Sri Lanka Institute of Information Technology

Data Warehousing and Business Intelligence (IT3021)

Loan Data for Dummy Bank

Assignment 1

IT20201678
Sandarangi R.M.N
Y3S1 Group 5.2
2022
(DS – Weekday)

Contents

Step 1: Data set selection

Background

Link to the selected source data set-
https://www.kaggle.com/datasets/mrferozi/loan-data-for-dummy-bank

Note: Some modifications have been done to the data set to get better ETL process.

The Irish Dummy Banks is a bank based in Ireland, in which bank provides funds for potential borrowers and bank earn a profit depending on the risk they take (the borrowers credit score). The complete data set is borrowed from Lending Club for more basic information about the company please check out the Wikipedia article about the company.

The Bank currently operates network of 53 branches. Each branch of Dummy Bank is able to provide many Loans.

The loyal customers can get loans from the Bank as the Borrowers. All the relevant details regarding Borrowers are available in the database.

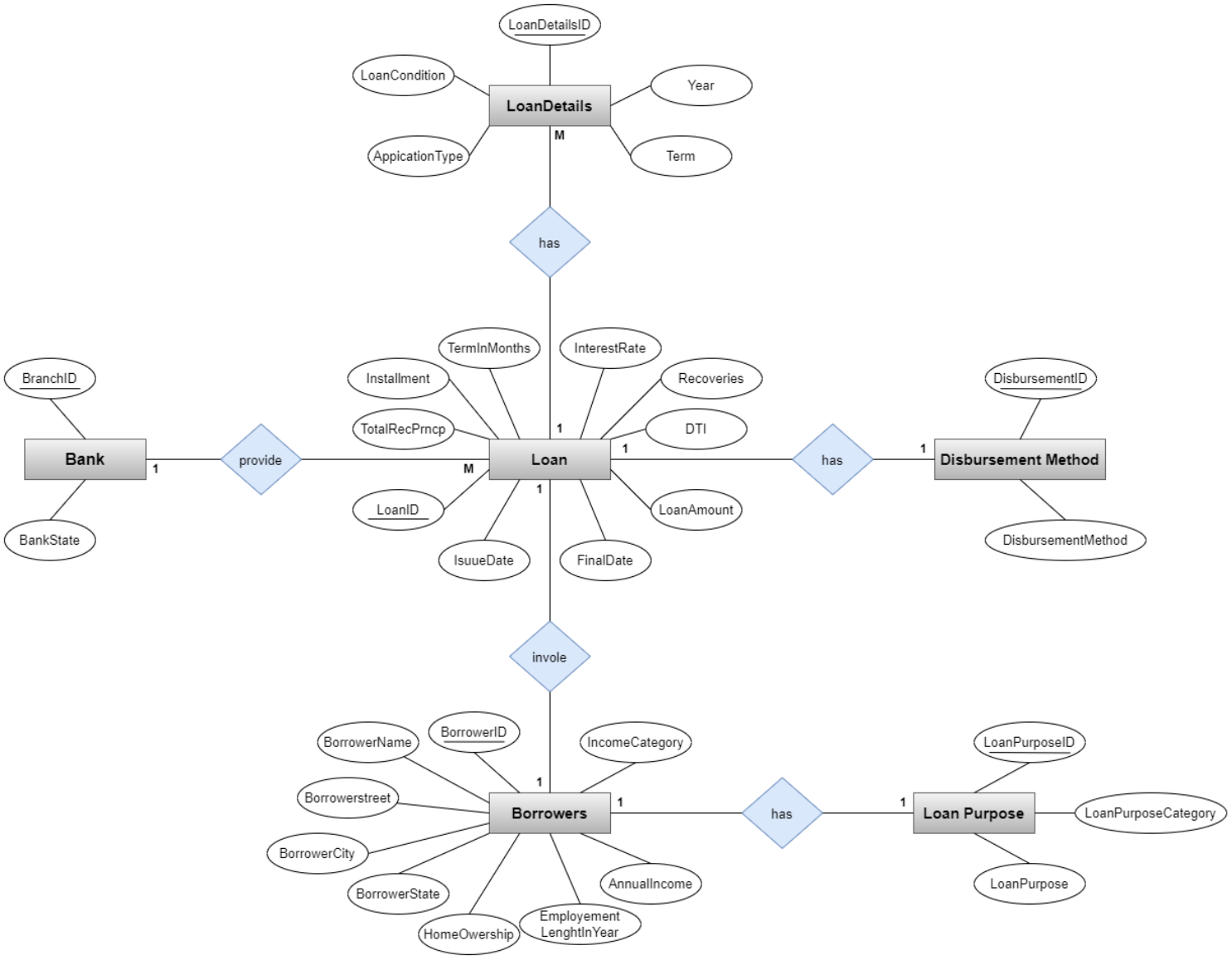
The system records all the matters and its involvers for a particular Loan.

Each Loan has a unique Loan Detail record, which is combined particular LoanID. This gives a good description about the loan that is got by the borrower.

Moreover, each Loan has a disbursement method. It can be either Cash or direct Pay.

ER Diagram

The high level of ER Diagram is attached below to get a better overview of the selected data set for this.



Step 2: Preparation of data sources

The original data source contains only one xlsx file and contains about one hundred thousand records.

Based on the requirements that file separated into multiple files, which are from two different types of data sources.

Text files: LoanPurpose
CSV files: Loan, Bank, Borrower, DisbursementMethod, LoanDetails

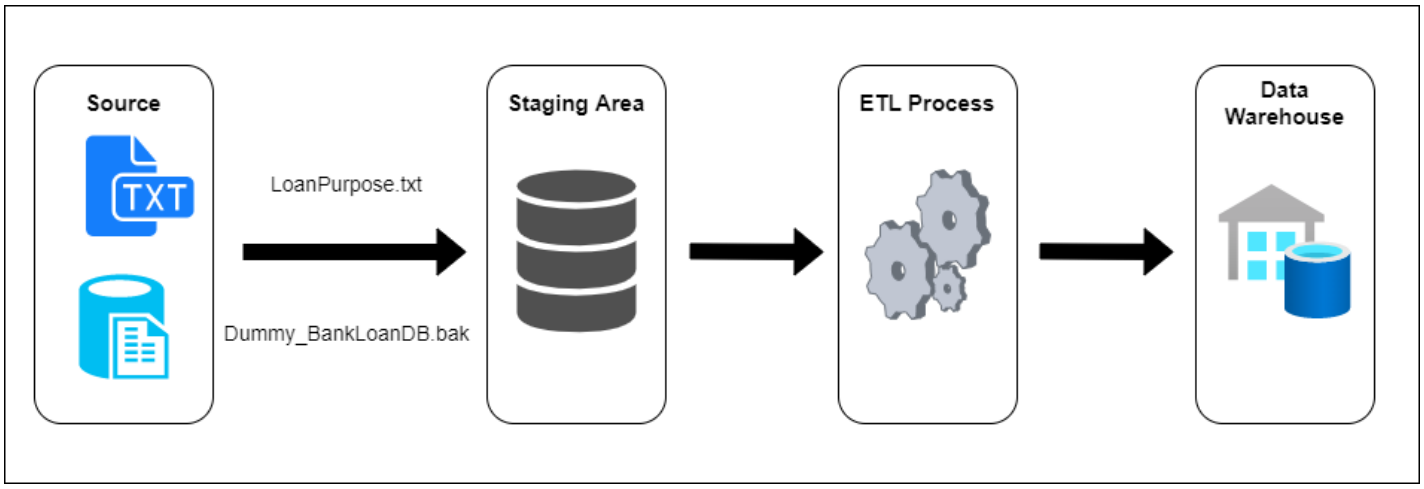
Following table contains the description of the Dataset.

Table Name	Column Name	Data Type	Description
Loan	LoanID IssueDate FinalDate LoanAmount InterestRate DTI TotalRecPrncp Recoveries Installment TermInMonths	Nvarchar(50) DateTime Varchar(50) Money Real Real Money Real Real Integer	Details about all loan transactions taken place
Borrower	BorrowerID BorrowerName BorrowerStreet BorrowerCity BorrowerState HomeOwnership EmployementInLeghtInYear AnnualIncome IncomeCategory	Nvarchar(50) Nvarchar(50) Nvarchar(50) Nvarchar(50) Nvarchar(50) Nvarchar(50) Float Integer Nvarchar(50)	Details about Borrowers. Who gets Loans from banks

Bank	BranchID BankState	Nvarchar(50) Nvarchar(50)	Details about Bank Branch
Disbursement Method	DisbursementID DisbursementMethod	Nvarchar(50) Nvarchar(50)	Details about loan delivery method
LoanDetails	LoanDetailsID Year Term ApplicationType LoanCondition	Nvarchar(50) Nvarchar(50) Nvarchar(50) Nvarchar(50) Nvarchar(50)	About loan details
LoanPurpose	LoanPurposeID LoanPurpose LoanPurposeCategory	Nvarchar(50) Nvarchar(50) Nvarchar(50)	Loan purpose details

Step 3: Solution architecture

The diagram below presents the overall architecture of the Datawarehouse and Business Intelligence.



The architecture comprises of four components.
1.Data Sources
2. Staging Area
3.ETL process
4.Data warehouse

- **Data sources**: This comprises of structured data in the format of text and excel and the formats are stored in the local folder.
- **Staging area:** In this, it loads source data into the DWH environment for further processing (the process from source-to-staging). In other words, the Staging Layer is responsible for the physical movement of data from the source platform onto the DWH platform.
- **ETL**: This is performed at two occurrences, to begin with occasion when extracting data from the sources and stacking it to the Staging Layer and in moment occurrence when performing extraction and change on Staging Layer to load data into Data warehouse Layer.
- **Data Warehouse:** enable and support business intelligence (BI) activities, especially analytics.

Step 4: Data warehouse design & development

Snowflake schema was selected to design the Data Warehouse of **Dummy Bank Loan Data** according to the behavior and the number of dimensional tables and fact tables.

Dimension Tables:

- dbo.DimBank
- dbo.DimBorrower → Slowly Changing Dimension
- dbo.DimDate
- dbo.DimDisbursementMethod
- dbo.DimLoanDetails
- dbo.DimLoanPurpose

Fact Table:

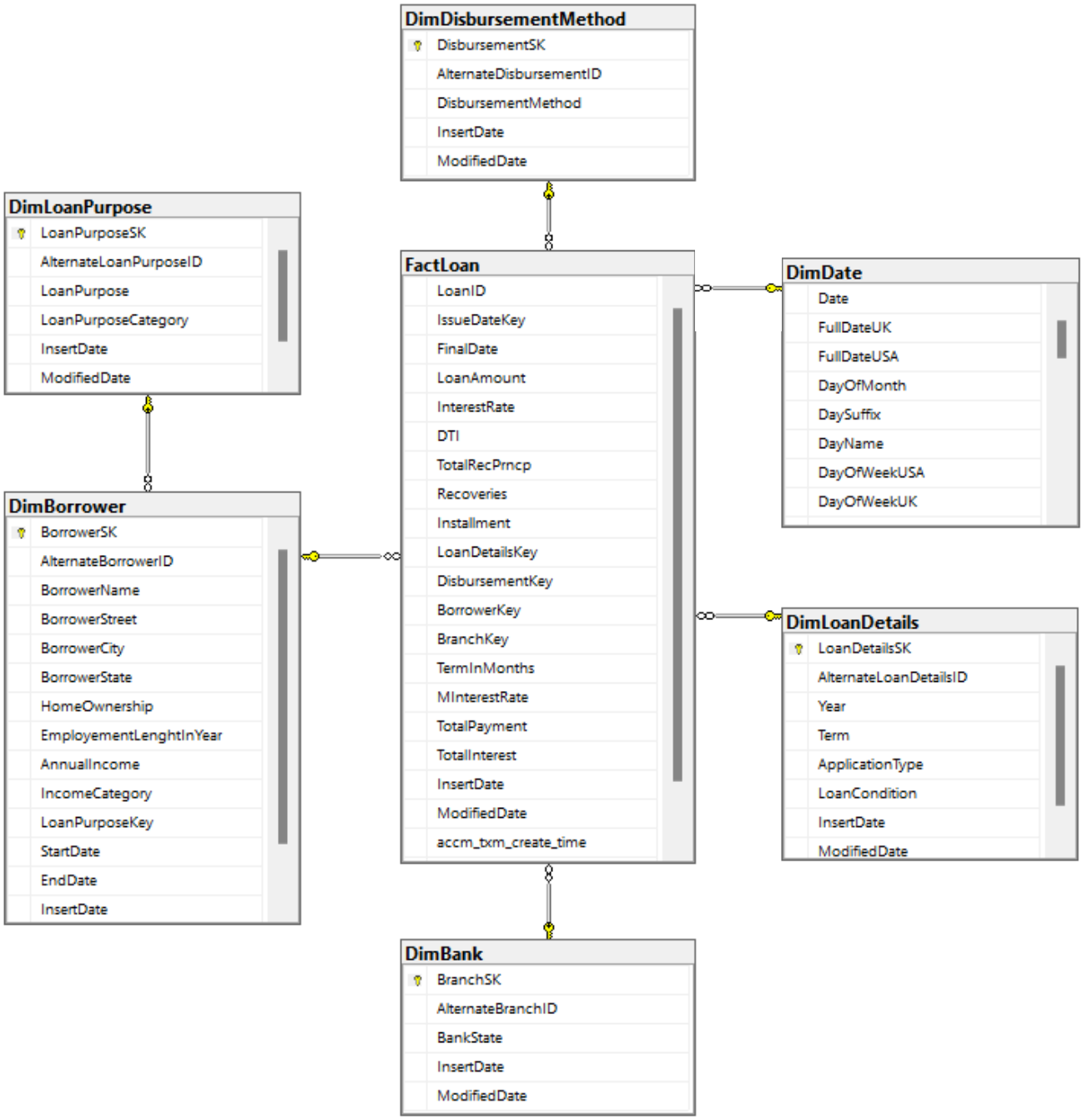
- dbo.FactLoan

Hierarchies:

- DimDate hierarchical breakdown from Year, Quarter, Week, Date
- DimBorrower hierarchical breakdown from BorrowerState, BorrowerCity, BorrowerStreet

Assumptions:

- The table "Borrower" was taken as the slowly changing dimension.


DimDisbursementMethod
DisbursementSK
AlternateDisbursementID
DisbursementMethod
InsertDate
ModifiedDate
DimLoanPurpose
LoanPurposeSK
AlternateLoanPurposeID
LoanPurpose
LoanPurposeCategory
InsertDate
ModifiedDate
FactLoan
LoanID
IssueDateKey
FinalDate
LoanAmount
InterestRate
DTI
TotalRecPrncp
Recoveries
Installment
LoanDetailsKey
DisbursementKey
BorrowerKey
BranchKey
TermInMonths
MInterestRate
TotalPayment
TotalInterest
InsertDate
ModifiedDate
accm_txm_create_time
DimDate
Date
FullDateUK
FullDateUSA
DayOfMonth
DaySuffix
DayName
DayOfWeekUSA
DayOfWeekUK
DimBorrower
BorrowerSK
AlternateBorrowerID
BorrowerName
BorrowerStreet
BorrowerCity
BorrowerState
HomeOwnership
EmployementLenghtInYear
AnnualIncome
IncomeCategory
LoanPurposeKey
StartDate
EndDate
InsertDate
DimLoanDetails
LoanDetailsSK
AlternateLoanDetailsID
Year
Term
ApplicationType
LoanCondition
InsertDate
ModifiedDate
DimBank
BranchSK
AlternateBranchID
BankState
InsertDate
ModifiedDate

Calculations in Fact Table

- Monthly Interest Rate

```
MInterestRate as
([InterestRate]/(100*12))
```

- Total Payment

```
TotalPayment as
((([LoanAmount]*(([InterestRate]/(1200))*power((1)+[InterestRate]/
(1200),[TermInMonths])))/(power((1)+[InterestRate]/(1200),
[TermInMonths])-(1)))*[TermInMonths])
```

- Total Interest

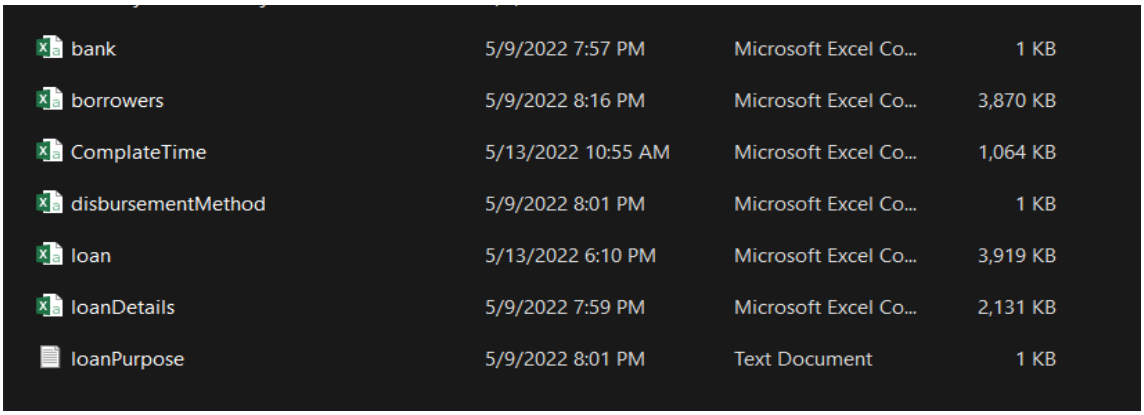
```
TotalInterest as
((([LoanAmount]*(([InterestRate]/(1200))*power((1)+[InterestRate]/
(1200),[TermInMonths])))/(power((1)+[InterestRate]/(1200),
[TermInMonths])-(1)))*[TermInMonths]-[LoanAmount])
```

Step 5: ETL development

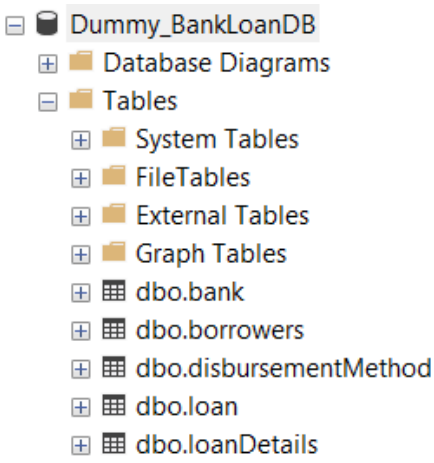
ETL development process

- ✓ Step 01 : Setting up the Environment

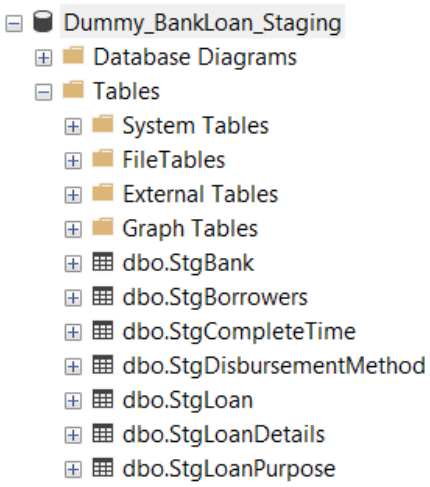
Text and Excel File



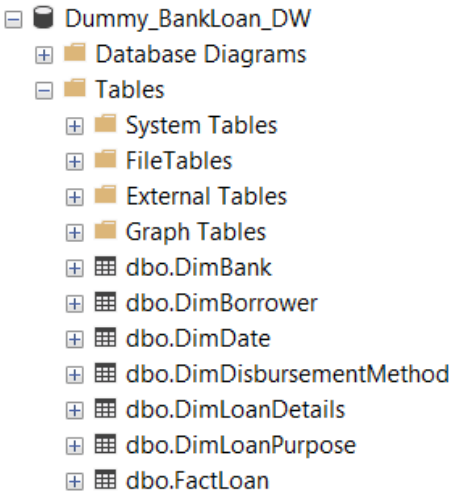
SourceDB in ssms



Staging in ssms

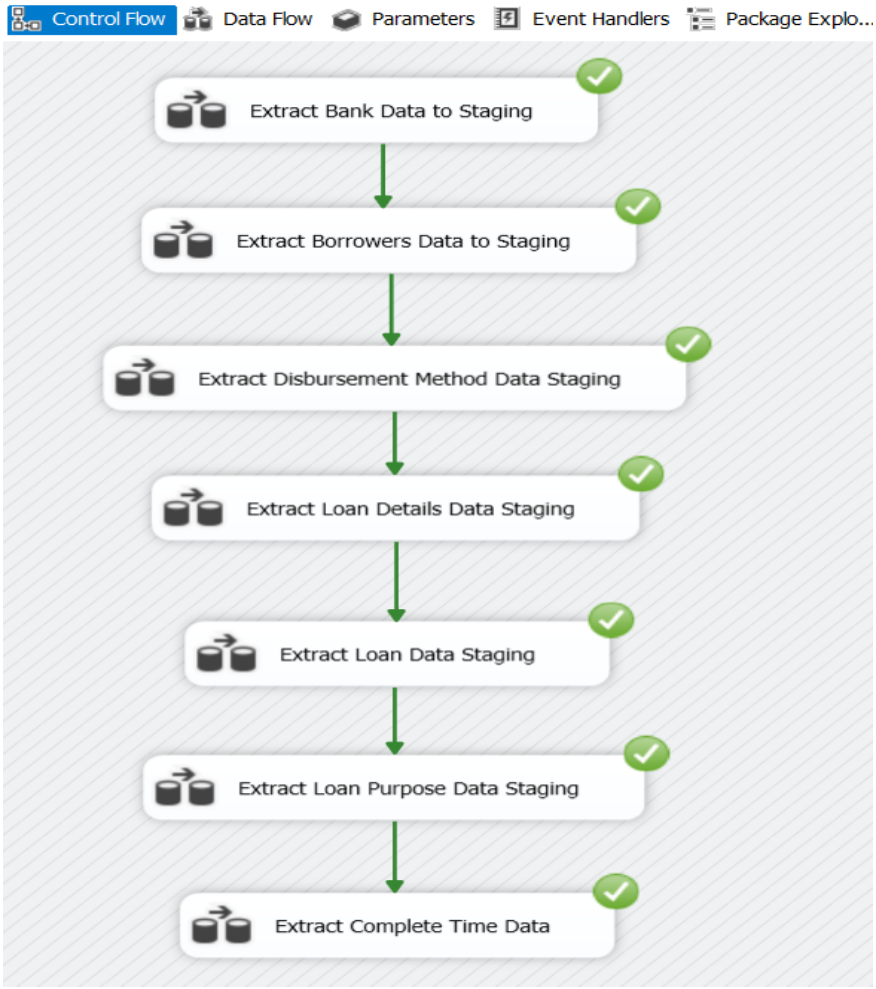


Data Warehouse in ssms

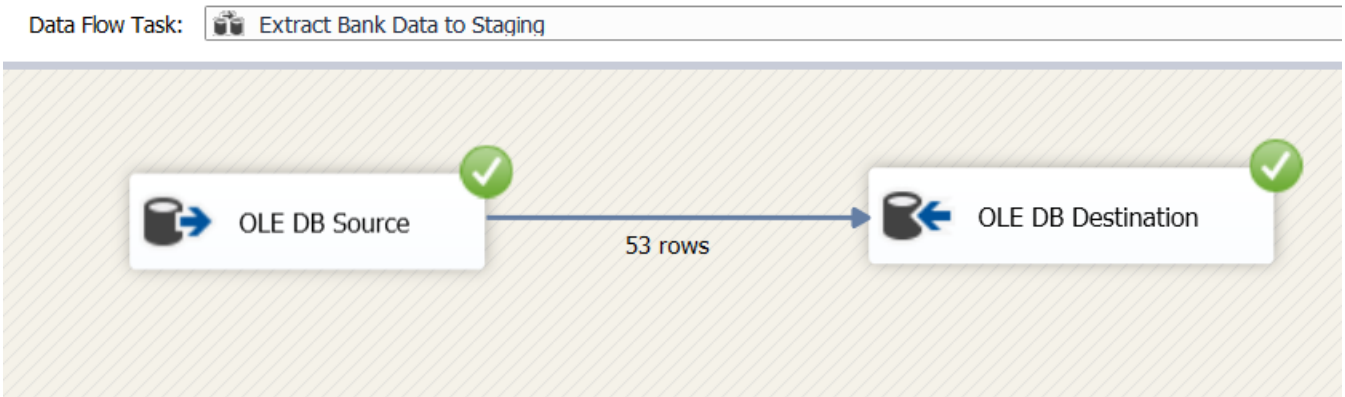


✓ Step 02 : Data Extracting from source to staging tables

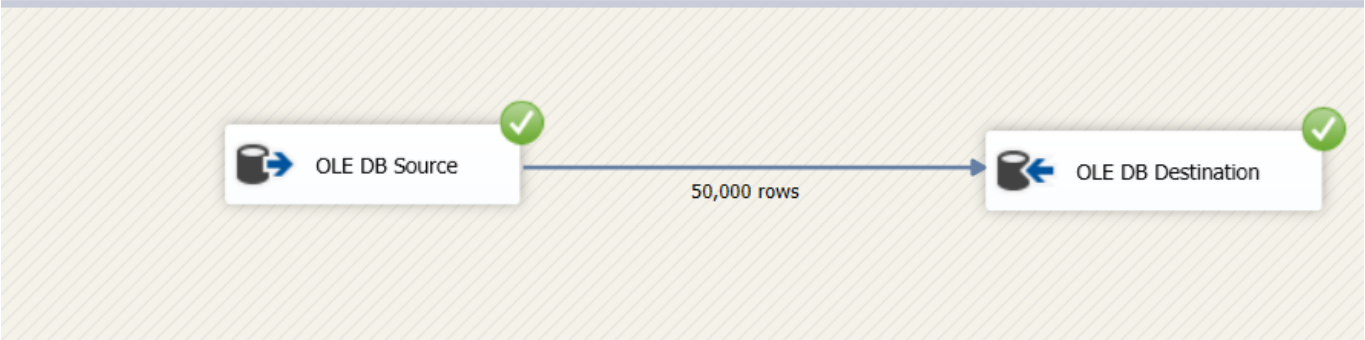
Control Flow



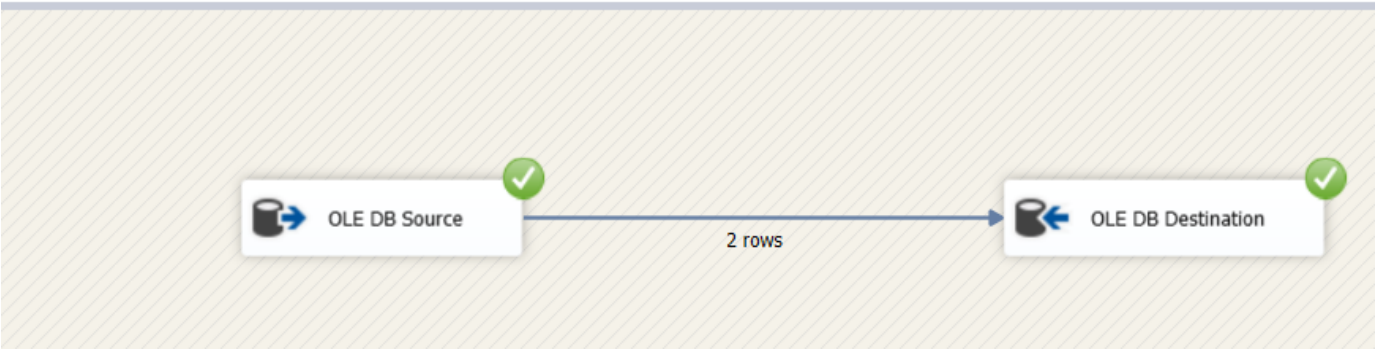
Data Flow



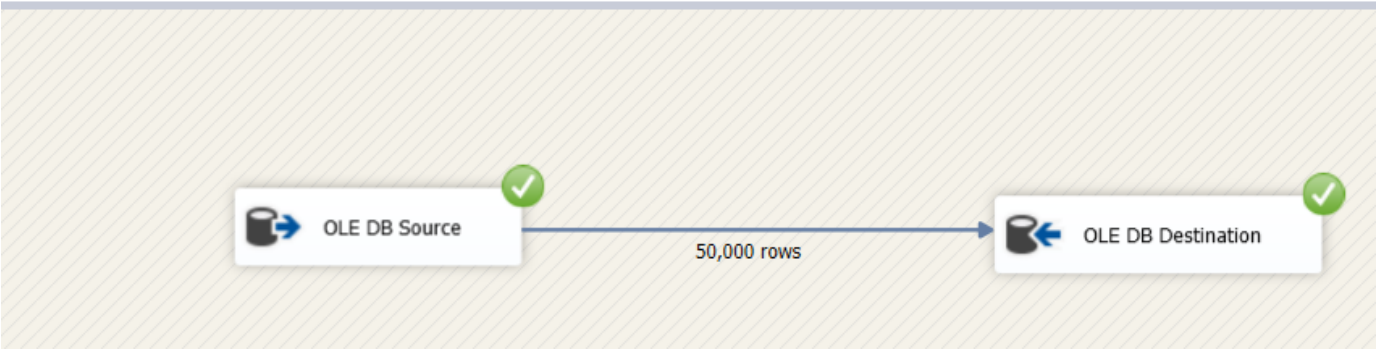
Data Flow Task: Extract Borrowers Data to Staging



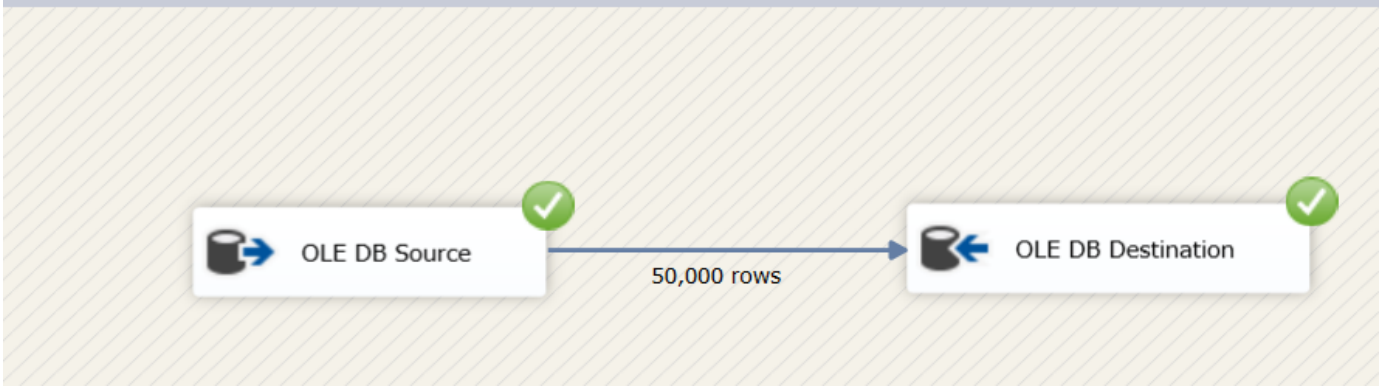
Data Flow Task: Extract Disbursement Method Data Staging

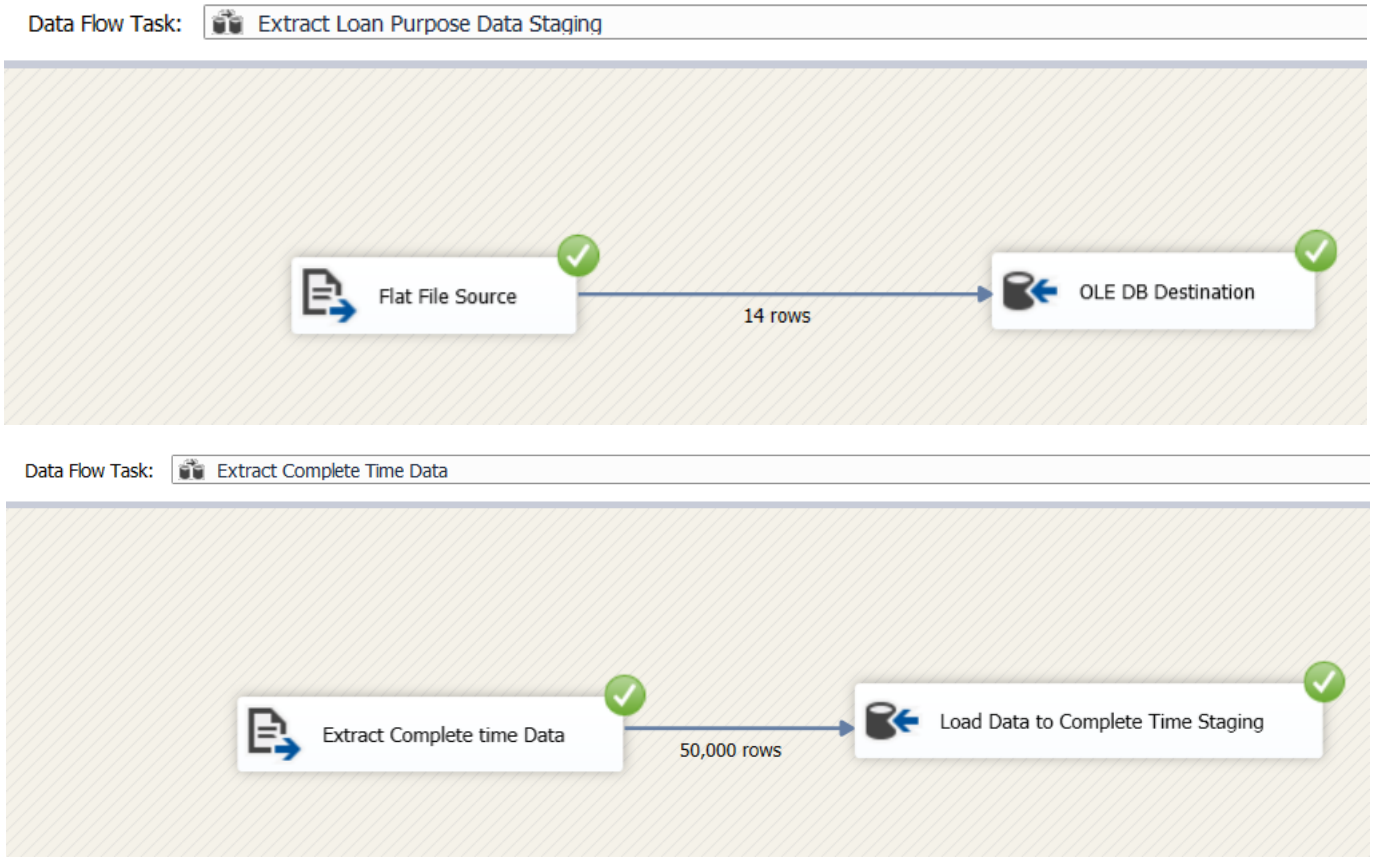


Data Flow Task: Extract Loan Details Data Staging

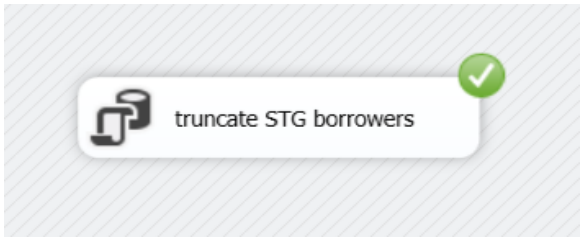


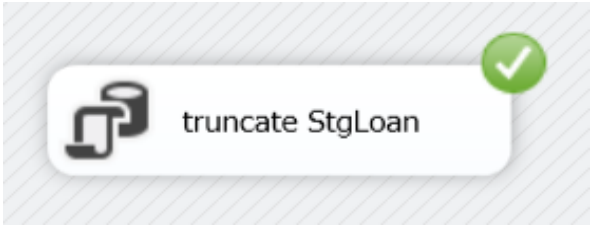
Data Flow Task: Extract Loan Data Staging

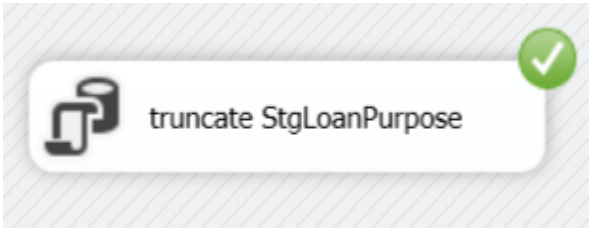




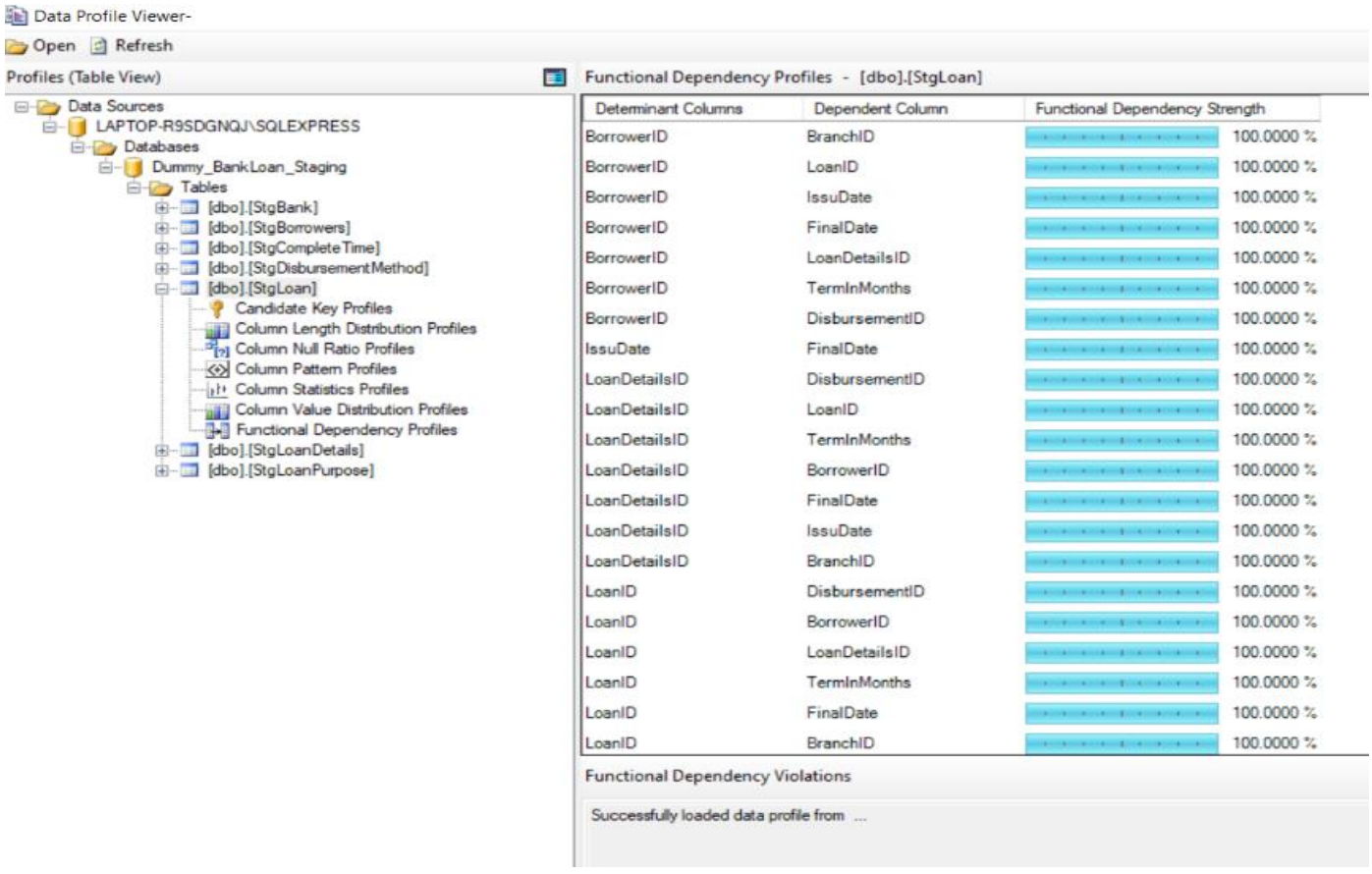
Event Handlers





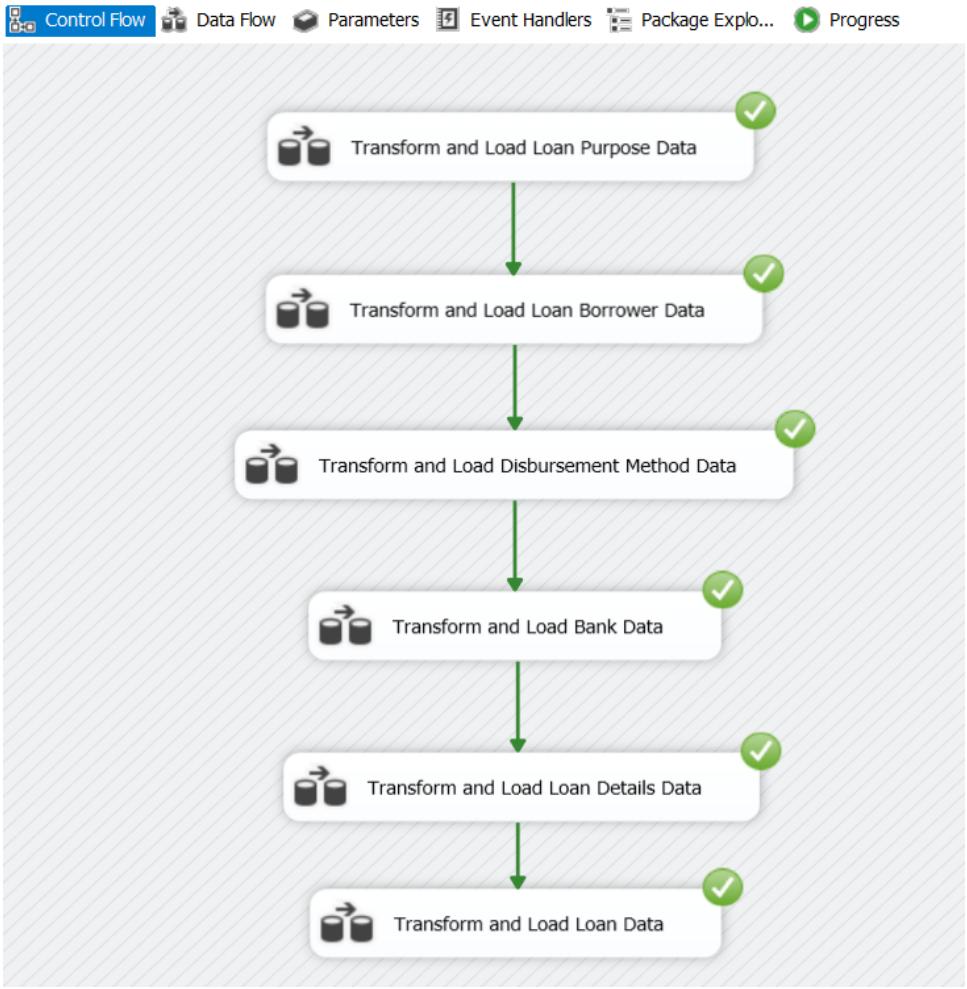


✓ Step 03 : Data Profiling

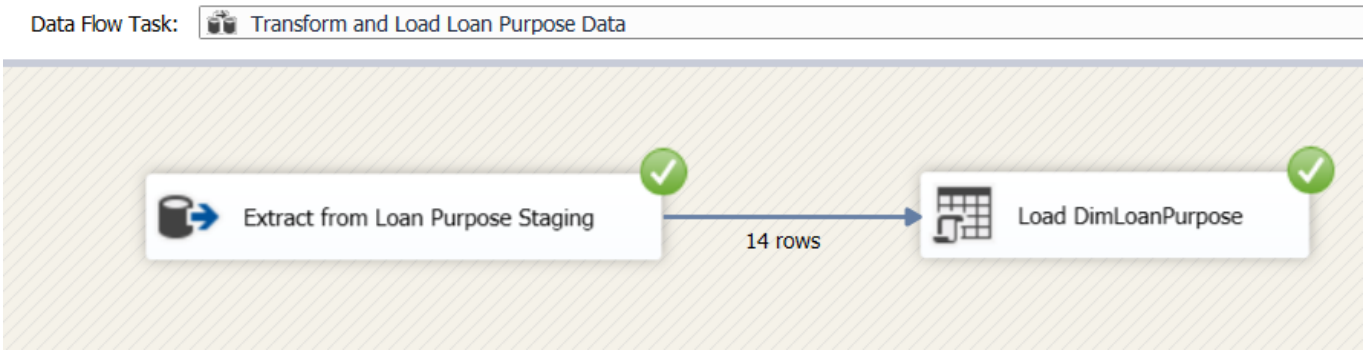


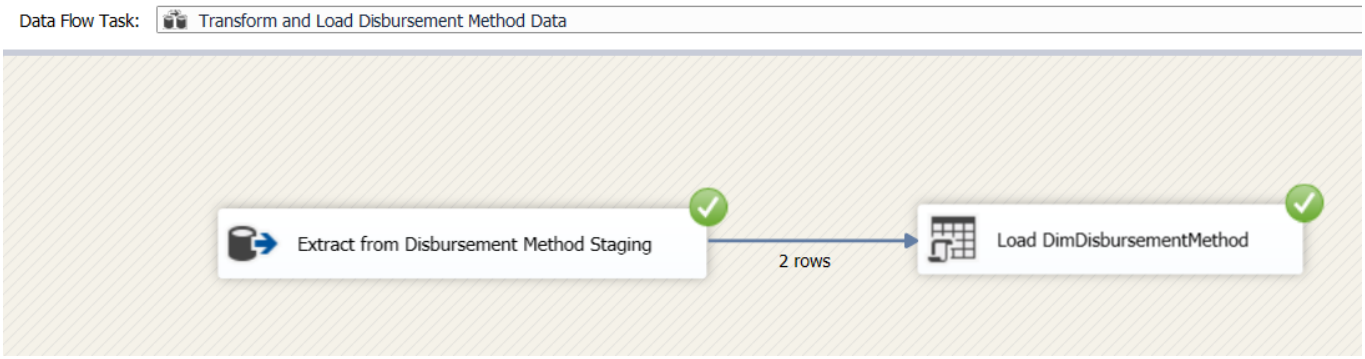
✓ Step 04 : Data Transformation from staging to warehouse

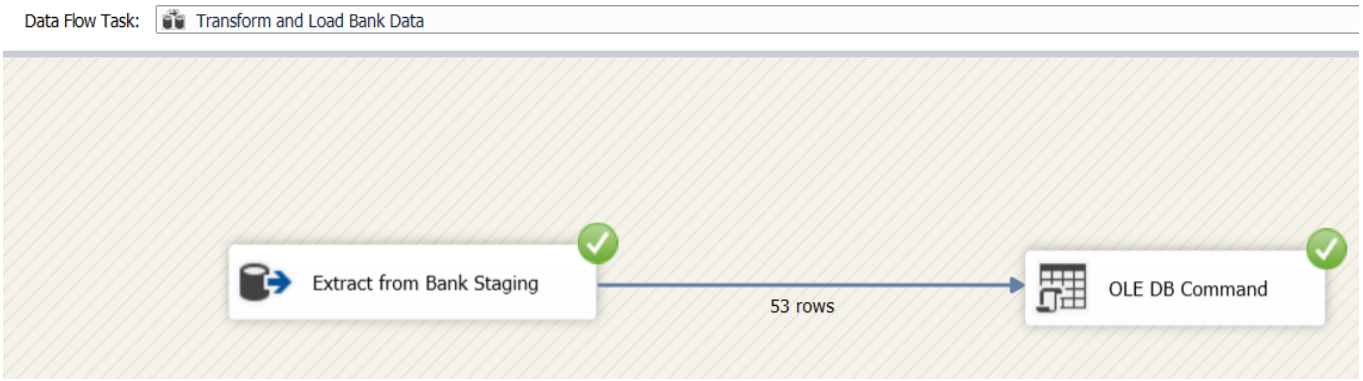
Control Flow

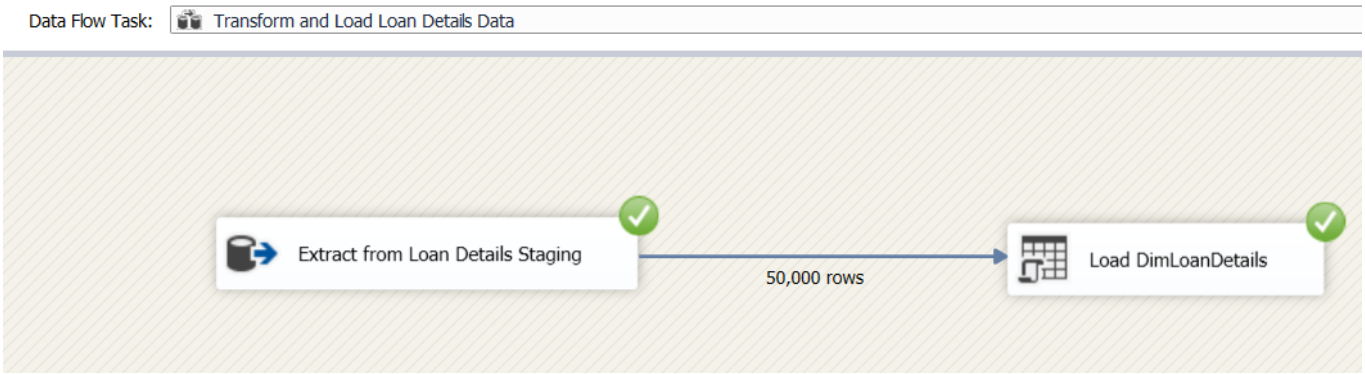


Data Flow

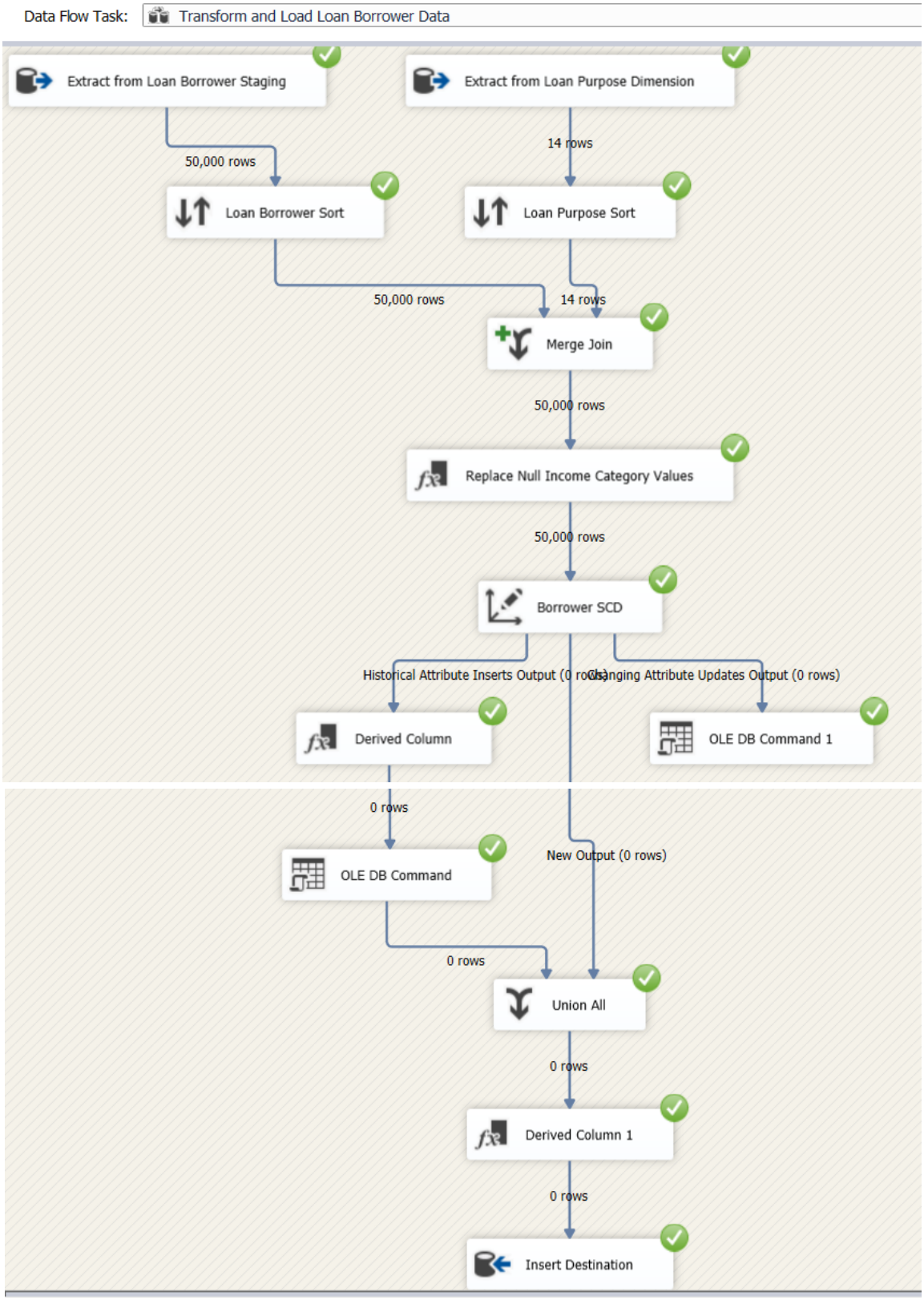



Data Flow Task:
Transform and Load Disbursement Method Data
Extract from Disbursement Method Staging
2 rows
Load DimDisbursementMethod

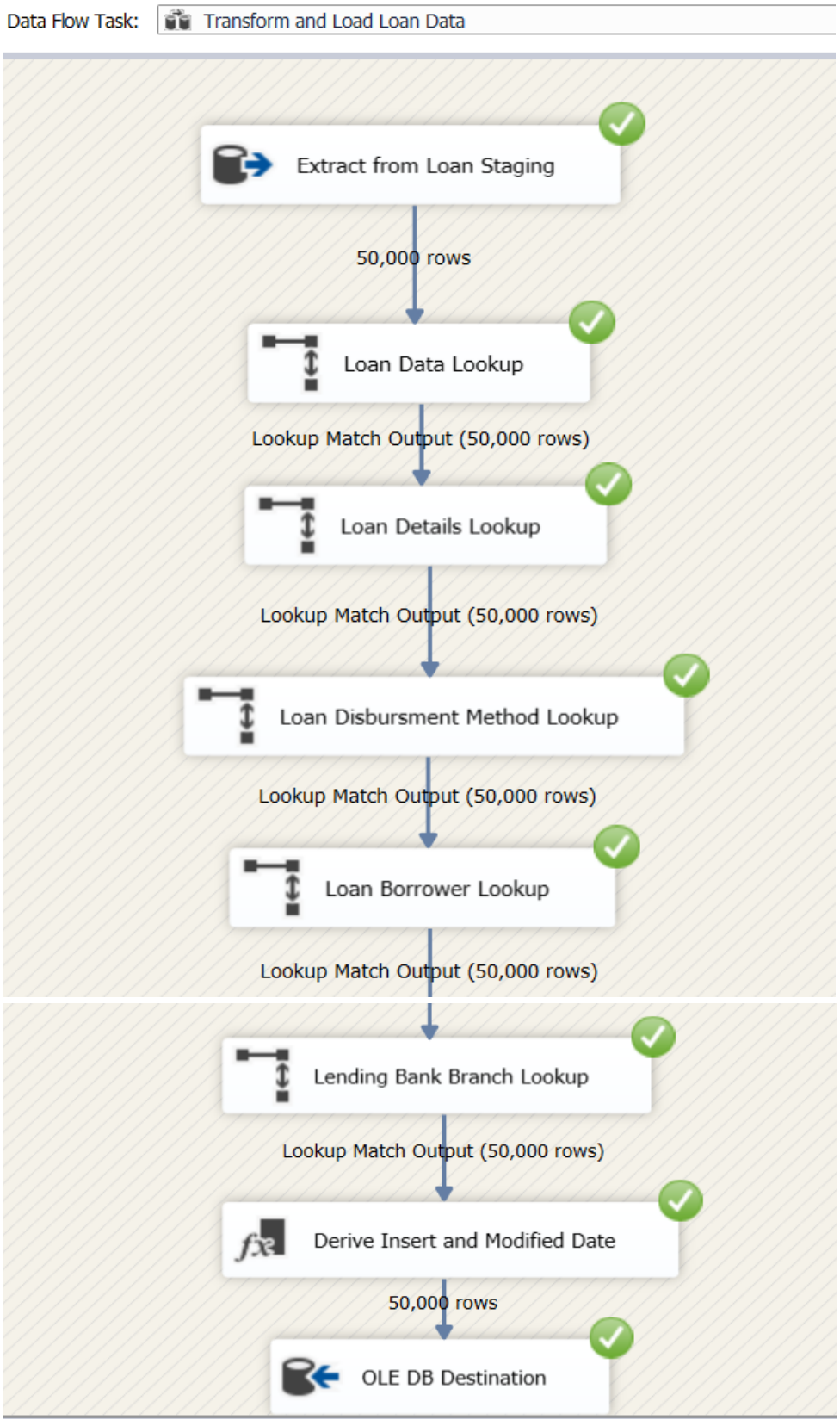

Data Flow Task:
Transform and Load Bank Data
Extract from Bank Staging
53 rows
OLE DB Command


Data Flow Task:
Transform and Load Loan Details Data
Extract from Loan Details Staging
50,000 rows
Load DimLoanDetails

Transforming, Loading Borrower with a Slowly Changing Dimension



Transforming, Loading Fact Table



Procedures

```
-------Procedure for DimLoanPurpose---------
CREATE PROCEDURE dbo.UpdateDimLoanPurpose
@LoanPurposeID varchar(50),
@LoanPurpose varchar(50),
@LoanPurposeCategory varchar(50)
AS BEGIN
if not exists (select LoanPurposeSK
from dbo.DimLoanPurpose
where AlternateLoanPurposeID = @LoanPurposeID)
BEGIN
insert into dbo.DimLoanPurpose
(AlternateLoanPurposeID, LoanPurpose, LoanPurposeCategory, InsertDate, ModifiedDate)
values
(@LoanPurposeID, @LoanPurpose, @LoanPurposeCategory, GETDATE(), GETDATE())
END;
if exists (select LoanPurposeSK
from dbo.DimLoanPurpose
where AlternateLoanPurposeID = @LoanPurposeID)
BEGIN
update dbo.DimLoanPurpose
set LoanPurpose = @LoanPurpose, LoanPurposeCategory = @LoanPurposeCategory, ModifiedDate
= GETDATE()
where AlternateLoanPurposeID = @LoanPurposeID
END;
END;
```

```
-------Procedure for DimDisbursementMethod---------
CREATE PROCEDURE dbo.UpdateDimDisbursementMethod
@DisbursementID nvarchar(50),
@DisbursementMethod nvarchar(50)
AS BEGIN
if not exists (select DisbursementSK
from dbo.DimDisbursementMethod
where AlternateDisbursementID = @DisbursementID)
BEGIN
insert into dbo.DimDisbursementMethod
(AlternateDisbursementID, DisbursementMethod, InsertDate, ModifiedDate)
values
(@DisbursementID, @DisbursementMethod, GETDATE(), GETDATE())
END;
if exists (select DisbursementSK
from dbo.DimDisbursementMethod
where AlternateDisbursementID = @DisbursementID)
BEGIN
update dbo.DimDisbursementMethod
set DisbursementMethod = @DisbursementMethod, ModifiedDate = GETDATE()
where AlternateDisbursementID = @DisbursementID
END;
END;
```

```
-------Procedure for DimBank---------
CREATE PROCEDURE dbo.UpdateDimBank
@BranchID nvarchar(50),
@BankState nvarchar(50)
AS BEGIN
if not exists (select BranchSK
from dbo.DimBank
where AlternateBranchID = @BranchID)
BEGIN
insert into dbo.DimBank
(AlternateBranchID, BankState, InsertDate, ModifiedDate)
values
(@BranchID, @BankState, GETDATE(), GETDATE())
END;
if exists (select BranchSK
from dbo.DimBank
where AlternateBranchID = @BranchID)
BEGIN
update dbo.DimBank
set BankState = @BankState, ModifiedDate = GETDATE()
where AlternateBranchID = @BranchID
END;
END;
```

```
-------Procedure for DimLoanDetails---------
CREATE PROCEDURE dbo.UpdateDimLoanDetails
@LoanDetailsID nvarchar(50),
@Year nvarchar(50),
@Term nvarchar(50),
@ApplicationType nvarchar(50),
@LoanCondition nvarchar(50)
AS BEGIN
if not exists (select LoanDetailsSK
from dbo.DimLoanDetails
where AlternateLoanDetailsID = @LoanDetailsID)
BEGIN
insert into dbo.DimLoanDetails
(AlternateLoanDetailsID, Year, Term, ApplicationType, LoanCondition, InsertDate,
ModifiedDate)
values
(@LoanDetailsID, @Year, @Term, @ApplicationType, @LoanCondition, GETDATE(), GETDATE())
END;
if exists (select LoanDetailsSK
from dbo.DimLoanDetails
where AlternateLoanDetailsID = @LoanDetailsID)
BEGIN
update dbo.DimLoanDetails
set Year = @Year, Term = @Term, ApplicationType = @ApplicationType, LoanCondition =
@LoanCondition, ModifiedDate = GETDATE()
where AlternateLoanDetailsID = @LoanDetailsID
END;
END;
```

Step 6: ETL development – Accumulating fact tables

✓ Step 01 : Extending Fact Table

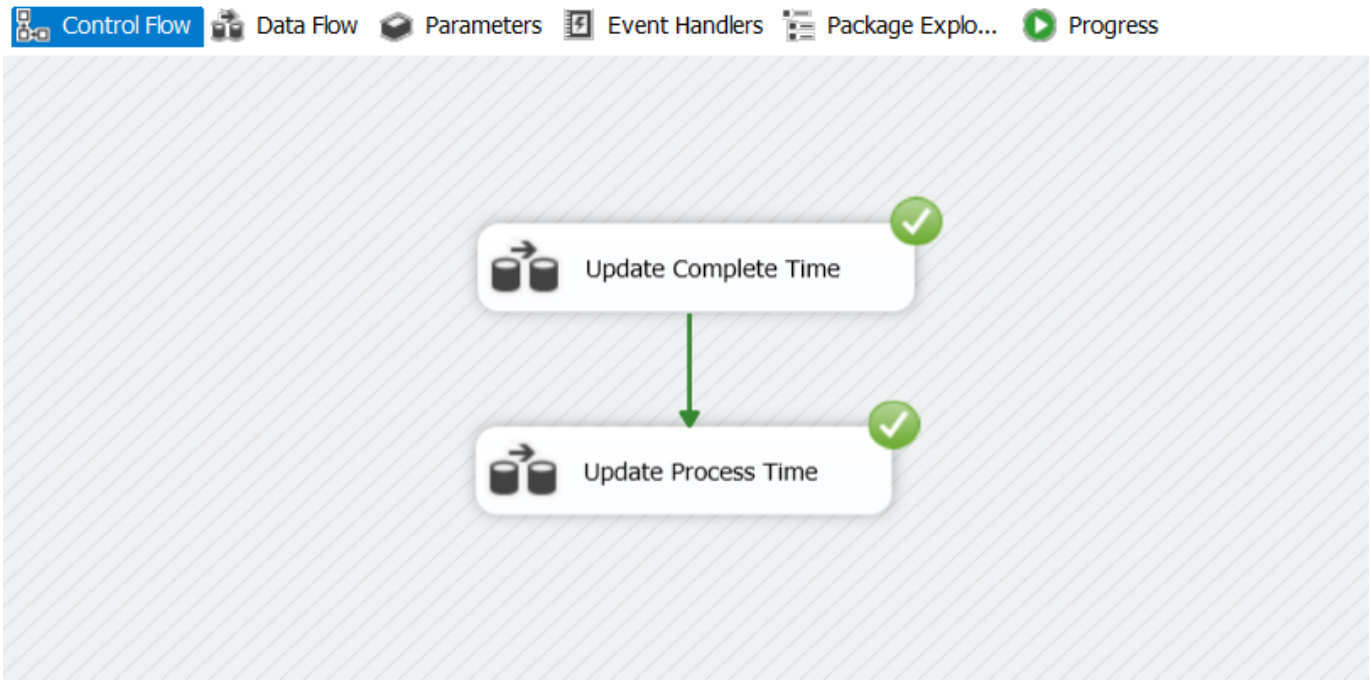
- dbo.FactLoan
  - Columns
    - LoanID (nvarchar(50), null)
    - IssueDateKey (int, null)
    - FinalDate (varchar(50), null)
    - LoanAmount (money, null)
    - InterestRate (real, null)
    - DTI (real, null)
    - TotalRecPrncp (money, null)
    - Recoveries (real, null)
    - Installment (real, null)
    - LoanDetailsKey (FK, int, null)
    - DisbursementKey (FK, int, null)
    - BorrowerKey (FK, int, null)
    - BranchKey (FK, int, null)
    - TermInMonths (int, null)
    - MInterestRate (Computed, real, null)
    - TotalPayment (Computed, float, null)
    - TotalInterest (Computed, float, null)
    - InsertDate (datetime, null)
    - ModifiedDate (datetime, null)
    - accm_txm_create_time (datetime, null)
    - accm_txm_complete_time (datetime, null)
    - txn_process_time_hours (int, null)

✓ Step 02 : Create Data set for Complete Time

ComplateTime	5/13/2022 10:55 AM	Microsoft Excel Co...	1,064 KB

✓ Step 03 : Update Complete Time and Process Time in Fact Table

Control Flow



Data Flow