Sri Lanka Institute of Information Technology

Data Warehousing and Business Intelligence

(IT3021)

Automobile Loan Default Program

Assignment 1

IT20135270

Kollure K.A.D.D

Y3S1 Group 4 (DS – Weekend)

Contents

Step 1: Dataset Selection

Link to the selected dataset → **https://www.kaggle.com/datasets/saurabhbagchi/dish-network-hackathon**

This dataset contains automobile loan data of a non-banking financial institution (NBFI) which is a Financial Institution does not have a full banking license or is not supervised by a national or international banking regulatory agency. NBFI facilitates bank-related financial services, such as investment, risk pooling, contractual savings, and market brokering.

An NBFI is struggling to mark profits due to an increase in defaults in the vehicle loan category. The company aims to determine the client's loan repayment abilities and understand the relative importance of each parameter contributing to a borrower's ability to repay the loan.

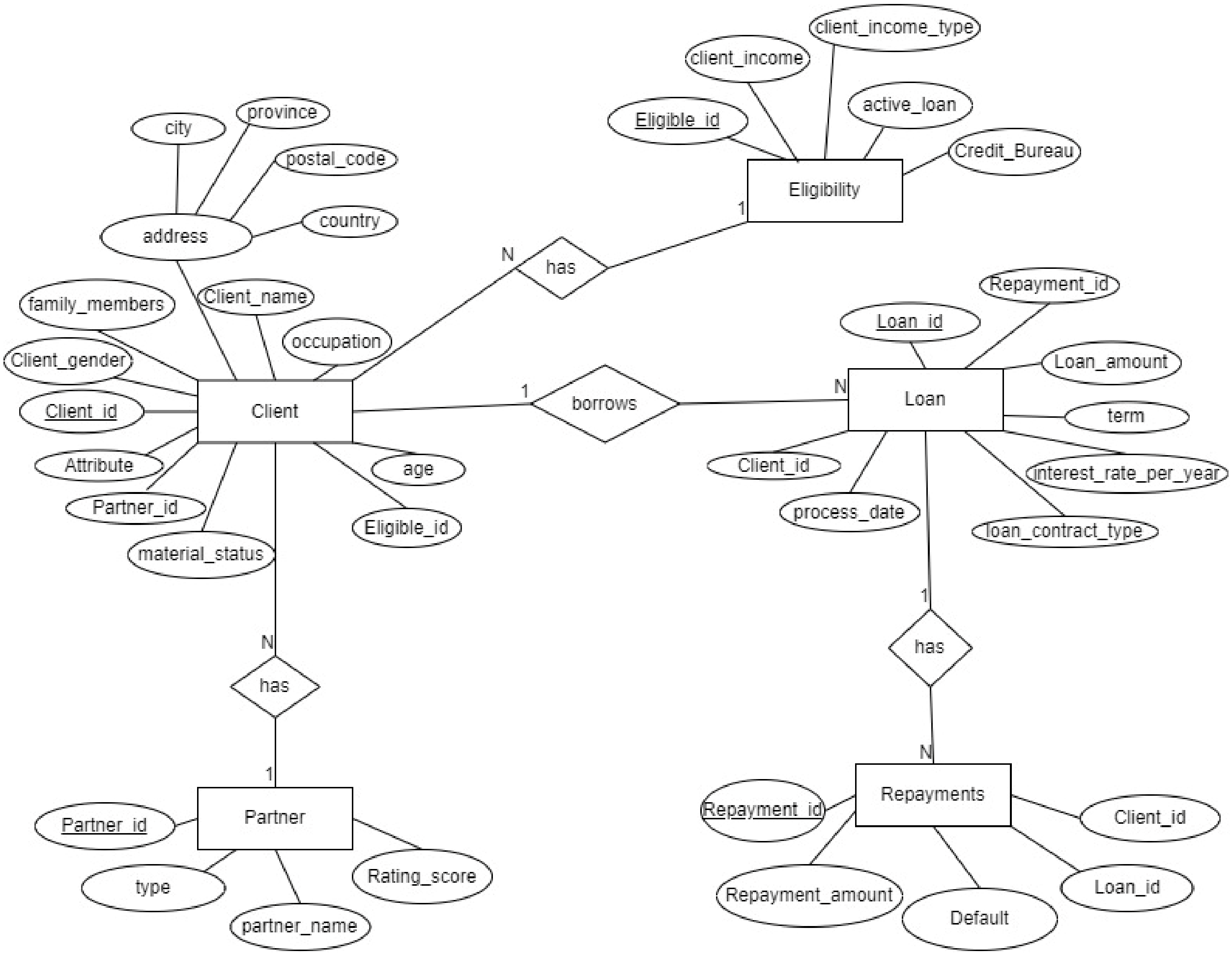
The goal of the problem is to predict whether a client will default on the vehicle loan payment or not.

The original dataset has less tables. I cut the columns of original source tables and put them into different source tables to get more dimensions and a hierarchy, because the assignment document says that we need to enrich the ETL process.

The dataset contains Automobile Loan details.

- Client Details
- Client Partner Details
- Loan Details
- Repayments Details
- Eligibility Details

ER Diagram


client_income_type
client_income
active_loan
Eligible_id
Credit_Bureau
Eligibility
1
N
has
province
city
postal_code
country
address
family_members
Client_name
occupation
Client_gender
Client_id
Client
Attribute
Partner_id
material_status
age
Eligible_id
1
borrows
N
Repayment_id
Loan_id
Loan_amount
Loan
term
Client_id
interest_rate_per_year
process_date
loan_contract_type
1
has
N
N
has
1
Partner_id
Partner
Rating_score
type
partner_name
Repayment_id
Repayments
Client_id
Repayment_amount
Default
Loan_id

Step 2: Preparation of Data Sources

The whole of data was in ‘csv’ file type, and they were separated into the following data sources, Text and csv. And they were used to create the following,

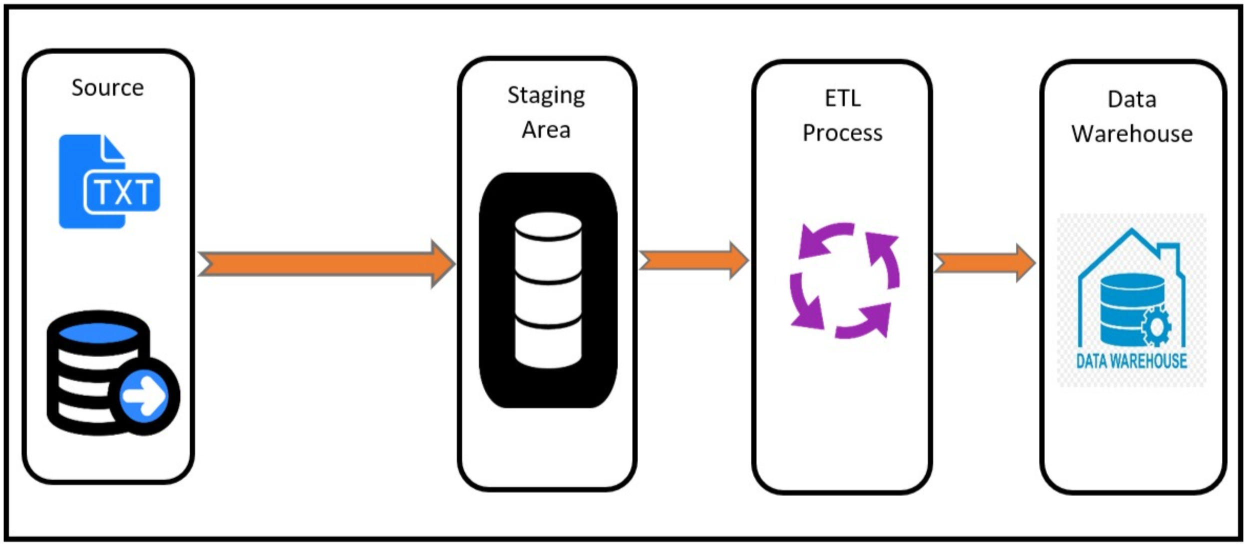
1. **Comma Separated Values (.csv)**
   Client.csv, partner.csv, loan.csv, repayments.csv, eligibility.csv
2. **Text (.txt)**
   client address.txt

Data Source Type	Source Name	Column Name	Data Type	Description
CSV Files	Client.csv	Client_id	int	Unique ID
		Client_name	Varchar (50)	Client’s name
		Client_Gender	Varchar (50)	Client’s gender
		age	Varchar (50)	Client’s age
		address	Varchar (100)	Client’d address
		city	Varchar (30)	Client’s city
		province	Varchar (50)	Client’s province
		postalCode	Varchar (15)	Client’s postalCode
		country	Varchar (50)	Client’s country
		Martial_Status	Varchar (50)	Client’s martial status
		Family_Members	Varchar (50)	Number of family members in client’s family
		Child_Count	Varchar (50)	Number of child in client’s family
		Occupation	Varchar (50)	Client’s occupation
		Eligible_id	int	id of the eligibility
		Partner_id	int	Id of the client partner
	eligibility.csv	Eligible_id	int	Unique ID
		Client_Income	Varchar (50)	Client’s income
		Client_Income_Type	Varchar (50)	Client’s income type
		Active_Loan	Varchar (50)	Number of active loans
		Credit_Bureau	Varchar (50)	Total number of enquiries in last year
	partner.csv	Partner_id	int	Unique ID
		type	Varchar (50)	Who accompanied the client when client applied for the loan
		Partner_name	Varchar (50)	Accompony partner’s name

		Rating_score	Varchar (50)	Accompony partner’s rating score
		Client_id	int	Client’s ID
	loan.csv	Loan_id	int	Unique ID
		LoanAmount	int	Amount of the loan
		Term	int	Terms for loan
		InterestRatePerYear	float	Interest Rate Per Year
		Loan_Contract_Type	Varchar (50)	Loan Type (CL-Cash Loan, RL-Revolving Loan)
		Process_date	Varchar (50)	Date that process the loan
		Client_id	int	Client’s ID
		Repayment_id	int	ID of the repayments
	Repayments.csv	Repayment_id	int	Unique ID
		repayment_amount	Varchar (50)	Amount of repayment
		Default	Varchar (50)	1 means the client defaulted on loan payments and 0 means otherwise
		Loan_id	int	ID for the laon
		Client_id	int	Client’s ID
Text File	client address.txt	Client_id	int	Unique ID
		address	Varchar (100)	Client’s address
		city	Varchar (30)	Client’s city
		province	Varchar (50)	Client’s province
		postalCode	Varchar (15)	Client’s postal code
		country	Varchar (50)	Client’s country

Step3: 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 Process**: 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.

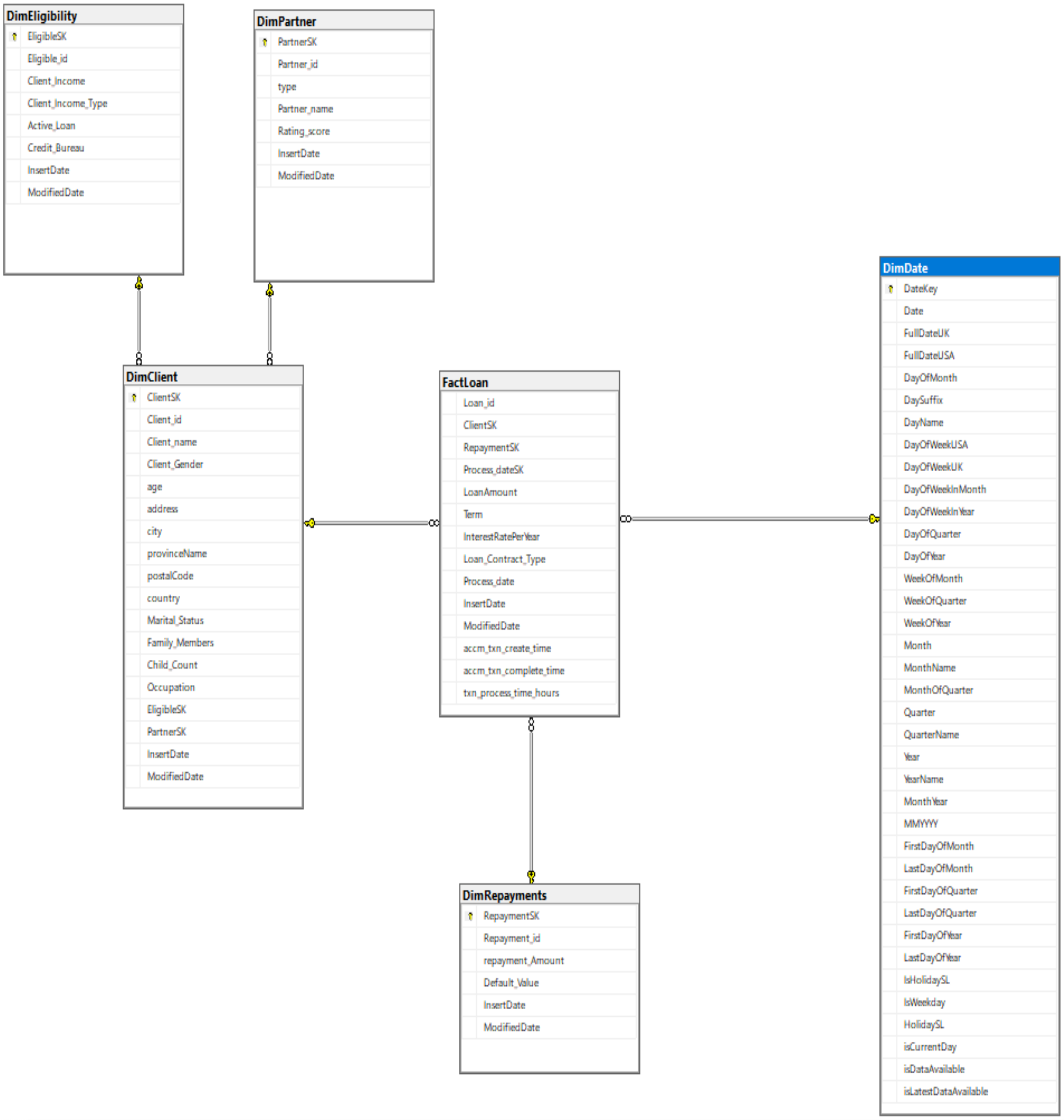
Step4: Datawarehouse design and development

Dimensional Model

Snowflake schema was selected to design the Data Warehouse of **Automobile Loan Data** an 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:

- DimClient→ Slowly Changing Dimension
- DimEligibility
- DimPartner
- DimRepayments
- DimDate
- FactLoan→ Fact Table



Hierarchies

Hierarchies in DimClient: City→ Province → Country

Hierarchies in DimDate: Day → Month → Quarter → Year

Stored Procedures

Stored procedure for DimClient

```
CREATE PROCEDURE dbo.UpdateDimClient
@Client_id int,
@EligibleSK int,
@PartnerSK int,
@Client_name varchar(50),
@Client_Gender       varchar(50),
@age varchar(50),
@address varchar(50),
@city  varchar(50),
@provinceName varchar(50),
@postalCode   varchar(50),
@country varchar(50),
@Marital_Status varchar(50),
@Family_Members varchar(50),
@Child_Count varchar(50),
@Occupation varchar(50)

AS
BEGIN
if not exists (select ClientSK
from dbo.DimClient
where Client_id= @Client_id)
BEGIN
insert into dbo.DimClient
(Client_id, EligibleSK, PartnerSK, Client_name,Client_Gender,age,address,city,
provinceName,postalCode,country,Marital_Status,Family_Members,Child_Count,Occupation,
InsertDate, ModifiedDate)
values
(@Client_id, @EligibleSK, @PartnerSK,
@Client_name,@Client_Gender,@age,@address,@city,@provinceName,@postalCode,@country,@Marit
al_Status,@Family_Members,@Child_Count,@Occupation,GETDATE(), GETDATE())

END;
if exists (select ClientSK
from dbo.DimClient
where Client_id = @Client_id)
BEGIN
update dbo.DimClient
set  EligibleSK= @EligibleSK,
PartnerSK=@PartnerSK,
Client_name = @Client_name,
Client_Gender = @Client_Gender,
age=@age,
address=@address,
city=@city,
provinceName=@provinceName,
postalCode=@postalCode,
```

```sql
country=@country,
Marital_Status=@Marital_Status,
Family_Members=@Family_Members,
Child_Count=@Child_Count,
Occupation=@Occupation,
ModifiedDate = GETDATE()

where Client_id = @Client_id
END;
END;
```

Stored procedure for DimEligibility

```sql
CREATE PROCEDURE dbo.UpdateDimEligibility
@Eligible_id int,
@Client_Income varchar(50),
@Client_Income_Type varchar(50),
@Active_Loan varchar(50),
@Credit_Bureau varchar(50)
AS
BEGIN
if not exists (select EligibleSK
from dbo.DimEligibility
where Eligible_id = @Eligible_id)
BEGIN
insert into dbo.DimEligibility
(Eligible_id, Client_Income, Client_Income_Type,Active_Loan, Credit_Bureau,InsertDate,
ModifiedDate)
values
(@Eligible_id, @Client_Income, @Client_Income_Type,@Active_Loan,@Credit_Bureau,
GETDATE(), GETDATE())
END;

if exists (select EligibleSK
from dbo.DimEligibility
where Eligible_id = @Eligible_id)
BEGIN
update dbo.DimEligibility
set  Client_Income= @Client_Income,
Client_Income_Type = @Client_Income_Type,
Active_Loan = @Active_Loan,
Credit_Bureau=@Credit_Bureau,
ModifiedDate = GETDATE()

where  Eligible_id = @Eligible_id

END;
END;
```

Stored procedure for DimPartner

```
CREATE PROCEDURE dbo.UpdateDimPartner
@Partner_id int,
@type varchar(50),
@Partner_name varchar(50),
@Rating_score varchar(50)
AS
BEGIN
if not exists (select PartnerSK
from dbo.DimPartner
where Partner_id = @Partner_id)
BEGIN
insert into dbo.DimPartner
(Partner_id,type,Partner_name ,Rating_score,InsertDate, ModifiedDate )
values
(@Partner_id, @type, @Partner_name,@Rating_score, GETDATE(), GETDATE())
END;
if exists (select PartnerSK
from dbo.DimPartner
where Partner_id = @Partner_id)
BEGIN
update dbo.DimPartner
set  type= @type,
Partner_name = @Partner_name,
Rating_score = @Rating_score,
ModifiedDate = GETDATE()

where Partner_id  = @Partner_id
END;
END;
```

Stored procedure for DimRepayments

```
CREATE PROCEDURE dbo.UpdateDimRepayments
@Repayment_id int,
@repayment_amount varchar(50),
@Default_Value varchar(50)

AS
BEGIN
if not exists (select RepaymentSK
from dbo.DimRepayments
where Repayment_id = @Repayment_id)
BEGIN
insert into dbo.DimRepayments
(Repayment_id,repayment_amount,Default_Value, InsertDate, ModifiedDate )
values
(@Repayment_id, @repayment_amount, @Default_Value, GETDATE(), GETDATE())
END;
if exists (select RepaymentSK
from dbo.DimRepayments
where Repayment_id = @Repayment_id)
BEGIN
update dbo.DimRepayments
set  repayment_amount= @repayment_amount,
Default_Value = @Default_Value,
ModifiedDate = GETDATE()
```

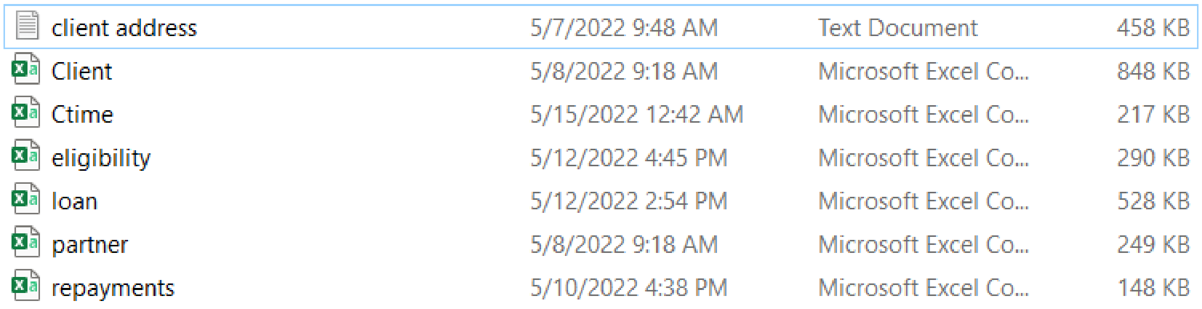
```
where Repayment_id  = @Repayment_id
END;
END;
```

Step 5: ETL development

ETL development process

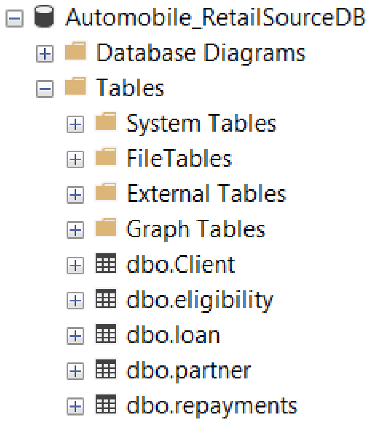
- Step 1: Setting up the environment

Text and CSV Files



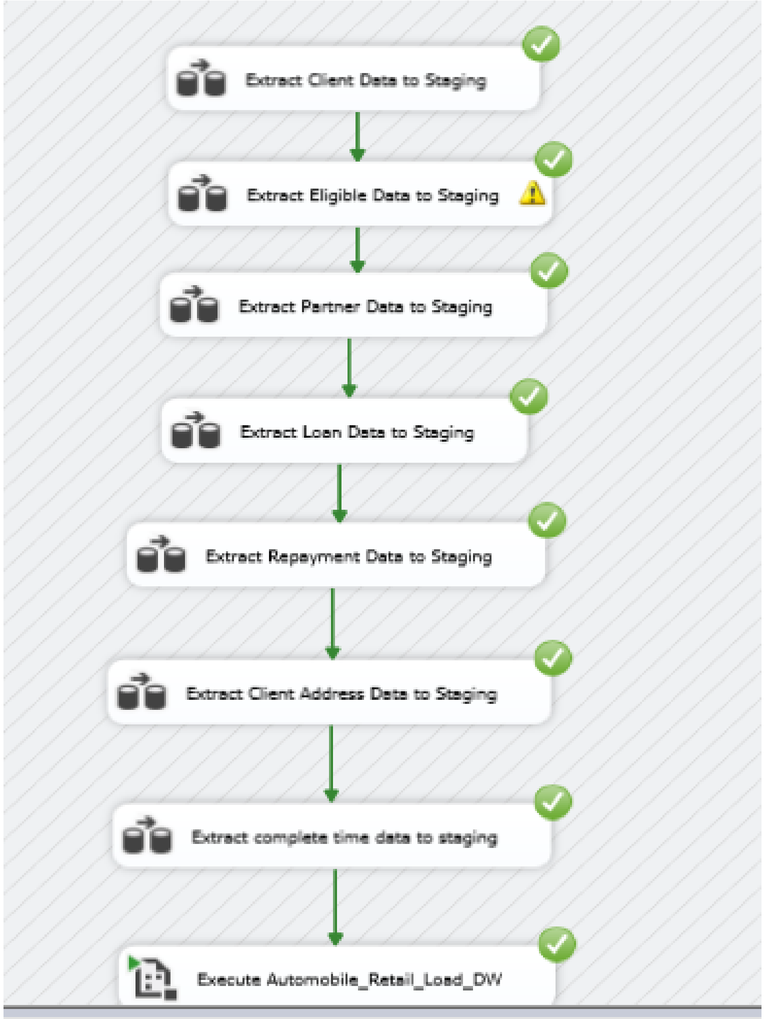
client address	5/7/2022 9:48 AM	Text Document	458 KB
Client	5/8/2022 9:18 AM	Microsoft Excel Co...	848 KB
Ctime	5/15/2022 12:42 AM	Microsoft Excel Co...	217 KB
eligibility	5/12/2022 4:45 PM	Microsoft Excel Co...	290 KB
loan	5/12/2022 2:54 PM	Microsoft Excel Co...	528 KB
partner	5/8/2022 9:18 AM	Microsoft Excel Co...	249 KB
repayments	5/10/2022 4:38 PM	Microsoft Excel Co...	148 KB

SourceDB in SSMS



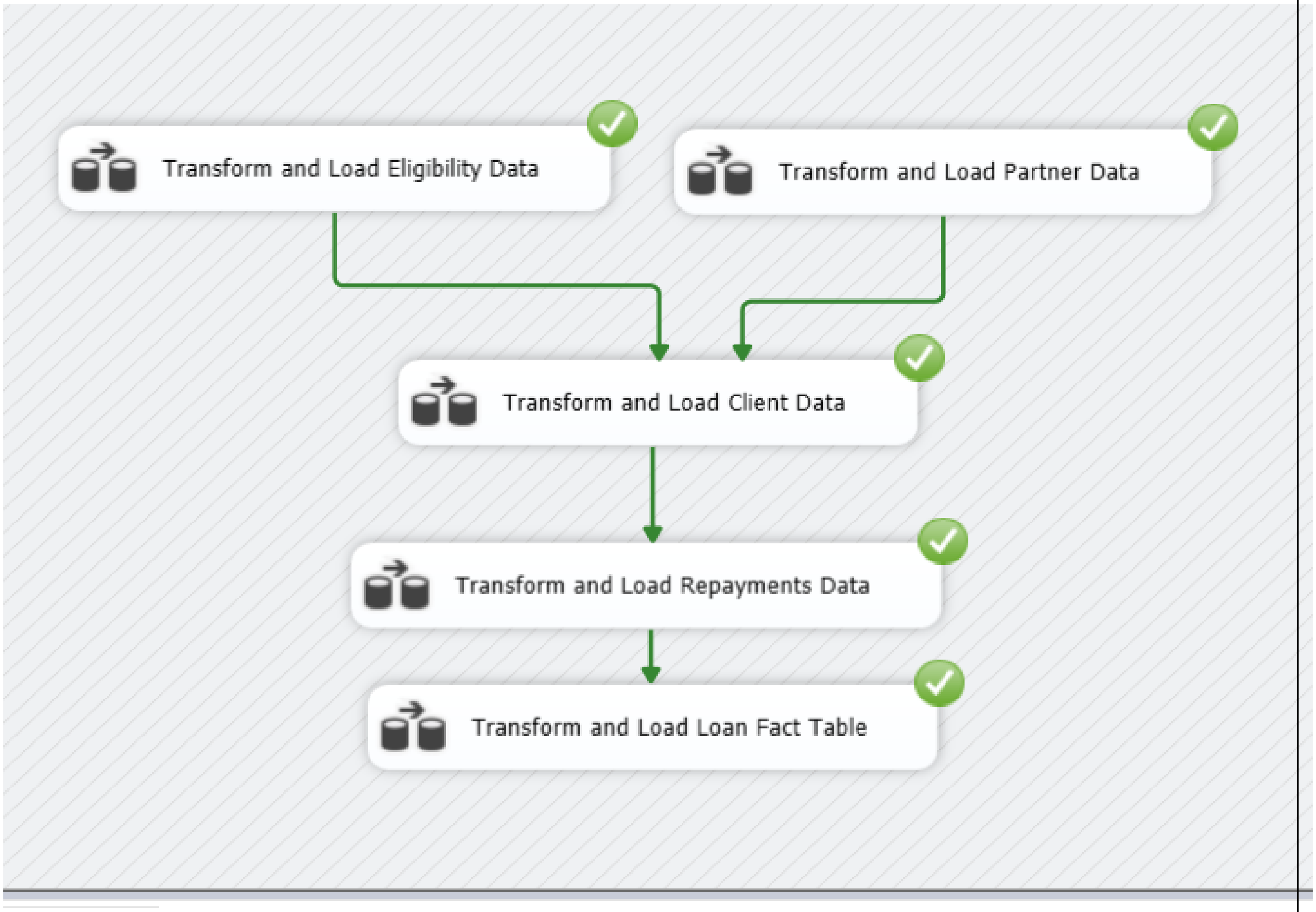
```
⊟ Automobile_RetailSourceDB
    ⊞ Database Diagrams
    ⊟ Tables
        ⊞ System Tables
        ⊞ FileTables
        ⊞ External Tables
        ⊞ Graph Tables
        ⊞ dbo.Client
        ⊞ dbo.eligibility
        ⊞ dbo.loan
        ⊞ dbo.partner
        ⊞ dbo.repayments
```

- Step 2: Data Extracting from Source to Staging Area

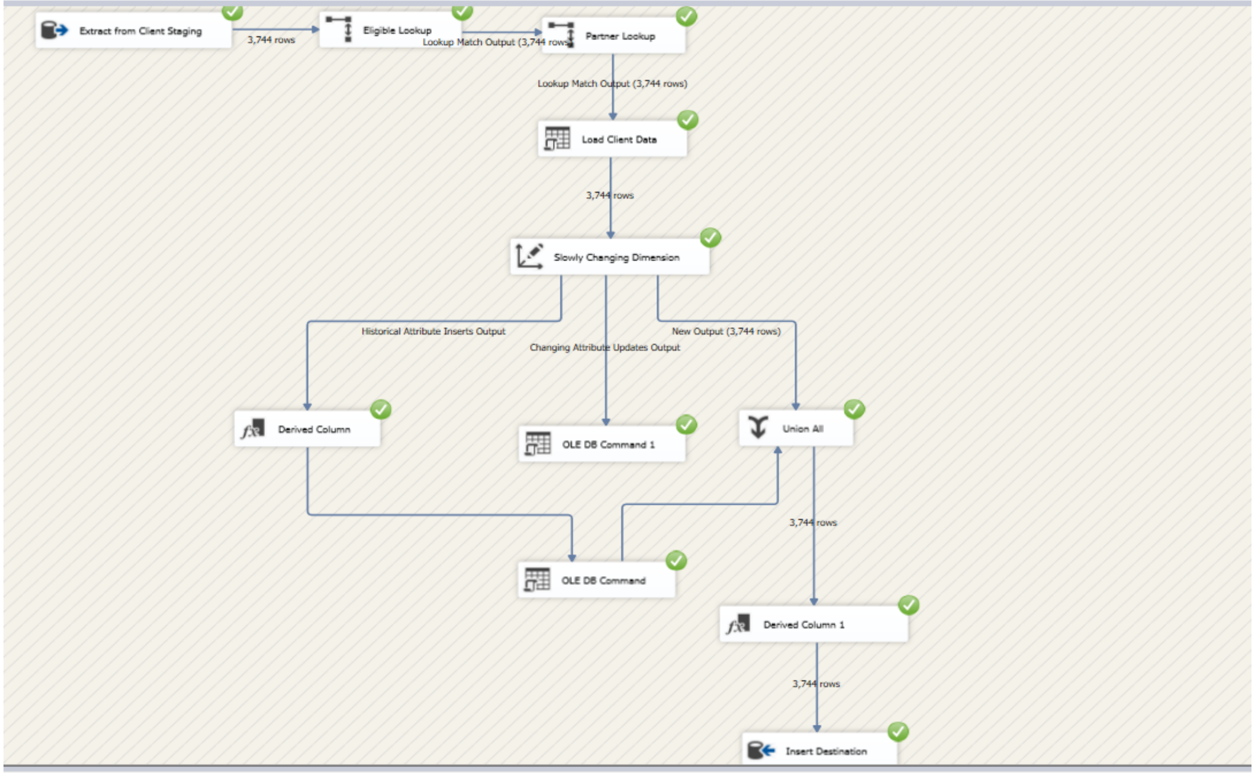

Extract Client Data to Staging
Extract Eligible Data to Staging
Extract Partner Data to Staging
Extract Loan Data to Staging
Extract Repayment Data to Staging
Extract Client Address Data to Staging
Extract complete time data to staging
Execute Automobile_Retail_Load_DW

- Step 4: Transform and Load Data to Datawarehouse from Staging

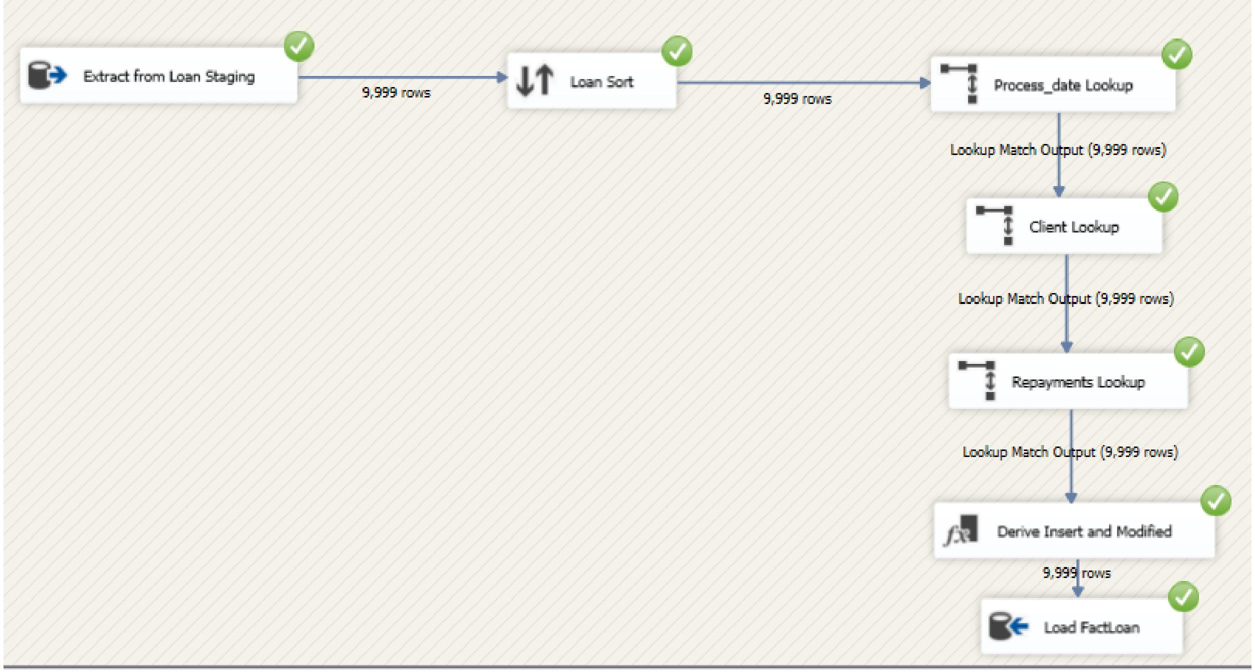
  Control flow


Transform and Load Eligibility Data
Transform and Load Partner Data
Transform and Load Client Data
Transform and Load Repayments Data
Transform and Load Loan Fact Table

Data Flow for Slowly Changing Dimension

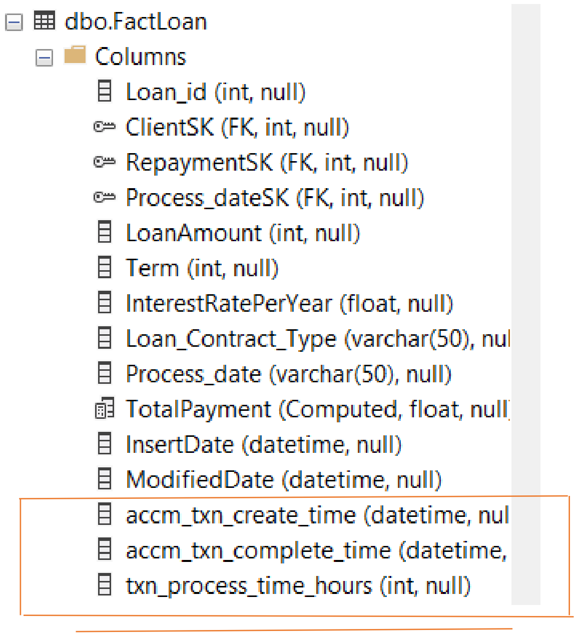


Data Flow for Fact Table


Extract from Loan Staging
9,999 rows
Loan Sort
9,999 rows
Process_date Lookup
Lookup Match Output (9,999 rows)
Client Lookup
Lookup Match Output (9,999 rows)
Repayments Lookup
Lookup Match Output (9,999 rows)
Derive Insert and Modified
9,999 rows
Load FactLoan

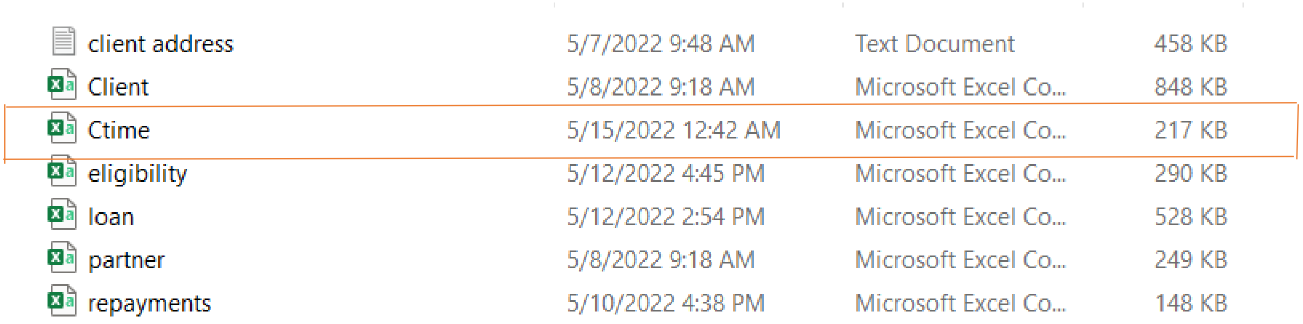
Step6: ETL Development - Accumulating Fact Tables

- Step 1: Extending Fact Table with Additional Columns



```
⊟ dbo.FactLoan
   ⊟ Columns
        Loan_id (int, null)
        ClientSK (FK, int, null)
        RepaymentSK (FK, int, null)
        Process_dateSK (FK, int, null)
        LoanAmount (int, null)
        Term (int, null)
        InterestRatePerYear (float, null)
        Loan_Contract_Type (varchar(50), nul
        Process_date (varchar(50), null)
        TotalPayment (Computed, float, null
        InsertDate (datetime, null)
        ModifiedDate (datetime, null)
        accm_txn_create_time (datetime, nul
        accm_txn_complete_time (datetime,
        txn_process_time_hours (int, null)
```

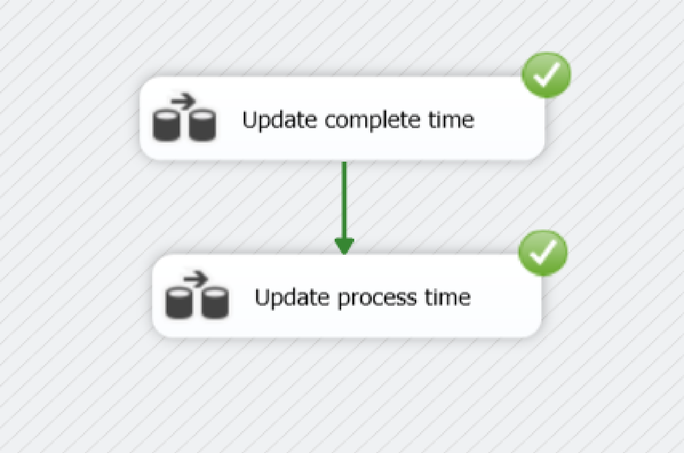
- Step 2: Prepare separate data set for complete time



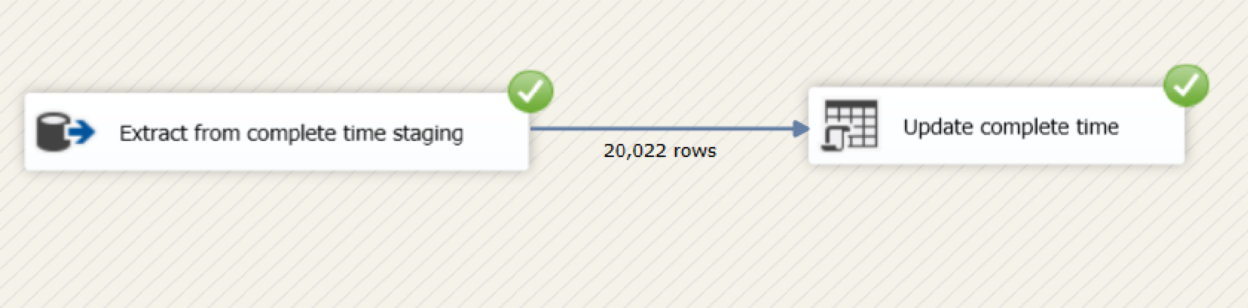
client address	5/7/2022 9:48 AM	Text Document	458 KB
Client	5/8/2022 9:18 AM	Microsoft Excel Co...	848 KB
Ctime	5/15/2022 12:42 AM	Microsoft Excel Co...	217 KB
eligibility	5/12/2022 4:45 PM	Microsoft Excel Co...	290 KB
loan	5/12/2022 2:54 PM	Microsoft Excel Co...	528 KB
partner	5/8/2022 9:18 AM	Microsoft Excel Co...	249 KB
repayments	5/10/2022 4:38 PM	Microsoft Excel Co...	148 KB

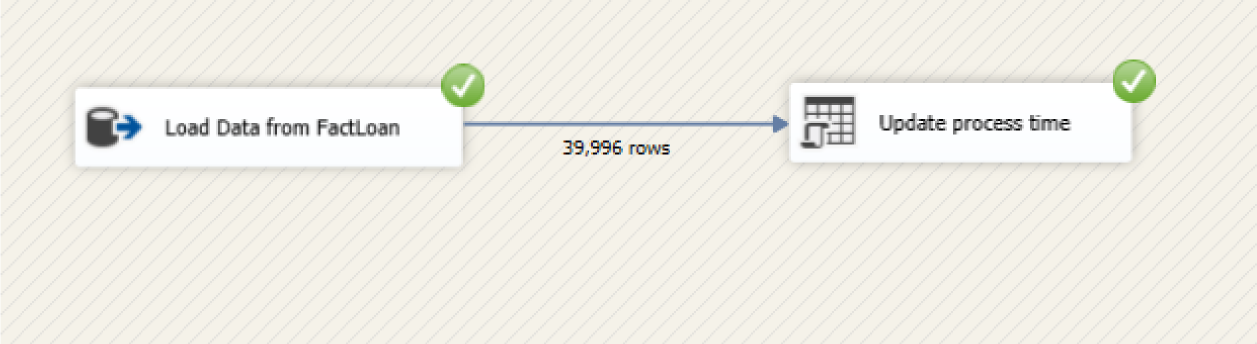
- Step 3: Update Complete Time and Process Time in Fact Table

Control Flow



Data Flows


Extract from complete time staging
20,022 rows
Update complete time


Load Data from FactLoan
39,996 rows
Update process time