Assignment 1

Submitted to

Sri Lanka Institute of Information Technology

Submitted on

17-May-2022

IT20187828

Y3S1-WD.5.2

Isurika M.D.A

Step 1 – Data Set Selection

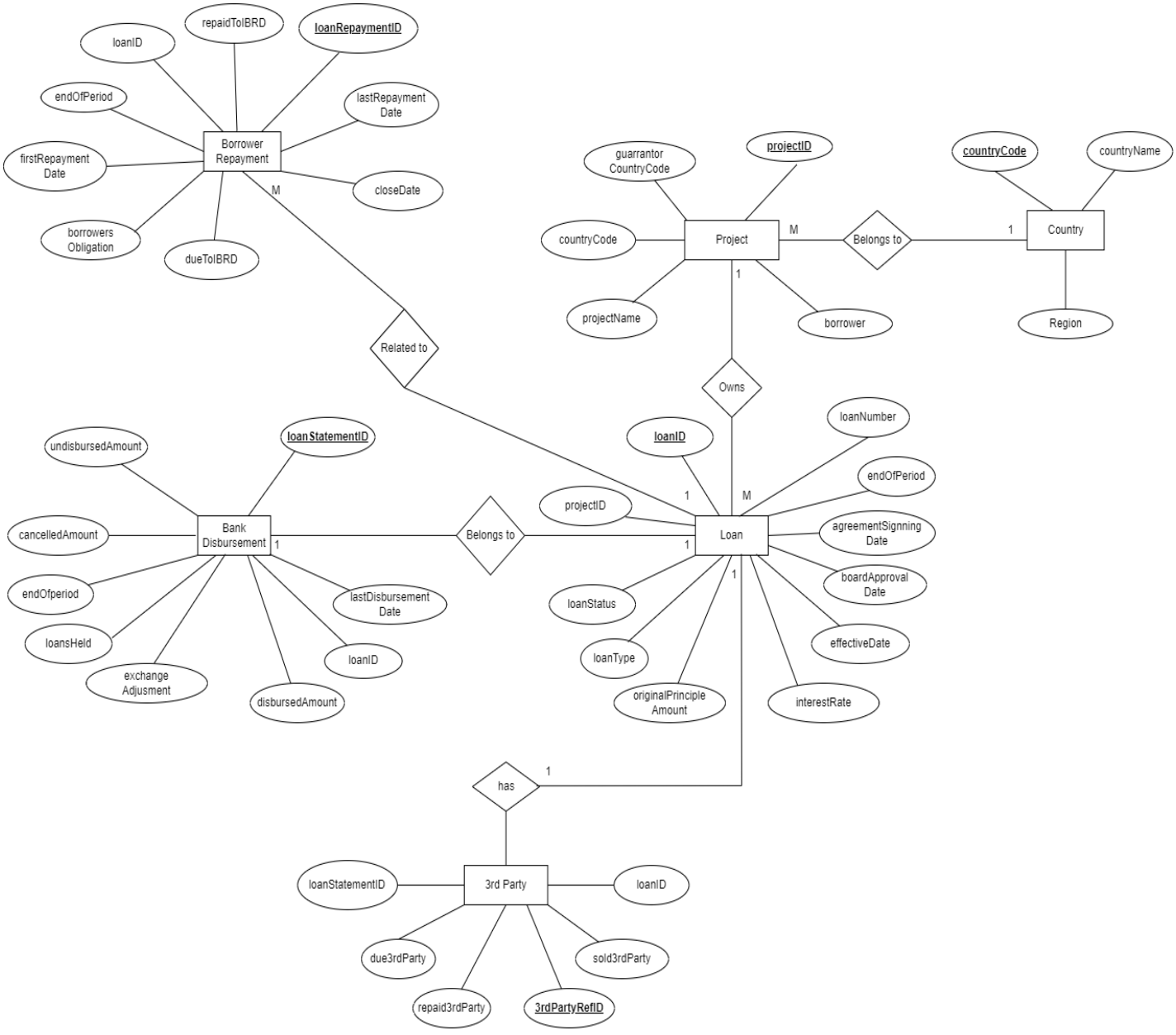
I have chosen a dataset of IBRD bank. This dataset hosted by the World Bank.

The International Bank for Reconstruction and Development (IBRD) is a global development cooperative owned by 189 member countries. As the largest development bank in the world, it supports the World Bank Group's mission by providing loans.

This dataset includes bank loan details which is given to countries to continue their large developing projects. It includes all the loan bank statement details which issues on the last day of every month.

Data Set Link: https://www.kaggle.com/datasets/theworldbank/ibrd-statement-of-loans-data

ER diagram Of Data Sources.

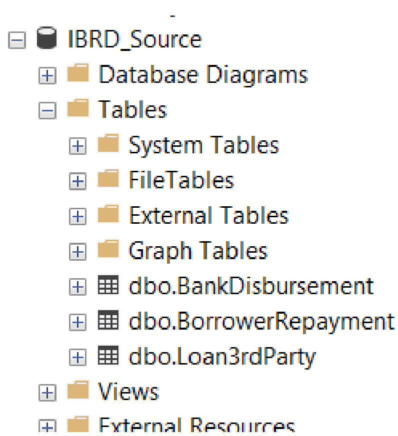


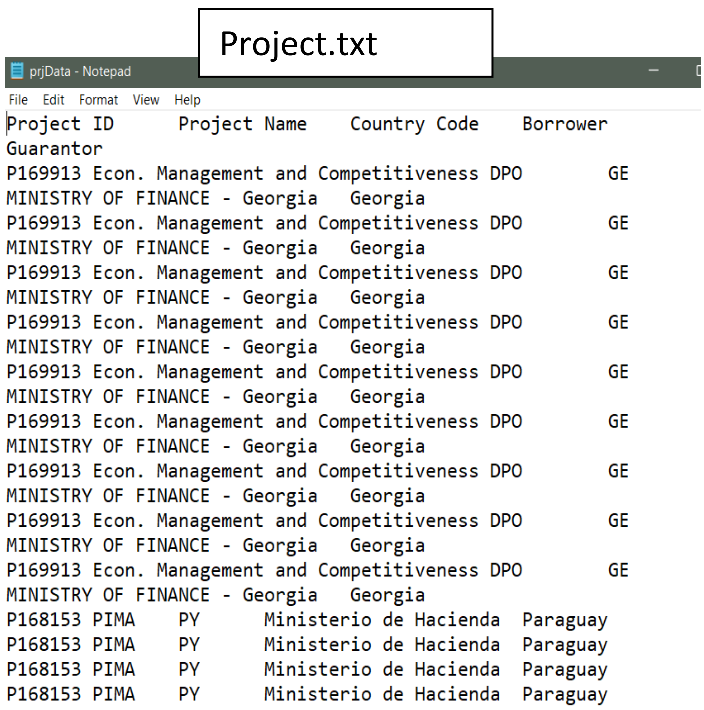
Step 2 – Preparation of data sources

All the data consists in a one large CSV file along with over 900,000 rows. So, I partitioned main large CSV file into seven small sub files. As 3rdParty.CSV, BorrowerRepayment.CSV, Country.CSV, BankDisbursement.CSV, LoanDisbursement.CSV, Loans.CSV and Projects.CSV.

Then depend on the guidelines I need to reconvert my CSV files into another two different formats as text and database. Here I used some ETL processes to load data into targeted formats.

There I convert BorrowerRepayment.CSV, BankDisbursement.CSV and 3rdParty.CSV files into database format. Country.CSV and Projects.CSV into text format. And Loans.CSV keep as it is. So, I can take these as my source files for staging database.





Country.txt

CountryData - Notepad

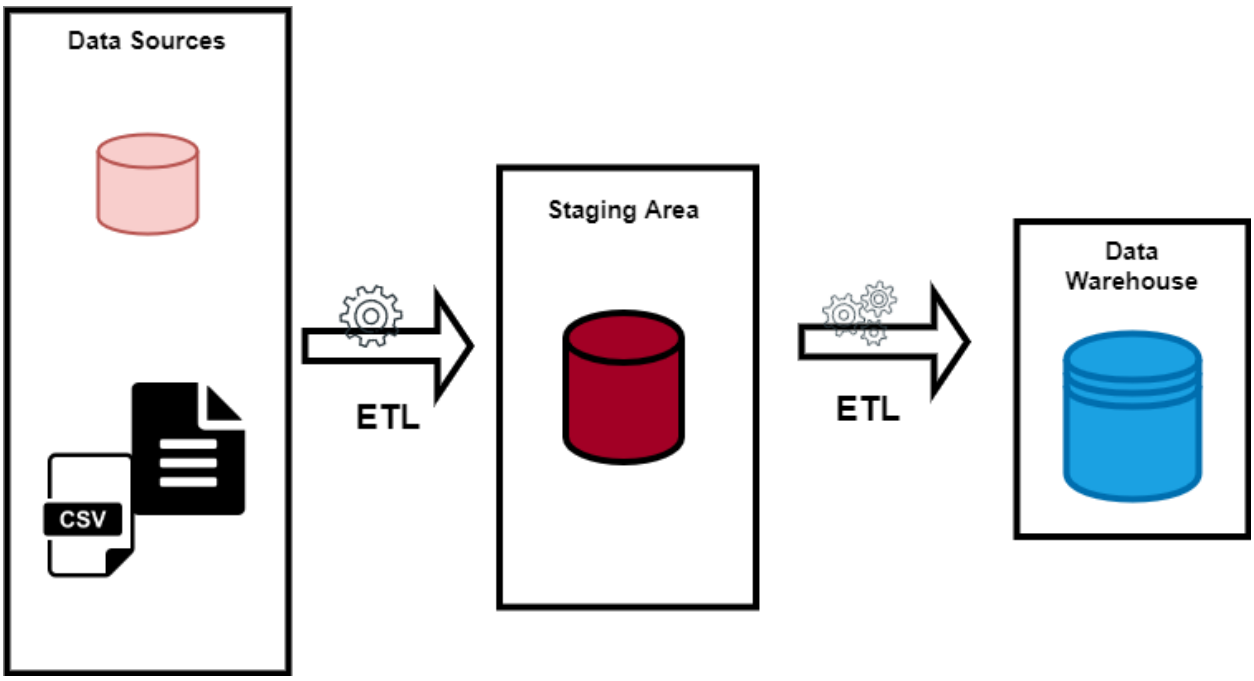
File Edit Format View Help

```
CountryCode     CountryName     Region
GE      Georgia EUROPE AND CENTRAL ASIA
GE      Georgia EUROPE AND CENTRAL ASIA
GE      Georgia EUROPE AND CENTRAL ASIA
GE      Georgia EUROPE AND CENTRAL ASIA
GE      Georgia EUROPE AND CENTRAL ASIA
GE      Georgia EUROPE AND CENTRAL ASIA
GE      Georgia EUROPE AND CENTRAL ASIA
GE      Georgia EUROPE AND CENTRAL ASIA
GE      Georgia EUROPE AND CENTRAL ASIA
PY      Paraguay        LATIN AMERICA AND CARIBBEAN
PY      Paraguay        LATIN AMERICA AND CARIBBEAN
PY      Paraguay        LATIN AMERICA AND CARIBBEAN
PY      Paraguay        LATIN AMERICA AND CARIBBEAN
PY      Paraguay        LATIN AMERICA AND CARIBBEAN
PY      Paraguay        LATIN AMERICA AND CARIBBEAN
PY      Paraguay        LATIN AMERICA AND CARIBBEAN
PY      Paraguay        LATIN AMERICA AND CARIBBEAN
PY      Paraguay        LATIN AMERICA AND CARIBBEAN
SV      El Salvador     LATIN AMERICA AND CARIBBEAN
```

Loan.CSV

	A	B	C	D	E	F	G	H	I	J
1	Loan Num	Project ID	Loan Type	Loan Status	Original Principal Amount	Interest Rate	End of Period	Agreement Signing Date	Board Approval Date	Effective Date (Most Recent)
2	IBRD9063(	P169913	FSL	Approved	49600000	0	2020-03-31T00:00:00.000		2020-03-26T00:00:00.000	
3	IBRD9063(	P169913	FSL	Disbursing	49600000	0	2020-04-30T00:00:00.000	2020-03-30T00:00:00.000	2020-03-26T00:00:00.000	2020-04-30T00:00:00.000
4	IBRD9063(	P169913	FSL	Disbursing	49600000	0	2020-05-31T00:00:00.000	2020-03-30T00:00:00.000	2020-03-26T00:00:00.000	2020-04-30T00:00:00.000
5	IBRD9063(	P169913	FSL	Disbursing	49600000	0	2020-06-30T00:00:00.000	2020-03-30T00:00:00.000	2020-03-26T00:00:00.000	2020-04-30T00:00:00.000
6	IBRD9063(	P169913	FSL	Disbursing	49600000	0	2020-07-31T00:00:00.000	2020-03-30T00:00:00.000	2020-03-26T00:00:00.000	2020-04-30T00:00:00.000
7	IBRD9063(	P169913	FSL	Disbursing	49600000	0	2020-08-31T00:00:00.000	2020-03-30T00:00:00.000	2020-03-26T00:00:00.000	2020-04-30T00:00:00.000
8	IBRD9063(	P169913	FSL	Disbursing	49600000	0	2020-09-30T00:00:00.000	2020-03-30T00:00:00.000	2020-03-26T00:00:00.000	2020-04-30T00:00:00.000
9	IBRD9063(	P169913	FSL	Disbursing	49600000	0	2020-10-31T00:00:00.000	2020-03-30T00:00:00.000	2020-03-26T00:00:00.000	2020-04-30T00:00:00.000
10	IBRD9063(	P169913	FSL	Disbursing	49600000	0	2020-11-30T00:00:00.000	2020-03-30T00:00:00.000	2020-03-26T00:00:00.000	2020-04-30T00:00:00.000
11	IBRD9064(	P168153	FSL	Approved	100000000	0	2020-03-31T00:00:00.000		2020-03-19T00:00:00.000	
12	IBRD9064(	P168153	FSL	Signed	100000000	0	2020-04-30T00:00:00.000	2020-04-14T00:00:00.000	2020-03-19T00:00:00.000	2020-10-11T00:00:00.000
13	IBRD9064(	P168153	FSL	Signed	100000000	0	2020-05-31T00:00:00.000	2020-04-14T00:00:00.000	2020-03-19T00:00:00.000	2020-10-11T00:00:00.000
14	IBRD9064(	P168153	FSL	Signed	100000000	0	2020-06-30T00:00:00.000	2020-04-14T00:00:00.000	2020-03-19T00:00:00.000	2020-10-11T00:00:00.000
15	IBRD9064(	P168153	FSL	Disbursing	100000000	0	2020-07-31T00:00:00.000	2020-04-14T00:00:00.000	2020-03-19T00:00:00.000	2020-07-06T00:00:00.000
16	IBRD9064(	P168153	FSL	Disbursing	100000000	0	2020-08-31T00:00:00.000	2020-04-14T00:00:00.000	2020-03-19T00:00:00.000	2020-07-06T00:00:00.000
17	IBRD9064(	P168153	FSL	Disbursing	100000000	0	2020-09-30T00:00:00.000	2020-04-14T00:00:00.000	2020-03-19T00:00:00.000	2020-07-06T00:00:00.000
18	IBRD9064(	P168153	FSL	Disbursing	100000000	0	2020-10-31T00:00:00.000	2020-04-14T00:00:00.000	2020-03-19T00:00:00.000	2020-07-06T00:00:00.000
19	IBRD9064(	P168153	FSL	Disbursing	100000000	0	2020-11-30T00:00:00.000	2020-04-14T00:00:00.000	2020-03-19T00:00:00.000	2020-07-06T00:00:00.000
20	IBRD9065(	P169677	FSL	Approved	250000000	0	2020-03-31T00:00:00.000		2020-03-19T00:00:00.000	
21	IBRD9065(	P169677	FSL	Approved	250000000	0	2020-04-30T00:00:00.000		2020-03-19T00:00:00.000	
22	IBRD9065(	P169677	FSL	Approved	250000000	0	2020-05-31T00:00:00.000		2020-03-19T00:00:00.000	
23	IBRD9065(	P169677	FSL	Approved	250000000	0	2020-06-30T00:00:00.000		2020-03-19T00:00:00.000	
24	IBRD9065(	P169677	FSL	Approved	250000000	0	2020-07-31T00:00:00.000		2020-03-19T00:00:00.000	
25	IBRD9065(	P169677	FSL	Approved	250000000	0	2020-08-31T00:00:00.000		2020-03-19T00:00:00.000	
26	IBRD9065(	P169677	FSL	Approved	250000000	0	2020-09-30T00:00:00.000		2020-03-19T00:00:00.000	
27	IBRD9065(	P169677	FSL	Approved	250000000	0	2020-10-31T00:00:00.000		2020-03-19T00:00:00.000	

Step 3 – Solution Architecture



Data Sources: As data sources in this BI solution, I used database files and flat files sources as txt and CSV.

Staging Area: inside the staging area I stored all the source files as database files. When upload to the staging area from source database I used some transformations as sorting data by remove duplicate values, remove null values etc. therefor I have used some ETL processes when extracting from Source to Staging.

Data Warehouse: When upload to the data warehouse from staging database I used to manage historical attributes of slowly changing dimensions. And keep track of changing attributes and fixed attributes. Here also I have used some ETL processes when extracting from Staging to warehouse.

Step 4 – Data warehouse design & development

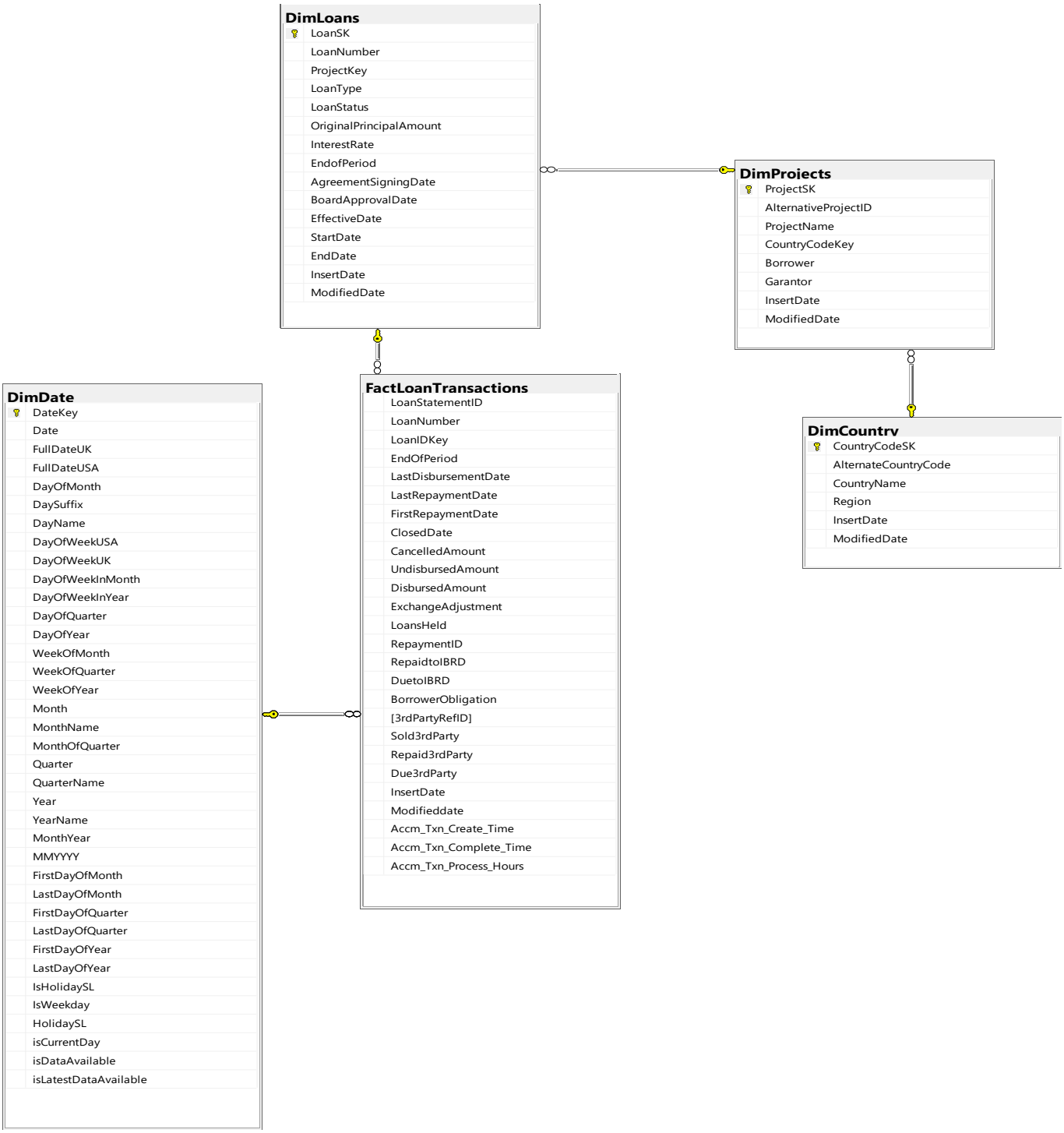
I have designed the data warehouse schema using **Snowflake Schema.**

When loading data to fact table I have combined Bank Disbursement Staging, Borrower Repayment Staging and 3rdParty Staging into one fact Table. Therefore, data warehouse includes four dimension tables including date dimension and one fact table.

Fact table have relationship with Loan dimension and Date dimension. Then Loan dimension is connected to project dimension. And Project dimension is connected to Country dimension.

Assumptions

I have taken Dim Loan Data as slowly changing dimension, because **loan status** and **interest rate** can be changed time to time, and we need to keep track of their historical data.


DimLoans
LoanSK
LoanNumber
ProjectKey
LoanType
LoanStatus
OriginalPrincipalAmount
InterestRate
EndofPeriod
AgreementSigningDate
BoardApprovalDate
EffectiveDate
StartDate
EndDate
InsertDate
ModifiedDate
DimProjects
ProjectSK
AlternativeProjectID
ProjectName
CountryCodeKey
Borrower
Garantor
InsertDate
ModifiedDate
DimDate
DateKey
Date
FullDateUK
FullDateUSA
DayOfMonth
DaySuffix
DayName
DayOfWeekUSA
DayOfWeekUK
DayOfWeekInMonth
DayOfWeekInYear
DayOfQuarter
DayOfYear
WeekOfMonth
WeekOfQuarter
WeekOfYear
Month
MonthName
MonthOfQuarter
Quarter
QuarterName
Year
YearName
MonthYear
MMYYYY
FirstDayOfMonth
LastDayOfMonth
FirstDayOfQuarter
LastDayOfQuarter
FirstDayOfYear
LastDayOfYear
IsHolidaySL
IsWeekday
HolidaySL
isCurrentDay
isDataAvailable
isLatestDataAvailable
FactLoanTransactions
LoanStatementID
LoanNumber
LoanIDKey
EndOfPeriod
LastDisbursementDate
LastRepaymentDate
FirstRepaymentDate
ClosedDate
CancelledAmount
UndisbursedAmount
DisbursedAmount
ExchangeAdjustment
LoansHeld
RepaymentID
RepaidtoIBRD
DuetoIBRD
BorrowerObligation
[3rdPartyRefID]
Sold3rdParty
Repaid3rdParty
Due3rdParty
InsertDate
Modifieddate
Accm_Txn_Create_Time
Accm_Txn_Complete_Time
Accm_Txn_Process_Hours
DimCountry
CountryCodeSK
AlternateCountryCode
CountryName
Region
InsertDate
ModifiedDate

Project Dimension

DESKTOP-4JL2VCE.I...- dbo.DimProjects

Column Name	Data Type	Allow Nulls
ProjectSK	int	☐
AlternativeProjectID	varchar(50)	☑
ProjectName	varchar(50)	☑
CountryCodeKey	int	☑
Borrower	varchar(50)	☑
Garantor	varchar(50)	☑
InsertDate	datetime	☑
ModifiedDate	datetime	☑
		☐

Loan Dimension

DESKTOP-4JL2VCE....W - dbo.DimLoans

Column Name	Data Type	Allow Nulls
LoanSK	int	☐
LoanNumber	varchar(50)	☑
ProjectKey	int	☑
LoanType	varchar(50)	☑
LoanStatus	varchar(50)	☑
OriginalPrincipalAmount	varchar(50)	☑
InterestRate	varchar(50)	☑
EndofPeriod	varchar(50)	☑
AgreementSigningDate	varchar(50)	☑
BoardApprovalDate	varchar(50)	☑
EffectiveDate	varchar(50)	☑
StartDate	datetime	☑
EndDate	datetime	☑
InsertDate	datetime	☑
ModifiedDate	datetime	☑
		☐

Country dimension

DESKTOP-4JL2VCE.I...- dbo.DimCountry

Column Name	Data Type	Allow Nulls
CountryCodeSK	int	☐
AlternateCountryCode	varchar(50)	☑
CountryName	varchar(50)	☑
Region	nvarchar(50)	☑
InsertDate	datetime	☑
ModifiedDate	datetime	☑
		☐

Date dimension

DESKTOP-4JL2VCE....DW - dbo.DimDate

Column Name	Data Type	Allow Nulls
DateKey	int	☐
Date	datetime	☑
FullDateUK	char(10)	☑
FullDateUSA	char(10)	☑
DayOfMonth	varchar(2)	☑
DaySuffix	varchar(4)	☑
DayName	varchar(9)	☑
DayOfWeekUSA	char(1)	☑
DayOfWeekUK	char(1)	☑
DayOfWeekInMonth	varchar(2)	☑
DayOfWeekInYear	varchar(2)	☑
DayOfQuarter	varchar(3)	☑
DayOfYear	varchar(3)	☑
WeekOfMonth	varchar(1)	☑
WeekOfQuarter	varchar(2)	☑
WeekOfYear	varchar(2)	☑
Month	varchar(2)	☑
MonthName	varchar(9)	☑
MonthOfQuarter	varchar(2)	☑
Quarter	char(1)	☑
QuarterName	varchar(9)	☑
Year	char(4)	☑
YearName	char(7)	☑
MonthYear	char(10)	☑
MMYYYY	char(6)	☑
FirstDayOfMonth	date	☑

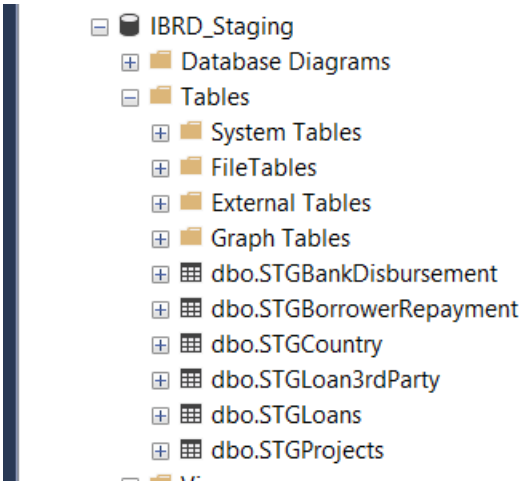
Fact Loan Transactions

DESKTOP-4JL2VCE.I...ctLoanTransactions

Column Name	Data Type	Allow Nulls
LoanStatementID	int	☑
LoanNumber	varchar(50)	☑
LoanIDKey	int	☑
EndOfPeriod	int	☑
LastDisbursementDate	int	☑
LastRepaymentDate	int	☑
FirstRepaymentDate	int	☑
ClosedDate	int	☑
CancelledAmount	money	☑
UndisbursedAmount	money	☑
DisbursedAmount	money	☑
ExchangeAdjustment	float	☑
LoansHeld	money	☑
RepaymentID	int	☑
RepaidtoIBRD	money	☑
DuetoIBRD	money	☑
BorrowerObligation	money	☑
[3rdPartyRefID]	int	☑
Sold3rdParty	money	☑
Repaid3rdParty	money	☑
Due3rdParty	money	☑
InsertDate	datetime	☑
Modifieddate	datetime	☑
Accm_Txn_Create_Time	datetime	☑
Accm_Txn_Complete_Time	datetime	☑
Accm_Txn_Process_Hours	int	☑
		☐

Step 5 - ETL Development

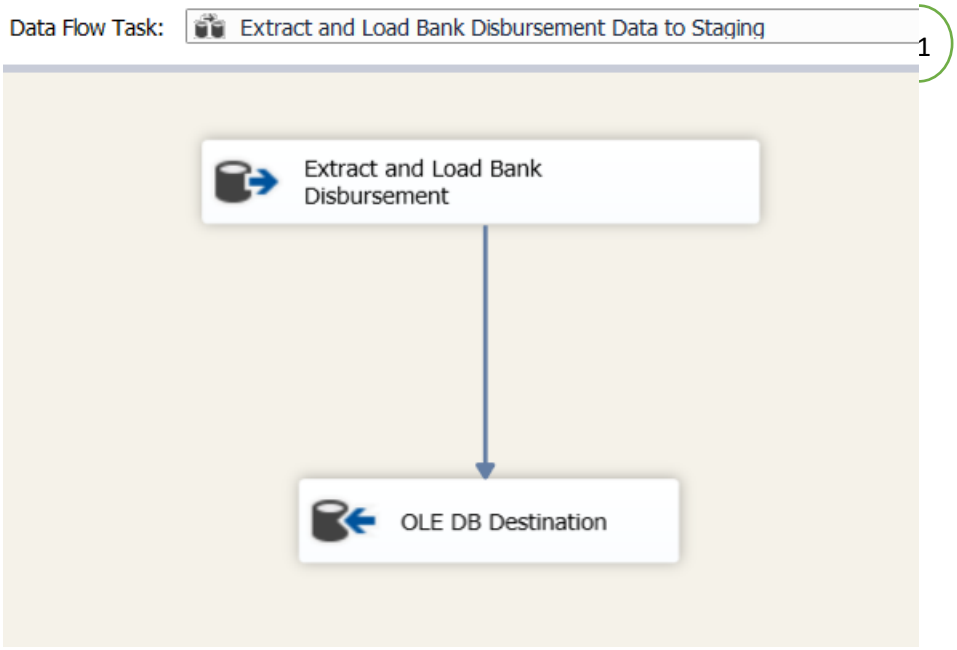
From Data Sources to Staging

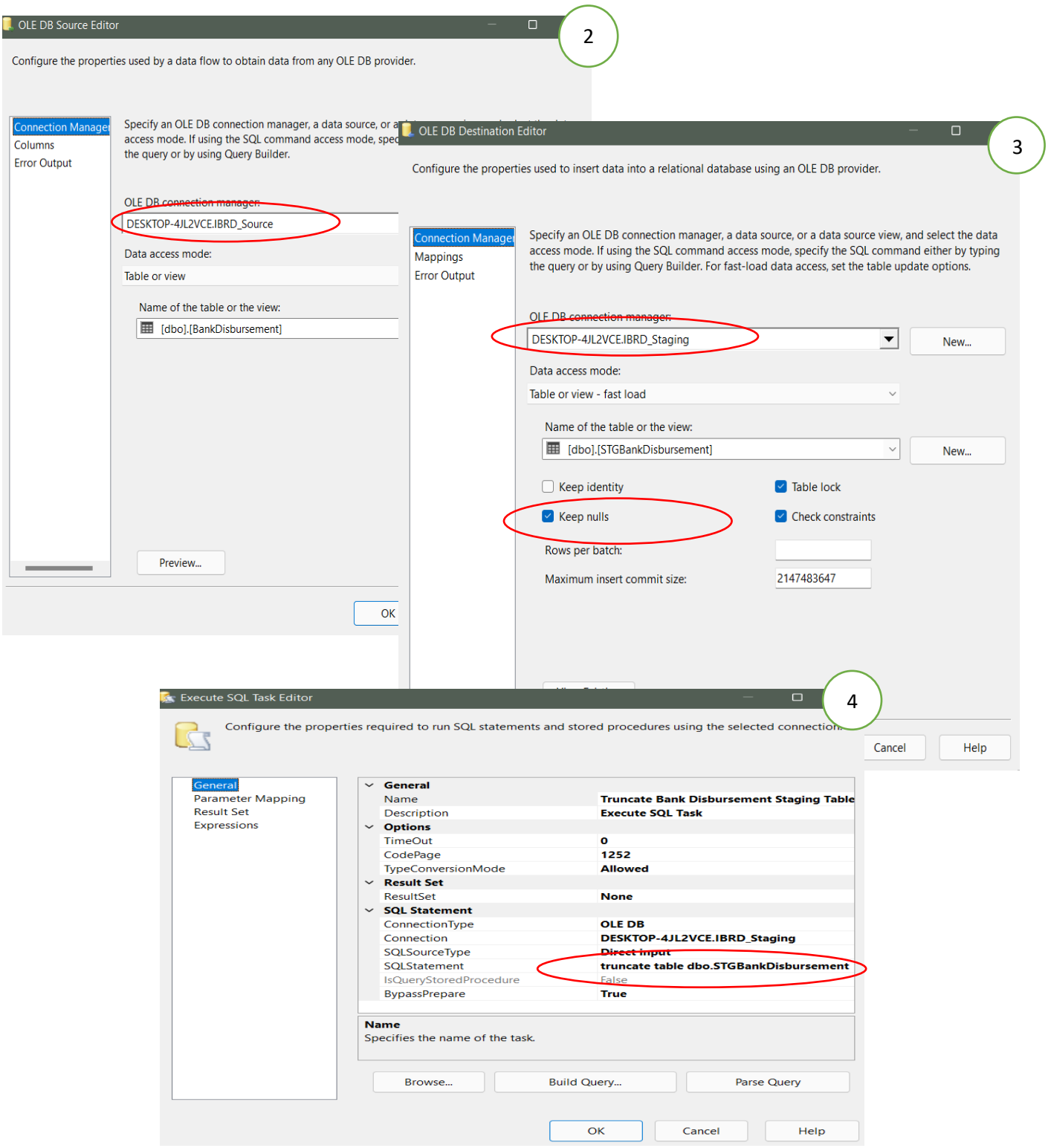


Load Data to Staging from IBRD Source database

** As mentioned in step 2 I have three tables in IBRD Source database. Therefore, I used same procedure to load all the three tables to Staging.

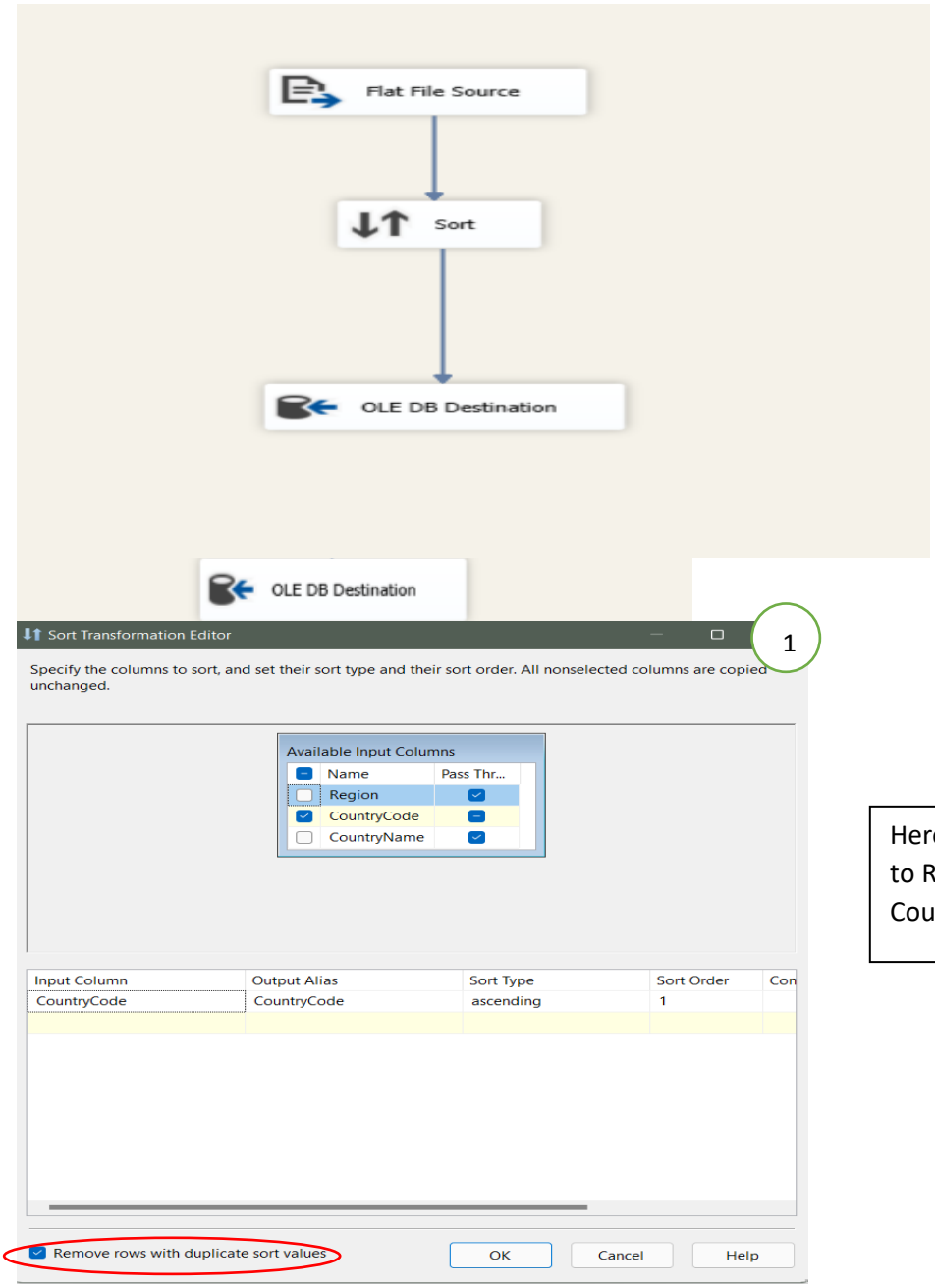
** here I only demonstrating Extract and load procedure of Bank Disbursement Table



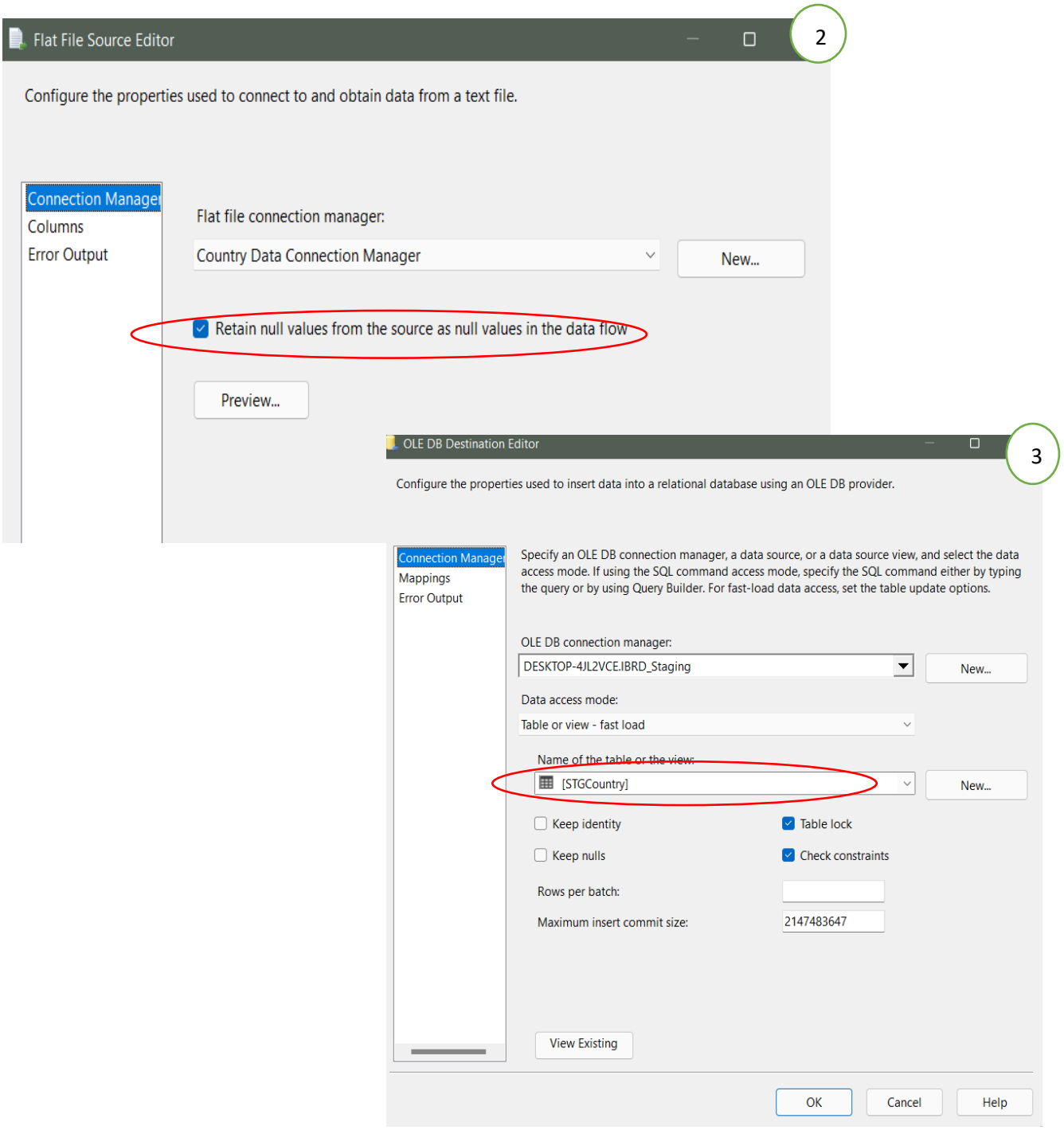

2
OLE DB Source Editor
Configure the properties used by a data flow to obtain data from any OLE DB provider.
Connection Manager
Columns
Error Output
Specify an OLE DB connection manager, a data source, or a
access mode. If using the SQL command access mode, spec
the query or by using Query Builder.
OLE DB connection manager:
DESKTOP-4JL2VCE.IBRD_Source
Data access mode:
Table or view
Name of the table or the view:
[dbo].[BankDisbursement]
Preview...
OK
3
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
Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command either by typing the query or by using Query Builder. For fast-load data access, set the table update options.
OLE DB connection manager:
DESKTOP-4JL2VCE.IBRD_Staging
New...
Data access mode:
Table or view - fast load
Name of the table or the view:
[dbo].[STGBankDisbursement]
New...
Keep identity
Table lock
Keep nulls
Check constraints
Rows per batch:
Maximum insert commit size:
2147483647
4
Execute SQL Task Editor
Configure the properties required to run SQL statements and stored procedures using the selected connection.
Cancel
Help
General
Parameter Mapping
Result Set
Expressions
General
Name
Truncate Bank Disbursement Staging Table
Description
Execute SQL Task
Options
TimeOut
0
CodePage
1252
TypeConversionMode
Allowed
Result Set
ResultSet
None
SQL Statement
ConnectionType
OLE DB
Connection
DESKTOP-4JL2VCE.IBRD_Staging
SQLSourceType
Direct input
SQLStatement
truncate table dbo.STGBankDisbursement
IsQueryStoredProcedure
False
BypassPrepare
True
Name
Specifies the name of the task.
Browse...
Build Query...
Parse Query
OK
Cancel
Help

** As mentioned in step 2 I have two test files as Country.txt and Project.txt and one Loans.CSV file. Therefore, I used same procedure to load all the three files to Staging by using FLAT FILE SOURCE COMPONENT.

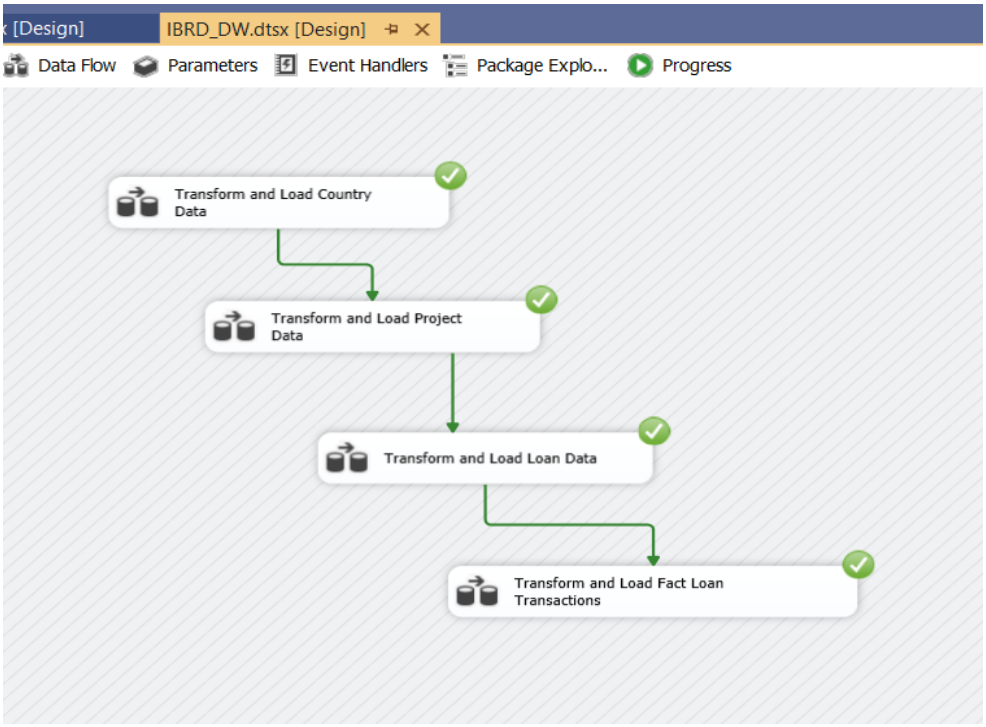
** here I only demonstrating Extract and load procedure of Country Staging Table

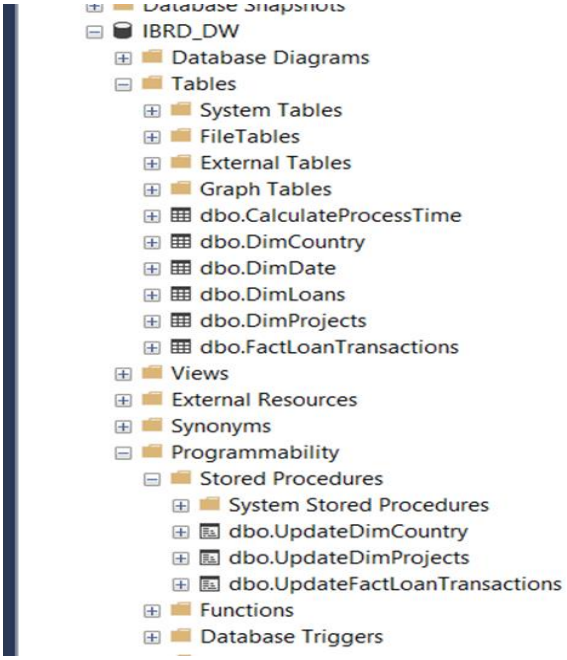


Here I have used this Sort Component to Remove duplicate Values In Country Table

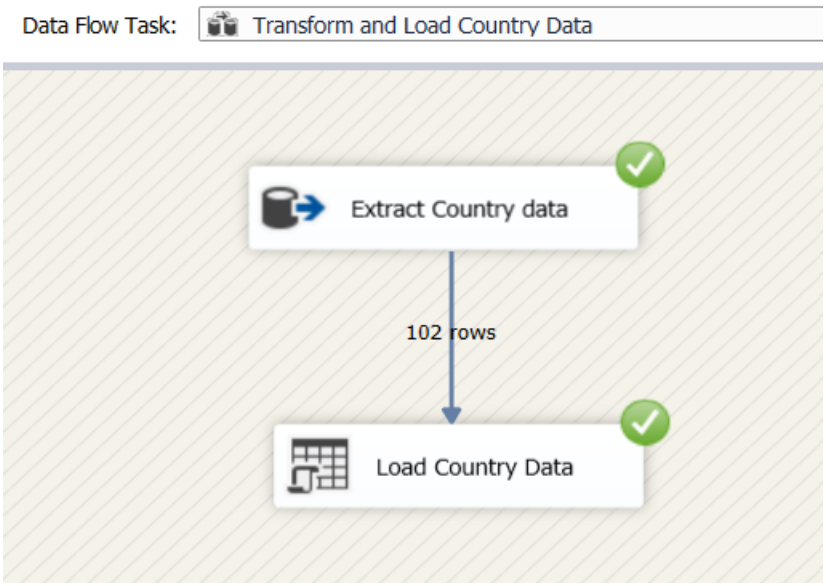

2
Flat File Source Editor
Configure the properties used to connect to and obtain data from a text file.
Connection Manager
Columns
Error Output
Flat file connection manager:
Country Data Connection Manager
New...
Retain null values from the source as null values in the data flow
Preview...
3
OLE DB Destination Editor
Configure the properties used to insert data into a relational database using an OLE DB provider.
Connection Manager
Mappings
Error Output
Specify an OLE DB connection manager, a data source, or a data source view, and select the data access mode. If using the SQL command access mode, specify the SQL command either by typing the query or by using Query Builder. For fast-load data access, set the table update options.
OLE DB connection manager:
DESKTOP-4JL2VCE.IBRD_Staging
New...
Data access mode:
Table or view - fast load
Name of the table or the view:
[STGCountry]
New...
Keep identity
Table lock
Keep nulls
Check constraints
Rows per batch:
Maximum insert commit size:
2147483647
View Existing
OK
Cancel
Help

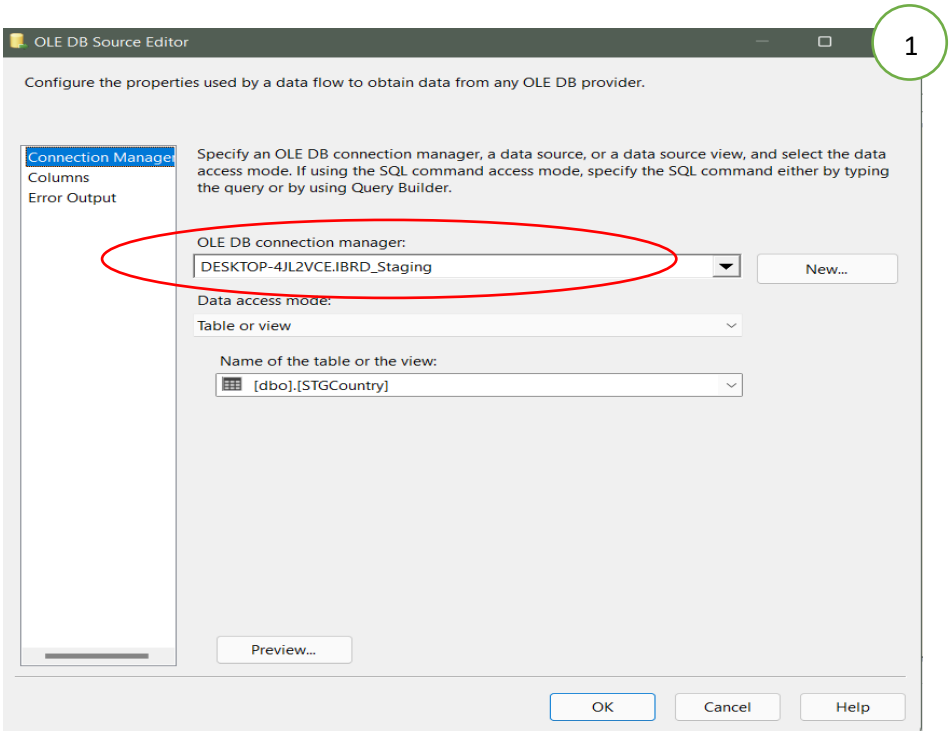
From Data Staging to Data Warehouse





1 step: Transform and Loading from Country Staging Table to Dim Country



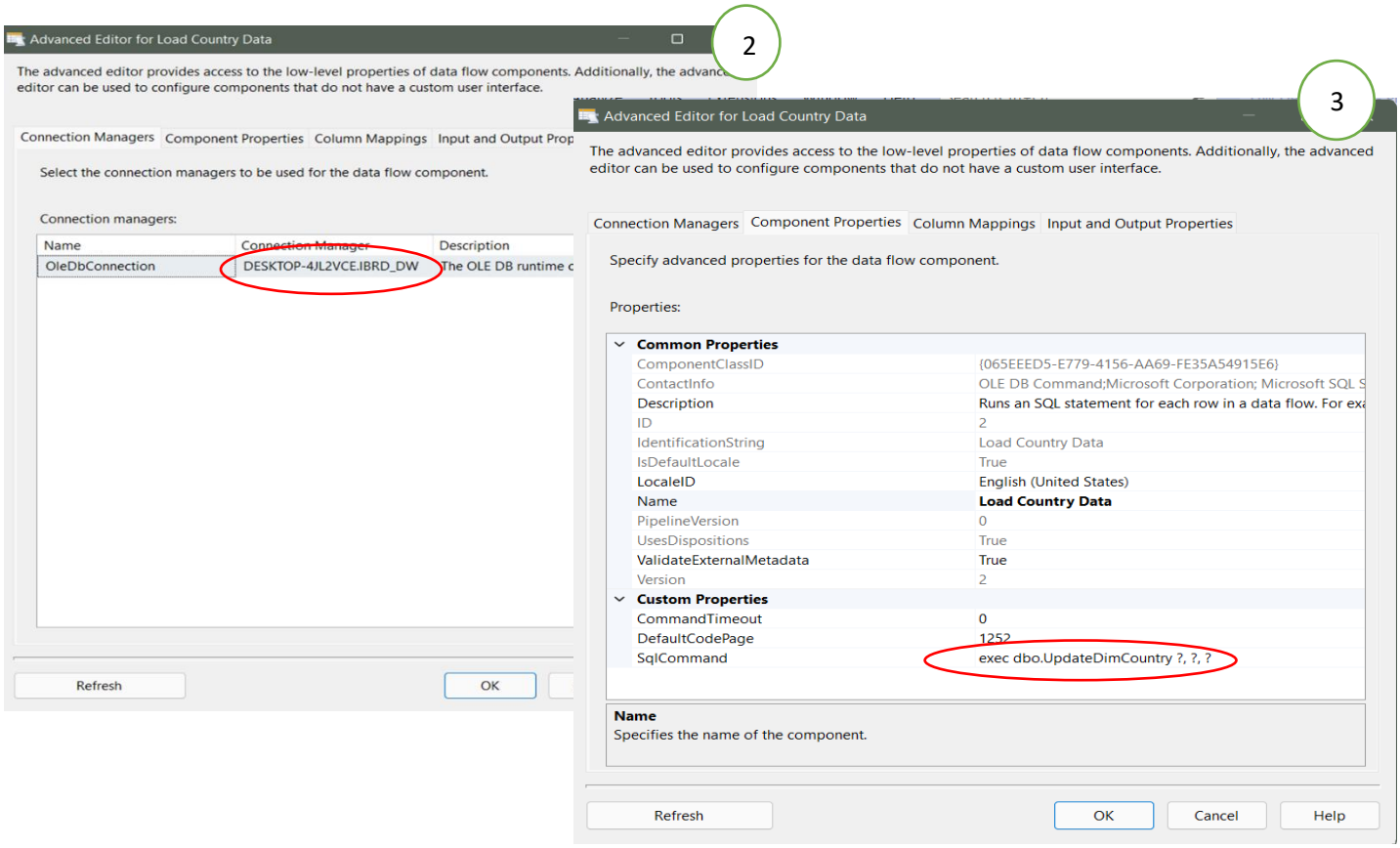


First step is to configure OLE DB Source Connection

Steps of Configuring OLE DB Command Object

When Load to DW I used Stored Procedure Because in Country Dimension I have not any historical data to manage. I only want update rows if already existing in the table or add a new record.

There for I used OLE DB COMMAND COMPONENT to load the data to DW.



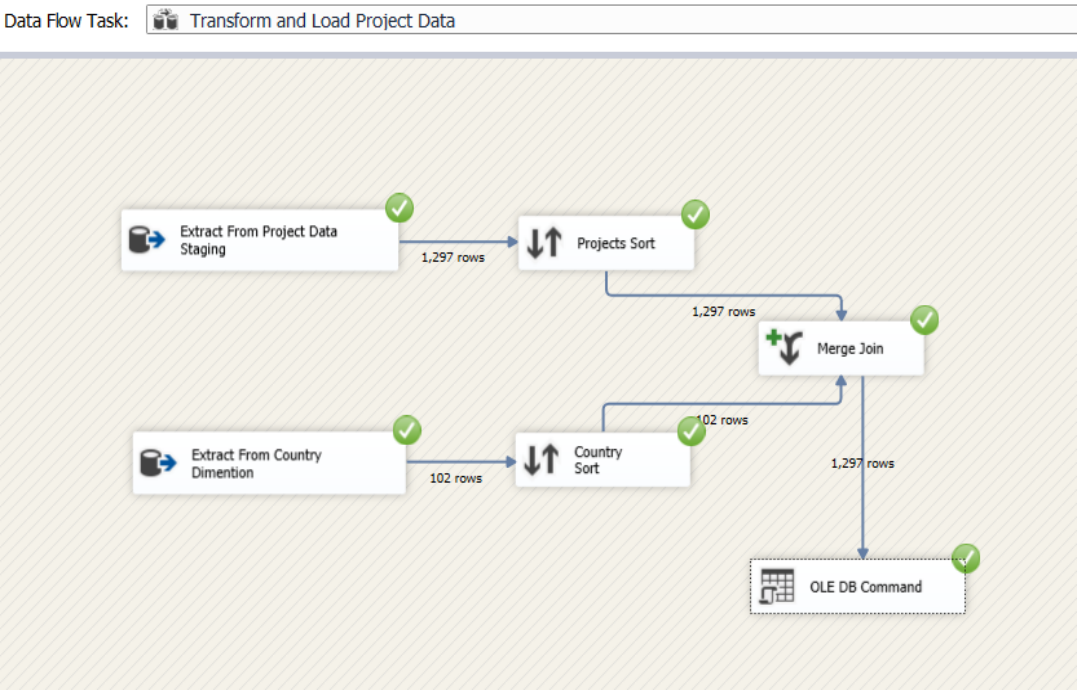
Stored Procedure to manage Dim Country table data

Query1.sql - DE...-4JL2VCE\acer (62))* DESKTOP-4JL2VCE.I...ctLoanTransactions

```
USE [IBRD_DW]
GO
/****** Object:  StoredProcedure [dbo].[UpdateDimCountry]    Script Date: 17-May-22 13:52:38 ******/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[UpdateDimCountry]
@CountryCode varchar(50),
@CountryName varchar(50),
@Region nvarchar(50)
AS
BEGIN
if not exists (select CountryCodeSK
from dbo.DimCountry
where AlternateCountryCode = @CountryCode)
BEGIN
insert into dbo.DimCountry
(AlternateCountryCode, CountryName, Region, InsertDate, ModifiedDate)
values
(@CountryCode, @CountryName, @Region, GETDATE(), GETDATE())
END;
if exists (select CountryCodeSK
from dbo.DimCountry
where AlternateCountryCode = @CountryCode)
BEGIN
update dbo.DimCountry
set CountryName = @CountryName,
Region = @Region,
ModifiedDate = GETDATE()
where AlternateCountryCode = @CountryCode
END;
END;
```

2 step: Transform and Loading from Project Staging Table to Dim Project

**** Same procedure used as above-mentioned step 1.**



Difference between 1st step and this 2nd step is here I have to use another source connection manager to connect with Country Dimension to get data to 'CountryCodeSK' column .

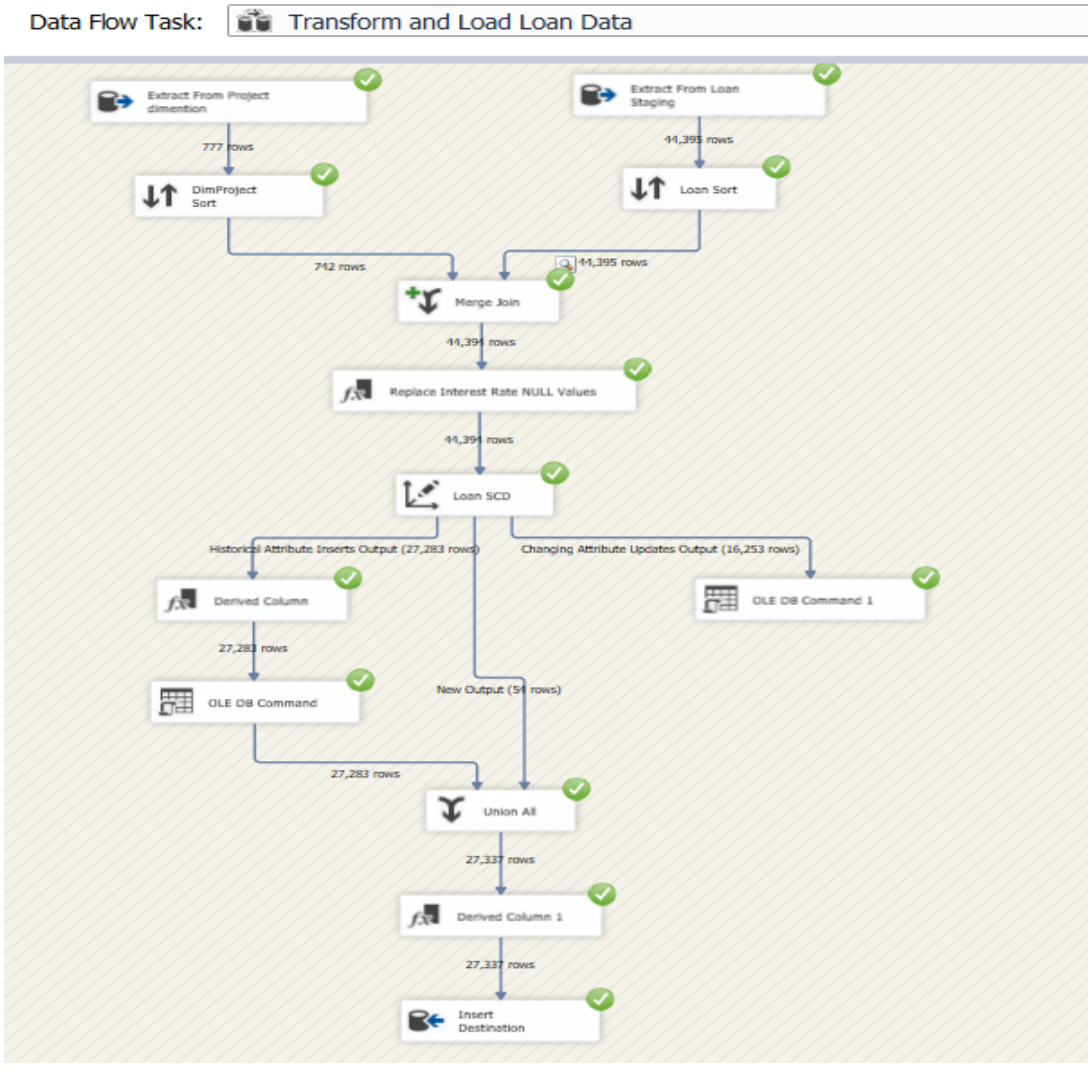
Stored Procedure to manage Dim Project table data

LQuery2.sql - DE...-4JL2VCE\acer (62)) DESKTOP-4JL2VCE.I...ctLoanTransactions

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[UpdateDimProjects]
@ProjectID varchar(50),
@ProjectName varchar(50),
@CountryCodeKey int,
@Borrower varchar(50),
@Garantor varchar(50)
AS
BEGIN
if not exists (select ProjectSK
from dbo.DimProjects
where AlternativeProjectID = @ProjectID)
BEGIN
insert into dbo.DimProjects
(AlternativeProjectID , ProjectName, CountryCodeKey, Borrower, Garantor,
InsertDate, ModifiedDate)
values
(@ProjectID, @ProjectName, @CountryCodeKey, @Borrower, @Garantor,
GETDATE(), GETDATE())
END;
if exists (select ProjectSK
from dbo.DimProjects
where AlternativeProjectID  = @ProjectID)
BEGIN
update dbo.DimProjects
set
ProjectName = @ProjectName,
CountryCodeKey = @CountryCodeKey,
Borrower = @Borrower,
Garantor = @Garantor,
ModifiedDate = GETDATE()
where AlternativeProjectID  = @ProjectID
END;
```

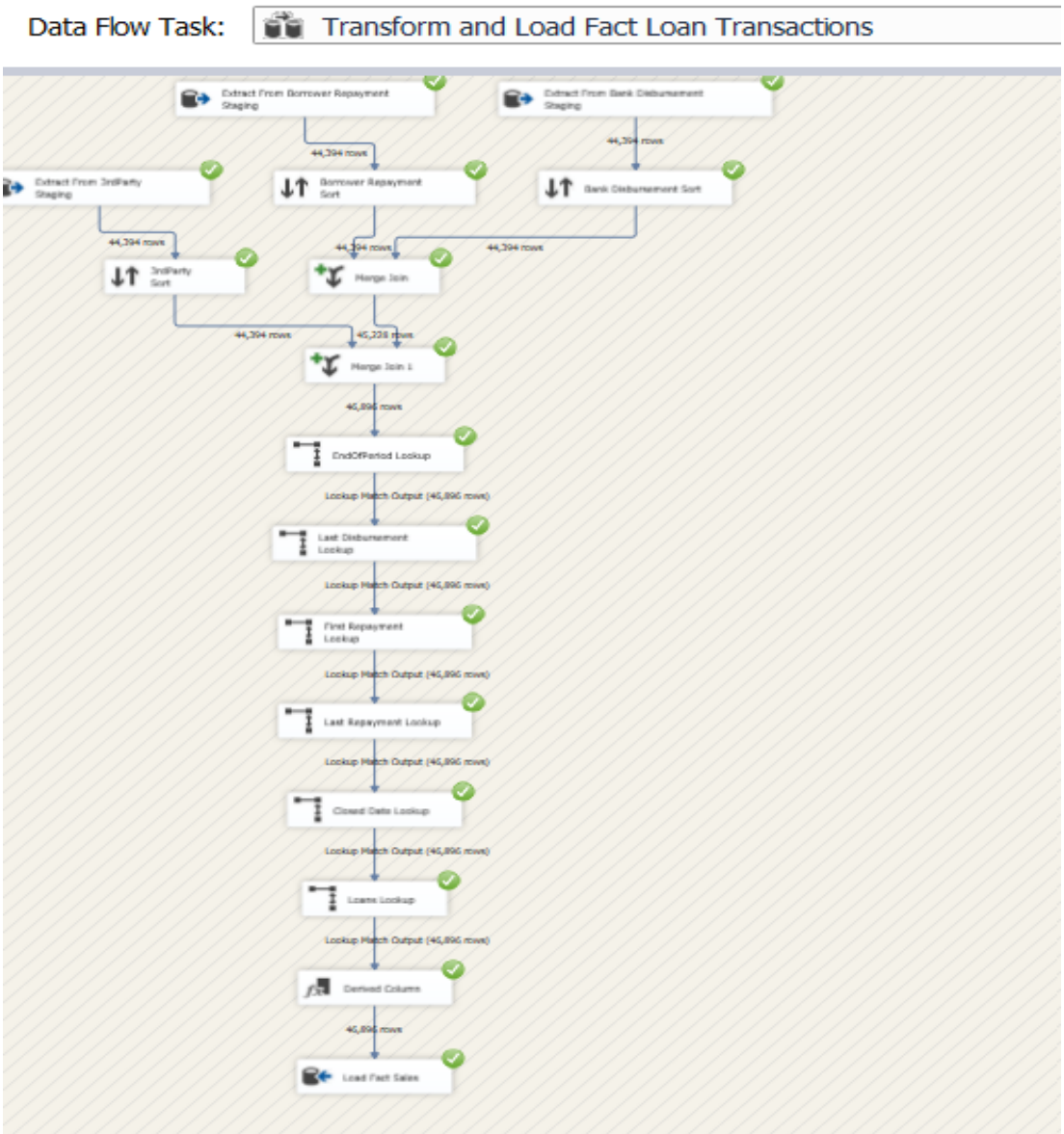
3 step: Transform and Loading from Loan Staging Table to Dim Loan

Here I have loaded the loan dimension as a slowly changing dimension. Loan data needs some history management. When existing data row needs update, we cannot update that data row as it but we have to add that as a new record while updating modify date and end date column of already existing record.

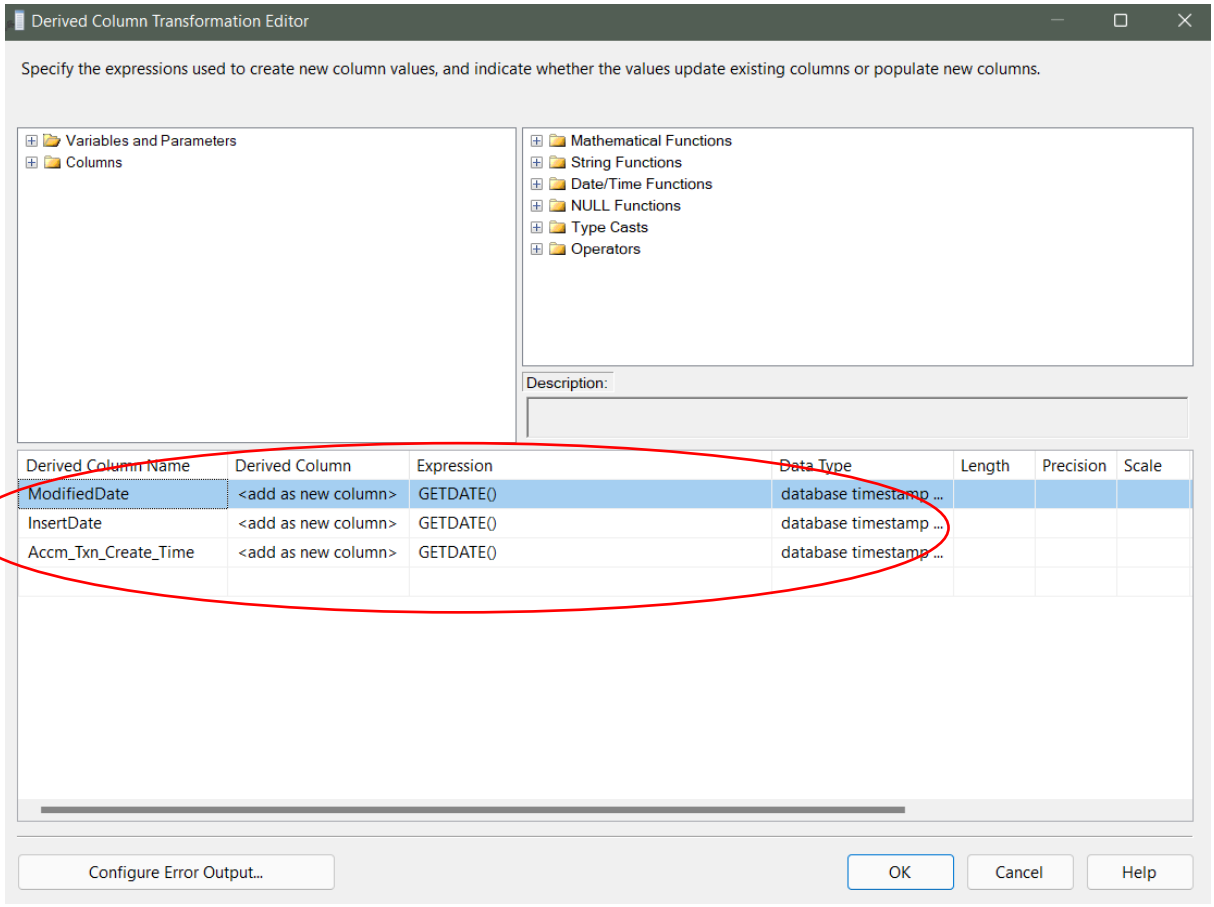


4 step: Transform and Loading Fact Loan Transaction Table

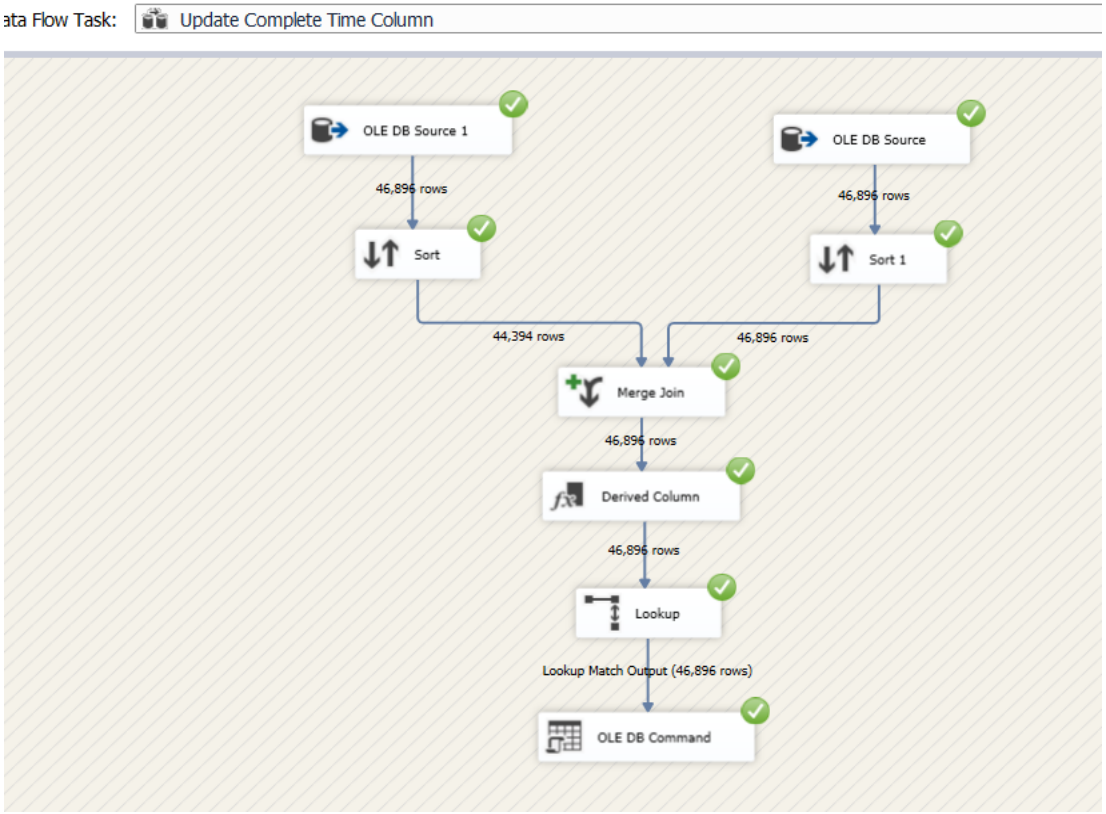
Here I have used three staging tables Borrower Repayment, Bank Disbursement and 3rdparty and then all are loaded to Fact Table. Since fact table have relationship with date dimension and loan dimension, I used LOOKUP COMPONENT to get surrogate keys of them.



I have used DERIVED COLUMN COMPONENT to add modified data column, insert date column. Additionally, Accm_Txn_Create_Time column which I will have to use in Step 6 of implementation.



Step 6 - ETL Development – Accumulating Fact Tables



	LoanStatementID	Derived Column.Accm_Txn_Complete_Time
1	24411	2022-05-13 19:20:24.790
2	24412	2022-05-13 19:20:24.790
3	24413	2022-05-13 19:20:24.790
4	24414	2022-05-13 19:20:24.790
5	24415	2022-05-13 19:20:24.790
6	24416	2022-05-13 19:20:24.790
7	24417	2022-05-13 19:20:24.790
8	24418	2022-05-13 19:20:24.790
9	24419	2022-05-13 19:20:24.790
10	24420	2022-05-13 19:20:24.790
11	24421	2022-05-13 19:20:24.790
12	24422	2022-05-13 19:20:24.790
13	24423	2022-05-13 19:20:24.790
14	24424	2022-05-13 19:20:24.790
15	24425	2022-05-13 19:20:24.790
16	24426	2022-05-13 19:20:24.790
17	24427	2022-05-13 19:20:24.790

This is the table I created to update Txn_Complete_Time.

I generate it by using a excel sheet and then import to DW.

	Modifieddate	Accm_Txn_Create_Time	Accm_Txn_Complete_Time	Accm_Txn_Process_Hours
:27.463	2022-05-17 16:56:27.463	2022-05-17 16:56:27.463	2022-05-13 19:20:24.790	-93
:27.463	2022-05-17 16:56:27.463	2022-05-17 16:56:27.463	2022-05-13 19:20:24.790	-93
:27.463	2022-05-17 16:56:27.463	2022-05-17 16:56:27.463	2022-05-13 19:20:24.790	-93
:27.463	2022-05-17 16:56:27.463	2022-05-17 16:56:27.463	2022-05-13 19:20:24.790	-93
:27.463	2022-05-17 16:56:27.463	2022-05-17 16:56:27.463	2022-05-13 19:20:24.790	-93
:27.463	2022-05-17 16:56:27.463	2022-05-17 16:56:27.463	2022-05-13 19:20:24.790	-93
:27.463	2022-05-17 16:56:27.463	2022-05-17 16:56:27.463	2022-05-13 19:20:24.790	-93
:27.463	2022-05-17 16:56:27.463	2022-05-17 16:56:27.463	2022-05-13 19:20:24.790	-93
:27.463	2022-05-17 16:56:27.463	2022-05-17 16:56:27.463	2022-05-13 19:20:24.790	-93
:27.463	2022-05-17 16:56:27.463	2022-05-17 16:56:27.463	2022-05-13 19:20:24.790	-93
:27.463	2022-05-17 16:56:27.463	2022-05-17 16:56:27.463	2022-05-13 19:20:24.790	-93
:27.463	2022-05-17 16:56:27.463	2022-05-17 16:56:27.463	2022-05-13 19:20:24.790	-93
:27.463	2022-05-17 16:56:27.463	2022-05-17 16:56:27.463	2022-05-13 19:20:24.790	-93
:27.463	2022-05-17 16:56:27.463	2022-05-17 16:56:27.463	2022-05-13 19:20:24.790	-93
:27.463	2022-05-17 16:56:27.463	2022-05-17 16:56:27.463	2022-05-13 19:20:24.790	-93

```
ALTER PROCEDURE [dbo].[UpdateFactLoanTransactions]
       @LoanStatementID int
      ,@LoanNumber varchar(50)
      ,@LoanIDKey int
      ,@EndOfPeriod int
      ,@LastDisbursementDate int
      ,@LastRepaymentDate int
      ,@FirstRepaymentDate int
      ,@ClosedDate int
      ,@CancelledAmount varchar(50)
      ,@UndisbursedAmount varchar(50)
      ,@DisbursedAmount varchar(50)
      ,@ExchangeAdjustment varchar(50)
      ,@LoansHeld varchar(50)
      ,@RepaymentID int
      ,@RepaidtoIBRD varchar(50)
      ,@DuetoIBRD varchar(50)
      ,@BorrowerObligation varchar(50)
      ,@3rdPartyRefID int
      ,@Sold3rdParty varchar(50)
      ,@Repaid3rdParty varchar(50)
      ,@Due3rdParty varchar(50)
      ,@InsertDate datetime
      ,@Modifieddate datetime
      ,@Accm_Txn_Create_Time datetime
      ,@Accm_Txn_Complete_Time datetime
      ,@Accm_Txn_Process_Hours int
       As
       update dbo.FactLoanTransactions
       set
       Accm_Txn_Complete_Time = @Accm_Txn_Complete_Time,
       Accm_Txn_Process_Hours = @Accm_Txn_Process_Hours
       where LoanStatementID = @LoanStatementID
```

Stored Procedure I used to truncate Fact Loan Table to avoid duplicating data when reloading.