**Reports for Warehousing concepts on Bank Database concerning Branch, Account and Customer Services**

**By,**

**ANUDEEP RAJU BASAVARAJU**

# **INDEX**

1. Background for the Development of chosen system.......................................................................3
   1. Source of Data Selection....................................................................................................................................3
   2. ER-Diagram...............................................................................................................................4
   3. Description of Source Table..........................................................................................................................................5
   4. Subject Area of Analysis............................................................................................................5
   5. Key Stakeholders..............................................................................................................................5
   6. Vision and Goals of Data warehouse.........................................................................................6
   7. Report Generation Requirements.............................................................................................6
2. Schema Generation for Data warehouse.........................................................................................7
   1. Selection of Schema..................................................................................................................7
   2. ER-Diagram and Illustration.................................................................................................................................8
3. Implementation of Data warehouse..............................................................................................10
   1. Creation of Dimension Table...................................................................................................10

3.1.1 ETL Process for Dimension Tables......................................................................................................................................10

* 1. Creation of Fact Table........................................................................................................................................13

3.2.1 ETL Process for Fact Tables.............................................................................................13

1. Report Generations...................................................................................................................................15
   1. SSRS Reports............................................................................................................................15
   2. Visualization Reports(R studio).....................................................................................................................................20
2. XML Schema Generation.....................................................................................................................................24
   1. XML and XSD...........................................................................................................................24
   2. Validation of Schema...............................................................................................................26
3. Modelling and Graph Data Models................................................................................................27
   1. Graph using Neo4j...................................................................................................................27
   2. Importing CSV file....................................................................................................................27
   3. Relation Between Tables.........................................................................................................28

BIBLIOGRAPHY...............................................................................................................................38

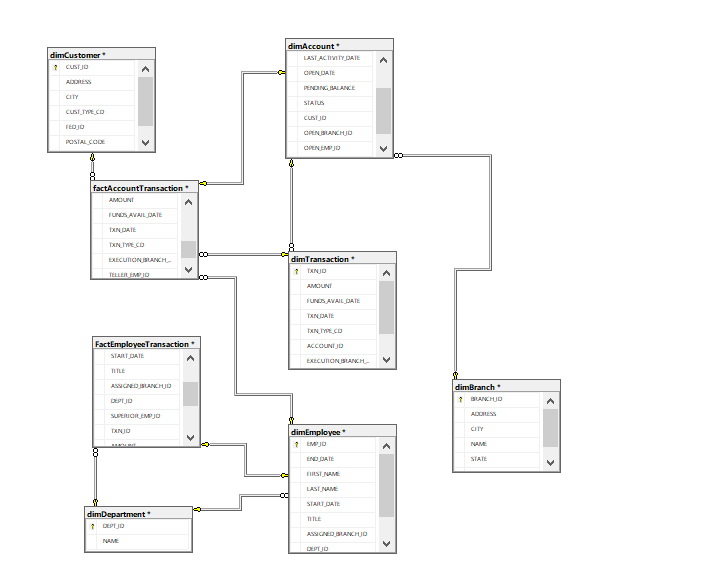
# **1. Background for the Development of chosen system:**

Banking is an ever-growing sector of finance and the economy. To keep track of areas of growth and areas where improvement is required is of great interest. We are using the data warehousing concepts to provide the important data at the same place using the ETL process and also providing a report for the data collected and visualizing the data using different angles.

## 1.1 Source of Data Selection:

By making use of a sample SQL database with bank account details that provides valuable and good quality data for the branch of the bank. We can directly run the SQL query to restore the dataset tables and data into our database server.

## 1.2 ER-Diagram :



## 1.3 Description of Source Table :

**The tables available in the dataset are as follows:**

* Account
* Branch
* Customer
* Department
* Employee
* Transaction

## 1.4 Subject Area of Analysis:

We are targeting the business part of the bank system where our main target will be the different transactions happening and the frequent employees and customers related to it and the branches they belong to so that we have a better picture of the whole bank system and its environment.

## 1.5 Key Stakeholders :

**The key stakeholders that we have selected for our data warehouse from the available datasets are:**

* Account
* Branch
* Customer
* Department
* Employee
* Transaction

## 1.6 Vision and Goals of Data warehouse:

To have the best banking performance by keeping track of transactions and the customers and branches involved.

## 1.7 Report Generation Requirements:

**SSRS Report**

* Report to show names of people having balance greater than 2000
* Report to show the number of accounts per customer
* Report to show the pending balance after each transaction
* The report showing the total available balance.

**R Visualization**

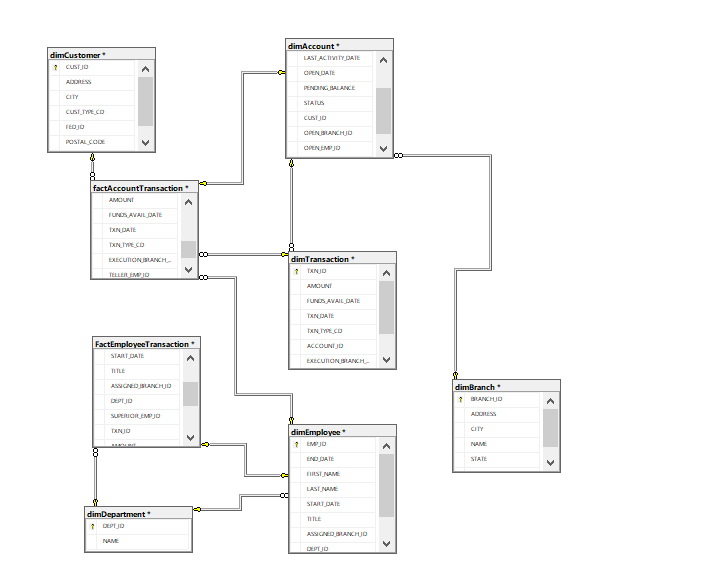
* Show names of people having a balance greater than 2000
* Show number of accounts per customer
* Show pending balance after each transaction
* Show total available balance.

# **2. Schema Generation for Data warehouse**

## 2.1 Selection of Schema:

Making use of the WHERE clause and JOIN operations is very common for our warehouse as fact tables are related to multiple dimension tables. Since STAR schema provides better JOIN operation and WHERE clauses usage to create a table we make use of STAR Schema.

## 2.2 ER-Diagram and Illustration :



Our warehouse makes use of five dimension tables which are as follows :

* dimAccount - keeps track of the details of accounts made in the banks.
* dimBranch- keeps track of the branches of the bank.
* dimCustomer - keeps track of the details of the customers linked to the bank.
* dimEmployees- keeps track of Employees working for the bank
* dimTransaction- keeps track of the transactions done in the bank.

.

**We have two fact tables for the data warehouse:**

* FactAccountTransaction - keeps track of transaction details linked to the accounts made in the banks.
* FactEmployeeTransaction - keeps track of Employees related to the transaction done.

# **3. Implementation of Data warehouse**

## 3.1 Creation of Dimension Table:

--we will create the transaction dimension table using the following query :

CREATE TABLE dimTransaction (

TXN\_ID numeric(19,0) not null,

AMOUNT float,

FUNDS\_AVAIL\_DATE datetime,

TXN\_DATE datetime,

TXN\_TYPE\_CD varchar(10),

ACCOUNT\_ID int ,

EXECUTION\_BRANCH\_ID int,

TELLER\_EMP\_ID int

)

alter table dimTransaction

add primary key(TXN\_ID)

--we will create the branch dimension table using the following query :

CREATE TABLE dimBranch (

BRANCH\_ID int not null,

ADDRESS varchar(30),

CITY varchar(20),

NAME varchar(20),

STATE varchar(10),

ZIP\_CODE varchar(12)

)

alter table dimBranch

add primary key(BRANCH\_ID)

--we will create the Account dimension table using the following query :

CREATE TABLE dimAccount (

ACCOUNT\_ID int not null,

AVAIL\_BALANCE float,

CLOSE\_DATE datetime,

LAST\_ACTIVITY\_DATE datetime,

OPEN\_DATE datetime,

PENDING\_BALANCE float,

STATUS varchar(10),

CUST\_ID int,

OPEN\_BRANCH\_ID int,

OPEN\_EMP\_ID int,

PRODUCT\_CD varchar(10)

)

alter table dimAccount

add primary key(ACCOUNT\_ID)

--we will create the Employee dimension table using the following query :

CREATE TABLE dimEmployee (

EMP\_ID int not null,

END\_DATE datetime,

FIRST\_NAME varchar(20),

LAST\_NAME varchar(20),

START\_DATE datetime,

TITLE varchar(20),

ASSIGNED\_BRANCH\_ID int,

DEPT\_ID int,

SUPERIOR\_EMP\_ID int

)

alter table dimEmployee

add primary key(EMP\_ID)

--we will create the customer dimension table using the following query :

CREATE TABLE dimCustomer (

CUST\_ID int not null,

ADDRESS varchar(30),

CITY varchar(20),

CUST\_TYPE\_CD varchar(1),

FED\_ID varchar(12),

POSTAL\_CODE varchar(10),

STATE varchar(20)

)

alter table dimCustomer

add primary key(CUST\_ID)

--we will create the department dimension table using the following query :

CREATE TABLE dimDepartment (

DEPT\_ID int not null,

NAME varchar(20)

)

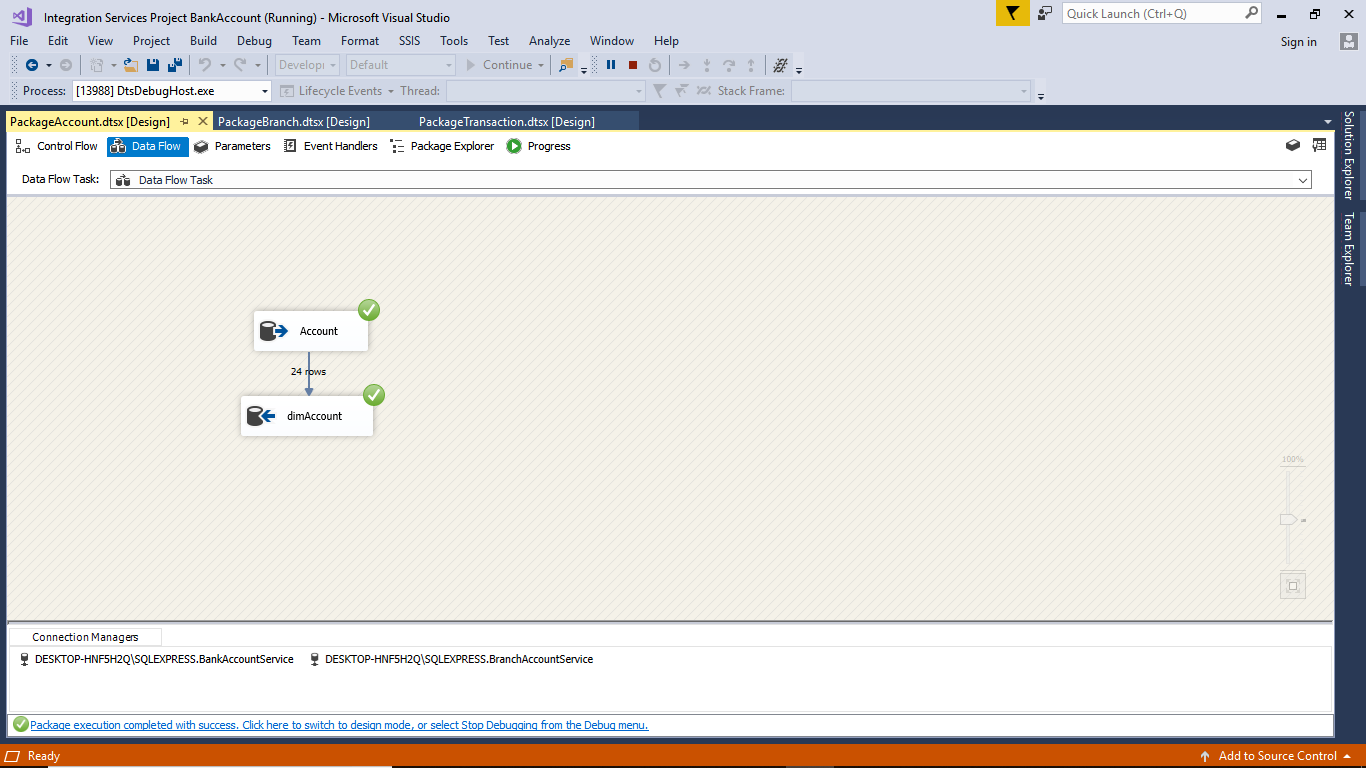
alter table dimDepartment

add primary key(DEPT\_ID)

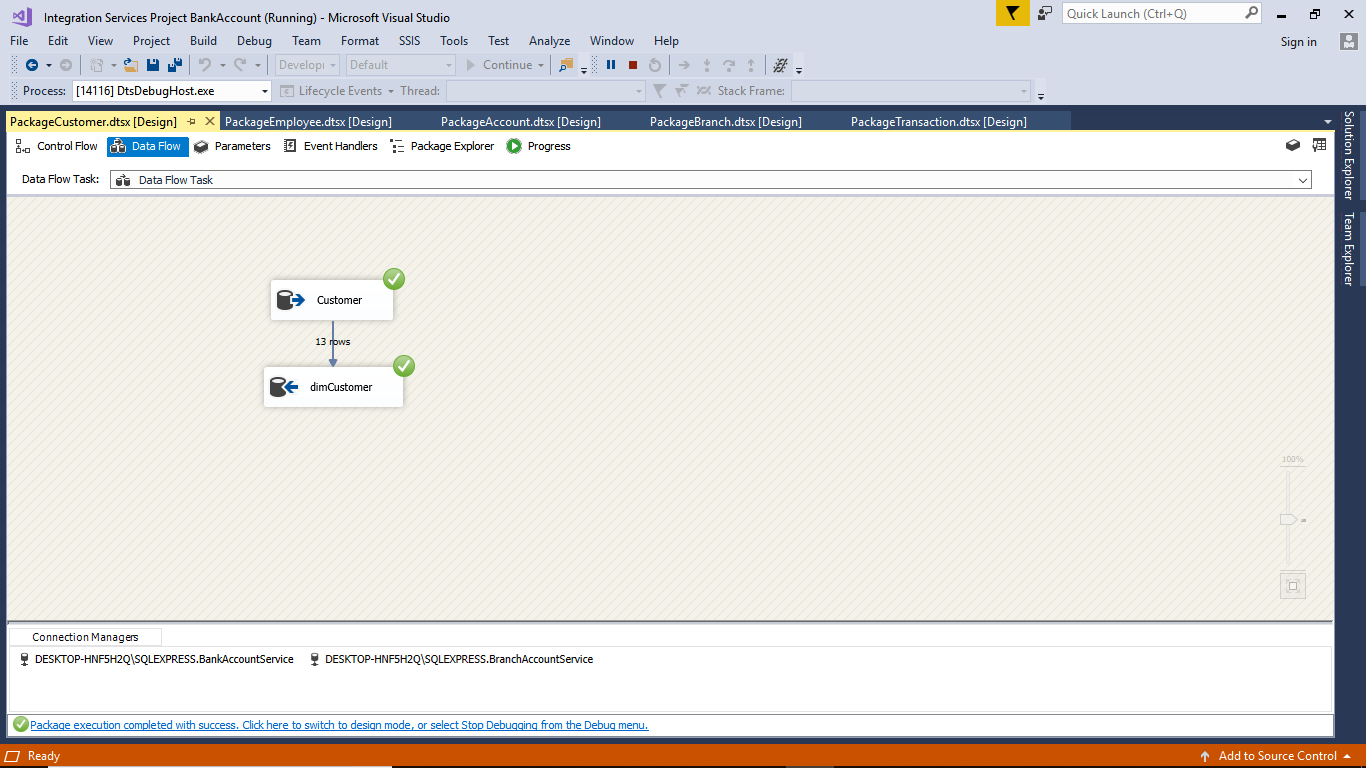
## 3.1.1 ETL Process for Dimension Tables:

Once the creation of the dimension table is done by executing the SQL queries we will define SSIS packages for defining the flow of data from our source dataset to our destination dimension table and once the package is executed the data is transferred to our destination.

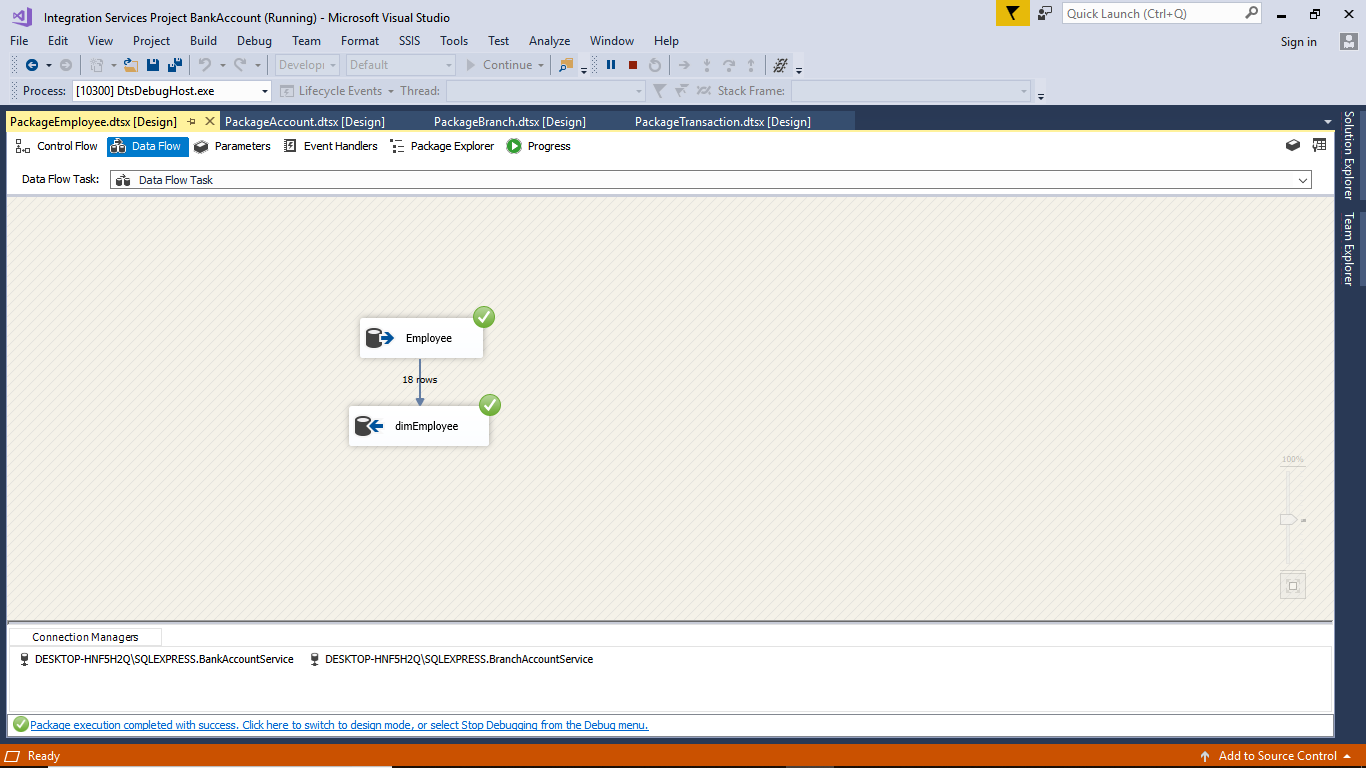
-- Package for our dimension table ‘dimAccount’ is defined as below:



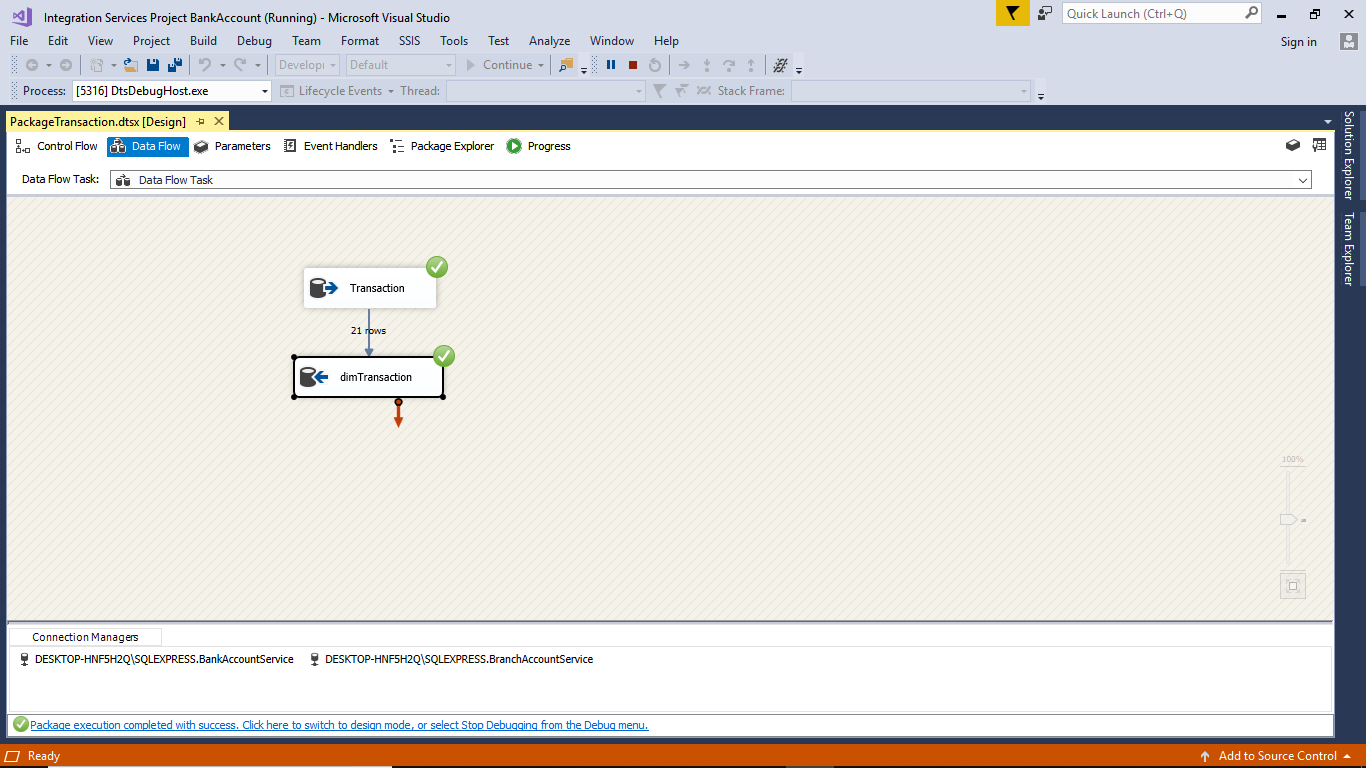
-- Package for our dimension table ‘dimCustomer’ is defined as below:



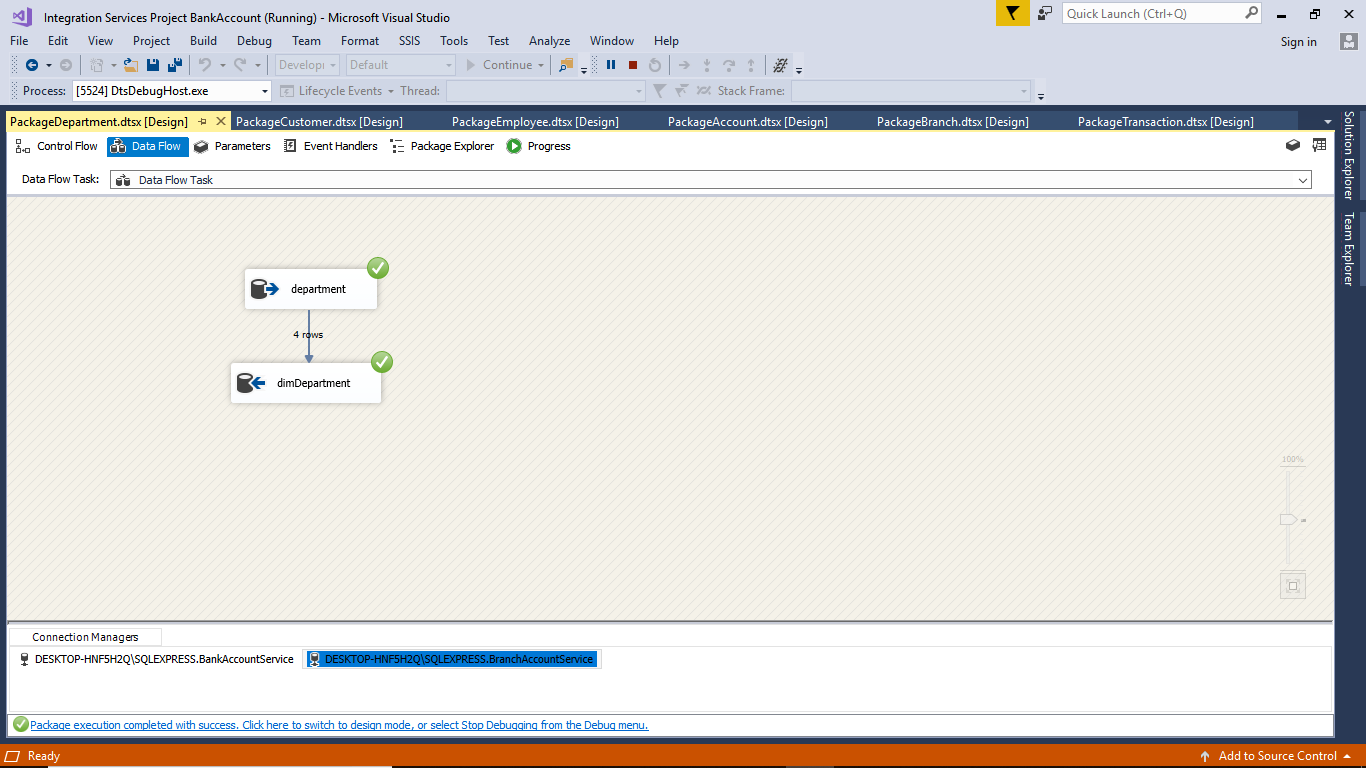
-- Package for our dimension table ‘dimEmployee’ is defined as below:



-- Package for our dimension table ‘dimTransaction’ is defined as below:



-- Package for our dimension table ‘dimDepartment’ is defined as below:



## 3.2 Creation of Fact Table:

--we will create the ‘FactAccountTransaction’ fact table using the following query :

CREATE TABLE factAccountTransaction (

ACCOUNT\_ID int not null,

AVAIL\_BALANCE float,

CLOSE\_DATE datetime,

LAST\_ACTIVITY\_DATE datetime,

OPEN\_DATE datetime,

PENDING\_BALANCE float,

STATUS varchar(10),

CUST\_ID int,

OPEN\_BRANCH\_ID int,

OPEN\_EMP\_ID int,

PRODUCT\_CD varchar(10),

TXN\_ID numeric(19,0),

AMOUNT float,

FUNDS\_AVAIL\_DATE datetime,

TXN\_DATE datetime,

TXN\_TYPE\_CD varchar(10),

EXECUTION\_BRANCH\_ID int,

TELLER\_EMP\_ID int

)

alter table factAccountTransaction

add primary key(ACCOUNT\_ID)

--we will create the ‘FactEmployeeTransaction’ fact table using the following query :

CREATE TABLE FactEmployeeTransaction (

EMP\_ID int not null,

END\_DATE datetime,

FIRST\_NAME varchar(20),

LAST\_NAME varchar(20),

START\_DATE datetime,

TITLE varchar(20),

ASSIGNED\_BRANCH\_ID int,

DEPT\_ID int,

SUPERIOR\_EMP\_ID int,

TXN\_ID numeric(19,0),

AMOUNT float,

FUNDS\_AVAIL\_DATE datetime,

TXN\_DATE datetime,

TXN\_TYPE\_CD varchar(10),

ACCOUNT\_ID int,

EXECUTION\_BRANCH\_ID int

)

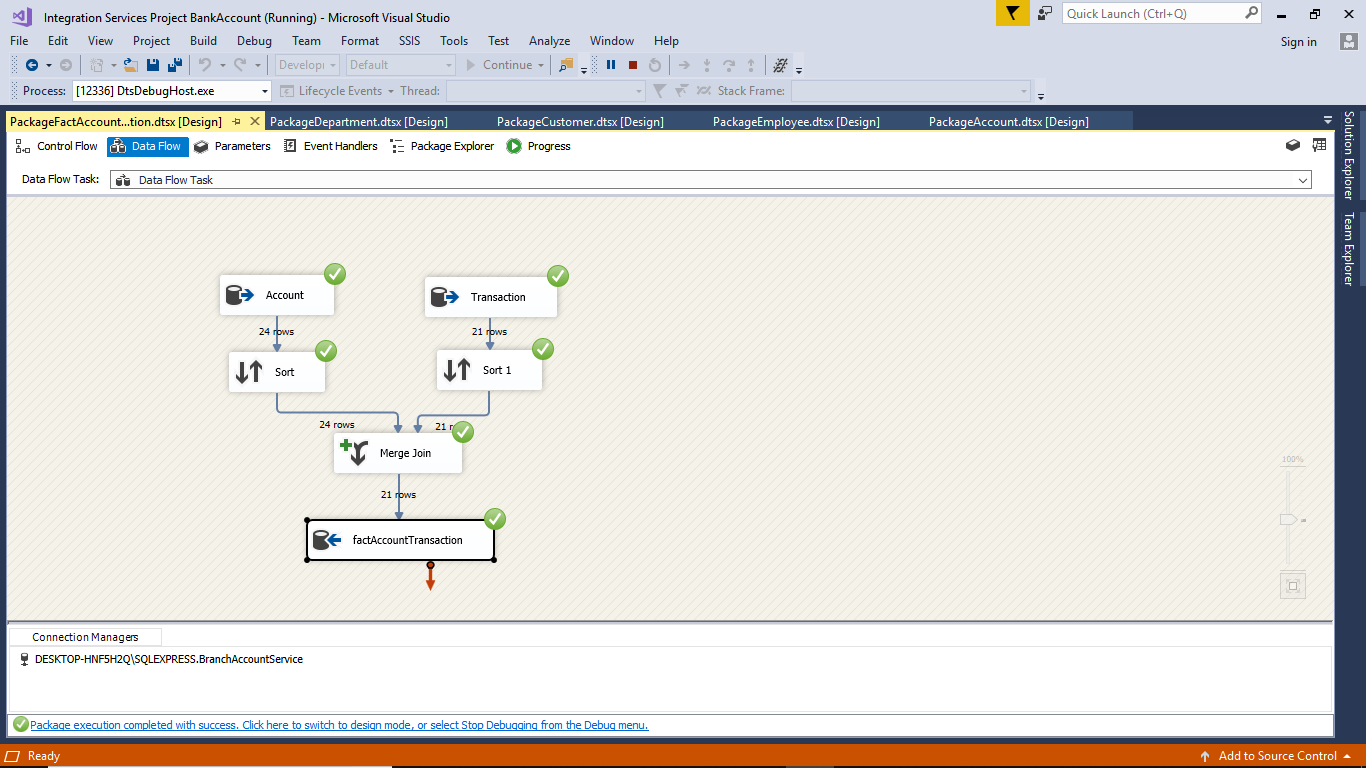
alter table FactEmployeeTransaction

add primary key(EMP\_ID)

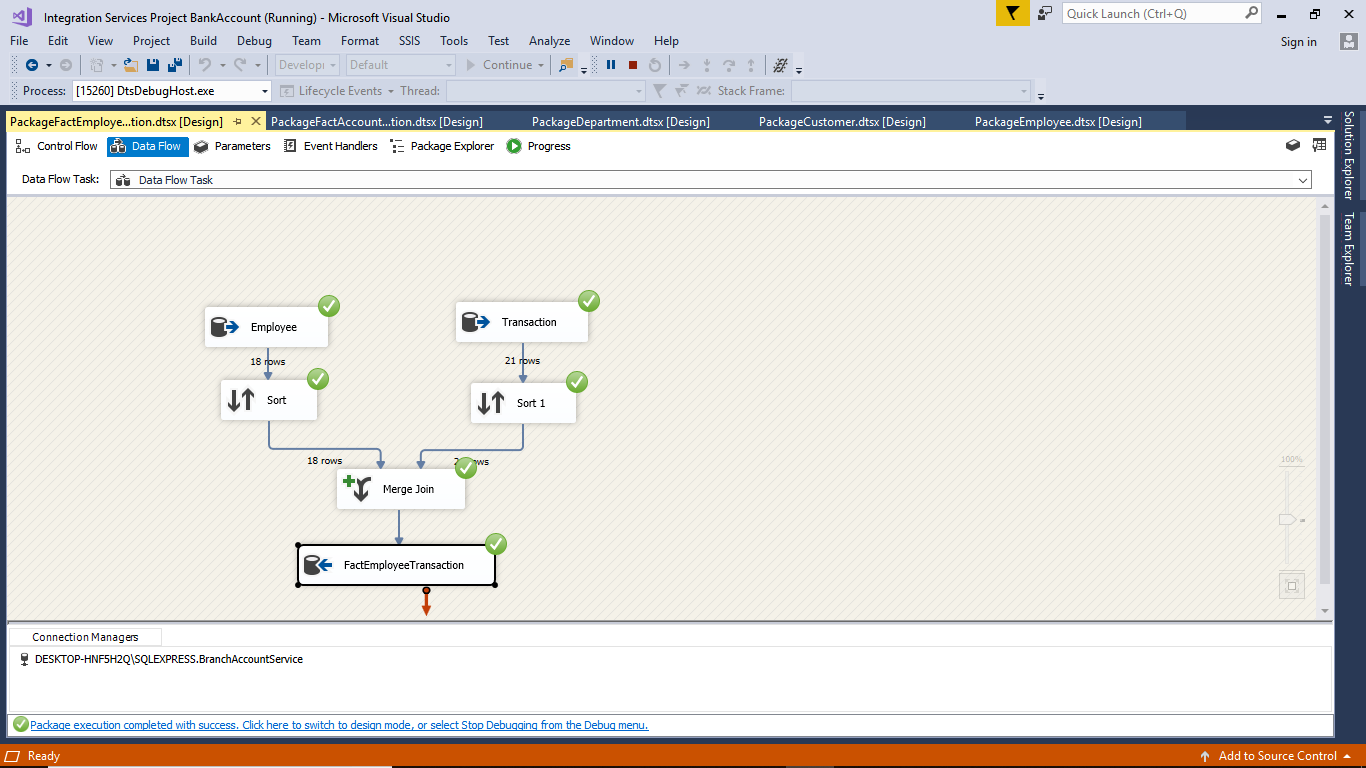
## 3.2.1 ETL Process for Fact Tables:

Once we are done with the creation of fact tables we can create packages for data flow from multiple dimension tables to our fact tables.

-- Package for our fact table ‘FactAccountTransaction’ is defined as below:



-- Package for our fact table ‘FactEmployeeTransaction’ is defined as below:

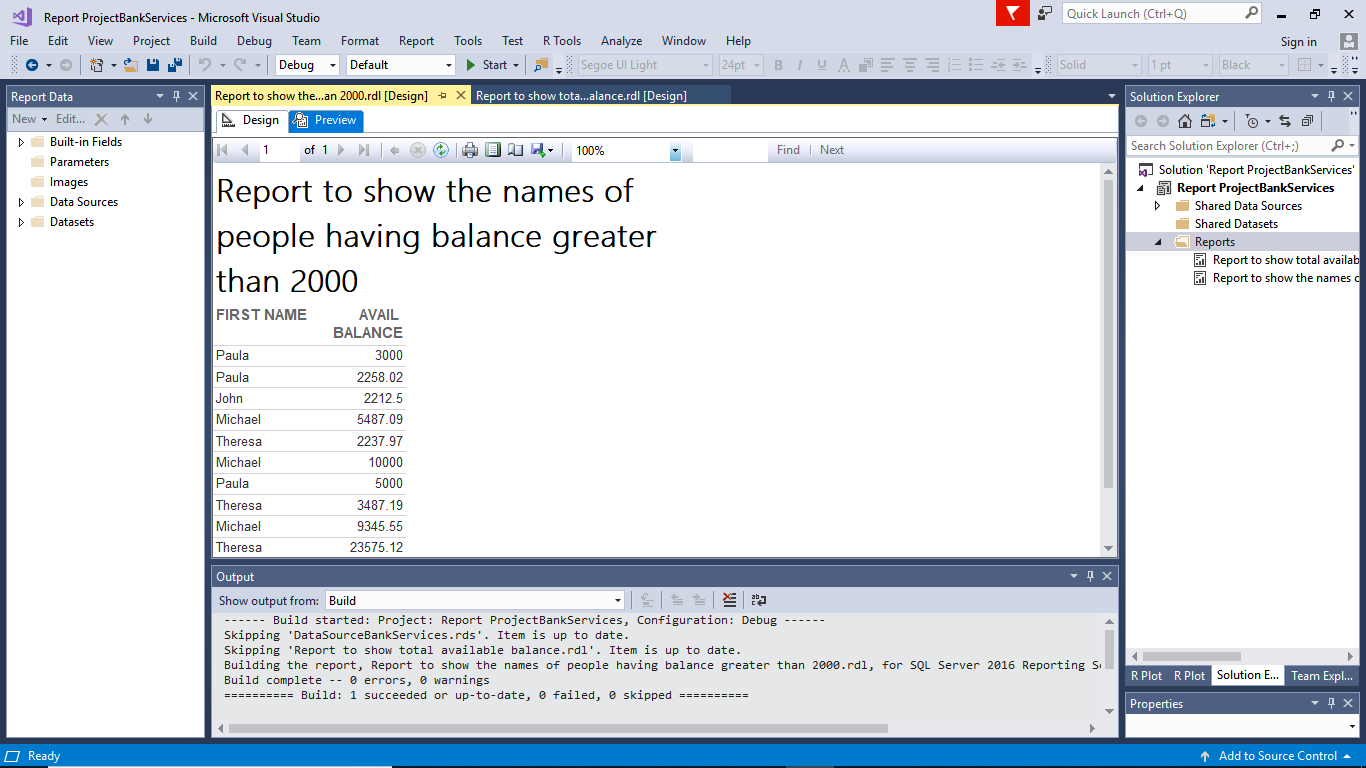


# **4. Report Generations**

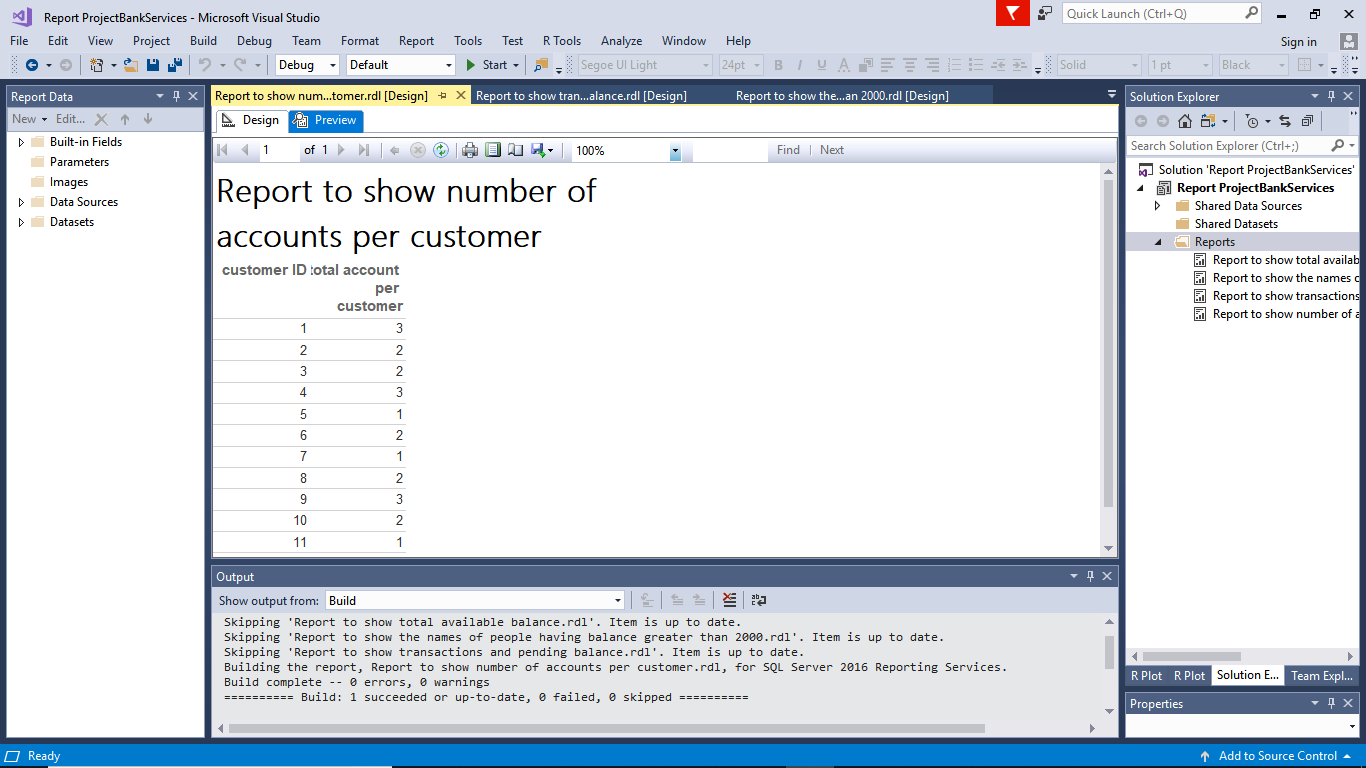
## 4.1 SSRS Reports:

We will write SQL queries for getting the important data from the fact tables and the dimension tables and we will use these queries to generate reports using SSRS report. Once we have the report we can store it in different formats and can use it for reference.

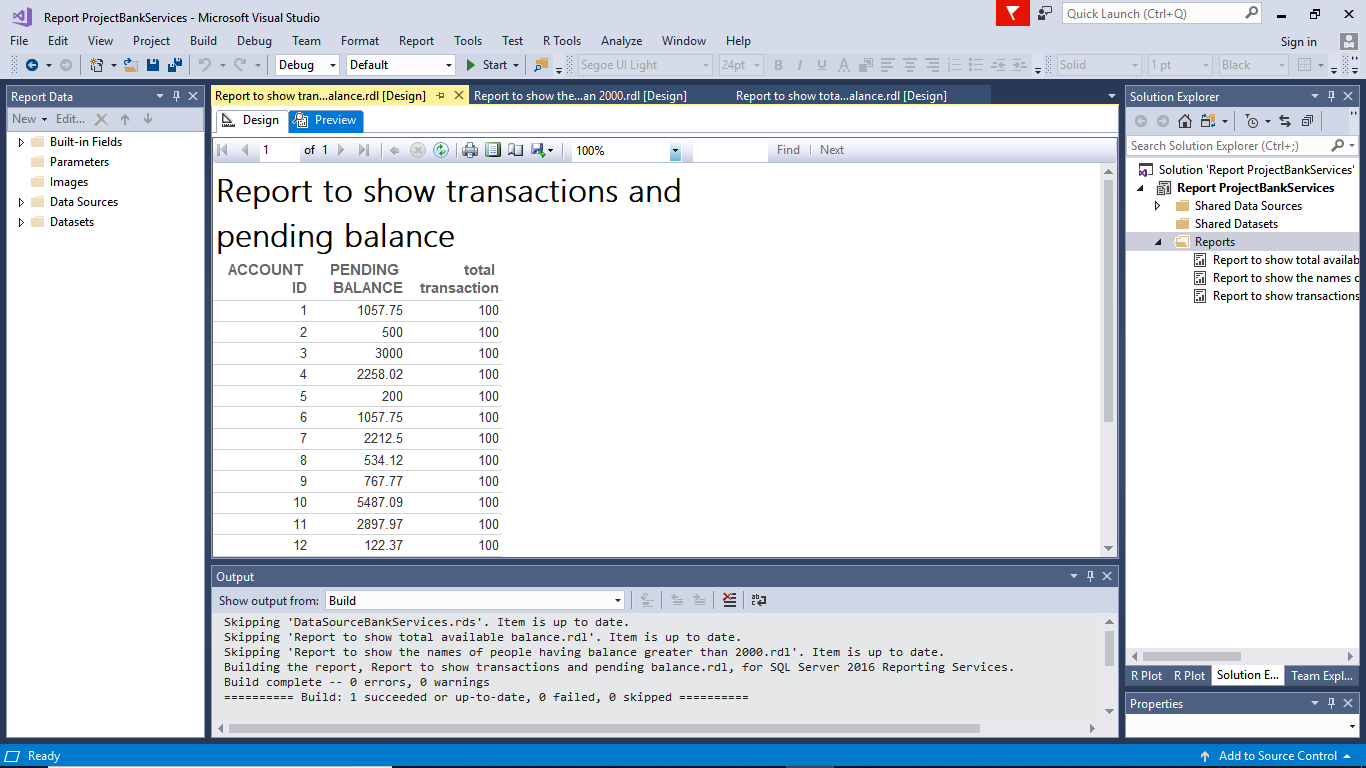
Report to show names of people having balance greater than 2000



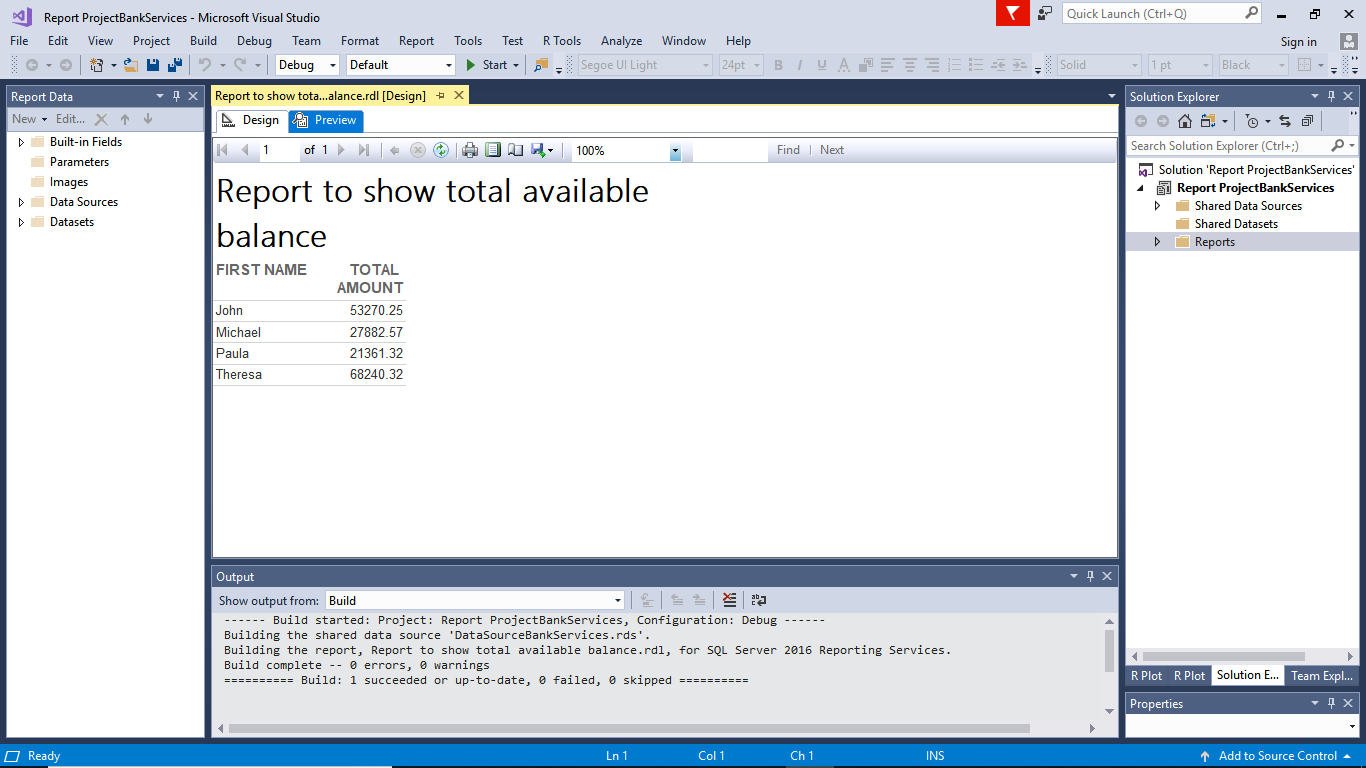
The report showing the number of accounts per customer -



The report showing pending balance after each transaction-

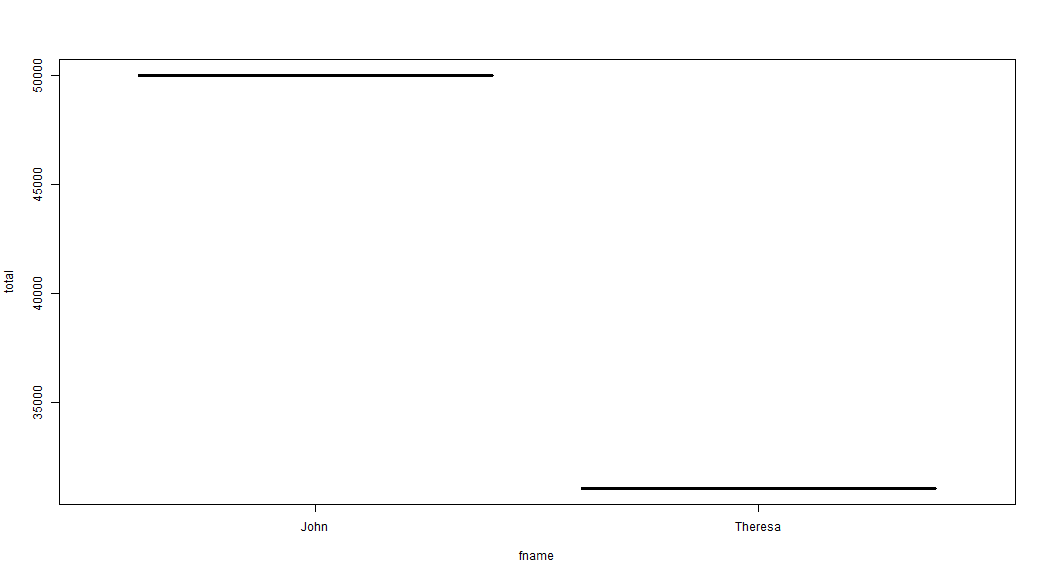


The report showing total available balance-

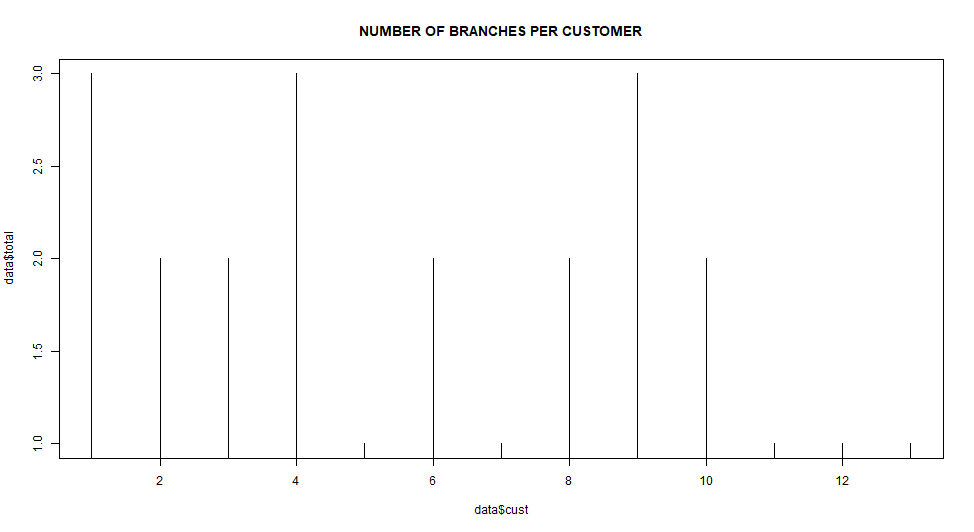


We will then generate the visualization for the report that we have created earlier using the r scripts. We establish a connection to our database and then we write queries and plot the graph

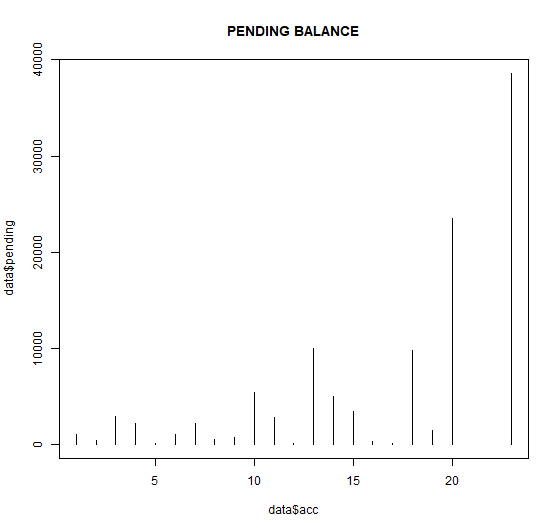
--Visualizing the average balance above 10000



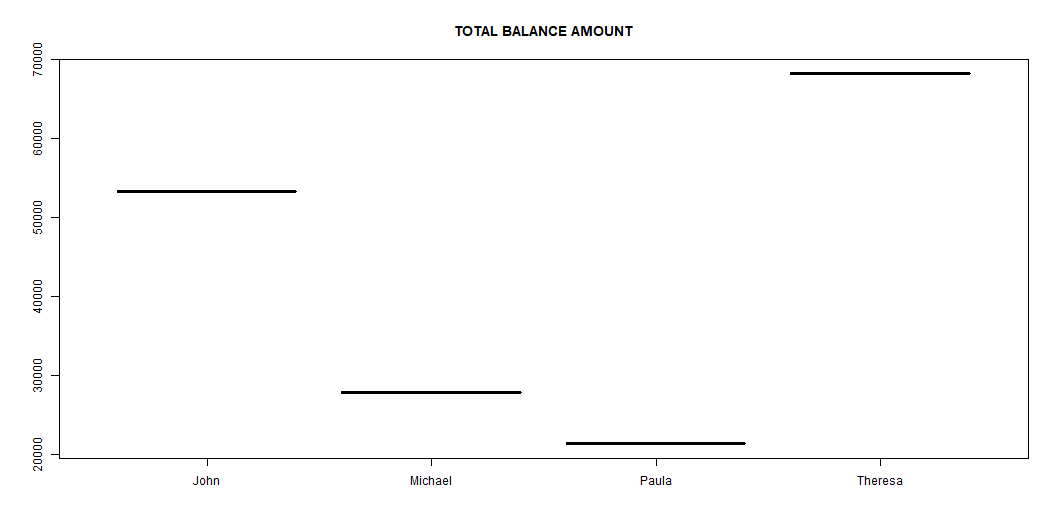
--Visualization of the number of branches per customer



Visualization of the pending balance after the transaction is done



--Visualization of the total amount per person.

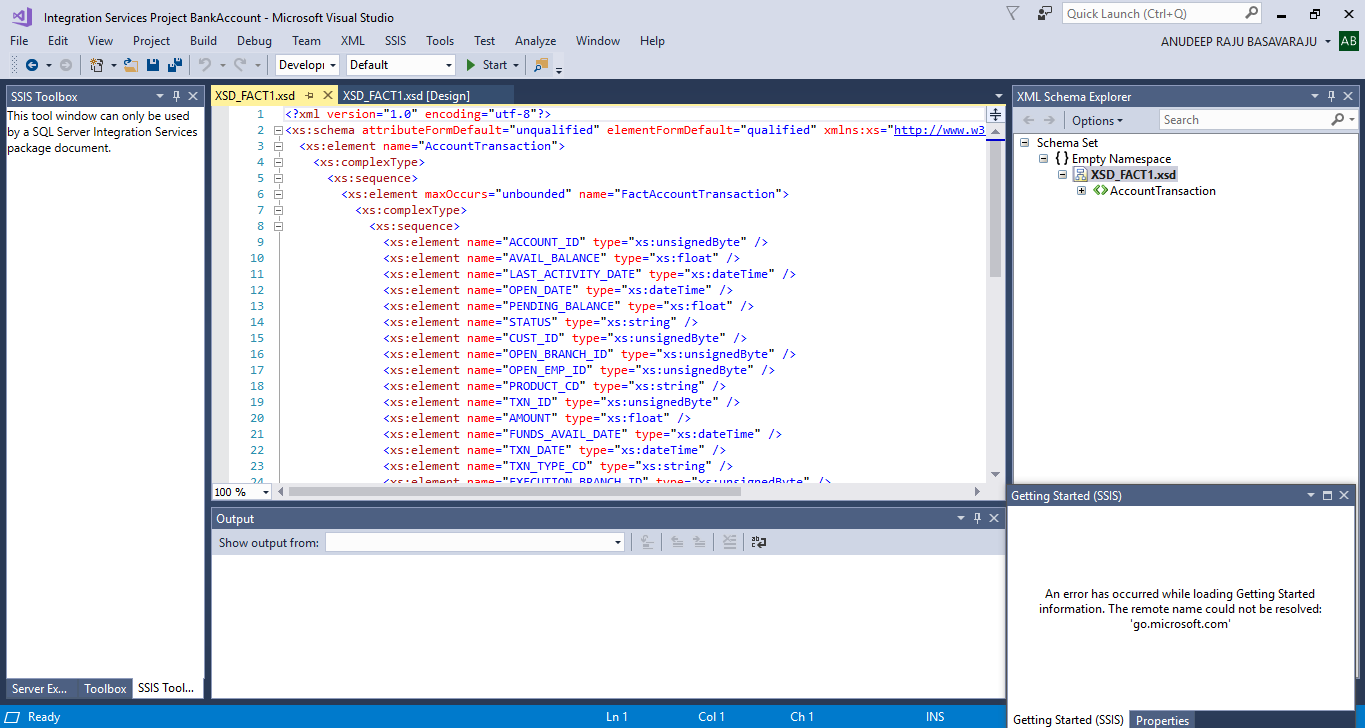


# **5. XML Schema Generation**

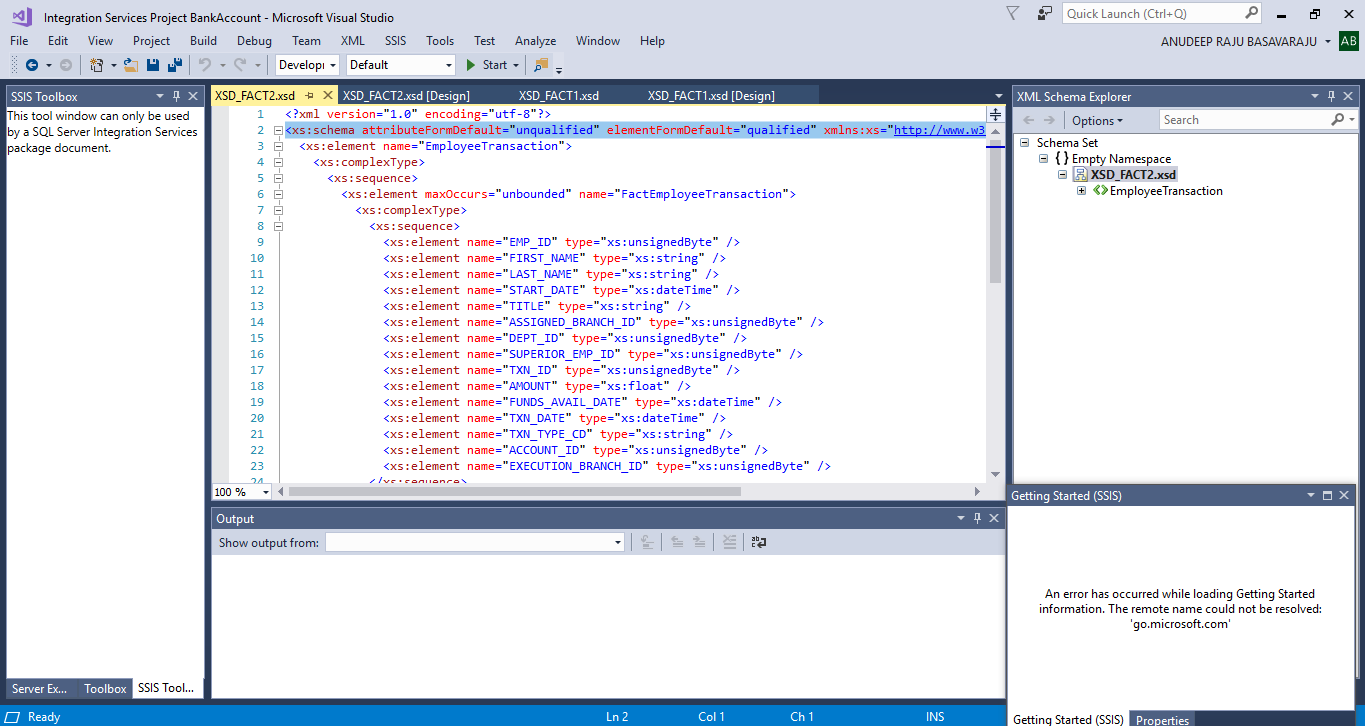
## 5.1 XML and XSD

Once we have data in our fact tables we can run our stored procedure to get data as XML format and using this XML we can generate the Schema for the fact tables which can be used for inserting data in XML format in future.

-- XML Schema is created for the fact table ‘FactAccountTransaction’



-- XML Schema is created for the fact table ‘FactEmployeeTransaction’



--Validating XML schema of ‘FactAccountTransaction’



--Validating XML schema of ‘FactEmployeeTransaction’



# **6 Modelling and Graph Data Models**

## 6.1 graph using Neo4j:

The neo4j is the tool that we are using to create a better visualisation of the data warehouse in terms of labels and nodes and then we create MATCHES between the labels to show the relationship existing between the dimension and fact tables.

We have to set up the neo4j server and start the neo4j browser and import CSV files as labels and then create MATCH.

## 6.2 Importing CSV file:

LOAD CSV WITH HEADERS FROM "file:///dimAccount.csv" AS row

CREATE (n:Br\_Account)

SET n = row {A\_ID:row.ACCOUNT\_ID,

AB:row.AVAIL\_BALANCE,

LAD:row.LAST\_ACTIVITY\_DATE,

OD:row.OPEN\_DATE,

PD:row.PENDING\_BALANCE,

S:row.STATUS,

CID:row.CUST\_ID,

OBID:row.OPEN\_BRANCH\_ID,

OEID:row.OPEN\_EMP\_ID,

PCD:row.PRODUCT\_CD}

return n

LOAD CSV WITH HEADERS FROM "file:///dimBranch.csv" AS row

CREATE (n:Br\_Branch)

SET n = row {B\_ID:row.BRANCH\_ID,

ADD:row.ADDRESS,

City:row.CITY,

Name:row.NAME,

State:row.STATE,

ZC:row.ZIP\_CODE}

return n

LOAD CSV WITH HEADERS FROM "file:///dimCustomer.csv" AS row

CREATE (n:Br\_Customer)

SET n = row {CCID:row.CUST\_ID,

ADDC:row.ADDRESS,

Ccity:row.CITY,

CTC:row.CUST\_TYPE\_CD,

FID:row.FED\_ID,

PC:row.POSTAL\_CODE,

Cstate:row.STATE}

return n

LOAD CSV WITH HEADERS FROM "file:///dimDepartment.csv" AS row

CREATE (n:Br\_Department)

SET n = row {DID:row.DEPT\_ID,

Dname:row.NAME}

return n

LOAD CSV WITH HEADERS FROM "file:///dimEmployee.csv" AS row

CREATE (n:Br\_Employee)

SET n = row{EID:row.EMP\_ID,

Efn:row.FIRST\_NAME,

Eln:row.LAST\_NAME,

Est:row.START\_DATE,

Etitle:row.TITLE,

ABID:row.ASSIGNED\_BRANCH\_ID,

eDID:row.DEPT\_ID,

SEID:row.SUPERIOR\_EMP\_ID}

return n

LOAD CSV WITH HEADERS FROM "file:///dimTransaction.csv" AS row

CREATE (n:Br\_Transaction)

SET n = row{TID:row.TXN\_ID,

AMT:row.AMOUNT,

FAD:row.FUNDS\_AVAIL\_DATE,

TDate:row.TXN\_DATE,

TTCD:row.TXN\_TYPE\_CD,

TAID:row.ACCOUNT\_ID,

TEBID:row.EXECUTION\_BRANCH\_ID,

TTEID:row.TELLER\_EMP\_ID}

return n

LOAD CSV WITH HEADERS FROM "file:///factAccountTransaction.csv" AS row

CREATE (n:Br\_AccountTransaction)

SET n = row

{

FATA\_ID:row.ACCOUNT\_ID,

FATAB:row.AVAIL\_BALANCE,

FATLAD:row.LAST\_ACTIVITY\_DATE,

FATOD:row.OPEN\_DATE,

FATPD:row.PENDING\_BALANCE,

FATS:row.STATUS,

FATCID:row.CUST\_ID,

FATOBID:row.OPEN\_BRANCH\_ID,

FATOEID:row.OPEN\_EMP\_ID,

FATPCD:row.PRODUCT\_CD,

FATTID:row.TXN\_ID,

FATAMT:row.AMOUNT,

FATFAD:row.FUNDS\_AVAIL\_DATE,

FATTDate:row.TXN\_DATE,

FATTTCD:row.TXN\_TYPE\_CD,

FATTAID:row.ACCOUNT\_ID,

FATTEBID:row.EXECUTION\_BRANCH\_ID,

FATTTEID:row.TELLER\_EMP\_ID}

return n

## 6.3 Relation Between Tables:

MATCH (a:Br\_Account),(l:Br\_Customer)

where a.CID = l.CCID

CREATE (a)-[ac:BranchAccountCustomer]->(l)

return a,l

MATCH (a:Br\_Account),(l:Br\_Branch)

where a.OBID = l.B\_ID

CREATE (a)-[ba:BranchAccount]->(l)

RETURN a,l

MATCH (a:Br\_Account),(l:Br\_Employee)

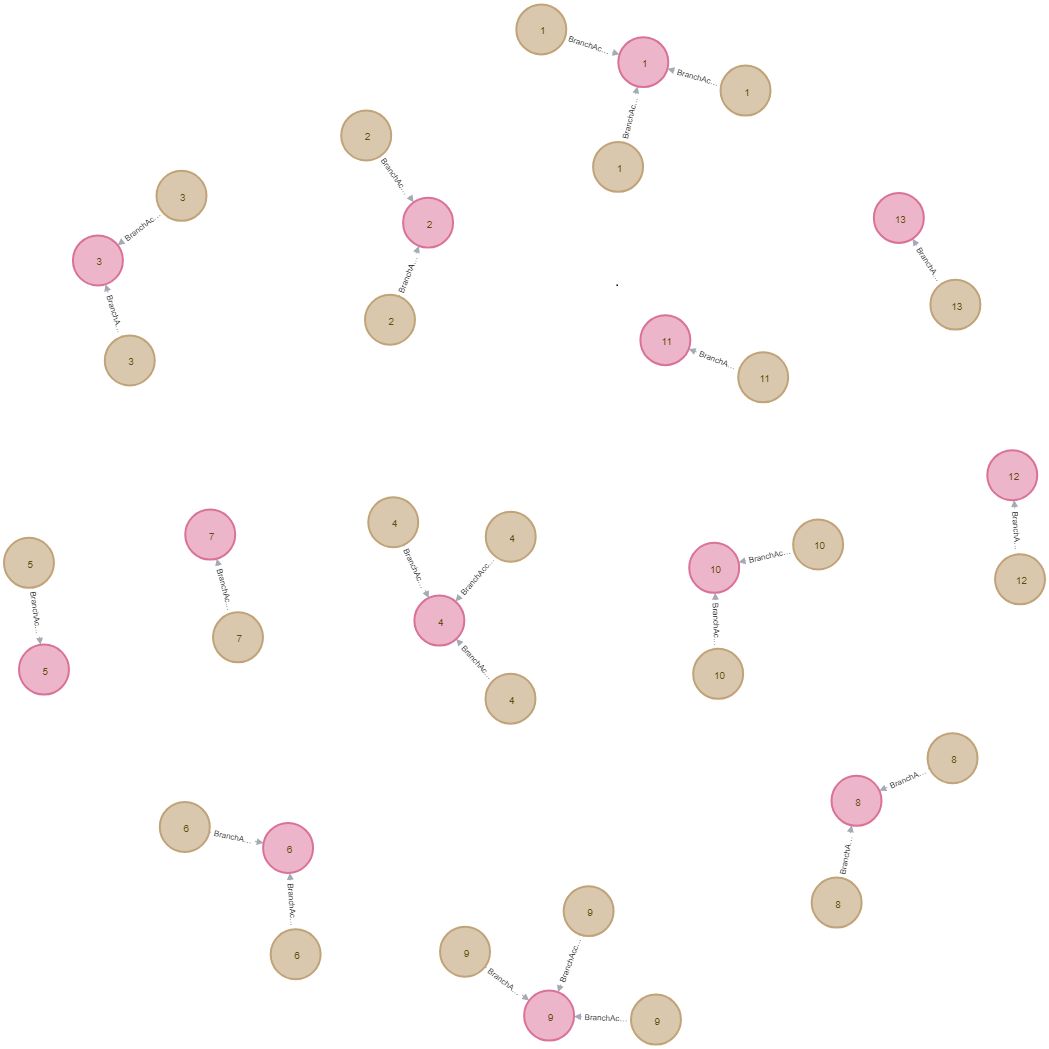
where a.OEID = l.EID

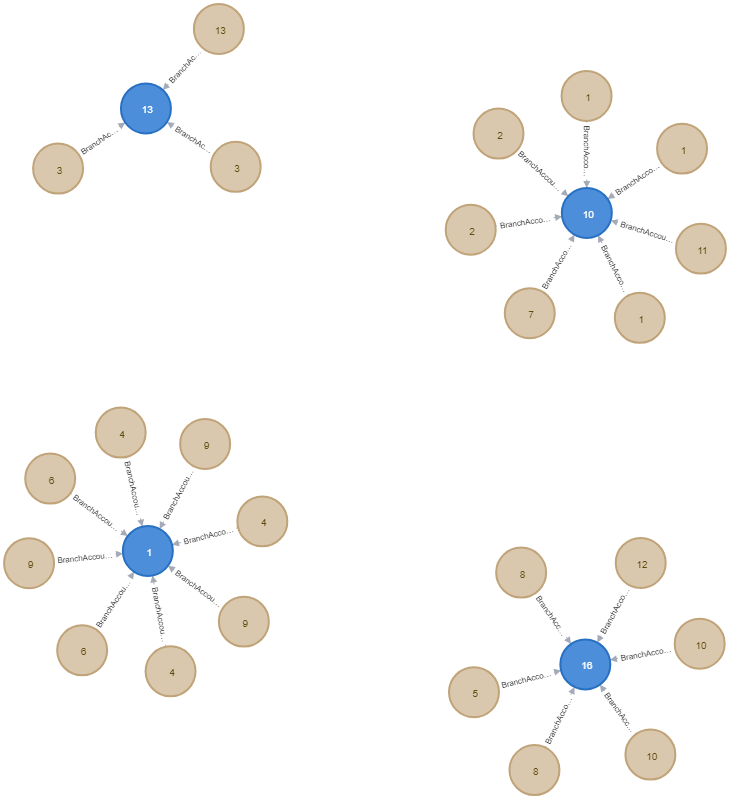
CREATE (a)-[ae:BranchAccountEmployee]->(l)

return a,l

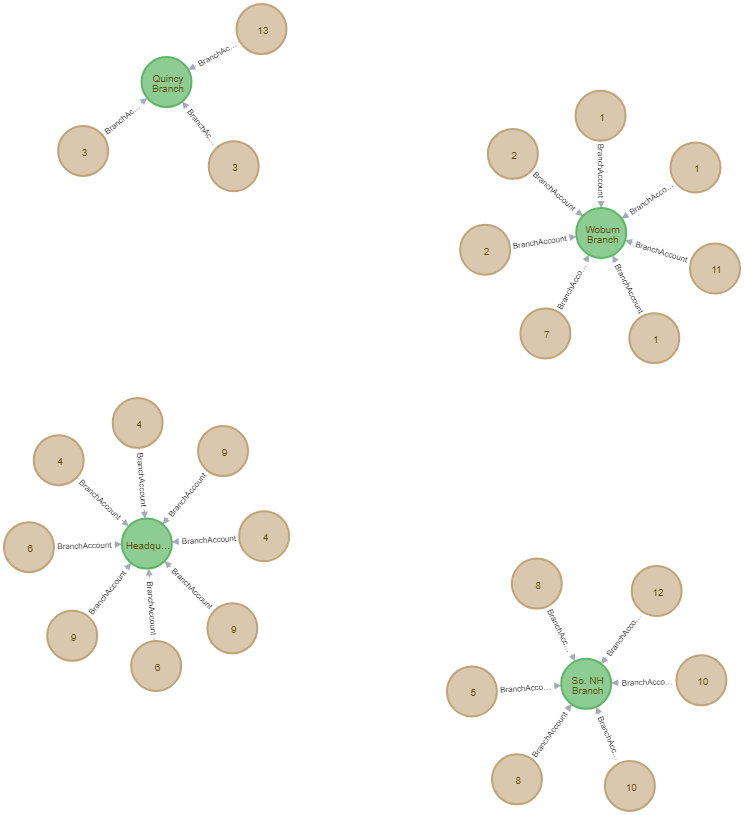
## 6.4 Displaying all nodes related to the fact table:

-- to show details of Account and Customers to which they belong



A graph to show Account and employees related to it

-- to show details of Account and Branch to which they belong



## BIBLIOGRAPHY

1. <https://archive.ics.uci.edu/ml/datasets/Bank%2BMarketing>
2. <https://github.com/sidgvpta/BankDatabases>