

Sri Lanka Institute of Information Technology



Data Warehousing and Business Intelligence
(IT3021)

SBA Guaranteed 7(a) Loan Program

Assignment 1

IT20202736

Withanage W.D.U.I.

Y3S1 Group 5.2 (DS – Weekday)

CONTENTS

STEP 1: DATA SET SELECTION	3
BACKGROUND	3
ER DIAGRAM	4
STEP 2: PREPARATION OF DATA SOURCES	5
DESCRIPTION OF THE DATASET.....	5
STEP 3: SOLUTION ARCHITECTURE	7
STEP 4: DATA WAREHOUSE DESIGN & DEVELOPMENT	8
DIMENSIONAL MODEL.....	8
DIMENSIONS AND FACT TABLES:	8
HIERARCHIES:	9
ASSUMPTIONS:	9
CALCULATIONS FOR THE FACT TABLE	9
STORED PROCEDURES.....	9
STEP 5: ETL DEVELOPMENT	13
ETL DEVELOPMENT PROCESS	13
STEP 6: ETL DEVELOPMENT – ACCUMULATING FACT TABLES.....	18

Step 1: Data set selection

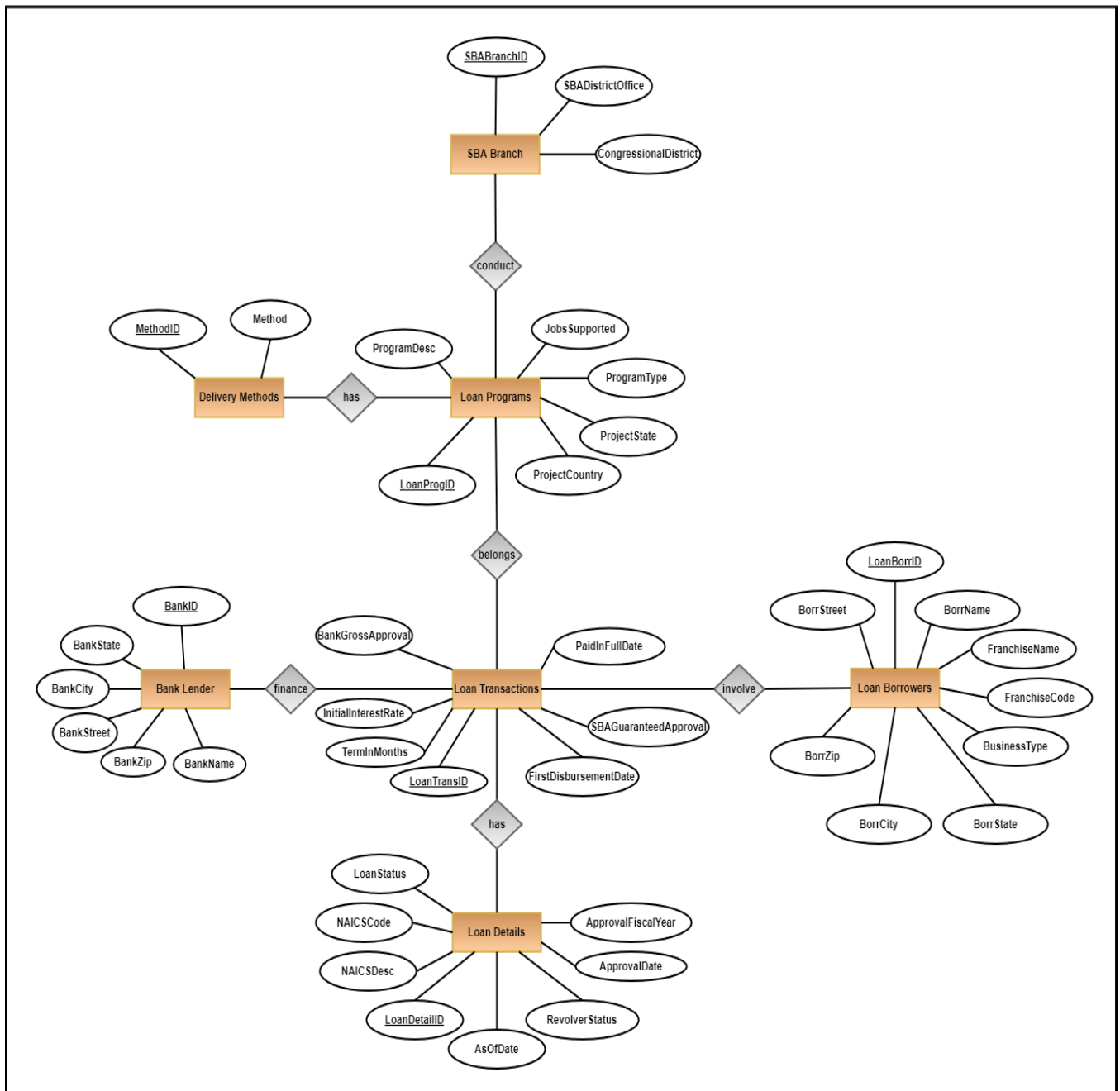
Background

Link to the selected source data set: <https://data.world/nerb/sba-loan-guarantee-data>

Note: Above dataset includes U.S. Small Business Administration 7(a) & 504 Detailed Loan Data since 1990. For the assignment purpose I have selected data related to 7(a) loan programs only and some modifications have been done to the data set to get a better ETL process.

The U.S. Small Business Administration (SBA) helps small businesses get funding by setting guidelines for loans and reducing lender risk. These SBA-backed loans make it easier for small businesses to get the funding they need. The SBA is not a direct lender for those loans, but rather is a guarantor to a loan that is financed by a bank. 7(a) is a such loan program conducted by SBA and the branches of SBA are involved for the loan transaction process. For a particular 7(a) loan program there are several delivery methods which informs how the loan amount is delivered to the relevant loan borrowers who are small business owners. For each loan transaction which belongs to a particular 7(a) loan program, a loan borrower and a lending bank is involved. All the details of a particular loan transaction are also recorded with relevant dates. Lenders appreciate SBA loan programs because the SBA shoulders some of the risk for the lender by guaranteeing a portion of the loan amount. Because of that guarantee, lenders can offer more flexible payment terms and lower interest rates than most small businesses would otherwise be able to get. To get more clear idea about the task an Entity Relationship diagram is provided below.

ER Diagram



Step 2: Preparation of data sources

The original data source contained one xlsx file and I separated files according to requirements as follows.

Text files: BankLender, DeliveryMethods, LoanBorrower, SBABranch

Database File: SBA_7a_LoanProgram_SourceDB (LoansDetails, LoanProgram, LoansTransactions tables)

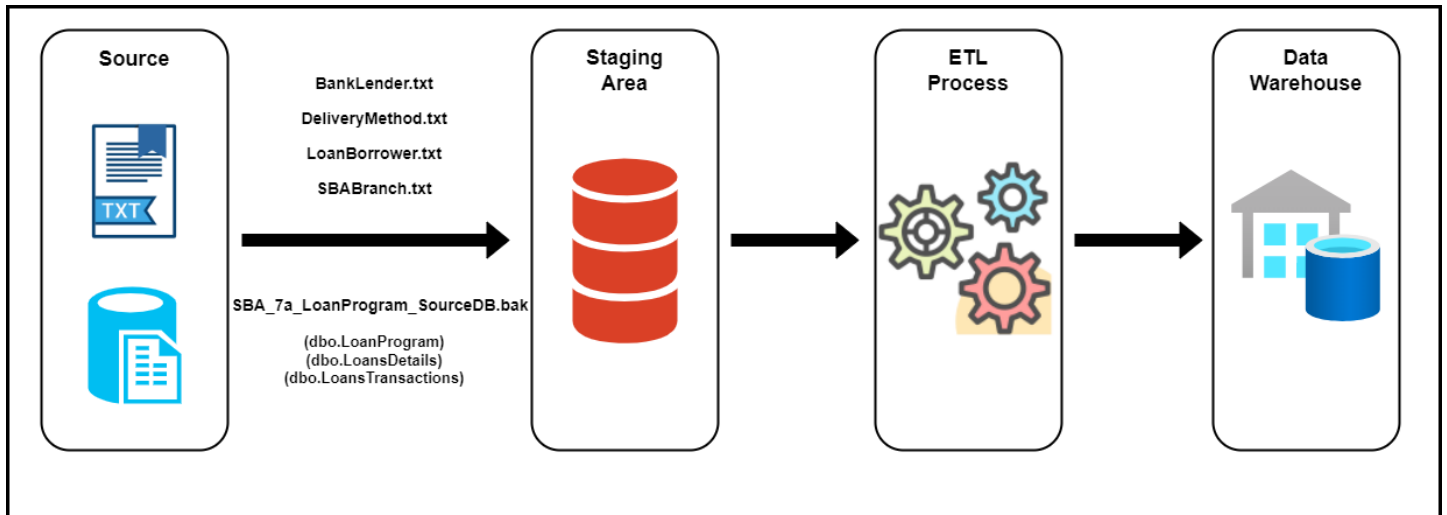
Description of the Dataset

Table Name	Column Name	Data Type	Description
Loan Borrower	LoanBorrID BorrName BorrStreet BorrCity BorrState BorrZip BusinessType FranchiseCode FranchiseName	Varchar (20) Nvarchar (250) Nvarchar (250) Nvarchar (250) Nvarchar (250) Int Nvarchar (250) Nvarchar (250) Nvarchar (250)	Details about all loan borrowers
Bank Lender	BankID BankName BankStreet BankCity BankState BankZip	Int Nvarchar (250) Nvarchar (250) Nvarchar (250) Nvarchar (250) Int	Details about all bank lenders
SBA Branch	SBABranchID SBADistrictOffice CongressionalDistrict	Nvarchar (50) Nvarchar (250) Nvarchar (50)	Details about SBA Company Branch
Delivery Method	MethodID Method	Varchar (20) Nvarchar (250)	Details about loan delivery method
Loan Program	LoanProgID ProjectCountry ProjectState ProgramType ProgramDesc JobsSupported MethodID SBABranchID	Varchar (20) Nvarchar (250) Nvarchar (250) Nvarchar (250) Nvarchar (250) Varchar (20) Varchar (20) Nvarchar (50)	Details about loan programs

Loans Transactions	LoanTransID BankGrossApproval SBAGuaranteedApproval FirstDisbursementDate InitialInterestRate TermInMonths PaidInFullDate LoanBorrID BankID LoanDetailID LoanProgID	Int Money Money Datetime Real Int Datetime Varchar (20) Int Int Varchar (20)	All details about loan transactions
Loans Details	LoanDetailID NAICSCode NAICSDescription LoanStatus RevolverStatus AsOfDate ApprovalDate ApprovalFiscalYear	Int Int Nvarchar (250) Nvarchar (250) Int Datetime Datetime Int	All loan details

Step 3: Solution architecture

The architectural diagram provided below describes the components of the Datawarehouse solution.



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 database files and the formats are stored in a local folder.
 - **Staging area:** In this, it extracts data from sources and load data into the staging area. Through staging area data can be moved from the sources to the DWH.
 - **ETL:** ETL is performed in two occasions. First is when extracting data from the sources and loading into staging area and secondly when extracting data from staging and do necessary transformations and loading them to data warehouse.
 - **Data Warehouse:** Data Warehouse supports Business Intelligence activities such as analytics.

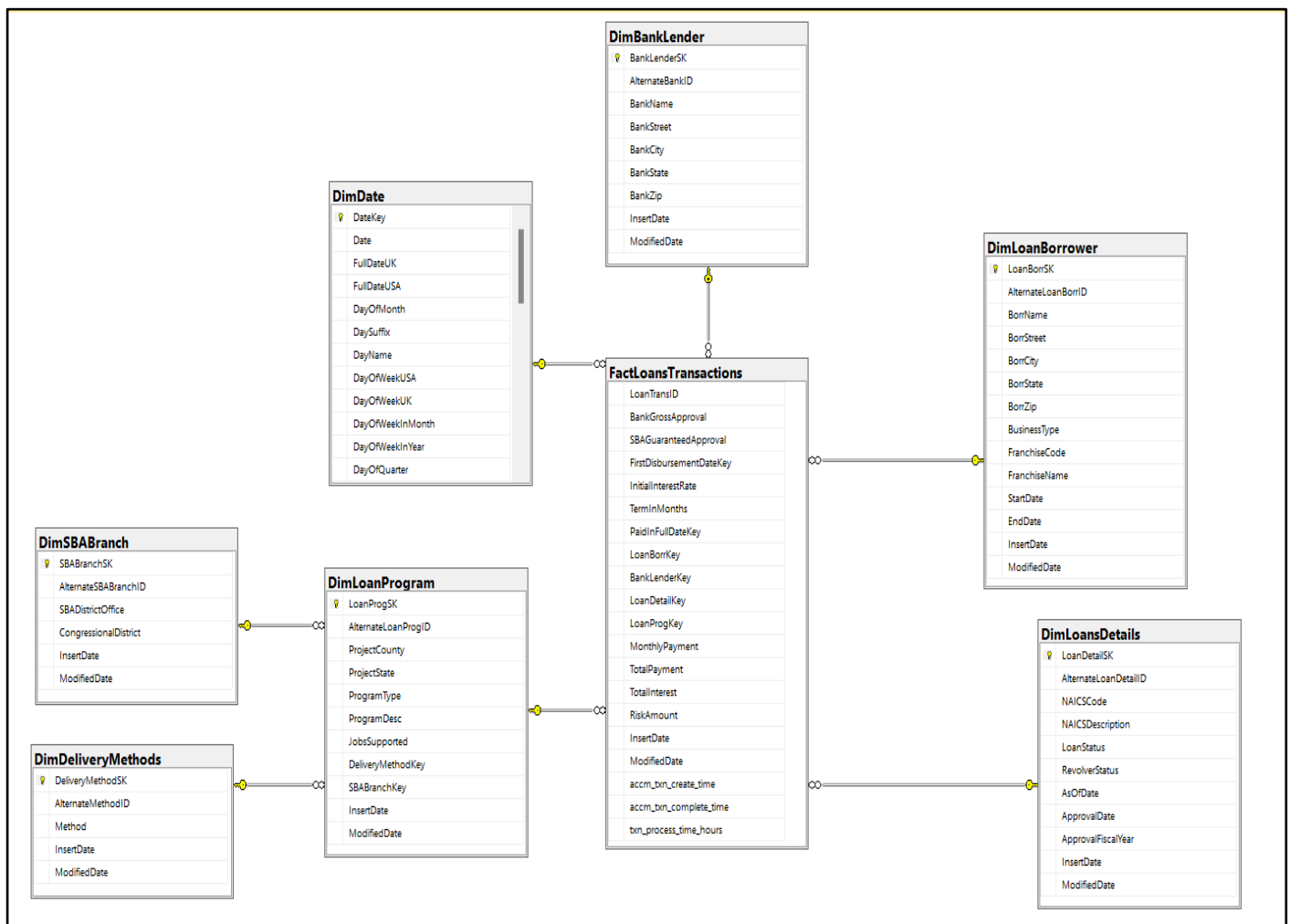
Step 4: Data warehouse design & development

Dimensional Model

Snowflake schema was selected to design the Data Warehouse of **SBA Guaranteed 7(a) Loan Program Data** according to the behavior and the number of dimensional tables and fact tables. All the dimensional tables are connected with the fact table.

Dimensions and Fact tables:

- DimLoanBorrower → Slowly Changing Dimension
- DimBankLender
- DimDeliveryMethods
- DimSBABranch
- DimLoanProgram
- DimLoansDetails
- DimDate
- FactLoansTransactions → Fact Table



Hierarchies:

Hierarchies in DimBankLender dimension: Street → City → State

Hierarchies in DimLoanBorrower dimension: Street → City → State

Hierarchies in DimDate dimension: Day → Month → Quarter → Year

Assumptions:

In DimLoanBorrower slowly changing dimension, Borrower City, State, Street and Zip code are considered as Historical Attributes and Franchise Code and Franchise Name are considered as Changing Attributes.

Calculations for the Fact Table

```
[MonthlyPayment] AS  
((( [BankGrossApproval] * ((( [InitialInterestRate] / (1200)) * power((1) + [InitialInterestRate] /  
(1200), [TermInMonths])))) / (power((1) + [InitialInterestRate] / (1200), [TermInMonths]) - (1)))  
[TotalPayment] AS  
((( [BankGrossApproval] * ((( [InitialInterestRate] / (1200)) * power((1) + [InitialInterestRate] /  
(1200), [TermInMonths])))) / (power((1) + [InitialInterestRate] / (1200), [TermInMonths]) -  
(1))) * [TermInMonths])  
[TotalInterest] AS  
((( [BankGrossApproval] * ((( [InitialInterestRate] / (1200)) * power((1) + [InitialInterestRate] /  
(1200), [TermInMonths])))) / (power((1) + [InitialInterestRate] / (1200), [TermInMonths]) -  
(1))) * [TermInMonths] - [BankGrossApproval])  
[RiskAmount] AS ([BankGrossApproval] - [SBAGuaranteedApproval])
```

Stored Procedures

Stored procedure for DimDeliveryMethods Dimension

```
CREATE PROCEDURE dbo.UpdateDimDeliveryMethods  
@MethodID varchar(20),  
@Method nvarchar(250)  
AS BEGIN  
if not exists (select DeliveryMethodSK  
from dbo.DimDeliveryMethods  
where AlternateMethodID = @MethodID)  
BEGIN  
insert into dbo.DimDeliveryMethods  
(AlternateMethodID, Method, InsertDate, ModifiedDate)  
values
```

```

(@MethodID, @Method, GETDATE(), GETDATE())
END;
if exists (select DeliveryMethodSK
from dbo.DimDeliveryMethods
where AlternateMethodID = @MethodID)
BEGIN
update dbo.DimDeliveryMethods
set Method = @Method, ModifiedDate = GETDATE()
where AlternateMethodID = @MethodID
END;
END;

```

Stored procedure for DimSBABranch Dimension

```

CREATE PROCEDURE dbo.UpdateDimSBABranch
@SBABranchID nvarchar(50),
@SBADistrictOffice nvarchar(250),
@CongressionalDistrict nvarchar(50)
AS BEGIN
if not exists (select SBABranchSK
from dbo.DimSBABranch
where AlternateSBABranchID = @SBABranchID)
BEGIN
insert into dbo.DimSBABranch
(AlternateSBABranchID, SBADistrictOffice, CongressionalDistrict, InsertDate,
ModifiedDate)
values
(@SBABranchID, @SBADistrictOffice, @CongressionalDistrict, GETDATE(), GETDATE())
END;
if exists (select SBABranchSK
from dbo.DimSBABranch
where AlternateSBABranchID = @SBABranchID)
BEGIN
update dbo.DimSBABranch
set SBADistrictOffice = @SBADistrictOffice, CongressionalDistrict =
@CongressionalDistrict, ModifiedDate = GETDATE()
where AlternateSBABranchID = @SBABranchID
END;
END;

```

Stored procedure for DimLoanProgram Dimension

```

CREATE PROCEDURE dbo.UpdateDimLoanProgram
@LoanProgID varchar(20),
@ProjectCountry nvarchar(250),
@ProjectState nvarchar(250),
@ProgramType nvarchar(250),
@ProgramDesc nvarchar(250),
@JobsSupported varchar(20),
@DeliveryMethodKey int,
@SBABranchKey int
AS
BEGIN
if not exists (select LoanProgSK
from dbo.DimLoanProgram
where AlternateLoanProgID = @LoanProgID)

```

```

BEGIN
insert into dbo.DimLoanProgram
(AlternateLoanProgID, ProjectCounty, ProjectState, ProgramType, ProgramDesc,
JobsSupported, DeliveryMethodKey, SBABranchKey, InsertDate, ModifiedDate)
values
(@LoanProgID, @ProjectCountry, @ProjectState, @ProgramType, @ProgramDesc,
@JobsSupported, @DeliveryMethodKey, @SBABranchKey, GETDATE(), GETDATE())
END;
if exists (select LoanProgSK
from dbo.DimLoanProgram
where AlternateLoanProgID = @LoanProgID)
BEGIN
update dbo.DimLoanProgram
set ProjectCounty = @ProjectCountry, ProjectState = @ProjectState, ProgramType =
@ProgramType, ProgramDesc = @ProgramDesc, JobsSupported = @JobsSupported,
DeliveryMethodKey = @DeliveryMethodKey, SBABranchKey = @SBABranchKey, ModifiedDate =
GETDATE()
where AlternateLoanProgID = @LoanProgID
END;
END;

```

Stored procedure for DimBankLender Dimension

```

CREATE PROCEDURE dbo.UpdateDimBankLender
@BankID int,
@BankName nvarchar(250),
@BankStreet nvarchar(250),
@BankCity nvarchar(250),
@BankState nvarchar(250),
@BankZip int
AS
BEGIN
if not exists (select BankLenderSK
from dbo.DimBankLender
where AlternateBankID = @BankID)
BEGIN
insert into dbo.DimBankLender
(AlternateBankID, BankName, BankStreet, BankCity, BankState, BankZip, InsertDate,
ModifiedDate)
values
(@BankID, @BankName, @BankStreet, @BankCity, @BankState, @BankZip, GETDATE(),
GETDATE())
END;
if exists (select BankLenderSK
from dbo.DimBankLender
where AlternateBankID = @BankID)
BEGIN
update dbo.DimBankLender
set BankName = @BankName, BankStreet = @BankStreet, BankCity = @BankCity, BankState =
@BankState, BankZip = @BankZip,
ModifiedDate = GETDATE()
where AlternateBankID = @BankID
END;
END;

```

Stored procedure for DimLoansDetails Dimension

```

CREATE PROCEDURE dbo.UpdateDimLoansDetails
@LoanDetailID int,
@NAICSCode int,
@NAICSDescription nvarchar(250),
@LoanStatus nvarchar(50),

```

```

@RevolverStatus int,
@AsOfDate datetime,
@ApprovalDate datetime,
@ApprovalFiscalYear int
AS
BEGIN
if not exists (select LoanDetailSK
from dbo.DimLoansDetails
where AlternateLoanDetailID = @LoanDetailID)
BEGIN
insert into dbo.DimLoansDetails
(AlternateLoanDetailID, NAICSCode, NAICSDescription, LoanStatus, RevolverStatus,
AsOfDate, ApprovalDate, ApprovalFiscalYear, InsertDate, ModifiedDate)
values
(@LoanDetailID, @NAICSCode, @NAICSDescription, @LoanStatus, @RevolverStatus,
@AsOfDate, @ApprovalDate, @ApprovalFiscalYear, GETDATE(), GETDATE())
END;
if exists (select LoanDetailSK
from dbo.DimLoansDetails
where AlternateLoanDetailID = @LoanDetailID)
BEGIN
update dbo.DimLoansDetails
set NAICSCode = @NAICSCode, NAICSDescription = @NAICSDescription, LoanStatus =
@LoanStatus, RevolverStatus = @RevolverStatus, AsOfDate = @AsOfDate, ApprovalDate =
@ApprovalDate, ApprovalFiscalYear = @ApprovalFiscalYear,
ModifiedDate = GETDATE()
where AlternateLoanDetailID = @LoanDetailID
END;
END;









```

Step 5: ETL development

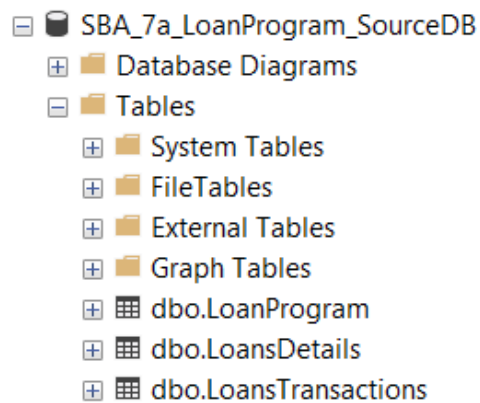
ETL development process

✓ Step 01: Setting up the Environment

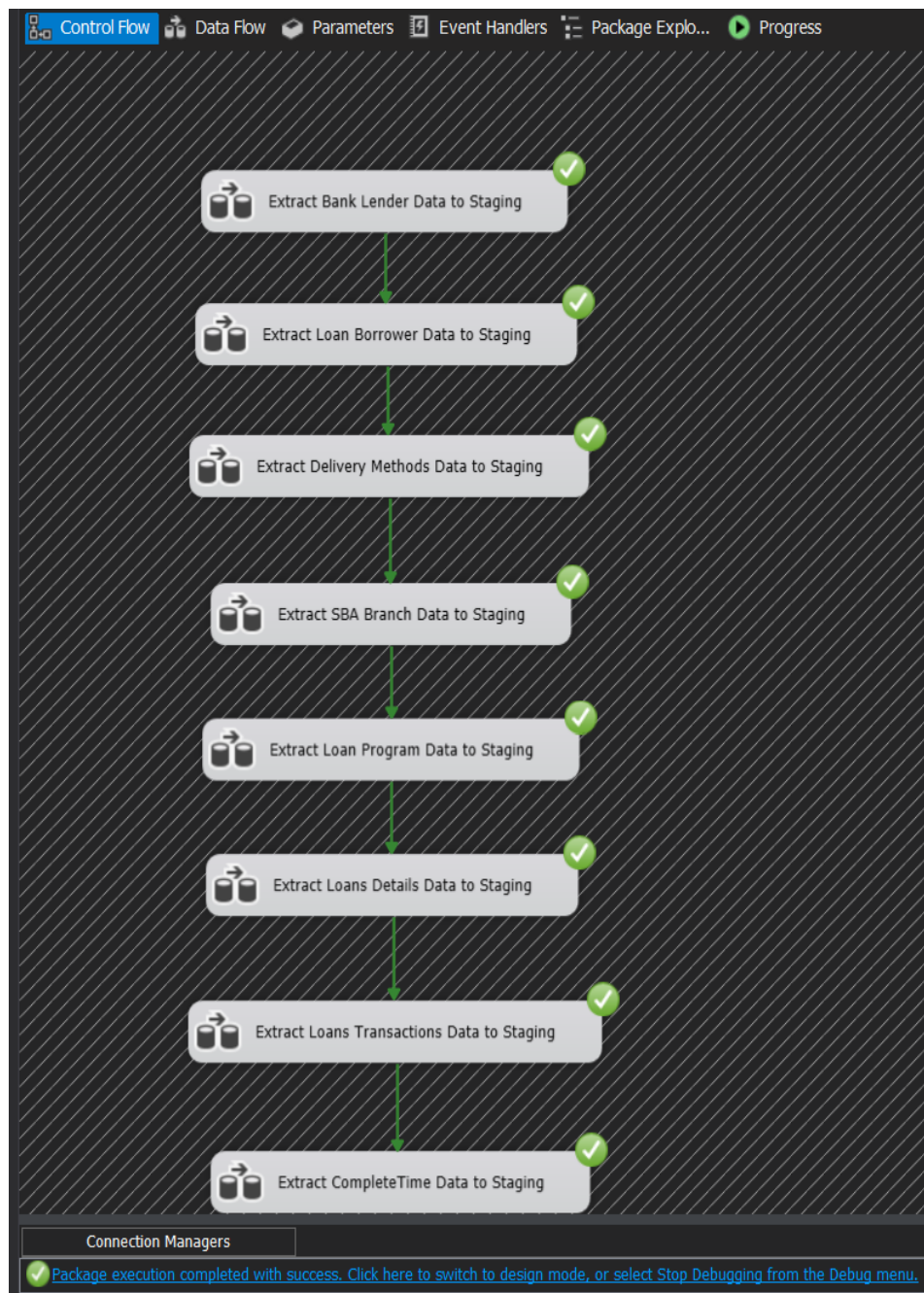
Text and CSV Files

	BankLender	5/6/2022 11:31 PM	Text Document	3,444 KB
	CompleteTime	5/12/2022 8:35 PM	Microsoft Excel Co...	1,064 KB
	DeliveryMethods	5/6/2022 11:32 PM	Text Document	1 KB
	LoanBorrower	5/6/2022 11:30 PM	Text Document	4,284 KB
	LoanProgram	5/7/2022 10:24 AM	Microsoft Excel Co...	2,753 KB
	LoansDetails	5/7/2022 12:57 PM	Microsoft Excel Co...	3,951 KB
	LoansTransactions	5/12/2022 12:17 PM	Microsoft Excel Co...	3,602 KB
	SBABranch	5/6/2022 11:31 PM	Text Document	19 KB

SourceDB in SSMS

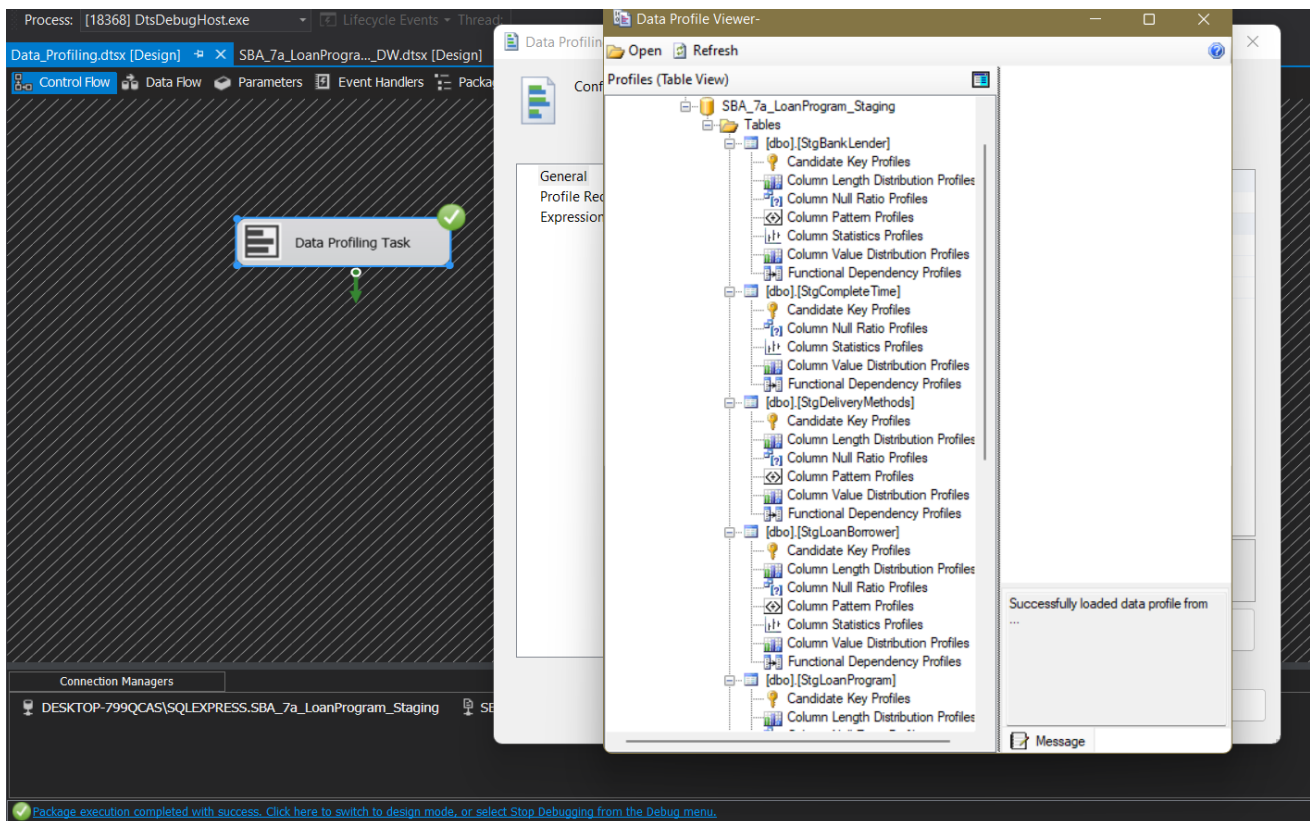


✓ Step 02: Data Extracting from Source to Staging Area



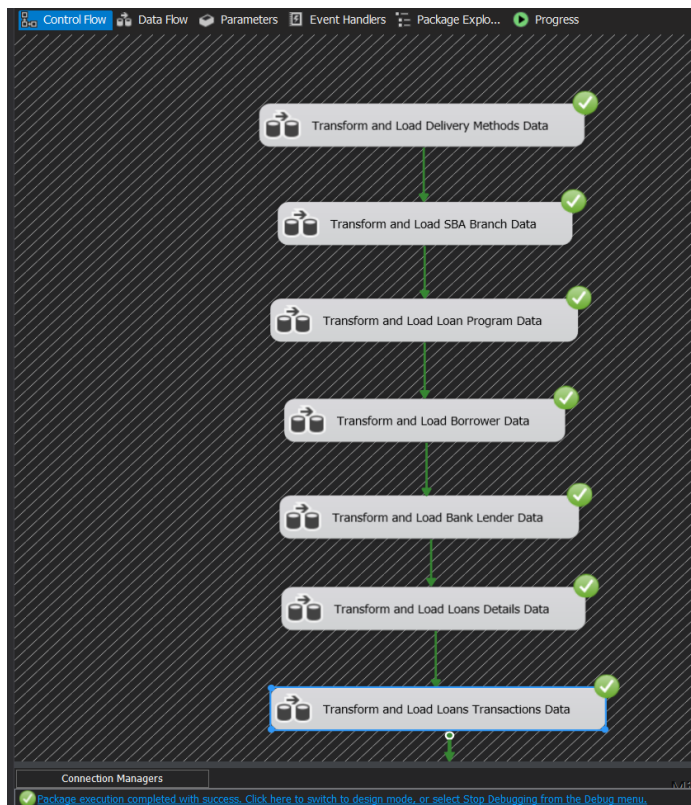
✓ Step 03: Data Profiling

Performed data profiling for all tables at once.

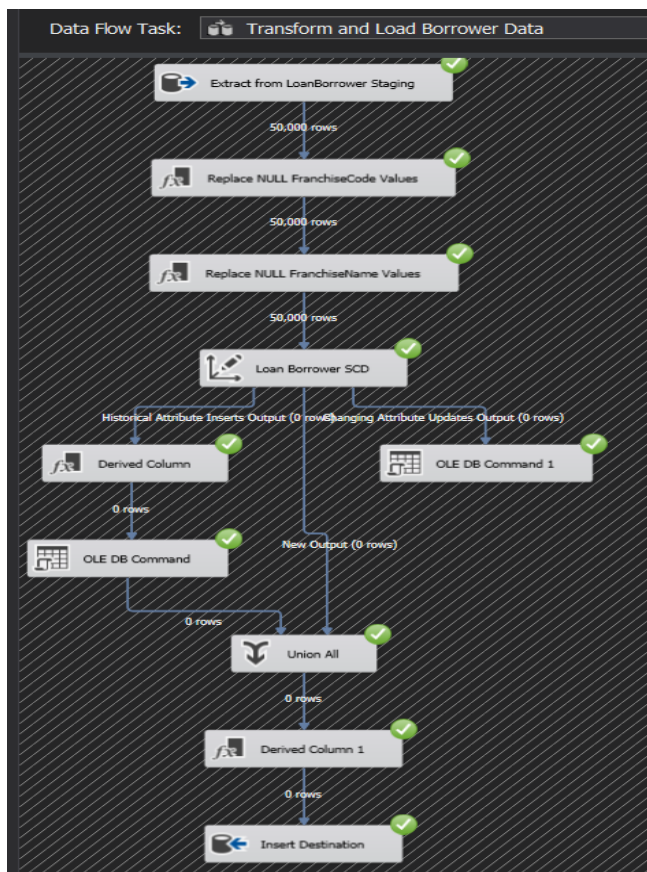


✓ Step 04: Transform and Load Data to Data Warehouse from Staging

Control flow



Data flow for Slowly Changing Dimension



Data flow for Fact Table



Step 6: ETL Development – Accumulating Fact Tables

✓ Step 01: Extending Fact Table with Additional Columns

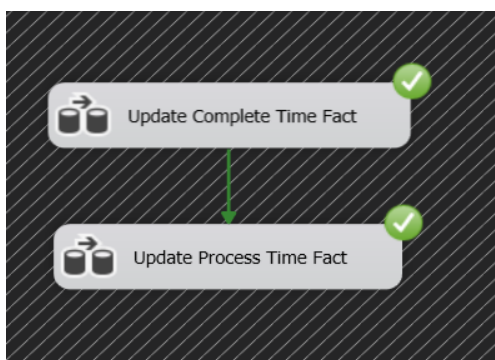
dbo.FactLoansTransactions	
Columns	
LoanTransID (int, null)	
BankGrossApproval (money, null)	
SBAGuaranteedApproval (money, null)	
FirstDisbursementDateKey (FK, int, null)	
InitialInterestRate (real, null)	
TermInMonths (int, null)	
PaidInFullDateKey (FK, int, null)	
LoanBorrKey (FK, int, null)	
BankLenderKey (FK, int, null)	
LoanDetailKey (FK, int, null)	
LoanProgKey (FK, int, null)	
MonthlyPayment (Computed, float, null)	
TotalPayment (Computed, float, null)	
TotalInterest (Computed, float, null)	
RiskAmount (Computed, money, 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)	

✓ Step 02: Prepare separate data set for complete time

BankLender	5/6/2022 11:31 PM	Text Document	3,444 KB
CompleteTime	5/12/2022 8:35 PM	Microsoft Excel Co...	1,064 KB
DeliveryMethods	5/6/2022 11:32 PM	Text Document	1 KB
LoanBorrower	5/6/2022 11:30 PM	Text Document	4,284 KB
LoanProgram	5/7/2022 10:24 AM	Microsoft Excel Co...	2,753 KB
LoansDetails	5/7/2022 12:57 PM	Microsoft Excel Co...	3,951 KB
LoansTransactions	5/12/2022 12:17 PM	Microsoft Excel Co...	3,602 KB
SBABranch	5/6/2022 11:31 PM	Text Document	19 KB

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

Control flow



Data Flows

