

Data Migration and SCD Implementation in Azure SQL Database Assignment

By Amandeep
Singh

Overview of the Project:

- We need to transfer the tables' data from on-premises SQL server to ADLS G2
- From ADLS G2 csv files, we need to transform the data into SCD type 1 and SCD Type 2 dimension tables.

Technical Requirement:

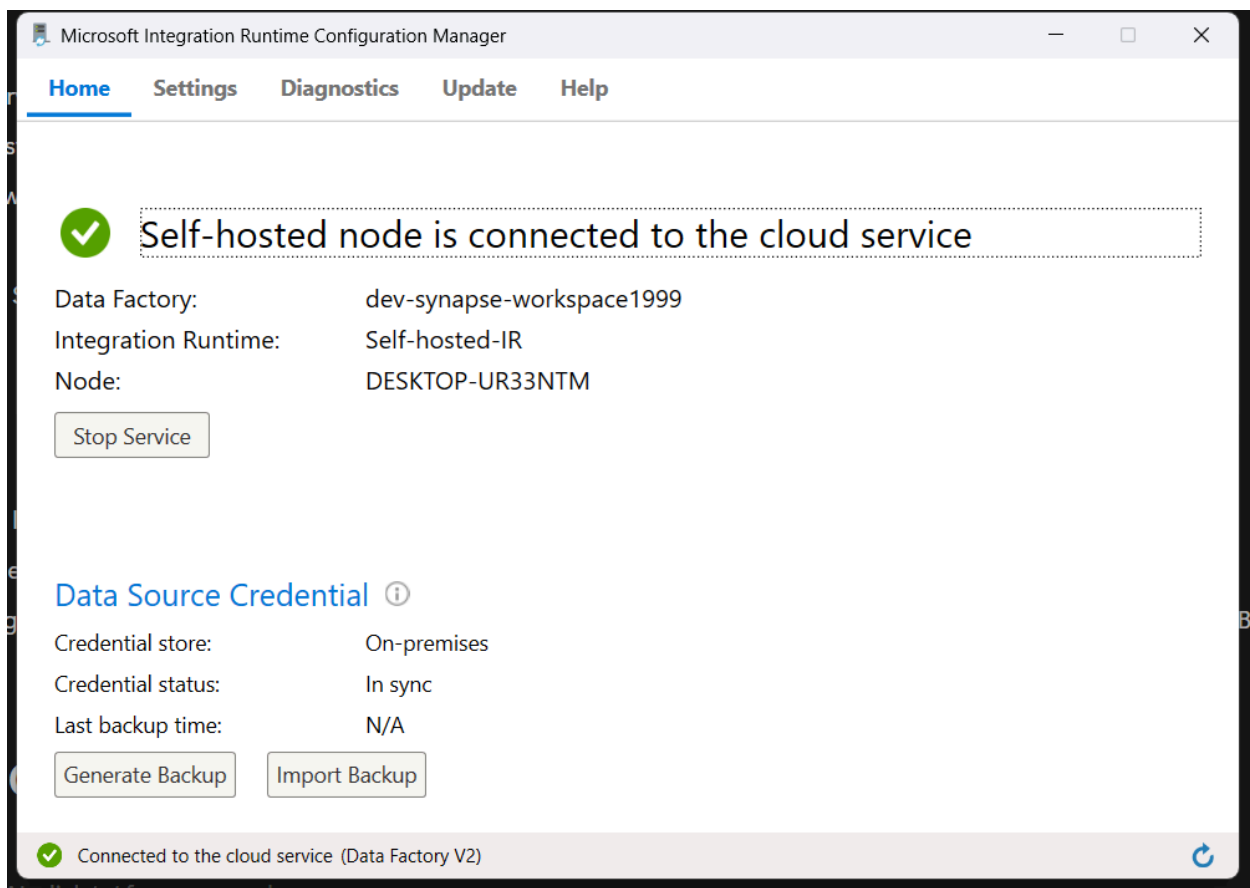
- We need SQL server and SSMS on local system.
- We need self hosted Integration runtime on local system.
- Connection between local SH-IR and Synapse's linked service is required.
- Access to ADLS
- Access to Azure SQL Database

Planning:

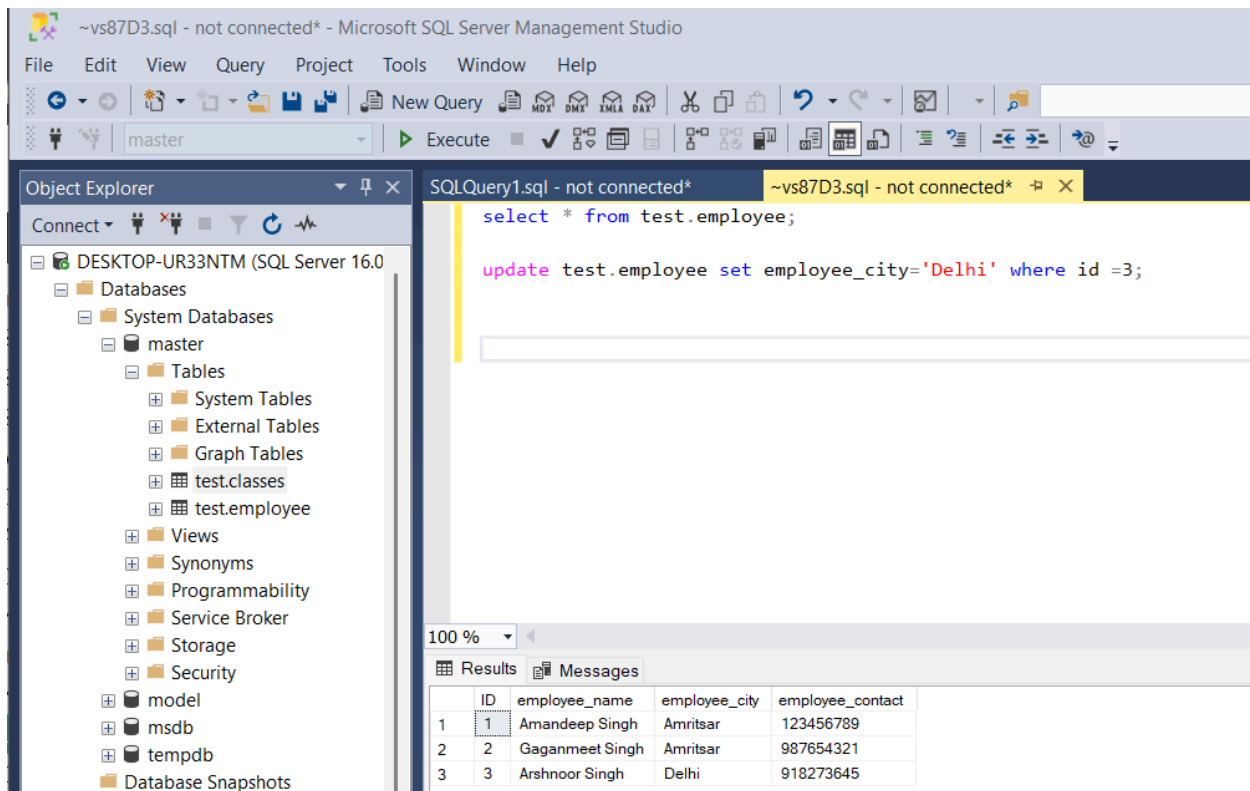
- create tables on local SQL server DB and create connection through self-hosted IR to Azure.

- Through Synapse Pipeline, transfer the SQL tables' data to ADLS.
- With Maintaining the data version, implement SCD type 1 and SCD type 2 Dimension tables for those two csv files on Azure SQL DB through Dataflow.

Setting up SQL server and Self-hosted IR on local System:



This is status of SH-IR that is connected with linked service in Synapse



- Here, I have setup the SQL server on local PC and I'm accessing it by SSMS

Creation of Tables and populating them on-premises:

Here are the codes for creation of two tables : employee and classes on on-premises SQL server:

```
create schema test;
--creation of first table
create table test.employee
(
id int identity(1,1),
employee_name varchar(100),
employee_city varchar(100),
employee_contact varchar(100)
```

```
);

--creation of 2nd table
create table test.classes
(
class_id int identity(1,1),
class_name varchar(100),
class_importance_factor int,
available_vacancy int
);

---insert data in first table
insert into test.employee values
('Amandeep Singh','Amritsar','987654321'),
('Gaganmeet Singh','Amritsar','918273644'),
('Arshnoor Singh','Delhi','987654321');

--insert data into second table
insert into test.classes values
('Physics',1,70),
('Chemistry',1,100),
('Mathematics',2,80),
('Biology',3,40);
```

SQLQuery1.sql - not connected* ~vs87D3.sql - DESK...TM.master (sa (52))*

```
select * from test.employee;  
  
select * from test.classes;
```

100 %

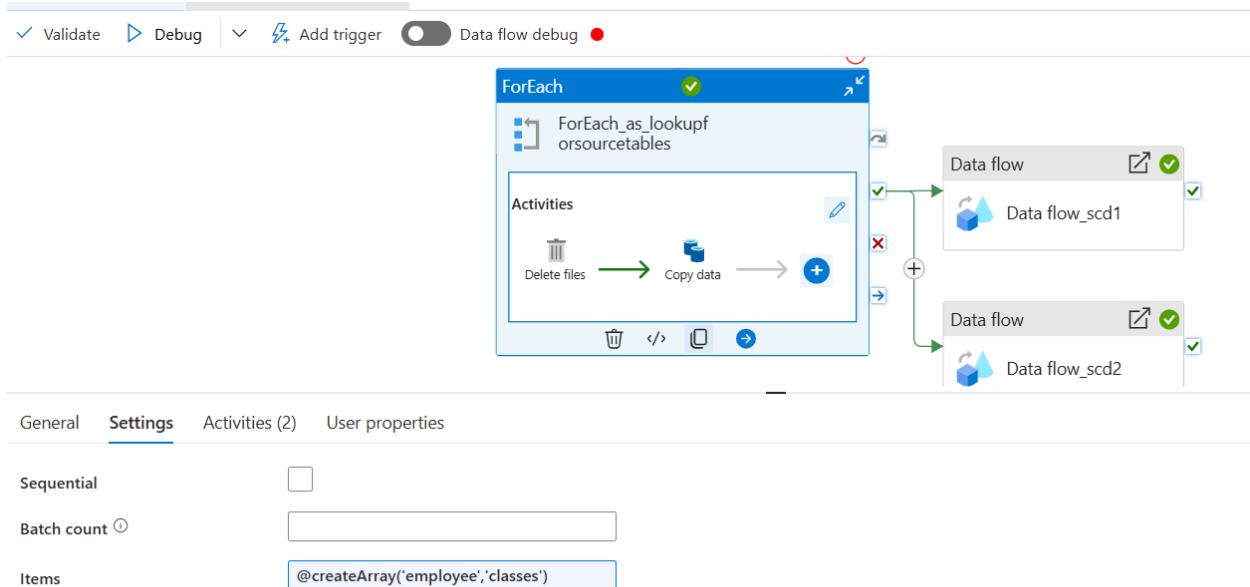
Results Messages

	ID	employee_name	employee_city	employee_contact
1	1	Amandeep Singh	Amritsar	123456789
2	2	Gaganmeet Singh	Amritsar	987654321
3	3	Arshnoor Singh	Delhi	918273645

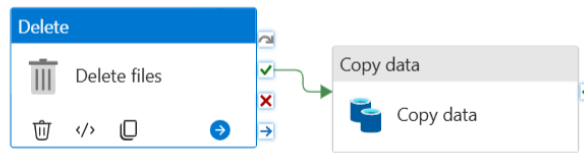
	class_ID	class_name	class_importance_factor	available_vacancy
1	1	Physics	1	100
2	2	Chemistry	1	70
3	3	Mathematics	2	80
4	4	Biology	3	30

Transferring the data from Local server to ADLS by Pipeline in Synapse:

- Here I'm using for-each loop in synapse pipeline and using custom array, I'm transferring the data from local SQL server to ADLS:



- Here you can see, I've used for each loop and it iterates over each item of array → ['employee','classes']
- Inside the for-each loop, first I'm deleting the previous files that have been copied before by **'Delete files'** activity and then transfer the SQL tables data to ADLS container by **'copy data'** activity.



General **Source** Logging settings User properties

Dataset * ⓘ ds_adls_general [Open](#) [New](#) [Preview data](#) [Learn more](#)

✓ Dataset properties ⓘ

Name	Value
folder_name	@item()
file_name	@concat(item(),'.csv')

File path type ☒ File path in dataset ☐ Wildcard file path ☐ List of files ⓘ

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
<input type="checkbox"/> classes	3/12/2025, 9:16:50 PM					...
<input type="checkbox"/> employee	3/12/2025, 9:16:42 PM					...

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
<input type="checkbox"/> [-]						...
<input type="checkbox"/> employee.csv	3/13/2025, 2:55:31 PM	Hot (Inferred)		Block blob	176 B	Available

Creation of SCD type 1 and SCD Type 2 Dimension Tables in Azure SQL DB

These are the SQL queries of tables creation in Azure SQL DB:

```
-- use first table employee for storing data in SCD type 1 format
create table dev.employee_scd_type1
(
    emp_id int,
    emp_name varchar(100),
    emp_city varchar(100),
    emp_contact varchar(100),
    createdby varchar(50),
    updatedby varchar(50),
    createddate datetime,
    updateddate datetime,
    hashkey bigint
);

--use second table classes for storing data in SCD TYPE 2 format

create table dev.classes_scd_type2
(
    class_id int,
    class_name varchar(100),
    class_imp_fact int,
    avail_vacancy int,
    createdby varchar(50),
    updatedby varchar(50),
    createddate datetime,
    updateddate datetime,
    hashkey bigint,
    isactive int
);
```


The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'dev' database is expanded, showing two tables: 'dev.classes_scd_type2' and 'dev.employee_scd_type1'. The 'dev.classes_scd_type2' table is selected, and its schema is visible: class_id (int, null), class_name (varchar, null), class_imp_fact (int, null), avail_vacancy (int, null), createdby (varchar, null), updatedby (varchar, null), createddate (datetime, null), updateddate (datetime, null), hashkey (bigint, null), and isactive (int, null). The 'dev.employee_scd_type1' table is also visible with columns: emp_id (int, null), emp_name (varchar, null), emp_city (varchar, null), emp_contact (varchar, null), createdby (varchar, null), updatedby (varchar, null), and createddate (datetime, null).

The 'Query 2' window is active, showing a SQL script with three lines:

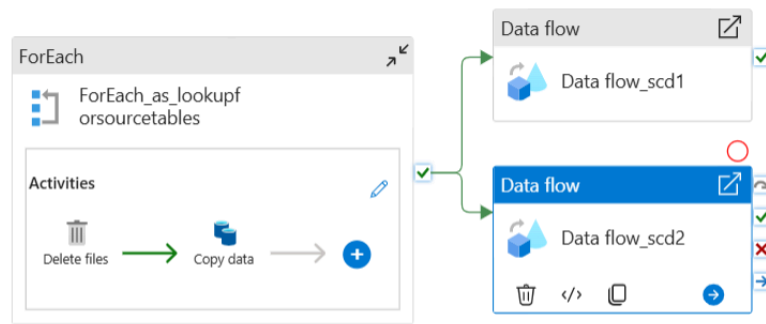
```
1 SELECT TOP (1000) * FROM [dev].[employee_scd_type1];
2
3 select * from dev.classes_scd_type2;
```

The 'Results' tab is selected, showing the output of the query. The results are displayed in a table with the following columns: class_id, class_name, class_imp_fact, avail_vacancy, and createdby. The data is as follows:

class_id	class_name	class_imp_fact	avail_vacancy	createdby
1	Physics	1	100	dataflow
2	Chemistry	1	70	dataflow
3	Mathematics	2	80	dataflow

Implementing SCD Type 1 and SCD Type 2 Conversion with help of Dataflows:

- After on-success of for-each loop, I start the both dataflow activities, which transform both tables' data into SCD type 1 and SCD type 2:



Self-Host Integration Runtime used :

- Here is the SH-IR that I used to make connection b/w azure and local SQL DB:

Integration runtimes

The integration runtime (IR) is the compute infrastructure to provide the following data integration capabilities across different network environment. [Learn more](#)

+ New Refresh

Filter by name

Showing 1 - 2 of 2 items

Name ↑↓	Type ↑↓	Sub-type ↑↓	Status ↑↓	Related ↑↓	Region ↑↓	Version ↑↓
AutoResolveIntegrationRuntime	Azure	Public	Running	13	Auto Resolve	---
Self-hosted-IR	Self-Hosted	---	Running	1	---	5.50.9162.1