Data warehousing

Phase 4

Deployment of star model

Submitted to: Sir Abid Sohail Bhutta

Submitted by: Ume-kalsoom Waheed

Samurah Shafique

Date:

Introduction:

The current phase focuses on the deployment of data warehouse. The phase comprises of the ETL activities, storage and information delivery components of the data warehousing lifecycle. The procedure and background of the project is explained in the succeeding sections.

Dataset:

The project used the Berka dataset based on the data of Czech bank. The data set is available on the following link: <https://www.researchgate.net/deref/http%3A%2F%2Fsorry.vse.cz%2F~berka%2Fchallenge%2Fpkdd1999%2Fdata_berka.zip>

The details are available on the link <http://webpages.uncc.edu/mirsad/itcs6265/group1/domain.html>

The data contains information about clients, accounts, demographics, cards, disposition, loan, bank orders and transactions.

Background:

The history of previous phases is as follows:

Phase I: relational model of the project dated

Phase II: relational database of the project dated

Phase III: dimensional model of the project dated

Development:

Dimensions and facts:

The excel files for the data were modified to meet the required format for the data warehouse. The changes were made to the files of client, account and transactions. The date dimension was populated from SQL script.

Client: the client file lacked gender attribute. The date of birth had month+50 for female and simple month for male. So, if the date(yy-mm-dd) is actually 93-02-21 then it would be 93-52-21 for female client and same for male client. The gender was extracted by applying formulas on the date\_of\_birth attribute.

The month was extracted by MID(cell\_num,3,2)

If month>12 then gender was female else male.

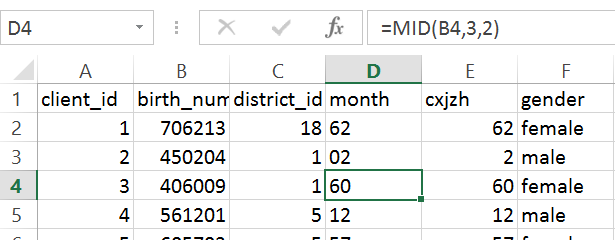


Table 1(edited client.xlsx with gender column)

Account:

For the account dimension, the disposition and card information was merged into a single file. The disp and card were copied as separate sheets in the account workbook and vlookup() function was used to match the ids. The disp sheet was matched with card and then account was matched with disp to get both card and disposition information.

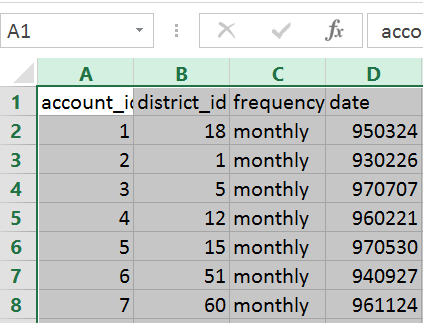


Table 2(actual account.xlsx)

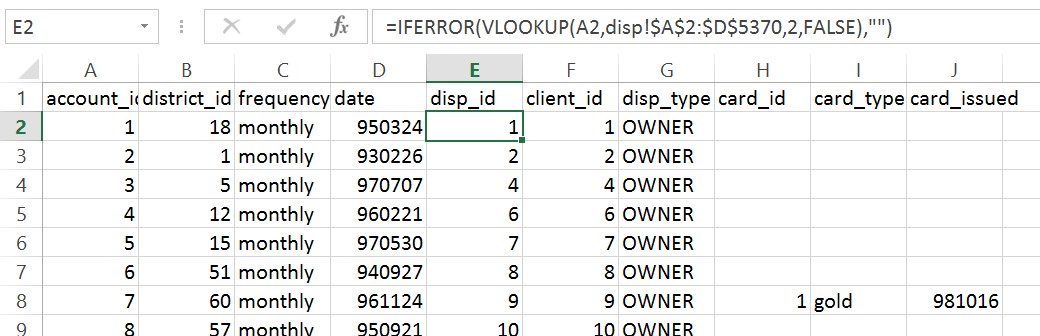


Table 3 (account.xlsx with disposition and card columns)

Transaction:

Similarly the transaction file was compared with account and client files to insert foreign keys for account, client and district.

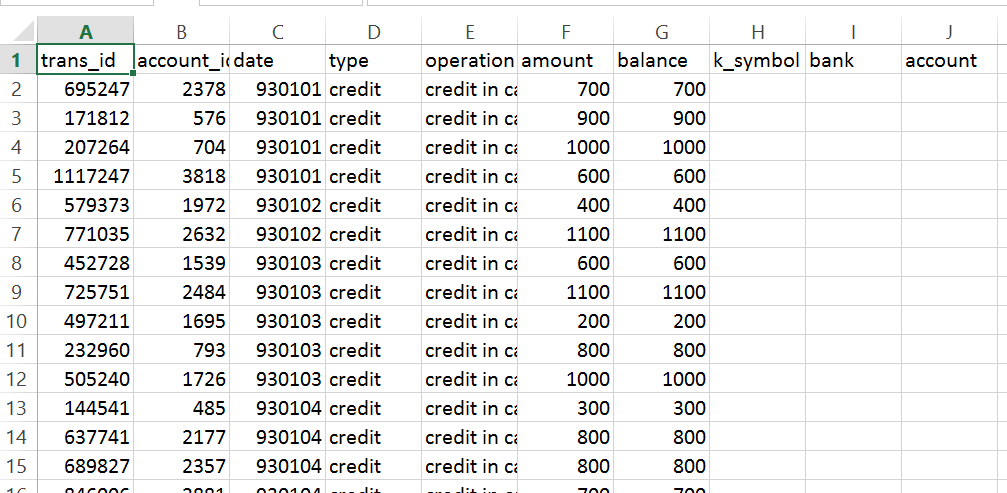


Table 4(actual table transaction)

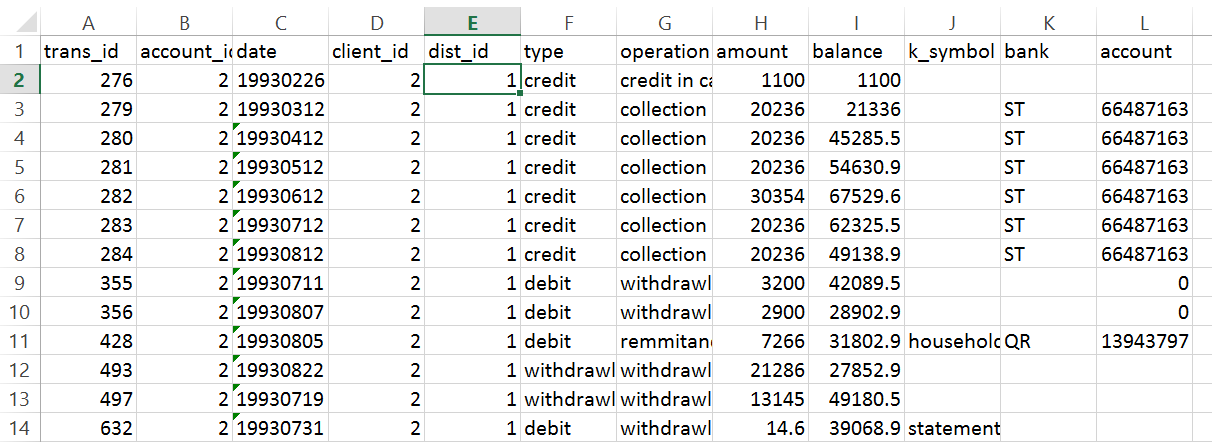


Table 5 (trans.xlsx with foreign keys of dimensions)

Date:

The script for date dimension is as follows:

DECLARE @startdate DATETIME = '01/01/1993'; --mm/dd/yy

declare @enddate DATETIME = '01/01/1999';

declare

@weekofmonth int,

@currentyear int,

@currentmonth int,

@currentquarter int

DECLARE @DayOfWeek TABLE (DOW INT, MonthCount INT, QuarterCount INT, YearCount INT)

INSERT INTO @DayOfWeek VALUES (1, 0, 0, 0);

INSERT INTO @DayOfWeek VALUES (2, 0, 0, 0);

INSERT INTO @DayOfWeek VALUES (3, 0, 0, 0);

INSERT INTO @DayOfWeek VALUES (4, 0, 0, 0);

INSERT INTO @DayOfWeek VALUES (5, 0, 0, 0);

INSERT INTO @DayOfWeek VALUES (6, 0, 0, 0);

INSERT INTO @DayOfWeek VALUES (7, 0, 0, 0);

DECLARE @CurrentDate AS DATETIME = @StartDate

SET @CurrentMonth = DATEPART(MM, @CurrentDate)

SET @CurrentYear = DATEPART(YY, @CurrentDate)

SET @CurrentQuarter = DATEPART(QQ, @CurrentDate)

select @CurrentDate as dat,@CurrentMonth as mont,@CurrentYear as yr,@CurrentQuarter as quart

WHILE @CurrentDate < @EndDate

BEGIN

IF @CurrentMonth != DATEPART(MM, @CurrentDate)

BEGIN

UPDATE @DayOfWeek

SET MonthCount = 0

SET @CurrentMonth = DATEPART(MM, @CurrentDate)

END

IF @CurrentQuarter != DATEPART(QQ, @CurrentDate)

BEGIN

UPDATE @DayOfWeek

SET QuarterCount = 0

SET @CurrentQuarter = DATEPART(QQ, @CurrentDate)

END

IF @CurrentYear != DATEPART(YY, @CurrentDate)

BEGIN

UPDATE @DayOfWeek

SET YearCount = 0

SET @CurrentYear = DATEPART(YY, @CurrentDate)

END

UPDATE @DayOfWeek

SET

MonthCount = MonthCount + 1,

QuarterCount = QuarterCount + 1,

YearCount = YearCount + 1

WHERE DOW = DATEPART(DW, @CurrentDate)

use star;

INSERT INTO date\_dim (datekey,[day],day\_of\_week,type\_of\_day,calendar\_week,calendar\_month,calendar\_quarter,calendar\_year)

SELECT

CONVERT (char(8),@CurrentDate,112) as datekey,

@CurrentDate AS day,

DATEPART(DW, @CurrentDate) AS day\_of\_week,

CASE DATEPART(DW, @CurrentDate)

WHEN 1 THEN 0

WHEN 2 THEN 1

WHEN 3 THEN 1

WHEN 4 THEN 1

WHEN 5 THEN 1

WHEN 6 THEN 1

WHEN 7 THEN 0

END AS type\_of\_day,

DATEPART(WW, @CurrentDate) + 1 - DATEPART(WW, CONVERT(VARCHAR, DATEPART(MM, @CurrentDate)) + '/1/' + CONVERT(VARCHAR, DATEPART(YY, @CurrentDate))) AS WeekOfMonth,

DATEPART(MM, @CurrentDate) AS calendar\_month,

DATEPART(QQ, @CurrentDate) AS calendar\_quarter,

DATEPART(YEAR, @CurrentDate) AS calendar\_year

SET @CurrentDate = DATEADD(DD, 1, @CurrentDate)

END

DECLARE

@dtFiscalYearStart SMALLDATETIME = 'January 01, 1993',

@FiscalYear INT = 1993,

@LastYear INT = 1999,

@FirstLeapYearInPeriod INT = 1996 --keep same

DECLARE

@iTemp INT,

@LeapWeek INT,

@Current DATETIME,

@FiscalDayOfYear INT,

@FiscalWeekOfYear INT,

@FiscalMonth INT,

@FiscalQuarter INT,

@LeapYear INT

DECLARE @LeapTable TABLE (leapyear INT)

DECLARE @tb TABLE(

PeriodDate DATETIME,

[FiscalDayOfYear] VARCHAR(3),

[FiscalWeekOfYear] VARCHAR(3),

[FiscalMonth] VARCHAR(2),

[FiscalQuarter] VARCHAR(1),

[FiscalYear] VARCHAR(4))

SET @LeapYear = @FirstLeapYearInPeriod

WHILE (@LeapYear < @LastYear)

BEGIN

INSERT INTO @leapTable VALUES (@LeapYear)

SET @LeapYear = @LeapYear + 5

END

SET @CurrentDate = @dtFiscalYearStart

SET @FiscalDayOfYear = 1

SET @FiscalWeekOfYear = 1

SET @FiscalMonth = 1

SET @FiscalQuarter = 1

SET @FiscalWeekOfYear = 1

IF (EXISTS (SELECT \* FROM @LeapTable WHERE @FiscalYear = leapyear))

BEGIN

SET @LeapWeek = 1

END

ELSE

BEGIN

SET @LeapWeek = 0

END

WHILE (DATEPART(yy,@CurrentDate) <= @LastYear)

BEGIN

SELECT @FiscalMonth = CASE

WHEN @FiscalWeekOfYear BETWEEN 1 AND 4 THEN 1 /\*4 weeks\*/

WHEN @FiscalWeekOfYear BETWEEN 5 AND 9 THEN 2 /\*5 weeks\*/

WHEN @FiscalWeekOfYear BETWEEN 10 AND 13 THEN 3 /\*4 weeks\*/

WHEN @FiscalWeekOfYear BETWEEN 14 AND 17 THEN 4 /\*4 weeks\*/

WHEN @FiscalWeekOfYear BETWEEN 18 AND 22 THEN 5 /\*5 weeks\*/

WHEN @FiscalWeekOfYear BETWEEN 23 AND 26 THEN 6 /\*4 weeks\*/

WHEN @FiscalWeekOfYear BETWEEN 27 AND 30 THEN 7 /\*4 weeks\*/

WHEN @FiscalWeekOfYear BETWEEN 31 AND 35 THEN 8 /\*5 weeks\*/

WHEN @FiscalWeekOfYear BETWEEN 36 AND 39 THEN 9 /\*4 weeks\*/

WHEN @FiscalWeekOfYear BETWEEN 40 AND 43 THEN 10 /\*4 weeks\*/

WHEN @FiscalWeekOfYear BETWEEN 44 AND (48+@LeapWeek) THEN 11 WHEN @FiscalWeekOfYear BETWEEN (49+@LeapWeek) AND (52+@LeapWeek) THEN 12

END

SELECT @FiscalQuarter = CASE

WHEN @FiscalMonth BETWEEN 1 AND 3 THEN 1

WHEN @FiscalMonth BETWEEN 4 AND 6 THEN 2

WHEN @FiscalMonth BETWEEN 7 AND 9 THEN 3

WHEN @FiscalMonth BETWEEN 10 AND 12 THEN 4

END

INSERT INTO @tb (PeriodDate, FiscalWeekOfYear,

fiscalMonth, FiscalQuarter, FiscalYear) VALUES

(@CurrentDate, @FiscalWeekOfYear, @FiscalMonth,

@FiscalQuarter, @FiscalYear)

SET @CurrentDate = DATEADD(dd, 1, @CurrentDate)

SET @FiscalDayOfYear = @FiscalDayOfYear + 1

SET @FiscalWeekOfYear = ((@FiscalDayOfYear-1) / 7) + 1

IF (@FiscalWeekOfYear > (52+@LeapWeek))

BEGIN

SET @FiscalDayOfYear = 1

SET @FiscalWeekOfYear = 1

SET @FiscalYear = @FiscalYear + 1

IF ( EXISTS (SELECT \* FROM @leapTable WHERE @FiscalYear = leapyear))

BEGIN

SET @LeapWeek = 1

END

ELSE

BEGIN

SET @LeapWeek = 0

END

END

END

UPDATE date\_dim

SET

fiscal\_week = a.FiscalWeekOfYear

, fiscal\_month = a.FiscalMonth

, fiscal\_quarter = a.FiscalQuarter

, fiscal\_year = a.FiscalYear

FROM @tb a

INNER JOIN date\_dim b ON a.PeriodDate = b.[day]

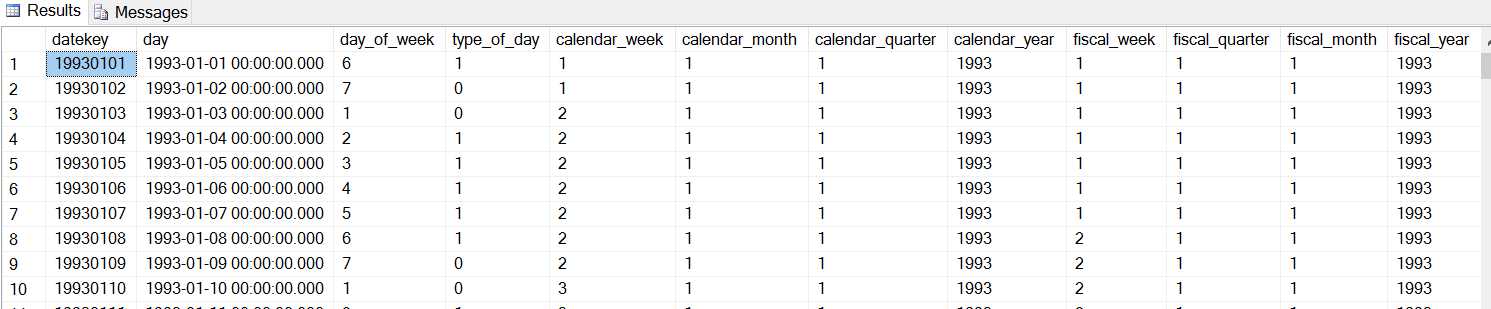


Table 6 (date dimension)

Database:

The phase IV is developed by creating a new database based on the dimensional star model of the project. The code for phase III was used to populate data from the excel files into the database.The database is named as “star” and contains the following tables:

Fact\_trans, dim\_account, dim\_client, dim\_dist, dim\_date.

The screenshots of tables are given below:

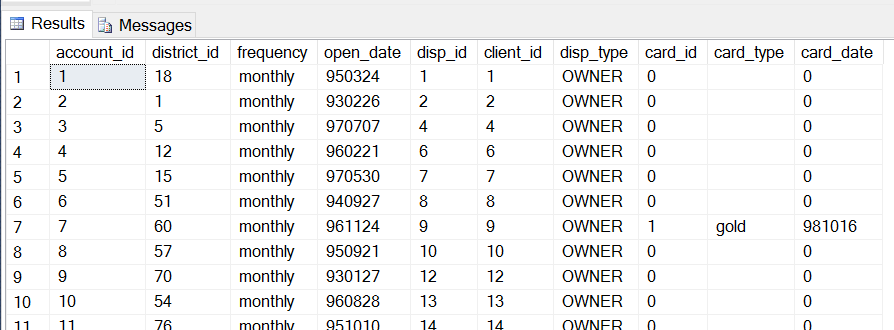


Table 7 (dim\_account)

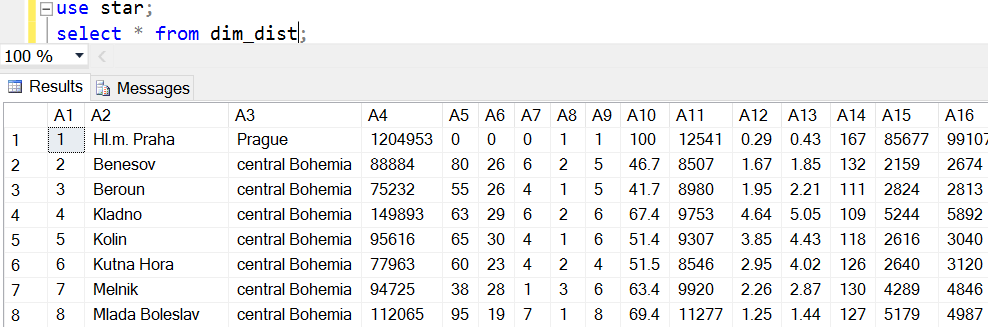


Table 8 (dim\_dist)

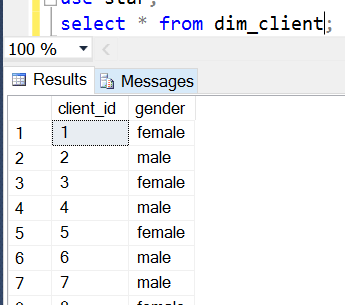


Table 9(dim\_client)

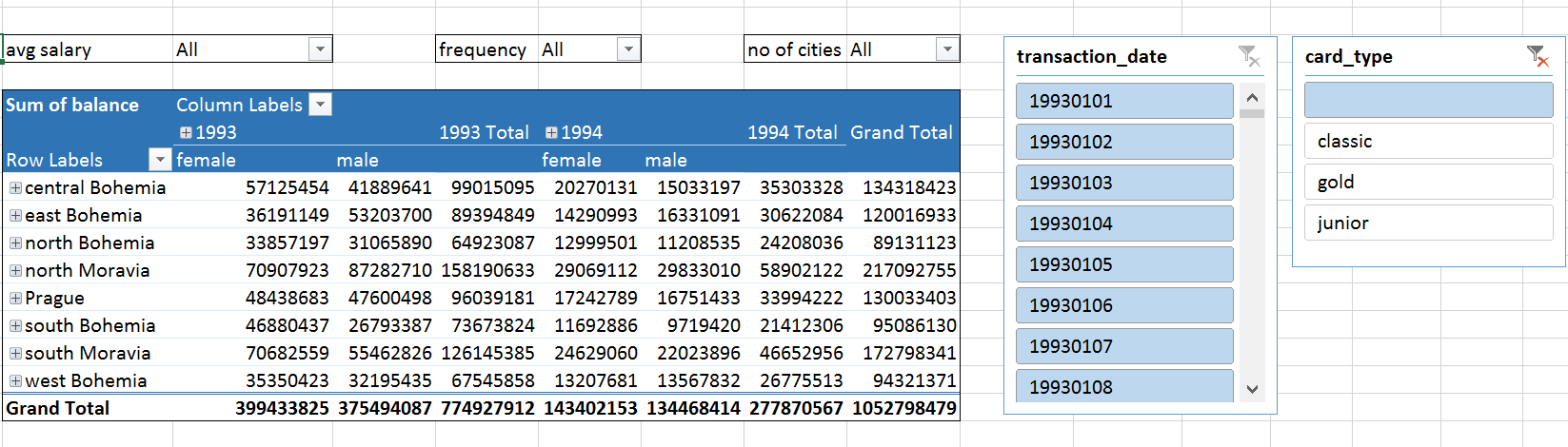
Views:

The view is provided using powerpivot add-in of excel. The database was imported into excel.

Dim\_acc

Dim\_date

Dim\_client



Dim\_dist

The charts produced are:

