THE Voice
SPEAKS FOR ITSELF

2021

Presented on: 27-04-2021.

Created By: Oz Maurer, Amit Avital & Alex Birbrair

The Voice

BI Project

Table of Contents

Introduction

As part of the BI programing course, we were required to use all of what we learn to sum it to a final project that will test all the aspect of the BI programs job.

The purpose of the project is to simulate real operational environment of a cellular company that needs to move to a data-driven way of working to make decision based on data and numbers rather than on feelings or hunches.

The requirement configurations and definitions of needs was given and based on a real production environment.

The company that the project is based on called "The Voice". It is a cellular company with more than 18,000 users. The company gives services of voice calls, data usage and SMS massaging across the world and working with 5 Israeli formal operators. The company has several service packages with different usage prices as well as discounts that differs between users.

This project included all the basic & advanced element of the process of creating the platform and developing the interface for a BI product; company business analysis, gathering of data from different sources, centralizing all sources into one structure, cleansing and manipulation of the data, automating of periodic refresh in efficient way, decentralizing access and permissions according the roles and perspectives of different future users, planning, designing and development of report & dashboard to present cross company data and turning data into valuable knowledge that make decisions making based on actual numbers & facts.

Operational System Diagram

Usage Main – a catalog of calls loged in a table based on the origin number.

Customer lines – a catalog of operating phone numbers that represents a distinct list of users based on their phone number.

Customer Invoice – a catalog of invoices bases on usage of users.

Coutries – a catalog of all the coutries that that the company in working with (incoming or outgoing calls).

Package Catalog – a catalog of all the packages the company has, active & inactive.

Customers – a catalog of all the costumers, past and present.

Xxcountrypre – a catalog of coutry prefixes.

OPFILEOPP - a catalog of operators coming from out side of the operation system in a form of CSV files.

CallType – a catalog of the veriaty of servises and sub-servies the company offers that comes from out side of the operational system in a form of CSV files.

customer_lines

PHONE_NO
createdate
enddate
status
TYPE
[DESC]
insert_date
update_date
discountpct
numberoffreeminutes

countries

COUNTRY_CODE
[DESC]
REGION
AREA
insert_date
update_date

customer

customer_id
CUST_NUMBER
cust_name
address
insert_date
update_date

USAGE_MAIN

CALL_NO
ANSWER_TIME
SEIZED_TIME
DISCONNECT_TIME
CALL_DATETIME
CALLING_NO
CALLED_NO
DES_NO
DURATION
CUST_ID
CALL_TYPE
PROD_TYPE
RATED_AMNT
RATED_CURR_CODE
CELL
CELL_ORIGIN
HIGH_LOW_RATE
insert_DATE
update_date

XXCOUNTRYPRE

COUNTRY_CODE
COUNTRY_PRE

CUSTOMER_INVOICE

INVOICE_NUM
PHONE_NO
INVOICE_TYPE
INVOICE_DATE
INVOICE_IND
INVOICE_DESC
INVOICE_CURRNCY
INVOICE_AMOUNT
insert_date
update_date

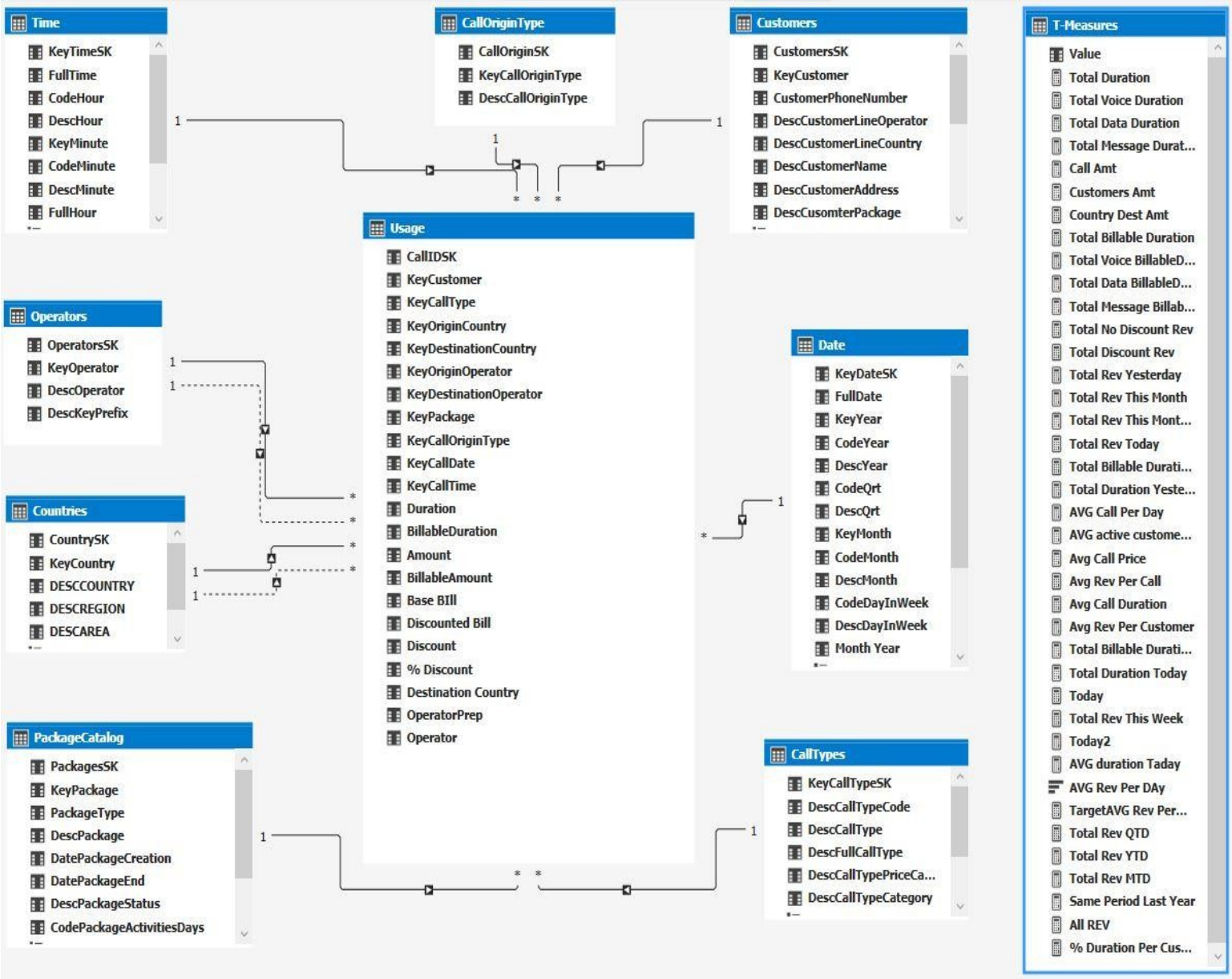
Package_Catalog

PACKAGE_NUM
createdate
enddate
status
pack_type
pack_desc
insert_date
update_date

Project Assumptions

- All 7 reports that ware mentioned in the requirement doc (appendix A) are required to be part of the project and any other report in a bonus.
- The following company challenges ware not treated due to lack of operational data regarding costs, employees & resources.
    - Proper pricing of various services.
    - Transfer resources to one service at another expense.
    - Finding customers who tend to abandon the company and give an answer before leaving.
    - Proper resource allocation to save expenses.
- A three-digit prefix of the telephone number is a unique value for the company name (for example 054 = Orange, Ignore number assignment)
- A country prefix is up to three digits (for example 1 = America, 972 = Israel)
- Some prefixes ware used for more than one country, to cope with that it was decided to merge those numbers into the first country that uses that prefix (For example USA & Canada both have 1 as a prefix, in this case call from Canada ware addressed as call from USA).
- The telephone number of a particular person constitutes a unique value for his number as a customer, that is, a particular customer has only one phone number.
- A customer is uniquely identified by a telephone number, i.e., a certain customer has one telephone number.

Data Warehouse Star Schema



Project Development Stages

Stage 1 – Configuration of the business environment

In this stage we went through all the business aspect of the company, learned and analyze the tables, files and the relationships between them, understand the business requirements and created a general & dip understanding of the BI need of the company based on the configuration file we recieved.

Stage 2 – Planning

In this stage we created a plan and a schedule for the project. Decided when and how to work on each stage, how to share the workload between the group members, decided on which platform to work and how to communicate the work between 3 computers and created a general vision of how the end result will look like.

Stage 3 – mirroring**

In this stage we copy the source files or tables, with no logic or enrichment. Source data is copied and added to the target mirror tables, which then hold historical raw data that is ready to be transformed.

Stage 4 – Staging**

In this stage data cleansing and transformation is done. Once the raw data from the mirror tables is transformed, all transformations are stored in staging tables. These tables hold the final form of the data.

Stage 5 – DW**

In this stage we preper and copy to the destination tables, which contain all the data in its final form after cleansing, enrichment and transformation. We also aggregate data to a daily or store level from the full dataset. This can improve report performance, enable the addition of business logic to calculated measures and make it easier for report developers to understand the data.

Stage 6 – SSRS - report creation

In this stage we use the SQL Server Data Tools for Business Intelligence (SSDT BI) to create the SSRS report. We use our DW and measures to create reports for each required subject or aspect of the company's actives. In our case we created 3 report: Usage by country/Usage by call type/Dormant users.

Stage 7 – SSAS – Creation and analyzing Tabular DB

In this stage we when from a relational data base to tabular data set that runs on the in-memory. This is done with the state-of-the-art data compression algorithms. In this stage we worked on creating the relationships between the new tabular table to create a star schema model, we worked on tuning the tabular data base with calculated tables, calculating columns, modifying data types, creating measures table and parameter table and much more all by using the DAX programing language.

Stage 8 – Power BI – report creation

In this stage we moved to creating the actual UI/UX part of the project. By using the Power BI Desktop program, we created the reports and the dashboard for all the users using the tabular model we created in the last stage. We used different visualizations to present the requires data in the most easy and intuitive way. This way instead of querying thousands of entries in different table by SQL and T-SQL the end user (managers) could look and 4/5 visualizations and in few second understand the current state of business. For professional users (analysts) it will be very easy to use the star schema model we created to work on their analysis of different aspect of the business.

Stage 9 – QA and Touch-up

In this stage we worked on designing the look of the reports and the general view of the "Application" we created like color scales, sizing, and fonts. Also, we done a QA stage where we tried to fail the system and verify that all calculation are correct, jobs are running smooth and security work as planned.

** Stages 3-5 where done with the SSIS program.

Challenges & Difficulties

Duplicated information

We encountered in several location in duplicated or conflicting information. For example, one phone number with more than one user, in this case it was decided to allocate the phone number to the users that used it last. Another example is that several countries had the same prefix (Italy & the Vatican/USA & Canada etc.) in this case we combined the call to the country we choose to continue with (Italy & USA).

-One unique bug we found is that the RATED_AMNT in the "THE VOICE" operational DB is INT but when we tested the data, we found that it is the result of "Duration"*"pricepermin". That means that all the partial AMNT after the decimal point is not calculated for future data and it sums to a loss of 46% in the revenue calculation. We fixed it to maintain high data integrity.

Data Cleansing

There were a lot of data cleansing needed to be done so it took a big part of our time especially because the initial idea of working "by the book" was the wrong choice. We found that in many cases understanding of what the information will used for at the beginning will helps you create data cleansing default more relevant and suitable for final stages of the project.

Formatting problems in SSIS

While working on the ETL with the Microsoft SSIS tool we found the formatting and data type conversion on of the most common cases for the process to fail. There was a need for a lot of data conversions and derived columns, that lead to creation a lot of unused columns, what increased the chances for confusion and mistakes in association between columns in those components.

SSIS ETL process

One of the most time-consuming parts of the project was the SSIS ETL process. There were a lot of components with a lot of limitation, that we were not aware of. For each component, many steps and set-ups were needed. Maybe it was only for us, but we felt that this part of the course was the least thorough. Therefor we needed to improve and train our google searching to finalize this part. Another

big issue was that after we enlarged the operational data base, the ETL process in the SSIS tool was unable to run properly using only the components gives in the tool. Instead we had to give up a lot of “SORT” & “MERGE JOIN” components and write SQL script and nest them in “OLE DB SUARCE” component.

Teamwork difficulties

Our team includes 3 developers. Although we learned and practices different languages and software, we did not go through the basic tool of working in a team while developing a BI solution. It begins with separating the work loud and how to share development components, how to create a shared workspace that could contain inputs from 3 different computers without setting up the connections and other setup processes.

Data Enlargement

Some of the most difficult part was the assignment to enlarge the DB from 2 days of usage to at least 1 year. We created data for 4 years and although we used **rand()** functions the data was very artificial looking.

Summery

We are very proud of our result. A full end to end solution that includes accepting a new assignment from a "client", understanding a new company operations and business, setting up the infrastructure for the development environment and working many days and nights to complete the project on time. There were a lot of problem on the way, A lot of them were a simple solution that was found after complicated and long search and trouble shoot. We learned a lot about teamwork and compromise. it is not that simple to manage this type of project via zoom and several computers, but we found a way that work for us.

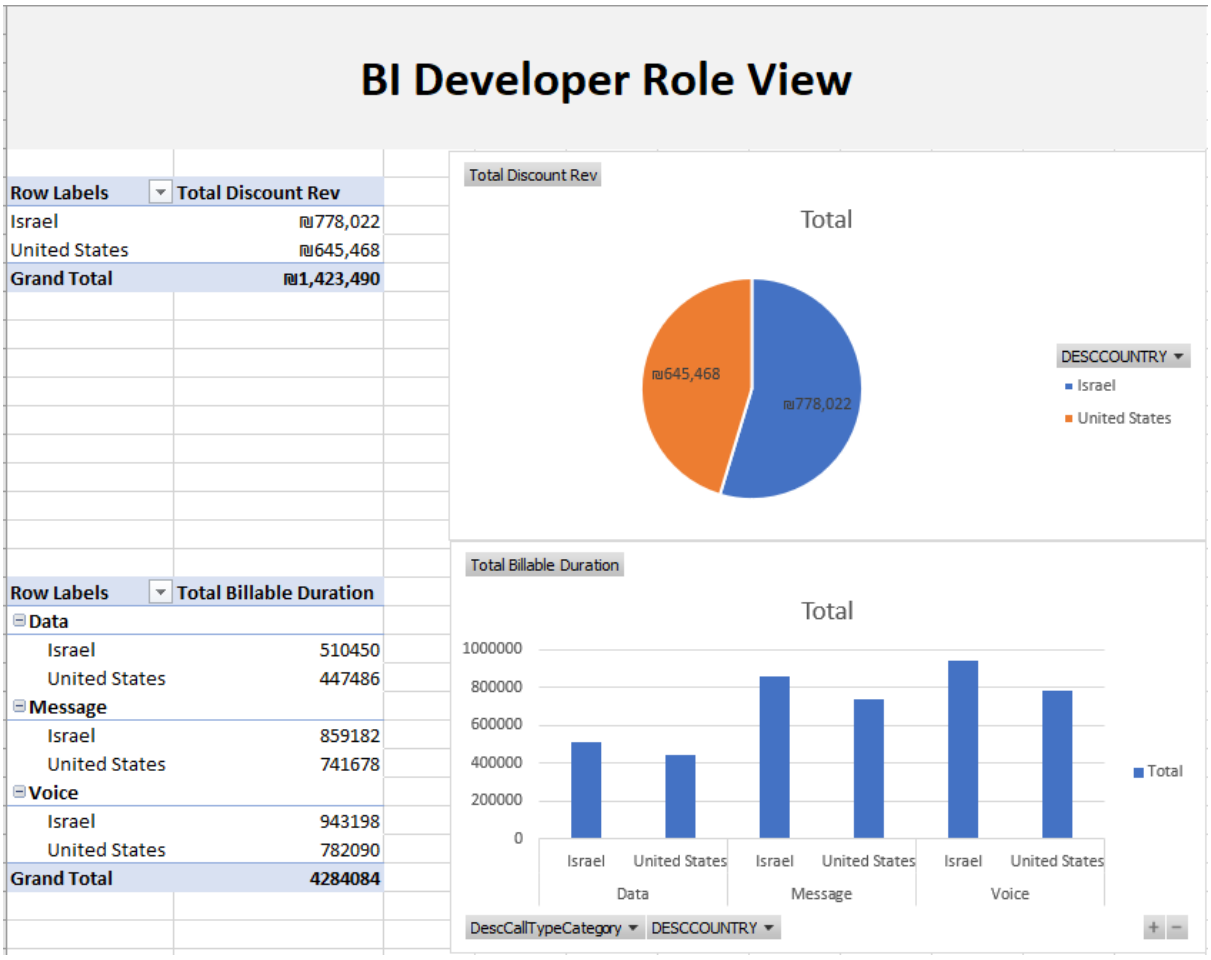
The result includes 8 reports and a dashboard, 3 Roles for different user types, automated job scheduler and three junior BI developers.

We want to say that this course was very thorough and included all the aspects we expected from a 6-month development course. From the presentations to the class exercise and the homework. The course was really testing our limits in term investing time and effort but was worth the while.

Appendixes

Appendix A – Excel screen shoots

1. BI Developer



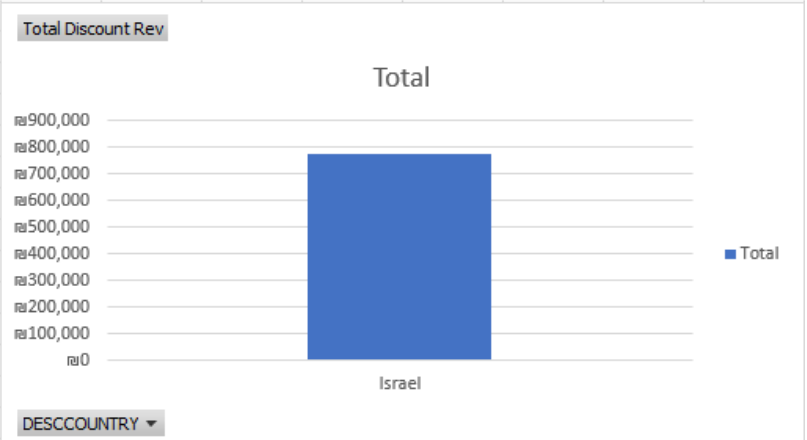
Row Labels	Total Discount Rev
Israel	₪778,022
United States	₪645,468
Grand Total	**₪1,423,490**

Row Labels	Total Billable Duration
Data	
Israel	510450
United States	447486
Message	
Israel	859182
United States	741678
Voice	
Israel	943198
United States	782090
Grand Total	**4284084**

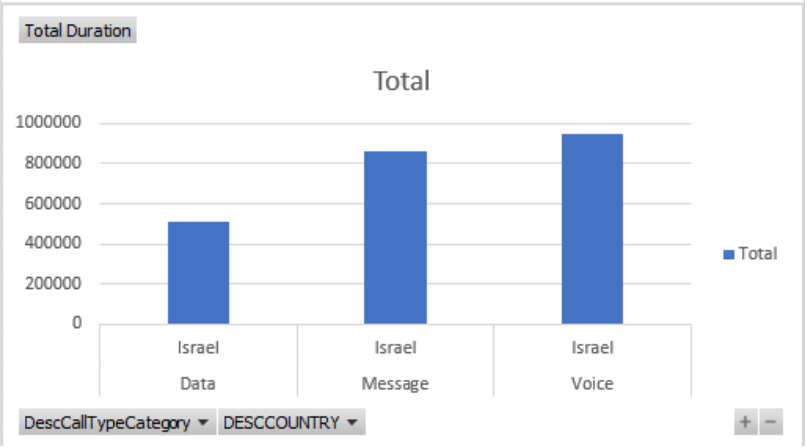
2. Israel Market Manager

Isarel Marketing Manager Role View

Row Labels	Total Discount Rev
Israel	₪778,022
Grand Total	**₪778,022**



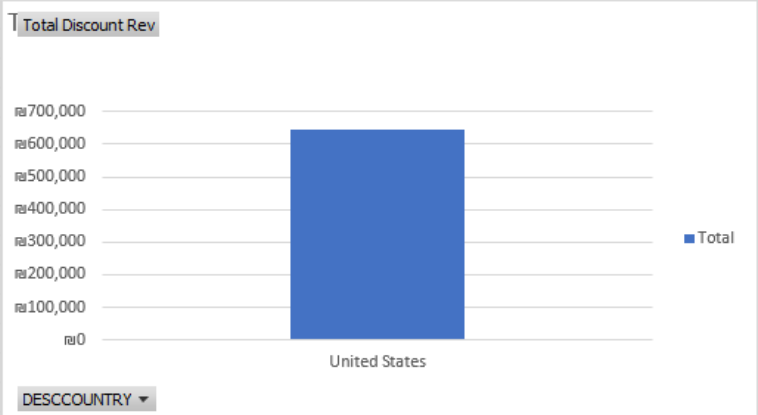
Row Labels	Total Duration
Data	
Israel	511144
Message	
Israel	860392
Voice	
Israel	944422
Grand Total	**2315958**



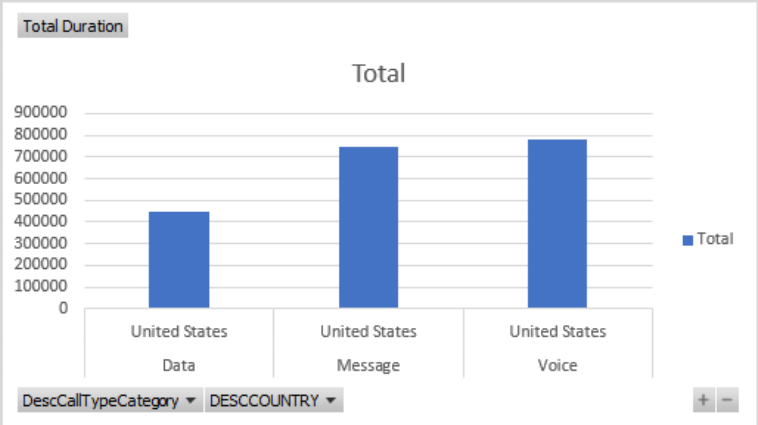
3. US Market Manager

US Marketing Manager Role View

Row Labels	Total Discount Rev
United States	₪645,468
Grand Total	**₪645,468**

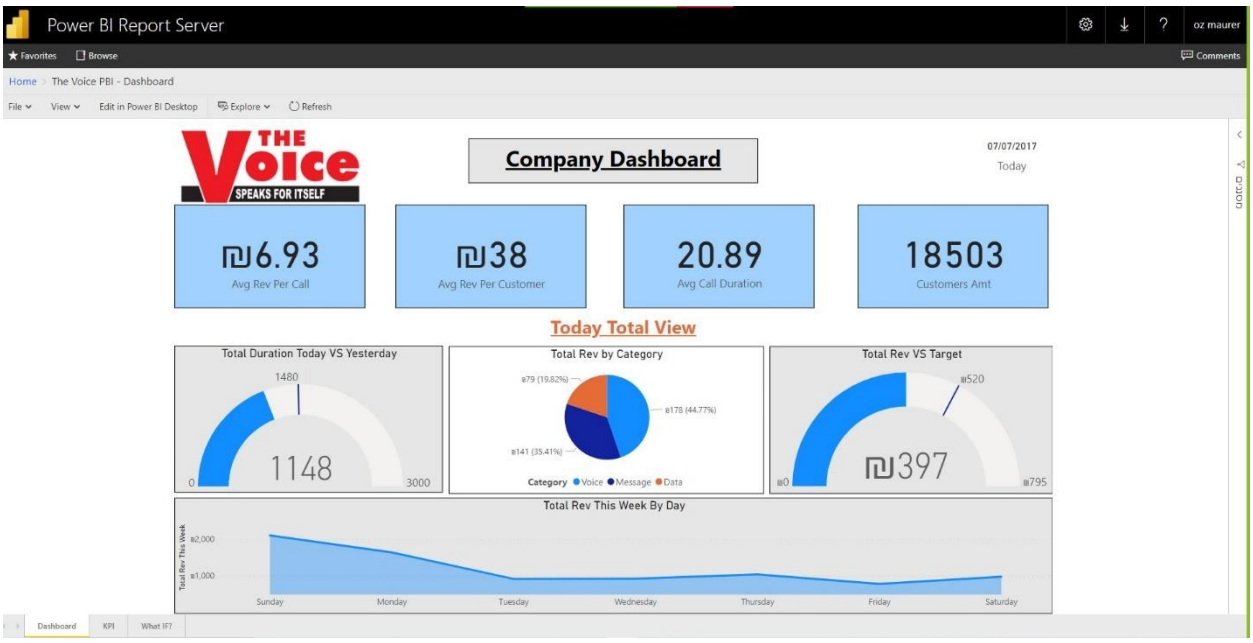


Row Labels	Total Duration
Data	
United States	448102
Message	
United States	742728
Voice	
United States	783292
Grand Total	**1974122**

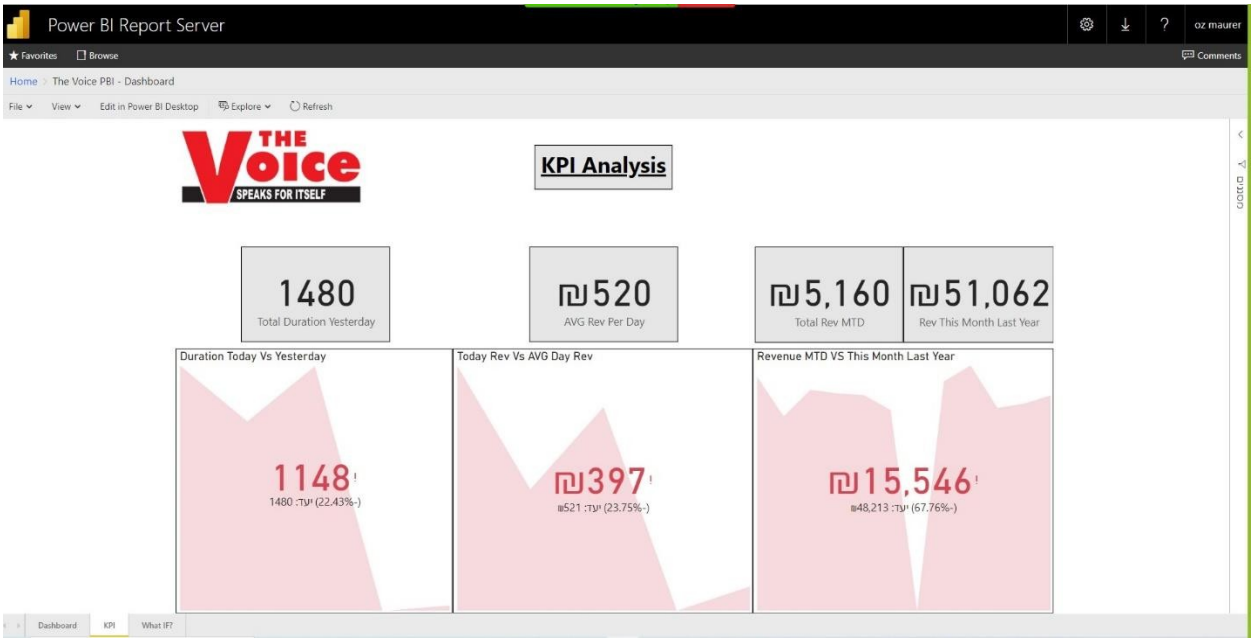


Appendix B – Power BI Published Reports

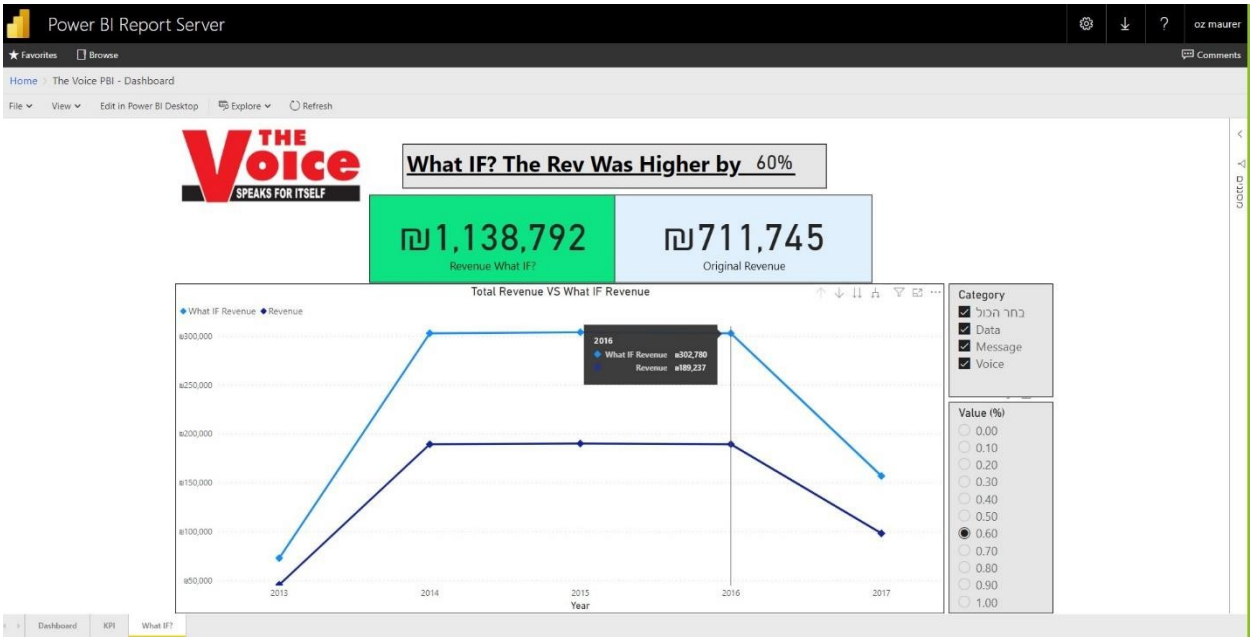
1. Dashboard



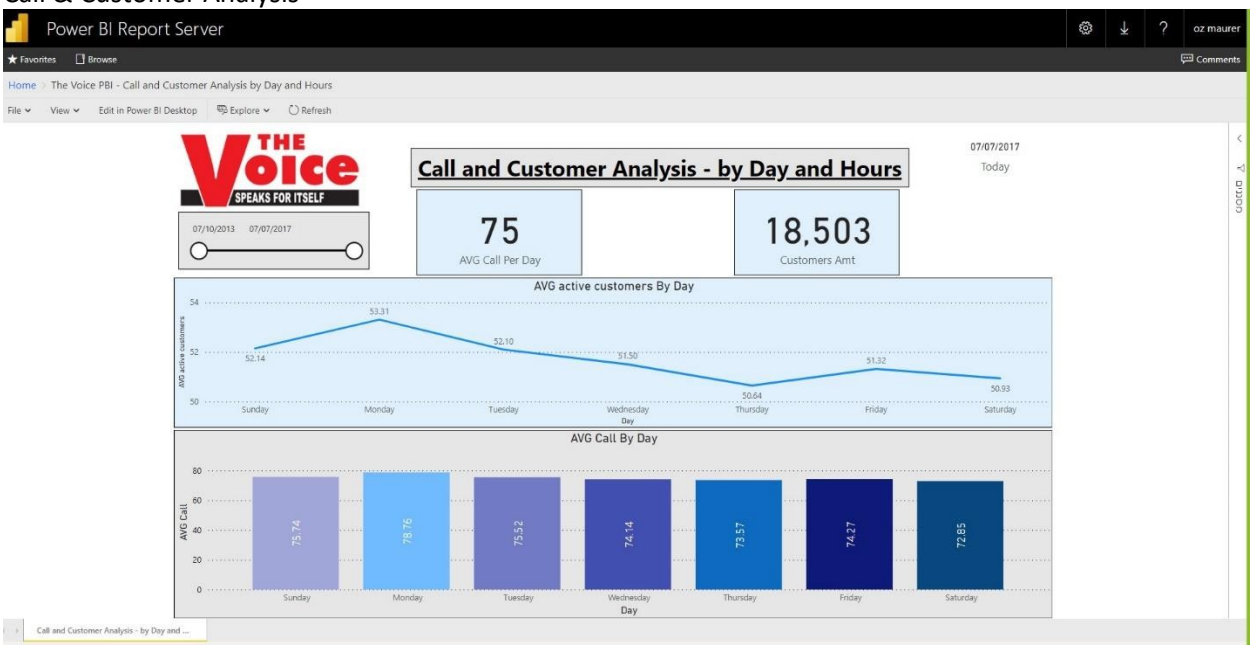
2. KPI



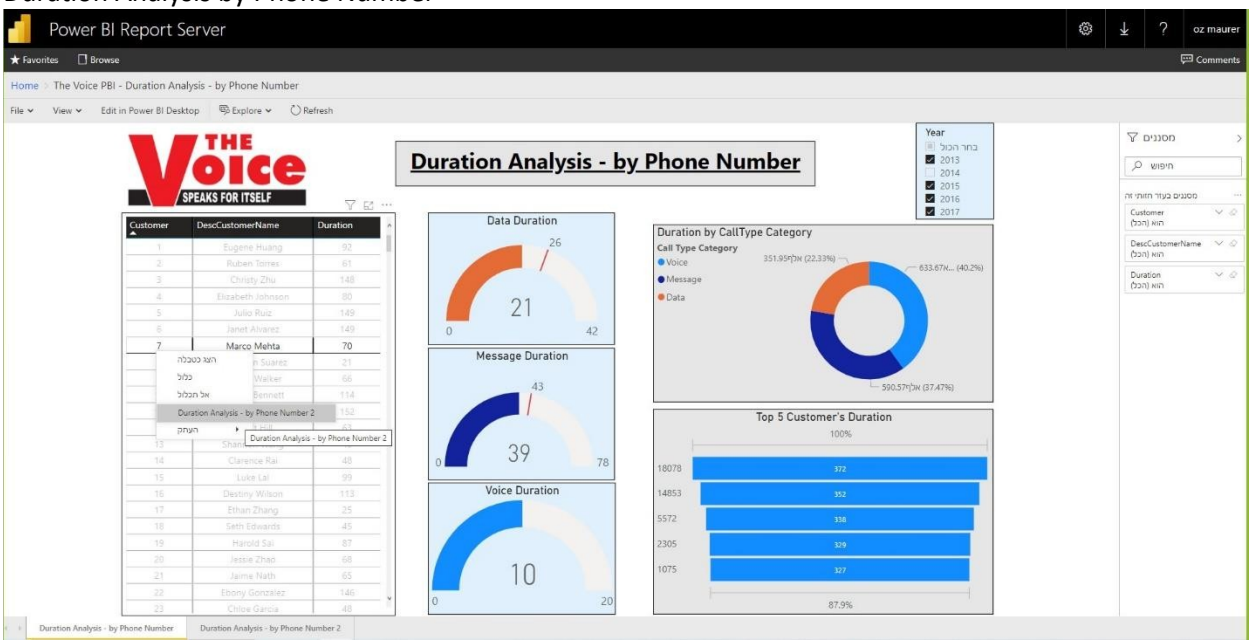
3. What If

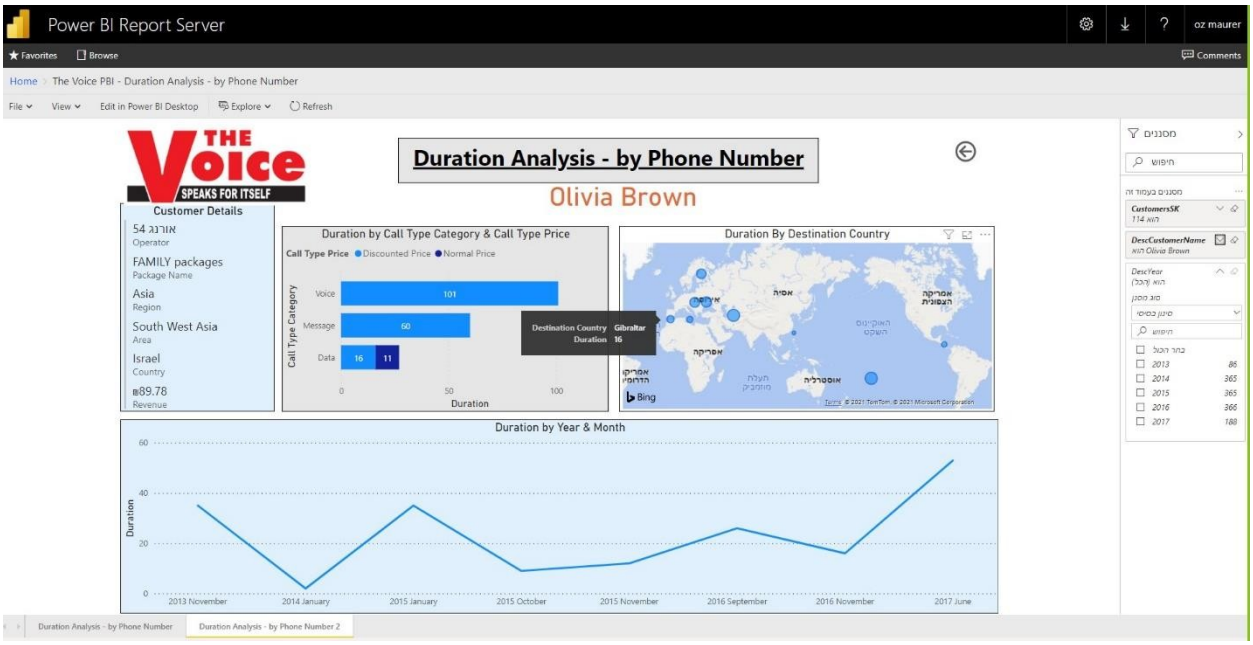


4. Call & Customer Analysis

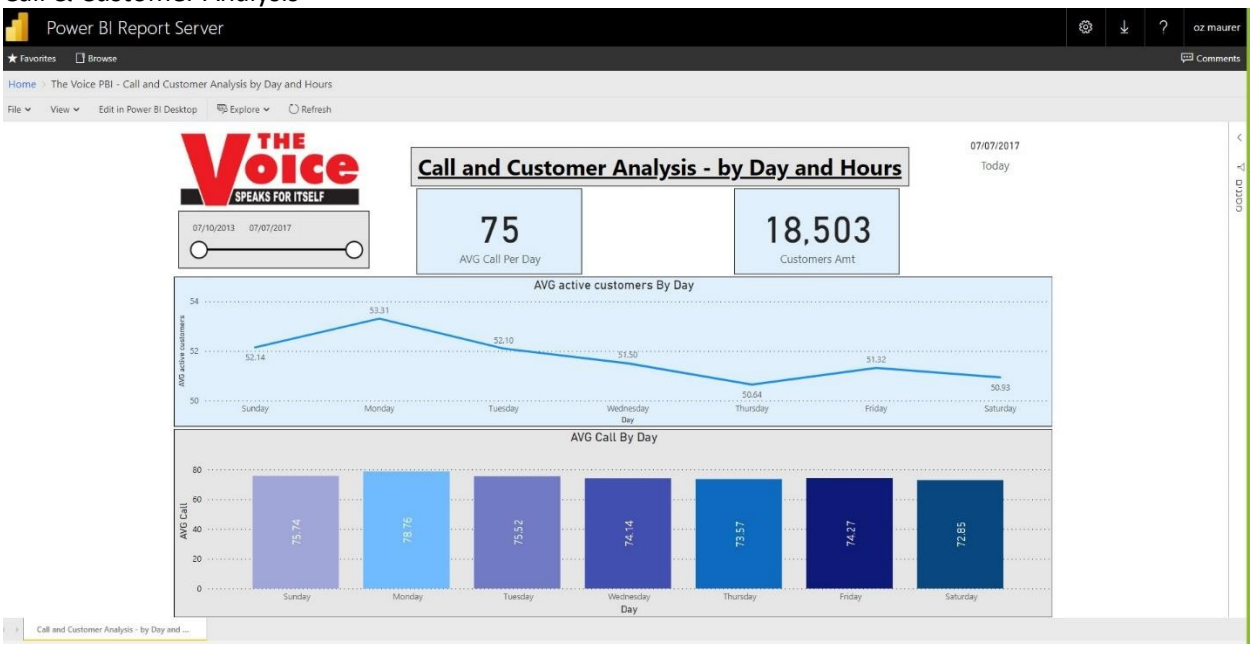


5. Duration Analysis by Phone Number

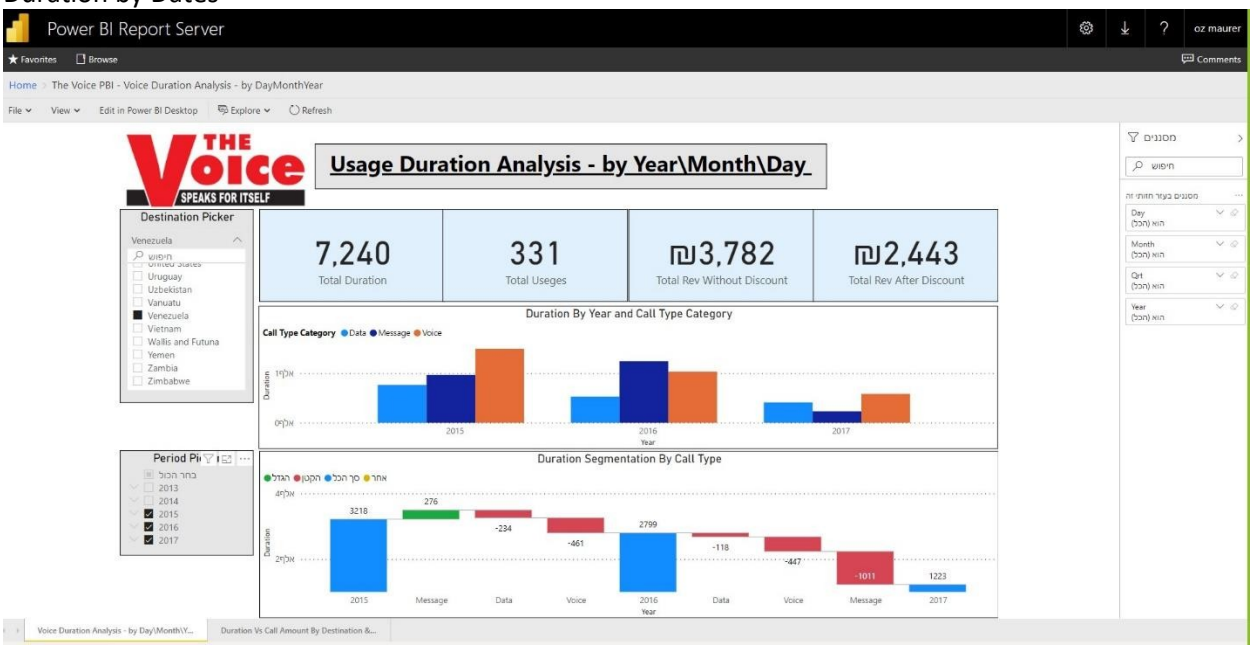




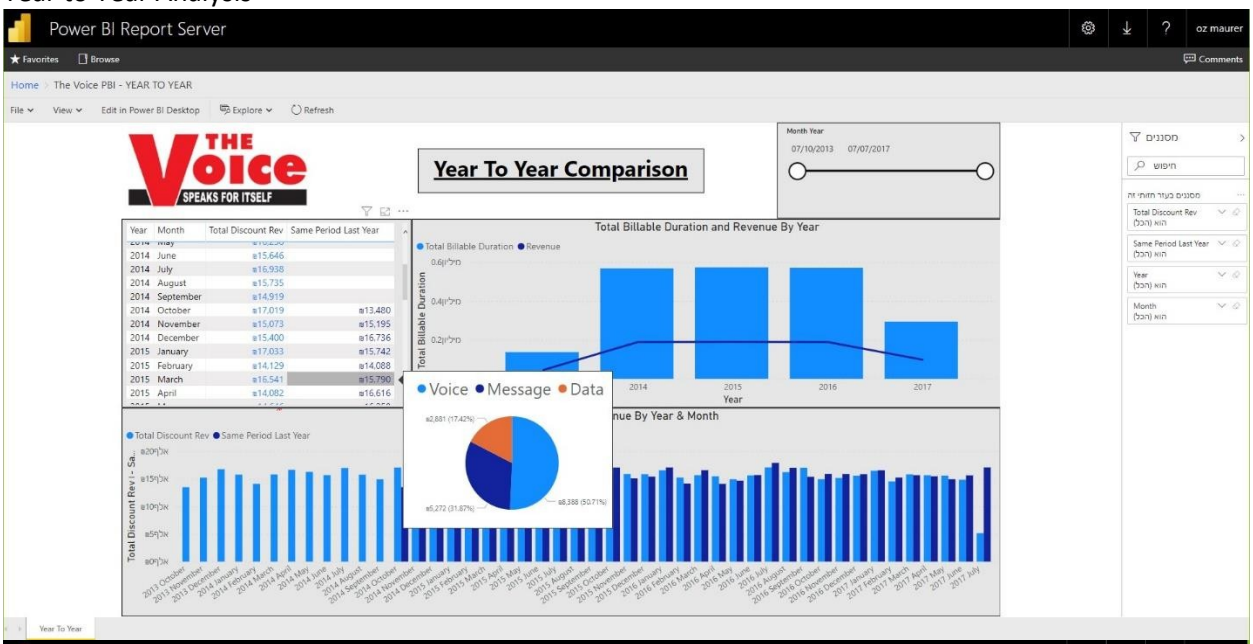
6. Call & Customer Analysis



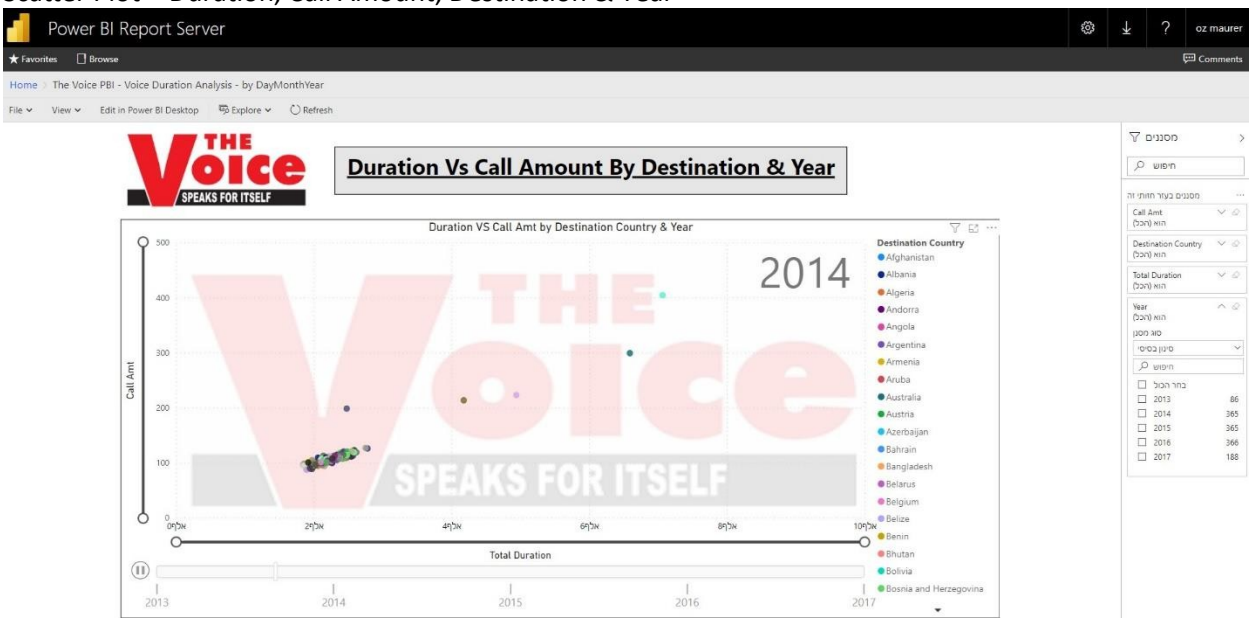
7. Duration by Dates



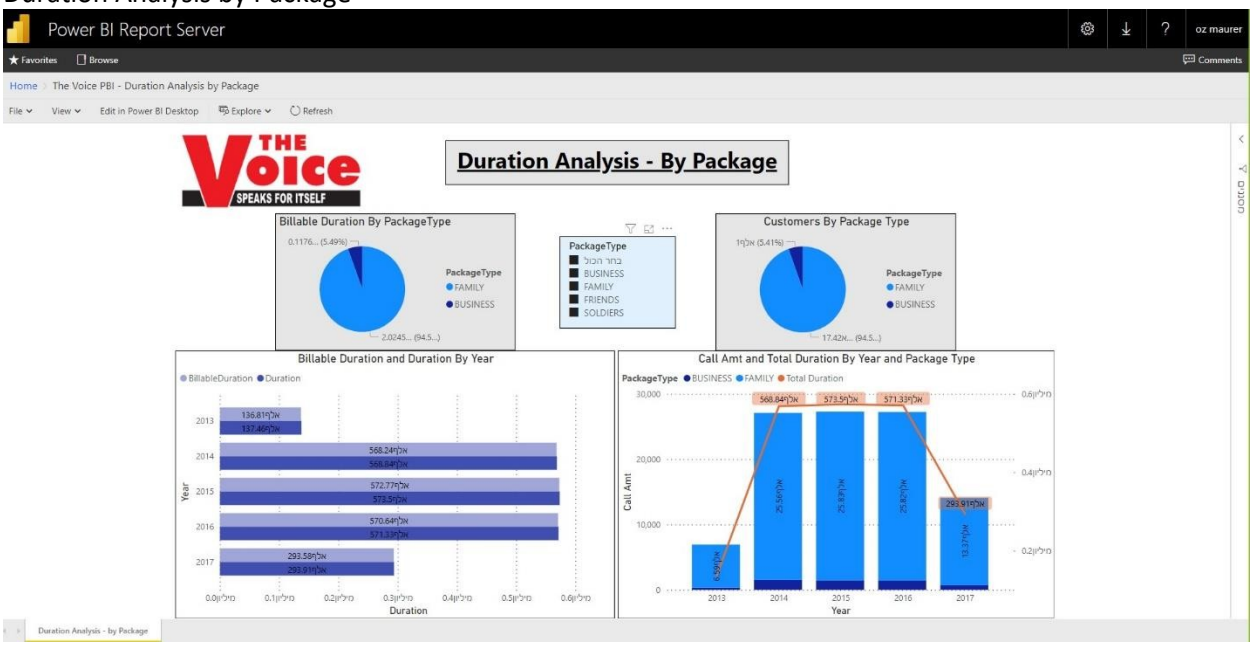
8. Year to Year Analysis

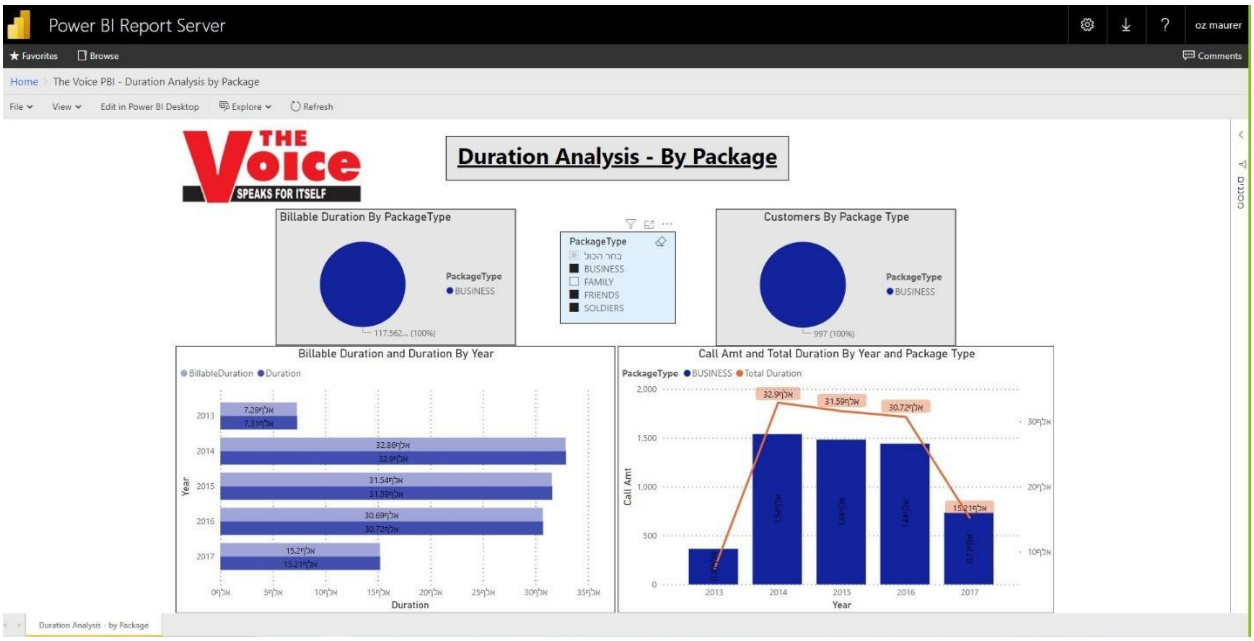


9. Scatter Plot – Duration, Call Amount, Destination & Year



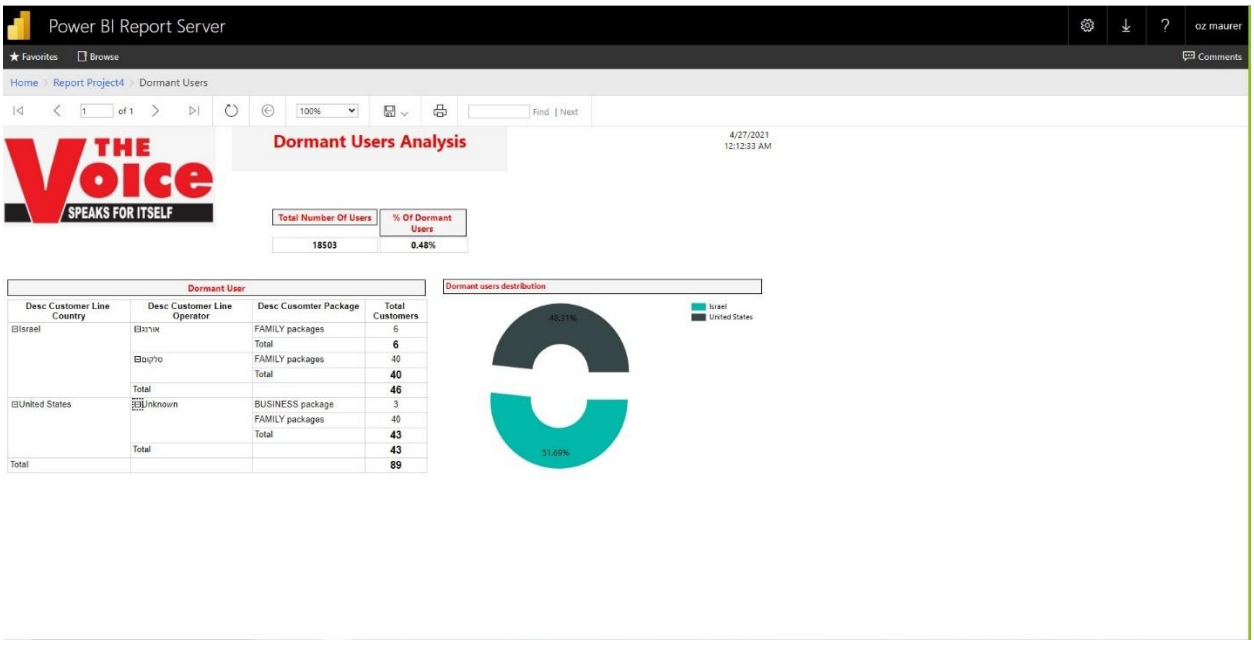
10. Duration Analysis by Package



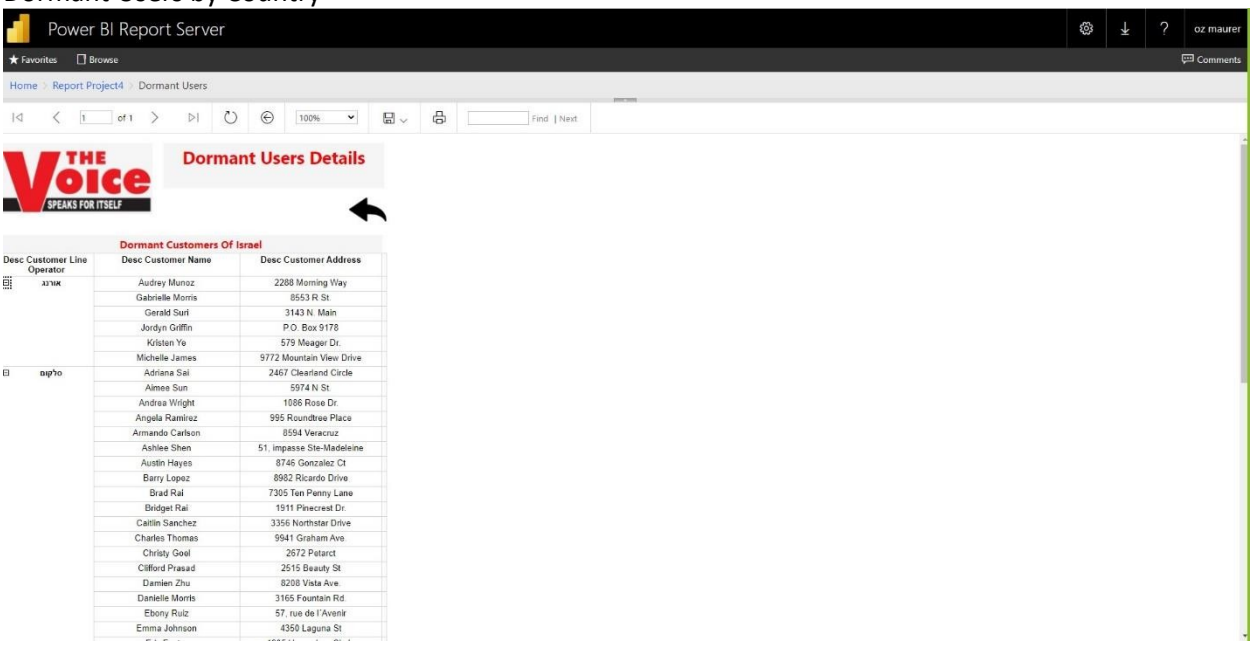


Appendix C - SSRS Reports

1. Dormant Users



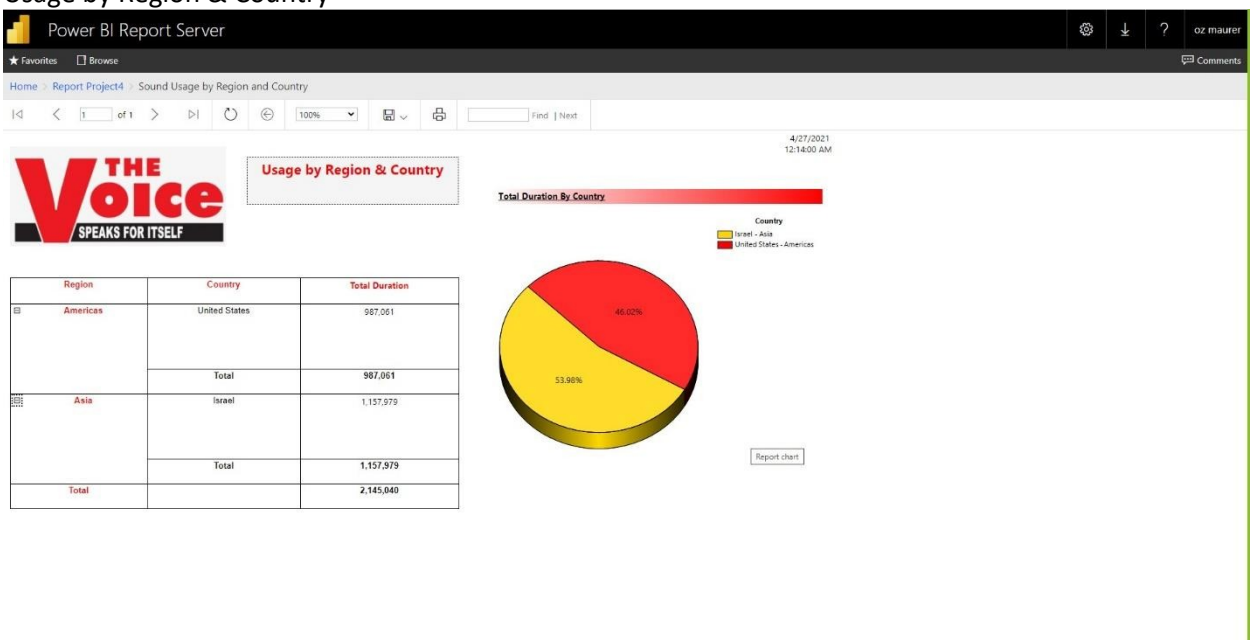
2. Dormant Users by Country



Dormant Customers Of Israel

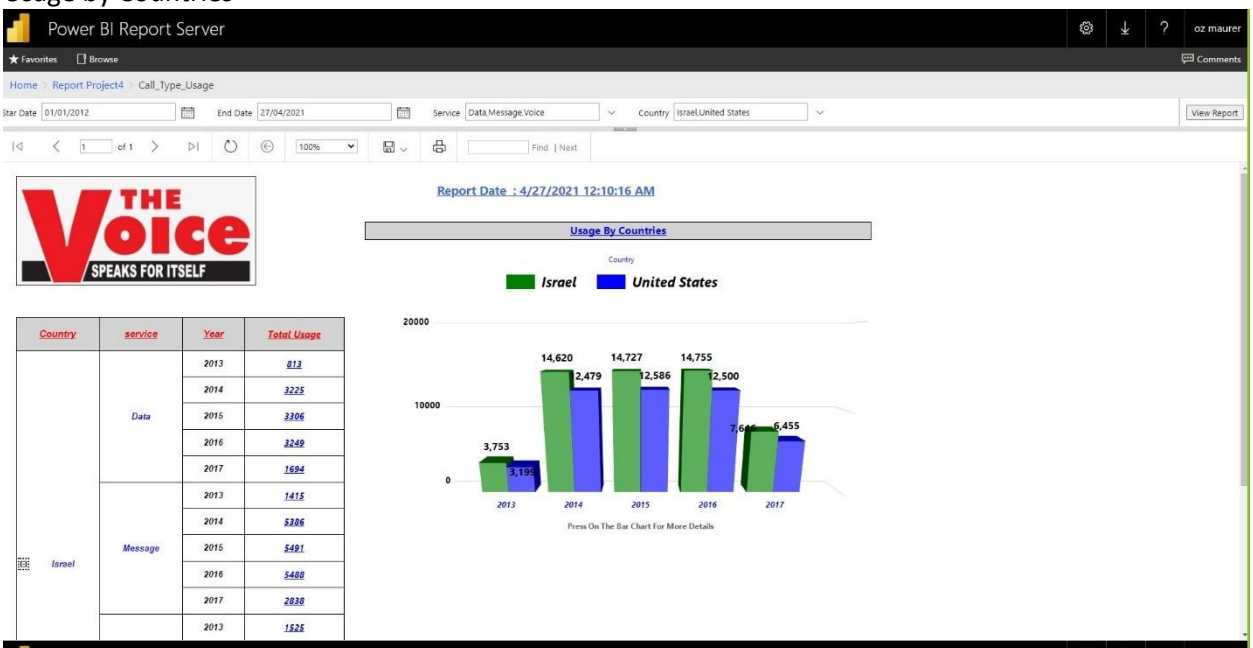
Desc Customer Line Operator	Desc Customer Name	Desc Customer Address
אורנה	Audrey Munoz	2288 Morning Way
	Gabrielle Morris	8553 R St
	Gerald Suri	3143 N. Main
	Jordyn Griffin	P.O. Box 9178
	Kristen Ye	579 Meager Dr.
	Michelle James	9772 Mountain View Drive
טליקוס	Adriana Sai	2467 Clearland Circle
	Aimee Sun	5974 N St
	Andrea Wright	1086 Rose Dr.
	Angela Ramirez	995 Roundtree Place
	Armando Carlson	8594 Veracruz
	Ashlee Shen	51, impasse Ste-Madeleine
	Austin Hayes	8746 Gonzalez Ct
	Barry Lopez	8982 Ricardo Drive
	Brad Rai	7305 Ten Penny Lane
	Bridget Rai	1911 Pinecrest Dr.
	Caitlin Sanchez	3356 Northstar Drive
	Charles Thomas	9941 Graham Ave.
	Christy Goel	2672 Petarct
	Clifford Prasad	2515 Beauty St
	Damien Zhu	8208 Vista Ave.
	Danielle Morris	3165 Fountain Rd.
	Ebony Ruiz	57, rue de l'Avenir
	Emma Johnson	4350 Laguna St

3. Usage by Region & Country

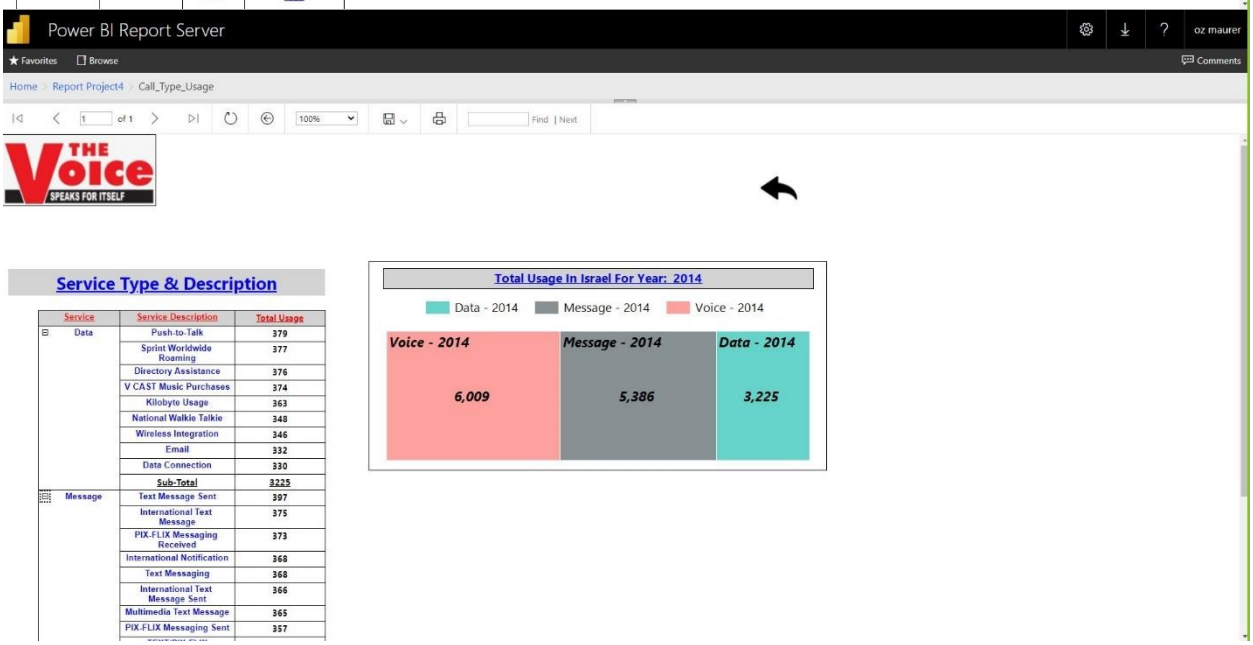


Region	Country	Total Duration
Americas	United States	987,061
	Total	987,061
Asia	Israel	1,157,979
	Total	1,157,979
Total		2,145,040

4. Usage by Countries



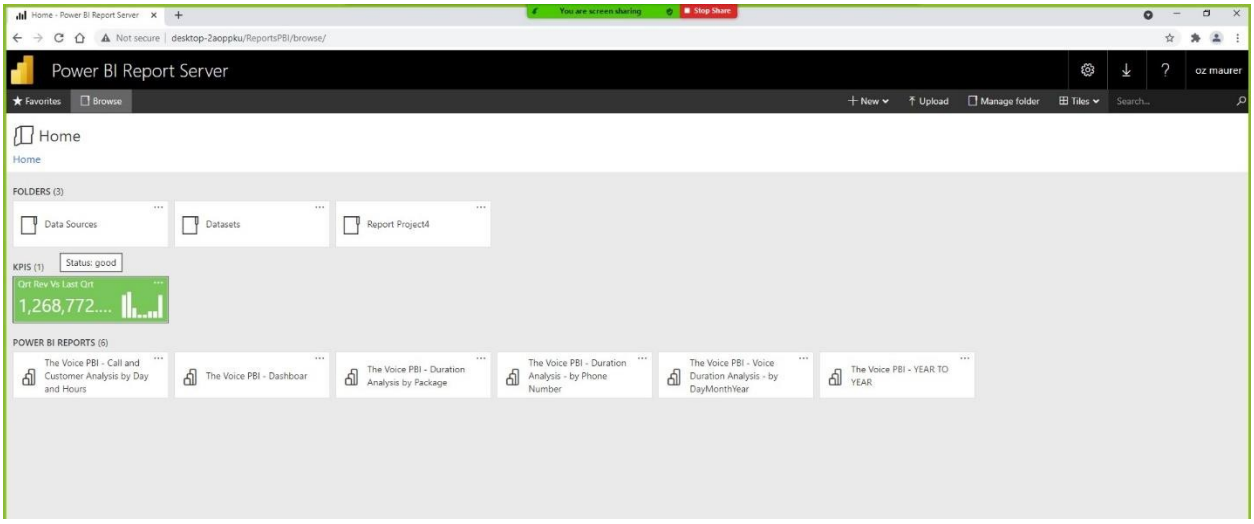
Country	service	Year	Total Usage
Israel	Data	2013	813
		2014	3225
		2015	3306
		2016	3249
		2017	1694
	Message	2013	1415
		2014	5386
		2015	5491
		2016	5488
		2017	2838
		2013	1525

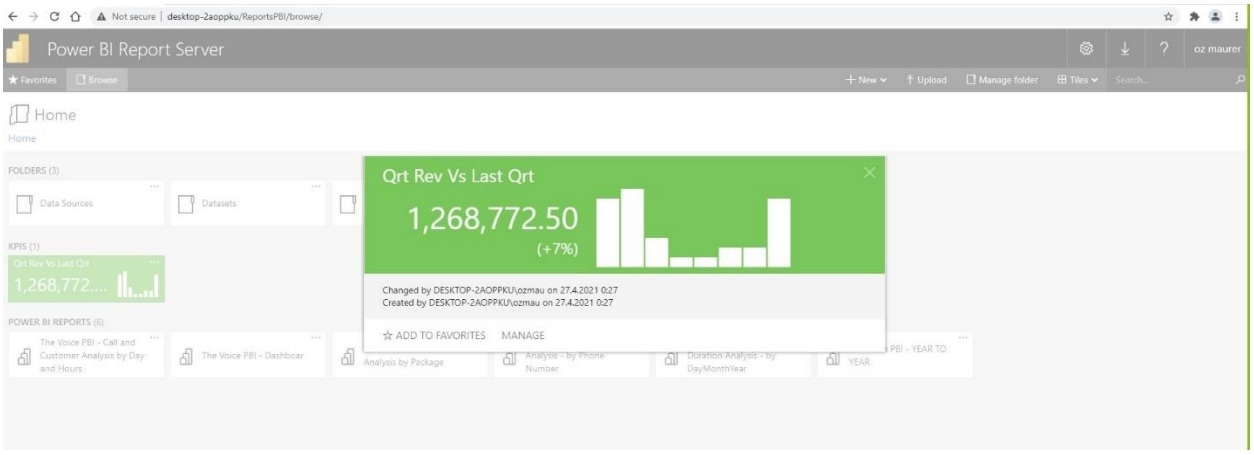


Service Type & Description

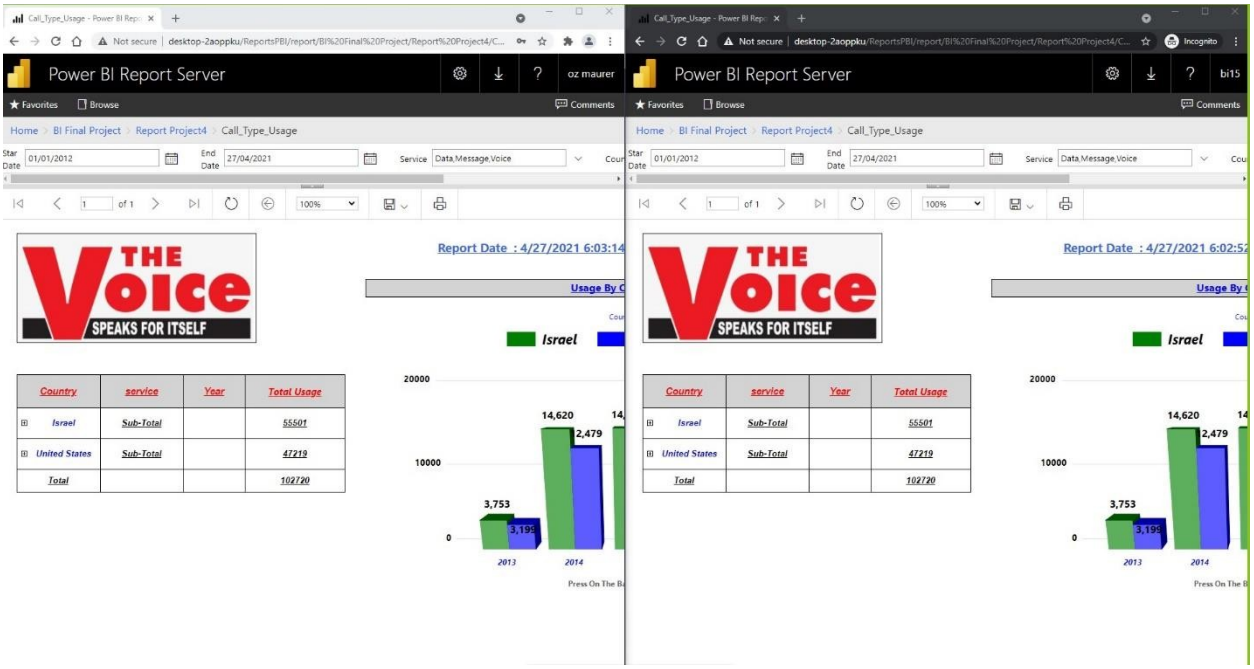
Service	Service Description	Total Usage
Data	Push-to-Talk	379
	Sprint Worldwide Roaming	377
	Directory Assistance	376
	V CAST Music Purchases	374
	Kilobyte Usage	363
	National Walkie Talkie	348
	Wireless Integration	346
	Email	332
	Data Connection	330
	Sub-Total	3225
Message	Text Message Sent	397
	International Text Message	375
	PIX-FLIX Messaging Received	373
	International Notification	368
	Text Messaging	368
	International Text Message Sent	366
	Multimedia Text Message	365
	PIX-FLIX Messaging Sent	357

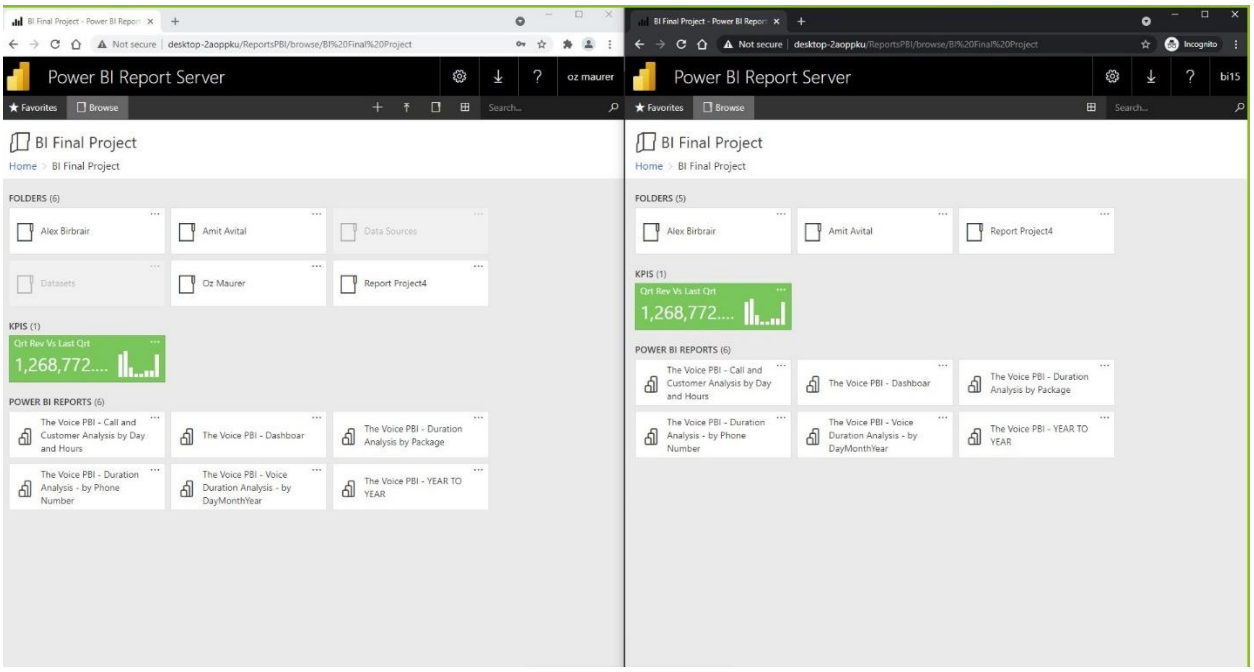
Appendix D – URL Screen Shoot





Appendix E – Security





Appendix F – SQL Queries

```
------------------------------ Final Project - The Vocie - O&A -------------------------
----------
------------------------------ Oz Maurer, Amit Avital & Alex Birbrair -------------------
----------

-------------------------------- Creation of The Voice MRR DB---------------------------
----------
CREATE DATABASE [TheVoice_MRR]
GO

USE [TheVoice_MRR]
GO

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

-------------------------------- Creation of Tables in TheVoice_Mrr ---------------------
----------
CREATE TABLE [dbo].[Countries_Mrr](
	[COUNTRY_CODE] [nvarchar](100) NOT NULL,
	[DESC] [nvarchar](100) NULL,
	[REGION] [nvarchar](100) NULL,
	[AREA] [nvarchar](100) NULL,
	[insert_date] [datetime] NULL,
	[update_date] [datetime] NULL)
GO

CREATE TABLE [dbo].[Customer_Mrr](
	[customer_id] [int] NOT NULL,
	[CUST_NUMBER] [nvarchar](20) NOT NULL,
	[cust_name] [nvarchar](100) NULL,
	[address] [nvarchar](100) NULL,
	[insert_date] [datetime] NULL,
	[update_date] [datetime] NULL)
GO


CREATE TABLE [dbo].[CUSTOMER_INVOICE_Mrr](
	[INVOICE_NUM] [int] NOT NULL,
	[PHONE_NO] [nvarchar](20) NOT NULL,
	[INVOICE_TYPE] [nvarchar](10) NULL,
	[INVOICE_DATE] [datetime] NULL,
	[INVOICE_IND] [tinyint] NULL,
	[INVOICE_DESC] [nvarchar](100) NULL,
	[INVOICE_CURRNCY] [nvarchar](10) NULL,
	[INVOICE_AMOUNT] [decimal](10, 4) NULL,
	[insert_date] [datetime] NULL,
	[update_date] [datetime] NULL)
GO

CREATE TABLE [dbo].[Customer_lines_Mrr](
	[PHONE_NO] [nvarchar](20) NOT NULL,
```

```
        [createdate] [datetime] NOT NULL,
        [enddate] [datetime] NULL,
        [status] [nvarchar](4) NULL,
        [TYPE] [nvarchar](10) NULL,
        [DESC] [nvarchar](100) NULL,
        [insert_date] [datetime] NULL,
        [update_date] [datetime] NULL,
        [discountpct] [int] NULL,
        [numberoffreeminutes] [int] NULL)
GO

CREATE TABLE [dbo].[Package_Catalog_Mrr](
        [PACKAGE_NUM] [int] NOT NULL,
        [createdate] [datetime] NULL,
        [enddate] [datetime] NULL,
        [status] [nvarchar](4) NULL,
        [pack_type] [nvarchar](10) NULL,
        [pack_desc] [nvarchar](100) NULL,
        [insert_date] [datetime] NULL,
        [update_date] [datetime] NULL)
GO

CREATE TABLE [dbo].[USAGE_MAIN_Mrr](
        [CALL_NO] [int] NOT NULL,
        [ANSWER_TIME] [datetime] NOT NULL,
        [SEIZED_TIME] [datetime] NOT NULL,
        [DISCONNECT_TIME] [datetime] NOT NULL,
        [CALL_DATETIME] [datetime] NULL,
        [CALLING_NO] [nvarchar](18) NULL,
        [CALLED_NO] [nvarchar](18) NULL,
        [DES_NO] [nvarchar](25) NULL,
        [DURATION] [int] NULL,
        [CUST_ID] [int] NULL,
        [CALL_TYPE] [nvarchar](20) NULL,
        [PROD_TYPE] [nvarchar](20) NULL,
        [RATED_AMNT] [int] NULL,
        [RATED_CURR_CODE] [nvarchar](10) NULL,
        [CELL] [int] NULL,
        [CELL_ORIGIN] [int] NULL,
        [HIGH_LOW_RATE] [int] NULL,
        [insert_DATE] [datetime] NULL,
        [update_date] [datetime] NULL)
GO

CREATE TABLE [dbo].[XXCOUNTRYPRE_Mrr](
        [COUNTRY_CODE] [nvarchar](100) NOT NULL,
        [COUNTRY_PRE] [nvarchar](3) NOT NULL)
GO

Create TABLE [dbo].[Call_Type_Mrr](
        [call_type_code] [nvarchar](100) NOT NULL,
        [call_type_desc] [nvarchar](100) NULL,
        [priceperminuter] [Float] NULL,
        [call_type] [nvarchar](100) NULL)
GO
```

```
Create TABLE [dbo].[OPFILEOPP_Mrr](
       [OPCCC] [nvarchar](100) NOT NULL,
       [OPDDD] [nvarchar](100) NULL,
       [prepre] [nvarchar](100) NULL)
GO

------------------------------------- Parssing Origin Number --------------------------
----------
USE [TheVoice_MRR]
GO

Create View vv_PhoneNumberParsing
as
select distinct CUST_NUMBER AS CUST_NUMBER ,
       case
                     when (right(substring(cust_number, 1, (len(cust_number) - 7)), 2)  )
in (select OPCCC from [dbo].[OPFILEOPP_Mrr])
                     then (substring(cust_number, 2, 3) )
                     else (substring(cust_number, 2, 1) )
              end as Country,

       case
                     when (right(substring(cust_number, 1, (len(cust_number) - 7)), 2)  )
in (select OPCCC from [dbo].[OPFILEOPP_Mrr] )
                     then (right(substring(cust_number, 1, (len(cust_number) - 7)), 2)  )
                     else right(substring(cust_number, 1, (len(cust_number) - 7)), 3)
              end as Operator,

              right(cust_number,7) as phone_num
       from customer_Mrr

GO

------------------------------------- Parssing destination Number ---------------------
----------
Create View [dbo].[vv_Phone_Parsing]
AS
select DISTINCT DES_NO AS DES_NO ,
       case
                     when (right(substring(DES_NO, 1, (len(DES_NO) - 7)), 2)) in (select
OPCCC from [dbo].[OPFILEOPP_Mrr])
                     then (substring(DES_NO, 2, 3))

                     when (left(substring(DES_NO, 2,2),1)) in (select COUNTRY_PRE from
[dbo].[XXCOUNTRYPRE_Mrr])
                     then (substring(DES_NO, 2, 1))

                     when (left(substring(DES_NO, 2,3),2)) in (select COUNTRY_PRE from
[dbo].[XXCOUNTRYPRE_Mrr])
                     then (substring(DES_NO, 2, 2))

                     when (left(substring(DES_NO, 2,4),3)) in (select COUNTRY_PRE from
[dbo].[XXCOUNTRYPRE_Mrr])
                     then  (substring(DES_NO, 2, 3))
       end as CountryDes,
```

```
        case
                        when (right(substring(DES_NO, 1, (len(DES_NO) - 9)), 3)  ) = 972
then  (right(substring(DES_NO, 1, (len(DES_NO) - 7)), 2)  )
                        when (right(substring(DES_NO, 1, (len(DES_NO) - 10)), 1)  ) = 1
then  (right(substring(DES_NO, 1, (len(DES_NO) - 7)), 3)  )
                        else (right(substring(DES_NO, 1, (len(DES_NO) - 7)), 3)  )
        end as OperatorDes,

                right(DES_NO,7) as Phone_Num
from USAGE_MAIN_Mrr
GO

------------------------------------- Phone parsing for the Loop Script ----------------
----------
CREATE View [dbo].[vv_Origin_NO_Parsing]
as
select DISTINCT PHONE_NO AS Origin_NO ,
        case
                        when (right(substring(PHONE_NO, 1, (len(PHONE_NO) - 7)), 2)  ) in
(select OPCCC from [dbo].[OPFILEOPP_Mrr])
                        then (substring(PHONE_NO, 2, 3) )
                        else (substring(PHONE_NO, 2, 1) )
                end as CountryOrigin,

        case
                        when (right(substring(PHONE_NO, 1, (len(PHONE_NO) - 7)), 2)  ) in
(select OPCCC from [dbo].[OPFILEOPP_Mrr] )
                        then (right(substring(PHONE_NO, 1, (len(PHONE_NO) - 7)), 2)  )
                        else right(substring(PHONE_NO, 1, (len(PHONE_NO) - 7)), 3)
                end as OperatorOrigin,

                right(PHONE_NO,7) as phone_num
        from Customer_lines_Mrr
GO

--------- Join Phone Number, Country & Operator with cleaning of double Country prefixes
----------
create View VV_PHONE_COUNRTY_OPP
as
select VV.CUST_NUMBER AS CUST_NUMBER , VV.Country , XX.COUNTRY_CODE , VV.Operator ,
OM.OPDDD
from [vv_PhoneNumberParsing] VV
left join XXCOUNTRYPRE_Mrr XX
on
XX.COUNTRY_PRE = VV.Country
left join  OPFILEOPP_Mrr OM
on
VV.Operator = OM.OPCCC
where COUNTRY_CODE not in ('puerto rico' , 'Canada','Kazakhstan','Christmas
Island','Cocos (Keeling) Islands','Holy See (Vatican City)')

GO

------------------------------------ View to Reveal old duplicates --------------------
----------
create view VV_Cancel_Duplicates
```

```
as
SELECT *
from customer_Mrr
    WHERE customer_ID NOT IN
    (
        SELECT max(customer_ID)
        FROM customer_Mrr
        GROUP BY cust_number
    )

GO
-------------------------- View to Reveal Customers Without Duplicates ----------------
----------
Create View VV_Without_Duplicates
as
       (select customer_id , CUST_NUMBER from Customer_Mrr
       where customer_id not in (select customer_id from VV_Cancel_Duplicates))
go
----------------------------- Rundom Number Function & View ---------------------------
----------
CREATE function [dbo].[FN_RandomValues](@Lower int, @Upper int)
returns int
as
Begin

DECLARE @Random INT;
if @Upper > @Lower
       SELECT @Random = (1 + @Upper - @Lower) * (SELECT [Value] FROM vv_GetRandomValue) +
@Lower
Else
       SELECT @Random = (1 + @Lower - @Upper) * (SELECT [Value] FROM vv_GetRandomValue) +
@Upper
return @Random
end
GO

-------------------------------------------------------------------------------------
----------
create view [dbo].[vv_GetRandomValue]
as
select rand() as [value]
GO

----------------------------- Creation of The Voice STG DB-----------------------------
----------
CREATE DATABASE [TheVoice_STG]
GO

USE [TheVoice_STG]
GO

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO
```

```
---------------------------------- First Creation of Tables in TheVoice_STG ------------
----------
CREATE TABLE [dbo].[Customers_STG](
	[customer_id] [int] NOT NULL,
	[PHONE_NO] [nvarchar](50) NOT NULL,
	[cust_name] [nvarchar](100) NOT NULL,
	[address] [nvarchar](100) NOT NULL,
	[DESC] [nvarchar](100) NOT NULL,
	[COUNTRY_CODE] [nvarchar](100) NOT NULL,
	[OPDDD] [nvarchar](50) NOT NULL)
Go

CREATE TABLE [dbo].[Customer_Lines_STG](
	[PHONE_NO] [nvarchar](20) NOT NULL,
	[DESC] [nvarchar](100) NOT NULL,
	[discountpct] [int] NOT NULL,
	[TYPE] [nvarchar](10) NOT NULL,
	[numberoffreeminutes] [int] NOT NULL)
GO

CREATE TABLE [dbo].[Package_Catalog_STG](
	[PACKAGE_NUM] [int] NOT NULL,
	[pack_type] [nvarchar](10) NOT NULL,
	[pack_desc] [nvarchar](100) NOT NULL,
	[createdate] [datetime] NOT NULL,
	[enddate] [datetime] NOT NULL,
	[status] [nvarchar](10) NOT NULL,
	[ActivitiesDays] [int] NOT NULL)
Go

CREATE TABLE [dbo].[USAGE_MAIN_STG](
	[CALL_NO] [int] NOT NULL,
	[SEIZED_TIME] [time] NOT NULL,
	[SEIZED_date] [date] NOT NULL,
	[CALLING_NO] [nvarchar](18) NULL,
	[Country] int NOT NULL,
	[Operator] int NOT NULL,
	[DES_NO] [nvarchar](25) NOT NULL,
	[CountryDes] int NOT NULL,
	[OperatorDes] int NOT NULL,
	[DURATION] [int] NOT NULL,
	[CUST_ID] [int] NOT NULL,
	[CALL_TYPE] [nvarchar](20) NOT NULL,
	[RATED_AMNT] [float] NOT NULL,
	[CELL_ORIGIN] [int] NOT NULL,
	[PACKAGE_NUM] [int] NOT NULL,
	[Updated_Duration] [int] NOT NULL,
	[Updated_RATED_AMNT] [Float] NOT NULL)
GO

CREATE TABLE [dbo].[XXCOUNTRYPRE_STG](
	[COUNTRY_CODE] [nvarchar](100) NOT NULL,
	[COUNTRY_PRE] int NOT NULL,
	[REGION] [nvarchar](100) NOT NULL,
	[AREA] [nvarchar](100) NOT NULL)
```

```
GO

Create TABLE [dbo].[Call_Type_STG](
       [call_type_code] [nvarchar](100) NOT NULL,
       [call_type_desc] [nvarchar](100) NOT NULL,
       [priceperminuter] [nvarchar](30) NOT NULL,
       [call_type] [nvarchar](100) NOT NULL,
       [Code&Desc] [nvarchar](100) NOT NULL)
GO

Create TABLE [dbo].[OPFILEOPP_STG](
       [OPCCC] int NOT NULL,
       [prepre&OPDDD] [nvarchar](50) NOT NULL,
       [prepre] [nvarchar](3) NOT NULL)
GO

Create Table [Managment Table](
       [Key] int identity(1,1) Primary Key NOT NULL,
       ParameterName nvarchar(20) NOT NULL,
       ParameterDescription nvarchar(100) NOT NULL,
       ParameterValue float NOT NULL)
GO

insert into [Managment Table]
values ('DiscountType' , 'This Parameter will deside if priceperminute is a DISCOUNTED
PRICE or NORMAL PRICE ' , 0.5)
GO

-------------- create surrogate table IN TheVoice_STG DB Date_all for any option date ---
----------
------------------------------------- dates for Dim_Date ------------------------------
-----------
CREATE  TABLE [dbo].[Date_All]
       (      [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,
            [IsHolidayUSA] BIT,-- Flag 1=National Holiday, 0-No National Holiday
            [IsWeekdayStyle1] BIT,-- 0=Week End ,1=Week Day
            [IsWeekdayStyle2] BIT,-- 0=Week End ,1=Week Day
            [HolidayUSA] VARCHAR(50),--Name of Holiday in US
            [IsHolidayUK] BIT Null,-- Flag 1=National Holiday, 0-No National Holiday
            [HolidayUK] VARCHAR(50) Null --Name of Holiday in UK
      )
GO
/*******************************************************************************
****/
--Specify Start Date and End date here
--Value of Start Date Must be Less than Your End Date

DECLARE @StartDate DATETIME = (SELECT min([SEIZED_TIME]) FROM
[TheVoice].[dbo].[USAGE_MAIN]) --Starting value of Date Range
DECLARE @EndDate DATETIME = (SELECT max([SEIZED_TIME]) FROM
[TheVoice].[dbo].[USAGE_MAIN]) --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].[Date_All]
```

```
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 IsHolidayUSA,
        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 IsWeekdayStyle1,
        CASE DATEPART(DW, @CurrentDate)
                WHEN 1 THEN 1
                WHEN 2 THEN 1
                WHEN 3 THEN 1
                WHEN 4 THEN 1
                WHEN 5 THEN 1
                WHEN 6 THEN 0
                WHEN 7 THEN 0
                END AS IsWeekdayStyle2,
        NULL AS HolidayUSA, Null, Null

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

go

---------- create surrogate table IN TheVoice_STG DB Time24 for any option in 24 Hr -----
----------
------------------------------ Time For dim_Time ---------------------------------------
----------
DROP TABLE IF EXISTS [dbo].[Time24]
GO
CREATE TABLE [dbo].[Time24]
(
      Id                    INT         NOT NULL CONSTRAINT PKC_DimTime PRIMARY KEY
CLUSTERED
```

```
    , Hour24                INT         NOT NULL
    , Hour24ShortString     VARCHAR(2)  NOT NULL
    , Hour24MinString       VARCHAR(5)  NOT NULL
    , Hour24FullString      VARCHAR(8)  NOT NULL
    , Hour12                INT         NOT NULL
    , Hour12ShortString     VARCHAR(2)  NOT NULL
    , Hour12MinString       VARCHAR(5)  NOT NULL
    , Hour12FullString      VARCHAR(8)  NOT NULL
    , AmPmCode              INT         NOT NULL
    , AmPmString            VARCHAR(2)  NOT NULL
    , Minute                INT         NOT NULL
    , MinuteCode            INT         NOT NULL
    , MinuteShortString     VARCHAR(2)  NOT NULL
    , MinuteFullString24    VARCHAR(8)  NOT NULL
    , MinuteFullString12    VARCHAR(8)  NOT NULL
    , HalfHour              INT         NOT NULL
    , HalfHourCode          INT         NOT NULL
    , HalfHourShortString   VARCHAR(2)  NOT NULL
    , HalfHourFullString24  VARCHAR(8)  NOT NULL
    , HalfHourFullString12  VARCHAR(8)  NOT NULL

)
GO

DECLARE   @hour     INT
        , @minute   INT

SET @hour = 0

WHILE @hour < 24
BEGIN

    SET @minute = 0

    WHILE @minute < 60
    BEGIN

            INSERT INTO [dbo].[Time24]
            (
                 Id
                , Hour24
                , Hour24ShortString
                , Hour24MinString
                , Hour24FullString
                , Hour12
                , Hour12ShortString
                , Hour12MinString
                , Hour12FullString
                , AmPmCode
                , AmPmString
                , Minute
                , MinuteCode
                , MinuteShortString
                , MinuteFullString24
                , MinuteFullString12
                , HalfHour
```

```
                , HalfHourCode
                , HalfHourShortString
                , HalfHourFullString24
                , HalfHourFullString12
            )
            SELECT
                  (@hour*100) + (@minute*1) AS TimeKey
                , @hour AS Hour24
                , RIGHT('0'+CONVERT(VARCHAR(2),@hour),2) Hour24ShortString
                , RIGHT('0'+CONVERT(VARCHAR(2),@hour),2)+':00' Hour24MinString
                , RIGHT('0'+CONVERT(VARCHAR(2),@hour),2)+':00:00' Hour24FullString
                , @hour%12 AS Hour12
                , RIGHT('0'+CONVERT(VARCHAR(2),@hour%12),2) Hour12ShortString
                , RIGHT('0'+CONVERT(VARCHAR(2),@hour%12),2)+':00' Hour12MinString
                , RIGHT('0'+CONVERT(VARCHAR(2),@hour%12),2)+':00:00' Hour12FullString
                , @hour/12 AS AmPmCode
                , CASE WHEN @hour<12 THEN 'AM' ELSE 'PM' END AS AmPmString
                , @minute AS MINUTE
                , (@hour*100) + (@minute) MinuteCode
                , RIGHT('0'+CONVERT(VARCHAR(2),@minute),2) MinuteShortString
                , RIGHT('0'+CONVERT(VARCHAR(2),@hour),2)+':'+
                  RIGHT('0'+CONVERT(VARCHAR(2),@minute),2)+':00' MinuteFullString24
                , RIGHT('0'+CONVERT(VARCHAR(2),@hour%12),2)+':'+
                  RIGHT('0'+CONVERT(VARCHAR(2),@minute),2)+':00' MinuteFullString12
                , @minute/30 AS HalfHour
                , (@hour*100) + ((@minute/30)*30) HalfHourCode
                , RIGHT('0'+CONVERT(VARCHAR(2),((@minute/30)*30)),2) HalfHourShortString
                , RIGHT('0'+CONVERT(VARCHAR(2),@hour),2)+':'+
                  RIGHT('0'+CONVERT(VARCHAR(2),((@minute/30)*30)),2)+':00'
HalfHourFullString24
                , RIGHT('0'+CONVERT(VARCHAR(2),@hour%12),2)+':'+
                  RIGHT('0'+CONVERT(VARCHAR(2),((@minute/30)*30)),2)+':00'
HalfHourFullString12

        SET @minute = @minute + 1
    END
    SET @hour = @hour + 1
END
GO
------------------------- SQL script for allocating the Discount value to a Parameter ---
----------
Select ParameterValue
from [Managment Table]
Where [Key] = 1
GO

----------------------------- Creation of The Voice DW DB -----------------------------
----------
CREATE DATABASE [TheVoice_DW]
GO

USE [TheVoice_DW]
GO

SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO

---------------------------- Fist Creation of Tables in TheVoice_DW ---------------------
----------
CREATE TABLE [dbo].[Dim_Countries](
	[CountrySK] int IDENTITY(1,1) PRIMARY KEY NOT NULL,
	[KeyCountry] int NOT NULL,
	[DESCCOUNTRY] [nvarchar](100) NOT NULL,
	[DESCREGION] [nvarchar](100) NOT NULL,
	[DESCAREA] [nvarchar](100) NOT NULL)

GO

CREATE TABLE [dbo].[Dim_Customers](
	[CustomersSK] int IDENTITY(1,1) PRIMARY KEY NOT NULL,
	[KeyCustomer] int NOT NULL,
	[CustomerPhoneNumber] [nvarchar](50) NOT NULL,
	[DescCustomerLineOperator] [nvarchar] (50) NOT NULL,
	[DescCustomerLineCountry] [nvarchar] (100) NOT NULL,
	[DescCustomerName] [nvarchar] (100) NOT NULL,
	[DescCustomerAddress] [nvarchar] (100) NOT NULL,
	[DescCusomterPackage] [nvarchar] (100) NOT NULL)
GO

CREATE TABLE [dbo].[Dim_PackageCatalog](
	[PackagesSK] int IDENTITY(1,1) PRIMARY KEY NOT NULL,
	[KeyPackage] int NOT NULL,
	[PackageType] [nvarchar] (10) NOT NULL,
	[DescPackage] [nvarchar](120) NOT NULL,
	[DatePackageCreation] Datetime NOT NULL,
	[DatePackageEnd] Datetime NOT NULL,
	[DescPackageStatus] [nvarchar](100) NOT NULL,
	[CodePackageActivitiesDays] int NOT NULL)
GO

Create TABLE [dbo].[Dim_CallTypes](
	[KeyCallTypeSK] int IDENTITY(1,1) PRIMARY KEY NOT NULL,
	[DescCallTypeCode] [nvarchar] (100) NOT NULL,
	[DescCallType] [nvarchar] (100) NOT NULL,
	[DescFullCallType] [nvarchar] (100) NOT NULL,
	[DescCallTypePriceCategory] [nvarchar] (100) NOT NULL,
	[DescCallTypeCategory] [nvarchar] (100) NOT NULL)
GO

Create TABLE [dbo].[Dim_Operators](
	[OperatorsSK] int IDENTITY(1,1) PRIMARY KEY NOT NULL,
	[KeyOperator] int NOT NULL,
	[DescOperator] [nvarchar](50) NOT NULL,
	[DescKeyPrefix] [nvarchar](3) NOT NULL)
GO

Create TABLE [dbo].[Dim_CallOriginType](
	[CallOriginSK] int IDENTITY(1,1) PRIMARY KEY NOT NULL,
	[KeyCallOriginType] int NOT NULL,
```

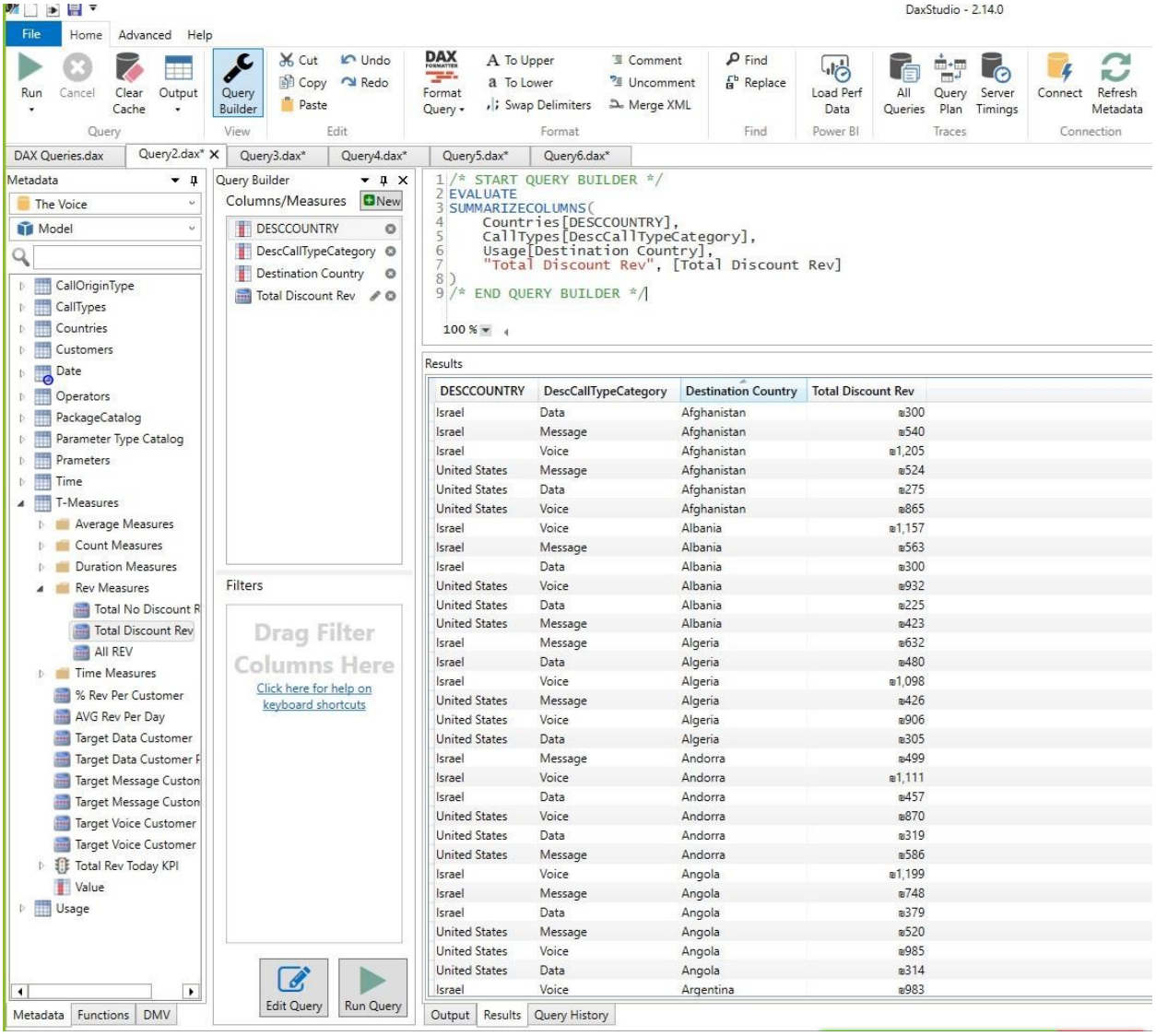
```
        [DescCallOriginType] nvarchar(100) NOT NULL)
GO

Create TABLE [dbo].[Dim_Date](
        [KeyDateSK] int PRIMARY KEY NOT NULL,
        [FullDate] date NOT NULL,
        [KeyYear] int NOT NULL,
        [CodeYear] int NOT NULL,
        [DescYear]    nvarchar(50) NOT NULL,
        [CodeQrt] int NOT NULL,
        [DescQrt]     nvarchar(50) NOT NULL,
        [KeyMonth] int NOT NULL,
        [CodeMonth] int NOT NULL,
        [DescMonth] nvarchar(50) NOT NULL,
        [CodeDayInWeek]      int NOT NULL,
        [DescDayInWeek]      nvarchar(50) NOT NULL)
GO

Create TABLE [dbo].[Dim_Time](
        [KeyTimeSK]   int PRIMARY KEY NOT NULL,
        [FullTime] time (0) NOT NULL,
        [CodeHour] int NOT NULL,
        [DescHour] nvarchar(50) NOT NULL,
        [KeyMinute]   int NOT NULL,
        [CodeMinute] int NOT NULL,
        [DescMinute] nvarchar(50)NOT NULL)
GO

CREATE TABLE [dbo].[Fact_Usage](
        [CallIDSK] int IDENTITY(1,1) PRIMARY KEY NOT NULL,
        [KeyCustomer] int NOT NULL,
        [KeyCallType] int NOT NULL,
        [KeyOriginCountry] int  NOT NULL,
        [KeyDestinationCountry] int  NOT NULL,
        [KeyOriginOperator] int  NOT NULL,
        [KeyDestinationOperator] int  NOT NULL,
        [KeyPackage] int  NOT NULL,
        [KeyCallOriginType] int NOT NULL,
        [KeyCallDate] int NOT NULL,
        [KeyCallTime] int NOT NULL,
        [Duration] int NOT NULL,
        [BillableDuration] int NOT NULL,
        [Amount] float NOT NULL,
        [BillableAmount] float NOT NULL)
GO

---------------------------- Dormant Users View - for SSRS Report -----------------------
----------
Create View vv_Dorment_Customers
AS
Select *
from Dim_Customers C2
Where C2.KeyCustomer  not in
                                                        (
                                                                select C1.KeyCustomer
                                                                from Fact_Usage FU
```

```
                                                        join Dim_Customers C1
                                                        on FU.KeyCustomer = C1.CustomersSK
                                                )
GO

---------------------------------------- Kpi Queries ----------------------------------
----------
--Value
Select sum(BillableAmount * BillableDuration) as RevenueCurrentQrt
from Fact_Usage F
join Dim_Date D
on F.keycalldate = D.KeyDateSK
where D.CodeYear = datepart(yyyy,DateAdd(yyyy,-7,GETDATE())) and D.CodeQrt =
datepart(qq,(DateAdd(yyyy,-7,GETDATE())))
Go

--Target
Select sum(BillableAmount * BillableDuration) as RevenueLastQrt
from Fact_Usage F
join Dim_Date D
on F.keycalldate = D.KeyDateSK
where D.CodeYear = datepart(yyyy,DateAdd(yyyy,-7,GETDATE())) and D.CodeQrt =
datepart(qq,(DateAdd(qq,-85,GETDATE())))

--Status
With Cte1 as
(
       Select sum(BillableAmount * BillableDuration) as RevenueCurrentQrt
       from Fact_Usage F
       join Dim_Date D
       on F.keycalldate = D.KeyDateSK
       where D.CodeYear = datepart(yyyy,DateAdd(yyyy,-7,GETDATE())) and D.CodeQrt =
datepart(qq,(DateAdd(yyyy,-7,GETDATE())))
)
,Cte2 as
(
       Select sum(BillableAmount * BillableDuration) as RevenueLastQrt
       from Fact_Usage F
       join Dim_Date D
       on F.keycalldate = D.KeyDateSK
       where D.CodeYear = datepart(yyyy,DateAdd(yyyy,-7,GETDATE())) and D.CodeQrt =
datepart(qq,(DateAdd(qq,-85,GETDATE())))
)
Select
       case
              when RevenueCurrentQrt > RevenueLastQrt then 1
              when RevenueCurrentQrt = RevenueLastQrt then 0
              when RevenueCurrentQrt < RevenueLastQrt then -1
              end as [Status]
from Cte1
cross join Cte2
GO

------------------------------ LOOP Creation For Usage Main ---------------------------
----------
Declare
```

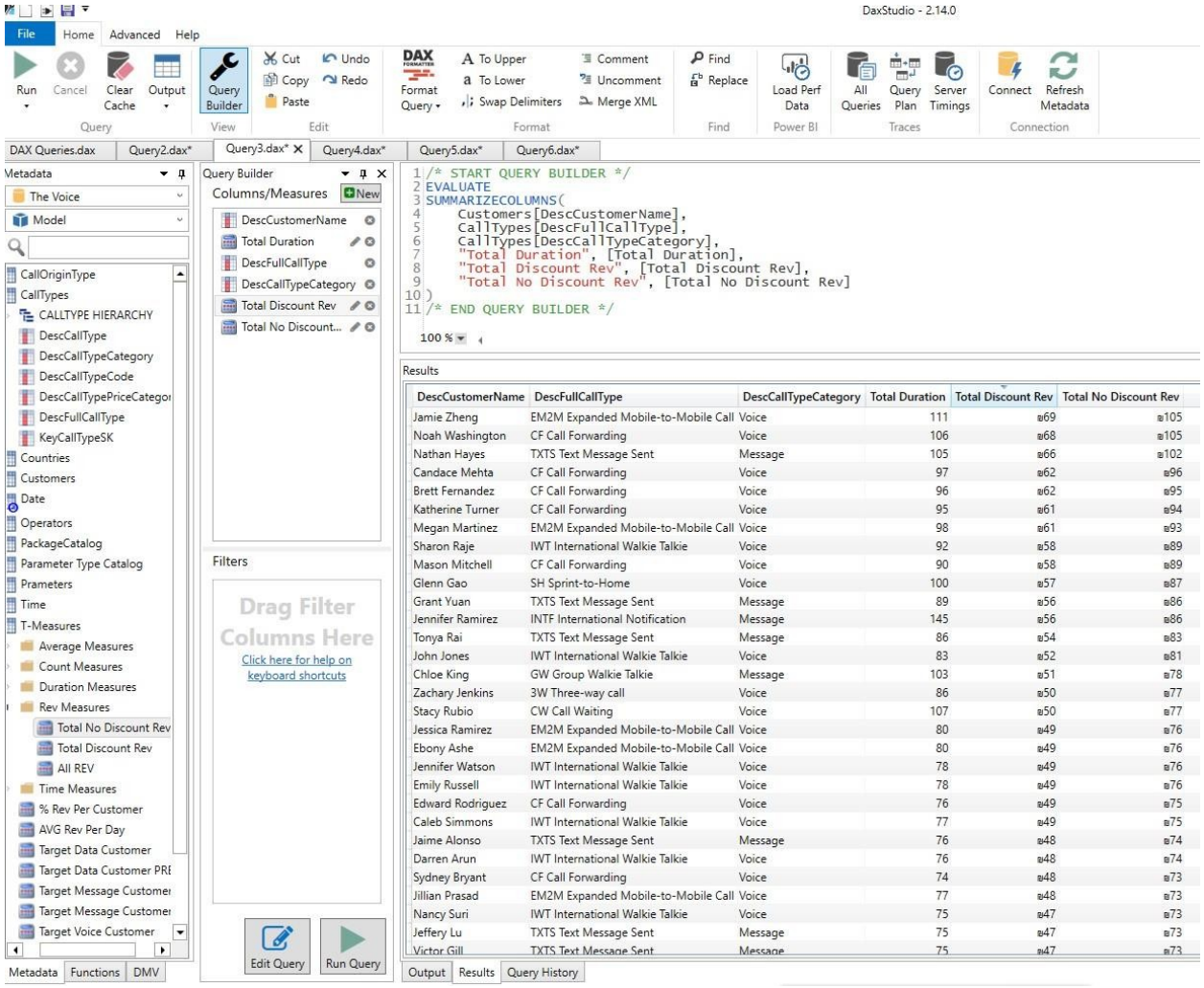
```
        @Dcount int ,                       --number of total days
        @Ccount int ,                       --number of calls per day
        @Rcount int ,                       --random number of calls in a single day

        @CallNo int ,
        @Ctime datetime ,                   --time call made
        @Atime datetime ,                   --time call answer
        @Dtime datetime ,                   --time call disconnected
        @Cdate datetime ,                   --date of call
        @custID int   ,                             --customer ID from customer table
        @Cnum nvarchar(18) ,         --customer phone number from customer table
        @Dnum nvarchar(25) ,         --destination number (randomal)
        @Dnum2 nvarchar(18) ,               --destination number (randomal)
        @Ctype nvarchar(20) ,               --call type - from calltype table
        @Dtype nvarchar(20) ,               --call type discription - from calltype table
        @Amt int ,                                  --between 0-6
        @Cell int ,                                 --israeli operator 1 other 0
        @CellOrigin int ,                   --israeli operator 1 other 0
        @Dur int ,                                  --duration
        @ratedcurr nvarchar (10) ,
------- viriables for genarate DES_NO and CALLED_NO -------
        @country  int ,
        @DesOP  int ,
        @OriginOP int ,
        @number int ,
        @DES_NO nvarchar(20)
------------ starting points viriables --------------------
set @Dcount = 1

set @Ctime = '00:00:00'
set @Cdate = (select convert(date,max(SEIZED_TIME)) from USAGE_MAIN)
set @ratedcurr = 'SHEKEL'
set @CallNo = ((select top 1 call_no from [TheVoice].[dbo].[USAGE_MAIN] order by CALL_NO
desc) + 1)

while @Dcount <= 1369
begin
        print @Dcount
        set @Cdate = DATEADD(DAY, 0, @Cdate) print @cdate
        set @Rcount = (Select Floor(Rand()*(100 - 50) )+ 50) print @Rcount
        set @Ccount = 1
        while @Ccount <=  @Rcount
                begin
                print @Ccount
                        Set @custID = (
                                                SELECT TOP 1 customer_id
                                                FROM
[TheVoice_MRR].[dbo].[VV_without_Duplicates]                                    --
select a random customer id
                                                ORDER BY NEWID()
                                                )
                        Set @Cnum = (
                                                SELECT CUST_NUMBER
                                                FROM
[TheVoice_MRR].[dbo].[VV_without_Duplicates]                                    --
select the phone number of the customer id
```

```
                                        where @custID = customer_id
                                        )
                    set @OriginOP = (
                                        select OperatorOrigin
                                        from
[TheVoice_MRR].[dbo].[vv_Origin_NO_Parsing]                                --select the
operation number
                                        where Origin_NO = @Cnum
                                        )

                    if @OriginOP in (
                                        select [OPCCC]
                                        from [TheVoice_MRR].[dbo].[OPFILEOPP_Mrr]
                                        )
                                        set @CellOrigin=1 else set @CellOrigin=0
GO
```

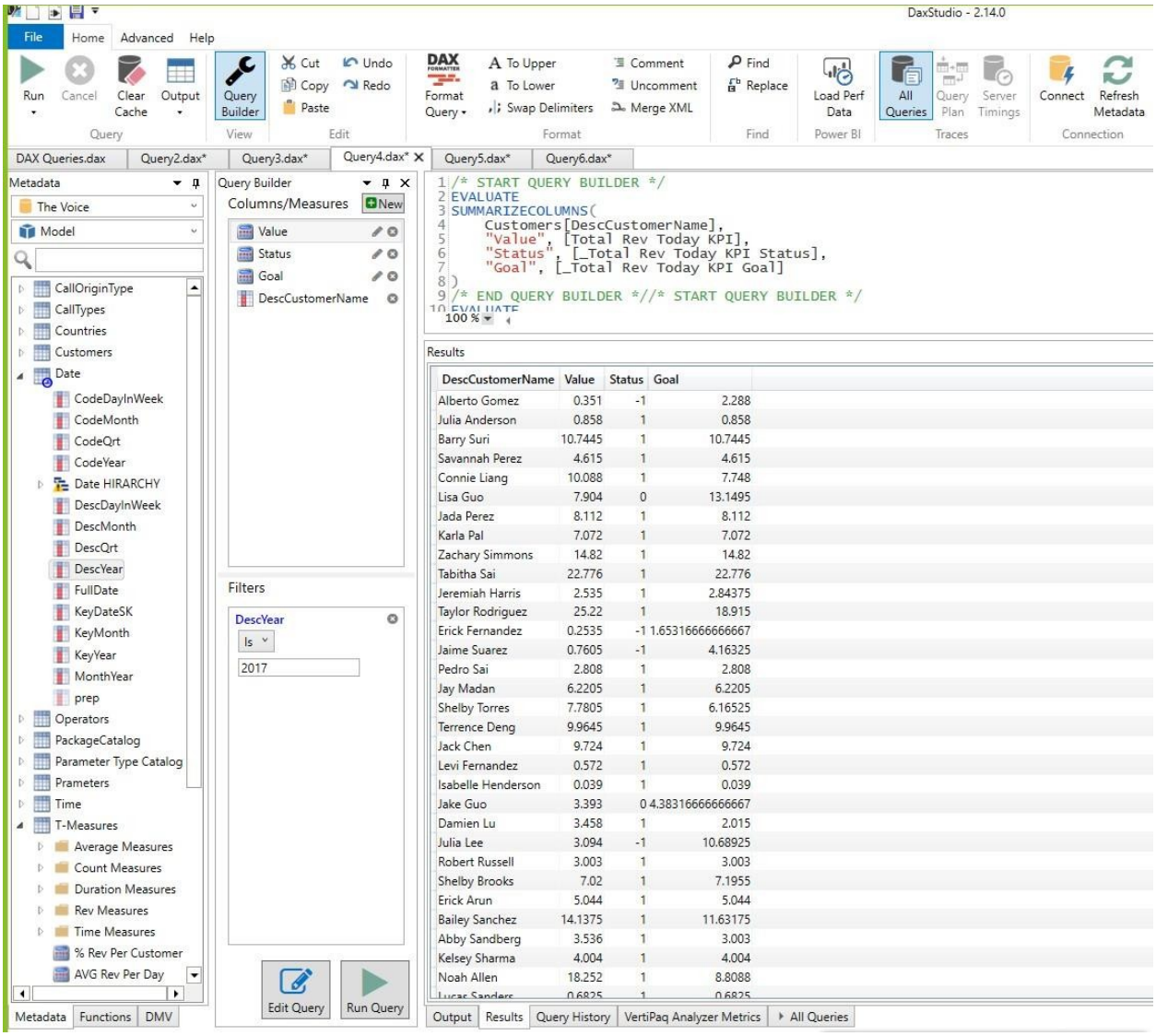
Appendix G – Dax Queries



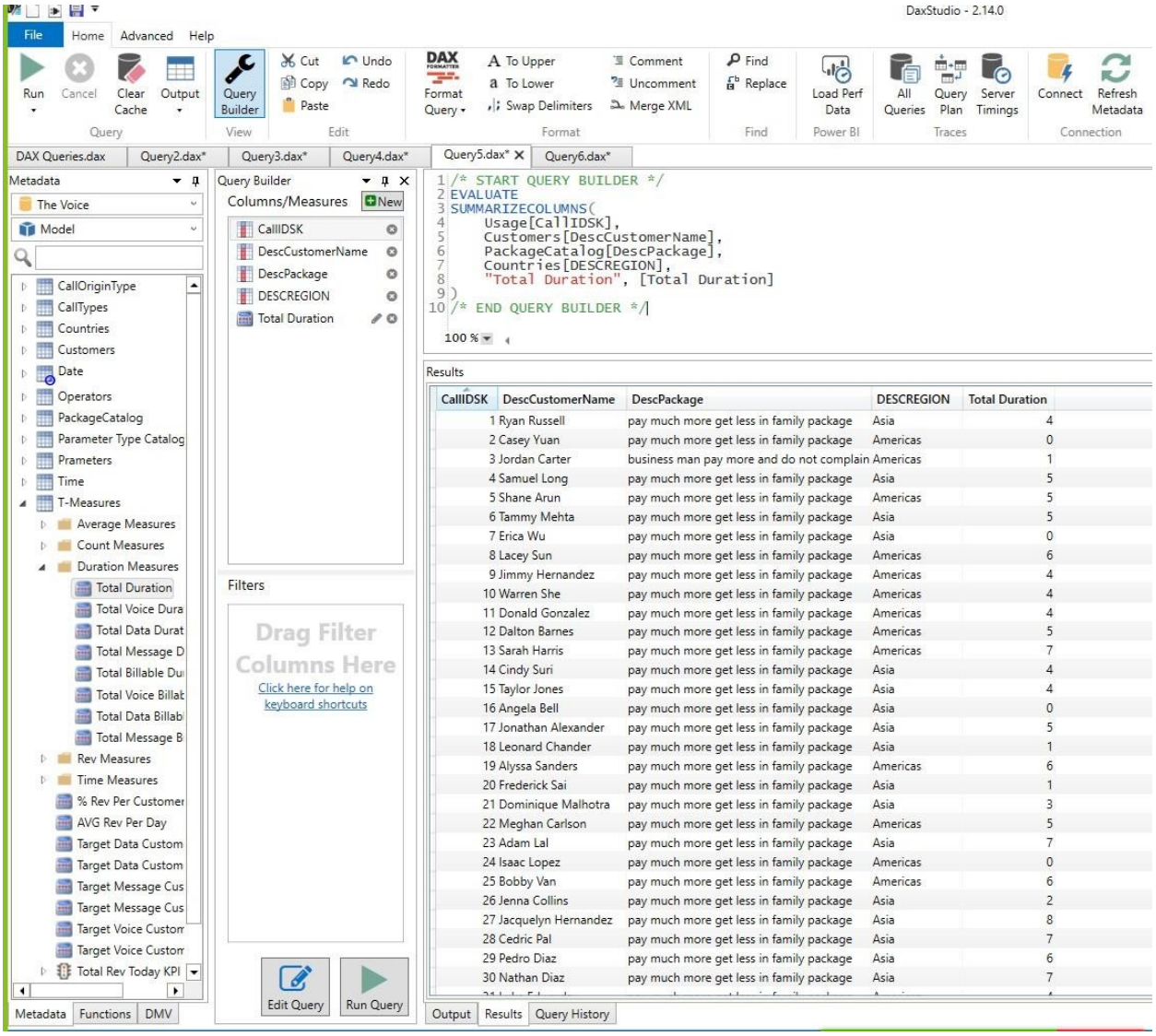
DESCCOUNTRY	DescCallTypeCategory	Destination Country	Total Discount Rev
Israel	Data	Afghanistan	₪300
Israel	Message	Afghanistan	₪540
Israel	Voice	Afghanistan	₪1,205
United States	Message	Afghanistan	₪524
United States	Data	Afghanistan	₪275
United States	Voice	Afghanistan	₪865
Israel	Voice	Albania	₪1,157
Israel	Message	Albania	₪563
Israel	Data	Albania	₪300
United States	Voice	Albania	₪932
United States	Data	Albania	₪225
United States	Message	Albania	₪423
Israel	Message	Algeria	₪632
Israel	Data	Algeria	₪480
Israel	Voice	Algeria	₪1,098
United States	Message	Algeria	₪426
United States	Voice	Algeria	₪906
United States	Data	Algeria	₪305
Israel	Message	Andorra	₪499
Israel	Voice	Andorra	₪1,111
Israel	Data	Andorra	₪457
United States	Voice	Andorra	₪870
United States	Data	Andorra	₪319
United States	Message	Andorra	₪586
Israel	Voice	Angola	₪1,199
Israel	Message	Angola	₪748
Israel	Data	Angola	₪379
United States	Message	Angola	₪520
United States	Voice	Angola	₪985
United States	Data	Angola	₪314
Israel	Voice	Argentina	₪983



DescCustomerName	DescFullCallType	DescCallTypeCategory	Total Duration	Total Discount Rev	Total No Discount Rev
Jamie Zheng	EM2M Expanded Mobile-to-Mobile Call	Voice	111	¤69	¤105
Noah Washington	CF Call Forwarding	Voice	106	¤68	¤105
Nathan Hayes	TXTS Text Message Sent	Message	105	¤66	¤102
Candace Mehta	CF Call Forwarding	Voice	97	¤62	¤96
Brett Fernandez	CF Call Forwarding	Voice	96	¤62	¤95
Katherine Turner	CF Call Forwarding	Voice	95	¤61	¤94
Megan Martinez	EM2M Expanded Mobile-to-Mobile Call	Voice	98	¤61	¤93
Sharon Raje	IWT International Walkie Talkie	Voice	92	¤58	¤89
Mason Mitchell	CF Call Forwarding	Voice	90	¤58	¤89
Glenn Gao	SH Sprint-to-Home	Voice	100	¤57	¤87
Grant Yuan	TXTS Text Message Sent	Message	89	¤56	¤86
Jennifer Ramirez	INTF International Notification	Message	145	¤56	¤86
Tonya Rai	TXTS Text Message Sent	Message	86	¤54	¤83
John Jones	IWT International Walkie Talkie	Voice	83	¤52	¤81
Chloe King	GW Group Walkie Talkie	Message	103	¤51	¤78
Zachary Jenkins	3W Three-way call	Voice	86	¤50	¤77
Stacy Rubio	CW Call Waiting	Voice	107	¤50	¤77
Jessica Ramirez	EM2M Expanded Mobile-to-Mobile Call	Voice	80	¤49	¤76
Ebony Ashe	EM2M Expanded Mobile-to-Mobile Call	Voice	80	¤49	¤76
Jennifer Watson	IWT International Walkie Talkie	Voice	78	¤49	¤76
Emily Russell	IWT International Walkie Talkie	Voice	78	¤49	¤76
Edward Rodriguez	CF Call Forwarding	Voice	76	¤49	¤75
Caleb Simmons	IWT International Walkie Talkie	Voice	77	¤49	¤75
Jaime Alonso	TXTS Text Message Sent	Message	76	¤48	¤74
Darren Arun	IWT International Walkie Talkie	Voice	76	¤48	¤74
Sydney Bryant	CF Call Forwarding	Voice	74	¤48	¤73
Jillian Prasad	EM2M Expanded Mobile-to-Mobile Call	Voice	77	¤48	¤73
Nancy Suri	IWT International Walkie Talkie	Voice	75	¤47	¤73
Jeffery Lu	TXTS Text Message Sent	Message	75	¤47	¤73
Victor Gill	TXTS Text Message Sent	Message	75	¤47	¤73



DescCustomerName	Value	Status	Goal
Alberto Gomez	0.351	-1	2.288
Julia Anderson	0.858	1	0.858
Barry Suri	10.7445	1	10.7445
Savannah Perez	4.615	1	4.615
Connie Liang	10.088	1	7.748
Lisa Guo	7.904	0	13.1495
Jada Perez	8.112	1	8.112
Karla Pal	7.072	1	7.072
Zachary Simmons	14.82	1	14.82
Tabitha Sai	22.776	1	22.776
Jeremiah Harris	2.535	1	2.84375
Taylor Rodriguez	25.22	1	18.915
Erick Fernandez	0.2535	-1	1.65316666666667
Jaime Suarez	0.7605	-1	4.16325
Pedro Sai	2.808	1	2.808
Jay Madan	6.2205	1	6.2205
Shelby Torres	7.7805	1	6.16525
Terrence Deng	9.9645	1	9.9645
Jack Chen	9.724	1	9.724
Levi Fernandez	0.572	1	0.572
Isabelle Henderson	0.039	1	0.039
Jake Guo	3.393	0	4.38316666666667
Damien Lu	3.458	1	2.015
Julia Lee	3.094	-1	10.68925
Robert Russell	3.003	1	3.003
Shelby Brooks	7.02	1	7.1955
Erick Arun	5.044	1	5.044
Bailey Sanchez	14.1375	1	11.63175
Abby Sandberg	3.536	1	3.003
Kelsey Sharma	4.004	1	4.004
Noah Allen	18.252	1	8.8088
Lucas Sanders	0.6825	1	0.6825


DaxStudio - 2.14.0
File
Home
Advanced
Help
Run
Cancel
Clear Cache
Output
Query
Query Builder
View
Cut
Copy
Paste
Undo
Redo
Edit
Format Query
To Upper
To Lower
Swap Delimiters
Comment
Uncomment
Merge XML
Format
Find
Replace
Load Perf Data
Power BI
All Queries
Query Plan
Server Timings
Traces
Connect
Refresh Metadata
Connection
DAX Queries.dax
Query2.dax*
Query3.dax*
Query4.dax*
Query5.dax*
Query6.dax*
Metadata
The Voice
Model
CallOriginType
CallTypes
Countries
Customers
Date
Operators
PackageCatalog
Parameter Type Catalog
Prameters
Time
T-Measures
Average Measures
Count Measures
Duration Measures
Total Duration
Total Voice Dura
Total Data Durat
Total Message D
Total Billable Du
Total Voice Billat
Total Data Billab
Total Message B
Rev Measures
Time Measures
% Rev Per Customer
AVG Rev Per Day
Target Data Custom
Target Data Custom
Target Message Cus
Target Message Cus
Target Voice Custom
Target Voice Custom
Total Rev Today KPI
Metadata
Functions
DMV
Query Builder
Columns/Measures
New
CallIDSK
DescCustomerName
DescPackage
DESCREGION
Total Duration
Filters
Drag Filter Columns Here
Click here for help on keyboard shortcuts
Edit Query
Run Query
/* START QUERY BUILDER */
EVALUATE
SUMMARIZECOLUMNS(
Usage[CallIDSK],
Customers[DescCustomerName],
PackageCatalog[DescPackage],
Countries[DESCREGION],
"Total Duration", [Total Duration]
)
/* END QUERY BUILDER */
100 %
Results
CallIDSK | DescCustomerName | DescPackage | DESCREGION | Total Duration
1 | Ryan Russell | pay much more get less in family package | Asia | 4
2 | Casey Yuan | pay much more get less in family package | Americas | 0
3 | Jordan Carter | business man pay more and do not complain | Americas | 1
4 | Samuel Long | pay much more get less in family package | Asia | 5
5 | Shane Arun | pay much more get less in family package | Americas | 5
6 | Tammy Mehta | pay much more get less in family package | Asia | 5
7 | Erica Wu | pay much more get less in family package | Asia | 0
8 | Lacey Sun | pay much more get less in family package | Americas | 6
9 | Jimmy Hernandez | pay much more get less in family package | Americas | 4
10 | Warren She | pay much more get less in family package | Americas | 4
11 | Donald Gonzalez | pay much more get less in family package | Americas | 4
12 | Dalton Barnes | pay much more get less in family package | Americas | 5
13 | Sarah Harris | pay much more get less in family package | Americas | 7
14 | Cindy Suri | pay much more get less in family package | Asia | 4
15 | Taylor Jones | pay much more get less in family package | Asia | 4
16 | Angela Bell | pay much more get less in family package | Asia | 0
17 | Jonathan Alexander | pay much more get less in family package | Asia | 5
18 | Leonard Chander | pay much more get less in family package | Asia | 1
19 | Alyssa Sanders | pay much more get less in family package | Americas | 6
20 | Frederick Sai | pay much more get less in family package | Asia | 1
21 | Dominique Malhotra | pay much more get less in family package | Asia | 3
22 | Meghan Carlson | pay much more get less in family package | Americas | 5
23 | Adam Lal | pay much more get less in family package | Asia | 7
24 | Isaac Lopez | pay much more get less in family package | Americas | 0
25 | Bobby Van | pay much more get less in family package | Americas | 6
26 | Jenna Collins | pay much more get less in family package | Asia | 2
27 | Jacquelyn Hernandez | pay much more get less in family package | Asia | 8
28 | Cedric Pal | pay much more get less in family package | Asia | 7
29 | Pedro Diaz | pay much more get less in family package | Asia | 6
30 | Nathan Diaz | pay much more get less in family package | Asia | 7
Output
Results
Query History