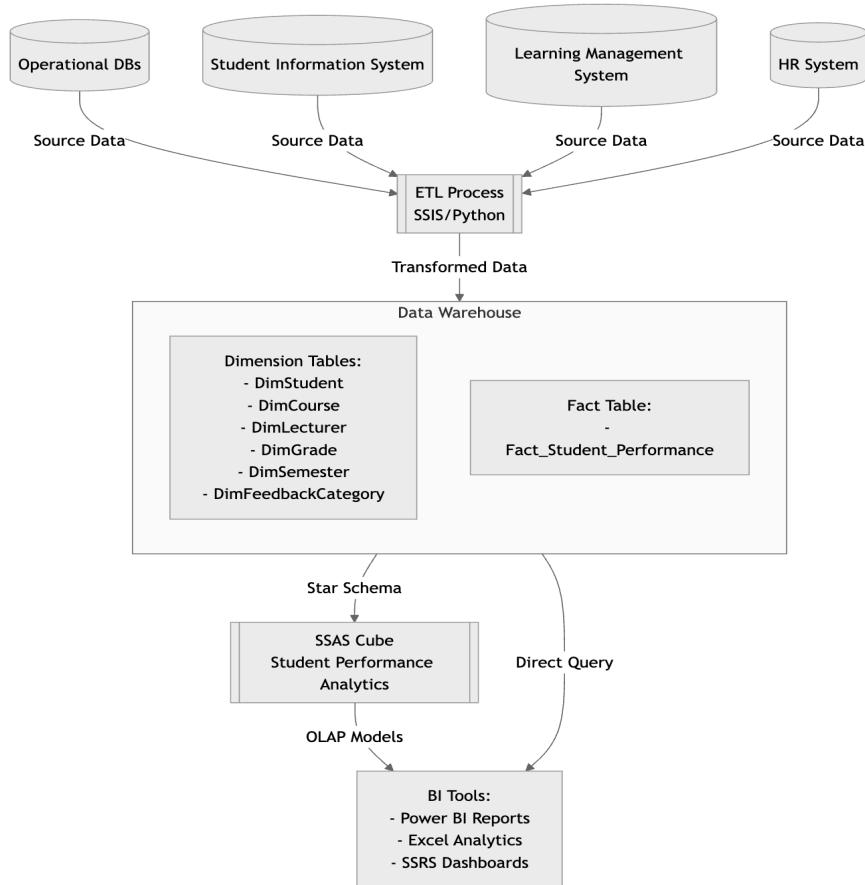
### Database schema



- \* Creating the University\_DataSourceDB database and importing the data files.
- \* Uploading .csv, .txt files using the Import Flat File method, and .xlsx files using the Import Data method.  
.CSV Files,
  - Students → student\_id, first\_name, last\_name, gender, dob, email, phone\_number, address, city, enrollment\_year, faculty, program, status.
  - Courses → course\_id, course\_name, faculty, credit\_hours, offered\_term, year\_offered, delivery\_mode.
  - Lecturers → lecturer\_id, full\_name, department, email, phone\_number, hire\_date, employment\_type.
  - Enrollments → enrollment\_id, student\_id, course\_id, lecturer\_id, enrollment\_date, semester, enrollment\_status.
  - Grades → grade\_id, enrollment\_id, grade, gpa\_points, grade\_date, retake, grade\_source.
- .xlsx Files,
  - Attendance → attendance\_id, enrollment\_id, date, location, session\_type.
  - Feedback → feedback\_id, enrollment\_id, feedback\_score, comment, submitted\_date

## Step 3: Solution architecture

## Step 4: Data warehouse design & development



## Data warehouse schema

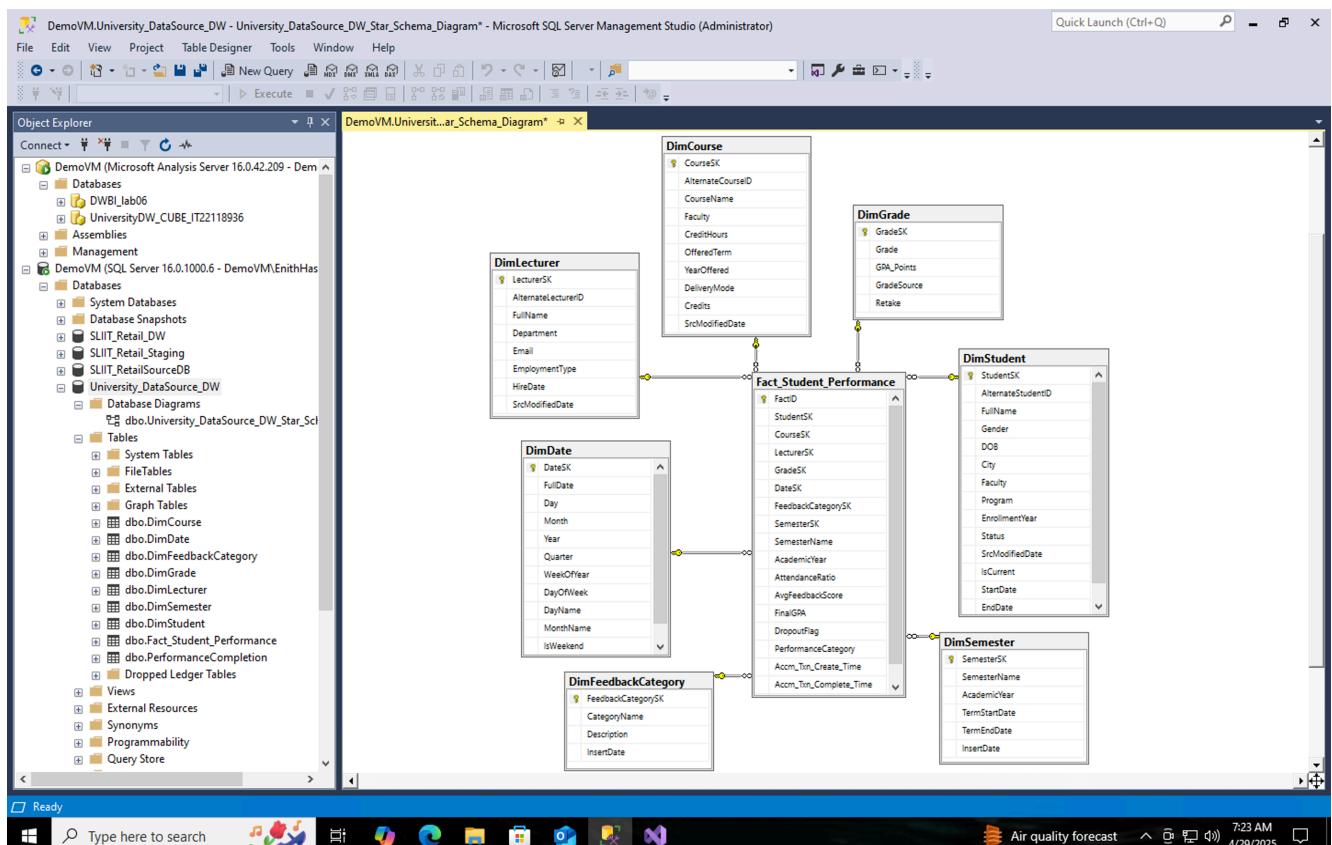
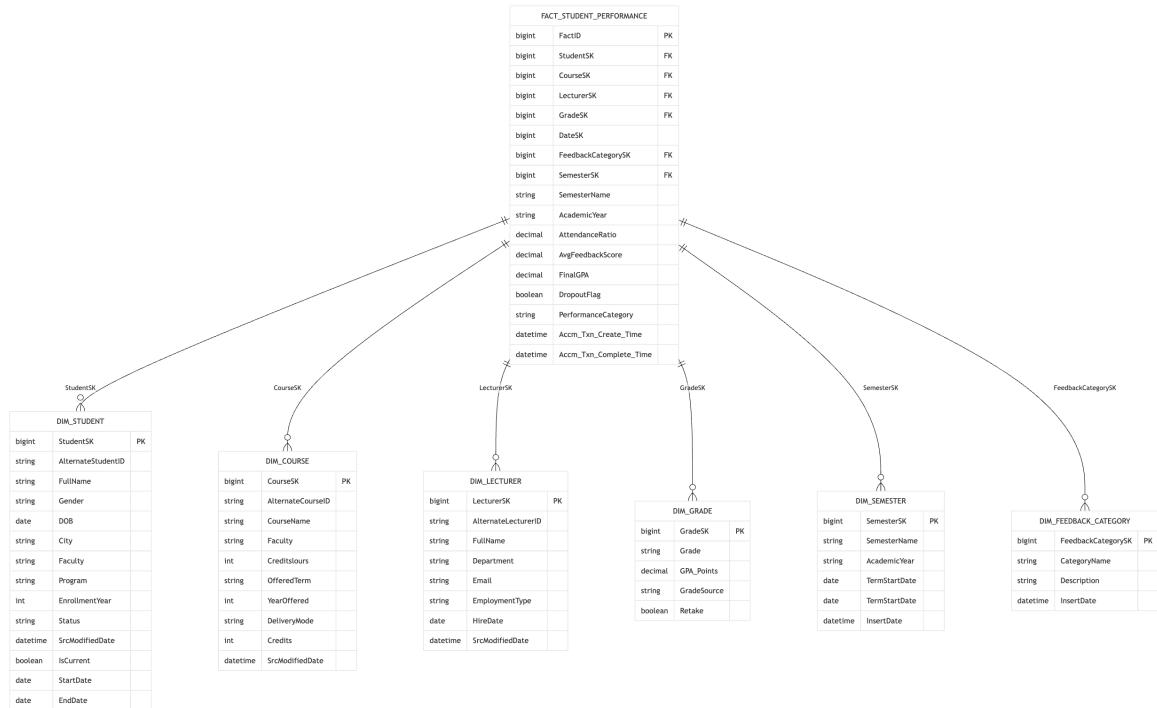
- \* Creating the University\\_DataSource\\_DW database and creating the dimension tables.  
Design a Star Schema with:
  1. DimStudent (Slowly Changing Dimension Type 2)  
(if a student's City, Status, or Program changes over time.)
  2. DimCourse
  3. DimLecturer
  4. DimGrade
  5. DimDate
  6. DimSemester
  7. DimFeedbackCategory
  8. Fact\_Student\_Performance (Fact table)
    - o Added Surrogate Keys (SK) to every Dim table.
    - o Created alternate IDs to maintain connection with Source Data.

[View All the Queries From here](#)

# ETL Design

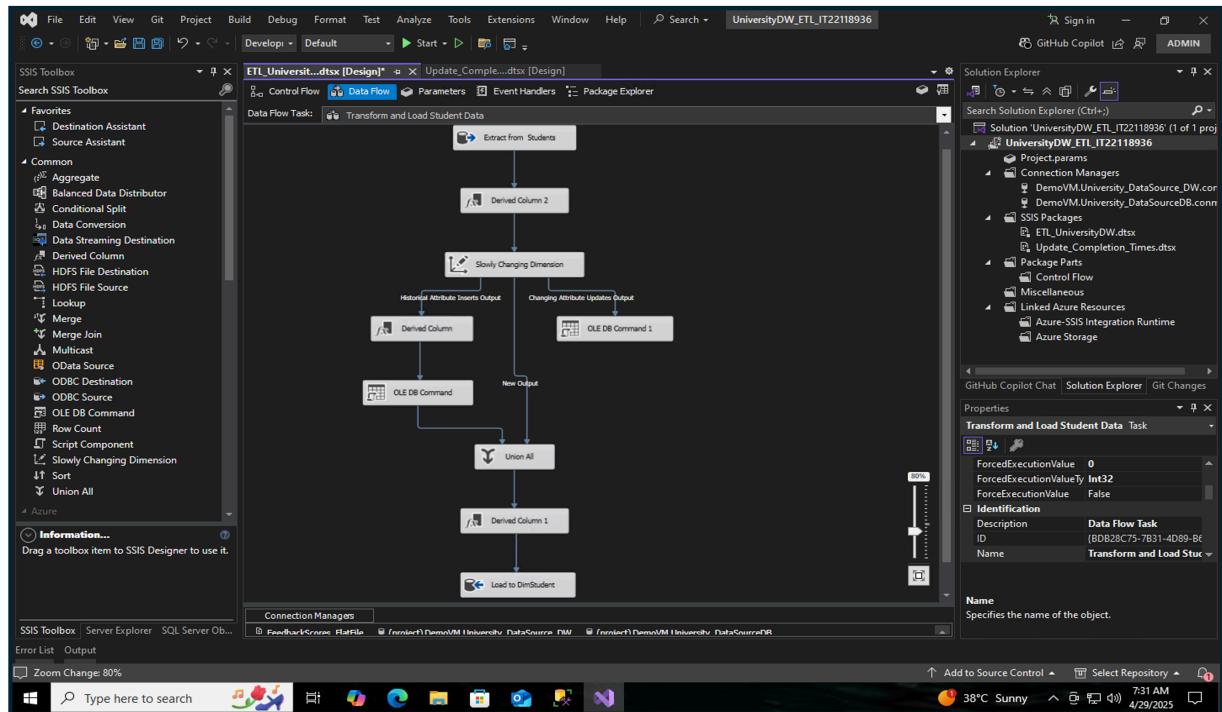
\* Main Objective: Extract → Transform →

Load from University\_DatasourceDB → University\_Datasource\_DW.

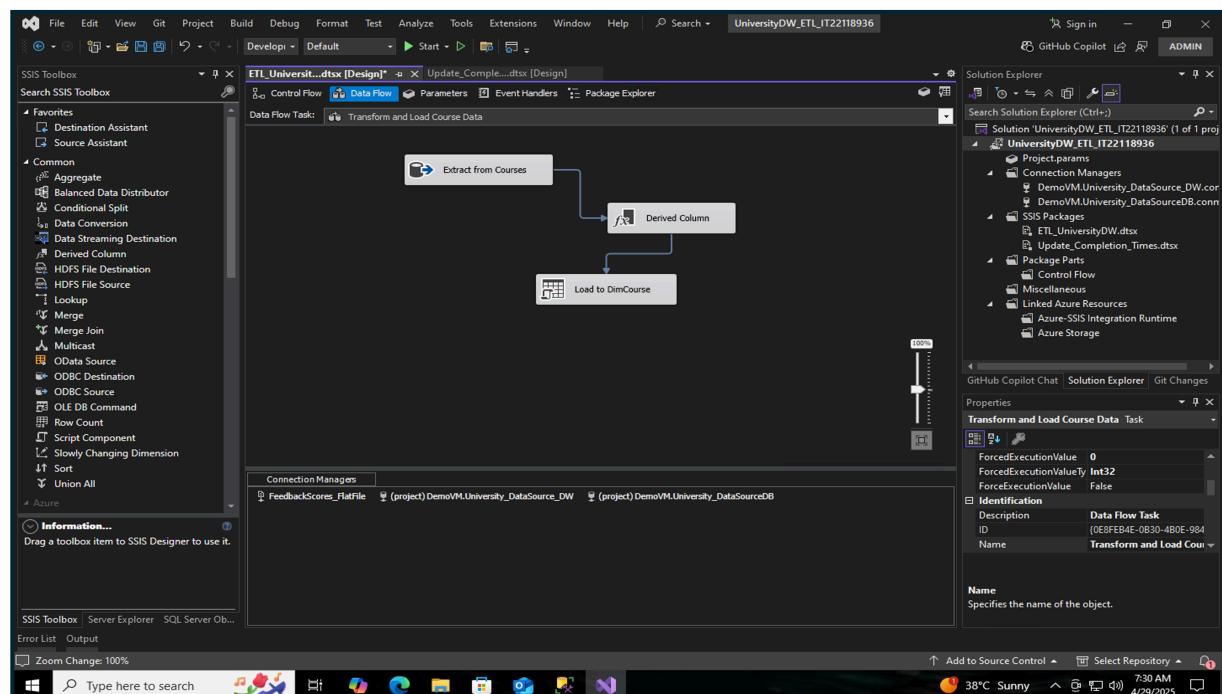


## Step 5: ETL development

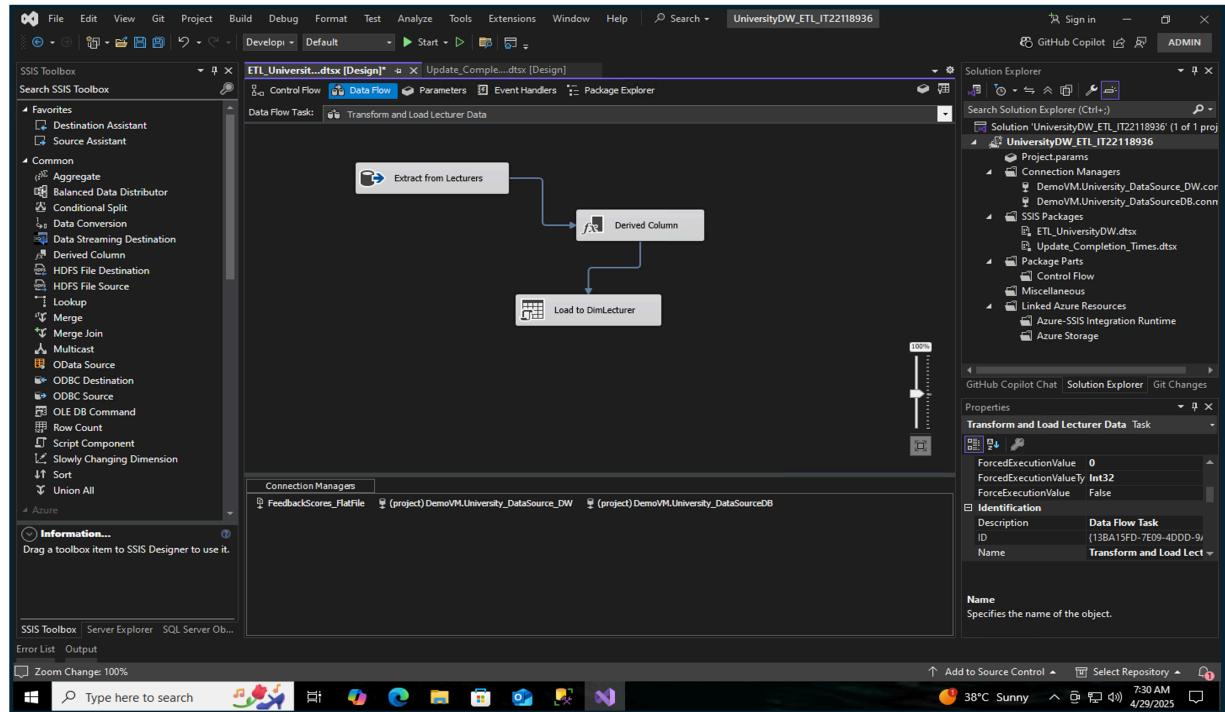
1. DimStudent : Used Stored Procedure (`UpsertDimStudent`) with SCD2 logic (handles slowly changing dimension)



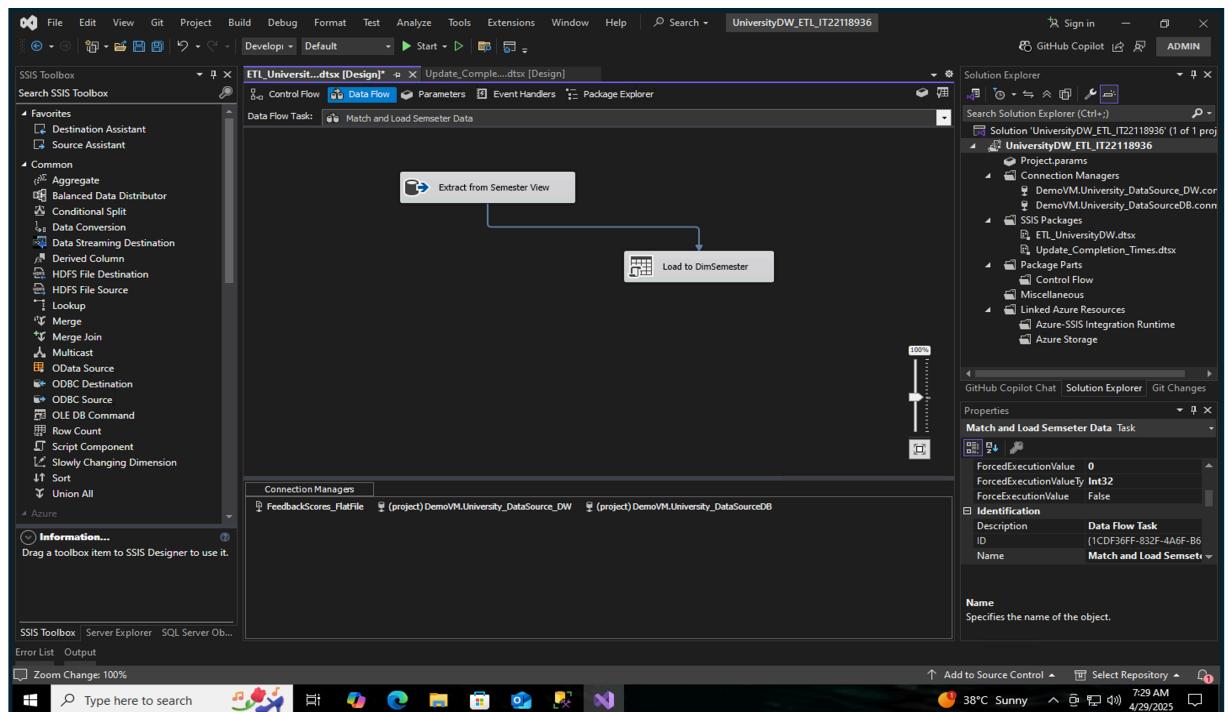
2. DimCourse : Upsert with (`UpsertDimCourse`) procedure



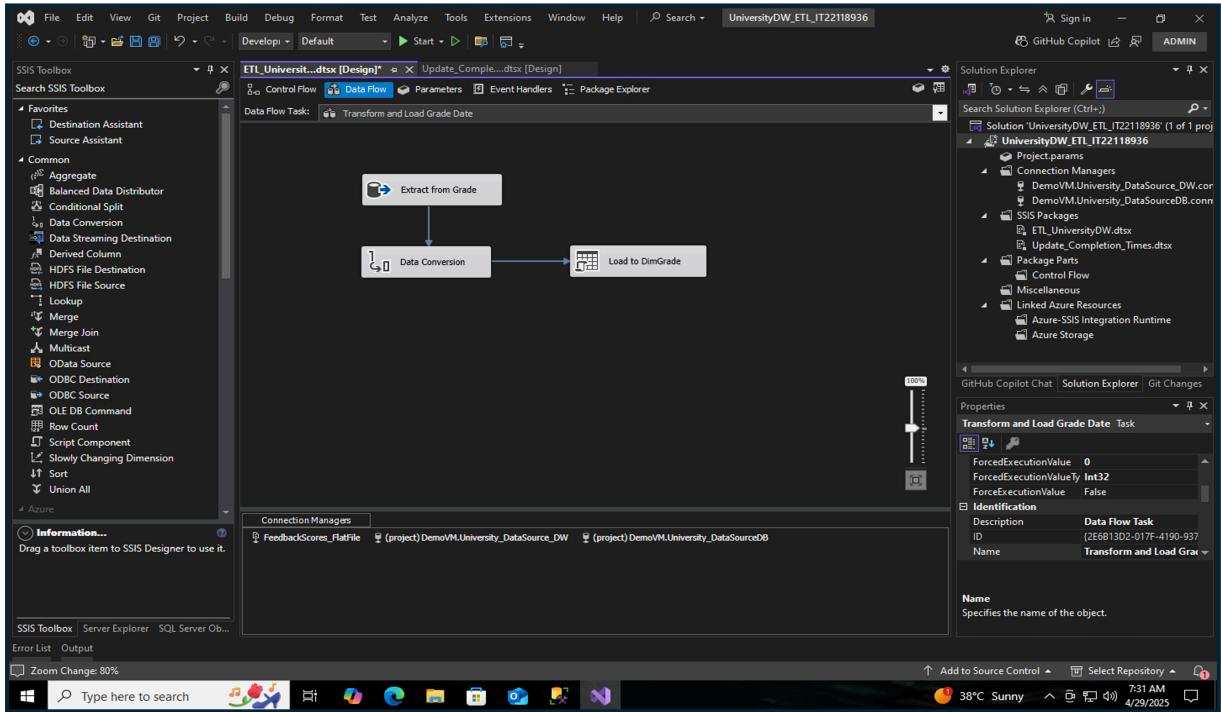
### 3. DimLecturer : Upsert with (UpsertDimLecturer) procedure



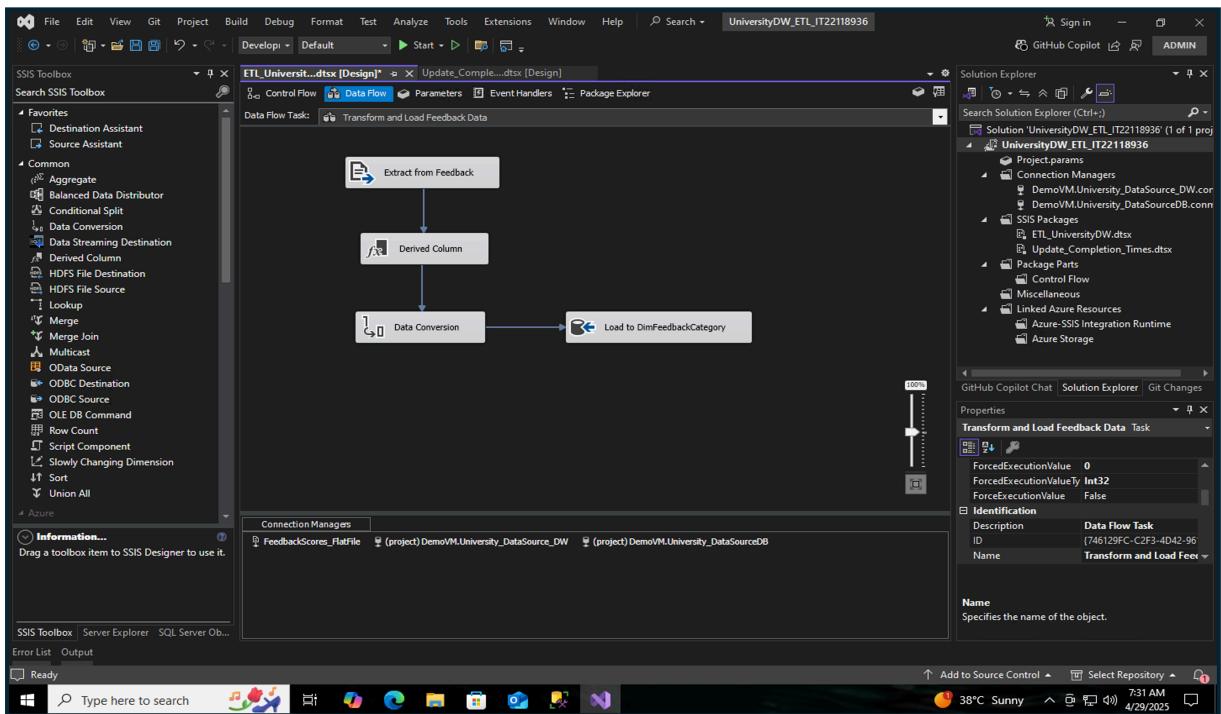
### 4. DimSemester : Derived from Enrollments (semester + year) with transformation logic



## 5. DimGrade : Used (UpsertDimGrade) procedure based on grade, gpa\_points, grade\_source, and retake



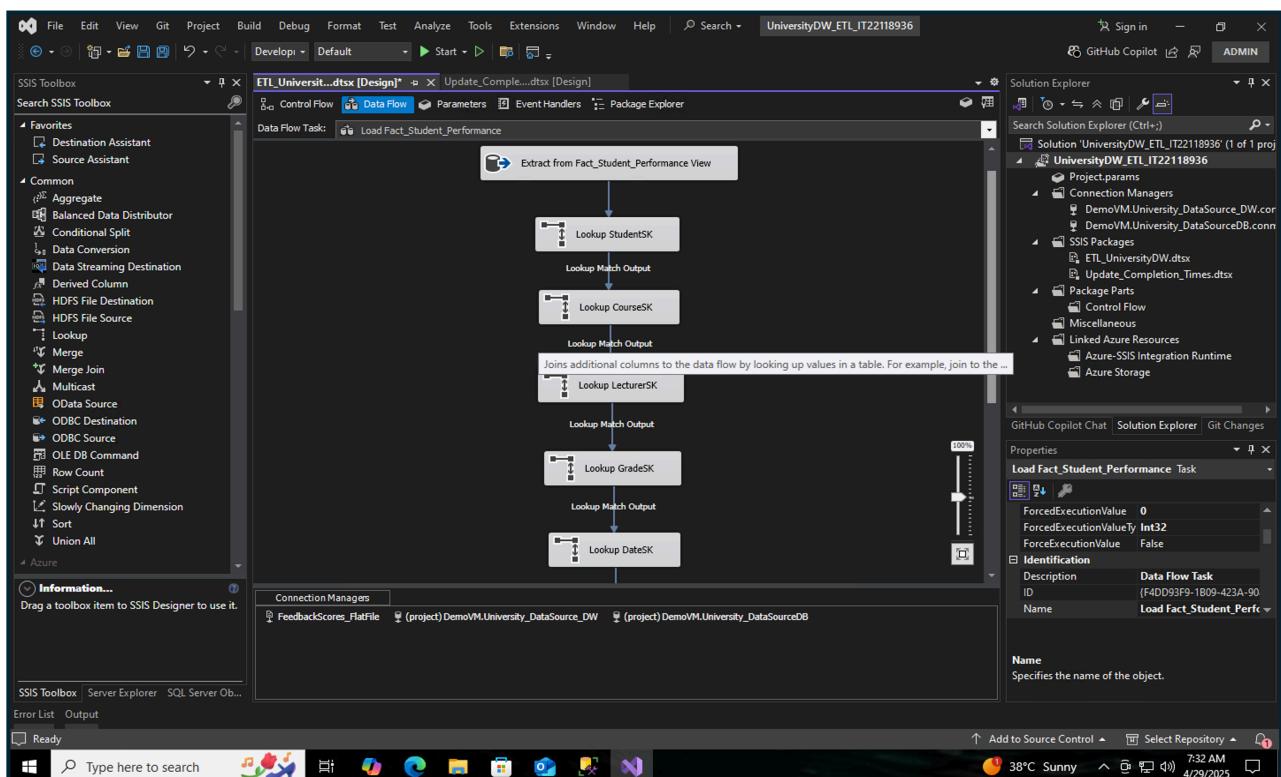
## 6. DimFeedbackCategory : Derived using Derived Column based on feedback score

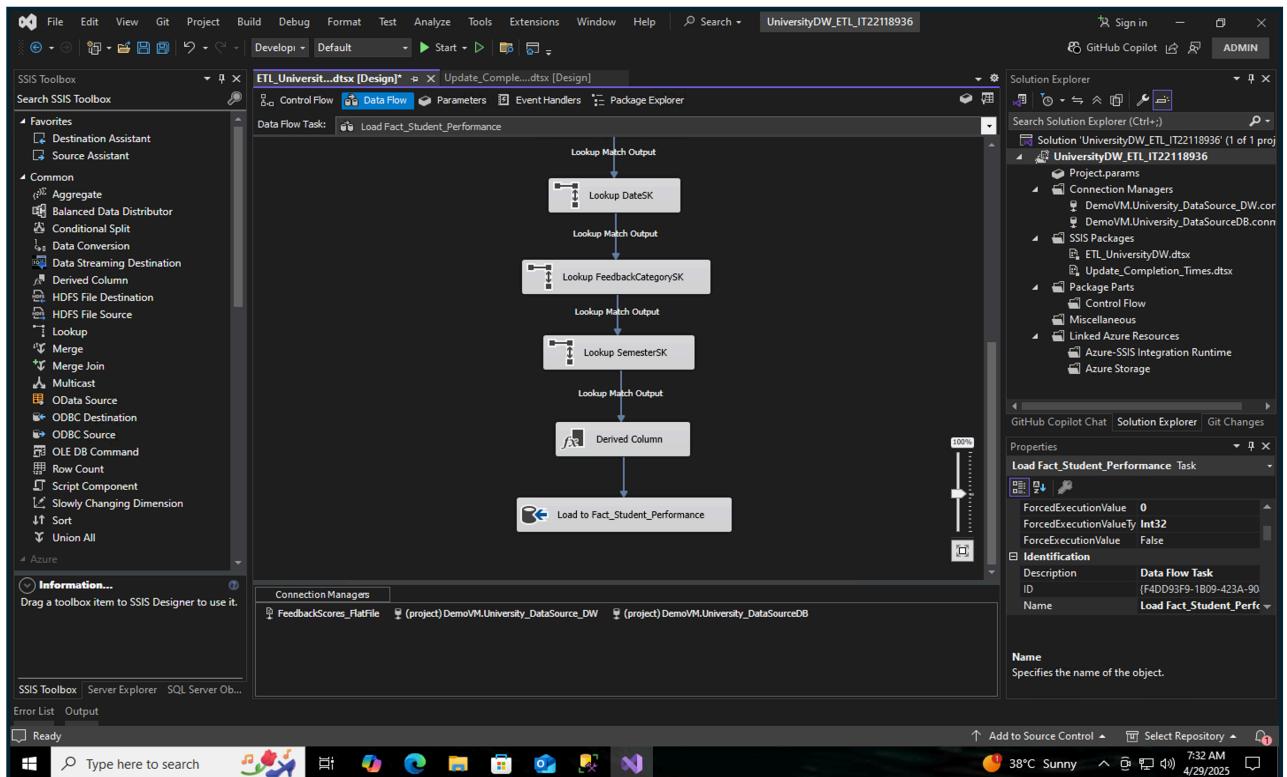


## 7. DimDate : Loaded separately using basic Date dimension logic

## 8. ETL Steps for Fact Table (Fact\_Student\_Performance) :

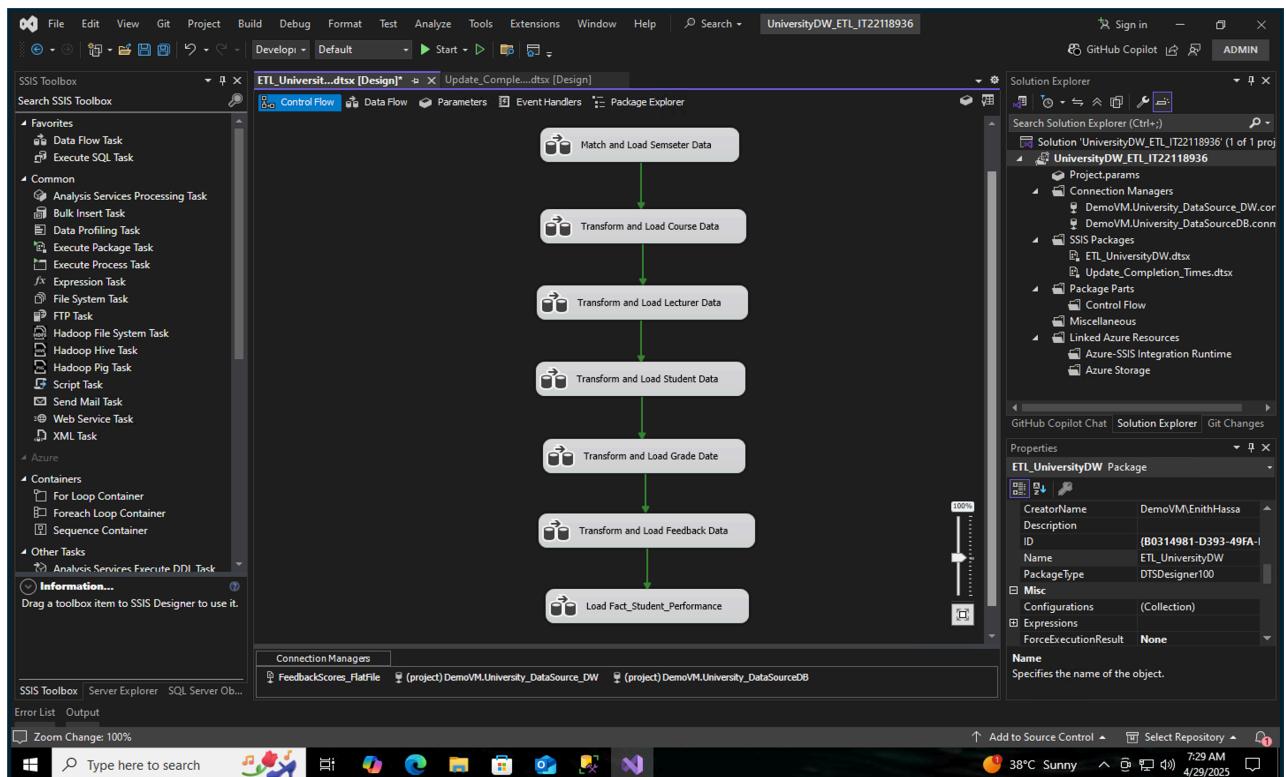
- **Built a source query** combining Enrollments, Grades, Attendance, Semester, and Feedback tables.
- **Added Lookups :**
  - Lookup Student → DimStudent
  - Lookup Course → DimCourse
  - Lookup Lecturer → DimLecturer
  - Lookup Grade → DimGrade
  - Lookup Date → DimDate
  - Lookup → DimFeedbackCategory
  - Lookup → DimSemester
- **Derived Columns :**
  - DropoutFlag (based on enrollment status)
  - PerformanceCategory (based on feedback score)
  - Accm\_Txn\_Create\_Time (Start Timestamp)
  - Accm\_Txn\_Complete\_Time (End Timestamp)





**Note:** Each Dim table was loaded carefully with OLEDB Command Task calling the Stored Procedures. When columns mapping, to avoid mismatch errors Data Conversion and Derived Columns components are used.

## Final Control Flow Structure:



## Step 6: ETL development – Accumulating fact tables

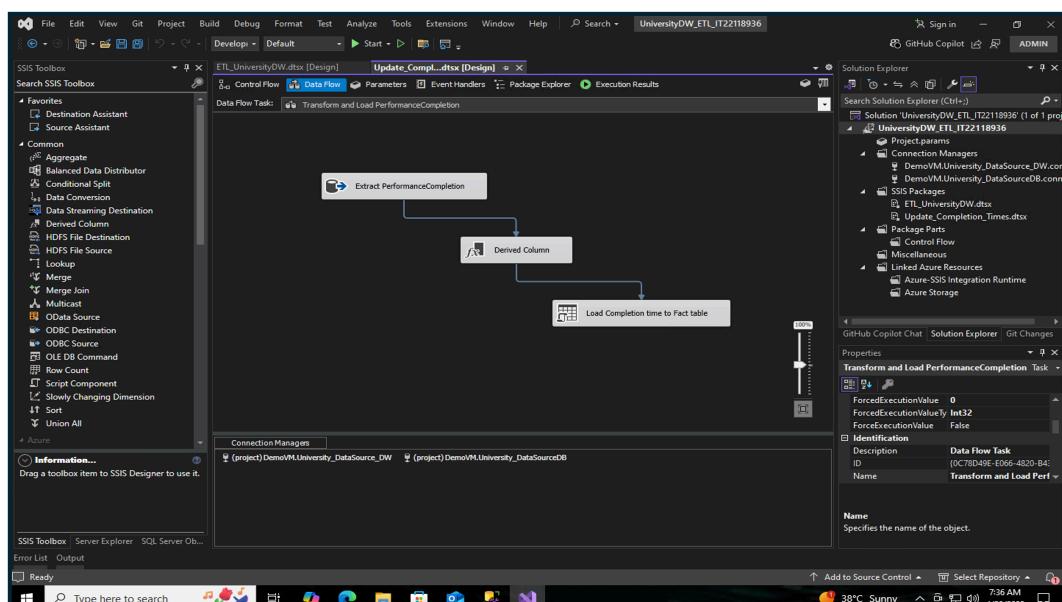
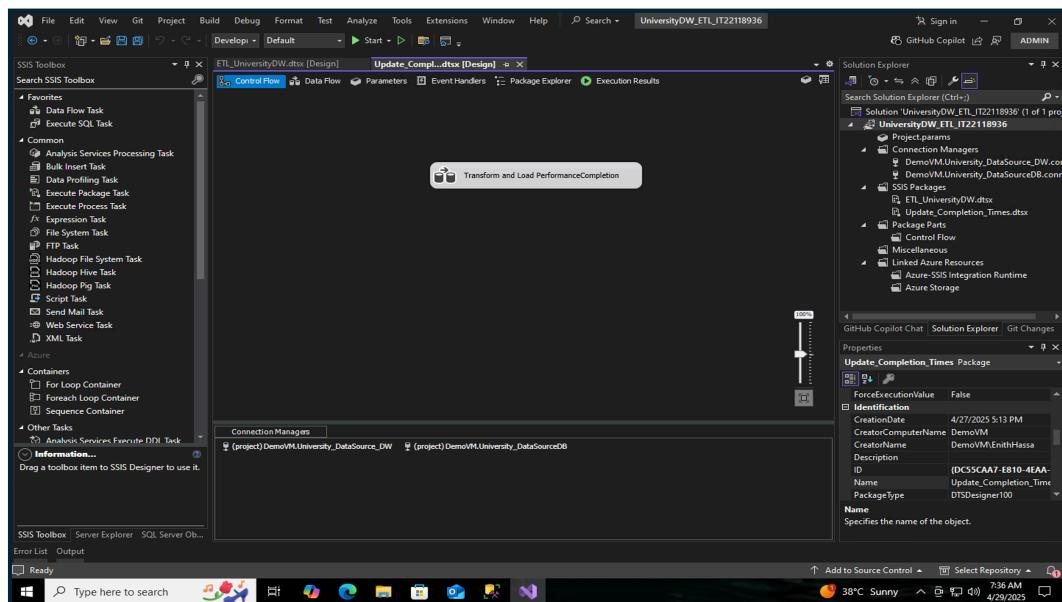
- \* Extended Fact Table Structure:
  - Accm\_Txn\_Create\_Time (datetime): Records when the transaction was created
  - Accm\_Txn\_Complete\_Time (datetime): Records when the transaction was completed
  - Txn\_Process\_Time\_Hours (int): Stores the time difference in hours between creation and completion
- \* Created a Separate Data Source:
 

created a separate data source (PerformanceCompletion table) to track transaction completion information with the structure:

  - TransactionID (corresponds to FactID in your fact table)
  - AccmTxnCompleteTime (when the transaction was actually completed)
- \* Developed a Separate ETL Process:
 

Created a new SSIS package specifically for updating the completion times:

  - Added an OLE DB Source to read from the PerformanceCompletion table
  - Connected it to an OLE DB Command configured to update the fact table



## Executing the Tasks;

