Sri Lanka Institute of Information Technology

Assignment I

Data Warehouse & Business Intelligence

2021

Submitted by:

K.R.Wickramasinghe

IT19050218

Y3S1.04 (DS)

Contents

Data set selection & Preparation 3

Scenario and Data set preparation 3

ER diagram of sources 3

High-level Solution Architecture 4

Staging area details 4

Data warehouse design 5

Data warehouse table design 6

Loading data into staging from sources 7

Loading data to Data warehouse from staging 8

- Transform and Load Property Data (Dim_Property) 9
  - ETL task will replace null values in Building Area column with 0 9
  - ETL task will replace null values in Council Area column with “N” 9
- Transform and Load Seller Data (Dim_SellerDetails) 10
  - When address in the staging table is updated 11
- Transform and Load Sold Fact Table (Fact_Sold) 11
  - ETL task will calculate final cost of the by using this equation. 12

Stored procedures and SQL statements 13

- Stored procedure to check update or insert in Dim_Property table 13
- Date dimension code 14

Data Profiling on staging tables 18

Data set selection & Preparation

The selected data source is a collection of transactional data available in kaggle. Which represent house selling details of the Melbourne city. The link to the source data set is mentioned below.

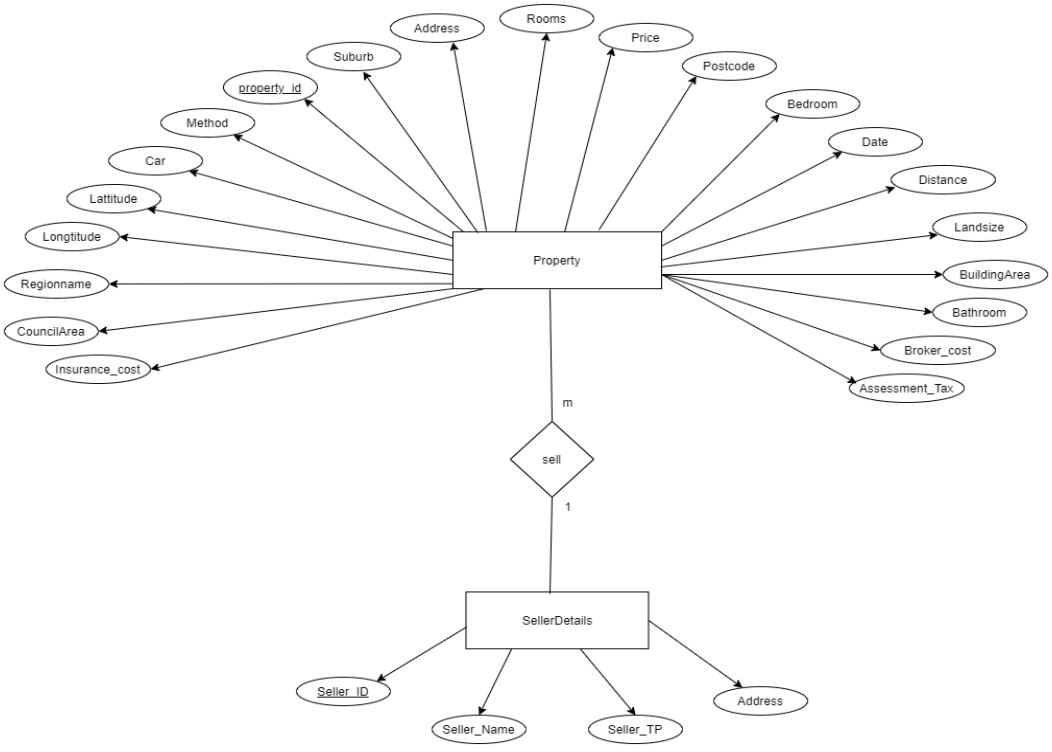
https://www.kaggle.com/dansbecker/melbourne-housing-snapshot

Scenario and Data set preparation

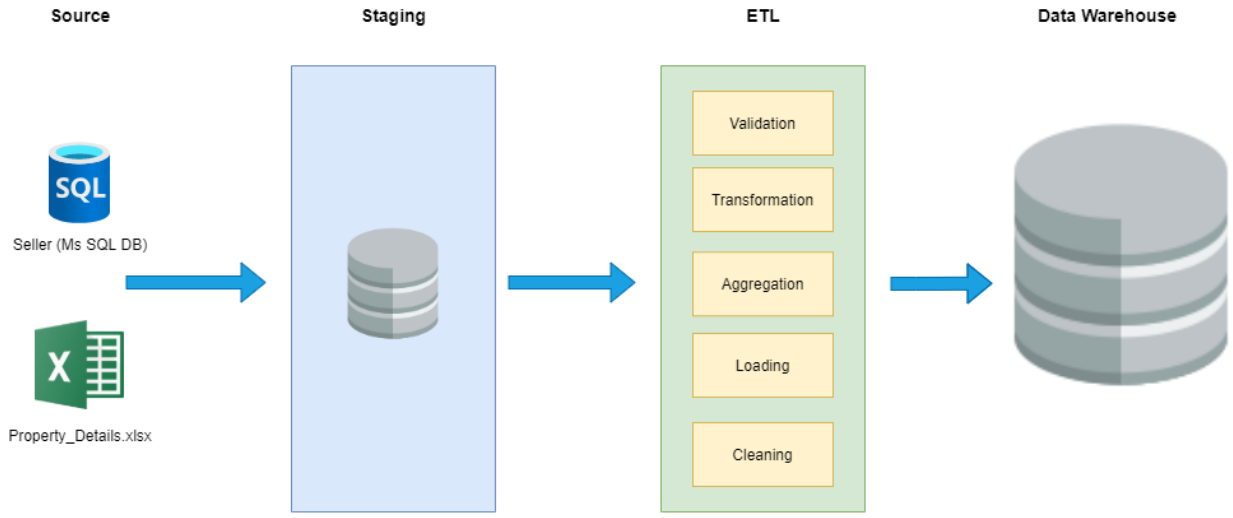
This data set represent house selling details of the Melbourne city. It contains the house details and the seller's details of the sellers who sold those properties. One csv file was divided into 2 parts as described below.

Source	Source Type	Object Name	Schema	Object Type	Description
Seller	SQL Database	sales	dbo	Table	All the seller details included
Property_Details	xlsx file			flat file	All the property details included

ER diagram of sources.



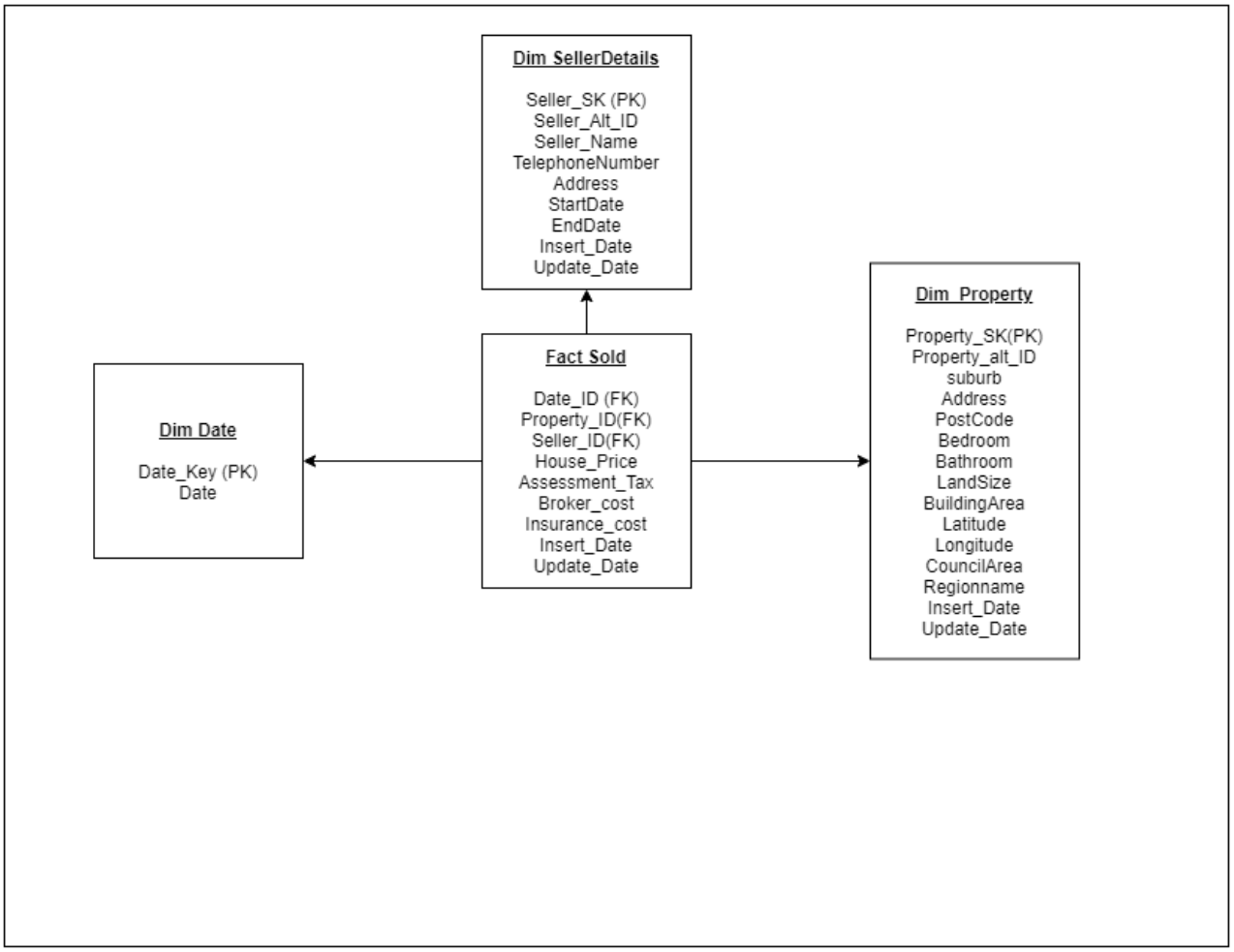
High-level Solution Architecture



Staging area details

Table Name	Column Name	Data Type	Description
Property	property_id	int	property id
	Suburb	nvarchar(255)	residential area name
	Address	nvarchar(255)	property address
	Rooms	int	number of rooms in the house
	Type	nvarchar(255)	house type
	Price	float	price of the house
	Method	nvarchar(255)	sold method
	Date	datetime	sold date
	Distance	float	distance from capital
	Postcode	int	postal code
	Bedroom	int	number of bedrooms in the house
	Bathroom	int	numbar of bathrooms in the house
	Car	int	number of carspots
	Landsize	int	size of the land
	BuildingArea	int	area of the building
	CouncilArea	nvarchar(255)	governing council for the area
	Lattitude	float	Lattitude
	Longtitude	float	Longtitude
	Regionname	nvarchar(255)	regional name
	Assessment_Tax	float	assessment Tax
	Broker_cost	float	broker cost
	Insurance_cost	float	insurance cost
	Seller_ID	int	seller ID
SellerDetails	Seller_ID	nvarchar(255)	seller id
	Seller_Name	nvarchar(255)	seller name
	Seller_TP	nvarchar(255)	seller telephone number
	Address	nvarchar(255)	seller address

Data warehouse design



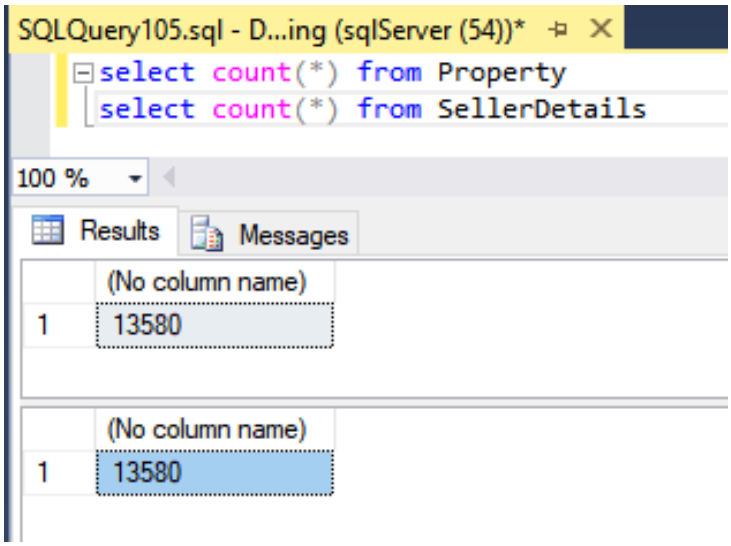
Data warehouse table design

Dimention Name	Dimention Attributes	Derived Attribute	Data Type	Key column	Derived Logic	Description
Dim_Property	Property_SK	no	int	Primary key	Auto increment	
	Property_Alt_ID	no	int			
	Suburb	no	nvarchar(255)			
	Address	no	nvarchar(255)			
	PostCode	no	int			
	Bedroom	no	int			
	Bathroom	no	int			
	LandSize	no	float			
	BuildingArea	no	float			
	CouncilArea	no	nvarchar(255)			
	Region_Name	no	nvarchar(255)			
	Insert_Date	yes	datetime		System Datetime	
	Update_Date	yes	datetime		System Datetime	
Dim_SellerDetails	Seller_SK	no	int			
	Seller_Alt_ID	no	nvarchar(255)			
	Seller_Name	no	nvarchar(255)			
	Telephone_Number	no	nvarchar(255)			
	Address	no	nvarchar(255)			
	Start_Date	yes	datetime		System Datetime (Historical)	
	End_Date	yes	datetime		System Datetime (Historical)	
	Insert_Date	yes	datetime		System Datetime	
	Update_Date	yes	datetime		System Datetime	
DimDate	DateKey		int	Primary key		static table
	Date		datetime			static table
	FullDateUK		char(10)			static table
	FullDateUSA		char(10)			static table
	DayOfMonth		varchar(4)			static table
	DaySuffix		varchar(9)			static table
	DayName		varchar(9)			static table
	More...					
Fact_Sold	Seller_ID	no	int	foreign key		
	Property_ID	no	int	foreign key		
	Date_ID	no	int	foreign key		
	House_Price	no	float			
	Assessment_Tax	no	float			
	Broker_Cost	no	float			
	Insurance_Cost	no	float			
	Final_Cost	yes	float		House_Price+Assessment_Tax+Broker_Cost+Insuarence_Cost	
	Insert_Date	yes	datetime		System Datetime	
	Update_Date	yes	datetime		System Datetime	

Loading data into staging from sources

Extract property information and loading into staging.

Extract seller details information and loading into staging.



All the data available in staging

Loading data to Data warehouse from staging

Transform and Load Property Data (Dim_Property)

ETL task will replace null values in Building Area column with 0

12339	Kew	57 Derby St	4	h	1690000	S	2020-11-10 00:00:00.000	5.4	3101	4	2	4	350	168	NULL	-37.80...	145.03733

Property_SK	Property_Alt_ID	Suburb	Address	PostCode	Bedroom	Bathroom	LandSize	BuildingArea	CouncilArea	Region_Name	Insert_Date	Update_Date
93820	12339	Kew	57 Derby St	3101	4	2	350	168	N	Southern Metropolit...	2021-05-13 07:57:19.210	2021-05-13 07:57:19.210

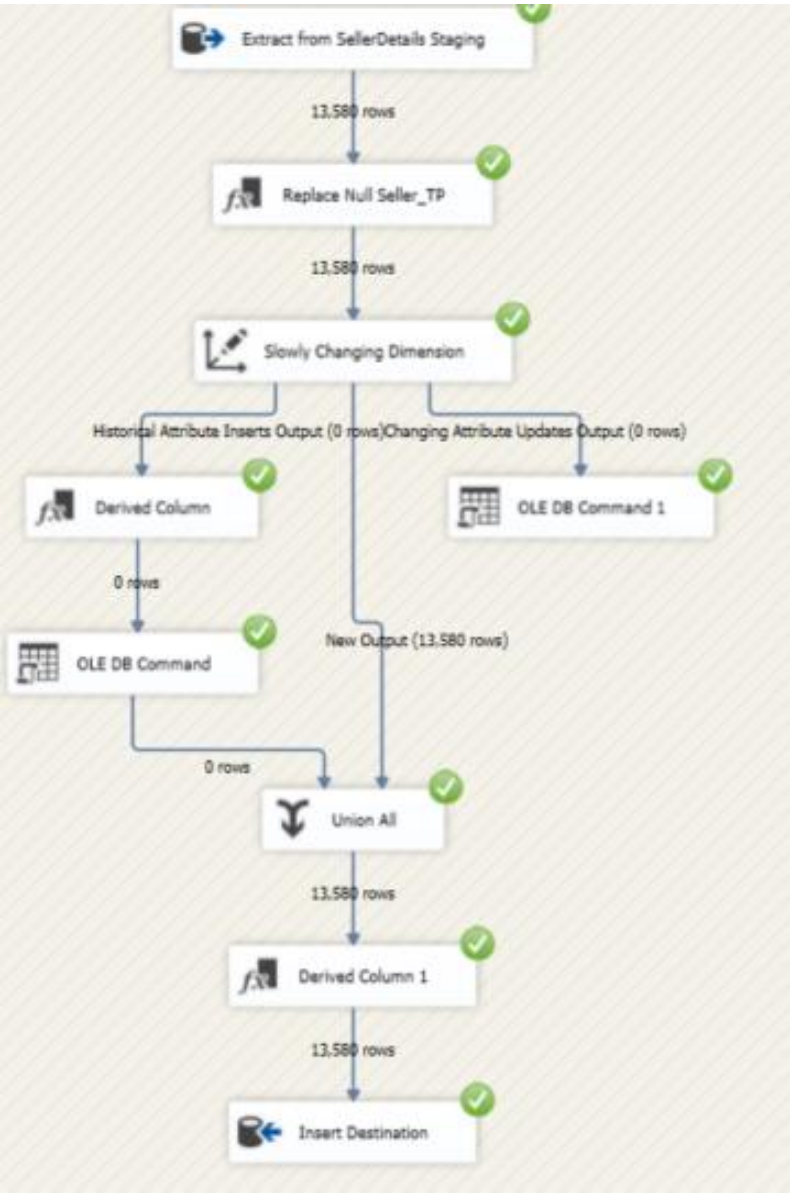
ETL task will replace null values in Council Area column with “N”

property_id	Suburb	Address	Rooms	Type	Price	Method	Date	Distance	Postcode	Bedroom	Bathroom	Car	Landsize	BuildingArea	CouncilArea	Lattitude	Longtitude
0	Abbotsford	85 Turner St	2	h	1480000	S	2019-09-12 00:00:00.000	2.5	3067	2	1	1	202	NULL	Yarra	-37.7996	144.9984

Property_SK	Property_Alt_ID	Suburb	Address	PostCode	Bedroom	Bathroom	LandSize	BuildingArea	CouncilArea	Region_Name	Insert_Date	Update_Date
81481	0	Abbotsford	85 Turner St	3067	2	1	202	0	Yarra	Northern Metropolitan	2021-05-13 07:56:18.117	2021-05-13 07:56:18.117

Transform and Load Seller Data (Dim_SellerDetails)

- Slowly changing dimension
- Address- Historical attribute
- Telephone_Number- changing attribute



When address in the staging table is updated

SQLQuery2.sql - DES...ing (sqlServer (53))*

```
update SellerDetails set Address = '2 Acom Place' WHERE Seller_ID = 'S-35100000';
```

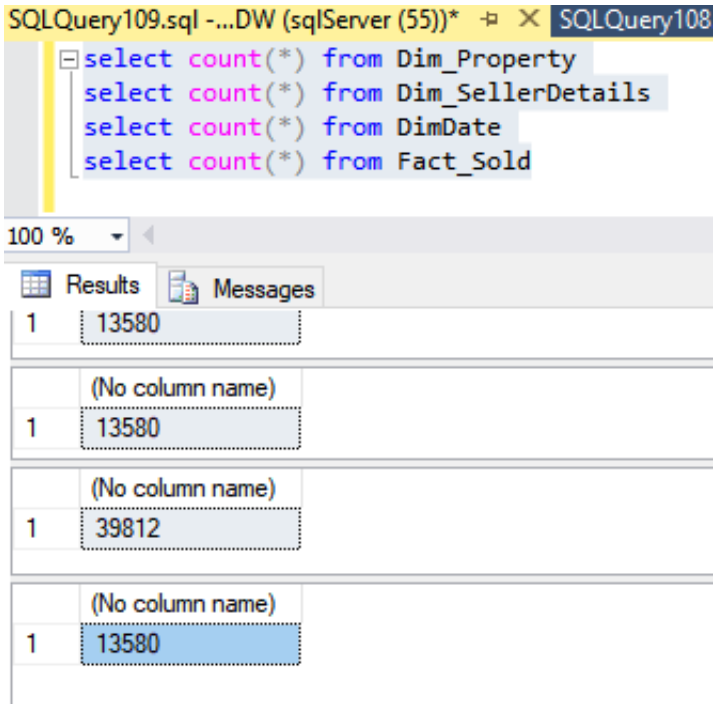
Seller_SK	Seller_Alt_ID	Seller_Name	Telephone_Number	Address	Start_Date	End_Date	Insert_Date	Update_Date
81481	S-35100000	Emma	866-410-0458	1 Acom Place	2021-05-13 07:56:16.000	2021-05-14 00:59:20.000	2021-05-13 07:57:28.583	2021-05-13 07:57:28.583
95061	S-35100000	Emma	866-410-0458	2 Acom Place	2021-05-14 00:59:20.000	NULL	2021-05-14 00:59:21.907	2021-05-14 00:59:21.907

Transform and Load Sold Fact Table (Fact_Sold)

ETL task will calculate final cost of the by using this equation.

Final_Cost = (House_Price + Assessment_Tax + Broker_Cost + Insurance_Cost)

Seller_ID	Property_ID	Date_ID	House_Price	Assessment_Tax	Broker_Cost	Insurance_Cost	Final_Cost	Insert_Date	Update_Date
81481	81481	20190912	1480000	14800	29600	7400	1531800	2021-05-13 07:57:38.427	2021-05-13 07:57:38.427



All the data available in data warehouse

Stored procedures and SQL statements

Stored procedure to check update or insert in Dim_Property table

```
CREATE PROCEDURE dbo.UpdateDim_Property
@property_id int,
@Suburb nvarchar(Max),
@Address nvarchar(MAX),
@Postcode int,
@Bedroom int,
@Bathroom int,
@Landsize float,
@BuildingArea float,
@CouncilArea nvarchar(MAX),
@Regionname nvarchar(MAX)
AS
BEGIN
if not exists (select Property_SK
from dbo.Dim_Property
where Property_Alt_ID = @property_id)
BEGIN
insert into dbo.Dim_Property
(Property_Alt_ID,Suburb,Address,PostCode,Bedroom,Bathroom,LandSize,BuildingArea,CouncilAr
ea,Region_Name,Insert_Date, Update_Date)
values
(@property_id, @Suburb, @Address, @Postcode,@Bedroom,@Bathroom,@Landsize
,@BuildingArea,@CouncilArea,@Regionname,GETDATE(),GETDATE())
END;
if exists (select Property_SK
from dbo.Dim_Property
where Property_Alt_ID = @property_id)
BEGIN
update dbo.Dim_Property
set Property_Alt_ID = @property_id,
Suburb = @Suburb,
Address = @Address,
PostCode = @Postcode,
Bedroom = @Bedroom,
Bathroom = @Bathroom,
LandSize = @Landsize,
BuildingArea = @BuildingArea,
CouncilArea = @CouncilArea,
Region_Name = @Regionname,
Insert_Date = GETDATE(),
Update_Date = GETDATE()
where Property_Alt_ID = @property_id
END;
END;
```

Date dimension code

```
CREATE TABLE   [dbo].[DimDate]
       (       [DateKey] INT primary key,
               [Date] DATETIME,
               [FullDateUK] CHAR(10), -- Date in dd-MM-yyyy format
               [FullDateUSA] CHAR(10),-- Date in MM-dd-yyyy format
               [DayOfMonth] VARCHAR(2), -- Field will hold day number of Month
               [DaySuffix] VARCHAR(4), -- Apply suffix as 1st, 2nd ,3rd etc
               [DayName] VARCHAR(9), -- Contains name of the day, Sunday, Monday
               [DayOfWeekUSA] CHAR(1),-- First Day Sunday=1 and Saturday=7
               [DayOfWeekUK] CHAR(1),-- First Day Monday=1 and Sunday=7
               [DayOfWeekInMonth] VARCHAR(2), --1st Monday or 2nd Monday in Month
               [DayOfWeekInYear] VARCHAR(2),
               [DayOfQuarter] VARCHAR(3),
               [DayOfYear] VARCHAR(3),
               [WeekOfMonth] VARCHAR(1),-- Week Number of Month
               [WeekOfQuarter] VARCHAR(2), --Week Number of the Quarter
               [WeekOfYear] VARCHAR(2),--Week Number of the Year
               [Month] VARCHAR(2), --Number of the Month 1 to 12
               [MonthName] VARCHAR(9),--January, February etc
               [MonthOfQuarter] VARCHAR(2),-- Month Number belongs to Quarter
               [Quarter] CHAR(1),
               [QuarterName] VARCHAR(9),--First,Second..
               [Year] CHAR(4),-- Year value of Date stored in Row
               [YearName] CHAR(7), --CY 2012,CY 2013
               [MonthYear] CHAR(10), --Jan-2013,Feb-2013
               [MMYYYY] CHAR(6),
               [FirstDayOfMonth] DATE,
               [LastDayOfMonth] DATE,
               [FirstDayOfQuarter] DATE,
               [LastDayOfQuarter] DATE,
               [FirstDayOfYear] DATE,
               [LastDayOfYear] DATE,
               [IsHolidaySL] BIT,-- Flag 1=National Holiday, 0-No National Holiday
               [IsWeekday] BIT,-- 0=Week End ,1=Week Day
               [HolidaySL] VARCHAR(50),--Name of Holiday in US
               [isCurrentDay] int, -- Current day=1 else = 0
               [isDataAvailable] int, -- data available for the day = 1, no data available for
the day = 0
               [isLatestDataAvailable] int
       )
GO


/*********************************************************************************************/
--Specify Start Date and End date here
--Value of Start Date Must be Less than Your End Date

DECLARE @StartDate DATETIME = '01/01/1990' --Starting value of Date Range
DECLARE @EndDate DATETIME = '01/01/2099' --End Value of Date Range

--Temporary Variables To Hold the Values During Processing of Each Date of Year
DECLARE
       @DayOfWeekInMonth INT,
       @DayOfWeekInYear INT,
       @DayOfQuarter INT,
       @WeekOfMonth INT,
       @CurrentYear INT,
```

```
        @CurrentMonth INT,
        @CurrentQuarter INT

/*Table Data type to store the day of week count for the month and year*/
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)

--Extract and assign various parts of Values from Current Date to Variable

DECLARE @CurrentDate AS DATETIME = @StartDate
SET @CurrentMonth = DATEPART(MM, @CurrentDate)
SET @CurrentYear = DATEPART(YY, @CurrentDate)
SET @CurrentQuarter = DATEPART(QQ, @CurrentDate)

/********************************************************************************************/
--Proceed only if Start Date(Current date ) is less than End date you specified above

WHILE @CurrentDate < @EndDate
BEGIN

/*Begin day of week logic*/

         /*Check for Change in Month of the Current date if Month changed then
          Change variable value*/
        IF @CurrentMonth != DATEPART(MM, @CurrentDate)
        BEGIN
               UPDATE @DayOfWeek
               SET MonthCount = 0
               SET @CurrentMonth = DATEPART(MM, @CurrentDate)
        END

        /* Check for Change in Quarter of the Current date if Quarter changed then change
         Variable value*/

        IF @CurrentQuarter != DATEPART(QQ, @CurrentDate)
        BEGIN
               UPDATE @DayOfWeek
               SET QuarterCount = 0
               SET @CurrentQuarter = DATEPART(QQ, @CurrentDate)
        END

        /* Check for Change in Year of the Current date if Year changed then change
         Variable value*/


        IF @CurrentYear != DATEPART(YY, @CurrentDate)
        BEGIN
               UPDATE @DayOfWeek
               SET YearCount = 0
               SET @CurrentYear = DATEPART(YY, @CurrentDate)
        END

        -- Set values in table data type created above from variables
```

```
        UPDATE @DayOfWeek
        SET
                MonthCount = MonthCount + 1,
                QuarterCount = QuarterCount + 1,
                YearCount = YearCount + 1
        WHERE DOW = DATEPART(DW, @CurrentDate)

        SELECT
                @DayOfWeekInMonth = MonthCount,
                @DayOfQuarter = QuarterCount,
                @DayOfWeekInYear = YearCount
        FROM @DayOfWeek
        WHERE DOW = DATEPART(DW, @CurrentDate)

/*End day of week logic*/


/* Populate Your Dimension Table with values*/

        INSERT INTO [dbo].[DimDate]
        SELECT

                CONVERT (char(8),@CurrentDate,112) as DateKey,
                @CurrentDate AS Date,
                CONVERT (char(10),@CurrentDate,103) as FullDateUK,
                CONVERT (char(10),@CurrentDate,101) as FullDateUSA,
                DATEPART(DD, @CurrentDate) AS DayOfMonth,
                --Apply Suffix values like 1st, 2nd 3rd etc..
                CASE
                        WHEN DATEPART(DD,@CurrentDate) IN (11,12,13)
                        THEN CAST(DATEPART(DD,@CurrentDate) AS VARCHAR) + 'th'
                        WHEN RIGHT(DATEPART(DD,@CurrentDate),1) = 1
                        THEN CAST(DATEPART(DD,@CurrentDate) AS VARCHAR) + 'st'
                        WHEN RIGHT(DATEPART(DD,@CurrentDate),1) = 2
                        THEN CAST(DATEPART(DD,@CurrentDate) AS VARCHAR) + 'nd'
                        WHEN RIGHT(DATEPART(DD,@CurrentDate),1) = 3
                        THEN CAST(DATEPART(DD,@CurrentDate) AS VARCHAR) + 'rd'
                        ELSE CAST(DATEPART(DD,@CurrentDate) AS VARCHAR) + 'th'
                        END AS DaySuffix,

                DATENAME(DW, @CurrentDate) AS DayName,
                DATEPART(DW, @CurrentDate) AS DayOfWeekUSA,

                -- check for day of week as Per US and change it as per UK format
                CASE DATEPART(DW, @CurrentDate)
                        WHEN 1 THEN 7
                        WHEN 2 THEN 1
                        WHEN 3 THEN 2
                        WHEN 4 THEN 3
                        WHEN 5 THEN 4
                        WHEN 6 THEN 5
                        WHEN 7 THEN 6
                        END
                        AS DayOfWeekUK,

                @DayOfWeekInMonth AS DayOfWeekInMonth,
                @DayOfWeekInYear AS DayOfWeekInYear,
                @DayOfQuarter AS DayOfQuarter,
                DATEPART(DY, @CurrentDate) AS DayOfYear,
                DATEPART(WW, @CurrentDate) + 1 - DATEPART(WW, CONVERT(VARCHAR,
                DATEPART(MM, @CurrentDate)) + '/1/' + CONVERT(VARCHAR,
```

```
                DATEPART(YY, @CurrentDate))) AS WeekOfMonth,
                (DATEDIFF(DD, DATEADD(QQ, DATEDIFF(QQ, 0, @CurrentDate), 0),
                @CurrentDate) / 7) + 1 AS WeekOfQuarter,
                DATEPART(WW, @CurrentDate) AS WeekOfYear,
                DATEPART(MM, @CurrentDate) AS Month,
                DATENAME(MM, @CurrentDate) AS MonthName,
                CASE
                        WHEN DATEPART(MM, @CurrentDate) IN (1, 4, 7, 10) THEN 1
                        WHEN DATEPART(MM, @CurrentDate) IN (2, 5, 8, 11) THEN 2
                        WHEN DATEPART(MM, @CurrentDate) IN (3, 6, 9, 12) THEN 3
                        END AS MonthOfQuarter,
                DATEPART(QQ, @CurrentDate) AS Quarter,
                CASE DATEPART(QQ, @CurrentDate)
                        WHEN 1 THEN 'First'
                        WHEN 2 THEN 'Second'
                        WHEN 3 THEN 'Third'
                        WHEN 4 THEN 'Fourth'
                        END AS QuarterName,
                DATEPART(YEAR, @CurrentDate) AS Year,
                'CY ' + CONVERT(VARCHAR, DATEPART(YEAR, @CurrentDate)) AS YearName,
                LEFT(DATENAME(MM, @CurrentDate), 3) + '-' + CONVERT(VARCHAR,
                DATEPART(YY, @CurrentDate)) AS MonthYear,
                RIGHT('0' + CONVERT(VARCHAR, DATEPART(MM, @CurrentDate)),2) +
                CONVERT(VARCHAR, DATEPART(YY, @CurrentDate)) AS MMYYYY,
                CONVERT(DATETIME, CONVERT(DATE, DATEADD(DD, - (DATEPART(DD,
                @CurrentDate) - 1), @CurrentDate))) AS FirstDayOfMonth,
                CONVERT(DATETIME, CONVERT(DATE, DATEADD(DD, - (DATEPART(DD,
                (DATEADD(MM, 1, @CurrentDate)))), DATEADD(MM, 1,
                @CurrentDate)))) AS LastDayOfMonth,
                DATEADD(QQ, DATEDIFF(QQ, 0, @CurrentDate), 0) AS FirstDayOfQuarter,
                DATEADD(QQ, DATEDIFF(QQ, -1, @CurrentDate), -1) AS LastDayOfQuarter,
                CONVERT(DATETIME, '01/01/' + CONVERT(VARCHAR, DATEPART(YY,
                @CurrentDate))) AS FirstDayOfYear,
                CONVERT(DATETIME, '12/31/' + CONVERT(VARCHAR, DATEPART(YY,
                @CurrentDate))) AS LastDayOfYear,
                NULL AS IsHolidaySL,
                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 IsWeekday,
                NULL AS HolidaySL, (case when @CurrentDate = convert(date, sysdatetime()) then 1
else 0 end), 0, 0

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

Data Profiling on staging tables