ETL Technical Manual

Version: 1.0
Date: 03/25/2025
Author: Girum Obse
UW ETL Processing Final

Table of Contents

1. Introduction

1.1 Purpose of the Document

This document serves as a comprehensive technical guide for understanding and implementing ETL (Extract, Transform, Load) processes, of a small medical clinic data analytics project. The project involves automating the data ingestion, transformation, and loading processes from CSV files to a structured Data Warehouse. The ETL solution prioritizes scalability, efficiency, and maintainability using SQL Server, SSIS, and Python. This manual provides step-by-step instructions on ETL development, best practices, and troubleshooting techniques.

Additionally, it outlines the four milestones of the project, covering file imports, data warehouse integration, non-SQL ETL processes, and automation. Security, compliance, and performance optimization strategies are also discussed to ensure smooth data flow between systems. Whether building an ETL pipeline for the first time or optimizing an existing one, this document serves as a valuable reference for designing scalable and efficient ETL solutions.

Current Situation

Currently, individual clinics send daily CSV files to the corporate office, where they are manually reviewed, cleaned, and added to one of two databases. This manual ETL process is time-consuming, prone to human error, and lacks scalability, especially as the number of clinics increases. The process delays data availability, affecting decision-making and efficiency.

2. Project Overview

The ETL project automates data movement from daily CSV uploads into structured databases, providing analytical insights. The existing system is entirely manual, and this solution introduces automation to streamline data integration.

- **Source Systems**: CSV files from clinics
- **Destination Systems**: Patients and DoctorsSchedules databases, DWClinicReportData (Data Warehouse), and Excel reports
- **ETL Tools**: SQL Server, SSIS, Python
- **Deployment**: Visual Studio-based structured solution

2.1 Current Workflow over view

Currently, the business has individual clinics send data to a corporate office by uploading CSV files each day. Those **files are then added to one of the two databases**. The current ETL process is **entirely manual**.

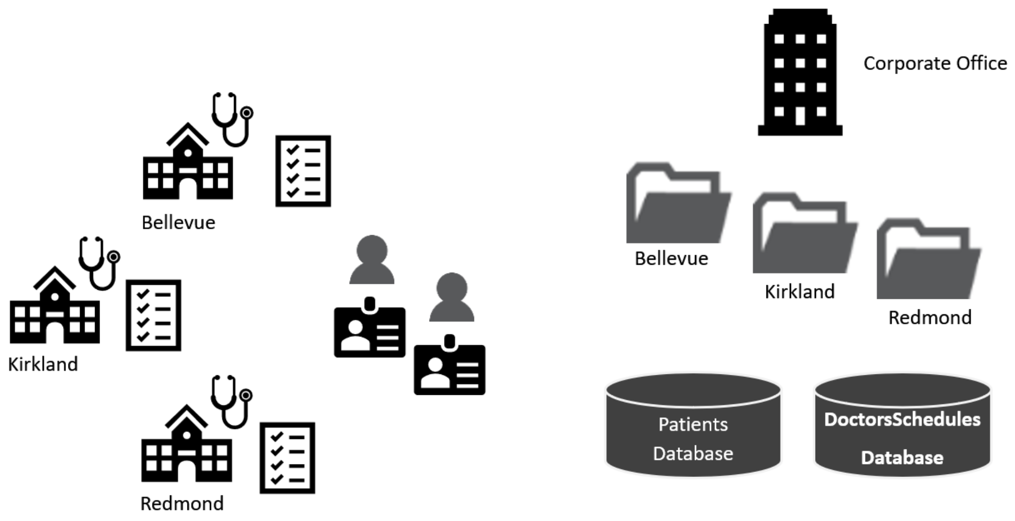


Figure 1. Topology of the current design.

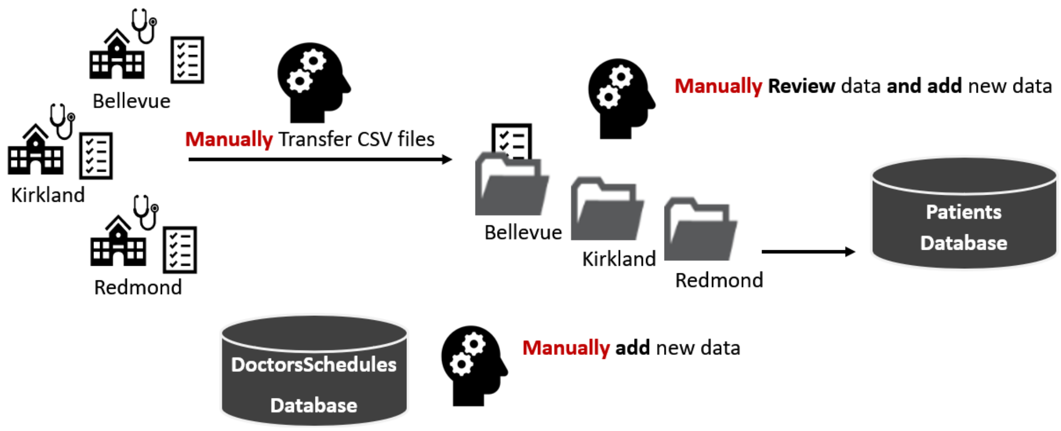

Bellevue
Kirkland
Redmond
Manually Transfer CSV files
Manually Review data and add new data
Bellevue
Kirkland
Redmond
Patients Database
DoctorsSchedules Database
Manually add new data

Figure 2. Visualization of the manual ETL file process

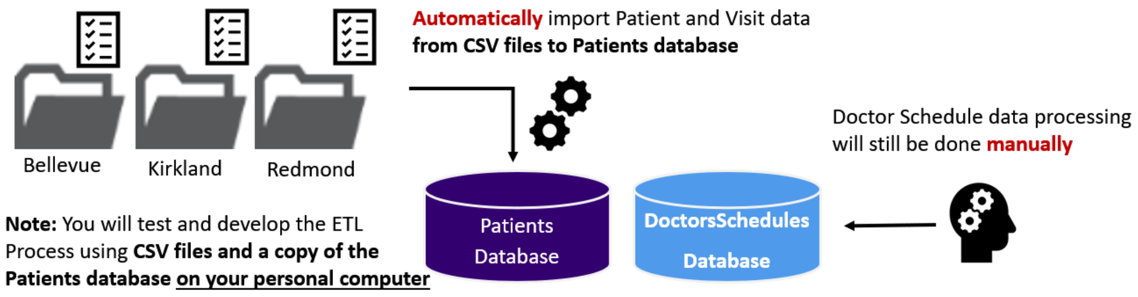

Automatically import Patient and Visit data
from CSV files to Patients database
Bellevue
Kirkland
Redmond
Note: You will test and develop the ETL Process using CSV files and a copy of the Patients database on your personal computer
Patients Database
DoctorsSchedules Database
Doctor Schedule data processing will still be done manually

Figure 3. Visualization of the manual ETL file process

2.2,Proposed ETL process

A new ETL process will automate parts of this workflow, with potential for full automation in the future. Below, you'll find detailed descriptions of the entire process and steps, along with sample code snippets and screenshots for clarity.

2.3 Audience

This manual is intended for a wide range of professionals involved in data management and analytics, including:
- ETL Developers – Those responsible for designing and implementing ETL pipelines.
- Data Engineers – Professionals who manage data pipelines and ensure data flows correctly.
- Database Administrators – Responsible for maintaining data integrity and optimizing database performance.
- Business Intelligence Analysts – Those who analyze and report on business data.
- IT Managers – Decision-makers overseeing data-related projects.

This document assumes some familiarity with databases, SQL, and basic scripting but does not require advanced ETL experience.

2.4 ETL Overview

2.4.1 What is ETL?

ETL stands for Extract, Transform, and Load—a three-step process used for integrating data from multiple sources into a centralized data warehouse or analytics system. The Extract phase retrieves data from various sources such as databases, web APIs, and flat files. The Transform phase applies business rules to clean, standardize, and enrich the data. The Load phase ensures the data is stored in the target system, ready for reporting and analysis. ETL is widely used in data warehousing, analytics, and business intelligence, ensuring that organizations work with high-quality, consistent, and structured data.

2.4.1 Why ETL is Important?

ETL plays a crucial role in modern data-driven organizations by enabling them to consolidate data from multiple sources and prepare it for analysis. Without ETL, businesses would struggle with inconsistent, incomplete, and redundant data. ETL ensures that data is cleaned, standardized, and formatted properly before it reaches reporting tools. Additionally, ETL processes support historical data storage, allowing companies to perform trend analysis, forecasting, and strategic decision-making. ETL also enhances data governance and compliance, ensuring that sensitive information is processed securely according to industry regulations such as GDPR and HIPAA.

3.Environment Setup

Before beginning ETL development, ensure the following components are installed and configured:

- SQL Server with Management Studio
- Visual Studio with SSIS extensions
- Python 3.9 with required libraries
- Access to database backup files (.BAK) for restoration
- CSV files for testing in the C:_BISolutions\ClinicDailyData directory

---

4. Milestone 1: File-Based ETL

Data Review

- Assess CSV file structures for inconsistencies (e.g., column names, data types, missing values).

Setting Up the Development Environment

- Create a Visual Studio solution with organized folders.

Database Restoration

Execute the following SQL script to restore databases:

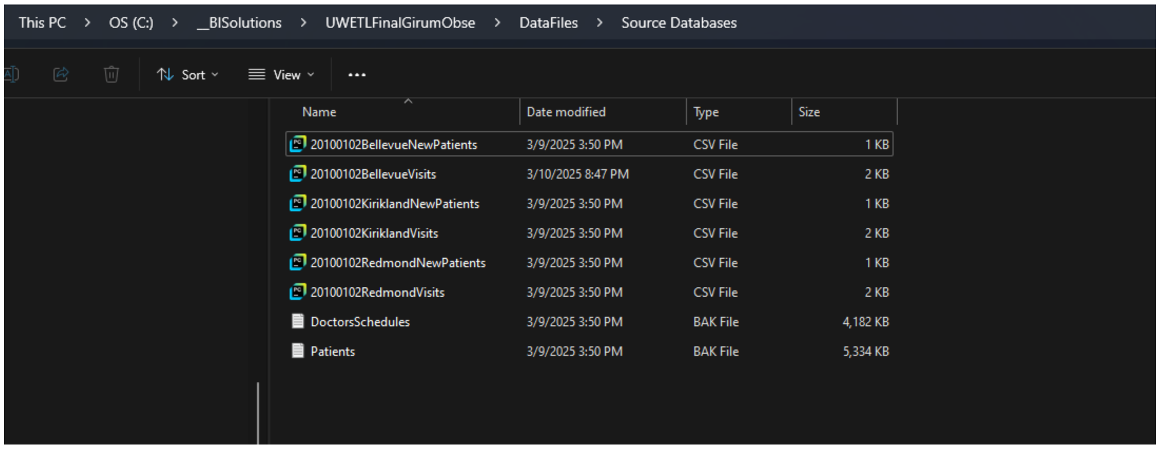
```
ALTER DATABASE [Patients] SET SINGLE_USER WITH ROLLBACK IMMEDIATE;
RESTORE DATABASE [Patients]
 FROM DISK = N'C:/_BISolutions/Databases/Patients.bak'
 WITH RECOVERY, REPLACE;
ALTER DATABASE [Patients] SET MULTI_USER;
GO
```

Data Transformation Documentation

- Maintain an Excel metadata sheet tracking transformations.

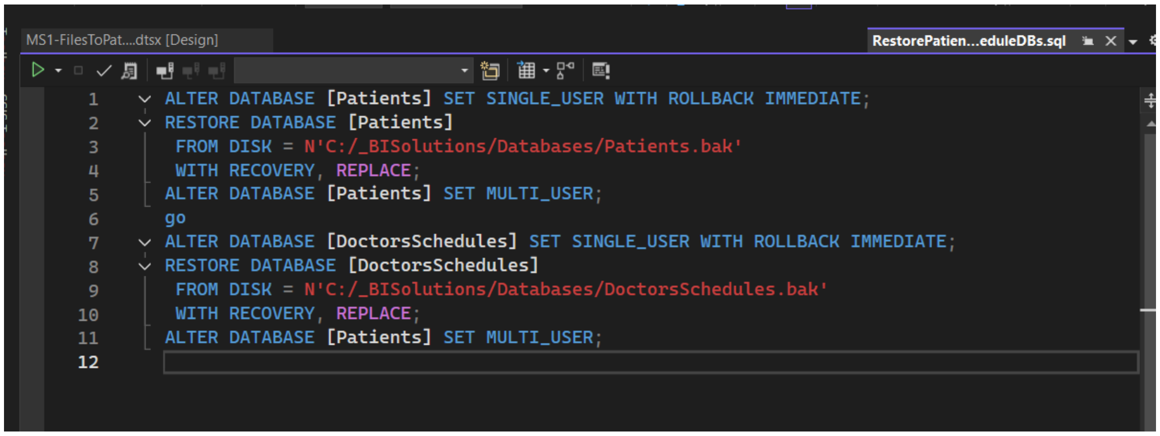
SSIS Package Implementation

- Develop an SSIS package (ETLFilesToDatabases.dtsx) to automate CSV imports.


This PC > OS (C:) > _BISolutions > UWETLFinalGirumObse > DataFiles > Source Databases
Sort
View

Name	Date modified	Type	Size
20100102BellevueNewPatients	3/9/2025 3:50 PM	CSV File	1 KB
20100102BellevueVisits	3/10/2025 8:47 PM	CSV File	2 KB
20100102KiriklandNewPatients	3/9/2025 3:50 PM	CSV File	1 KB
20100102KiriklandVisits	3/9/2025 3:50 PM	CSV File	2 KB
20100102RedmondNewPatients	3/9/2025 3:50 PM	CSV File	1 KB
20100102RedmondVisits	3/9/2025 3:50 PM	CSV File	2 KB
DoctorsSchedules	3/9/2025 3:50 PM	BAK File	4,182 KB
Patients	3/9/2025 3:50 PM	BAK File	5,334 KB

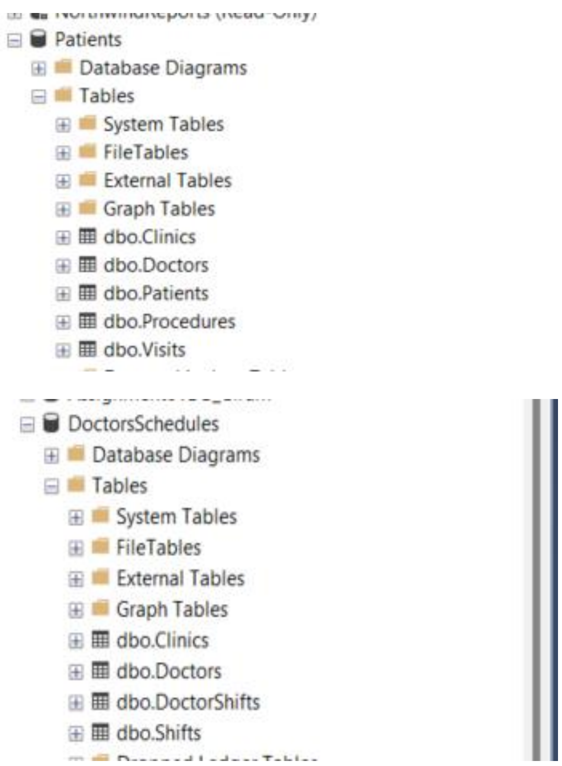
Figure 4 Excel source file for different clinics



MS1-FilesToPat....dtsx [Design]   RestorePatien...eduleDBs.sql

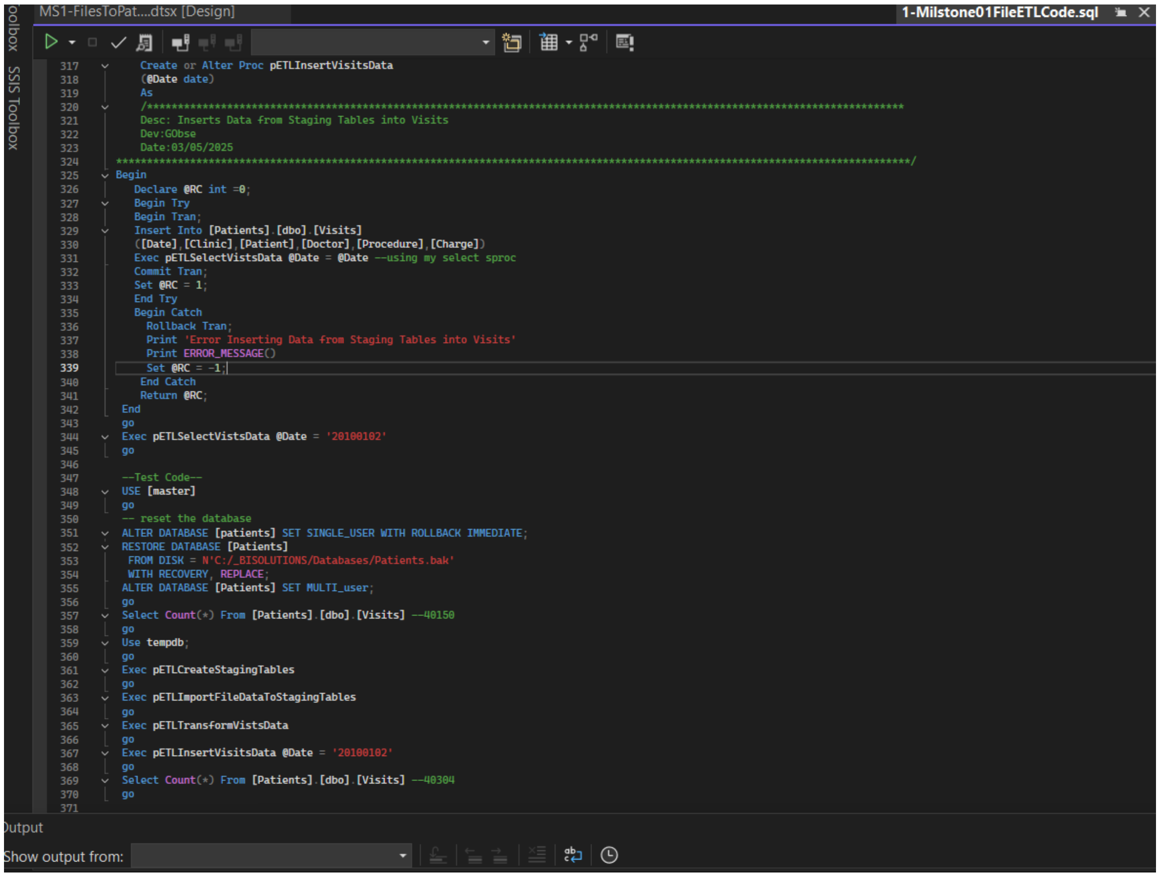
```sql
ALTER DATABASE [Patients] SET SINGLE_USER WITH ROLLBACK IMMEDIATE;
RESTORE DATABASE [Patients]
 FROM DISK = N'C:/_BISolutions/Databases/Patients.bak'
 WITH RECOVERY, REPLACE;
ALTER DATABASE [Patients] SET MULTI_USER;
go
ALTER DATABASE [DoctorsSchedules] SET SINGLE_USER WITH ROLLBACK IMMEDIATE;
RESTORE DATABASE [DoctorsSchedules]
 FROM DISK = N'C:/_BISolutions/Databases/DoctorsSchedules.bak'
 WITH RECOVERY, REPLACE;
ALTER DATABASE [Patients] SET MULTI_USER;
```

Figure5, sql code script showing restoring the patients and doctorschedule database



```
Patients
  Database Diagrams
  Tables
    System Tables
    FileTables
    External Tables
    Graph Tables
    dbo.Clinics
    dbo.Doctors
    dbo.Patients
    dbo.Procedures
    dbo.Visits

DoctorsSchedules
  Database Diagrams
  Tables
    System Tables
    FileTables
    External Tables
    Graph Tables
    dbo.Clinics
    dbo.Doctors
    dbo.DoctorShifts
    dbo.Shifts
```

Fiqure6, showing doctorschedule and patients database tables



```sql
    Create or Alter Proc pETLInsertVisitsData
    (@Date date)
    As
    /**********************************************************************************************************************
    Desc: Inserts Data from Staging Tables into Visits
    Dev:GObse
    Date:03/05/2025
************************************************************************************************************************/
Begin
   Declare @RC int =0;
   Begin Try
   Begin Tran;
   Insert Into [Patients].[dbo].[Visits]
   ([Date],[Clinic],[Patient],[Doctor],[Procedure],[Charge])
   Exec pETLSelectVistsData @Date = @Date --using my select sproc
   Commit Tran;
   Set @RC = 1;
   End Try
   Begin Catch
     Rollback Tran;
     Print 'Error Inserting Data from Staging Tables into Visits'
     Print ERROR_MESSAGE()
     Set @RC = -1;
    End Catch
    Return @RC;
 End
 go
 Exec pETLSelectVistsData @Date = '20100102'
 go

 --Test Code--
 USE [master]
 go
 -- reset the database
 ALTER DATABASE [patients] SET SINGLE_USER WITH ROLLBACK IMMEDIATE;
 RESTORE DATABASE [Patients]
  FROM DISK = N'C:/_BISOLUTIONS/Databases/Patients.bak'
  WITH RECOVERY, REPLACE;
 ALTER DATABASE [Patients] SET MULTI_user;
 go
 Select Count(*) From [Patients].[dbo].[Visits] --40150
 go
 Use tempdb;
 go
 Exec pETLCreateStagingTables
 go
 Exec pETLImportFileDataToStagingTables
 go
 Exec pETLTransformVistsData
 go
 Exec pETLInsertVisitsData @Date = '20100102'
 go
 Select Count(*) From [Patients].[dbo].[Visits] --40304
 go
```

Figure7: part of sql script showing how data loaded to databases

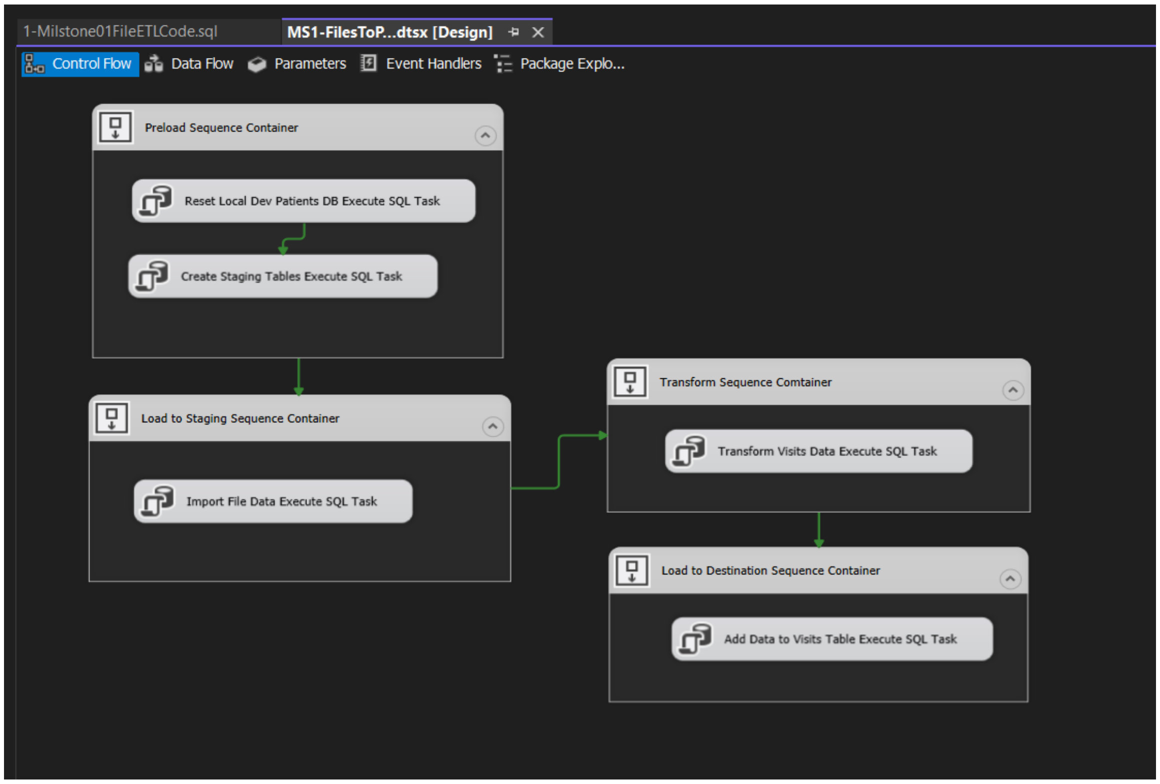
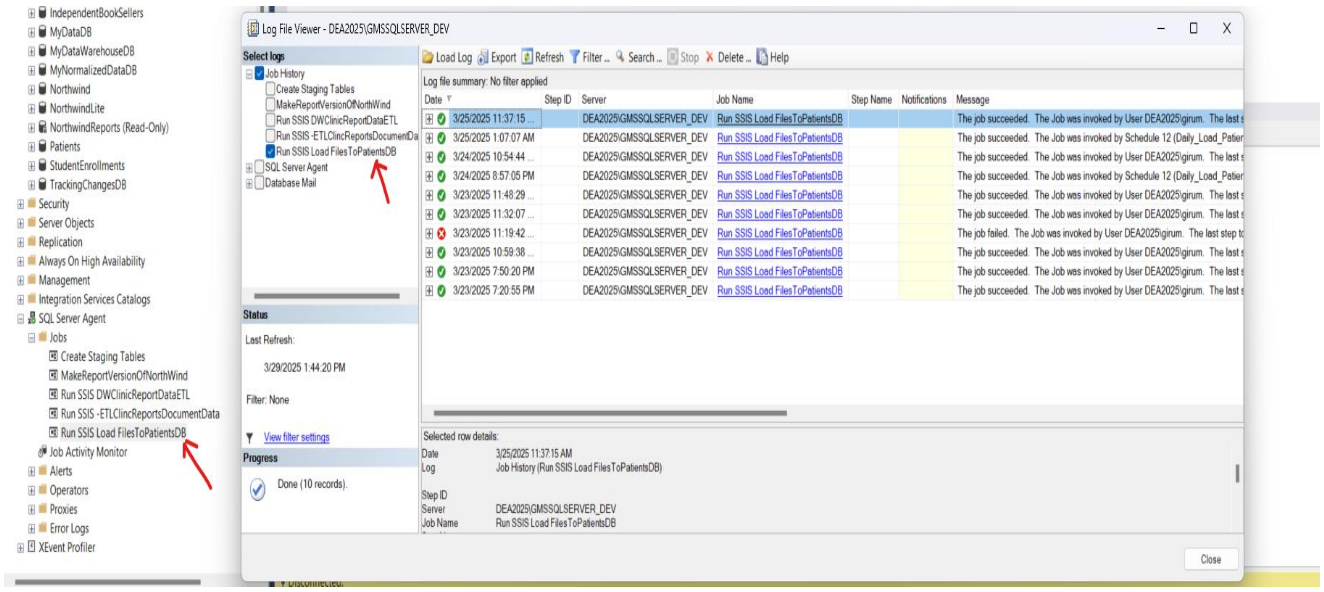

1-Milstone01FileETLCode.sql
MS1-FilesToP...dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explo...
Preload Sequence Container
Reset Local Dev Patients DB Execute SQL Task
Create Staging Tables Execute SQL Task
Load to Staging Sequence Container
Import File Data Execute SQL Task
Transform Sequence Comtainer
Transform Visits Data Execute SQL Task
Load to Destination Sequence Container
Add Data to Visits Table Execute SQL Task

Figure 8, ssis package for dataloaded from excel source to visits table in patients database


Log File Viewer - DEA2025\GMSSQLSERVER_DEV
Select logs
Job History
Create Staging Tables
MakeReportVersionOfNorthWind
Run SSIS DWClincReportDataETL
Run SSIS -ETLClincReportsDocumentDa
Run SSIS Load FilesToPatientsDB
SQL Server Agent
Database Mail
Load Log
Export
Refresh
Filter ...
Search ...
Stop
Delete ...
Help
Jobs
Create Staging Tables
MakeReportVersionOfNorthWind
Run SSIS DWClincReportDataETL
Run SSIS -ETLClincReportsDocumentData
Run SSIS Load FilesToPatientsDB
Job Activity Monitor
Close

Log file summary: No filter applied

Date	Step ID	Server	Job Name	Step Name	Notifications	Message
3/25/2025 11:37:15 ...		DEA2025\GMSSQLSERVER_DEV	Run SSIS Load FilesToPatientsDB			The job succeeded. The Job was invoked by User DEA2025\girum. The last s
3/25/2025 1:07:07 AM		DEA2025\GMSSQLSERVER_DEV	Run SSIS Load FilesToPatientsDB			The job succeeded. The Job was invoked by Schedule 12 (Daily_Load_Patien
3/24/2025 10:54:44 ...		DEA2025\GMSSQLSERVER_DEV	Run SSIS Load FilesToPatientsDB			The job succeeded. The Job was invoked by User DEA2025\girum. The last s
3/24/2025 8:57:05 PM		DEA2025\GMSSQLSERVER_DEV	Run SSIS Load FilesToPatientsDB			The job succeeded. The Job was invoked by Schedule 12 (Daily_Load_Patien
3/23/2025 11:48:29 ...		DEA2025\GMSSQLSERVER_DEV	Run SSIS Load FilesToPatientsDB			The job succeeded. The Job was invoked by User DEA2025\girum. The last s
3/23/2025 11:32:07 ...		DEA2025\GMSSQLSERVER_DEV	Run SSIS Load FilesToPatientsDB			The job succeeded. The Job was invoked by User DEA2025\girum. The last s
3/23/2025 11:19:42 ...		DEA2025\GMSSQLSERVER_DEV	Run SSIS Load FilesToPatientsDB			The job failed. The Job was invoked by User DEA2025\girum. The last step to
3/23/2025 10:59:38 ...		DEA2025\GMSSQLSERVER_DEV	Run SSIS Load FilesToPatientsDB			The job succeeded. The Job was invoked by User DEA2025\girum. The last s
3/23/2025 7:50:20 PM		DEA2025\GMSSQLSERVER_DEV	Run SSIS Load FilesToPatientsDB			The job succeeded. The Job was invoked by User DEA2025\girum. The last s
3/23/2025 7:20:55 PM		DEA2025\GMSSQLSERVER_DEV	Run SSIS Load FilesToPatientsDB			The job succeeded. The Job was invoked by User DEA2025\girum. The last s

Status

Last Refresh:

3/29/2025 1:44:20 PM

Filter: None

View filter settings

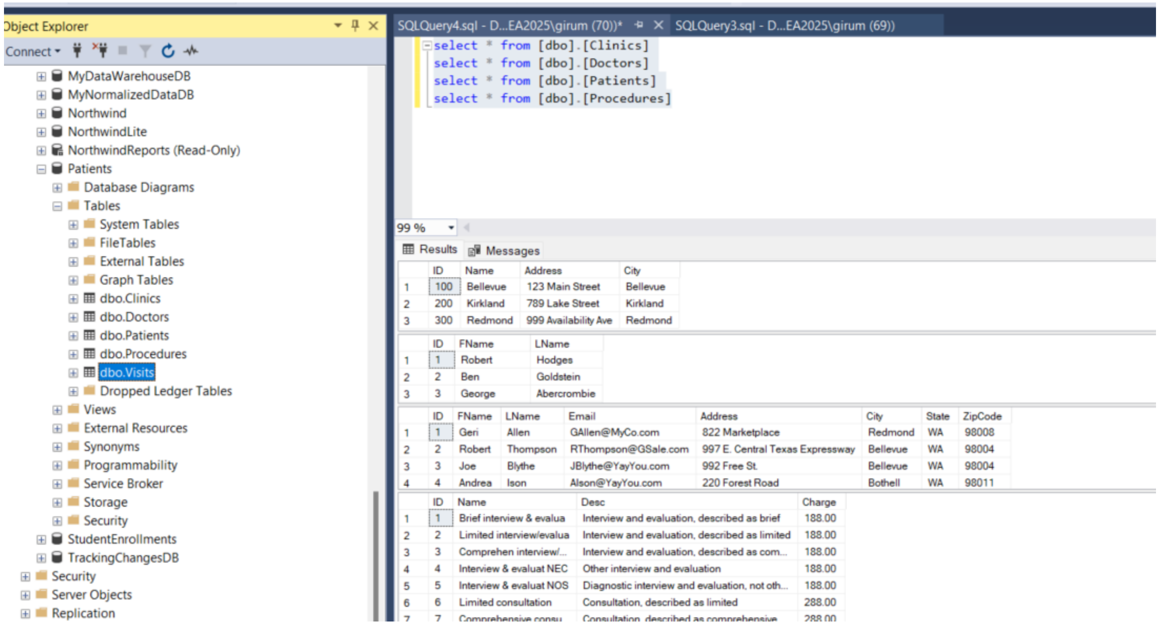
Progress

Done (10 records).

Selected row details:

Date	3/25/2025 11:37:15 AM
Log	Job History (Run SSIS Load FilesToPatientsDB)
Step ID	
Server	DEA2025\GMSSQLSERVER_DEV
Job Name	Run SSIS Load FilesToPatientsDB

Figure9, showing screenshot of scheduled job to load data to patients database tables



```sql
select * from [dbo].[Clinics]
select * from [dbo].[Doctors]
select * from [dbo].[Patients]
select * from [dbo].[Procedures]
```

	ID	Name	Address	City
1	100	Bellevue	123 Main Street	Bellevue
2	200	Kirkland	789 Lake Street	Kirkland
3	300	Redmond	999 Availability Ave	Redmond

	ID	FName	LName
1	1	Robert	Hodges
2	2	Ben	Goldstein
3	3	George	Abercrombie

	ID	FName	LName	Email	Address	City	State	ZipCode
1	1	Geri	Allen	GAllen@MyCo.com	822 Marketplace	Redmond	WA	98008
2	2	Robert	Thompson	RThompson@GSale.com	997 E. Central Texas Expressway	Bellevue	WA	98004
3	3	Joe	Blythe	JBlythe@YayYou.com	992 Free St.	Bellevue	WA	98004
4	4	Andrea	Ison	AIson@YayYou.com	220 Forest Road	Bothell	WA	98011

	ID	Name	Desc	Charge
1	1	Brief interview & evalua	Interview and evaluation, described as brief	188.00
2	2	Limited interview/evalua	Interview and evaluation, described as limited	188.00
3	3	Comprehen interview/...	Interview and evaluation, described as com...	188.00
4	4	Interview & evaluat NEC	Other interview and evaluation	188.00
5	5	Interview & evaluat NOS	Diagnostic interview and evaluation, not oth...	188.00
6	6	Limited consultation	Consultation, described as limited	288.00
7	7	Comprehensive consu	Consultation, described as comprehensive	288.00

Figure 10, showing data in patients database tables

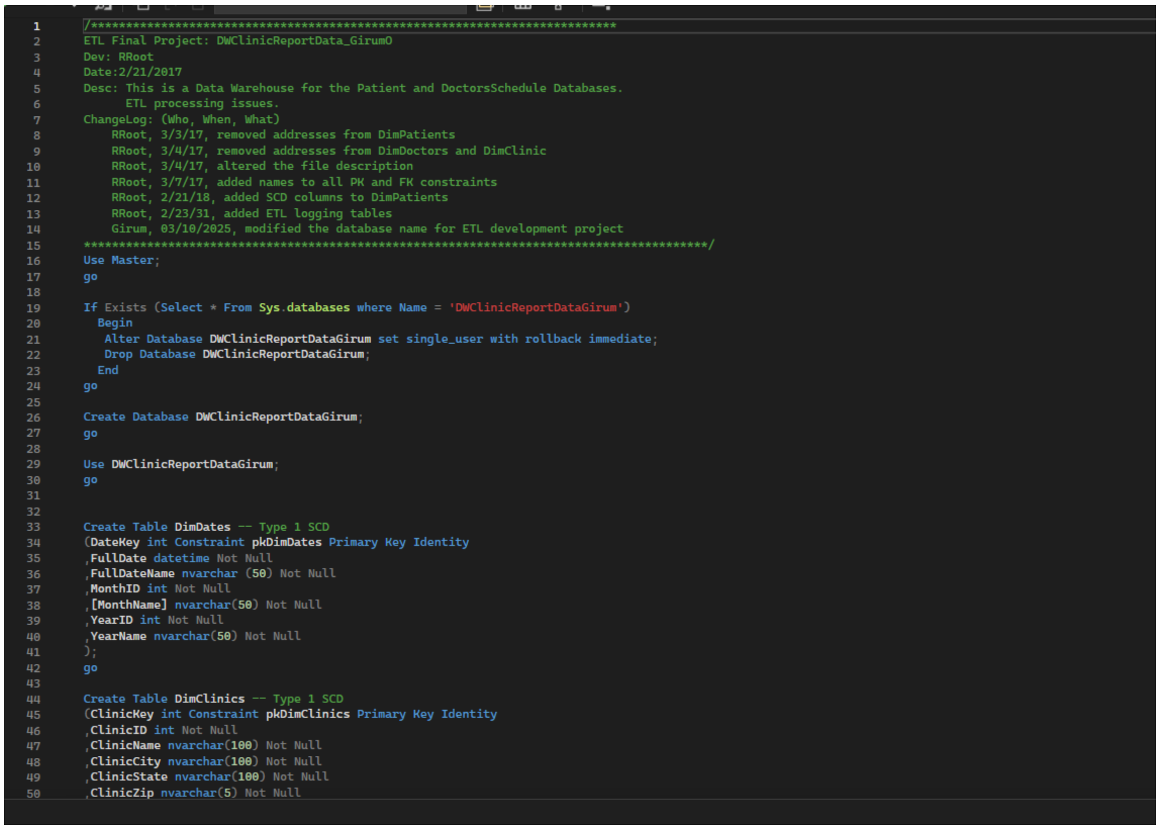
---

5. Milestone 2: Data Warehouse ETL

Process Steps

Creating the Data Warehouse

- Run the provided SQL script (`\Scripts\CreateDWClinicReportData.sql`) to create the **DWClinicReportData database.**



```sql
/*****************************************************************************
ETL Final Project: DWClinicReportData_GirumO
Dev: RRoot
Date:2/21/2017
Desc: This is a Data Warehouse for the Patient and DoctorsSchedule Databases.
      ETL processing issues.
ChangeLog: (Who, When, What)
    RRoot, 3/3/17, removed addresses from DimPatients
    RRoot, 3/4/17, removed addresses from DimDoctors and DimClinic
    RRoot, 3/4/17, altered the file description
    RRoot, 3/7/17, added names to all PK and FK constraints
    RRoot, 2/21/18, added SCD columns to DimPatients
    RRoot, 2/23/31, added ETL logging tables
    Girum, 03/10/2025, modified the database name for ETL development project
*****************************************************************************************/
Use Master;
go

If Exists (Select * From Sys.databases where Name = 'DWClinicReportDataGirum')
  Begin
   Alter Database DWClinicReportDataGirum set single_user with rollback immediate;
   Drop Database DWClinicReportDataGirum;
  End
go

Create Database DWClinicReportDataGirum;
go

Use DWClinicReportDataGirum;
go


Create Table DimDates -- Type 1 SCD
(DateKey int Constraint pkDimDates Primary Key Identity
,FullDate datetime Not Null
,FullDateName nvarchar (50) Not Null
,MonthID int Not Null
,[MonthName] nvarchar(50) Not Null
,YearID int Not Null
,YearName nvarchar(50) Not Null
);
go

Create Table DimClinics -- Type 1 SCD
(ClinicKey int Constraint pkDimClinics Primary Key Identity
,ClinicID int Not Null
,ClinicName nvarchar(100) Not Null
,ClinicCity nvarchar(100) Not Null
,ClinicState nvarchar(100) Not Null
,ClinicZip nvarchar(5) Not Null
```

Figure11, sql code script of the datawarehouse

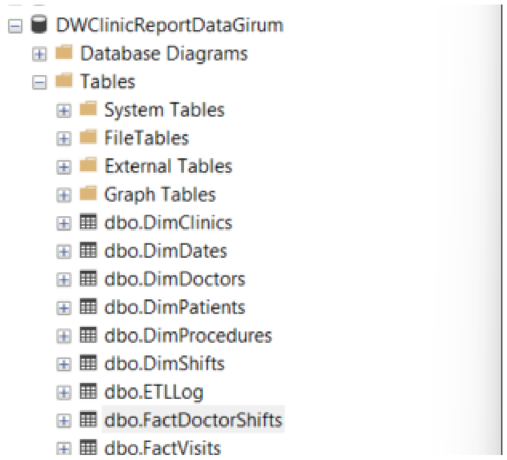
- Verify that all required tables, schemas, and constraints are properly created.

- Check for any errors or missing objects in the database after execution.

Reviewing the Database Design

- Identify the **dimension tables (e.g., DimPatients, DimDoctors, DimShifts) and fact tables(e.g., FactDoctorShifts and FactVisits).**

- Understand relationships between tables, primary and foreign keys.



```
DWClinicReportDataGirum
  Database Diagrams
  Tables
    System Tables
    FileTables
    External Tables
    Graph Tables
    dbo.DimClinics
    dbo.DimDates
    dbo.DimDoctors
    dbo.DimPatients
    dbo.DimProcedures
    dbo.DimShifts
    dbo.ETLLog
    dbo.FactDoctorShifts
    dbo.FactVisits
```

SQL ETL and SSIS Implementation

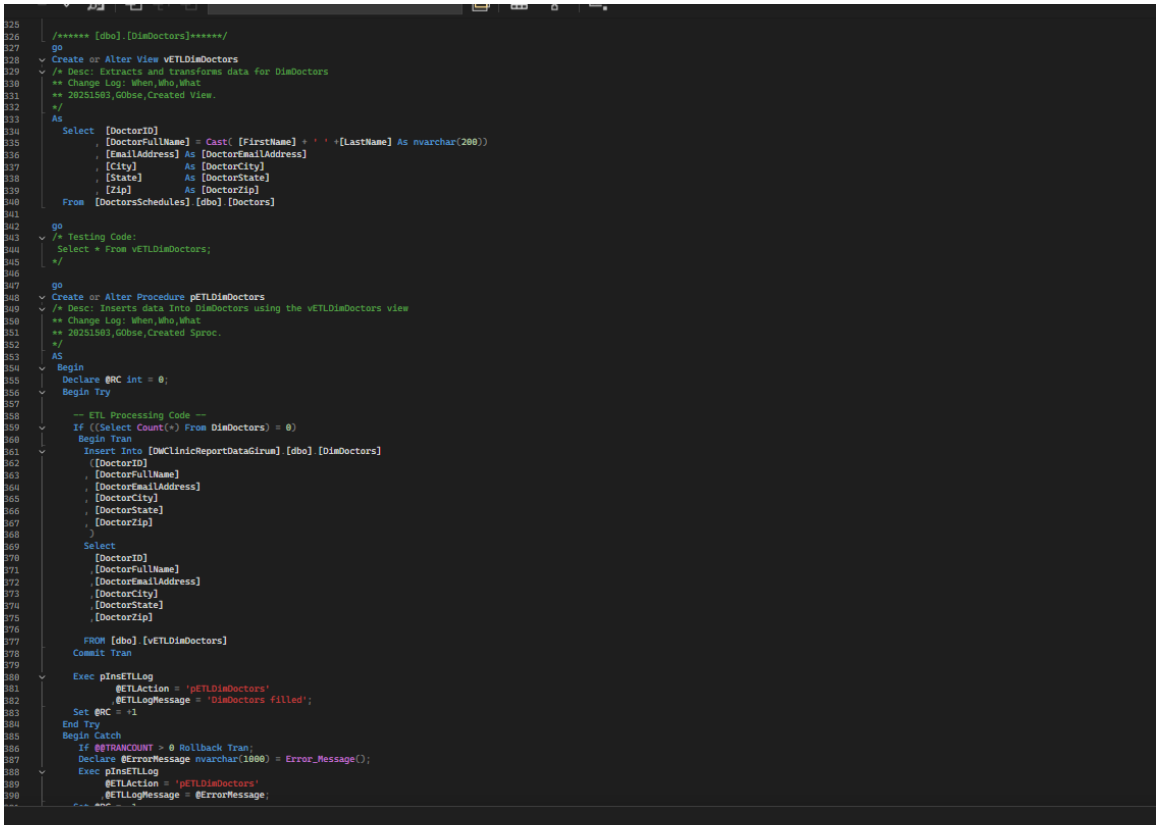
- **Create SQL Views and Stored Procedures:**

- Develop **views to extract necessary data from the **DoctorsSchedules** and **Patients** OLTP databases.**

- Write **stored procedures to transform and load the data into the **DWClinicReportData** warehouse.**

- Implement **data cleansing to remove duplicates, handle null values, and ensure consistency.**

Figure12, showing ETL script of data load from OLTP database to the OLAP database example of Dimdoctor table



```
/****** [dbo].[DimDoctors]******/
go
Create or Alter View vETLDimDoctors
/* Desc: Extracts and transforms data for DimDoctors
** Change Log: When,Who,What
** 20251503,GObse,Created View.
*/
As
  Select  [DoctorID]
        , [DoctorFullName] = Cast( [FirstName] + ' ' +[LastName] As nvarchar(200))
        , [EmailAddress] As [DoctorEmailAddress]
        , [City]         As [DoctorCity]
        , [State]        As [DoctorState]
        , [Zip]          As [DoctorZip]
  From  [DoctorsSchedules].[dbo].[Doctors]

go
/* Testing Code:
 Select * From vETLDimDoctors;
*/

go
Create or Alter Procedure pETLDimDoctors
/* Desc: Inserts data Into DimDoctors using the vETLDimDoctors view
** Change Log: When,Who,What
** 20251503,GObse,Created Sproc.
*/
AS
 Begin
  Declare @RC int = 0;
  Begin Try

    -- ETL Processing Code --
    If ((Select Count(*) From DimDoctors) = 0)
     Begin Tran
      Insert Into [DWClinicReportDataGirum].[dbo].[DimDoctors]
       ([DoctorID]
      , [DoctorFullName]
      , [DoctorEmailAddress]
      , [DoctorCity]
      , [DoctorState]
      , [DoctorZip]
       )
      Select
        [DoctorID]
       ,[DoctorFullName]
       ,[DoctorEmailAddress]
       ,[DoctorCity]
       ,[DoctorState]
       ,[DoctorZip]

      FROM [dbo].[vETLDimDoctors]
    Commit Tran

    Exec pInsETLLog
           @ETLAction = 'pETLDimDoctors'
          ,@ETLLogMessage = 'DimDoctors filled';
    Set @RC = +1
  End Try
  Begin Catch
     If @@TRANCOUNT > 0 Rollback Tran;
     Declare @ErrorMessage nvarchar(1000) = Error_Message();
     Exec pInsETLLog
          @ETLAction = 'pETLDimDoctors'
         ,@ETLLogMessage = @ErrorMessage;
```

- **Develop an SSIS Package (DWClinicReportDataETL.dtsx):**

- Create an **SSIS package to automate the ETL process.**

- Configure **Data Flow tasks to extract data from the source OLTP databases.**

- Use **Staging Tables for incremental loads and ensure proper error handling.**

- Implement **Logging and Notifications to track process execution and failures.**

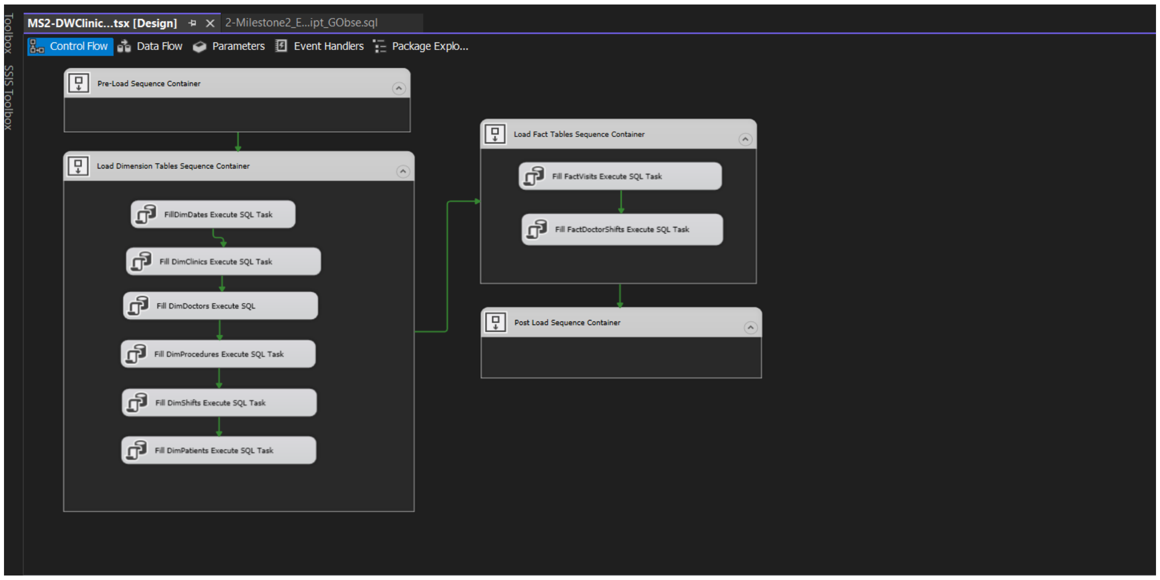
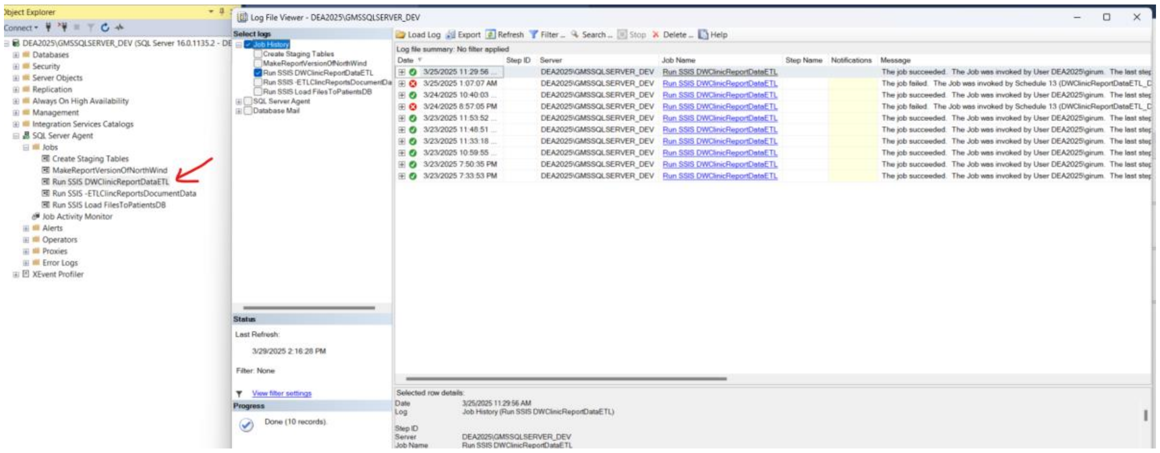

MS2-DWClinic...tsx [Design]
2-Milestone2_E...ipt_GObse.sql
Control Flow
Data Flow
Parameters
Event Handlers
Package Explo...
Toolbox
SSIS Toolbox
Pre-Load Sequence Container
Load Dimension Tables Sequence Container
FillDimDates Execute SQL Task
Fill DimClinics Execute SQL Task
Fill DimDoctors Execute SQL
Fill DimProcedures Execute SQL Task
Fill DimShifts Execute SQL Task
Fill DimPatients Execute SQL Task
Load Fact Tables Sequence Container
Fill FactVisits Execute SQL Task
Fill FactDoctorShifts Execute SQL Task
Post Load Sequence Container

Figure13, showing ssis package of the datawarehouse

- Schedule the SSIS package for execution via **SQL Server Agent** or manual triggers.


Object Explorer
Connect ▾
DEA2025\GMSSQLSERVER_DEV (SQL Server 16.0.1135.2 - DE
Databases
Security
Server Objects
Replication
Always On High Availability
Management
Integration Services Catalogs
SQL Server Agent
Jobs
Create Staging Tables
MakeReportVersionOfNorthWind
Run SSIS DWClinicReportDataETL
Run SSIS -ETLClincReportsDocumentData
Run SSIS Load FilesToPatientsDB
Job Activity Monitor
Alerts
Operators
Proxies
Error Logs
XEvent Profiler
Log File Viewer - DEA2025\GMSSQLSERVER_DEV
Select logs
Job History
Create Staging Tables
MakeReportVersionOfNorthWind
Run SSIS DWClinicReportDataETL
Run SSIS -ETLClincReportsDocumentDa
Run SSIS Load FilesToPatientsDB
SQL Server Agent
Database Mail
Load Log
Export
Refresh
Filter ...
Search ...
Stop
Delete ...
Help
View filter settings

Status
Last Refresh:
3/29/2025 2:16:28 PM
Filter: None

Progress
Done (10 records).

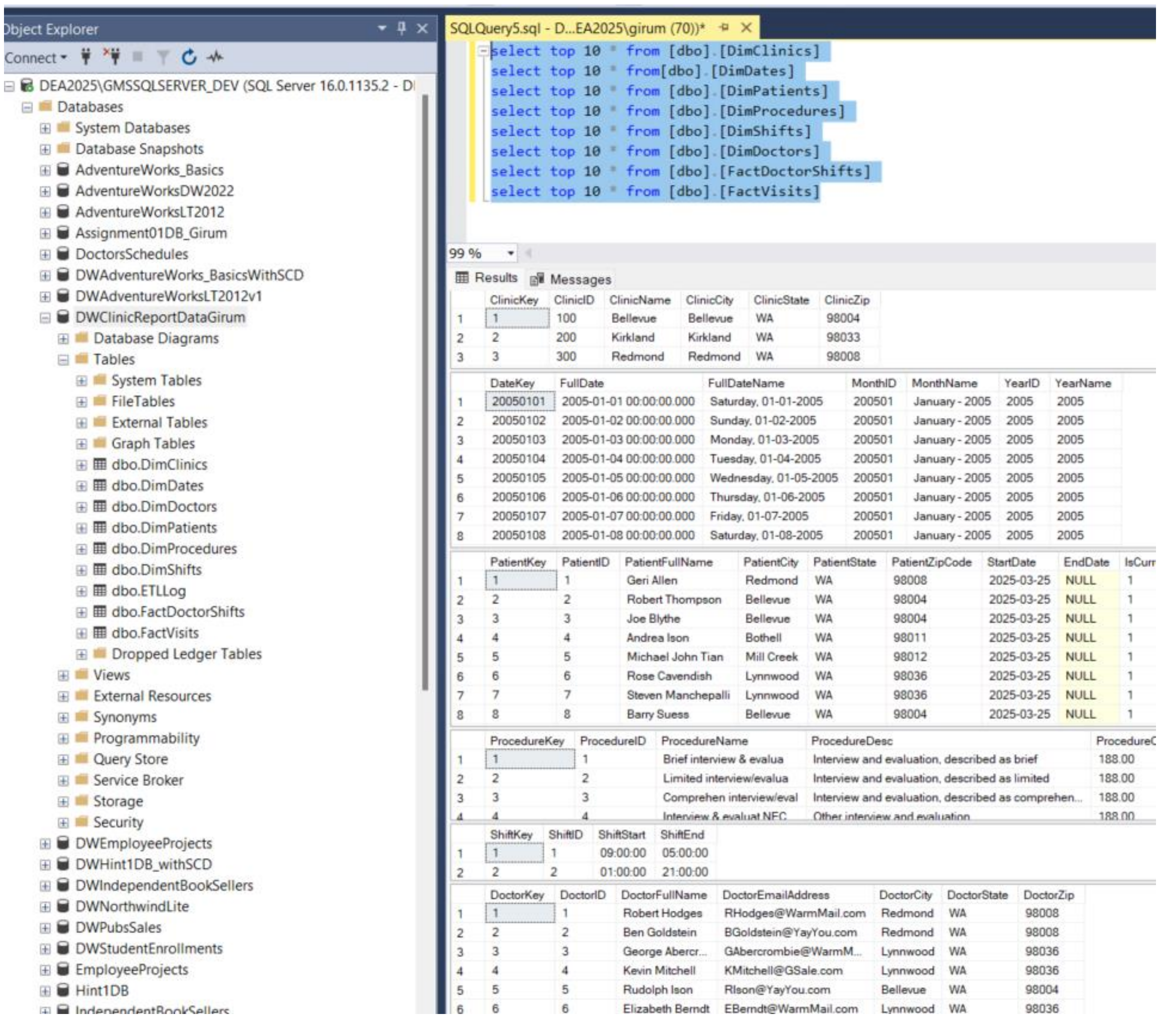
Log file summary: No filter applied

Date	Step ID	Server	Job Name	Step Name	Notifications	Message
✓ 3/25/2025 11:29:56 ...		DEA2025\GMSSQLSERVER_DEV	Run SSIS DWClinicReportDataETL			The job succeeded. The Job was invoked by User DEA2025\girum. The last ste
✗ 3/25/2025 1:07:07 AM		DEA2025\GMSSQLSERVER_DEV	Run SSIS DWClinicReportDataETL			The job failed. The Job was invoked by Schedule 13 (DWClinicReportDataETL_C
✓ 3/24/2025 10:40:03 ...		DEA2025\GMSSQLSERVER_DEV	Run SSIS DWClinicReportDataETL			The job succeeded. The Job was invoked by User DEA2025\girum. The last ste
✗ 3/24/2025 8:57:05 PM		DEA2025\GMSSQLSERVER_DEV	Run SSIS DWClinicReportDataETL			The job failed. The Job was invoked by Schedule 13 (DWClinicReportDataETL_C
✓ 3/23/2025 11:53:52 ...		DEA2025\GMSSQLSERVER_DEV	Run SSIS DWClinicReportDataETL			The job succeeded. The Job was invoked by User DEA2025\girum. The last ste
✓ 3/23/2025 11:48:51 ...		DEA2025\GMSSQLSERVER_DEV	Run SSIS DWClinicReportDataETL			The job succeeded. The Job was invoked by User DEA2025\girum. The last ste
✓ 3/23/2025 11:33:18 ...		DEA2025\GMSSQLSERVER_DEV	Run SSIS DWClinicReportDataETL			The job succeeded. The Job was invoked by User DEA2025\girum. The last ste
✓ 3/23/2025 10:59:55 ...		DEA2025\GMSSQLSERVER_DEV	Run SSIS DWClinicReportDataETL			The job succeeded. The Job was invoked by User DEA2025\girum. The last ste
✓ 3/23/2025 7:50:35 PM		DEA2025\GMSSQLSERVER_DEV	Run SSIS DWClinicReportDataETL			The job succeeded. The Job was invoked by User DEA2025\girum. The last ste
✓ 3/23/2025 7:33:53 PM		DEA2025\GMSSQLSERVER_DEV	Run SSIS DWClinicReportDataETL			The job succeeded. The Job was invoked by User DEA2025\girum. The last ste

Selected row details:

Date	3/25/2025 11:29:56 AM
Log	Job History (Run SSIS DWClinicReportDataETL)
Step ID	
Server	DEA2025\GMSSQLSERVER_DEV
Job Name	Run SSIS DWClinicReportDataETL

Figure 14, showing the ssis scheduled job and job history.



```
bject Explorer
Connect ▾
⊟ DEA2025\GMSSQLSERVER_DEV (SQL Server 16.0.1135.2 - D
  ⊟ Databases
    ⊞ System Databases
    ⊞ Database Snapshots
    ⊞ AdventureWorks_Basics
    ⊞ AdventureWorksDW2022
    ⊞ AdventureWorksLT2012
    ⊞ Assignment01DB_Girum
    ⊞ DoctorsSchedules
    ⊞ DWAdventureWorks_BasicsWithSCD
    ⊞ DWAdventureWorksLT2012v1
    ⊟ DWClinicReportDataGirum
      ⊞ Database Diagrams
      ⊟ Tables
        ⊞ System Tables
        ⊞ FileTables
        ⊞ External Tables
        ⊞ Graph Tables
        ⊞ dbo.DimClinics
        ⊞ dbo.DimDates
        ⊞ dbo.DimDoctors
        ⊞ dbo.DimPatients
        ⊞ dbo.DimProcedures
        ⊞ dbo.DimShifts
        ⊞ dbo.ETLLog
        ⊞ dbo.FactDoctorShifts
        ⊞ dbo.FactVisits
        ⊞ Dropped Ledger Tables
      ⊞ Views
      ⊞ External Resources
      ⊞ Synonyms
      ⊞ Programmability
      ⊞ Query Store
      ⊞ Service Broker
      ⊞ Storage
      ⊞ Security
    ⊞ DWEmployeeProjects
    ⊞ DWHint1DB_withSCD
    ⊞ DWIndependentBookSellers
    ⊞ DWNorthwindLite
    ⊞ DWPubsSales
    ⊞ DWStudentEnrollments
    ⊞ EmployeeProjects
    ⊞ Hint1DB
    ⊞ IndependentBookSellers
```

SQLQuery5.sql - D...EA2025\girum (70))*

```sql
select top 10 * from [dbo].[DimClinics]
select top 10 * from[dbo].[DimDates]
select top 10 * from [dbo].[DimPatients]
select top 10 * from [dbo].[DimProcedures]
select top 10 * from [dbo].[DimShifts]
select top 10 * from [dbo].[DimDoctors]
select top 10 * from [dbo].[FactDoctorShifts]
select top 10 * from [dbo].[FactVisits]
```

99 %

Results | Messages

	ClinicKey	ClinicID	ClinicName	ClinicCity	ClinicState	ClinicZip
1	1	100	Bellevue	Bellevue	WA	98004
2	2	200	Kirkland	Kirkland	WA	98033
3	3	300	Redmond	Redmond	WA	98008

	DateKey	FullDate	FullDateName	MonthID	MonthName	YearID	YearName
1	20050101	2005-01-01 00:00:00.000	Saturday, 01-01-2005	200501	January - 2005	2005	2005
2	20050102	2005-01-02 00:00:00.000	Sunday, 01-02-2005	200501	January - 2005	2005	2005
3	20050103	2005-01-03 00:00:00.000	Monday, 01-03-2005	200501	January - 2005	2005	2005
4	20050104	2005-01-04 00:00:00.000	Tuesday, 01-04-2005	200501	January - 2005	2005	2005
5	20050105	2005-01-05 00:00:00.000	Wednesday, 01-05-2005	200501	January - 2005	2005	2005
6	20050106	2005-01-06 00:00:00.000	Thursday, 01-06-2005	200501	January - 2005	2005	2005
7	20050107	2005-01-07 00:00:00.000	Friday, 01-07-2005	200501	January - 2005	2005	2005
8	20050108	2005-01-08 00:00:00.000	Saturday, 01-08-2005	200501	January - 2005	2005	2005

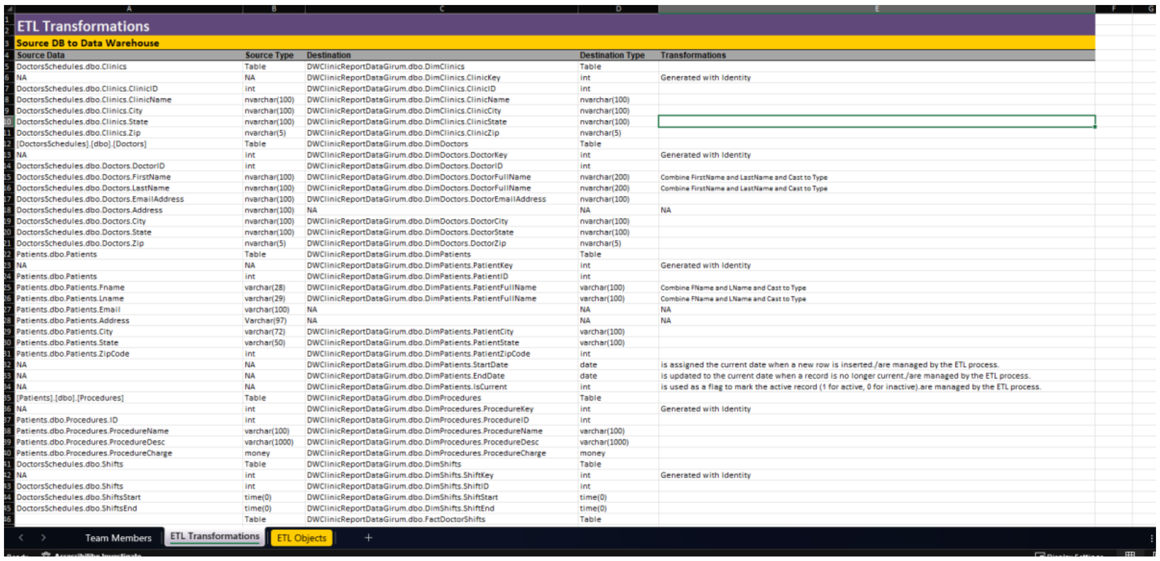
	PatientKey	PatientID	PatientFullName	PatientCity	PatientState	PatientZipCode	StartDate	EndDate	IsCurr
1	1	1	Geri Allen	Redmond	WA	98008	2025-03-25	NULL	1
2	2	2	Robert Thompson	Bellevue	WA	98004	2025-03-25	NULL	1
3	3	3	Joe Blythe	Bellevue	WA	98004	2025-03-25	NULL	1
4	4	4	Andrea Ison	Bothell	WA	98011	2025-03-25	NULL	1
5	5	5	Michael John Tian	Mill Creek	WA	98012	2025-03-25	NULL	1
6	6	6	Rose Cavendish	Lynnwood	WA	98036	2025-03-25	NULL	1
7	7	7	Steven Manchepalli	Lynnwood	WA	98036	2025-03-25	NULL	1
8	8	8	Barry Suess	Bellevue	WA	98004	2025-03-25	NULL	1

	ProcedureKey	ProcedureID	ProcedureName	ProcedureDesc	ProcedureC
1	1	1	Brief interview & evalua	Interview and evaluation, described as brief	188.00
2	2	2	Limited interview/evalua	Interview and evaluation, described as limited	188.00
3	3	3	Comprehen interview/eval	Interview and evaluation, described as comprehen...	188.00
4	4	4	Interview & evaluat NEC	Other interview and evaluation	188.00

	ShiftKey	ShiftID	ShiftStart	ShiftEnd
1	1	1	09:00:00	05:00:00
2	2	2	01:00:00	21:00:00

	DoctorKey	DoctorID	DoctorFullName	DoctorEmailAddress	DoctorCity	DoctorState	DoctorZip
1	1	1	Robert Hodges	RHodges@WarmMail.com	Redmond	WA	98008
2	2	2	Ben Goldstein	BGoldstein@YayYou.com	Redmond	WA	98008
3	3	3	George Abercr...	GAbercrombie@WarmM...	Lynnwood	WA	98036
4	4	4	Kevin Mitchell	KMitchell@GSale.com	Lynnwood	WA	98036
5	5	5	Rudolph Ison	RIson@YayYou.com	Bellevue	WA	98004
6	6	6	Elizabeth Berndt	EBerndt@WarmMail.com	Lynnwood	WA	98036

Figure15, demonstrating the data in DataClinicReport datawarehouse tables



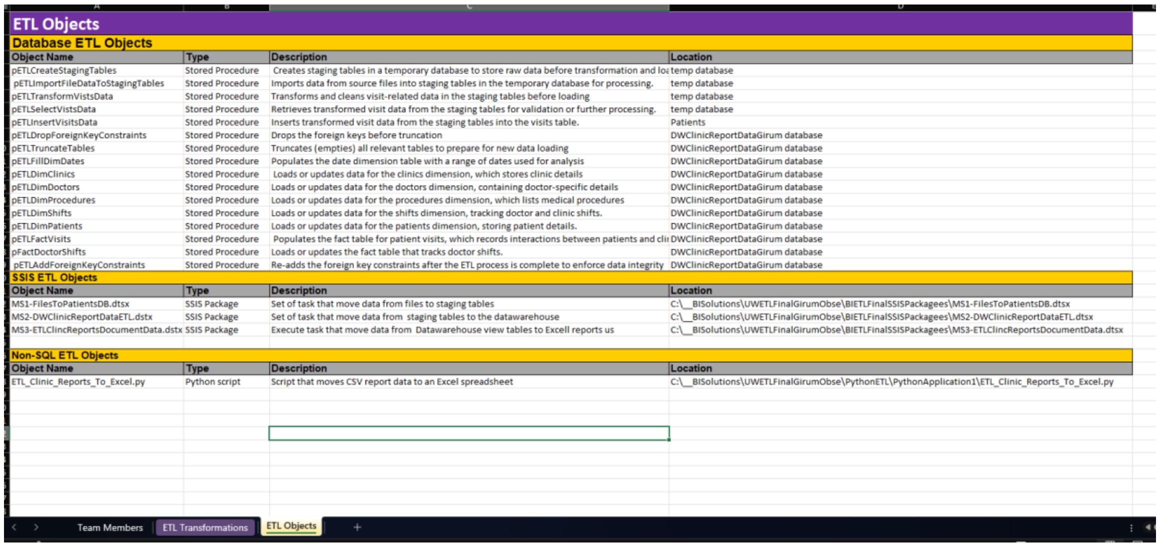
ETL Transformations

Source DB to Data Warehouse

Source Data	Source Type	Destination	Destination Type	Transformations
DoctorsSchedules.dbo.Clinics	Table	DWClinicReportDataGirum.dbo.DimClinics	Table	
NA	NA	DWClinicReportDataGirum.dbo.DimClinics.ClinicKey	int	Generated with Identity
DoctorsSchedules.dbo.Clinics.ClinicID	int	DWClinicReportDataGirum.dbo.DimClinics.ClinicID	int	
DoctorsSchedules.dbo.Clinics.ClinicName	nvarchar(100)	DWClinicReportDataGirum.dbo.DimClinics.ClinicName	nvarchar(100)	
DoctorsSchedules.dbo.Clinics.City	nvarchar(100)	DWClinicReportDataGirum.dbo.DimClinics.ClinicCity	nvarchar(100)	
DoctorsSchedules.dbo.Clinics.State	nvarchar(100)	DWClinicReportDataGirum.dbo.DimClinics.ClinicState	nvarchar(100)	
DoctorsSchedules.dbo.Clinics.Zip	nvarchar(5)	DWClinicReportDataGirum.dbo.DimClinics.ClinicZip	nvarchar(5)	
[DoctorsSchedules].[dbo].[Doctors]	Table	DWClinicReportDataGirum.dbo.DimDoctors	Table	
NA	int	DWClinicReportDataGirum.dbo.DimDoctors.DoctorKey	int	Generated with Identity
DoctorsSchedules.dbo.Doctors.DoctorID	int	DWClinicReportDataGirum.dbo.DimDoctors.DoctorID	int	
DoctorsSchedules.dbo.Doctors.FirstName	nvarchar(100)	DWClinicReportDataGirum.dbo.DimDoctors.DoctorFullName	nvarchar(200)	Combine FirstName and LastName and Cast to Type
DoctorsSchedules.dbo.Doctors.LastName	nvarchar(100)	DWClinicReportDataGirum.dbo.DimDoctors.DoctorFullName	nvarchar(200)	Combine FirstName and LastName and Cast to Type
DoctorsSchedules.dbo.Doctors.EmailAddress	nvarchar(100)	DWClinicReportDataGirum.dbo.DimDoctors.DoctorEmailAddress	nvarchar(100)	
DoctorsSchedules.dbo.Doctors.Address	nvarchar(100)	NA	NA	NA
DoctorsSchedules.dbo.Doctors.City	nvarchar(100)	DWClinicReportDataGirum.dbo.DimDoctors.DoctorCity	nvarchar(100)	
DoctorsSchedules.dbo.Doctors.State	nvarchar(100)	DWClinicReportDataGirum.dbo.DimDoctors.DoctorState	nvarchar(100)	
DoctorsSchedules.dbo.Doctors.Zip	nvarchar(5)	DWClinicReportDataGirum.dbo.DimDoctors.DoctorZip	nvarchar(5)	
Patients.dbo.Patients	Table	DWClinicReportDataGirum.dbo.DimPatients	Table	
NA	NA	DWClinicReportDataGirum.dbo.DimPatients.PatientKey	int	Generated with Identity
Patients.dbo.Patients	int	DWClinicReportDataGirum.dbo.DimPatients.PatientID	int	
Patients.dbo.Patients.Fname	varchar(28)	DWClinicReportDataGirum.dbo.DimPatients.PatientFullName	varchar(100)	Combine FName and LName and Cast to Type
Patients.dbo.Patients.Lname	varchar(29)	DWClinicReportDataGirum.dbo.DimPatients.PatientFullName	varchar(100)	Combine FName and LName and Cast to Type
Patients.dbo.Patients.Email	varchar(100)	NA	NA	NA
Patients.dbo.Patients.Address	Varchar(97)	NA	NA	NA
Patients.dbo.Patients.City	varchar(72)	DWClinicReportDataGirum.dbo.DimPatients.PatientCity	varchar(100)	
Patients.dbo.Patients.State	varchar(50)	DWClinicReportDataGirum.dbo.DimPatients.PatientState	varchar(100)	
Patients.dbo.Patients.ZipCode	int	DWClinicReportDataGirum.dbo.DimPatients.PatientZipCode	int	
NA	NA	DWClinicReportDataGirum.dbo.DimPatients.StartDate	date	is assigned the current date when a new row is inserted./are managed by the ETL process.
NA	NA	DWClinicReportDataGirum.dbo.DimPatients.EndDate	date	is updated to the current date when a record is no longer current./are managed by the ETL process.
NA	NA	DWClinicReportDataGirum.dbo.DimPatients.IsCurrent	int	is used as a flag to mark the active record (1 for active, 0 for inactive).are managed by the ETL process.
[Patients].[dbo].[Procedures]	Table	DWClinicReportDataGirum.dbo.DimProcedures	Table	
NA	int	DWClinicReportDataGirum.dbo.DimProcedures.ProcedureKey	int	Generated with Identity
Patients.dbo.Procedures.ID	int	DWClinicReportDataGirum.dbo.DimProcedures.ProcedureID	int	
Patients.dbo.Procedures.ProcedureName	varchar(100)	DWClinicReportDataGirum.dbo.DimProcedures.ProcedureName	varchar(100)	
Patients.dbo.Procedures.ProcedureDesc	varchar(1000)	DWClinicReportDataGirum.dbo.DimProcedures.ProcedureDesc	varchar(1000)	
Patients.dbo.Procedures.ProcedureCharge	money	DWClinicReportDataGirum.dbo.DimProcedures.ProcedureCharge	money	
DoctorsSchedules.dbo.Shifts	Table	DWClinicReportDataGirum.dbo.DimShifts	Table	
NA	int	DWClinicReportDataGirum.dbo.DimShifts.ShiftKey	int	Generated with Identity
DoctorsSchedules.dbo.Shifts	int	DWClinicReportDataGirum.dbo.DimShifts.ShiftID	int	
DoctorsSchedules.dbo.ShiftsStart	time(0)	DWClinicReportDataGirum.dbo.DimShifts.ShiftStart	time(0)	
DoctorsSchedules.dbo.ShiftsEnd	time(0)	DWClinicReportDataGirum.dbo.DimShifts.ShiftEnd	time(0)	
	Table	DWClinicReportDataGirum.dbo.FactDoctorShifts	Table	

Team Members | ETL Transformations | ETL Objects

Figure16, demonstrating the metadata worksheets



ETL Objects

Database ETL Objects

Object Name	Type	Description	Location
pETLCreateStagingTables	Stored Procedure	Creates staging tables in a temporary database to store raw data before transformation and loa	temp database
pETLImportFileDataToStagingTables	Stored Procedure	Imports data from source files into staging tables in the temporary database for processing.	temp database
pETLTransformVistsData	Stored Procedure	Transforms and cleans visit-related data in the staging tables before loading	temp database
pETLSelectVistsData	Stored Procedure	Retrieves transformed visit data from the staging tables for validation or further processing.	temp database
pETLInsertVisitsData	Stored Procedure	Inserts transformed visit data from the staging tables into the visits table.	Patients
pETLDropForeignKeyConstraints	Stored Procedure	Drops the foreign keys before truncation	DWClinicReportDataGirum database
pETLTruncateTables	Stored Procedure	Truncates (empties) all relevant tables to prepare for new data loading	DWClinicReportDataGirum database
pETLFillDimDates	Stored Procedure	Populates the date dimension table with a range of dates used for analysis	DWClinicReportDataGirum database
pETLDimClinics	Stored Procedure	Loads or updates data for the clinics dimension, which stores clinic details	DWClinicReportDataGirum database
pETLDimDoctors	Stored Procedure	Loads or updates data for the doctors dimension, containing doctor-specific details	DWClinicReportDataGirum database
pETLDimProcedures	Stored Procedure	Loads or updates data for the procedures dimension, which lists medical procedures	DWClinicReportDataGirum database
pETLDimShifts	Stored Procedure	Loads or updates data for the shifts dimension, tracking doctor and clinic shifts.	DWClinicReportDataGirum database
pETLDimPatients	Stored Procedure	Loads or updates data for the patients dimension, storing patient details.	DWClinicReportDataGirum database
pETLFactVisits	Stored Procedure	Populates the fact table for patient visits, which records interactions between patients and cli	DWClinicReportDataGirum database
pFactDoctorShifts	Stored Procedure	Loads or updates the fact table that tracks doctor shifts.	DWClinicReportDataGirum database
pETLAddForeignKeyConstraints	Stored Procedure	Re-adds the foreign key constraints after the ETL process is complete to enforce data integrity	DWClinicReportDataGirum database

SSIS ETL Objects

Object Name	Type	Description	Location
MS1-FilesToPatientsDB.dtsx	SSIS Package	Set of task that move data from files to staging tables	C:\__BISolutions\UWETLFinalGirumObse\BIETLFinalSSISPackagees\MS1-FilesToPatientsDB.dtsx
MS2-DWClinicReportDataETL.dtsx	SSIS Package	Set of task that move data from staging tables to the datawarehouse	C:\__BISolutions\UWETLFinalGirumObse\BIETLFinalSSISPackagees\MS2-DWClinicReportDataETL.dtsx
MS3-ETLClincReportsDocumentData.dtsx	SSIS Package	Execute task that move data from Datawarehouse view tables to Excell reports us	C:\__BISolutions\UWETLFinalGirumObse\BIETLFinalSSISPackagees\MS3-ETLClincReportsDocumentData.dtsx

Non-SQL ETL Objects

Object Name	Type	Description	Location
ETL_Clinic_Reports_To_Excel.py	Python script	Script that moves CSV report data to an Excel spreadsheet	C:\__BISolutions\UWETLFinalGirumObse\PythonETL\PythonApplication1\ETL_Clinic_Reports_To_Excel.py

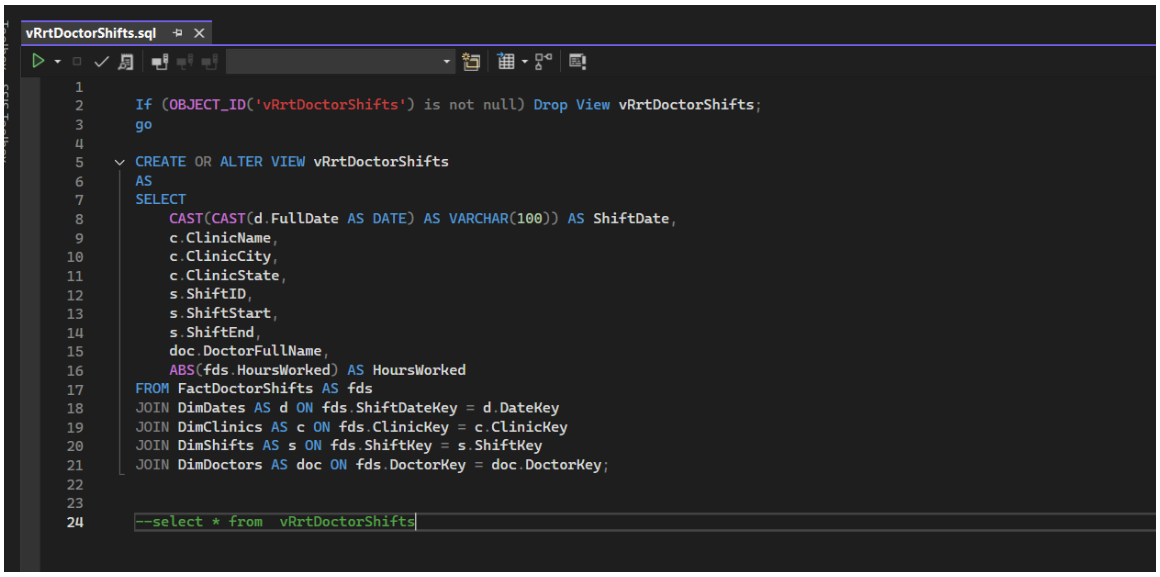
6. Milestone 3: Non-SQL ETL

Process Steps

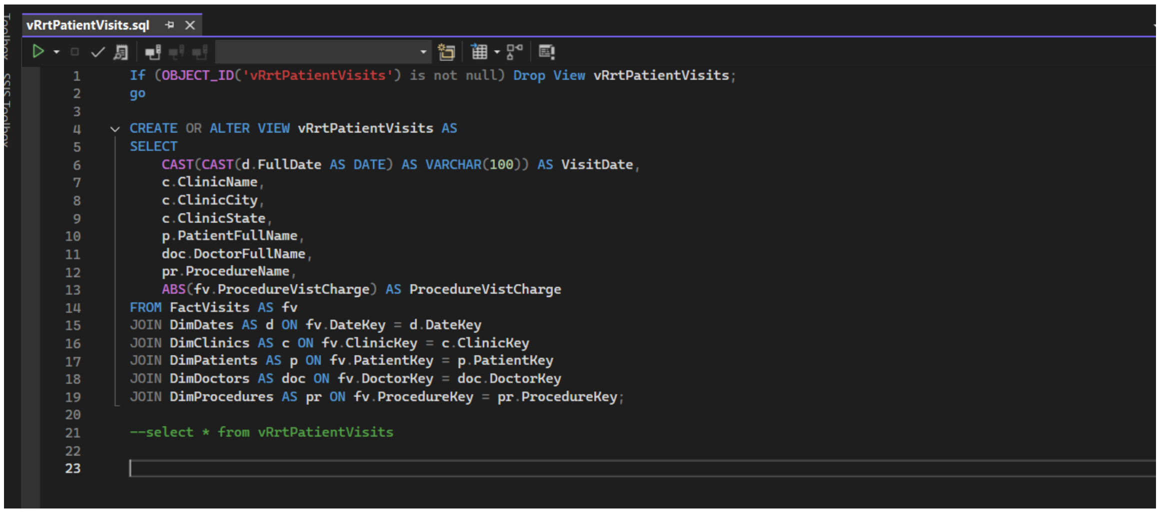
Creating SQL Views

- **Develop two SQL views in the DWClinicReportData database:**
  - **One for doctor shifts data.**
  - **One for patient visits data.**
- **Ensure the views contain relevant columns for analytical reporting.**
- **Validate the views by querying sample data.**

Sql code for the views..



```sql

If (OBJECT_ID('vRrtDoctorShifts') is not null) Drop View vRrtDoctorShifts;
go

CREATE OR ALTER VIEW vRrtDoctorShifts
AS
SELECT
    CAST(CAST(d.FullDate AS DATE) AS VARCHAR(100)) AS ShiftDate,
    c.ClinicName,
    c.ClinicCity,
    c.ClinicState,
    s.ShiftID,
    s.ShiftStart,
    s.ShiftEnd,
    doc.DoctorFullName,
    ABS(fds.HoursWorked) AS HoursWorked
FROM FactDoctorShifts AS fds
JOIN DimDates AS d ON fds.ShiftDateKey = d.DateKey
JOIN DimClinics AS c ON fds.ClinicKey = c.ClinicKey
JOIN DimShifts AS s ON fds.ShiftKey = s.ShiftKey
JOIN DimDoctors AS doc ON fds.DoctorKey = doc.DoctorKey;


--select * from  vRrtDoctorShifts
```

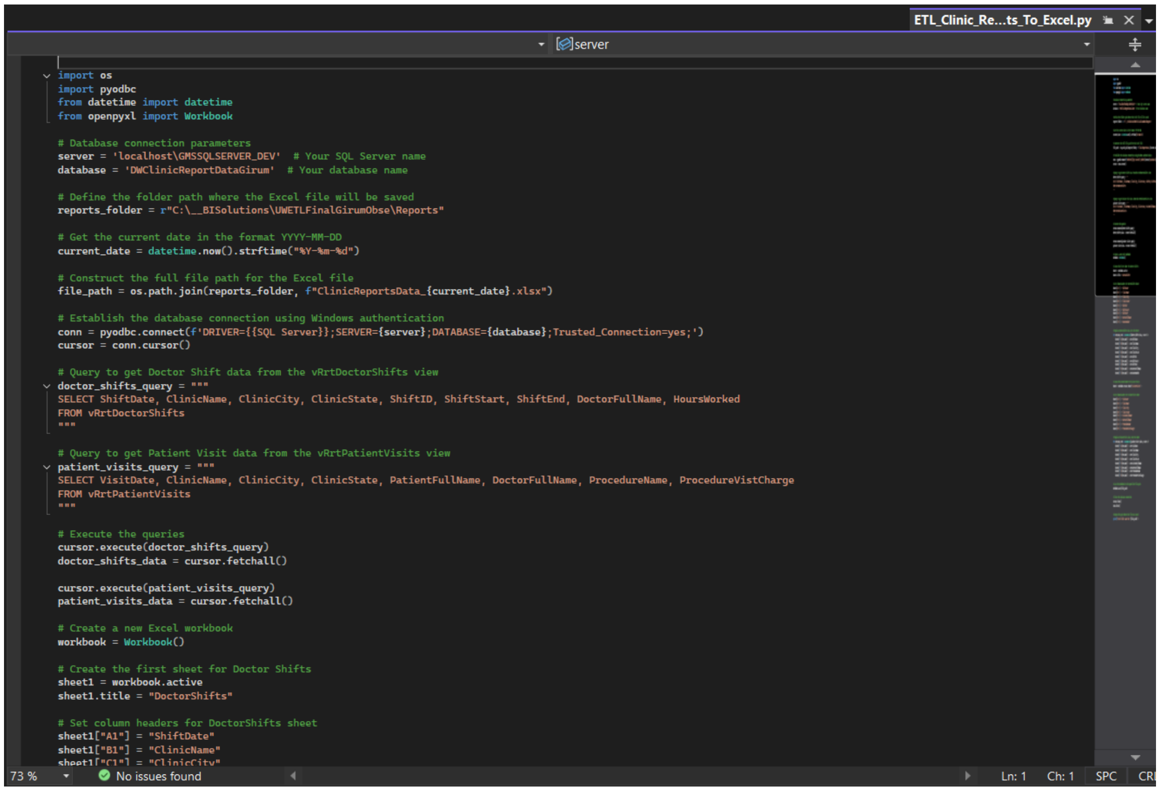


```sql
If (OBJECT_ID('vRrtPatientVisits') is not null) Drop View vRrtPatientVisits;
go

CREATE OR ALTER VIEW vRrtPatientVisits AS
SELECT
    CAST(CAST(d.FullDate AS DATE) AS VARCHAR(100)) AS VisitDate,
    c.ClinicName,
    c.ClinicCity,
    c.ClinicState,
    p.PatientFullName,
    doc.DoctorFullName,
    pr.ProcedureName,
    ABS(fv.ProcedureVistCharge) AS ProcedureVistCharge
FROM FactVisits AS fv
JOIN DimDates AS d ON fv.DateKey = d.DateKey
JOIN DimClinics AS c ON fv.ClinicKey = c.ClinicKey
JOIN DimPatients AS p ON fv.PatientKey = p.PatientKey
JOIN DimDoctors AS doc ON fv.DoctorKey = doc.DoctorKey
JOIN DimProcedures AS pr ON fv.ProcedureKey = pr.ProcedureKey;

--select * from vRrtPatientVisits

```

Extract, Transform, and Load (ETL) using Python

- **Develop a Python script to perform ETL operations:**
  - **Extract data from the SQL views.**
  - **Transform the data as needed (e.g., date formatting, column renaming).**
  - **Load the data into an Excel spreadsheet named ClinicReportsData_.xlsx.**
- **Store the Excel file in the solution's Reports folder.**
- **Validate that the data is correctly loaded into the two worksheets (one per view).**

Screernshot of the python script for the ETL for Clinic Reports



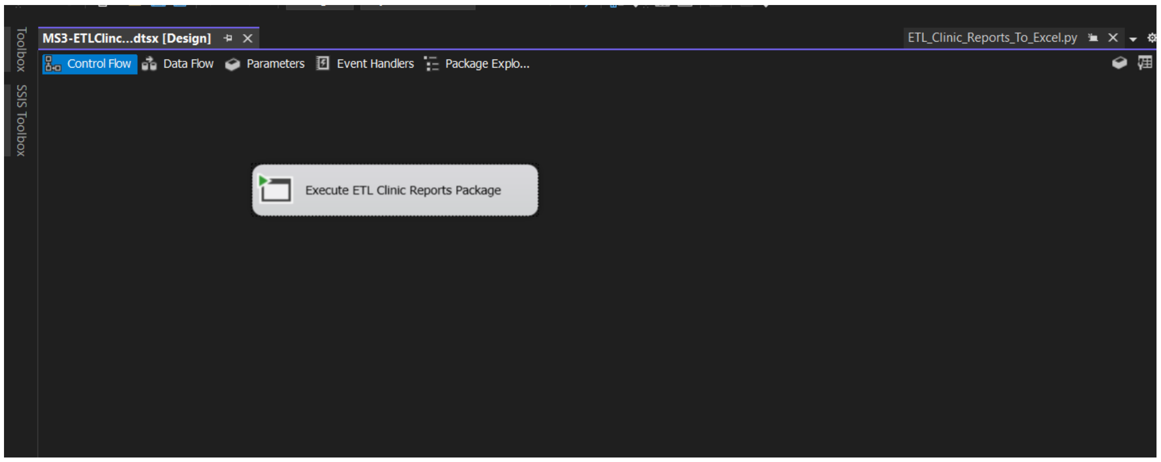
```python
import os
import pyodbc
from datetime import datetime
from openpyxl import Workbook

# Database connection parameters
server = 'localhost\GMSSQLSERVER_DEV'  # Your SQL Server name
database = 'DWClinicReportDataGirum'  # Your database name

# Define the folder path where the Excel file will be saved
reports_folder = r"C:\__BISolutions\UWETLFinalGirumObse\Reports"

# Get the current date in the format YYYY-MM-DD
current_date = datetime.now().strftime("%Y-%m-%d")

# Construct the full file path for the Excel file
file_path = os.path.join(reports_folder, f"ClinicReportsData_{current_date}.xlsx")

# Establish the database connection using Windows authentication
conn = pyodbc.connect(f'DRIVER={{SQL Server}};SERVER={server};DATABASE={database};Trusted_Connection=yes;')
cursor = conn.cursor()

# Query to get Doctor Shift data from the vRrtDoctorShifts view
doctor_shifts_query = """
SELECT ShiftDate, ClinicName, ClinicCity, ClinicState, ShiftID, ShiftStart, ShiftEnd, DoctorFullName, HoursWorked
FROM vRrtDoctorShifts
"""

# Query to get Patient Visit data from the vRrtPatientVisits view
patient_visits_query = """
SELECT VisitDate, ClinicName, ClinicCity, ClinicState, PatientFullName, DoctorFullName, ProcedureName, ProcedureVistCharge
FROM vRrtPatientVisits
"""

# Execute the queries
cursor.execute(doctor_shifts_query)
doctor_shifts_data = cursor.fetchall()

cursor.execute(patient_visits_query)
patient_visits_data = cursor.fetchall()

# Create a new Excel workbook
workbook = Workbook()

# Create the first sheet for Doctor Shifts
sheet1 = workbook.active
sheet1.title = "DoctorShifts"

# Set column headers for DoctorShifts sheet
sheet1["A1"] = "ShiftDate"
sheet1["B1"] = "ClinicName"
sheet1["C1"] = "ClinicCity"
```

SSIS Package Integration

- **Develop an SSIS package (ETLClinicReportsDocumentData.dtsx):**
    - **Automate execution of the Python script.**
    - **Handle errors and log execution results.**
    - **Ensure proper folder and file management for storing output reports.**

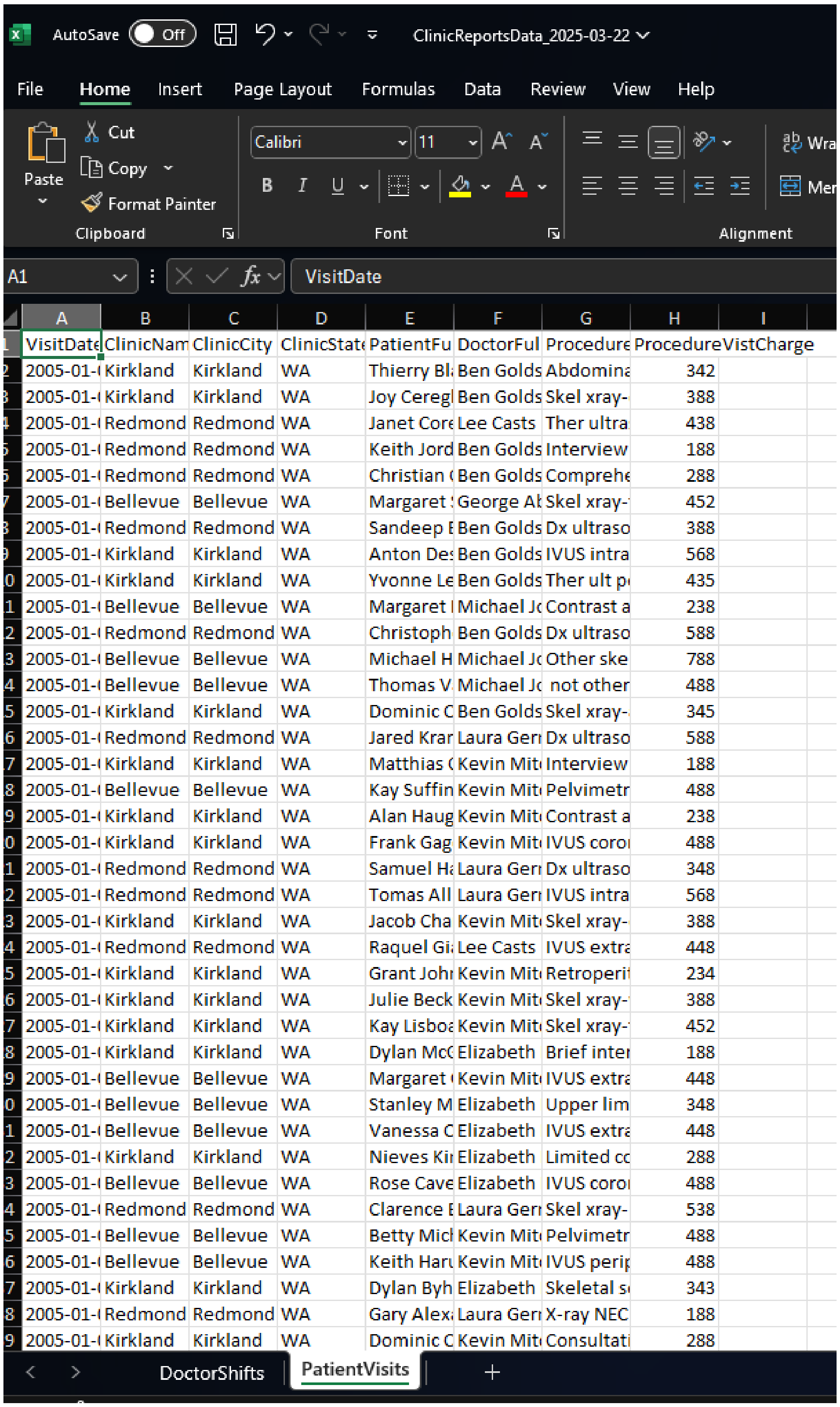
Updating ETL Process Metadata

- **Update the Excel metadata spreadsheet:**
    - **Document source, destination, and transformation rules.**
    - **Ensure all tabs are updated as per the ETL workflow.**
- **Verify the metadata consistency with the implemented ETL process.**


MS3-ETLClinc...dtsx [Design]
Control Flow
Data Flow
Parameters
Event Handlers
Package Explo...
ETL_Clinic_Reports_To_Excel.py
Toolbox
SSIS Toolbox
Execute ETL Clinic Reports Package

Ssis package the executes the python ETL script

Excel worksheet of the clinic reports



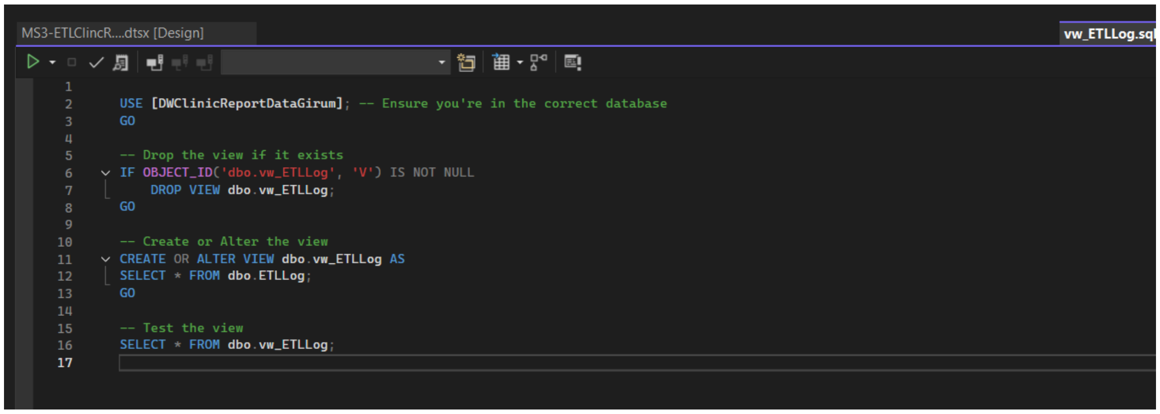
A	B	C	D	E	F	G	H	I
VisitDate	ClinicNam	ClinicCity	ClinicStat	PatientFu	DoctorFul	Procedure	Procedure	VistCharge
2005-01-(	Kirkland	Kirkland	WA	Thierry Bl	Ben Golds	Abdomina	342	
2005-01-(	Kirkland	Kirkland	WA	Joy Ceregl	Ben Golds	Skel xray-	388	
2005-01-(	Redmond	Redmond	WA	Janet Core	Lee Casts	Ther ultra	438	
2005-01-(	Redmond	Redmond	WA	Keith Jord	Ben Golds	Interview	188	
2005-01-(	Redmond	Redmond	WA	Christian (	Ben Golds	Comprehe	288	
2005-01-(	Bellevue	Bellevue	WA	Margaret !	George Al	Skel xray-'	452	
2005-01-(	Redmond	Redmond	WA	Sandeep E	Ben Golds	Dx ultraso	388	
2005-01-(	Kirkland	Kirkland	WA	Anton Des	Ben Golds	IVUS intra	568	
2005-01-(	Kirkland	Kirkland	WA	Yvonne Le	Ben Golds	Ther ult p	435	
2005-01-(	Bellevue	Bellevue	WA	Margaret I	Michael Jc	Contrast a	238	
2005-01-(	Redmond	Redmond	WA	Christoph	Ben Golds	Dx ultraso	588	
2005-01-(	Bellevue	Bellevue	WA	Michael H	Michael Jc	Other ske	788	
2005-01-(	Bellevue	Bellevue	WA	Thomas V.	Michael Jc	not other	488	
2005-01-(	Kirkland	Kirkland	WA	Dominic C	Ben Golds	Skel xray-	345	
2005-01-(	Redmond	Redmond	WA	Jared Krar	Laura Gerr	Dx ultraso	588	
2005-01-(	Kirkland	Kirkland	WA	Matthias (	Kevin Mit	Interview	188	
2005-01-(	Bellevue	Bellevue	WA	Kay Suffin	Kevin Mit	Pelvimetr	488	
2005-01-(	Kirkland	Kirkland	WA	Alan Haug	Kevin Mit	Contrast a	238	
2005-01-(	Kirkland	Kirkland	WA	Frank Gag	Kevin Mit	IVUS coro	488	
2005-01-(	Redmond	Redmond	WA	Samuel Ha	Laura Gerr	Dx ultraso	348	
2005-01-(	Redmond	Redmond	WA	Tomas All	Laura Gerr	IVUS intra	568	
2005-01-(	Kirkland	Kirkland	WA	Jacob Cha	Kevin Mit	Skel xray-	388	
2005-01-(	Redmond	Redmond	WA	Raquel Gi	Lee Casts	IVUS extra	448	
2005-01-(	Kirkland	Kirkland	WA	Grant Johr	Kevin Mit	Retroperi	234	
2005-01-(	Kirkland	Kirkland	WA	Julie Beck	Kevin Mit	Skel xray-'	388	
2005-01-(	Kirkland	Kirkland	WA	Kay Lisboa	Kevin Mit	Skel xray-'	452	
2005-01-(	Kirkland	Kirkland	WA	Dylan McC	Elizabeth	Brief inter	188	
2005-01-(	Bellevue	Bellevue	WA	Margaret (	Kevin Mit	IVUS extra	448	
2005-01-(	Bellevue	Bellevue	WA	Stanley M	Elizabeth	Upper lim	348	
2005-01-(	Bellevue	Bellevue	WA	Vanessa C	Elizabeth	IVUS extra	448	
2005-01-(	Kirkland	Kirkland	WA	Nieves Kir	Elizabeth	Limited cc	288	
2005-01-(	Bellevue	Bellevue	WA	Rose Cave	Elizabeth	IVUS coro	488	
2005-01-(	Redmond	Redmond	WA	Clarence E	Laura Gerr	Skel xray-	538	
2005-01-(	Bellevue	Bellevue	WA	Betty Micl	Kevin Mit	Pelvimetr	488	
2005-01-(	Bellevue	Bellevue	WA	Keith Haru	Kevin Mit	IVUS perip	488	
2005-01-(	Kirkland	Kirkland	WA	Dylan Byh	Elizabeth	Skeletal s	343	
2005-01-(	Redmond	Redmond	WA	Gary Alex	Laura Gerr	X-ray NEC	188	
2005-01-(	Kirkland	Kirkland	WA	Dominic C	Kevin Mit	Consultati	288	

7. Milestone 4: Automation, Reports, and Documentation

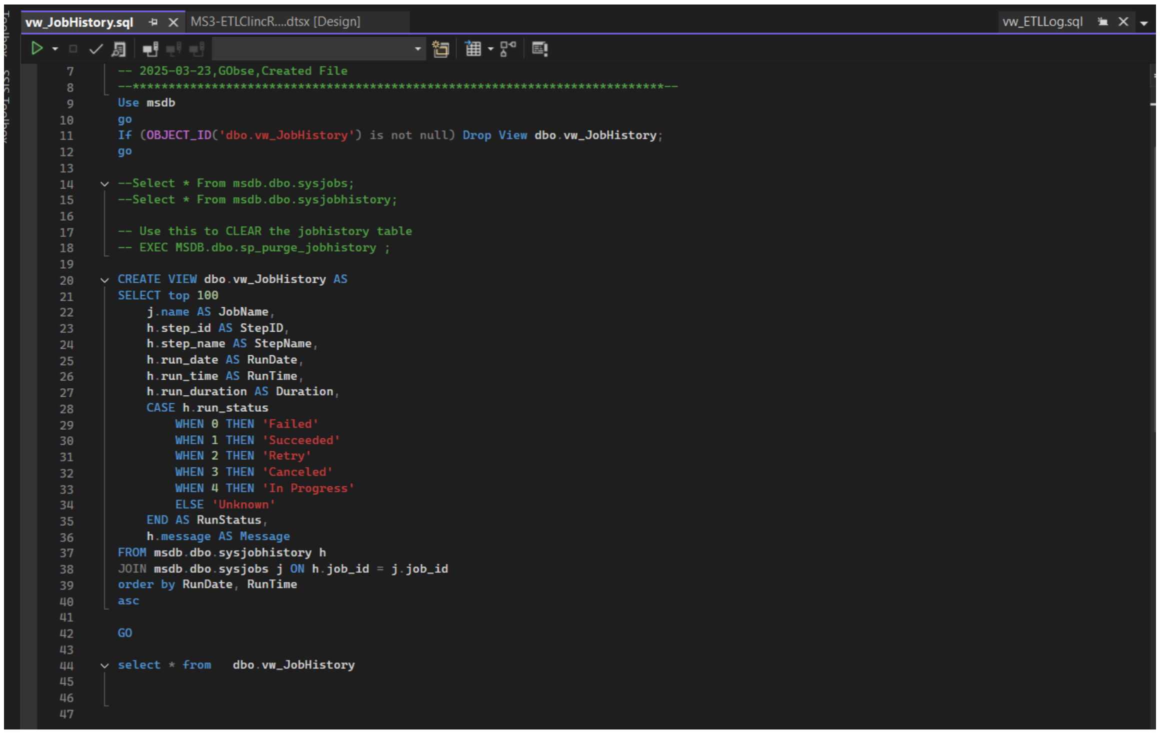
Process Steps

Creating SQL Reporting Views

• Develop ETL report views to extract data from the ETL log and MSDB jobs tables.
• Ensure the views include relevant columns for tracking ETL execution status, errors, and processing times.
• Validate the views by running queries to confirm correct data extraction and filtering.
• Optimize performance by adding appropriate indexing where necessary.



```sql

USE [DWClinicReportDataGirum]; -- Ensure you're in the correct database
GO

-- Drop the view if it exists
IF OBJECT_ID('dbo.vw_ETLLog', 'V') IS NOT NULL
    DROP VIEW dbo.vw_ETLLog;
GO

-- Create or Alter the view
CREATE OR ALTER VIEW dbo.vw_ETLLog AS
SELECT * FROM dbo.ETLLog;
GO

-- Test the view
SELECT * FROM dbo.vw_ETLLog;

```



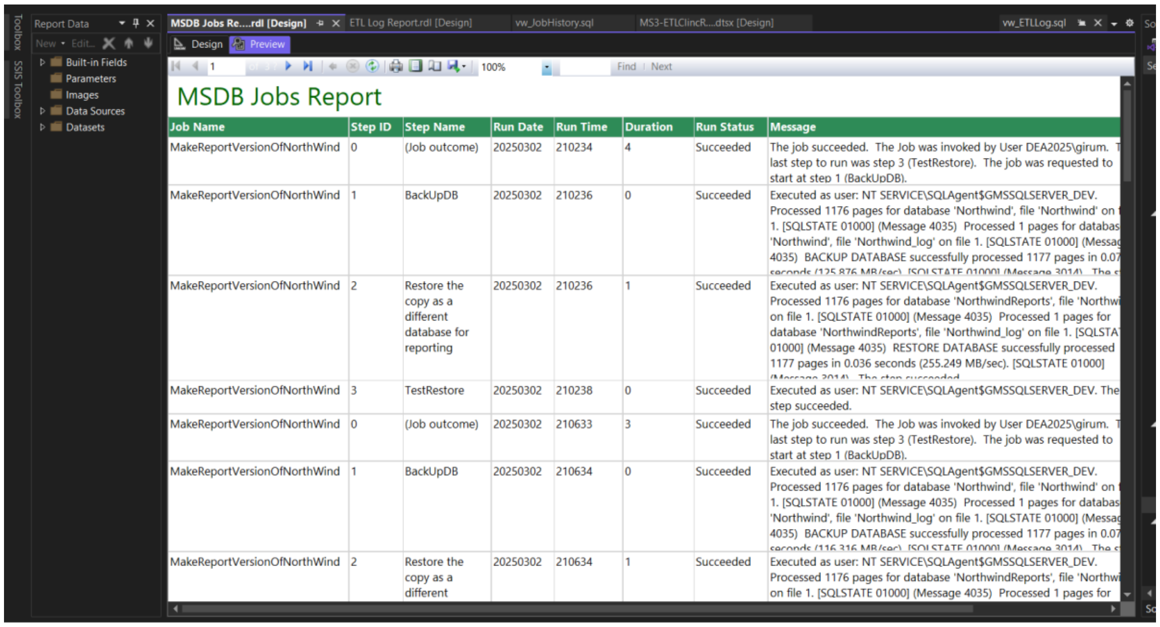
```sql
-- 2025-03-23,GObse,Created File
--*****************************************************************************--
Use msdb
go
If (OBJECT_ID('dbo.vw_JobHistory') is not null) Drop View dbo.vw_JobHistory;
go

--Select * From msdb.dbo.sysjobs;
--Select * From msdb.dbo.sysjobhistory;

-- Use this to CLEAR the jobhistory table
-- EXEC MSDB.dbo.sp_purge_jobhistory ;

CREATE VIEW dbo.vw_JobHistory AS
SELECT top 100
    j.name AS JobName,
    h.step_id AS StepID,
    h.step_name AS StepName,
    h.run_date AS RunDate,
    h.run_time AS RunTime,
    h.run_duration AS Duration,
    CASE h.run_status
        WHEN 0 THEN 'Failed'
        WHEN 1 THEN 'Succeeded'
        WHEN 2 THEN 'Retry'
        WHEN 3 THEN 'Canceled'
        WHEN 4 THEN 'In Progress'
        ELSE 'Unknown'
    END AS RunStatus,
    h.message AS Message
FROM msdb.dbo.sysjobhistory h
JOIN msdb.dbo.sysjobs j ON h.job_id = j.job_id
order by RunDate, RunTime
asc

GO

select * from   dbo.vw_JobHistory



```

Figure20, sql script for the views vw_ETL Log and vw_JobHistory

Creating ETL Dashboard Reports with SSRS

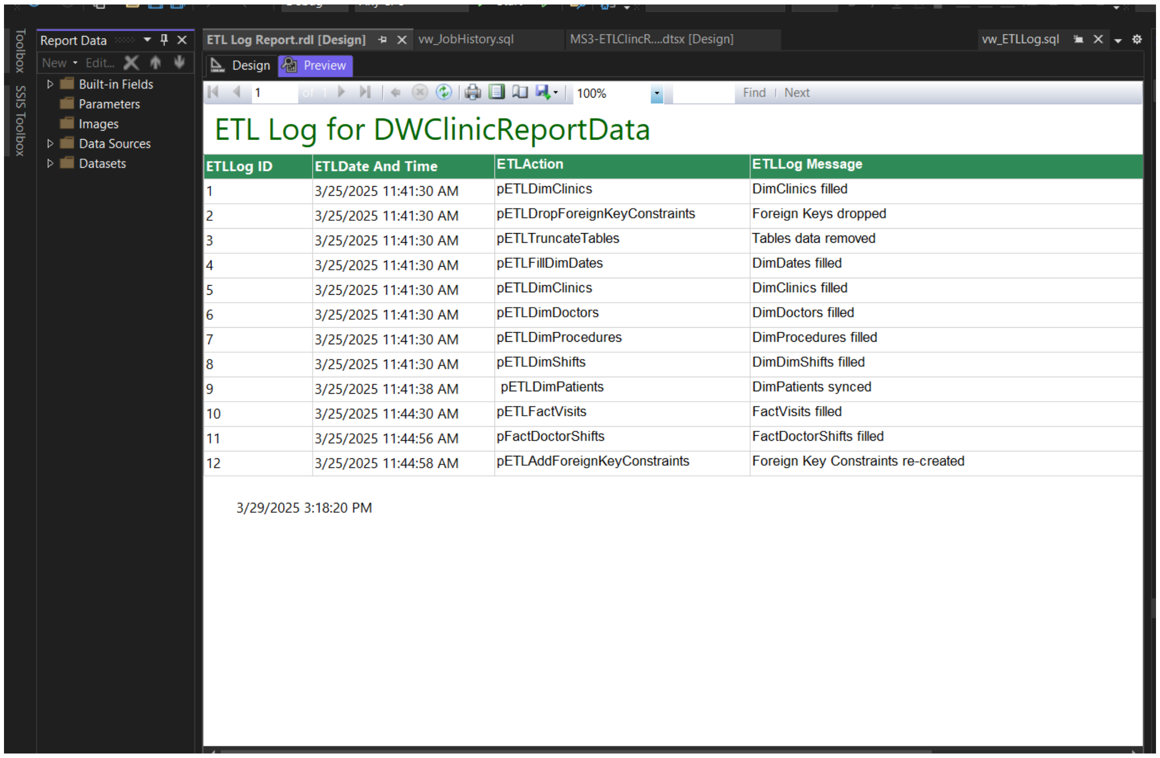
Create an ETL report that shows the contents of the ETL log and MSDB jobs table.

- The report should display execution details, success/failure statuses, error messages, and timestamps.
- Use appropriate filters and sorting options to ensure the report provides useful insights.
- Ensure that your Data Sources and DataSets are properly configured and named according to their respective database objects.



MSDB Jobs Report

Job Name	Step ID	Step Name	Run Date	Run Time	Duration	Run Status	Message
MakeReportVersionOfNorthWind	0	(Job outcome)	20250302	210234	4	Succeeded	The job succeeded. The Job was invoked by User DEA2025\girum. T last step to run was step 3 (TestRestore). The job was requested to start at step 1 (BackUpDB).
MakeReportVersionOfNorthWind	1	BackUpDB	20250302	210236	0	Succeeded	Executed as user: NT SERVICE\SQLAgent$GMSSQLSERVER_DEV. Processed 1176 pages for database 'Northwind', file 'Northwind' on 1. [SQLSTATE 01000] (Message 4035) Processed 1 pages for databas 'Northwind', file 'Northwind_log' on file 1. [SQLSTATE 01000] (Messag 4035) BACKUP DATABASE successfully processed 1177 pages in 0.07 seconds (125.876 MB/sec). [SQLSTATE 01000] (Message 3014) The s
MakeReportVersionOfNorthWind	2	Restore the copy as a different database for reporting	20250302	210236	1	Succeeded	Executed as user: NT SERVICE\SQLAgent$GMSSQLSERVER_DEV. Processed 1176 pages for database 'NorthwindReports', file 'Northwi on file 1. [SQLSTATE 01000] (Message 4035) Processed 1 pages for database 'NorthwindReports', file 'Northwind_log' on file 1. [SQLSTA 01000] (Message 4035) RESTORE DATABASE successfully processed 1177 pages in 0.036 seconds (255.249 MB/sec). [SQLSTATE 01000] (Message 3014) The step succeeded.
MakeReportVersionOfNorthWind	3	TestRestore	20250302	210238	0	Succeeded	Executed as user: NT SERVICE\SQLAgent$GMSSQLSERVER_DEV. The step succeeded.
MakeReportVersionOfNorthWind	0	(Job outcome)	20250302	210633	3	Succeeded	The job succeeded. The Job was invoked by User DEA2025\girum. T last step to run was step 3 (TestRestore). The job was requested to start at step 1 (BackUpDB).
MakeReportVersionOfNorthWind	1	BackUpDB	20250302	210634	0	Succeeded	Executed as user: NT SERVICE\SQLAgent$GMSSQLSERVER_DEV. Processed 1176 pages for database 'Northwind', file 'Northwind' on 1. [SQLSTATE 01000] (Message 4035) Processed 1 pages for databas 'Northwind', file 'Northwind_log' on file 1. [SQLSTATE 01000] (Messag 4035) BACKUP DATABASE successfully processed 1177 pages in 0.07 seconds (116.316 MB/sec). [SQLSTATE 01000] (Message 3014) The s
MakeReportVersionOfNorthWind	2	Restore the copy as a different	20250302	210634	1	Succeeded	Executed as user: NT SERVICE\SQLAgent$GMSSQLSERVER_DEV. Processed 1176 pages for database 'NorthwindReports', file 'Northwi on file 1. [SQLSTATE 01000] (Message 4035) Processed 1 pages for



ETL Log for DWClinicReportData

ETLLog ID	ETLDate And Time	ETLAction	ETLLog Message
1	3/25/2025 11:41:30 AM	pETLDimClinics	DimClinics filled
2	3/25/2025 11:41:30 AM	pETLDropForeignKeyConstraints	Foreign Keys dropped
3	3/25/2025 11:41:30 AM	pETLTruncateTables	Tables data removed
4	3/25/2025 11:41:30 AM	pETLFillDimDates	DimDates filled
5	3/25/2025 11:41:30 AM	pETLDimClinics	DimClinics filled
6	3/25/2025 11:41:30 AM	pETLDimDoctors	DimDoctors filled
7	3/25/2025 11:41:30 AM	pETLDimProcedures	DimProcedures filled
8	3/25/2025 11:41:30 AM	pETLDimShifts	DimDimShifts filled
9	3/25/2025 11:41:38 AM	pETLDimPatients	DimPatients synced
10	3/25/2025 11:44:30 AM	pETLFactVisits	FactVisits filled
11	3/25/2025 11:44:56 AM	pFactDoctorShifts	FactDoctorShifts filled
12	3/25/2025 11:44:58 AM	pETLAddForeignKeyConstraints	Foreign Key Constraints re-created

3/29/2025 3:18:20 PM

Figure22, screeshot of the SSRS report for MSDB jobs and ETL Log Report

Summery

The ETL Technical Manual outlines the implementation of an automated ETL solution for a small medical clinic, replacing the current manual CSV-based process with a structured system using SQL Server, SSIS, and Python.

The project involves automating data ingestion, transformation, and loading into a Data Warehouse, improving efficiency and accuracy. Key milestones include file-based ETL for CSV imports, data warehouse ETL for structured reporting, Python-based ETL for Excel-based analytics, and automation using SSIS.

The solution also integrates SQL views, stored procedures, and SSRS dashboards to track ETL performance and errors. Designed for scalability and reliability, this manual serves as a guide for data engineers, ETL developers, and BI analysts involved in data integration projects.