



Sri Lanka Institute of Information Technology

## **Starbucks Offers Data Warehouse Solution**

### **Assignment-2 Document**

IT3021 - Data Warehousing and Business Intelligence  
Assignment 2

Submitted by: IT19069432 – Dissanayake D.M.I.M.

Date of submission: 25/06/2021

## Table of Contents

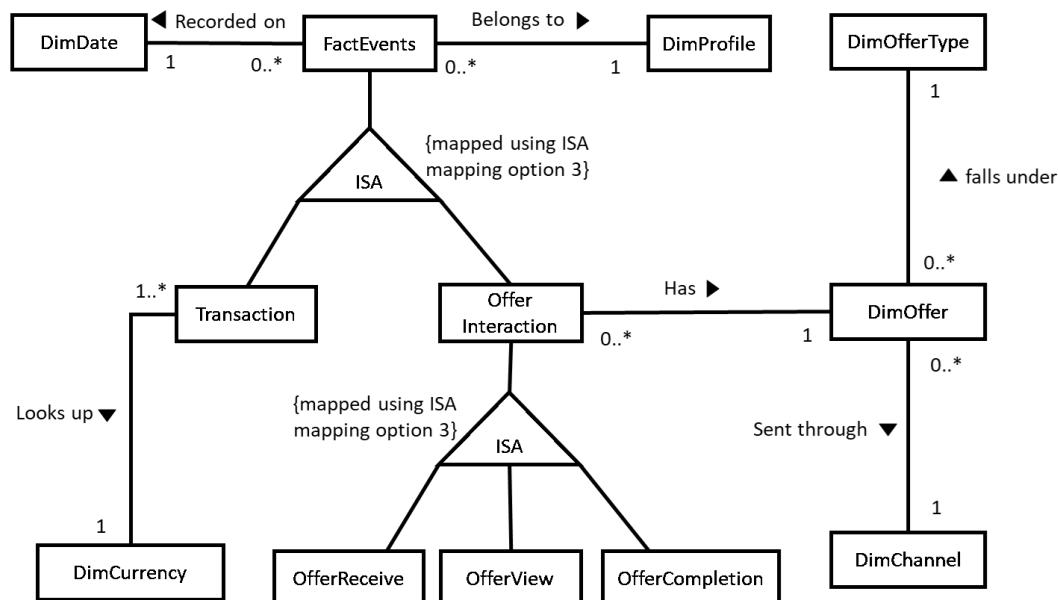
1.	Data Source for Assignment-2.....	4
1.1.	EER Design.....	4
1.2.	Dimensional Model Schema .....	4
1.3.	Dimension and Fact tables in Detail .....	5
1.3.1.	Dimension Tables .....	5
1.3.2.	Fact Tables .....	6
1.3.3.	Measures .....	6
1.3.4.	Identified Hierarchies.....	7
1.3.5.	Derived Columns .....	7
2.	SSAS Cube Implementation .....	9
2.1.	Creating the Project and Data Source .....	9
2.2.	Creating the Data Source View .....	10
2.2.1.	Named Queries.....	11
2.3.	Creating a SSAS Cube .....	14
2.3.1.	Creating the Cube .....	14
2.3.2.	Adding Unlisted Attributes .....	16
2.3.3.	Adding Hierarchies .....	17
2.4.	Deploying the Cube.....	19
3.	Demonstration of OLAP operations .....	20
3.1.	Connecting Excel to the OLAP Cube .....	20
3.2.	Roll-up Operation.....	21
3.3.	Drill-Down Operation .....	22
3.4.	Slice Operation.....	22

3.5.	Dice Operation .....	24
3.6.	Pivot Operation .....	25
3.7.	Excel Dashboard .....	25
4.	SSRS Reports.....	28
4.1.	Configuring the Report Server .....	28
4.2.	Report 1 – SSRS Report with a Matrix .....	29
4.3.	Report 2 – Report with more than one parameter .....	32
4.4.	Report 3 – SSRS Drill-down Report.....	36
4.5.	Report 4 – SSRS Drill-Through Report.....	37
5.	References.....	41

## 1. Data Source for Assignment-2

### 1.1. EER Design

The data warehouse database solution created during the first assignment was selected as the data source for the second assignment without changing the structure of it at all. The data warehouse solution was implemented following a well designed snowflake schema with one fact table and 6 dimension tables in total. Given below is an EER diagram designed to showcase the structure of the data warehouse database solution through explaining the relationships among the fact and dimension tables.



\* Both ISA inheritance relationships were mapped into single table “FactEvents” using ISA mapping option 3.

Figure 1.1. 1. EER Diagram of the Data Warehouse Database

### 1.2. Dimensional Model Schema

The Data warehouse design of the Starbucks\_Offers\_DW is given below, which was designed according to the snowflake schema. Notice that event type Inheritance (Transaction, Offer Interaction) and Offer Interaction Inheritance (OfferReceived, OfferView, OfferCompletion) both ISA relationships mentioned in the above EER are mapped into the same FactEvents table as a single table. Relationships between Dimension tables and Fact tables have been established using surrogate keys instead of foreign keys according to the data warehouse designing rules. DimOffer table has played a major role on making the given schema a snowflake as the table has been normalized by creating the DimChannel and DimOfferType tables separately.

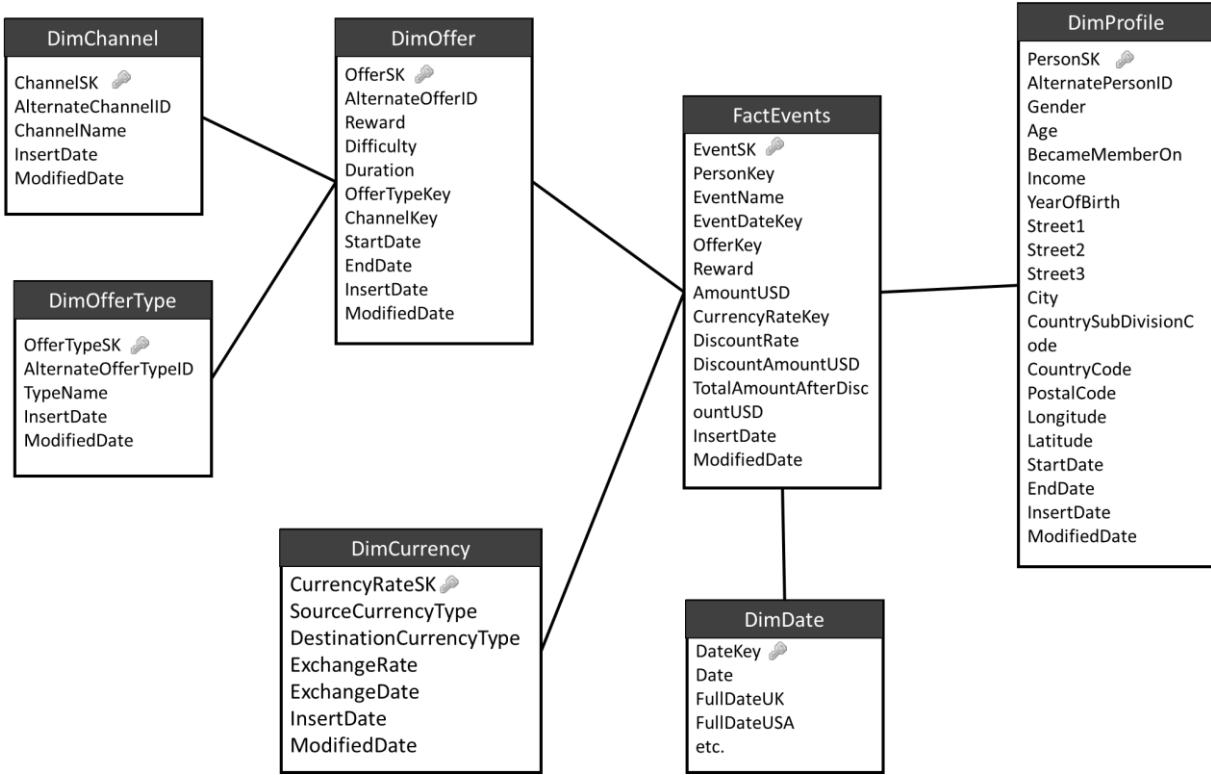


Figure 1.2. 1. Dimensional Model Schema of the Data Warehouse which is a snowflake schema

### 1.3. Dimension and Fact tables in Detail

#### 1.3.1. Dimension Tables

DimProfile dimension table contains Starbucks customer profile information along with their addresses. PersonSK is used as the surrogate key which uniquely identifies a customer, and the Names of the customers were not disclosed in the source tables, thus not included in the dimension table as well. DimProfile was treated as a slowly changing dimension assuming that there is a business requirement to track the locations of the customers over time.

DimOffer dimension table contains all the offer details offered by Starbucks and related details. DimOffer has also been treated as a slowly changing dimension under the assumption that tracking offer feature variation over time is a business need at Starbucks.

DimOffer table has presented in a normalized manner by connecting it to two additional dimensions DimOfferType and DimChannel that could have been presented in a denormalized manner in a single table as well.

DimDate is a static dimension table which is being used for date related calculations and the table is statically populated using a well-written SQL script at the very beginning.

DimCurrency is a special case of dimension tables where it can be considered both static and slowly changing. Since DimCurrency dimension is being populated using an API call, it was decided not to do the currency conversion related calculations at the time of ETL process from staging layer to data warehouse. Thus, in case a currency exchange rate is not populated for a given date, still there should be a default value present as having a surrogate key from fact table to the dimension is a must as the necessary currency conversion related calculations are to be done later at the data retrieving stage or OLAP cube implementation time. Thus, at first, DimCurrency table was treated as a static table and populated with default values. However, while performing the ETL process, these default values get overwritten based on the type of the currency and the date. Since exchange rate values can be accessed based on dates, the DimCurrency dimension can also be treated as a slowly changing dimension where it is possible to track how the exchange rates did vary over time.

### **1.3.2. Fact Tables**

FactEvents is the only transactional fact table in the data warehouse solution and, transaction made by customers and customer interactions with offers are being recorded with the respective dates and other information. There are several measures such as total amount of transactions made in USD and LKR, discounted amounts for a given transaction, and Rewards received through offers customers have interacted with.

### **1.3.3. Measures**

#### **Fully Additive Measures**

- Reward [reward received by a customer for offer completed events, given in USD]
- AmountUSD [original amount had to be paid for a transaction event, given in USD]
- TotalAmountAfterDiscountUSD [amount paid after deduction the discount, in USD]

#### **Semi-Additive Measures**

- There are no semi-additive measures in the FactEvents fact table.

#### **Non-Additive Measures**

- Discount Rate [Discount rate granted for transactions done by the customers]

#### **1.3.4. Identified Hierarchies**

The following Hierarchies were identified between columns of the given tables.

- Year > Month > Day [DimDate]
- SourceCurrencyType > DestinationCurrencyType [DimCurrency]
- OfferType > Offer [DimOffer and DimOfferType]
- Channel > Offer [DimOffer and DimChannel]
- Age > YearOfBirth > Person\_ID [DimProfile]
- Countrycode > CountrySubDivisionCode > City > PostalCode > Street1 > Street2 > Street3 [DimProfile]
- City > PostalCode > Street1 > Street2 > Street3 > geolocation (latitude and longitude) [DimProfile]

#### **1.3.5. Derived Columns**

- **YearOfBirth** – derived from using age and membership acquired date [DimProfile]
- **Surrogate Keys** – given to each data warehouse dimension table as a unique PK [int]
- **InsertDate and ModifiedDate** – given to all dimensional model dim and fact tables to record timestamps of insertions and modifications.
- **StartDate, EndDate** – given to each SCD table except DimCurrency, to keep track of historical value changes
- **DiscountAmountUSD** – (AmountUSD \* discount rate) [FactEvents Table]
- **TotalAmountAfterDiscountUSD** – (AmountUSD – DiscountAmountUSD) [FactEvents Table]

The complete list of fact and dimension tables and their column details with descriptions can be found in the following figure.

Dimension Name	- Truncate Before Update?	Dimension Attributes	- Derived Attribute?	- Data Type	- Null/ Not Null	Key Column?	- Data Length	- Derived Logic	- Description
DimChannel	No	ChannelSK	Yes	int	NOT NULL	Yes. SK	Arbitrary	Auto Increment	Dimension Table Unique Surrogate Key
		AlternateChannelID	No	nvarchar(6)	NOT NULL	No	6		Business Key in the Source Table
		ChannelName	No	nvarchar(50)	NOT NULL	No	50		Name of the Channel Group offers send through
		InsertDate	Yes	datetime	NOT NULL	No	N/A	System Date	Timestamp of the Insertion Date
		ModifiedDate	Yes	datetime	NOT NULL	No	N/A	System Date	Timestamp of the Modified Date
DimOfferType	No	OfferTypeSK	Yes	int	NOT NULL	Yes. SK	Arbitrary	Auto Increment	Dimension Table Unique Surrogate Key
		AlternateOfferTypeID	No	nvarchar(6)	NOT NULL	No	6		Business Key in the Source Table
		TypeName	No	nvarchar(15)	NOT NULL	No	15		Offer Type of offers offered by Starbucks
		InsertDate	Yes	datetime	NOT NULL	No	N/A	System Date	Timestamp of the Insertion Date
		ModifiedDate	Yes	datetime	NOT NULL	No	N/A	System Date	Timestamp of the Modified Date
DimOffer (Considered as SDC)	No	OfferSK	Yes	int	NOT NULL	Yes. SK	Arbitrary	Auto Increment	Dimension Table Unique Surrogate Key
		AlternateOfferID	No	nvarchar(50)	NOT NULL	No	50		Business Key in the Source Table
		Reward	No	tinyint	NULL	No	4		Reward offered on Offer completion as a USD value
		Difficulty	No	tinyint	NULL	No	4		Total Amount have to pay if offer is absent (in USD)
		Duration	No	tinyint	NULL	No	4		Duration till the offer is available after generated
		OfferTypeKey	No	int	NOT NULL	Yes. FK	Arbitrary		Surrogate Key of the Offer Type
		ChannelKey	No	int	NOT NULL	Yes. FK	Arbitrary		Surrogate Key of the Channel
		StartDate	Yes	datetime	NOT NULL	No	N/A	This Record Insert Time	This Record Insert Time
		EndDate	Yes	datetime	NULL	No	N/A	Next Record Insert Time	Next Record Insert Time
		InsertDate	Yes	datetime	NOT NULL	No	N/A	System Date	Timestamp of the Insertion Date
		ModifiedDate	Yes	datetime	NOT NULL	No	N/A	System Date	Timestamp of the Modified Date
DimProfile (Considered as SDC)	No	PersonSK	Yes	int	NOT NULL	Yes. SK	Arbitrary	Auto Increment	Dimension Table Unique Surrogate Key
		AlternatePersonID	No	nvarchar(50)	NOT NULL	No	50		Business Key in the Source Table
		Gender	No	nvarchar(2)	NULL	No	2		Gender of the Customer
		Age	No	tinyint	NULL = 118	No	4		Age of the Customer
		BecameMemberOn	No	datetime	NULL	No	N/A		BecameMemberOn of the Customer
		Income	No	money	NULL	No	N/A		Income of the Customer
		YearOfBirth	Yes	int	NULL	No	4	(YEAR([BecameMemberOn]) - Age)	YearOfBirth of the Customer
		Street1	No	nvarchar(60)	NOT NULL	No	60		Street1 of the Customer's Address
		Street2	No	nvarchar(60)	NULL	No	60		Street2 of the Customer's Address
		Street3	No	nvarchar(60)	NULL	No	60		Street3 of the Customer's Address
		City	No	nvarchar(50)	NOT NULL	No	50		City of the Customer's Address
		CountrySubDivisionCode	No	nvarchar(4)	NOT NULL	No	4		CountrySubDivisionCode of the Customer
		CountryCode	No	nvarchar(4)	NOT NULL	No	4		CountryCode of the Customer's Address
		PostalCode	No	nvarchar(50)	NOT NULL	No	50		PostalCode of the Customer's Address
		Longitude	No	nvarchar(50)	NULL	No	50		Longitude of the Customer's Location
		Latitude	No	nvarchar(50)	NULL	No	50		Latitude of the Customer's Location
		StartDate	Yes	datetime	NOT NULL	No	N/A	This Record Insert Time	This Record Insert Time
		EndDate	Yes	datetime	NULL	No	N/A	Next Record Insert Time	Next Record Insert Time
		InsertDate	Yes	datetime	NOT NULL	No	N/A	System Date	Timestamp of the Insertion Date
		ModifiedDate	Yes	datetime	NOT NULL	No	N/A	System Date	Timestamp of the Modified Date
FactEvents	No	EventSK	Yes	int	NOT NULL	Yes. SK	Arbitrary	Auto Increment	Dimension Table Unique Surrogate Key
		PersonKey	No	int	NOT NULL	Yes. FK	Arbitrary		Business Key in the Source Table
		EventName	No	nvarchar(50)	NOT NULL	No	50		EventName of the Customer's Address
		EventDateKey	No	int	NULL	Yes. FK	Arbitrary		EventDateKey of the Customer
		OfferKey	No	int	NULL	Yes. FK	Arbitrary		OfferKey of the Customer
		Reward	No	tinyint	NULL	No	4		Reward of the Customer
		AmountUSD	No	money	NULL	No	N/A		AmountUSD of the Customer
		CurrencyRateKey	Yes	int	NULL	Yes. FK	Arbitrary		CurrencyRateSK of the Customer
		DiscountRate	No	decimal(3,2)	NULL	No	3,2		DiscountRate of the Customer's Dication
		DiscountAmountUSD	Yes	money	NULL	No	N/A	((AmountUSD)*(DiscountRate)/100))	DiscountAmountUSD of the Customer
DimDate	No	DateKey	Yes	int	NOT NULL	Yes. SK	N/A	Static Dimensional Table	Dimension Table Unique Surrogate Key
		More Attributes...							...
DimCurrency (Considered as SDC but NO sd/ed)	No	CurrencyRateSK	Yes	int	NOT NULL	Yes. SK	Arbitrary	Auto Increment	Dimension Table Unique Surrogate Key
		SourceCurrencyType	No	nvarchar(4)	NOT NULL	No	4		Partial Business Key in the Source Table
		DestinationCurrencyType	No	nvarchar(4)	NOT NULL	No	4		Partial Business Key in the Source Table
		ExchangeRate	No	decimal(18,6)	NOT NULL	No	18,6		Actual Exchange Rate
		ExchangeDate	No	date	NOT NULL	No	N/A		Partial Business Key in the Source Table
		InsertDate	Yes	datetime	NOT NULL	No	N/A	System Date	Timestamp of the Insertion Date
		ModifiedDate	Yes	datetime	NOT NULL	No	N/A	System Date	Timestamp of the Modified Date

Figure 1.3. 1. The complete list of Dimension and Fact Tables in the Data Warehouse Database Design. 200% Zoom is recommended for making the text more readable.

## 2. SSAS Cube Implementation

### 2.1. Creating the Project and Data Source

To implement the SSAS Cube, first, an “Analysis Services Multidimensional and Data Mining Project” was created under the name of “Starbucks\_SSAS”.

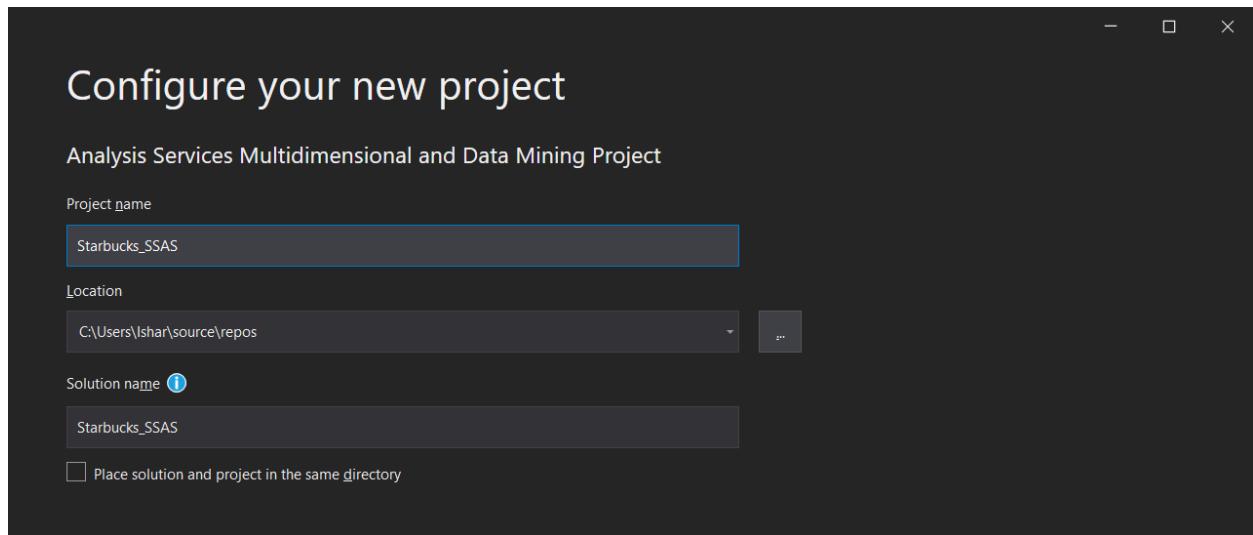


Figure 2.1. 1. Project creation of Starbucks\_SSAS Analysis Services Multidimensional and Data Mining Project

After project was created successfully, new Data Source was added using Data Source Wizard through a connection that was established to the existing Data Warehouse database in the localhost of the machine.

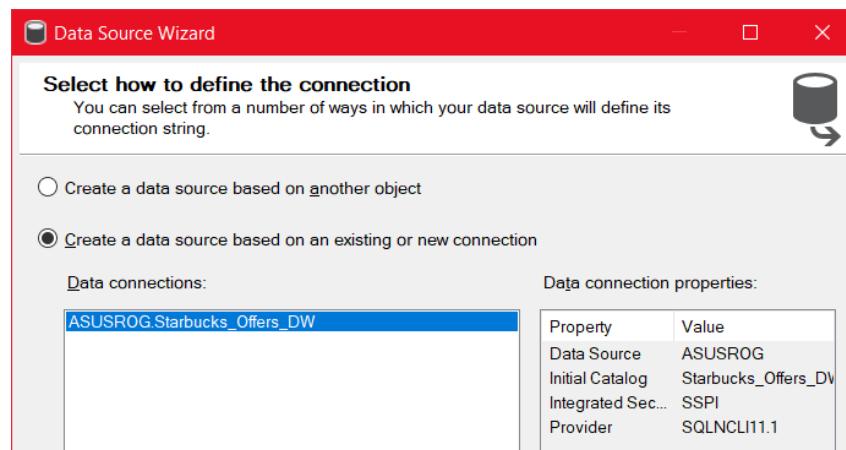


Figure 2.1. 2. Data Source Wizard - New Connection to the Data Warehouse Database

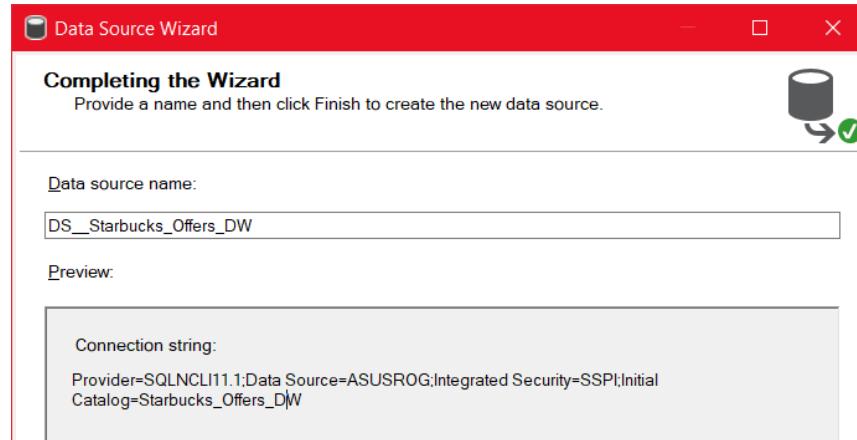


Figure 2.1. 3. Data Source Wizard - DS\_Starbucks\_Offers\_DW Data Source Creation

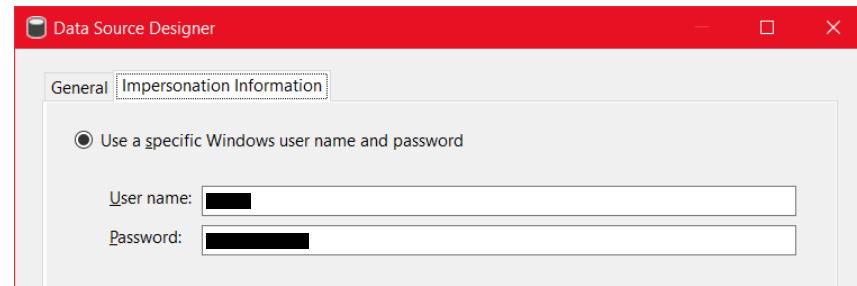


Figure 2.1. 4. Data Source Impersonation Credentials was set to an existing windows account which is used as the Log on by both SQL Server and SSAS

## 2.2. Creating the Data Source View

Based on the Data Source created above, a data source view was created using Data Source View Wizard. All Dimensions and Fact tables were added to Included Objects of the Data Source View as all of them are necessary for the following steps.

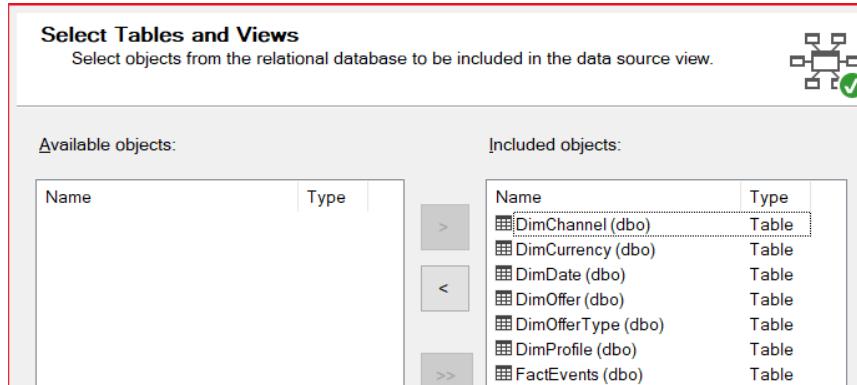


Figure 2.2. 1. List of Selected Tables and Views to be included in the Data Source View

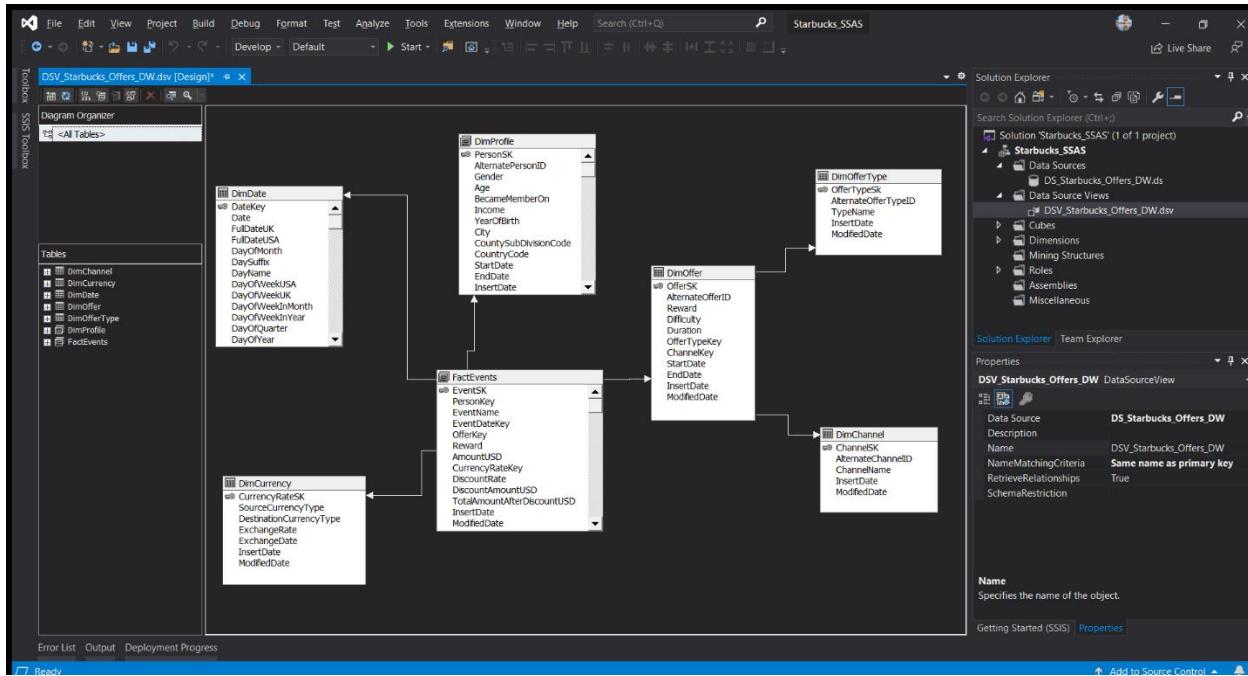


Figure 2.2. 2. Data Source View Designer Tab - Connected Dimension and Fact Tables

As shown in the Figure 2.2.2., Dimension tables and Fact table were linked using surrogate keys based on the Data Warehouse Dimensional Model Schema. List of Table links established are as follows.

- **PersonKey** of **FactEvents** fact table with **PersonSK** of **DimProfile** dimension table
- **OfferKey** of **FactEvents** fact table with **OfferSK** of **DimOffer** dimension table
- **CurrencyRateKey** of **FactEvents** fact table with **CurrencyRateSK** of **DimCurrency** dimension table
- **EventDateKey** of **FactEvents** fact table with **DateKey** of **DimDate** dimension table
- **ChannelKey** of **DimOffer** dimension table with **ChannelSK** of **DimChannel** dimension table
- **OfferTypeKey** of **DimOffer** dimension table with **OfferTypeSK** of **DimOfferType** dimension table

### 2.2.1. Named Queries

Named Queries help to alter the columns of an existing Dimension or Fact table, retrieve data from multiple tables to create a new dimension or simply to add new columns to an existing table. Named Query can be given a specific name, description and the data retrieval must be specified using an SQL Query. To replace or alter an existing table with a Named Query, table should be right clicked and select the ‘Replace Table’ and select ‘With New Named Query’ from the sub menu.

### 2.2.1.1.DimProfile Named Query

When getting DimProfile table data in the data source view designer, A named Query was used instead of connecting the table following the usual procedure. The reason for using a named query was to get rid of NULL values in “Street1”, “Street2”, “Street3”, “Latitude”, “Longitude”, and “PostalCode” columns. Without creating a named query, the deployment of the cube is failing with the error “SSAS Error: Duplicate attribute key found when processing” as SSAS does not support having null values as it writes distinct queries on the above-mentioned columns while the OLAP Cube deployment. Named Query used in DimProfile simply generates blank values by converting Null values to blank values for necessary columns only by using the following SQL Query.[1]

```
SELECT PersonSK, AlternatePersonID, Gender, Age, BecameMemberOn, Income,
YearOfBirth, ISNULL(Street1, '') AS Street1, ISNULL(Street2, '') AS Street2,
ISNULL(Street3, '') AS Street3, City, CountySubDivisionCode, CountryCode,
ISNULL(PostalCode, '') AS PostalCode, ISNULL(Longitude, '') AS Longitude,
ISNULL(Latitude, '') AS Latitude, StartDate, EndDate, InsertDate, ModifiedDate
FROM      DimProfile
```

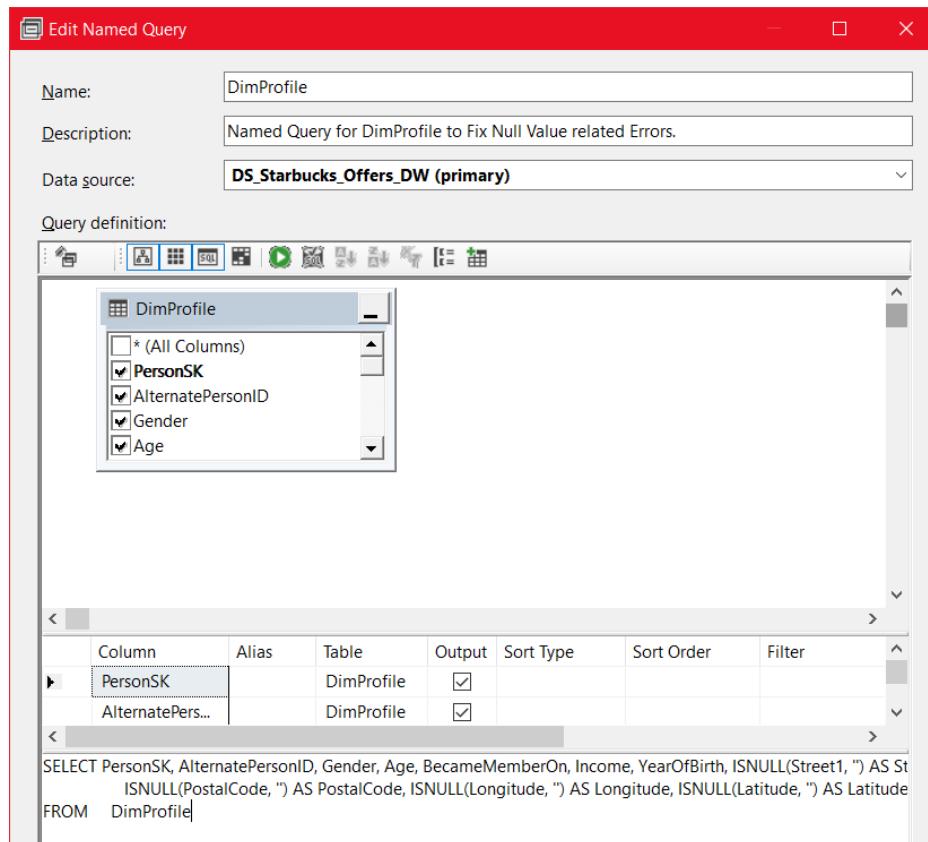


Figure 2.2.1.2. 1. DimProfile Named Query which replaces possible repeating NULL values with a blank value using ISNULL() SQL method for necessary columns.

### 2.2.1.2. FactEvents Named Query

FactEvents Named Query was created to add the additional measure columns ‘AmountLKR’ and ‘TotalAmountAfterDiscountLKR’ which converts existing measures in USD to LKR using a table join with the DimCurrency. This is done so that every time the currency exchange rate default values get overridden by the actual currency exchange rates, the measures can be calculated right in the cube without having to update the fact table. Named Query was written as given below.

```
SELECT a.EventSK, a.PersonKey, a.EventName, a.EventDateKey, a.OfferKey,
a.Reward, a.AmountUSD, a.CurrencyRateKey, a.DiscountRate, a.DiscountAmountUSD,
a.TotalAmountAfterDiscountUSD, a.InsertDate, a.ModifiedDate, a.AmountUSD *
b.ExchangeRate AS 'AmountLKR', a.TotalAmountAfterDiscountUSD * b.ExchangeRate
AS 'TotalAmountAfterDiscountLKR'

FROM FactEvents AS a INNER JOIN DimCurrency AS b ON a.CurrencyRateKey =
b.CurrencyRateSK
```

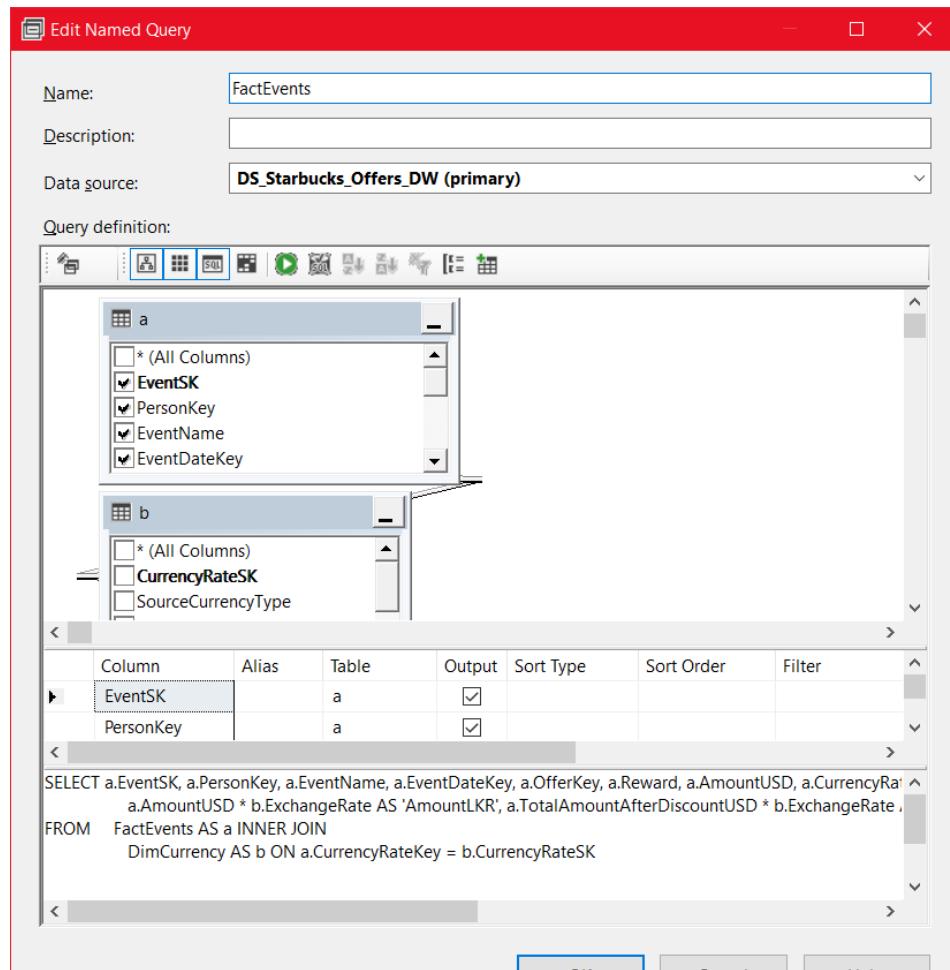


Figure 2.2.1.2. 2. FactEvents Named Query to generate additional measures to be able to use in the cube

## 2.3. Creating a SSAS Cube

### 2.3.1. Creating the Cube

If the data warehouse contains dimension and fact tables related to several different subjects/ areas, several cubes can be created accordingly. However, since Starbucks\_Offers\_DW data warehouse contains dimension and fact tables only related to Starbucks Offers, it was decided to create and design a single OLAP cube.

Cube Wizard was used to create the cube with existing tables which are imported and linked in the previous data source view creation step. FactEvents table was selected as the Measure group table.



Figure 2.3.1. 1. Cube Wizard - FactEvents table was selected as the Measure group tables.

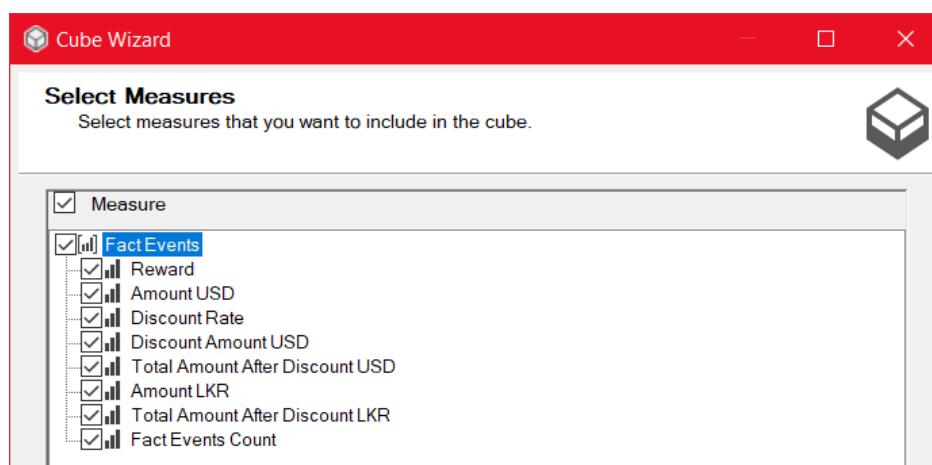


Figure 2.3.1. 2. Cube Wizard - Selected Measures

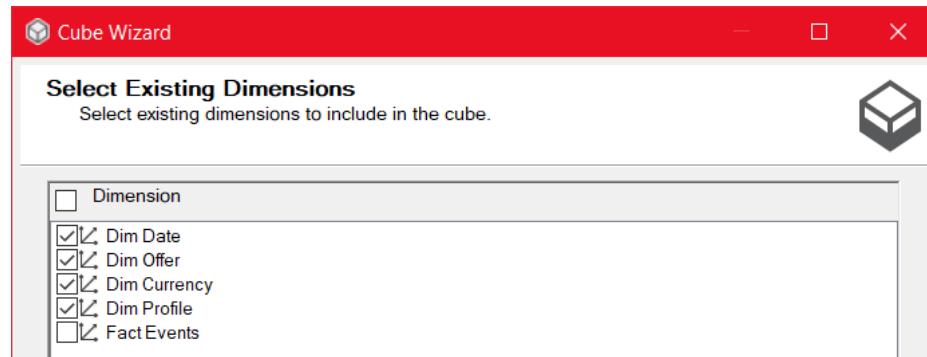


Figure 2.3.1. 3. Cube Wizard - Selected Dimensions

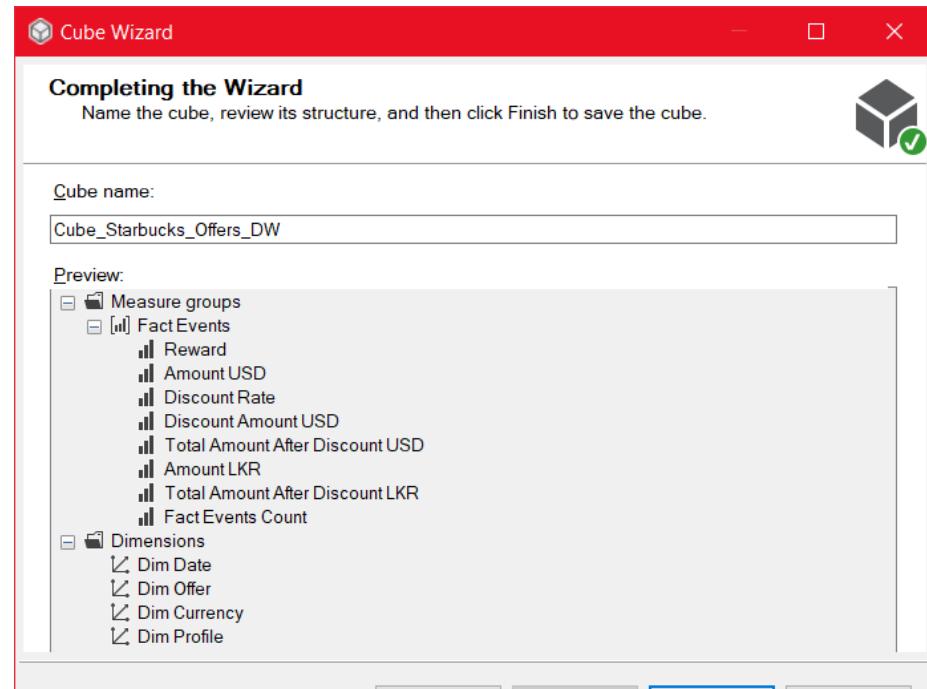


Figure 2.3.1. 4. Cube Wizard - Selected Measures and Dimensions Preview.

As it is indicated in the figure 2.3.4., Measure groups, Measures and dimensions were selected appropriately based on the existing data warehouse tables and their columns, and finally, the cube was created.

After the cube was created initially, missing attributes in the dimension tables were added using Edit dimension links in the Dimensions tab of the ‘cube structure’ main tab in the cube designer.

### 2.3.2. Adding Unlisted Attributes

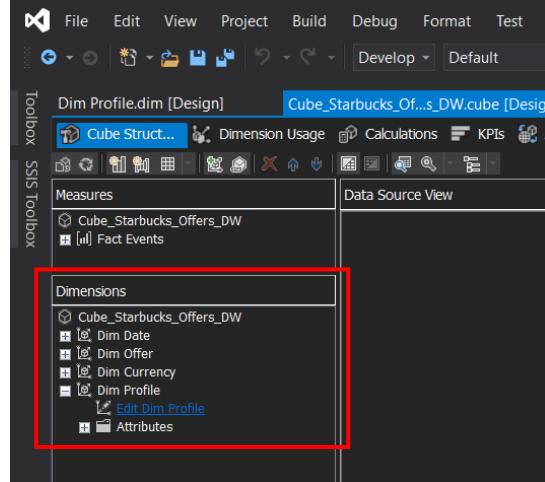


Figure 2.3.2. 1. Adding necessary attributes for the cube or creating hierarchies is done by clicking the Edit Dimension link given in the Dimensions Tab of the cube structure tab for each dimension.

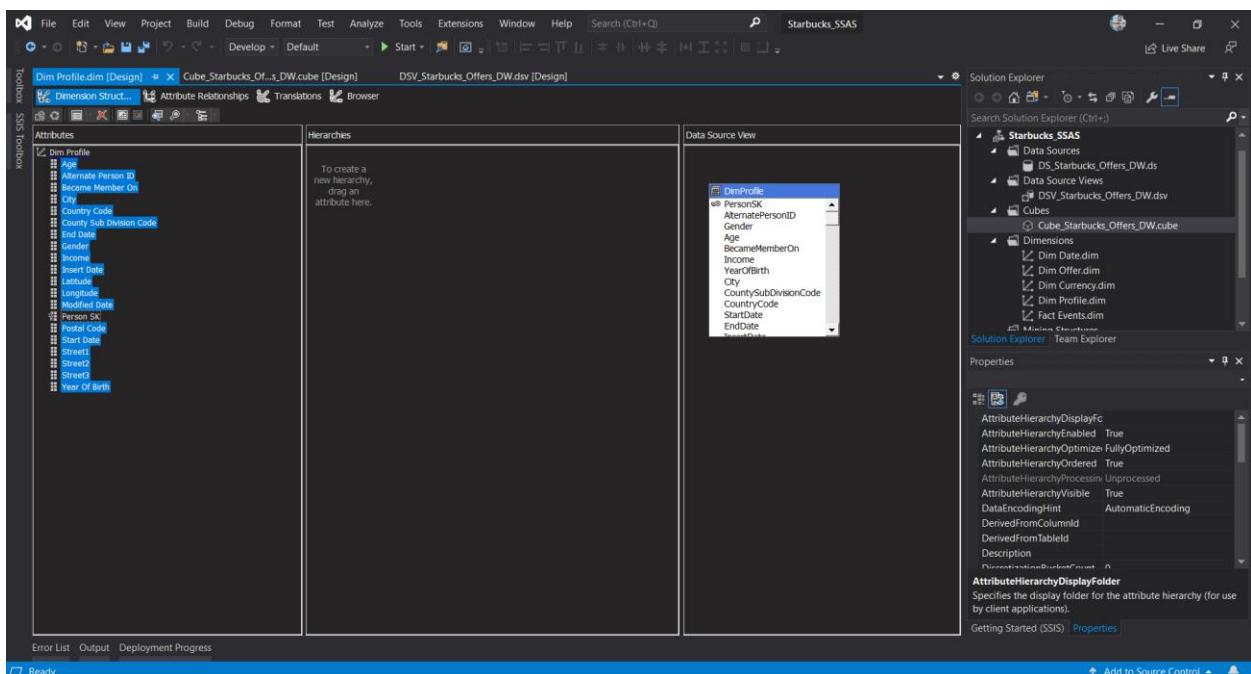


Figure 2.3.2. 2. DimProfile dimension after adding the necessary attributes for the cube.

As shown in the figure 2.2.1.2.4., unlisted attributes can be dragged and dropped from the data source view section to Attributes section. This was done for all the dimension tables and all the attributes were treated as necessary attributes for the cube.

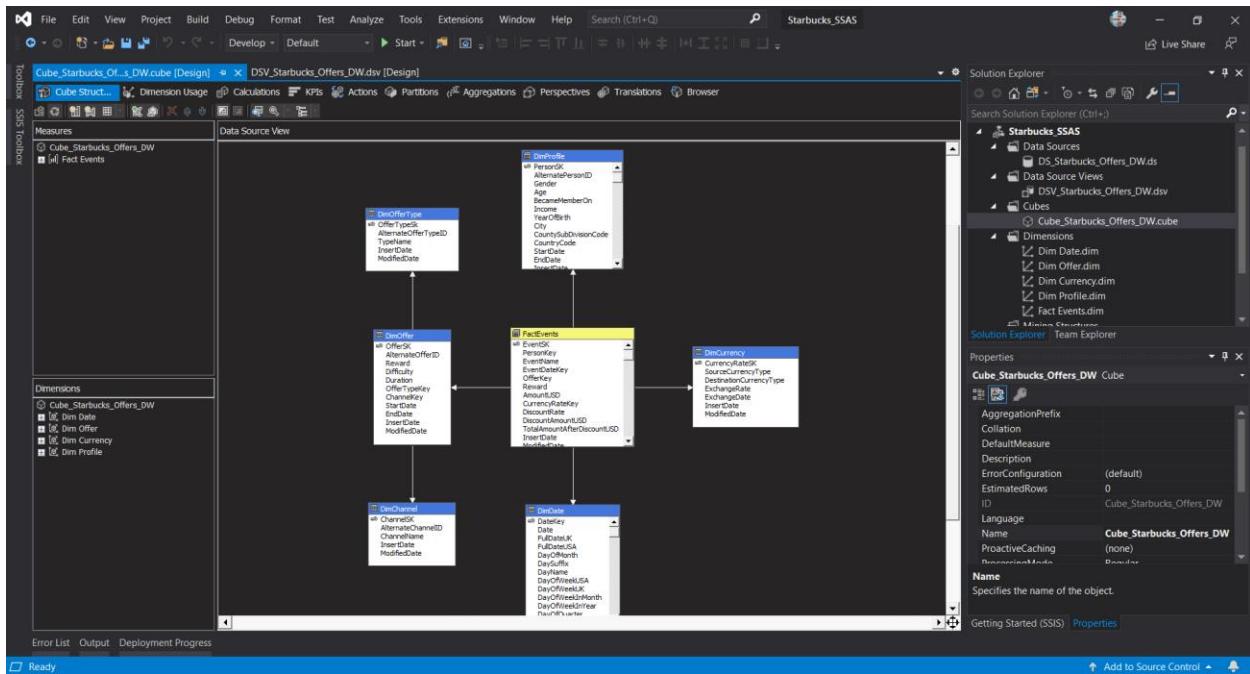


Figure 2.3.2. 3. Cube Designer - Data Source View of the Cube

### 2.3.3. Adding Hierarchies

Hierarchies can be added for dimensions in dimension designer window which can be accessed by clicking on the edit dimension link in cube structure tab of the cube designer window (refer figure 2.3.2.1). User-defined hierarchies can be used to organize the attributes of a dimension into hierarchical structures to provide navigation paths in a cube. They can be used to reduce complexity and guide users into a specific drill-down behavior. Hierarchies defined here can be easily used later when browsing the cube to obtain measures based on the pre-defined hierarchies without having to build them from scratch.

The following hierarchies were added to the cube designed earlier.

DimProfile Dimension

- City Hierarchy
- Location Hierarchy

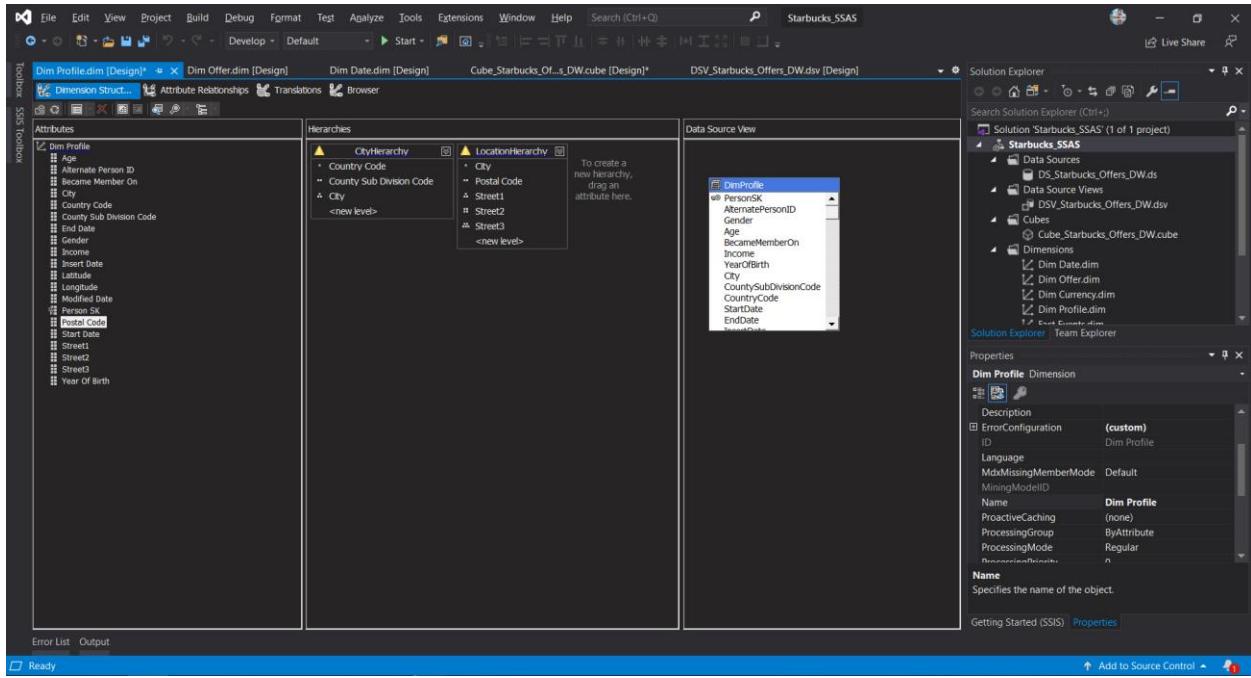


Figure 2.3.3. 1. Dim Profile Dimension Hierarchies

## DimOffer Dimension

- Offer Type Hierarchy
- Offer Channel Hierarchy

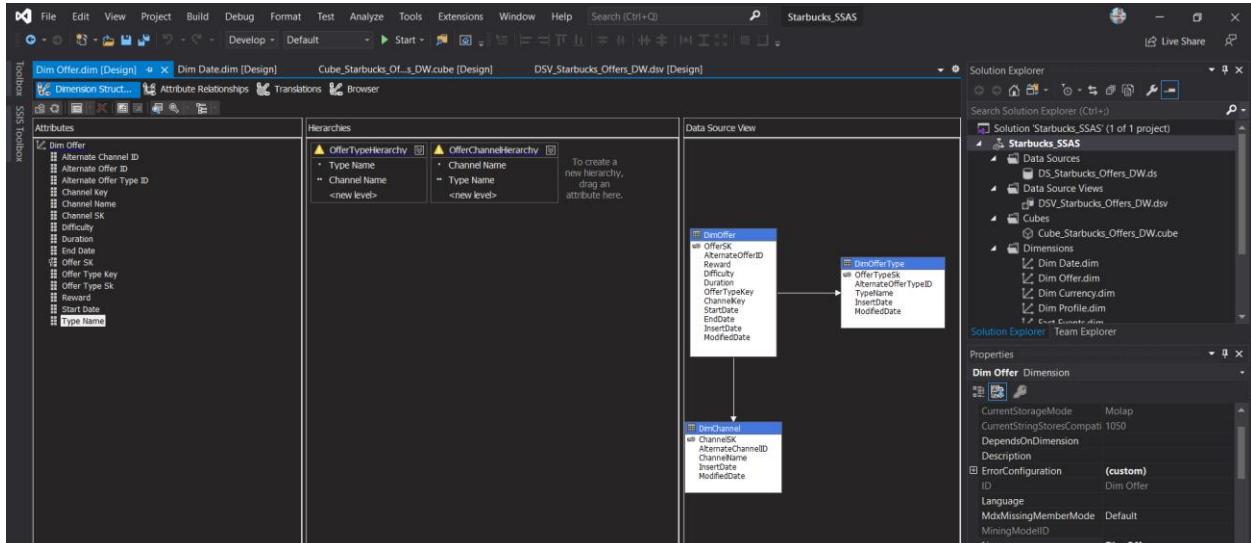


Figure 2.3.3. 2. DimOffer Dimension Hierarchies

## DimDate Dimension

- Date Hierarchy
- Month Hierarchy
- Period Hierarchy

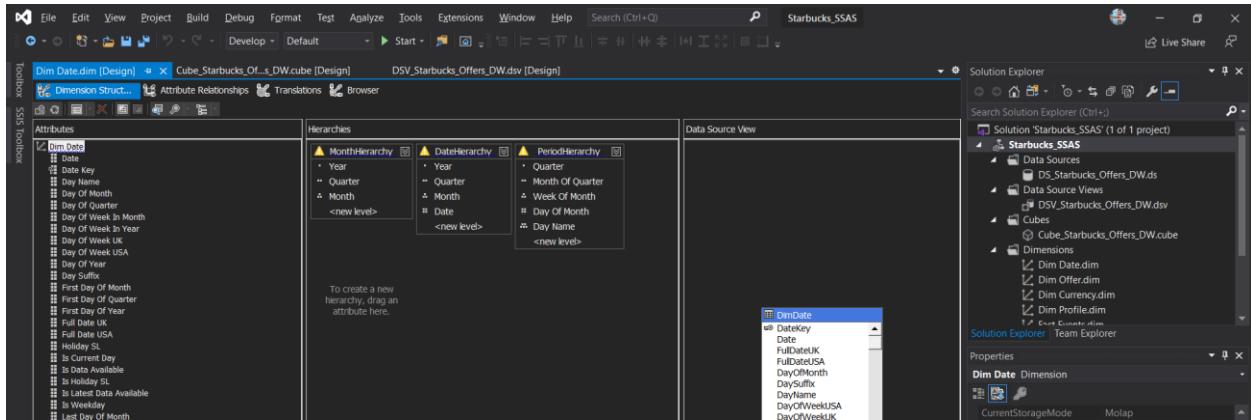


Figure 2.3.3. 3. DimDate Dimension Hierarchies.

## 2.4. Deploying the Cube

The project Starbucks\_SSAS was rebuilt and deployed by right clicking the project name and selecting deploy. Before the deployment it was checked that Dimension Usage were correctly configured.

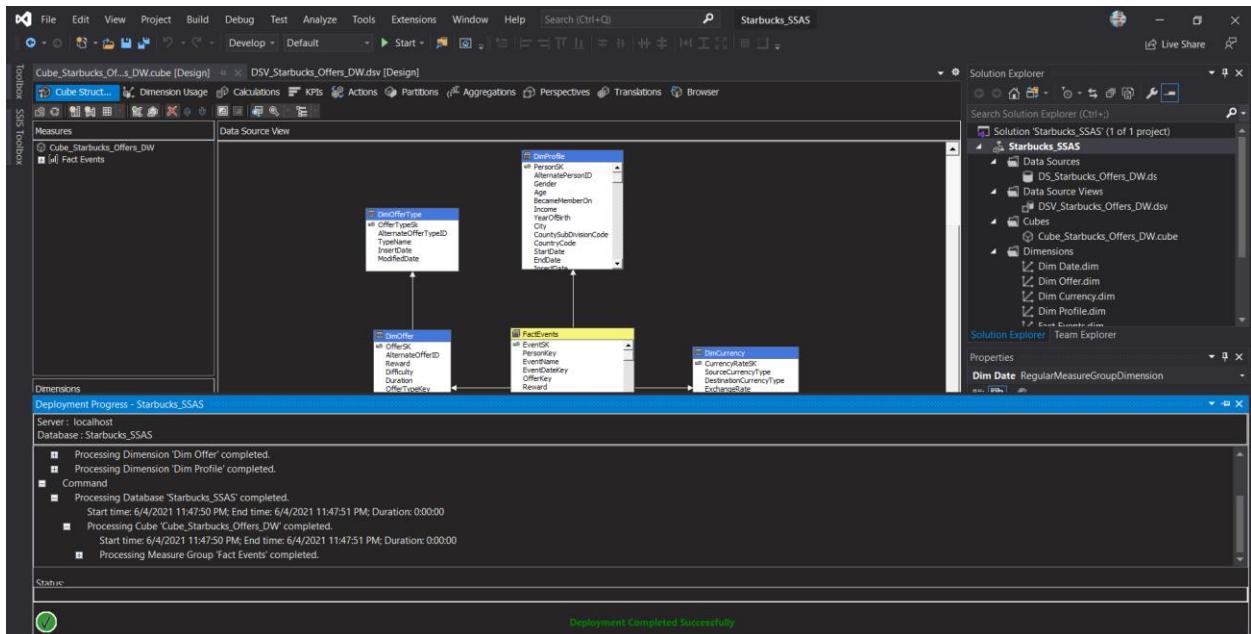


Figure 2.4. 1. Cube after the deployment was a success.

### 3. Demonstration of OLAP operations

#### 3.1. Connecting Excel to the OLAP Cube

To do the demonstrations using Microsoft Excel 365, Excel Data tab was used to directly connect to the cube to get all the tables rather than using the power pivot mode where writing MDX Expression is needed. When connecting to the deployed OLAP cube for the first time, it was done as follows.

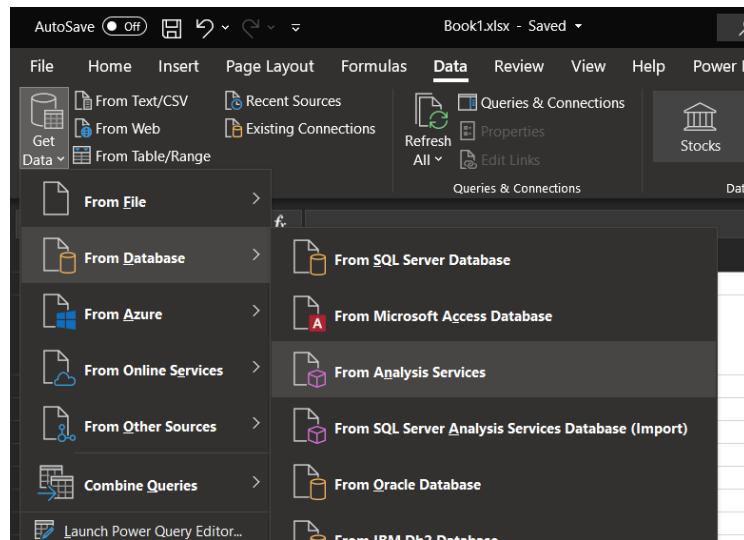


Figure 3.1. 1. Connecting to the cube using "Data" tab's Get Data option using Analysis Services where the cube is deployed.

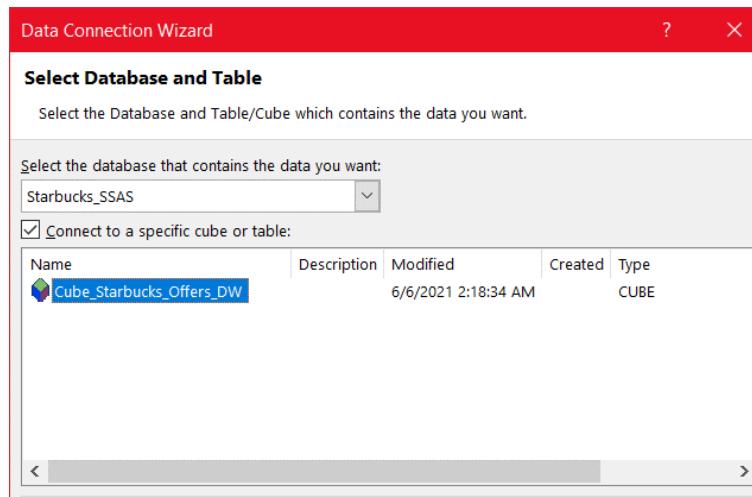
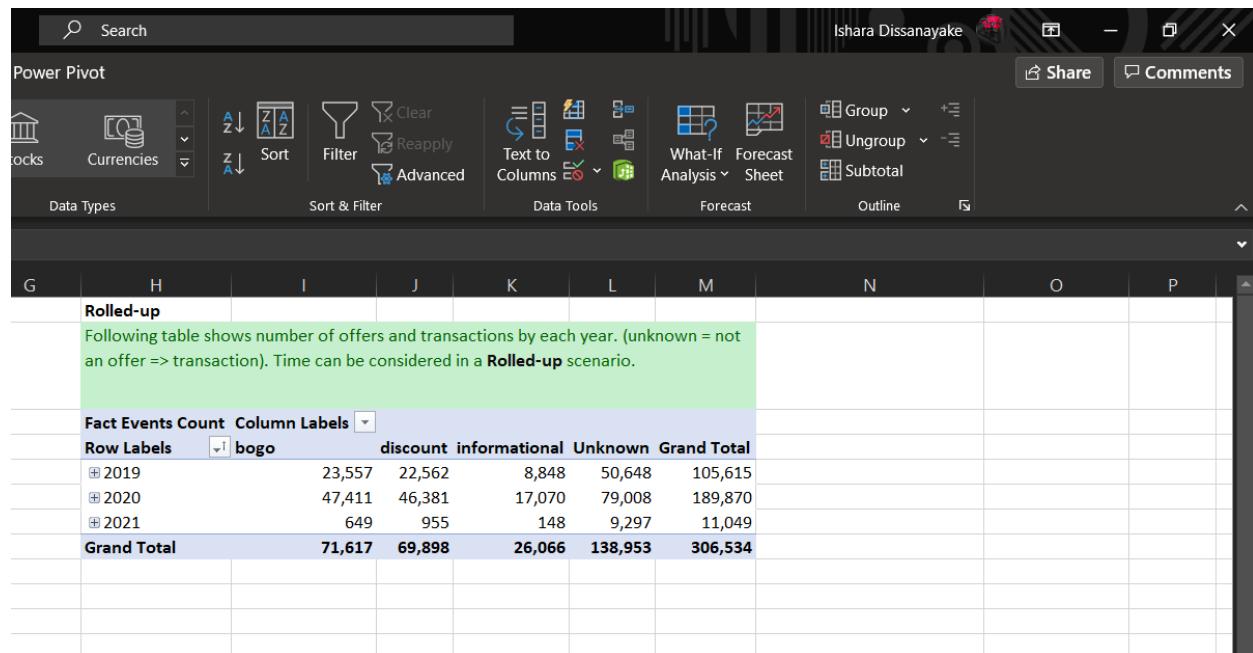


Figure 3.1. 2. Deployed cube was selected from Data Connection Wizard after providing connection details to SSAS.

After successfully connecting to the cube, Roll-up, Drill-down, Slice, Dice, and Pivot OLAP operations were demonstrated using different excel pivot tables using several worksheets of the same excel workbook, and finally, a excel dashboard was set up to enable visualizations to easily comprehend the derived analytics using the pivot charts and additional pivot tables along with a user-friendly dashboard interface designed right inside the excel workbook in a different worksheet.

### 3.2. Roll-up Operation

Roll-up operation was demonstrated in the “Roll-up vs Drill-down” worksheet with the help of created hierarchies in the cube. As shown in the following figure, number of offers and transactions done within a specified time(year/ quarter/ month/ day) is presented in the pivot table. The time axis/column is in a roll-up view where the table shows offers and transactions done based on specific years only. This can be further drilled down to days where it is possible to obtain number of offers and transactions within a given day of a week since the hierarchy used here allows it.



The screenshot shows the Microsoft Excel ribbon with the 'Power Pivot' tab selected. The main area displays a pivot table with the following data:

	Fact	Events	Count	Column Labels	
Row Labels	bogo	discount	informational	Unknown	Grand Total
2019	23,557	22,562	8,848	50,648	105,615
2020	47,411	46,381	17,070	79,008	189,870
2021	649	955	148	9,297	11,049
<b>Grand Total</b>	<b>71,617</b>	<b>69,898</b>	<b>26,066</b>	<b>138,953</b>	<b>306,534</b>

A green callout box highlights the first row of the pivot table with the text: "Following table shows number of offers and transactions by each year. (unknown = not an offer => transaction). Time can be considered in a **Rolled-up** scenario."

Figure 3.2. 1. Number of Offers and Transactions in a drilled down state based specific years given

### 3.3. Drill-Down Operation

In the same worksheet where roll-up was demonstrated, drill-down operation was also demonstrated using the same dimension attributes and measure using another pivot table which is shown below. The rows of the pivot table are now drilled down further till months where it allows to read number of transactions and offers based on months of a given quarter of a specific year. As it can be seen here, low level hierarchy attributes were used to hierarchy attributes were used to demonstrate data in a drilled-down manner.

The screenshot shows two pivot tables side-by-side in an Excel spreadsheet. The left pivot table is titled 'Drilled-down' and displays detailed transaction data categorized by month and year. The right pivot table is titled 'Rolled-up' and displays summary data by year. Both tables have columns for Fact, Events, Count, and Column Labels (bogo, discount, informational, Unknown, Grand Total). The data includes figures for 2019, 2020, and 2021, with specific values for each month and category.

	Fact	Events	Count	Column Labels	bogo	discount	informational	Unknown	Grand Total	
1	3	Fact	Events	Count	Column Labels	bogo	discount	informational	Unknown	Grand Total
2	Row Labels	1	2	3	4	5	6	7	8	
3	3	1	2	3	4	5	6	7	8	
4	Row Labels	1	2	3	4	5	6	7	8	
5	2019	1	2	3	4	5	6	7	8	
6	2019	1	2	3	4	5	6	7	8	
7	2019	1	2	3	4	5	6	7	8	
8	2019	1	2	3	4	5	6	7	8	
9	2019	1	2	3	4	5	6	7	8	
10	2019	1	2	3	4	5	6	7	8	
11	2019	1	2	3	4	5	6	7	8	
12	2019	1	2	3	4	5	6	7	8	
13	2019	1	2	3	4	5	6	7	8	
14	2019	1	2	3	4	5	6	7	8	
15	2019	1	2	3	4	5	6	7	8	
16	2019	1	2	3	4	5	6	7	8	
17	2019	1	2	3	4	5	6	7	8	
18	2019	1	2	3	4	5	6	7	8	
19	2019	1	2	3	4	5	6	7	8	
20	2020	1	2	3	4	5	6	7	8	
21	2020	1	2	3	4	5	6	7	8	
22	2020	1	2	3	4	5	6	7	8	
23	2020	1	2	3	4	5	6	7	8	
24	2020	1	2	3	4	5	6	7	8	

Figure 3.3. 1. Offer/Transaction Data Demonstrated in a drilled-down manner. (Left Pivot table). (Right side pivot table is the same data set demonstrated in a rolled-up manner)

### 3.4. Slice Operation

A pivot table which helps to identify channels which were used to send offers and the number of offers sent through each type of channel (channel trend) was used to demonstrate the slice operation. Here the data can be sliced based on channel type using the given slicer to see data related only for the selected channel type/ types.

Slicer was implemented by clicking on the pivot table and then heading to “PivotTable Analyze” tab in the Excel Ribbon and then selecting the “Insert Slicer”. Then from the check list, relevant dimension attributes can be selected to perform the slicing and, in this case, “Channel Name” from the “DimChannel” was selected as the only slicing attribute.

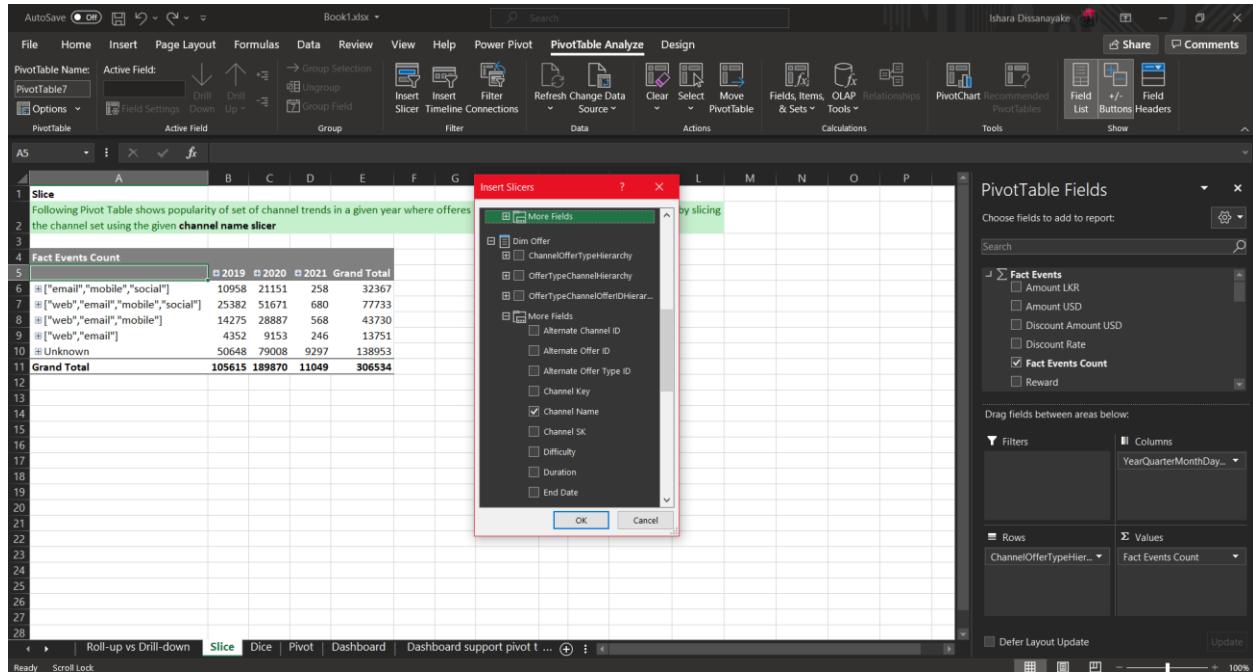


Figure 3.4. 2. Inserting the Channel Name Slicer

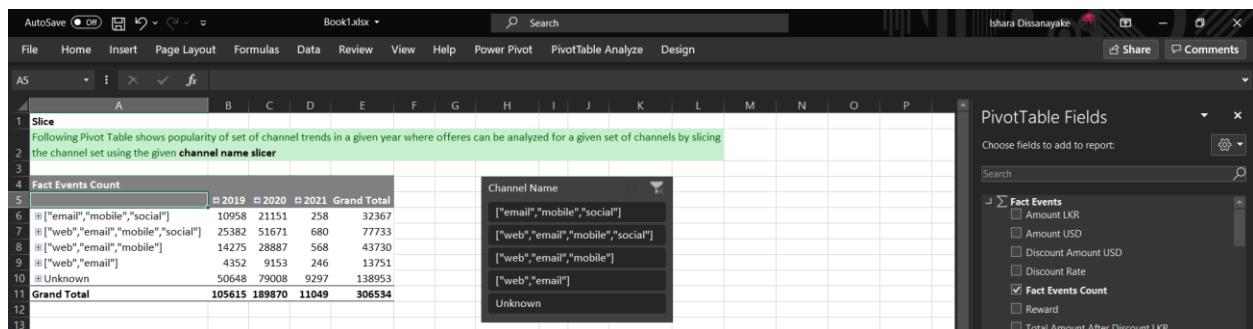


Figure 3.4. 1. Before the slicing was performed

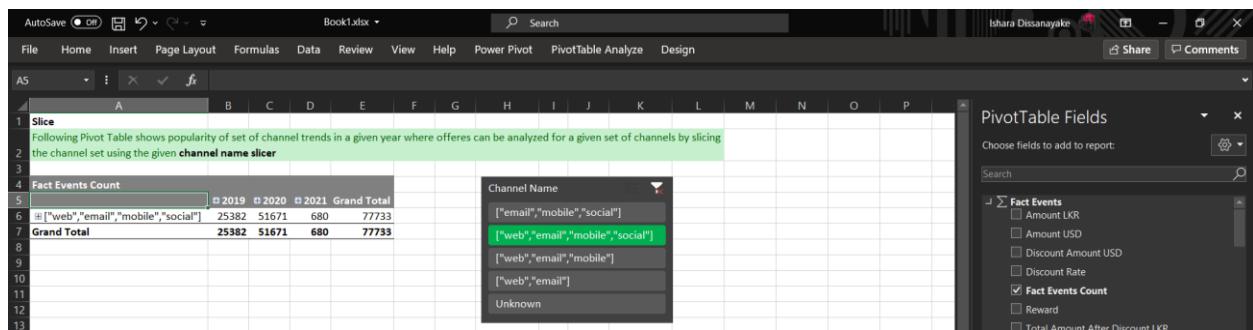


Figure 3.4. 3. After the Slicing was performed

### 3.5. Dice Operation

Dicing can be performed by slicing two dimensions in the pivot table. In this case, dicing was performed on the same pivot table data used while slicing in a new worksheet named “Dice” in the excel workbook. The DimOffer and DimDate dimension attributes were both sliced by Channel Name and Year to demonstrate the Dice OLAP operation as follows.

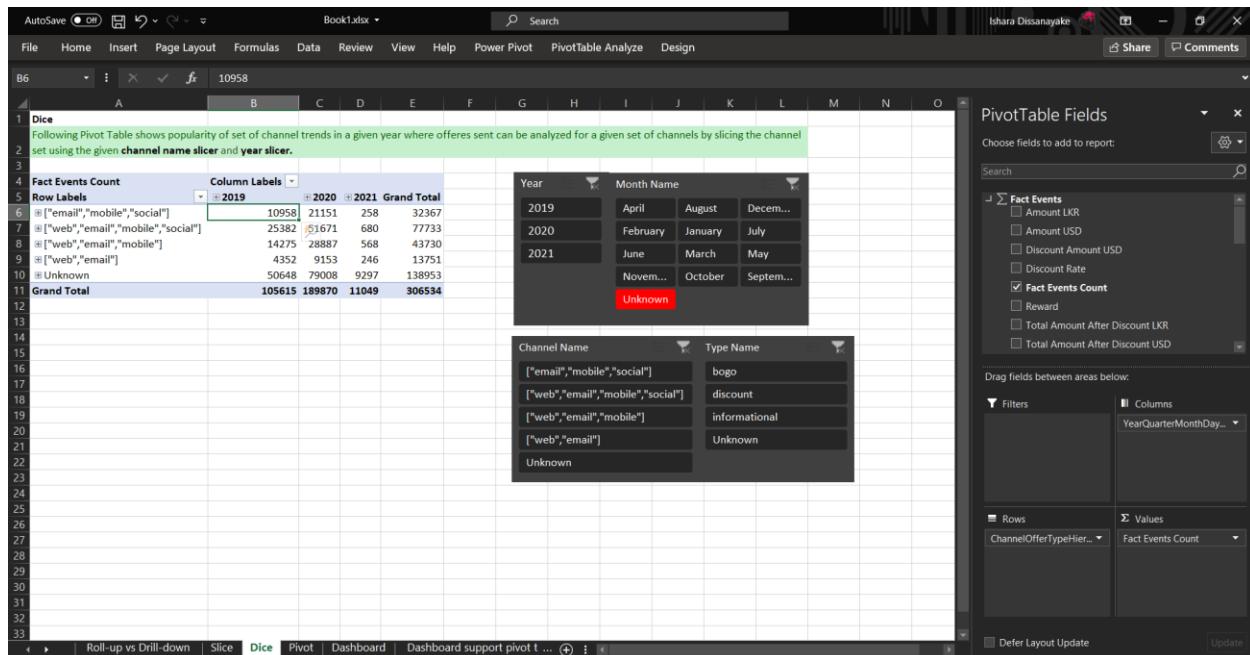


Figure 3.5. 1. Before the Dice OLAP operation was performed

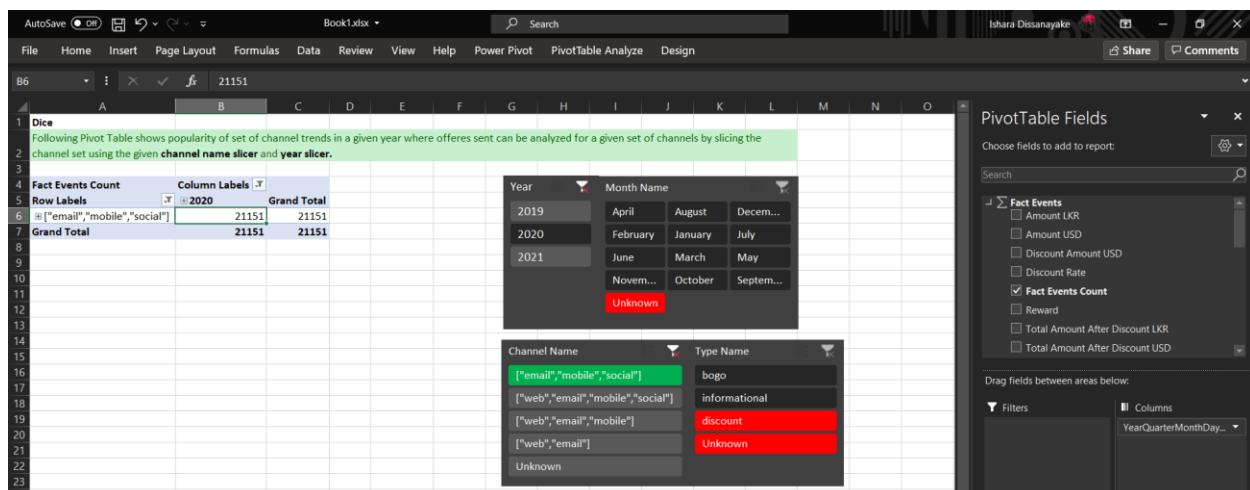


Figure 3.5. 2. After Dice Operation was performed by slicing data by year and Channel Name as shown in the figure

### 3.6. Pivot Operation

Pivot was performed on a pivot table which was designed to analyze the total transactions done by customers based on their Gender, Year, and County Code wise. In here the pivot operation was demonstrated by inter-changing the Year from being a column to being a row in the pivot table. It can be observed how the measure values are changing when the pivot is done and how the grand total of all transactions, at the very end, remains unchanged.

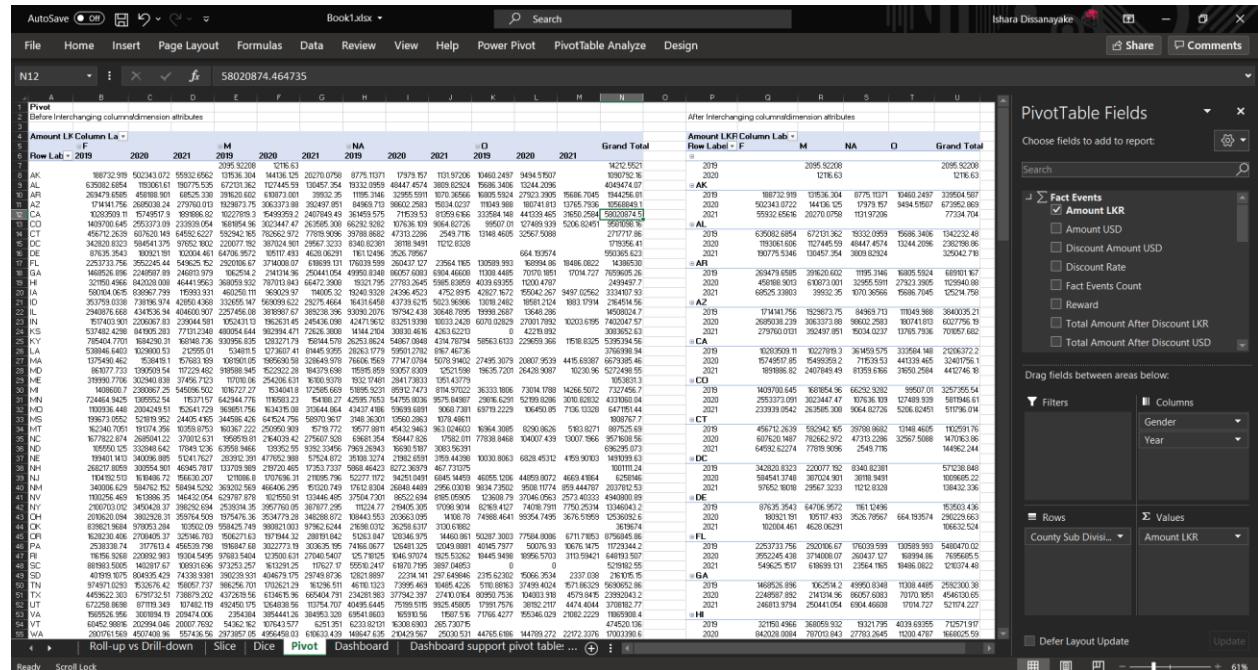


Figure 3.6. 1. Two pivot tables with same set of dimension and measures, left table has year as a column, right table was modified by changing year as a row. Notice that the measures have changed although dimensions are the same.

### 3.7. Excel Dashboard

In addition to the above pivot table analysis, a dashboard like data visualization was implemented using Microsoft excel pivot charts and tables along with slices to filter-out the results on the dashboard. The dashboard that was created is presenting data analytics mainly based on offers sent by Starbucks to their customers analyzed according to time and type of the offers, Total transactions done based on time, The popularity of offer sending channels based on time and type of the offers, and finally, the total transactions done based on the county code, and the gender of the customers.

It was designed in a user-friendly manner and in a way that the analytics can be easily interpreted using visualizations rather than looking at complex pivot tables. Pivot charts such as Line, Bar,

Radar charts were mainly used with slicers to filter the data faster than having to use drop down check lists.

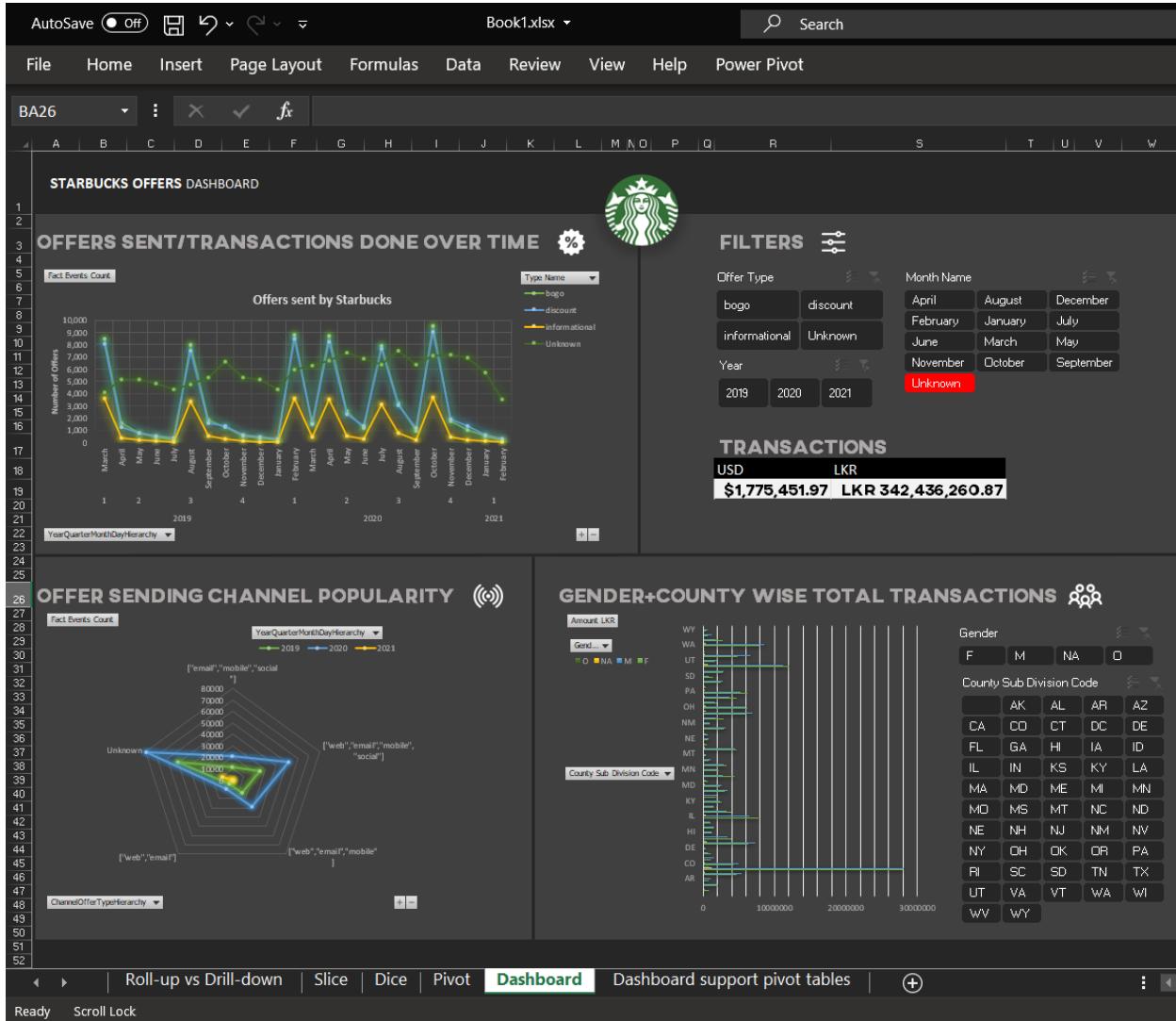


Figure 3.7. 1. Excel Dashboard made using Pivot Tables and Pivot Charts

Slicers given in the dashboard are connected to several pivot tables, making it possible to filter several charts at once. However, the slicers county and gender are only presented for the bottom-right corner chart only.

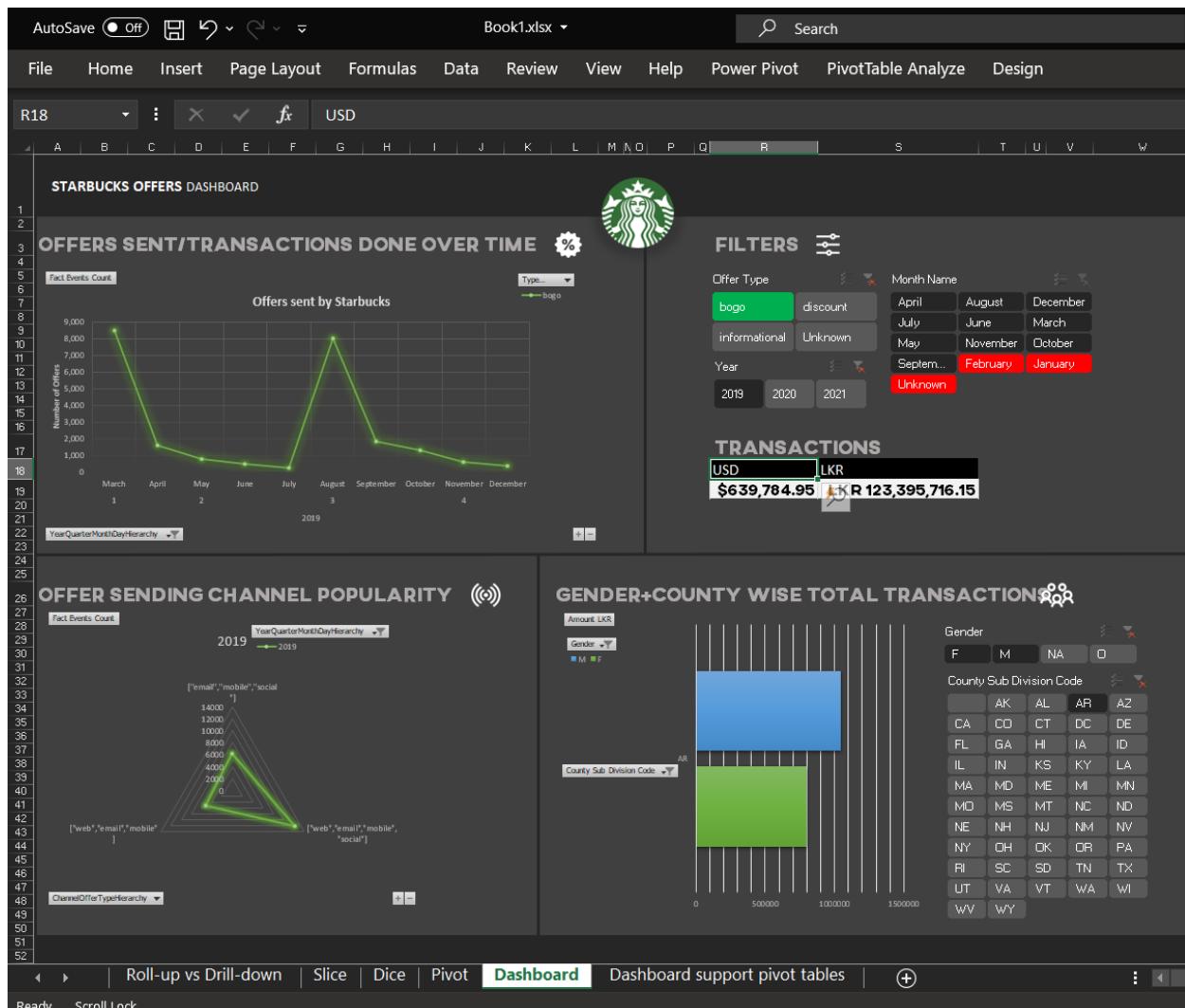


Figure 3.7. 2. The above instance of the dashboard shows,

- I. Number of "Bogo" offers sent in year 2019 in top-left chart
- II. Number of "Bogo" offers sent using special channels in the bottom-left chart
- III. Total transactions done by Male and Female customers in "AZ"-Arizona county in bottom-right chart
- IV. Total Transactions done by all customers in 2019 in "Transactions" (top-right) in USD and LKR

## 4. SSRS Reports

### 4.1. Configuring the Report Server

Before preparing SSRS Reports, the report server and web portal were set up using the Report Server Configuration Manager.

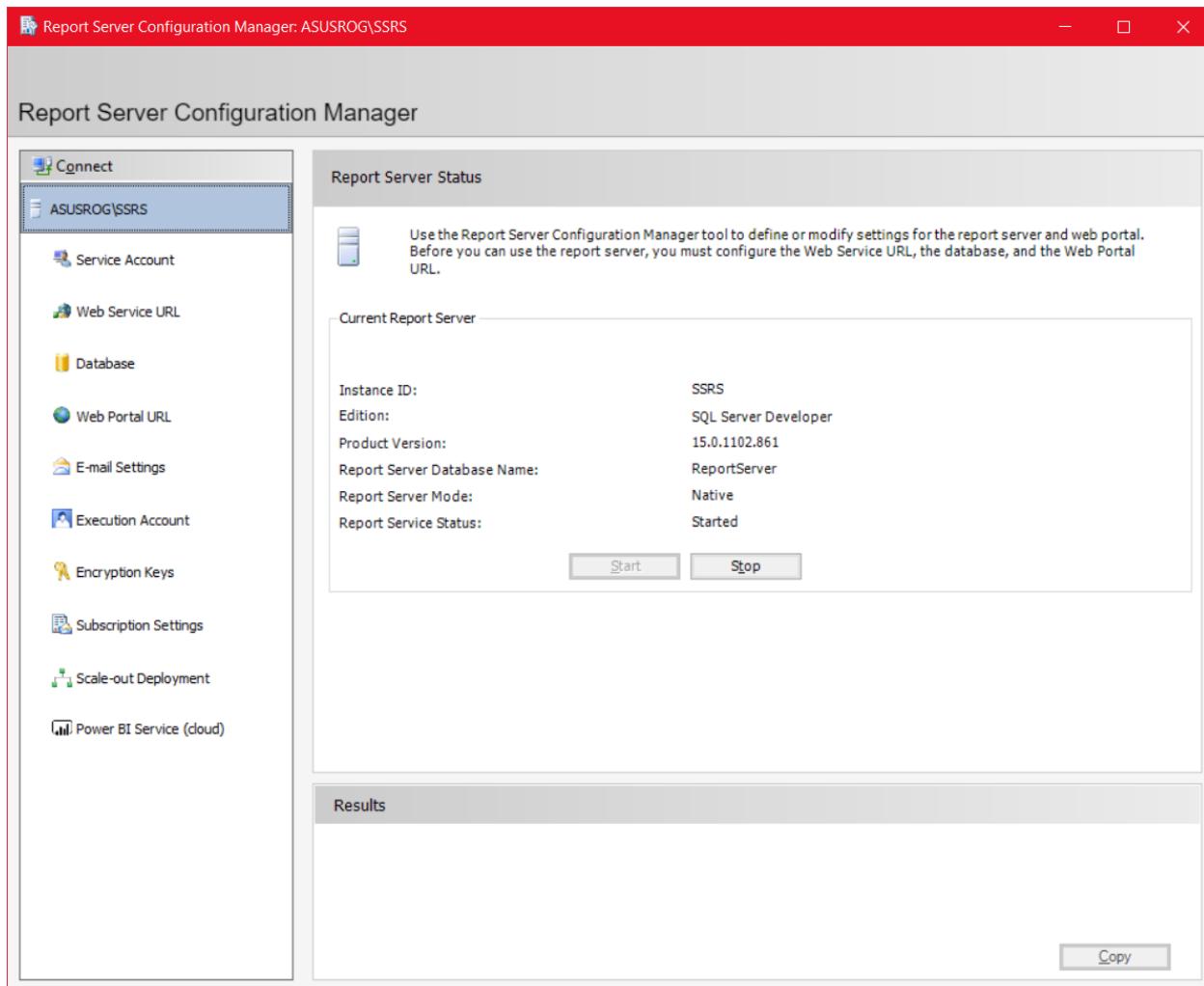


Figure 4.1. 1. Report Server Running after configurations were done

Based on the configurations, <http://asusrog/ReportServer> was given as the server URL and [http://asusrog/Reports/browse/Starbucks\\_SSRS](http://asusrog/Reports/browse/Starbucks_SSRS) was the folder located within the web portal which was created to store the reports to be built using Report Builder.

## 4.2. Report 1 – SSRS Report with a Matrix

The first report was created as “County and Gender wise YOY Transactions Report” which analyses the total transaction amount based on county and gender in a yearly manner. First, the Data warehouse was added as the Data Source to the Report Builder providing valid credentials.

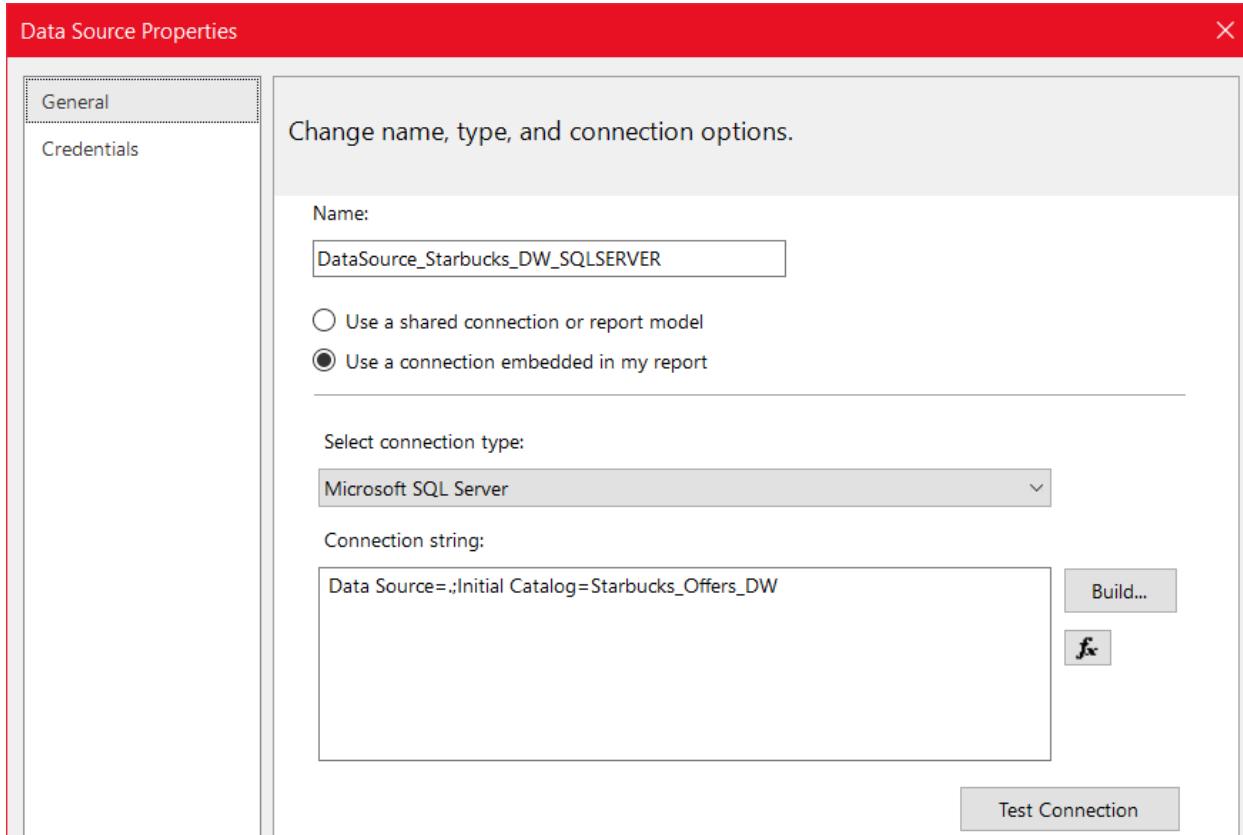


Figure 4.2. 1. Adding data warehouse as a Data Source to the Report Builder Tool

Then the necessary data were retrieved using the data source created, by providing a valid SQL Query using the Query Designer in the Dataset Properties and testing the validity of the query designed by running the query and examining the result. To retrieve all necessary fields from all dimension and fact tables, table joins were heavily used and, in this case, left outer joins were used with most of the tables. Examine the following figure to see the query used to retrieve data from the data warehouse directly.

Query Designer

Edit as Text Import... Command type: Text

```
select fe.[PersonKey], fe.[EventName], fe.[Reward], fe.[AmountUSD], fe.[DiscountRate], fe.[DiscountAmountUSD], fe.[TotalAmountAfterDiscountUSD], dp.[AlternatePersonID], dp.[Gender], dp.[Age], dp.[Income], dp.[YearOfBirth], dp.[City], dp.[CountySubDivisionCode], dp.[PostalCode], dp.[Street1], dp.[Street2], dp.[Street3], do.[AlternateOfferID], do.[Difficulty], do.[Duration], dot.[TypeName], dc.[ChannelName], dd.[Date], dd.[DayName], dd.[WeekOfMonth], dd.[MonthName], dd.[QuarterName], dd.[Quarter], dd.[Year], dcr.[ExchangeRate], dcr.[SourceCurrencyType]
from [dbo].[FactEvents] fe
left outer join [dbo].[DimProfile] dp on fe.[PersonKey] = dp.[PersonSK]
left outer join [dbo].[DimOffer] do on fe.[OfferKey] = do.[OfferSK]
left outer join [dbo].[DimChannel] dc on do.[ChannelKey] = dc.[ChannelSK]
left outer join [dbo].[DimOfferType] dot on do.[OfferTypeKey] = dot.[OfferTypeSK]
left outer join [dbo].[DimDate] dd on fe.[EventDateKey] = dd.[DateKey]
left outer join [dbo].[DimCurrency] dcr on fe.[CurrencyRateKey] = dcr.[CurrencyRateSK]
```

PersonKey	EventName	Reward	AmountUSD	DiscountRate	DiscountAoun...	TotalAmountAft...	AlternatePersonID
3848	offer completed	10	0.0000	0.00	0.0000	0.0000	3a298dbf99ef4...
3632	transaction	0	0.4500	0.00	0.0000	0.4500	375043f439a44...
7952	offer viewed	0	0.0000	0.00	0.0000	0.0000	77caa79f70754...
1721	offer viewed	0	0.0000	0.00	0.0000	0.0000	1ab040cd863d...
9411	transaction	0	0.8500	0.00	0.0000	0.8500	8d0222af15e4...
6998	transaction	0	9.2200	0.00	0.0000	9.2200	6960918e3a94...
6998	offer completed	5	0.0000	0.00	0.0000	0.0000	6960918e3a94...
366	offer viewed	0	0.0000	0.00	0.0000	0.0000	05929d20f3c24...
13529	transaction	0	1.5300	0.00	0.0000	1.5300	cb351d3c1d364...

Figure 4.2. 2. Query Designed to retrieve dataset from the data warehouse to build the report

Following Fields were selected, and the layout settings were left as it is From the New Table or Matrix Window and the Matrix was created.

New Table or Matrix

Arrange fields

Available fields

- PersonKey
- EventName
- Reward
- AmountUSD
- DiscountRate
- DiscountAmountUSD
- TotalAmountAfterDiscountUSD
- AlternatePersonID
- Gender
- Age
- Income
- YearOfBirth
- City
- CountySubDivisionCode
- PostalCode
- Street1
- Street2

Column groups

- Year
- Gender

Row groups

- CountySubDivisionCode

Values

- Sum(AmountUSD)
- Sum(TotalAmountAfterDiscountUSD)

Figure 4.2. 3. Selected Row Groups, Values, and Column Groups for the Matrix

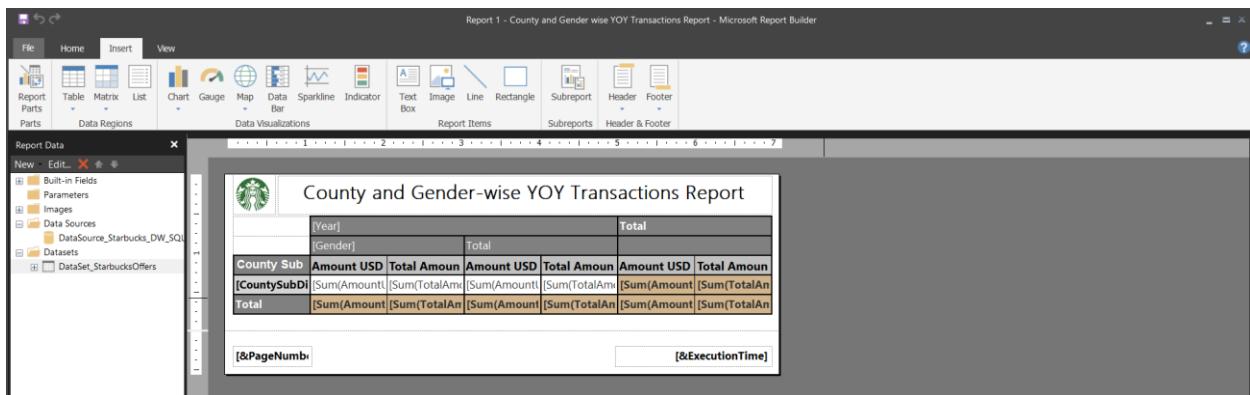


Figure 4.2. 4. Matrix Report was Designed using color codes and Images using the report builder tools

The cells were properly formatted as currency holding cells and appropriate type of currency selected using the “text box properties” for each cell which holds currency values.

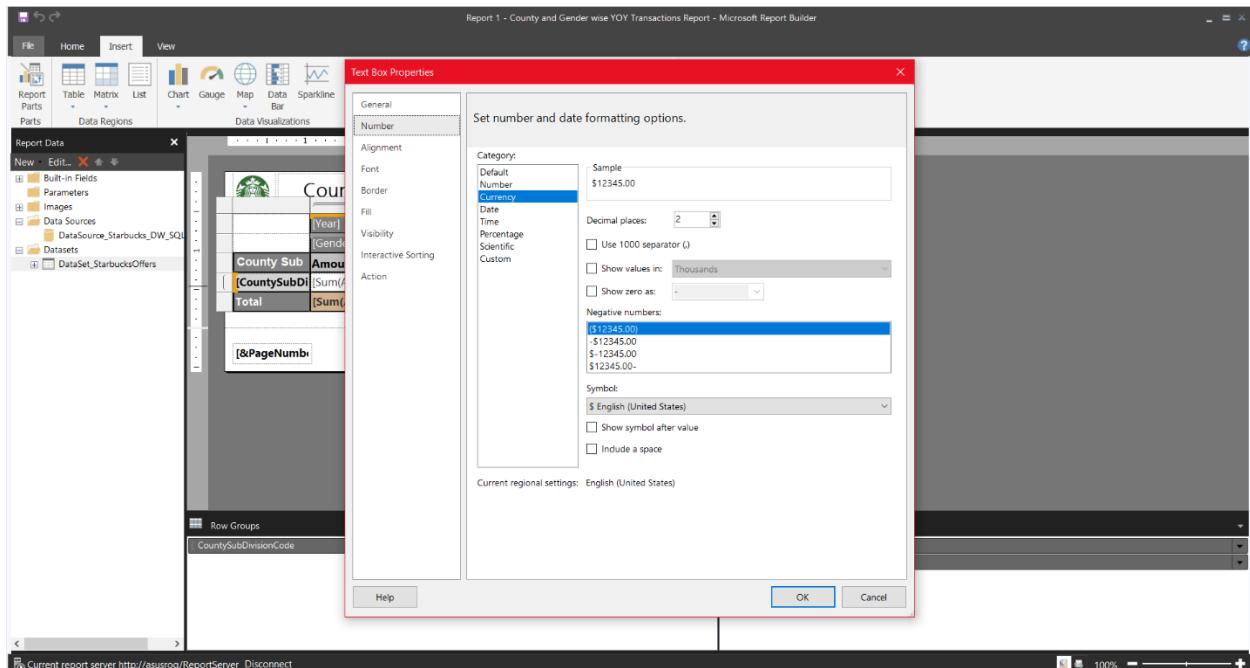


Figure 4.2. 5. Text Box properties was used to format the cells

Finally, the report was saved in the server using “save as” command and it was opened and inspected using the web portal of the SSRS as shown in the following figure.

County Sub Division Code	2019			2020			2021			Total	
	Total			Total			Total			Total	
	Amount USD	Total Amount After Discount USD	Amount USD	Total Amount After Discount USD	Amount USD	Total Amount After Discount USD	Amount USD	Total Amount After Discount USD	Amount USD	Total Amount After Discount USD	
	\$11.20	\$10.86	\$62.80	\$61.12					\$74.00	\$71.99	
AK	\$1764.53	\$1734.92	\$3495.00	\$3418.53	\$403.79	\$396.32	\$5663.32	\$5549.77			
AL	\$6952.97	\$6821.88	\$1231.73	\$1208.65	\$1704.29	\$1659.48	\$20970.99	\$20566.01			
AR	\$3567.99	\$3503.15	\$5876.96	\$5756.70	\$653.00	\$640.46	\$10097.95	\$9900.31			
AZ	\$19954.86	\$19576.36	\$31184.10	\$30623.23	\$3657.07	\$3574.91	\$54796.03	\$53774.50			
CA	\$109910.72	\$107855.00	\$167799.41	\$164557.34	\$23045.89	\$22614.73	\$300756.02	\$295027.07			
CO	\$16870.70	\$16531.17	\$30177.29	\$29649.89	\$2677.36	\$2624.39	\$49725.35	\$48805.45			
CT	\$5719.11	\$5605.77	\$7635.11	\$7481.46	\$759.75	\$745.80	\$14113.97	\$13833.04			
DC	\$2959.82	\$2901.05	\$5263.83	\$5159.68	\$728.99	\$711.60	\$8952.64	\$8772.33			
DE	\$795.84	\$782.15	\$1504.88	\$1478.77	\$562.02	\$560.27	\$2862.74	\$2821.19			
FL	\$28403.01	\$27847.85	\$39891.13	\$39131.46	\$6341.96	\$6240.65	\$74636.10	\$73219.96			
GA	\$13436.54	\$13189.65	\$23590.39	\$23162.18	\$2728.47	\$2668.57	\$39755.40	\$39020.39			
HI	\$3696.75	\$3625.40	\$8648.95	\$8483.56	\$620.10	\$610.10	\$12965.80	\$12719.06			
IA	\$5733.20	\$5626.12	\$10271.86	\$10071.59	\$1278.53	\$1257.08	\$17283.59	\$16954.79			

Figure 4.2. 6. Report with the Matrix opened using the SSRS Web Portal

### 4.3. Report 2 – Report with more than one parameter

In the same way as mentioned above, a new report was created and the data source and the dataset were added and a table was inserted which has Offer Type Name, Channel Name, and Year as columns. The expected report has two parameter lists Offer Type list and Channel list. When the Offer Type list item is selected, the Channel List is updated according to the selected offer type and then a channel name can be selected. When both are selected, the report can be viewed which shows year-wise rewards given through the selected offer type and the specific channel.

To add the parameters as lists, two new parameters were created namely ‘offer\_type\_para’ and ‘channel\_type\_para’. To populate the parameters with list of values, first, the list of distinct offer types was retrieved as a new dataset.

To add the matching list of Channel Names for a selected Offer Type, another dataset was retrieved as a list of channels where the offer type is equal to the parameter value of offer\_type\_para and the list was assigned to channel\_type\_para parameter. Carefully inspect the queries given in the following figures to see how the two lists were retrieved.

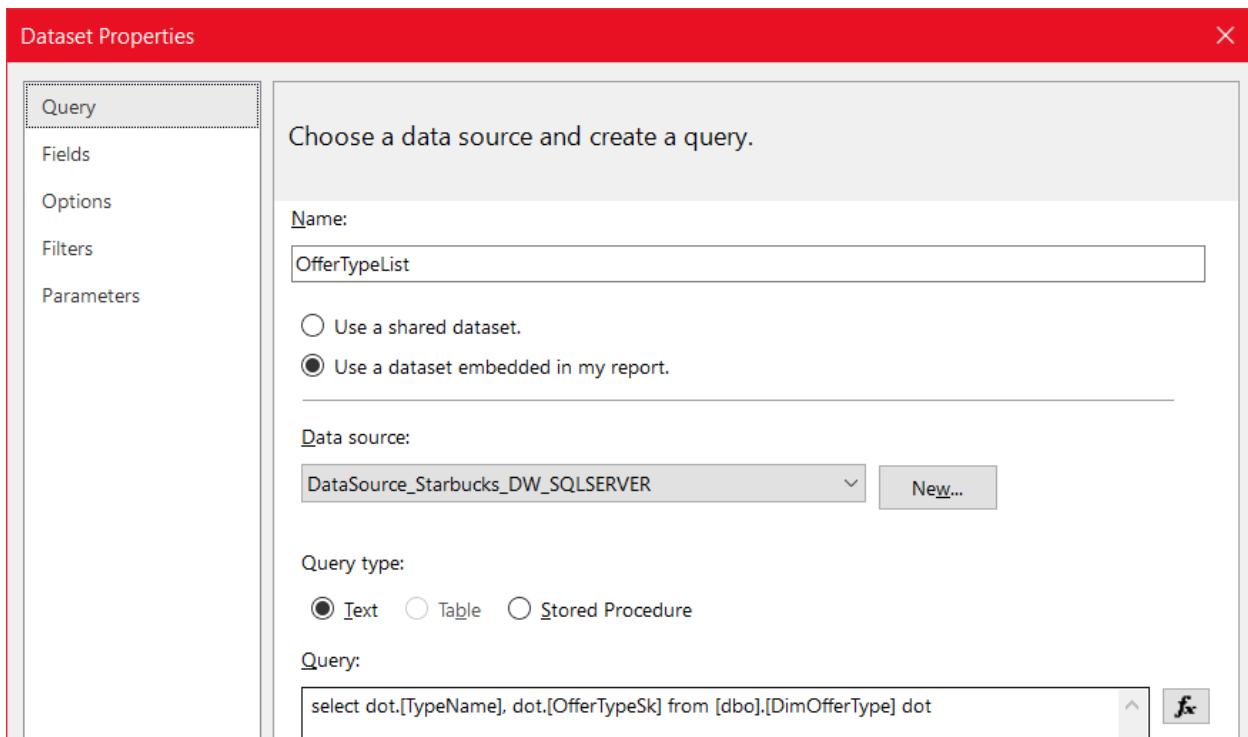


Figure 4.3. 2. Retrieving list of distinct offer types.

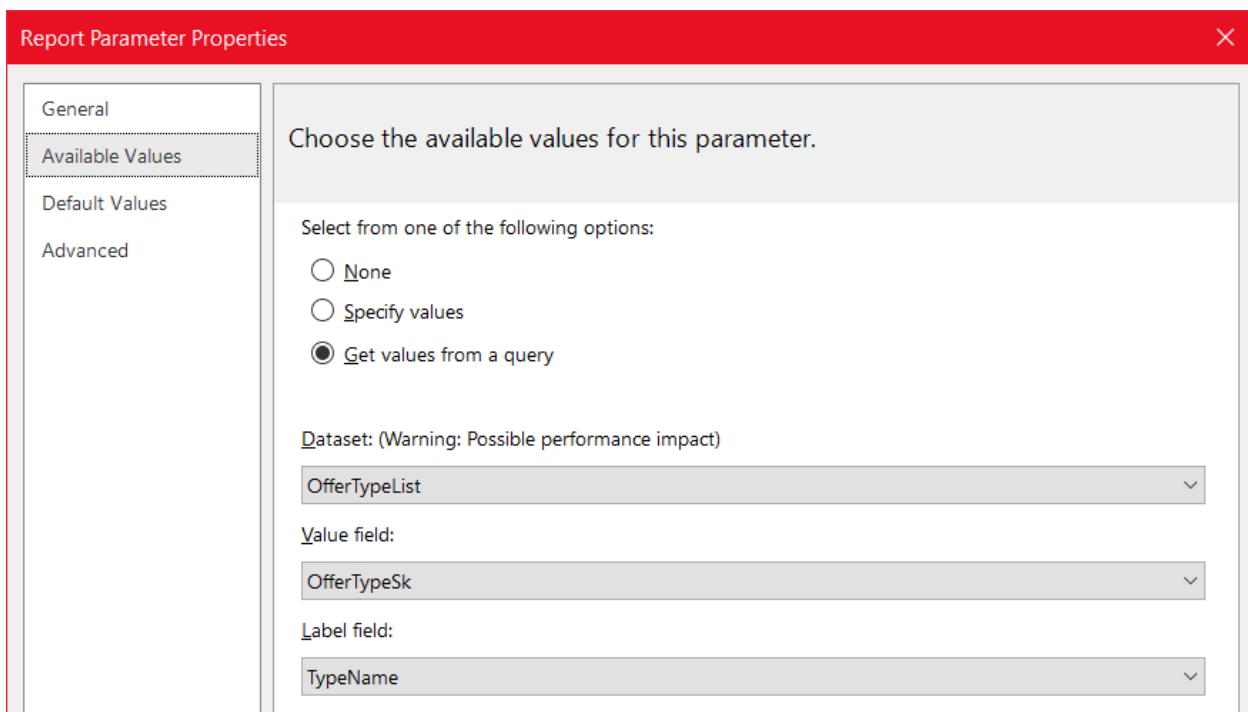


Figure 4.3. 1. Assigning Retrieved offer type names to the parameter as a list

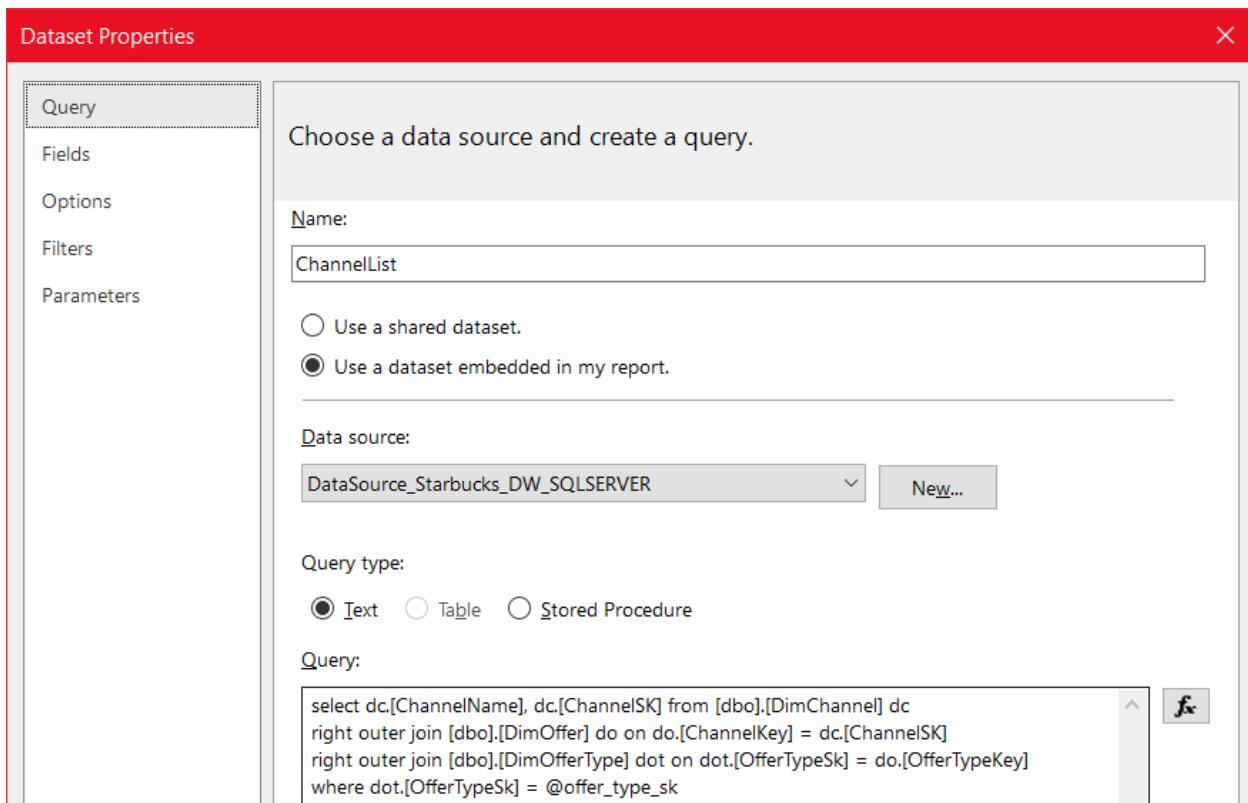


Figure 4.3. 4. Retrieving the list of Channel dataset based on the offer type selected by using the OfferTypeSK

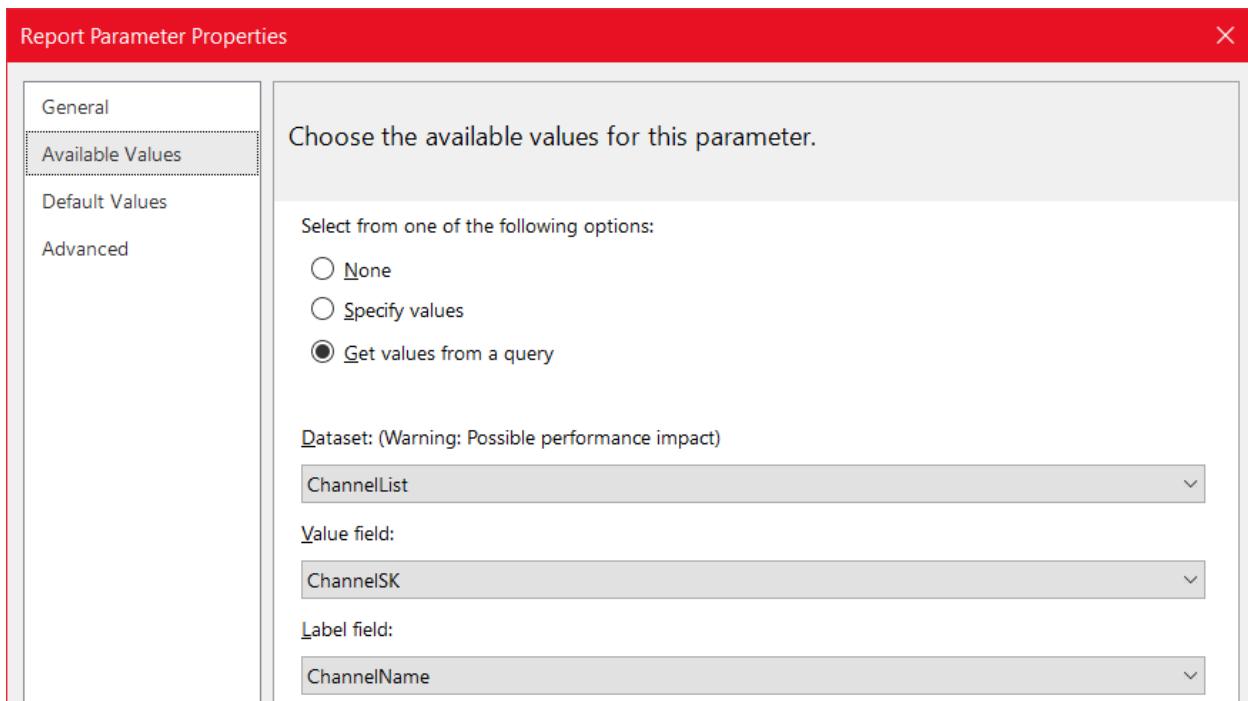


Figure 4.3. 3. How List of channels was assigned to the parameter channel\_type\_para

Additionally, a pie chart was added to visualize the changes when selecting different offer types and channel names by using the chart wizard selecting year as the series Group, Type Name and Channel Name as Category Groups, and Summation of Rewards received as the Value.

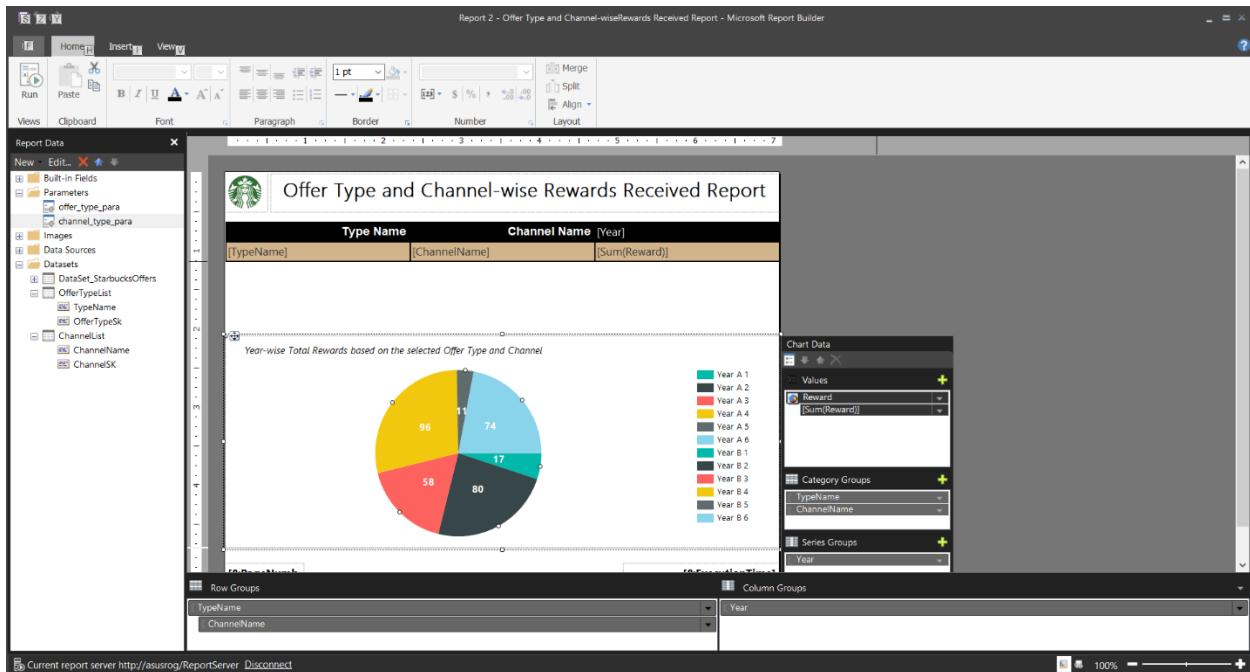


Figure 4.3. 5. Report layout Prepared with a pie chart to visualize the changing data in the table

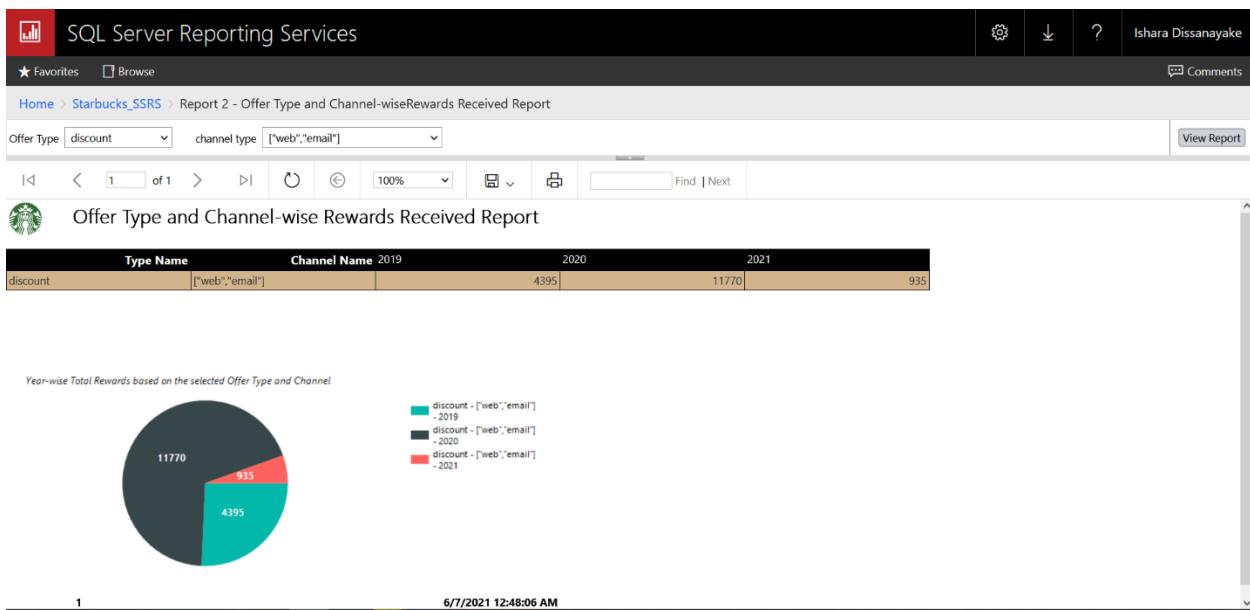


Figure 4.3. 6. Report with two list parameters viewed using the SSRS Web Portal. (Report Instance: offer type=discount and channel type = ["web", "email"])

#### 4.4. Report 3 – SSRS Drill-down Report

A drill-down report is basically a report which contains row groups or column groups which are expandable just like demonstrated in excel drill-down OLAP operation. Here, a report was built which gives Transactions done by consumers county-wise as a year-over-year report. However, the drill-down feature is enabled by allowing the year to be further expandable to view monthly transactions using the expand/collapse buttons in the report.

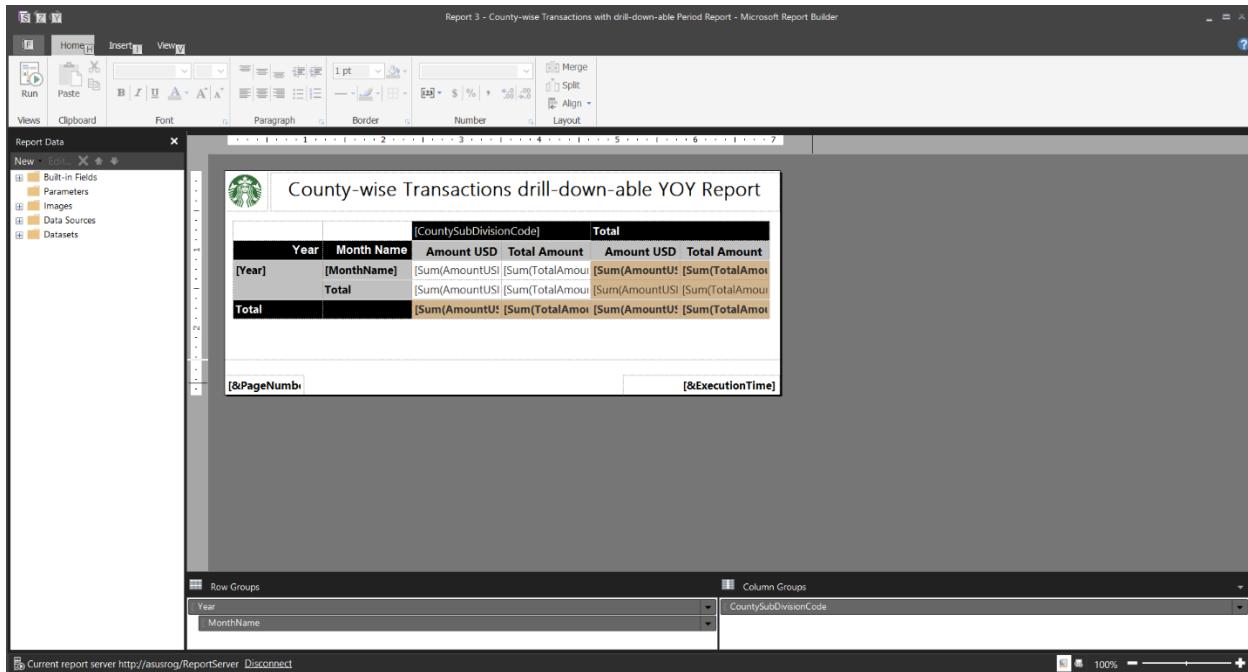


Figure 4.4. 1. Report layout design in Report Builder with drill-down-able row groups from Year-wise data to monthly transaction data.

		AK	AL	AR	AZ	CA
Year	Month Name	Amount USD After Discount USD				
2019	Total	\$11.20	\$10.86	\$1,764.53	\$1,734.92	\$6,952.97
2020	Total	\$62.80	\$61.12	\$3,495.00	\$3,418.53	\$12,313.73
2021	Total			\$403.79	\$396.32	\$1,704.29
<b>Total</b>		<b>\$74.00</b>	<b>\$71.99</b>	<b>\$5,663.32</b>	<b>\$5,549.77</b>	<b>\$20,970.99</b>
				<b>\$20,566.01</b>	<b>\$10,097.95</b>	<b>\$9,900.31</b>
					<b>\$54,796.03</b>	<b>\$53,774.50</b>
						<b>\$300,756.02</b>
						<b>\$295,027.4</b>

Figure 4.4. 2. Report in the SSRS Web Portal as yearly data (before drill-down)

Year	Month Name	AK		AL		AR		AZ		CA	
		Amount USD	Total Amount After Discount USD	Amount USD	Total Amount After Discount USD	Amount USD	Total Amount After Discount USD	Amount USD	Total Amount After Discount USD	Amount USD	Total Amo After Disco
2019	April		\$99.28	\$97.80	\$689.36	\$678.93	\$448.75	\$441.06	\$2,321.54	\$2,273.40	\$10,693.17
	August	\$0.00	\$0.00	\$191.16	\$188.28	\$666.18	\$655.01	\$310.65	\$304.79	\$1,763.88	\$1,730.98
	December	\$11.20	\$10.86	\$211.26	\$206.31	\$1,380.40	\$1,353.13	\$381.14	\$373.96	\$1,538.79	\$1,508.41
	July		\$165.86	\$164.75	\$537.73	\$527.00	\$266.38	\$262.41	\$1,655.56	\$1,619.58	\$10,359.68
	June		\$176.25	\$174.06	\$460.12	\$451.96	\$327.26	\$320.01	\$1,559.61	\$1,533.06	\$10,216.36
	March		\$186.68	\$182.57	\$504.16	\$494.17	\$284.80	\$279.01	\$1,705.17	\$1,672.49	\$9,543.34
	May		\$91.32	\$89.65	\$588.56	\$575.28	\$336.72	\$328.93	\$1,855.28	\$1,821.61	\$10,528.51
	November		\$132.42	\$131.14	\$729.05	\$715.85	\$388.47	\$381.68	\$2,514.12	\$2,454.33	\$11,569.56
	October	\$0.00	\$0.00	\$243.61	\$238.67	\$662.67	\$650.72	\$460.14	\$452.87	\$2,857.14	\$2,820.41
	September		\$266.69	\$261.69	\$734.74	\$719.84	\$363.68	\$358.41	\$2,183.77	\$2,142.10	\$11,688.63
2020	Total	\$11.20	\$10.86	\$1,764.53	\$1,734.92	\$6,952.97	\$6,821.88	\$3,567.99	\$3,503.15	\$19,954.86	\$19,576.36
	Total	\$62.80	\$61.12	\$3,495.00	\$3,418.53	\$12,313.73	\$12,084.65	\$5,876.96	\$5,756.70	\$31,184.10	\$30,623.23
2021	Total	\$403.79	\$396.32	\$1,704.29	\$1,659.48	\$653.00	\$640.46	\$3,657.07	\$3,574.91	\$23,045.89	\$22,6
	Total	\$74.00	\$71.99	\$5,663.32	\$5,549.77	\$20,970.99	\$20,566.01	\$10,097.95	\$9,900.31	\$54,796.03	\$53,774.50

Figure 4.4. 3. Same Report shown in Figure 4.4.2. as monthly data (after drill-down)

## 4.5. Report 4 – SSRS Drill-Through Report

Drill Through reports consists of two reports. Clicking on data visible in one report can be configured to open-up a separate report which gives further analysis and visualizations based on the clicked data in the first report.

In this case, two reports were built as follows.

- Level 1 Report (Main) – Performance of Offer Types based on the rewards offered as a YOY report along with number of offers sent.
  - Offer Types-Rewards Data is given for all the years
  - Offer Count is based on all years
- Level 2 Report (Sub) – Performance of Selected Offer Type and Year in Level 1 report based on total rewards with monthly analysis of number of offers sent within the selected year.
  - Rewards are analyzed county-wise, based on the selected year and offer type
  - Offer count is based on months of the selected year

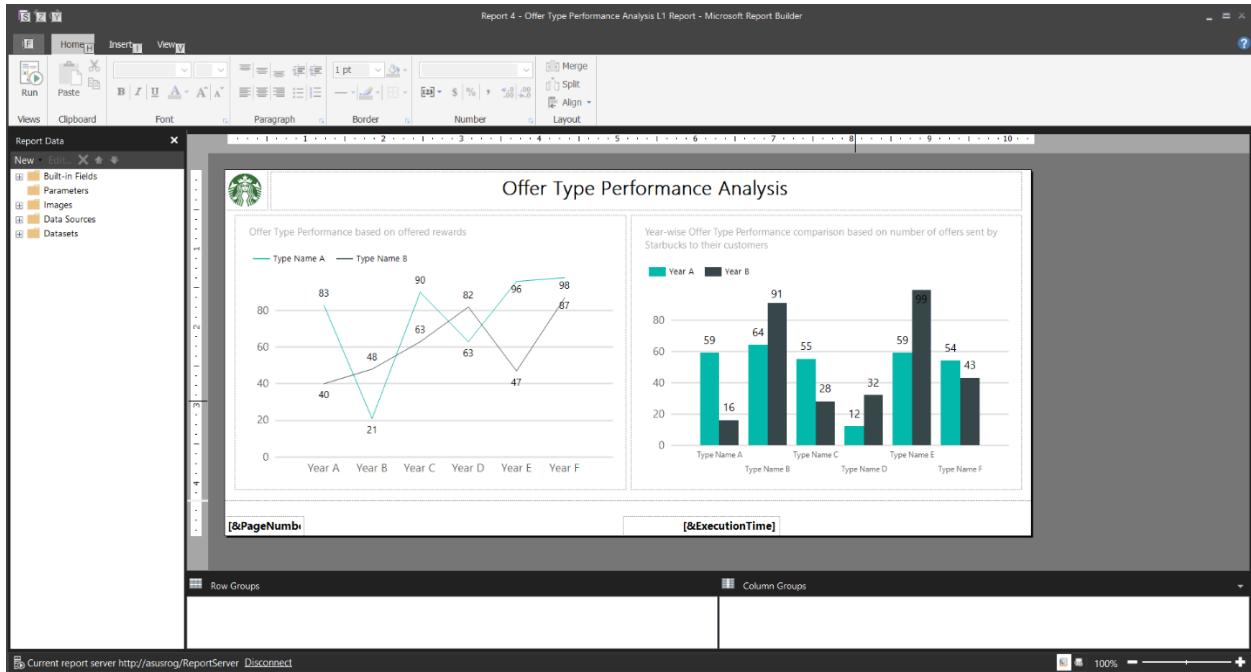


Figure 4.5. 1. Level 1 Report Design

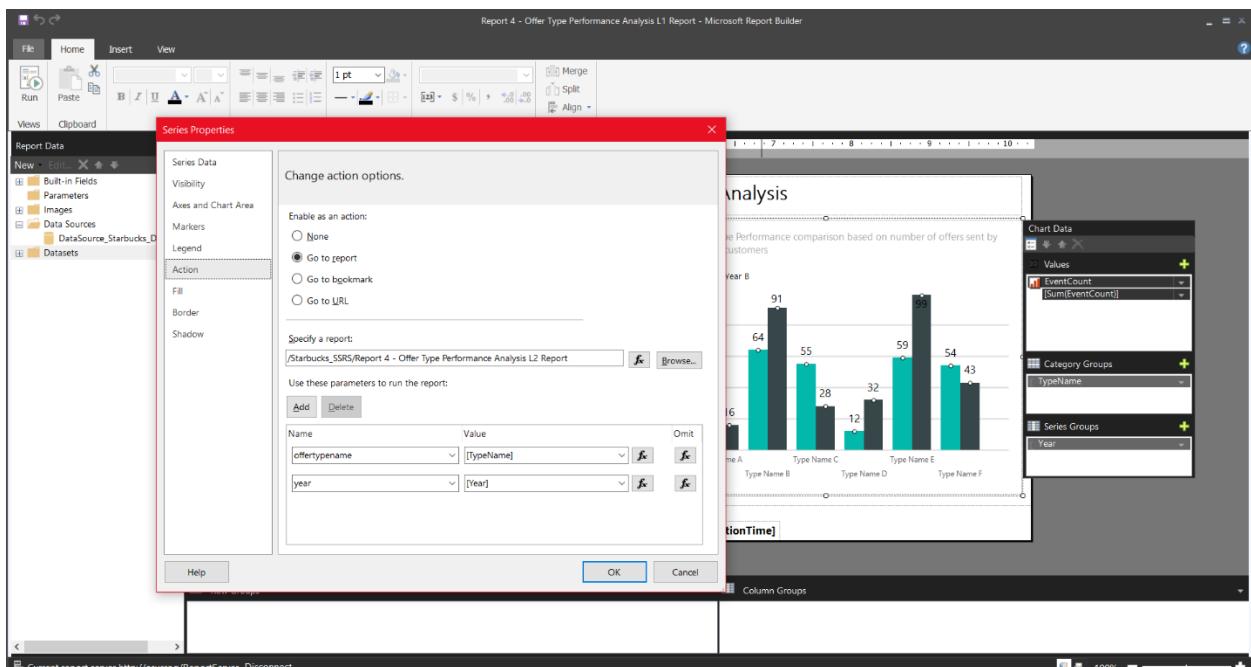


Figure 4.5. 2. The Right-side graph of the Level 1 report was added an action by editing the series properties to send the **year** and **offer type name** to the Level 2 Report as parameters.

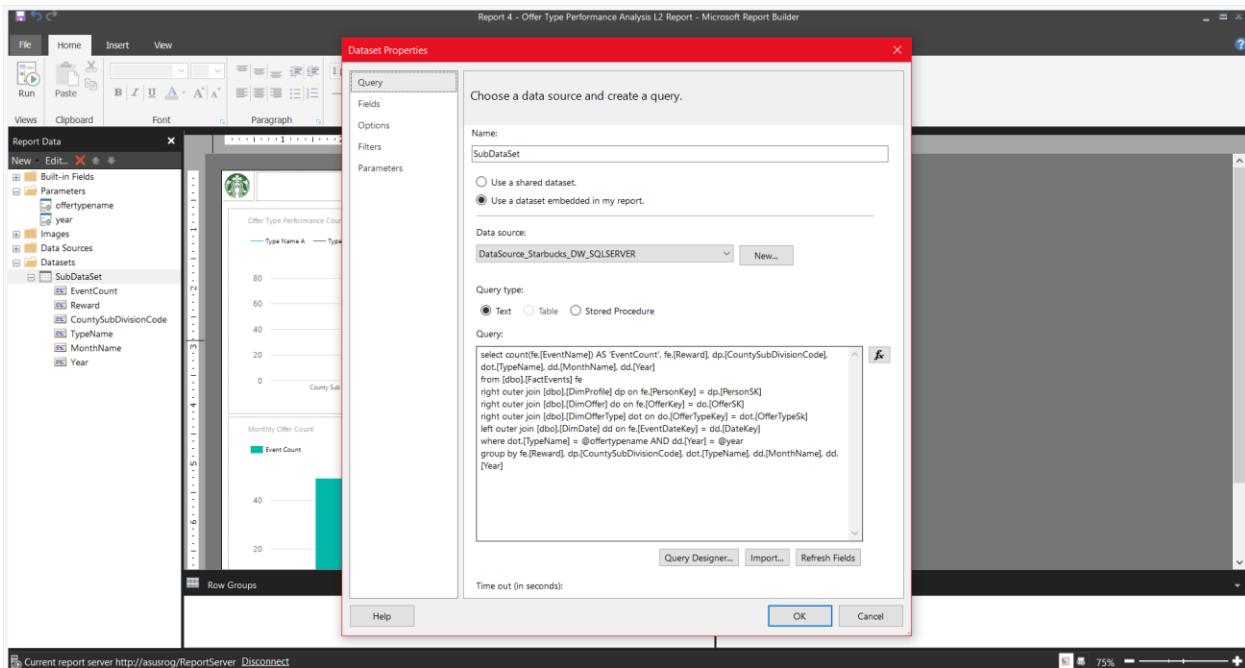


Figure 4.5. 3. In level 2 Report, a new dataset was retrieved based on the parameters received from the Level 1 report when a specific column of the column chart is clicked

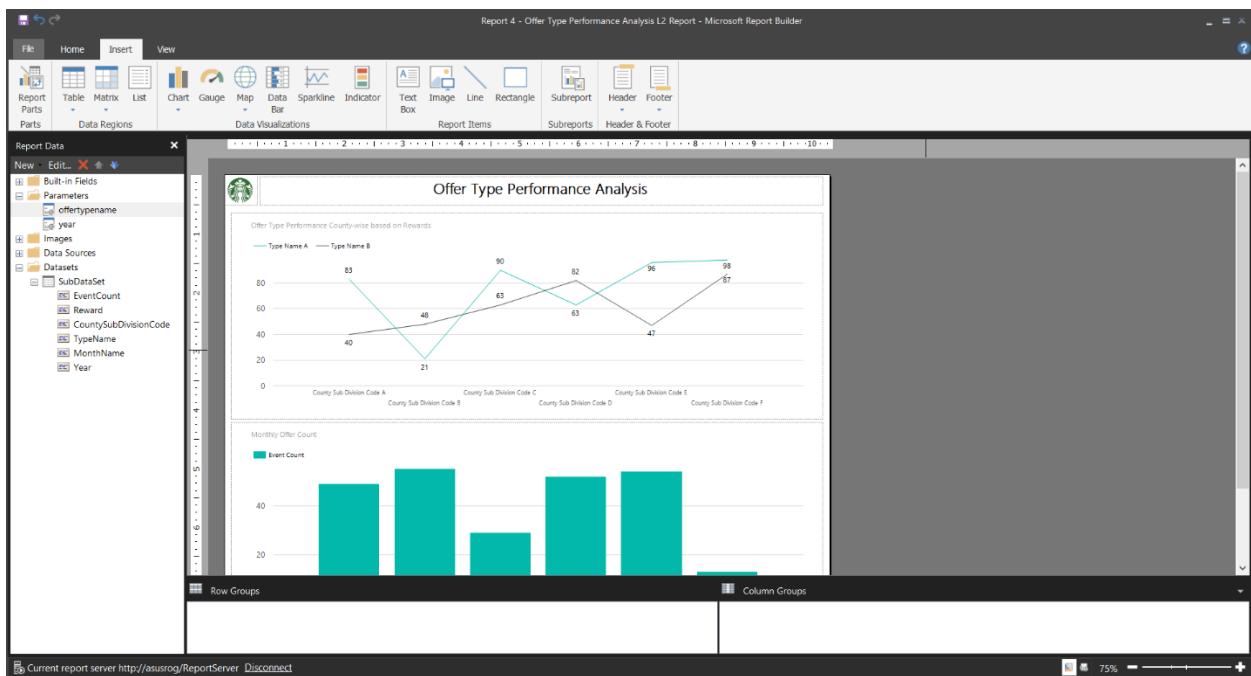


Figure 4.5. 4. Level 2 Report Designed to view Monthly and County-wise analysis for selected offer type and year in the main report.

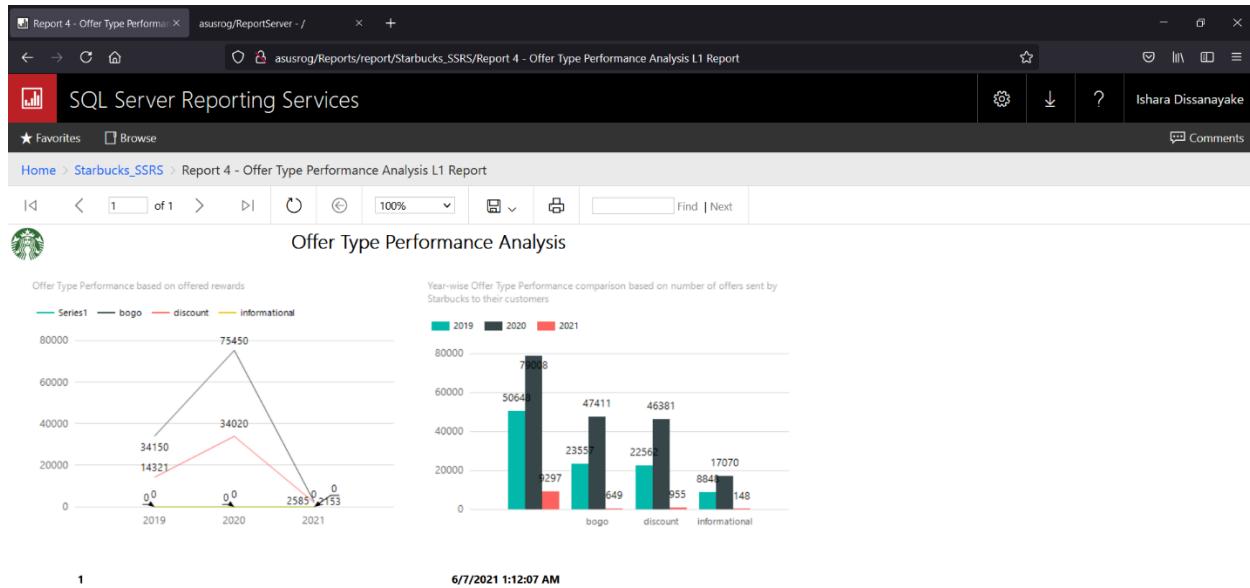


Figure 4.5. 5. Level 1 (Main) Report in SSRS Web Portal. (Note that from right-side column chart, **year 2020 column of Bogo Offer type** was clicked)

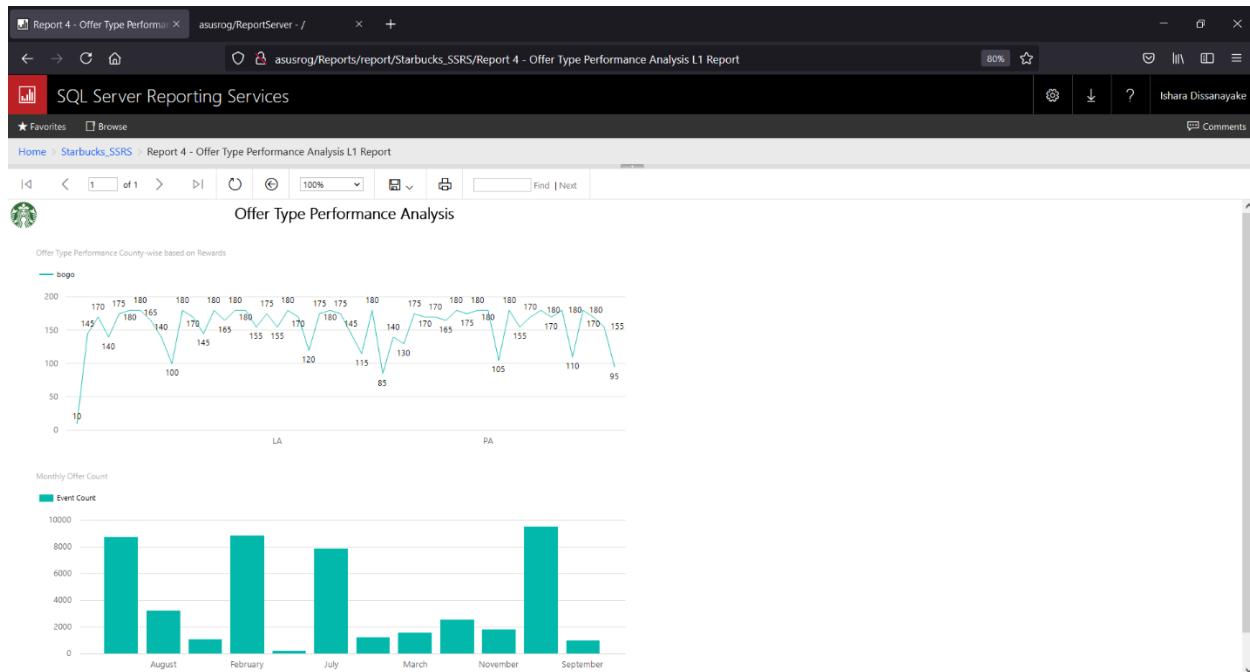


Figure 4.5. 6. Level 2 (Sub) Report in SSRS Web Portal. Note that after clicking year 2020 - Bogo column, Level 2 report views county-wise and monthly analyzed data for year 2020 for the "Bogo" offer type only, which is a drill-through.

## **5. References**

- [1].sqlservercentral.com, ‘SSAS Error: Duplicate attribute key found when processing’,  
<https://www.sqlservercentral.com/articles/ssas-error-duplicate-attribute-key-found-when-processing> [Accessed on: 02-June-2021]
- [2].microsoft.com, ‘Analysis Services documentation’, <https://docs.microsoft.com/en-us/analysis-services/?view=asallproducts-allversions> [Accessed on: 03-June-2021]
- [3].stackoverflow.com, ‘Can't add members to SSAS role through Visual Studio’,  
<https://stackoverflow.com/questions/56997515/cant-add-members-to-ssas-role-through-visual-studio> [Accessed on: 05-June-2021]