Sri Lanka Institute of Information Technology

Data Warehousing & Business Intelligence

Assignment 1

D.T. Mallikarachchi

IT20166892

Y3S1 – 04 (DS – WEEKEND)

Step 01 – Dataset Selection

The Selected Data Source is a collection of property loan data. The link to the source data set is given below:

https://www.kaggle.com/datasets/yasserh/loan-default-dataset

Modifications were done accordingly to the data set derived from the source This data set reflects Comprehensive details about transactional data of the loan and the details of the loan type and the associated loan recipient and the loan officer,

The given data set was separated into multiple different source tables depending on the type of columns and additional columns were integrated to some tables to obtain more dimensions and hierarchy – as the assignment document specifies the enrichment of the ETL process

Two Main Data sources:

One SQL Database

One text file – Bank Branch Data (Branch.txt)

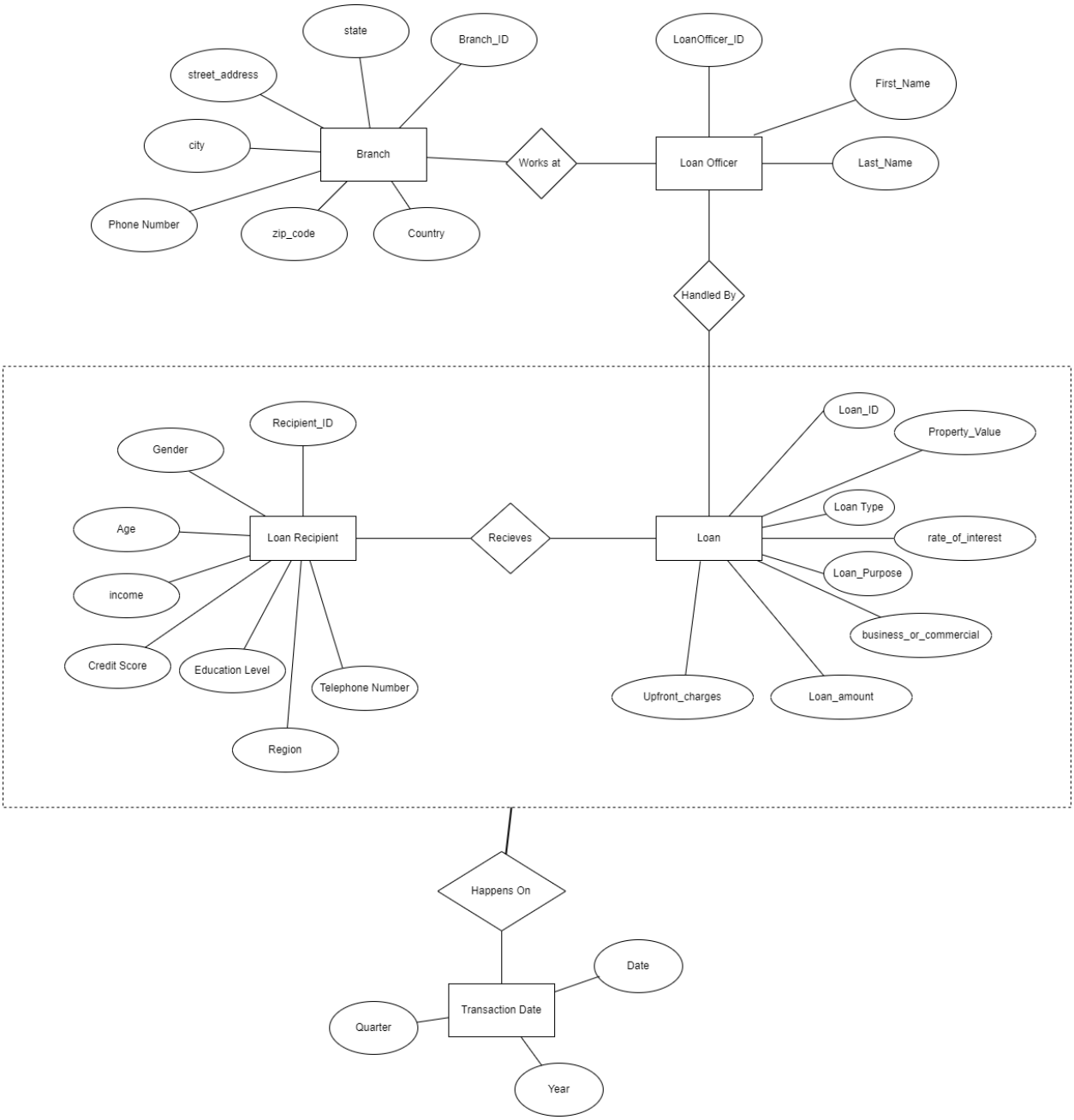
The below mentioned CSV files were imported to the SQL source Database.

Loan details data
Loan acquired Date data
Loan recipient Data
Loan type Data
Loan Officer Data

Description of the data set:

Table Name	Column name	Data Type	Description
Loan	Loan_ID	Int	Summar of the Details of the acquired loan including transactional data
	Date_ID	Varchar(50)	
	Recipient_ID	Int	
	LoanOfficer_ID	Int	
	Loan_amount	Float	
	Upfront_charges	Float	
	Property_value	float	
Loan_Type	Loan_ID	int	Details of the type and the purpose of loan and other important facts about the loan.
	Loan_type	nvarchar(50)	
	Loan_purpose	nvarchar(50)	
	business_or_commercial	nvarchar(50)	
	rate_of_interest	float	
LoanOfficer	LoanOfficer_ID	int	Details of the Loan Officer who is in charge of processing the loan
	First_Name	nvarchar(50)	
	Last_Name	nvarchar(50)	
	Branch_ID	int	
Recipient	Recipient_ID	int	Loan recipient details
	Gender	nvarchar(50)	
	age	nvarchar(50)	
	income	int	
	Credit_Score	smallint	
	Educational_Level	tinyint	
	Region	nvarchar(50)	
	Telephone_No	nvarchar(50)	
Date	Date_ID	Int	Details of the Loan transaction date
	VALUE_DATE	date	
	Quarter	tinyint	
	Year	smallint	
Branch	Branch_ID	int	Details of the Bank Branch of the Loan Officer(The branch at which the loan is processed)
	Street_address	varchar(50)	
	city	varchar(50)	
	state	varchar(50)	
	zip_code	varchar(50)	
	country	varchar(50)	
	phone_number	varchar(50)	

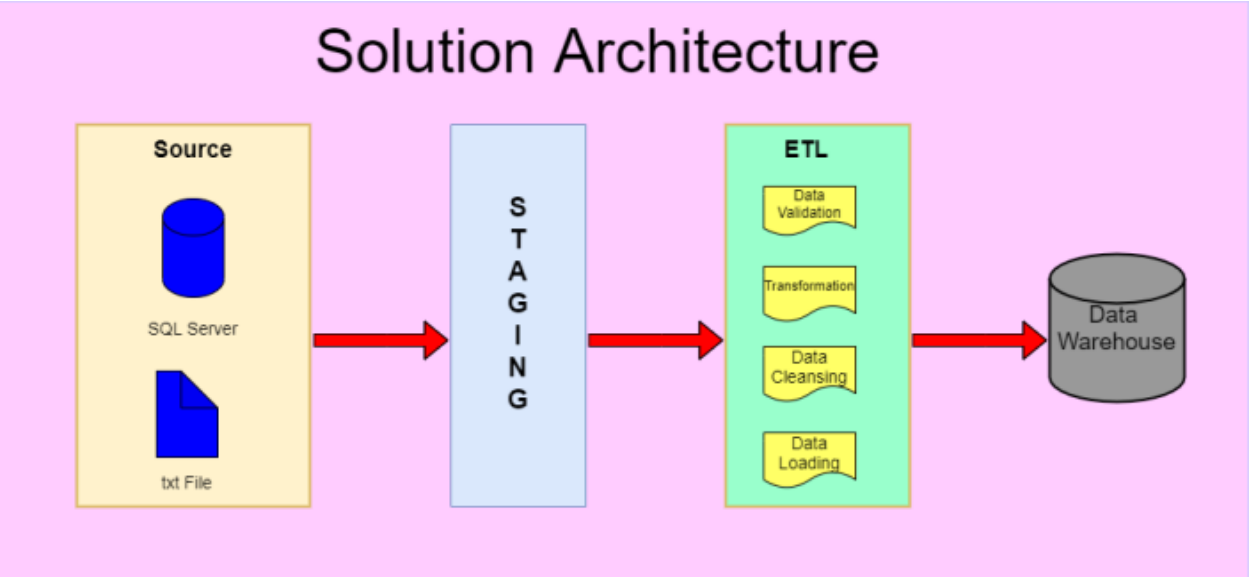
ER Diagram :



- This diagram shows the connection between the entities in the data set

Solution Architecture

Solution Architecture



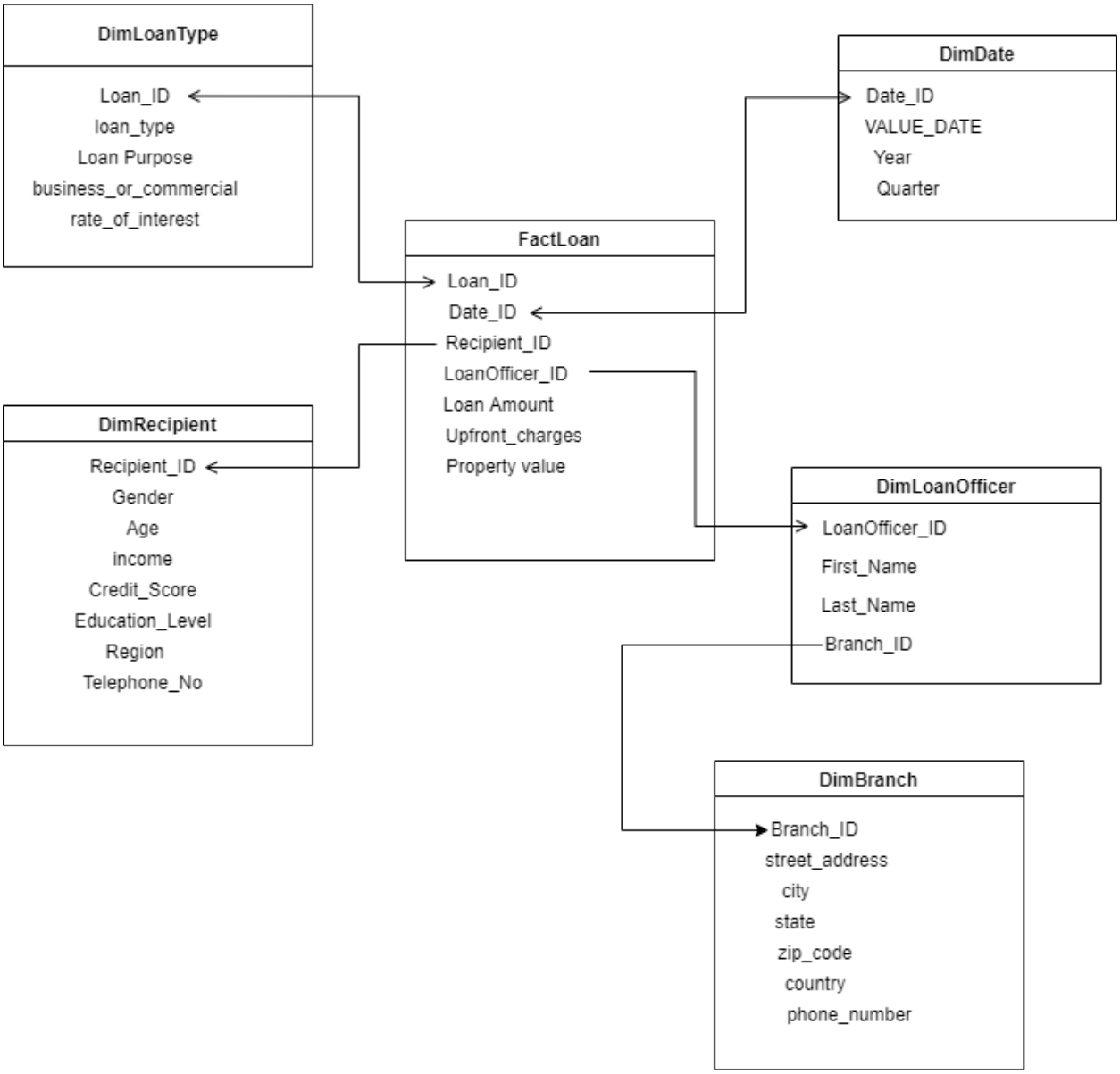
As explained First step is staging the source data set. After the staging layer the below mentioned staging tables are created:

1. Loan Staging
2. Branch Staging
3. Date Staging
4. Loan Officer Staging
5. Loan Type Staging
6. Recipient Staging

Next staged tables are profiled and aggregations are performed when necessary. As the next step data is transformed and loaded. After completing the described stages, data is tested and validated and the Datawarehouse is created.

After the warehouse is created BI results such as OLAP analysis, Reports, Data visualization, Data mining can be obtained as results after further modifications.

Data warehouse Design and Development



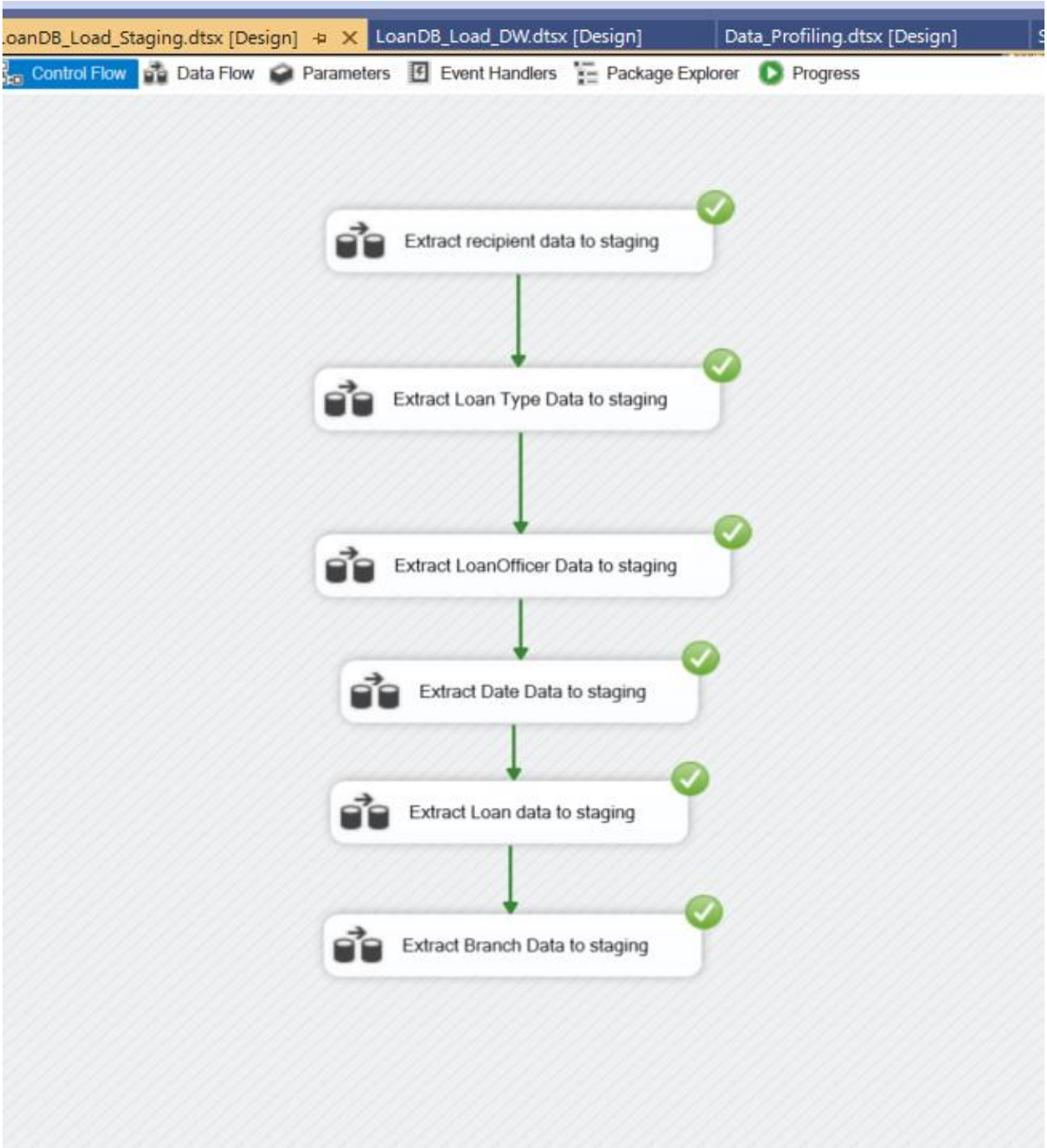
Snowflake schema is used to design the Data warehouse design. There is one fact table as FactLoan and 5 dimensions.

Assumptions.

DimRecipient was considered as a slowly changing dimension

ETL Development

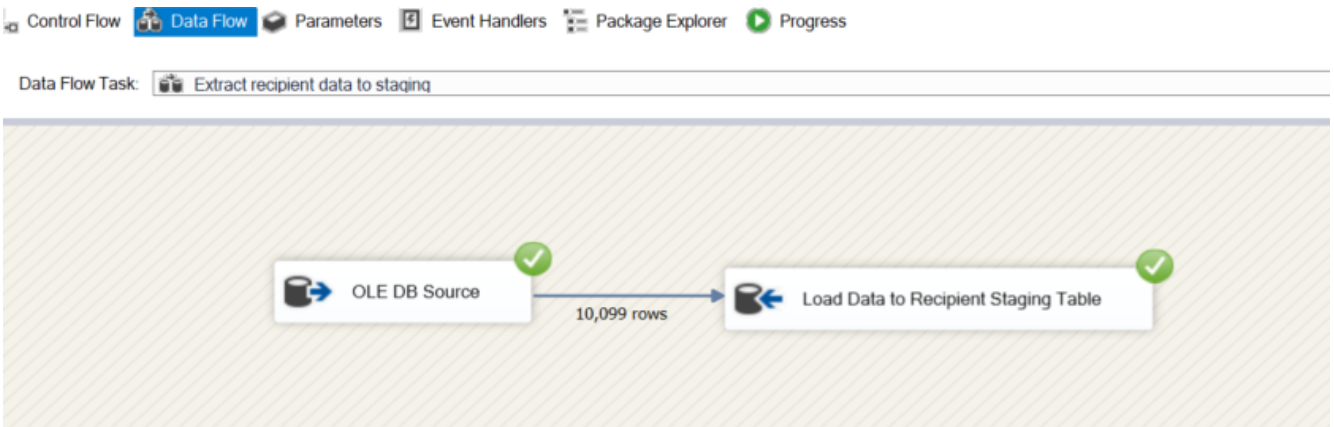
As the first step data was extracted from the sources (DB source & a text file). For every extraction, data flow task was used and data was extracted from the source to the staging table. Then for every staging table a truncate table was created. All the data flow tasks were joined as shown below.



Screenshots of all the data sources that were staged and truncate tables created are attached below: Staging customer details

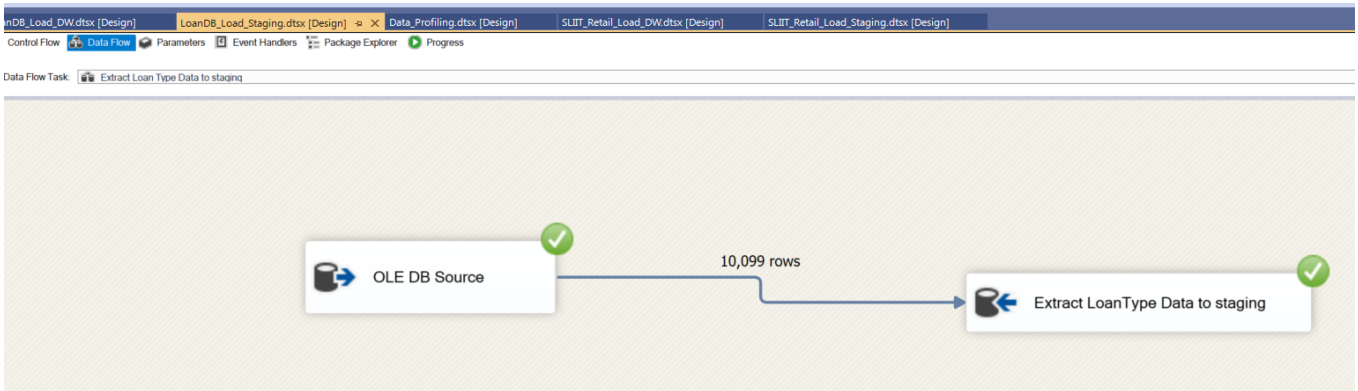
Staging Recipient Data

Data was extracted for, the Recipient table in the LoanDB_Resource Database and inserted to StgRecipient Staging table.



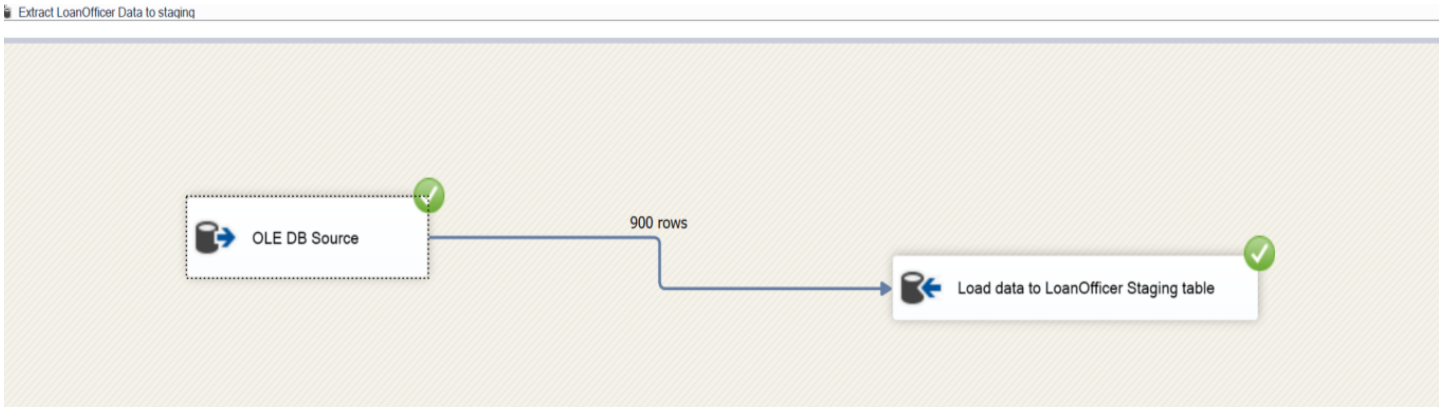
Staging Loan Type Data

Data was extracted from the LoanType table in the LoanDB_Resource Database and inserted to StgLoanType Staging table.



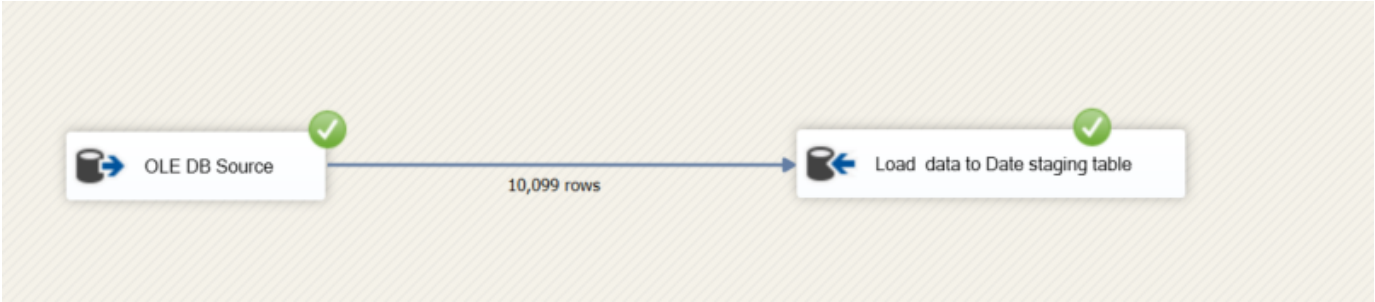
Staging Loan Officer Data

Data was extracted from the LoanOfficer table in the LoanDB_Resource Database and inserted to StgLoanOfficer Staging table.



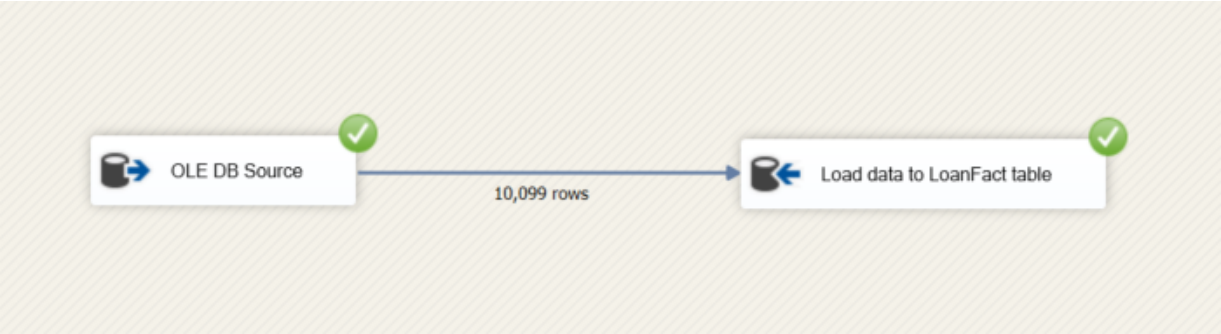
Staging Date Data

Data was extracted from the Date table in the LoanDB_Resource Database and inserted to StgDate Staging table.



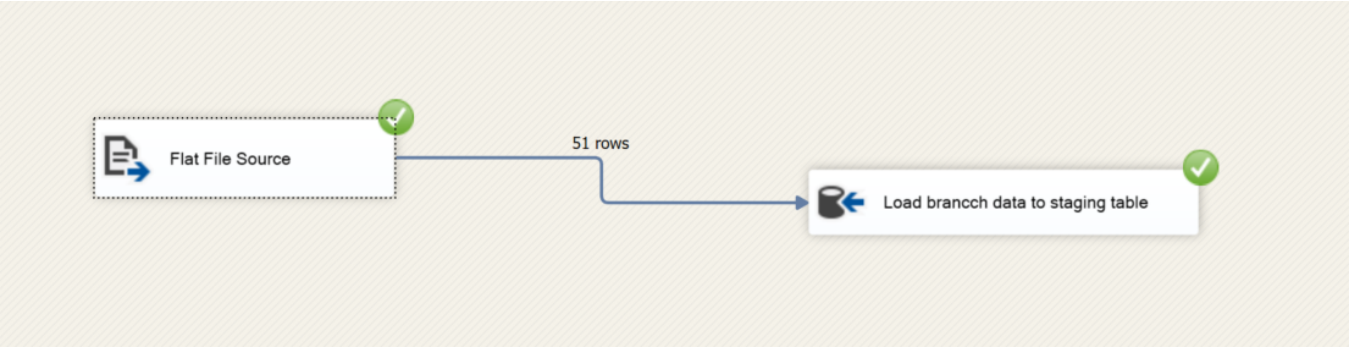
Staging Loan Data

Data was extracted from the Loan table in the LoanDB_Resource Database and inserted to StgLoan Staging table.



Staging Branch Data

Data was extracted from the Branchtable (whish is a CSV file)and inserted to StgBranch Staging table through a flat file source.

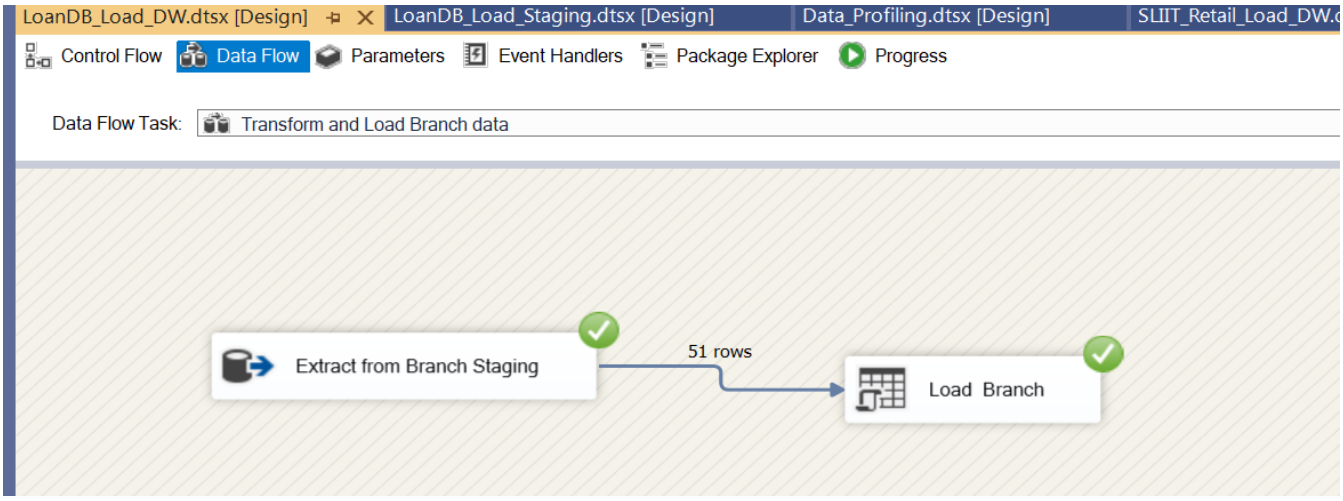


After staging all the tables, Data in the staging tables are transformed and loaded into the data warehouse using a package execution task.

Next step is data transformation and as explained in a previous step, the execution task connected to the last data flow task of the first package is attached to the transformation package used for transformation.

Transform and Load DimBranch

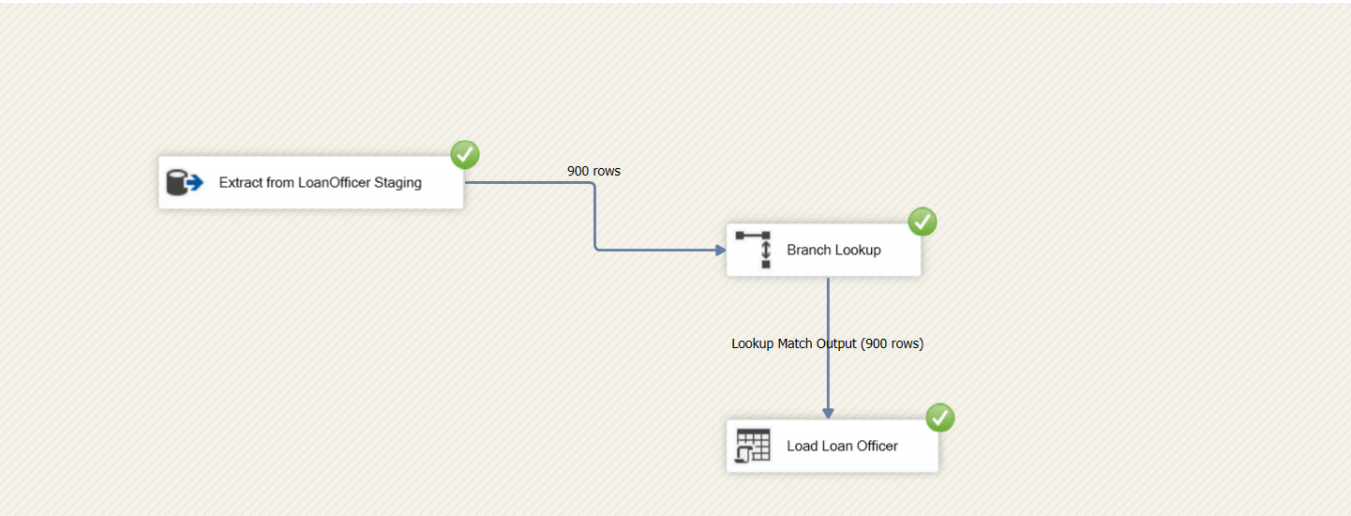
Branch dimension is loaded with data by executing a procedure. Here it shows how data is transformed and loaded in to the DimBranch Dimension table



Procedure used to update DimBranch table is given below:

```sql
CREATE PROCEDURE dbo.UpdateDimRecipient
@Recipient_ID int,
@Gender  nvarchar(50),
@age nvarchar(50),
@Telephone_No nvarchar(50),
@Region nvarchar(50),
@income int ,
@Credit_Score smallint,
@Education_Level tinyint



AS
BEGIN
if not exists (select RecipientSK
from dbo.DimRecipient
where AlternateRecipient_ID = @Recipient_ID)
BEGIN
insert into dbo.DimRecipient(AlternateRecipient_ID, Gender, age, Telephone_No,Region, income, Credit_Score, Education_Level, insertDate, ModifiedDate)
values
(@Recipient_ID, @Gender, @age, @Telephone_No,@Region, @income, @Credit_Score, @Education_Level, GETDATE(), GETDATE())
END;
if exists (select RecipientSK
from dbo.DimRecipient
where AlternateRecipient_ID = @Recipient_ID)
BEGIN
update dbo.DimRecipient
set Gender = @Gender,
age = @age,
Telephone_No = @Telephone_No,
Region = @Region,
income = @income,
Credit_Score = @Credit_Score,
Education_Level = @Education_Level,
ModifiedDate = GETDATE()
where AlternateRecipient_ID = @Recipient_ID
END;
END;
```

Transform and Load DimLoanOfficer



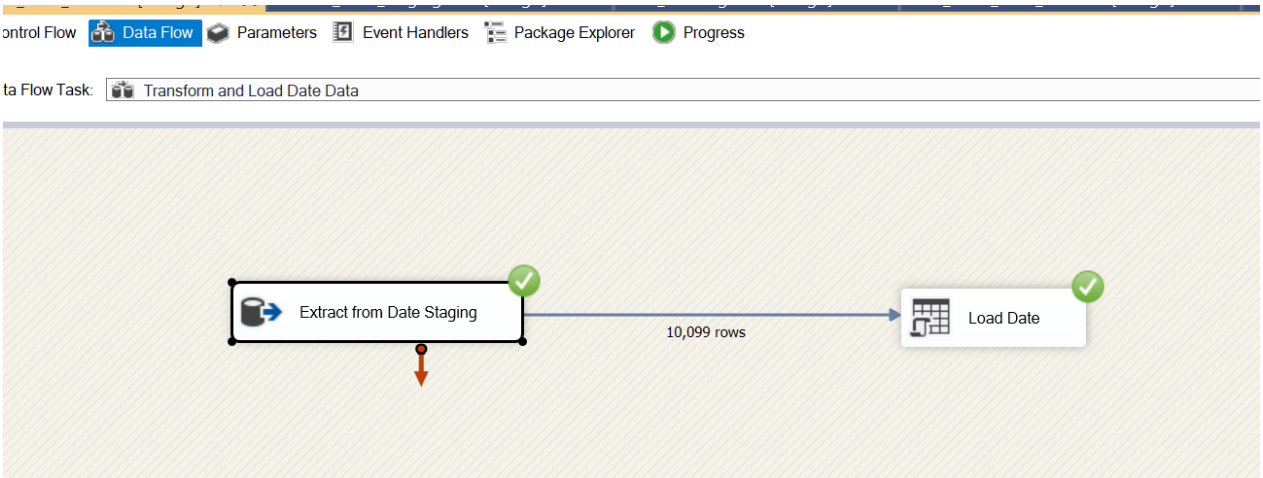
LoanOfficer dimension is transformed and loaded with data by first matching the corresponding Branch_ID s of the Branch table by using a lookup and then executing a procedure.

Procedure used to update DimLoanOfficer table is given below:

```
anTypeProcedure...FDI4TM\ASUS (58))    FactLoan.sql - LAP...2FDI4TM\ASUS (66))*    LoanOfficerProced...2FDI4TM\ASUS (54))    SQLQu
CREATE PROCEDURE dbo.UpdateDimLoanOfficer
@LoanOfficer_ID int,
@First_Name  nvarchar(50),
@Last_Name nvarchar(50),
@Branch_ID int

AS
BEGIN
if not exists (select LoanOfficerSK
from dbo.DimLoanOfficer
where AlternateLoanOfficer_ID = @LoanOfficer_ID)
BEGIN
insert into dbo.DimLoanOfficer(AlternateLoanOfficer_ID, First_Name, Last_Name, Branch_ID,insertDate, ModifiedDate)
values
(@LoanOfficer_ID, @First_Name, @Last_Name, @Branch_ID, GETDATE(), GETDATE())
END;
if exists (select LoanOfficerSK
from dbo.DimLoanOfficer
where AlternateLoanOfficer_ID = @LoanOfficer_ID)
BEGIN
update dbo.DimLoanOfficer
set First_Name = @First_Name,
Last_Name = @Last_Name,
Branch_ID = @Branch_ID,
ModifiedDate = GETDATE()
where AlternateLoanOfficer_ID = @LoanOfficer_ID
END;
END;
```

Transform and Load DimDate

Date dimension is loaded with data by executing a procedure. Here it shows how data is transformed and loaded in to the DimDate Dimension table



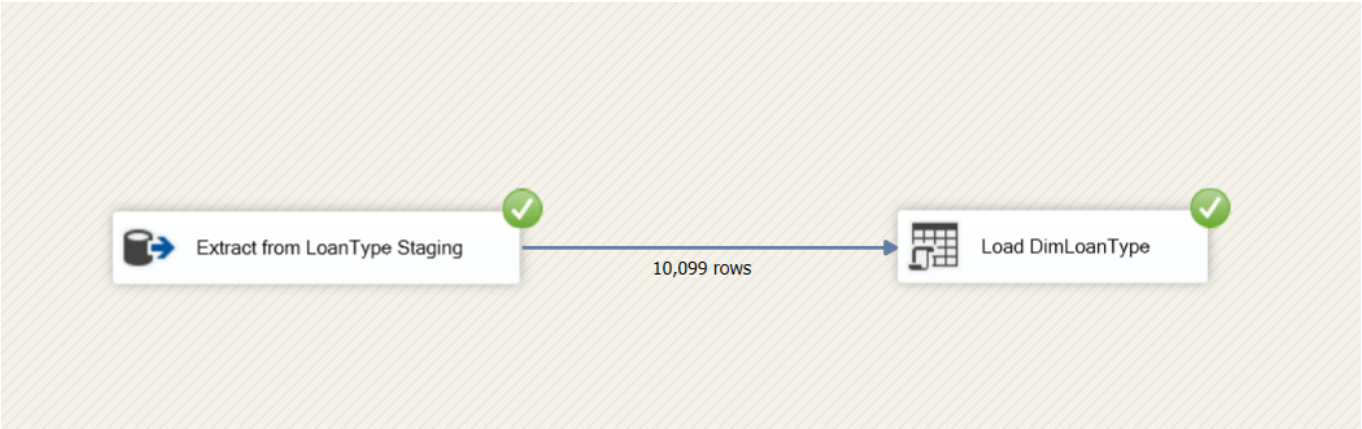
Procedure used to update DimDate table is given below:

DateProcedure.sql...2FDI4TM\ASUS (59)) | LoanTypeProcedure...FDI4TM\ASUS (58)) | FactLoan.sql - LAP...2FDI4TM\A

```sql
CREATE PROCEDURE dbo.UpdateDimDate
@Date_ID int,
@VALUE_DATE  date,
@Quarter tinyint,
@Year smallint

AS
BEGIN
if not exists (select DateSK
from dbo.DimDate
where AlternateDate_ID = @Date_ID)
BEGIN
insert into dbo.DimDate(AlternateDate_ID, VALUE_DATE, "Quarter","Year",insertDate, ModifiedDate)
values
(@Date_ID, @VALUE_DATE, @Quarter, @Year, GETDATE(), GETDATE())
END;
if exists (select DateSK
from dbo.DimDate
where AlternateDate_ID = @Date_ID)
BEGIN
update dbo.DimDate
set VALUE_DATE = @VALUE_DATE,
"Quarter" = @Quarter,
"Year" = @Year,
ModifiedDate = GETDATE()
where AlternateDate_ID = @Date_ID
END;
END;
```

Transform and Load DimLoanType

LoanType dimension is loaded with data by executing a procedure which was connected to a DBCommand. Here it shows how data is transformed and loaded in to the DimLoanType Dimension table.

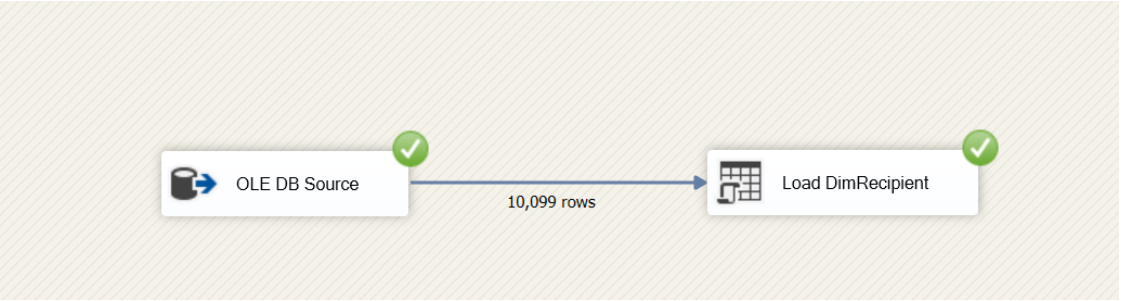


Procedure used to update DimLoanType table is given below:

```
CREATE PROCEDURE dbo.UpdateDimLoanType
@Loan_ID int,
@loan_type  nvarchar(50),
@loan_purpose nvarchar(50),
@business_or_commercial nvarchar(50),
@rate_of_interest float

AS
BEGIN
if not exists (select LoanTypeSK
from dbo.DimLoanType
where AlternateLoan_ID = @Loan_ID)
BEGIN
insert into dbo.DimLoanType(AlternateLoan_ID, loan_type, loan_purpose, business_or_commercial, rate_of_interest, insertDate, ModifiedDate)
values
(@Loan_ID, @loan_type, @loan_purpose, @business_or_commercial, @rate_of_interest, GETDATE(), GETDATE())
END;
if exists (select LoanTypeSK
from dbo.DimLoanType
where AlternateLoan_ID = @Loan_ID)
BEGIN
update dbo.DimLoanType
set loan_type = @loan_type,
loan_purpose = @loan_purpose,
business_or_commercial = @business_or_commercial,
rate_of_interest = @rate_of_interest,
ModifiedDate = GETDATE()
where AlternateLoan_ID = @Loan_ID
END;
END;
```

Transform and Load DimRecipient

Recipient dimension is loaded with data by executing a procedure which was connected to a DBCommand. Here it shows how data is transformed and loaded in to the DimRecipient Dimension table



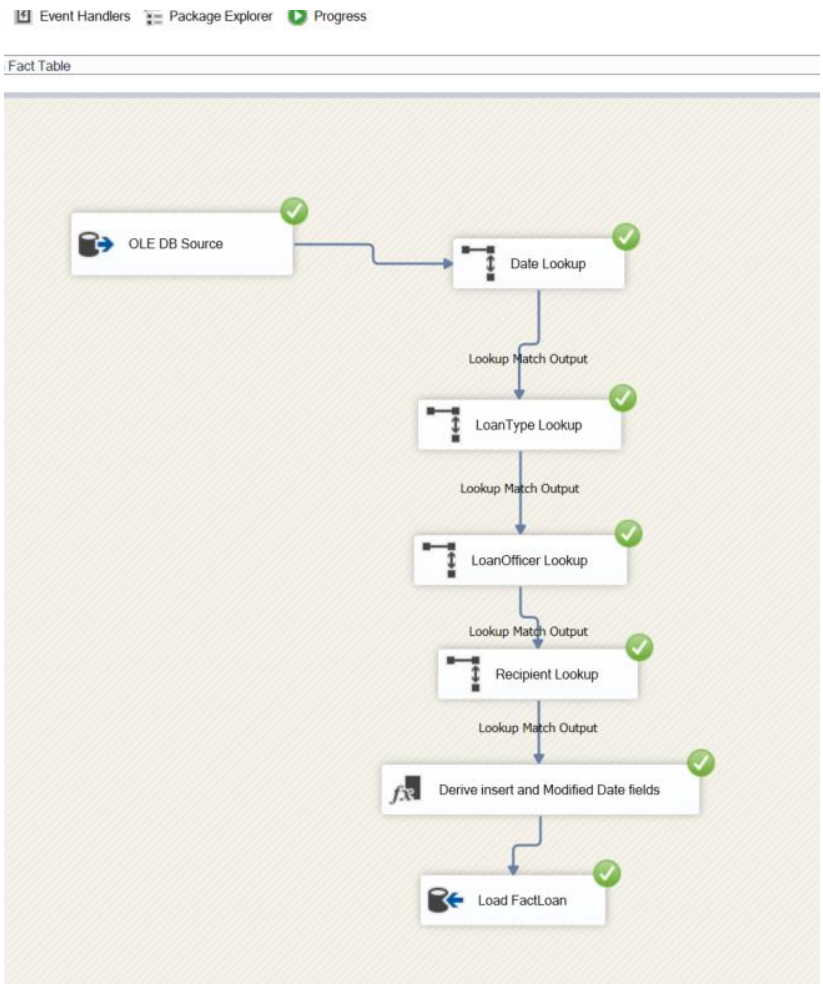
Procedure used to update DimRecipient table is given below:

```
CREATE PROCEDURE dbo.UpdateDimRecipient
@Recipient_ID int,
@Gender  nvarchar(50),
@age nvarchar(50),
@Telephone_No nvarchar(50),
@Region nvarchar(50),
@income int ,
@Credit_Score smallint,
@Education_Level tinyint



AS
BEGIN
if not exists (select RecipientSK
from dbo.DimRecipient
where AlternateRecipient_ID = @Recipient_ID)
BEGIN
insert into dbo.DimRecipient(AlternateRecipient_ID, Gender, age, Telephone_No,Region, income, Credit_Score, Education_Level, insertDate, ModifiedDate)
values
(@Recipient_ID, @Gender, @age, @Telephone_No,@Region, @income, @Credit_Score, @Education_Level, GETDATE(), GETDATE())
END;
if exists (select RecipientSK
from dbo.DimRecipient
where AlternateRecipient_ID = @Recipient_ID)
BEGIN
update dbo.DimRecipient
set Gender = @Gender,
age = @age,
Telephone_No = @Telephone_No,
Region = @Region,
income = @income,
Credit_Score = @Credit_Score,
Education_Level = @Education_Level,
ModifiedDate = GETDATE()
where AlternateRecipient_ID = @Recipient_ID
END;
END;
```

Transform and Load Loan Fact Table

To Transform and Load Loan Fact sales a series of transformations and lookups were used.

1. Using an OLEDB source, Data is extracted from StgLoan Staging table.
2. 'Date Lookup' lookup was used to match the corresponding Date fields of the loan table
3. 'LoanType' lookup was used to match the corresponding LoanType fields of the loan table
4. 'LoanOfficer Lookup' lookup was used to match the corresponding LoanOfficer fields of the loan table
5. 'Recipient Lookup' lookup was used to match the corresponding Recipient fields of the loan table.
6. A Derived Column is used to Derive insertDate and ModifiedDate to the LoanFact table.



- As mentioned earlier under assumptions, Recipient Dimension was considered as the slowly changing Dimension

The below mentioned columns were set as changing attributes,

1. Gender
2. age
3. income
4. Credit_Score
5. Education_Level
6. Region
7. Telephone_No
8. InsertDate
9. Modified_date

The below diagram shows the LoanFact Table after loading data to all the dimensions and the fact table:

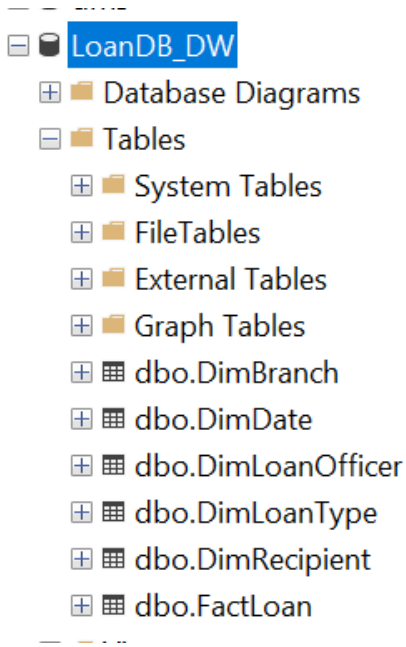
96 %

Results | Messages

	Loan_Key	Date_key	LoanOfficer_Key	Recipient_Key	Loan_ID	Date_ID	LoanOfficer_ID	Recipient_ID	loan_amount	Upfront_charges	property_value	InsertDate	ModifiedDate
6	6	6	631	6	24895	1000006	629	1000006	706500	370	1008000	2022-05-19 18:45:59.887	2022-05-19 18:45:
7	7	7	83	7	24896	1000007	81	1000007	346500	5120	438000	2022-05-19 18:45:59.887	2022-05-19 18:45:
8	8	8	22	8	24897	1000008	20	1000008	266500	5609.88	308000	2022-05-19 18:45:59.887	2022-05-19 18:45:
9	9	9	788	9	24898	1000009	786	1000009	376500	1150	478000	2022-05-19 18:45:59.887	2022-05-19 18:45:
10	10	10	450	10	24899	1000010	448	1000010	436500	2316.5	688000	2022-05-19 18:45:59.887	2022-05-19 18:45:
11	11	11	827	11	24900	1000011	825	1000011	136500	0	168000	2022-05-19 18:45:59.887	2022-05-19 18:45:
12	12	12	345	12	24901	1000012	343	1000012	466500	1150	708000	2022-05-19 18:45:59.887	2022-05-19 18:45:
13	13	13	171	13	24902	1000013	169	1000013	206500	0	258000	2022-05-19 18:45:59.887	2022-05-19 18:45:
14	14	14	862	14	24903	1000014	860	1000014	436500	0	508000	2022-05-19 18:45:59.887	2022-05-19 18:45:
15	15	15	219	15	24904	1000015	217	1000015	226500	3953.13	298000	2022-05-19 18:45:59.887	2022-05-19 18:45:
16	16	16	553	16	24905	1000016	551	1000016	76500	0	138000	2022-05-19 18:45:59.887	2022-05-19 18:45:
17	17	17	203	17	24906	1000017	201	1000017	356500	0	368000	2022-05-19 18:45:59.887	2022-05-19 18:45:
18	18	18	385	18	24907	1000018	383	1000018	156500	0	168000	2022-05-19 18:45:59.887	2022-05-19 18:45:
19	19	19	224	19	24908	1000019	222	1000019	406500	895	598000	2022-05-19 18:45:59.887	2022-05-19 18:45:
20	20	20	847	20	24909	1000020	845	1000020	586500	650	748000	2022-05-19 18:45:59.887	2022-05-19 18:45:
21	21	21	402	21	24910	1000021	400	1000021	306500	10470	558000	2022-05-19 18:45:59.887	2022-05-19 18:45:
22	22	22	4	22	24911	1000022	2	1000022	136500	4156.25	198000	2022-05-19 18:45:59.887	2022-05-19 18:45:
23	23	23	643	23	24912	1000023	641	1000023	306500	0	448000	2022-05-19 18:45:59.887	2022-05-19 18:45:
24	24	24	636	24	24913	1000024	634	1000024	316500	6661.88	508000	2022-05-19 18:45:59.887	2022-05-19 18:45:
25	25	25	634	25	24914	1000025	632	1000025	336500	0	428000	2022-05-19 18:45:59.887	2022-05-19 18:45:

Disconnected.

The final fact tables and the Dimension tables of the Data Warehouse is as follows:



ETL development – Accumulating fact tables

It was needed to add the following three columns and the relevant to the LoanFact table,

1. accm_txn_create_time
2. txn_process_time_hours
3. accm_txn_complete_time

In order to add the three columns, the following procedure was used.

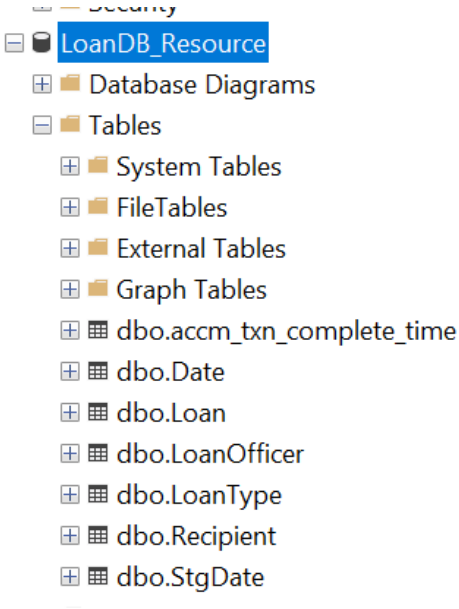
1. Firstly, the three columns were added to the LoanFact tale using the design function.

accm_txn_create_time	datetime	☑
accm_txn_complete_time	datetime	☑
txn_process_time_hours	int	☑
		☐

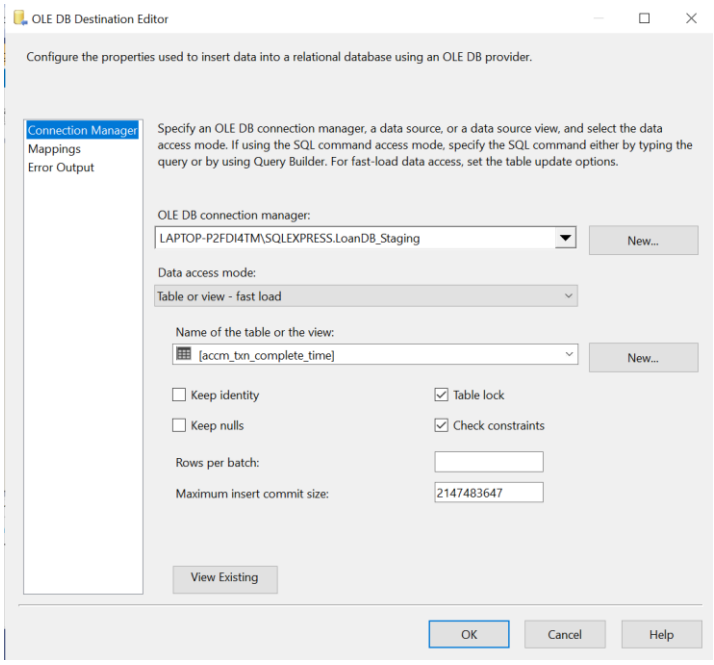
The FactTable looks like below after the three fields were added

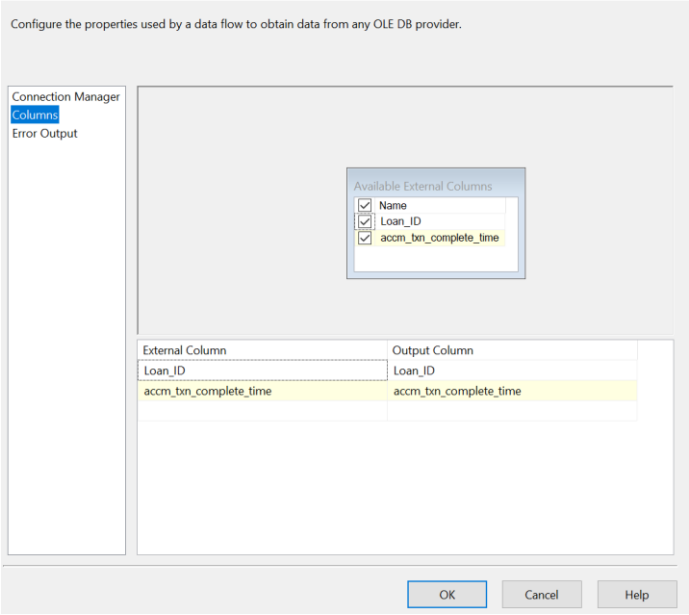
- dbo.FactLoan
  - Columns
    - Loan_Key (FK, int, null)
    - Date_Key (FK, int, null)
    - LoanOfficer_Key (FK, int, null)
    - Recipient_Key (FK, int, null)
    - Loan_ID (int, null)
    - Date_ID (int, null)
    - LoanOfficer_ID (int, null)
    - Recipient_ID (int, null)
    - loan_amount (float, null)
    - Upfront_charges (float, null)
    - property_value (float, null)
    - InsertDate (datetime, null)
    - ModifiedDate (datetime, null)
    - accm_txn_create_time (datetime, null)
    - accm_txn_complete_time (datetime, null)
    - txn_process_time_hours (int, null)

2. Afterwards a csv table was created with the Loan_ID and the corresponding accm_txn_complete_time date value as columns and then it was imported to the LoanDB_Resouce using the import flat files feature in the tasks menu.

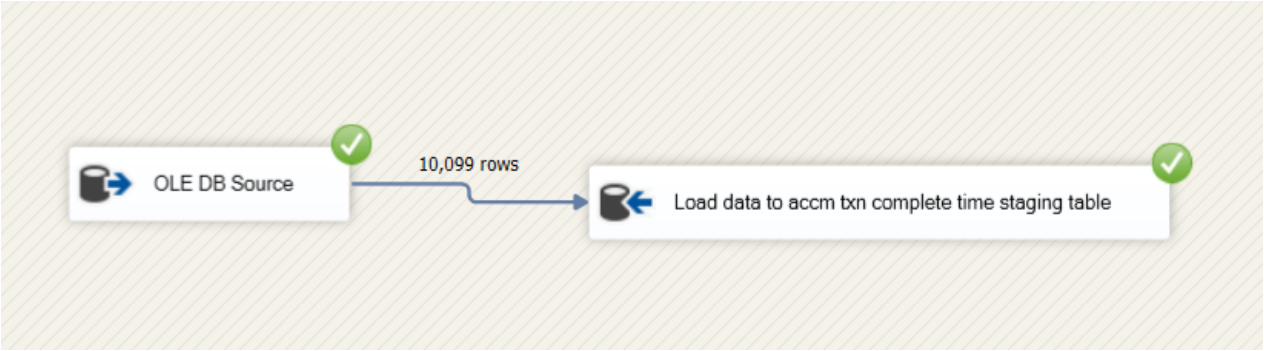


3. Then the dbo. accm_txn_complete_time table was loaded to the LoanDB_Staging staging database through an OLEDB Source and a OLEDB Destination as shown below.

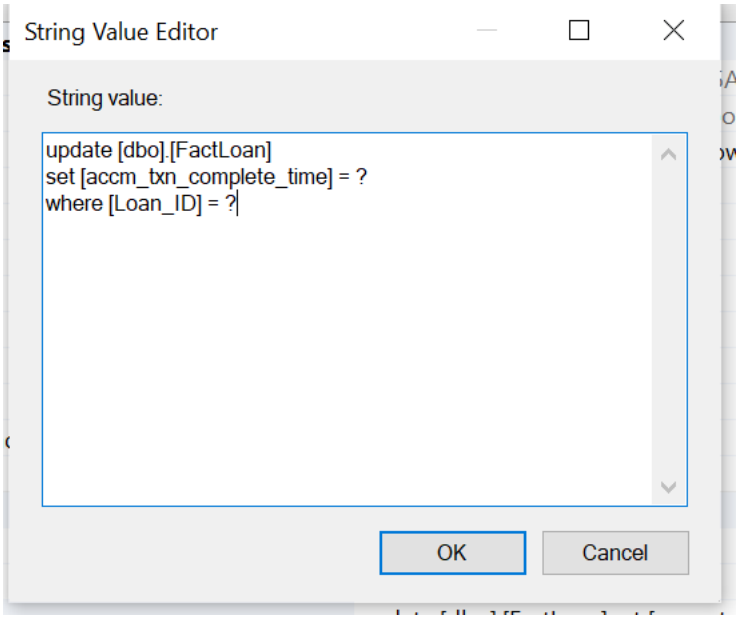




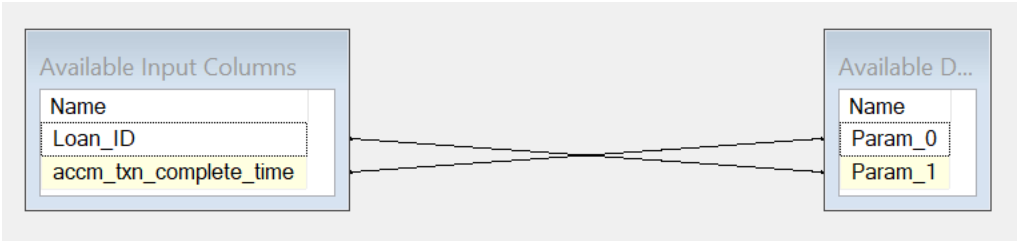
A truncate SQL task was added to the staging table and then was executed succesfully as shown below.

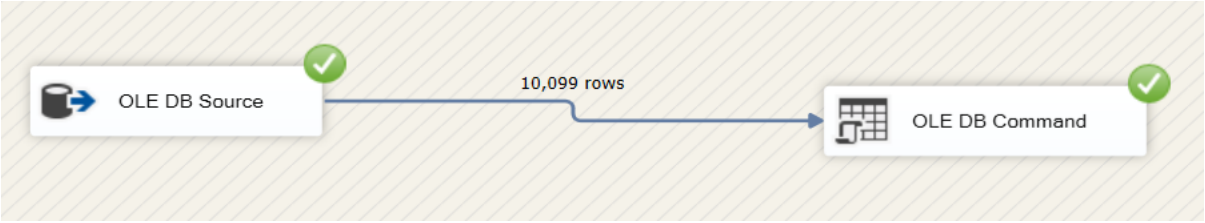


4. Then a new SSIS package was created for the Datawarehouse update transformations.then an OLEDB Source and an OLEDB command was used to update the accm_txn_complete_time column.

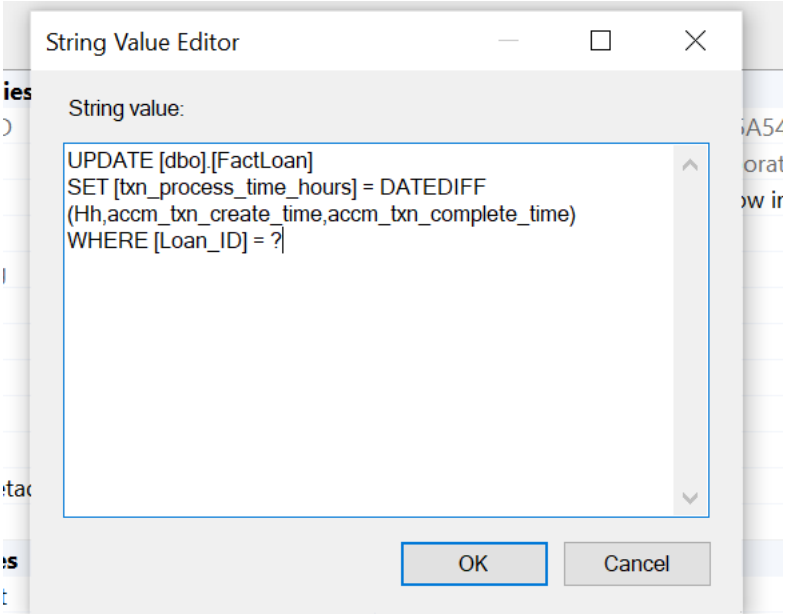


This SQL querry was used to update the accm_txn_complete_time field in the FactLoan table



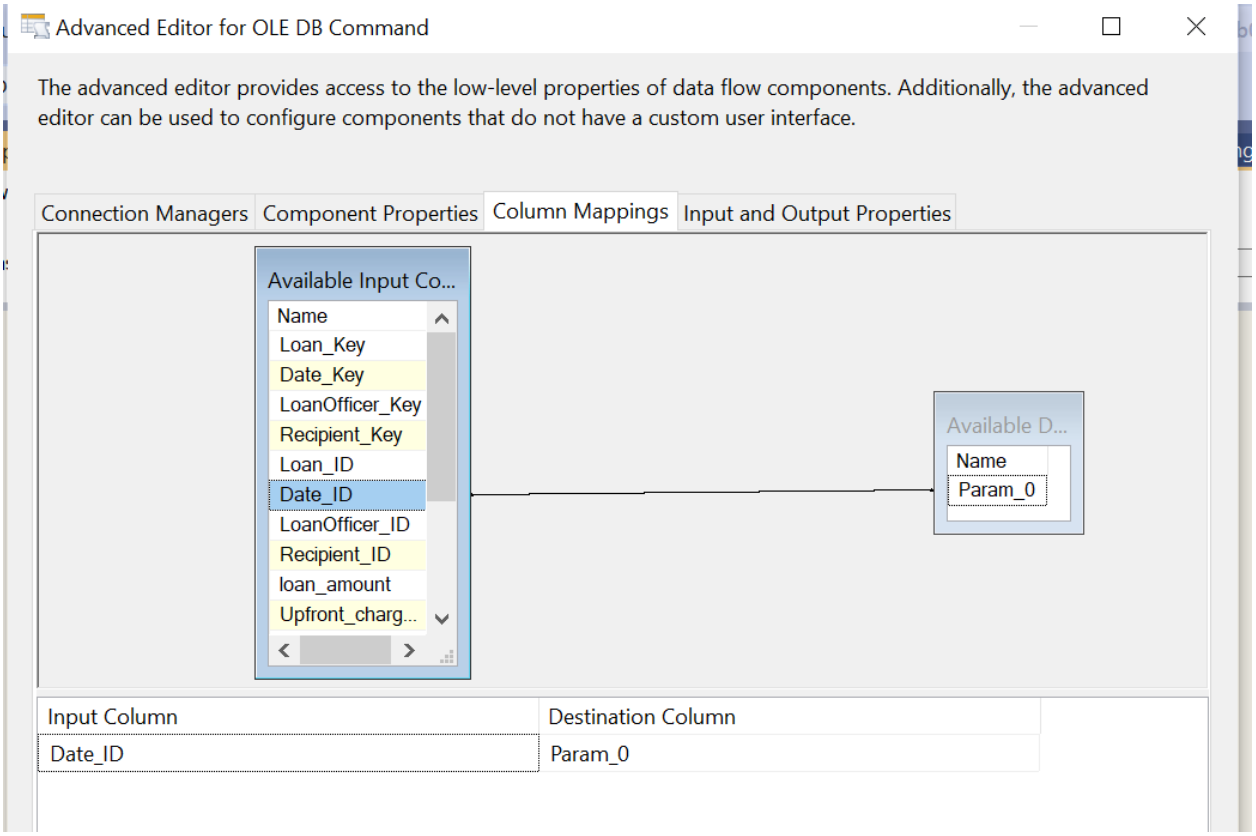


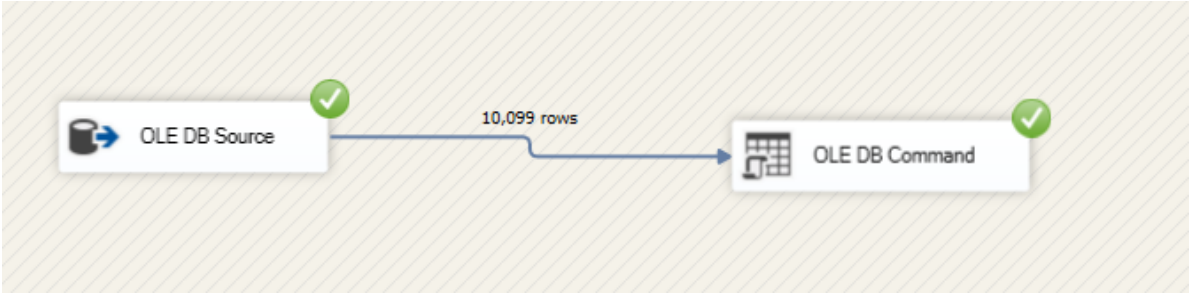
5. To calculate and populate the column, an OLEDB source and an OLEDB command was used.



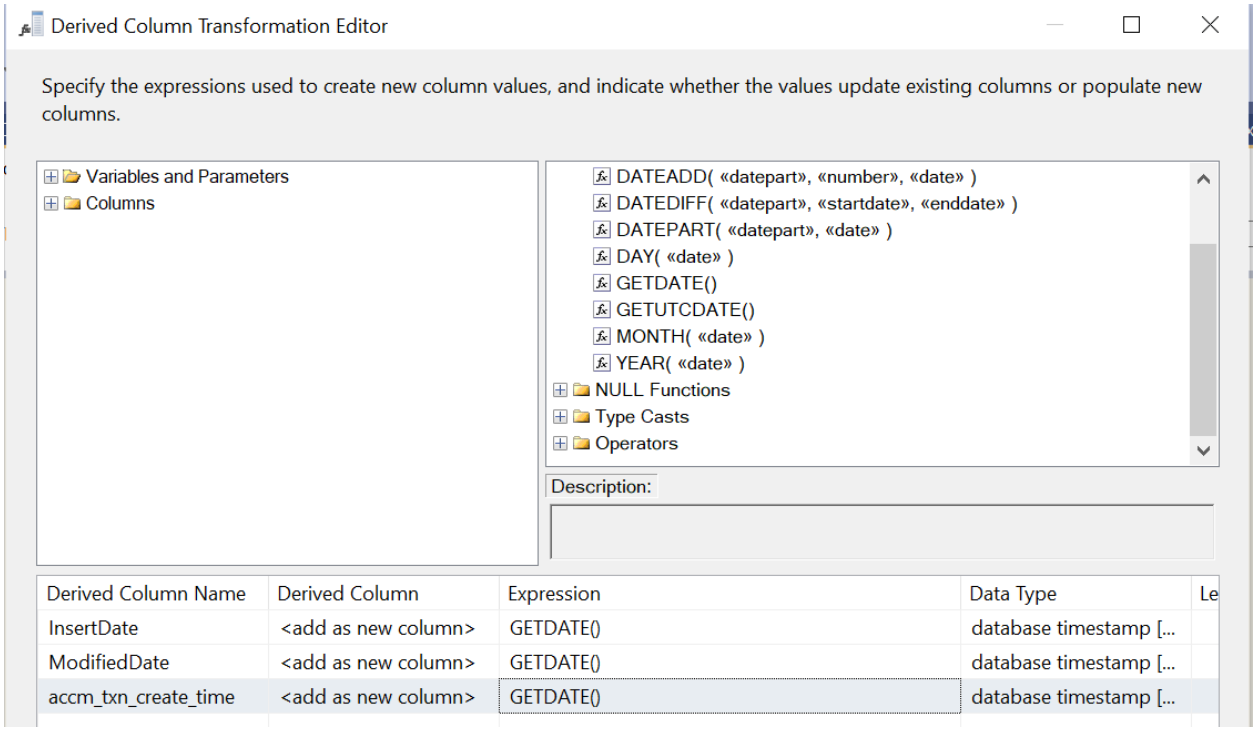
This SQL command was used to calculate txn_process_time_ and update the factloan table.

Column mapping for the transformation are as follows





6. Using the previous derived Column feature, accm_txn_create_time was assigned the current date

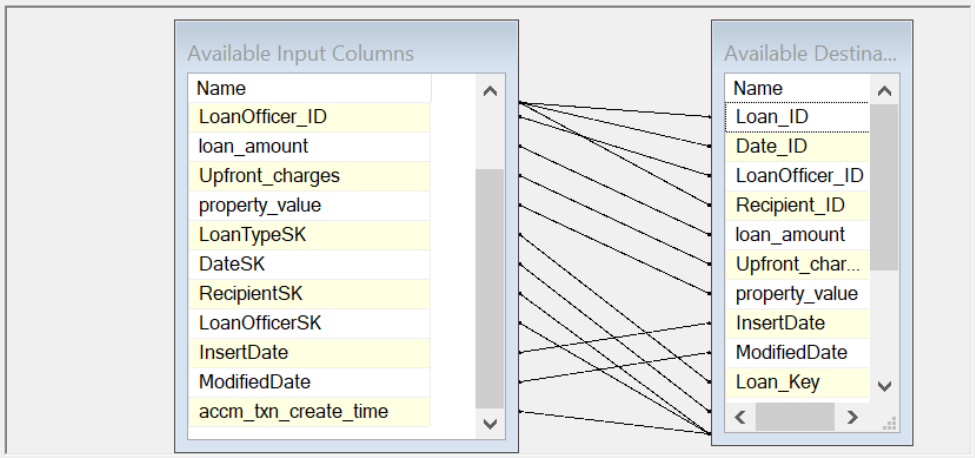


accm_txn_create_time was mapped to the LoanFact table in the following LoadFactLoan OLEDB Destination.

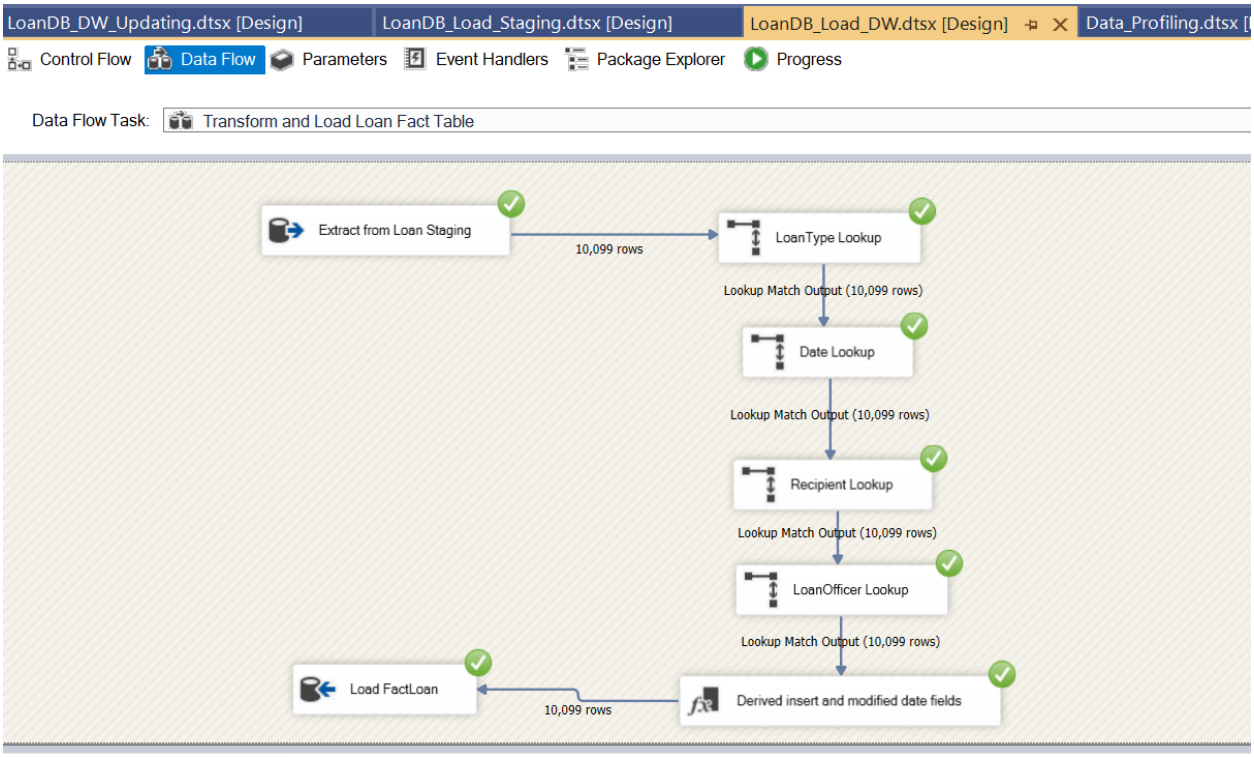
OLE DB Destination Editor

Configure the properties used to insert data into a relational database using an OLE DB provider.

- Connection Manager
- Mappings
- Error Output



Input Column	Destination Column
Upfront_charges	Upfront_charges
property_value	property_value
InsertDate	InsertDate
ModifiedDate	ModifiedDate
LoanTypeSK	Loan_Key
DateSK	Date_Key
LoanOfficerSK	LoanOfficer_Key
RecipientSK	Recipient_Key
accm_txn_create_time	accm_txn_create_time
<ignore>	accm_txn_complete_time
<ignore>	txn_process_time_hours



When all the elements are executed, the FactLoan table in the data warehouse gets populated with the new three column data as shown below.

96 %

Results | Messages

	ey	Loan_ID	Date_ID	LoanOfficer_ID	Recipient_ID	loan_amount	Upfront_charges	property_value	InsertDate	ModifiedDate	accm_txn_create_time	accm_txn_complete_time	txn_process_time_hours
6		24895	1000006	629	1000006	706500	370	1008000	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 00:00:00.000	-18
7		24896	1000007	81	1000007	346500	5120	438000	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 00:00:00.000	-18
8		24897	1000008	20	1000008	266500	5609.88	308000	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 00:00:00.000	-18
9		24898	1000009	786	1000009	376500	1150	478000	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 00:00:00.000	-18
10		24899	1000010	448	1000010	436500	2316.5	688000	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 00:00:00.000	-18
11		24900	1000011	825	1000011	136500	0	168000	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 00:00:00.000	-18
12		24901	1000012	343	1000012	466500	1150	708000	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 00:00:00.000	-18
13		24902	1000013	169	1000013	206500	0	258000	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 00:00:00.000	-18
14		24903	1000014	860	1000014	436500	0	508000	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 00:00:00.000	-18
15		24904	1000015	217	1000015	226500	3953.13	298000	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 00:00:00.000	-18
16		24905	1000016	551	1000016	76500	0	138000	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 00:00:00.000	-18
17		24906	1000017	201	1000017	356500	0	368000	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 00:00:00.000	-18
18		24907	1000018	383	1000018	156500	0	168000	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 00:00:00.000	-18
19		24908	1000019	222	1000019	406500	895	598000	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 00:00:00.000	-18
20		24909	1000020	845	1000020	586500	650	748000	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 00:00:00.000	-18
21		24910	1000021	400	1000021	306500	10470	558000	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 00:00:00.000	-18
22		24911	1000022	2	1000022	136500	4156.25	198000	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 00:00:00.000	-18
23		24912	1000023	641	1000023	306500	0	448000	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 00:00:00.000	-18
24		24913	1000024	634	1000024	316500	6661.88	508000	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 00:00:00.000	-18
25		24914	1000025	632	1000025	336500	0	428000	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 18:45:59.887	2022-05-19 00:00:00.000	-18