



**Data Warehousing & Business Intelligence  
(DS)  
3<sup>rd</sup> Year, 1<sup>st</sup> Semester**

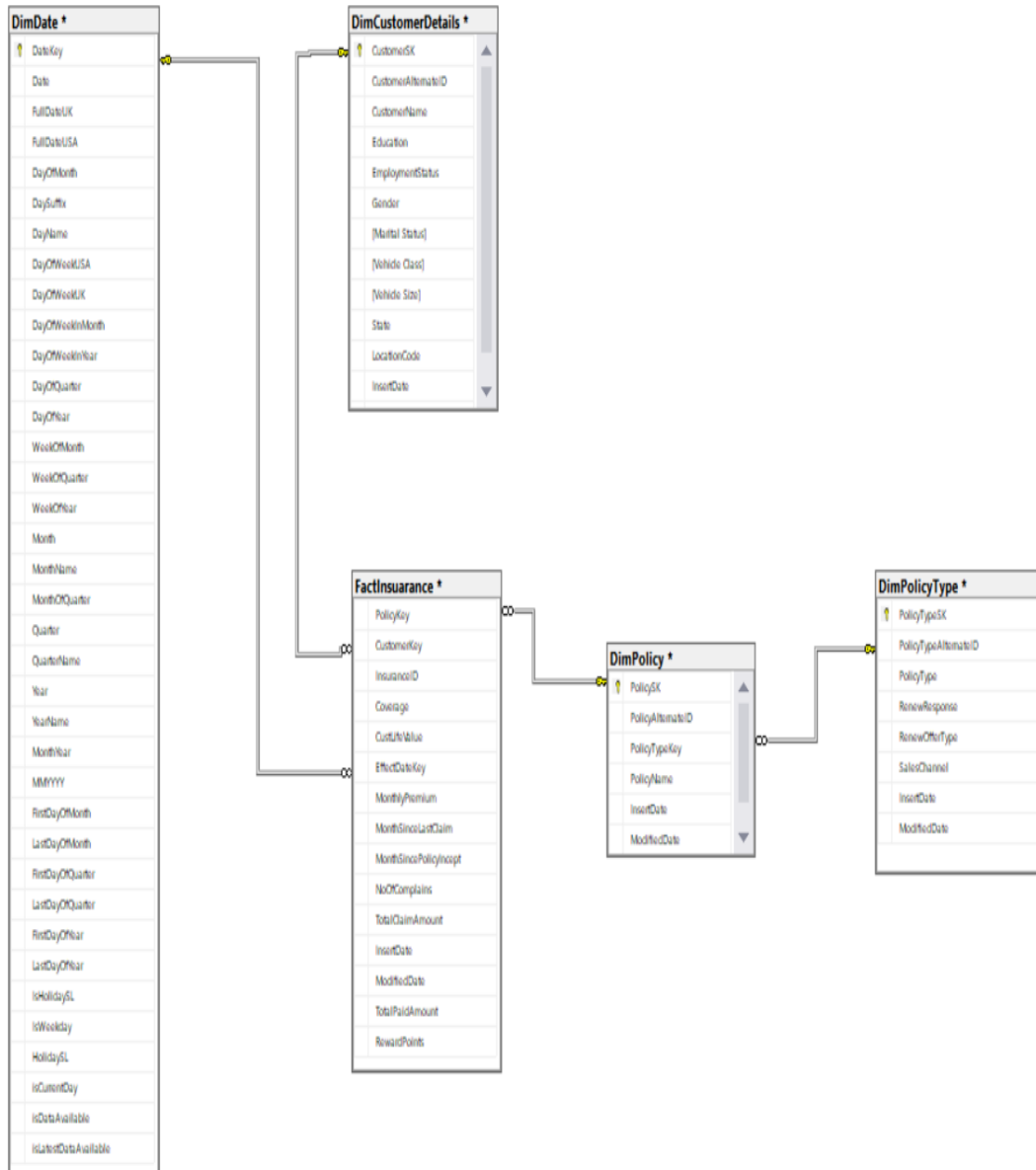
**Assignment 02**

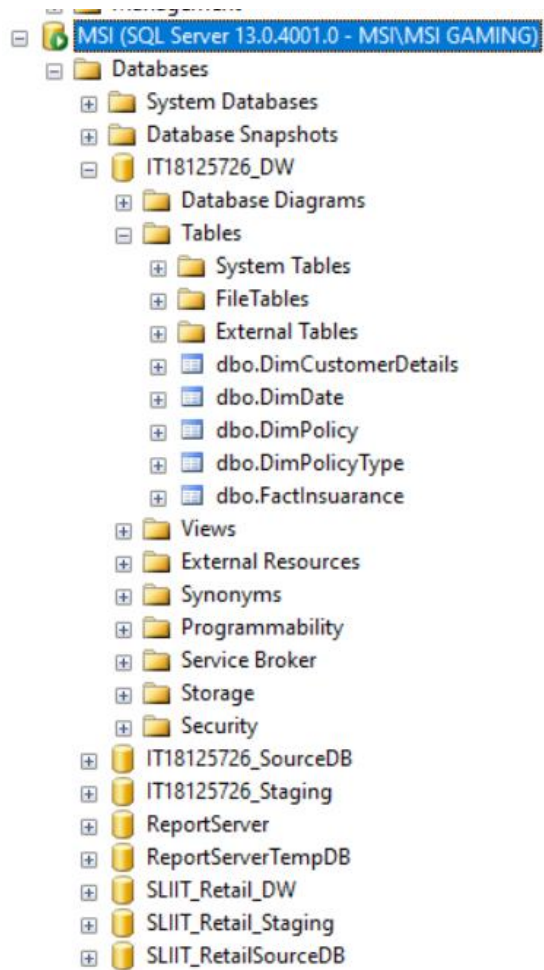
Submitted to  
Srilanka Institute of Information Technology

IT18125726  
L.V.A Kumarage  
Weekday batch

## Step 1: Data source for the assignment 2

- **IT18125726\_DW** that I have implemented and loaded with data in Assignment 1 as the data source for the assignment 2.

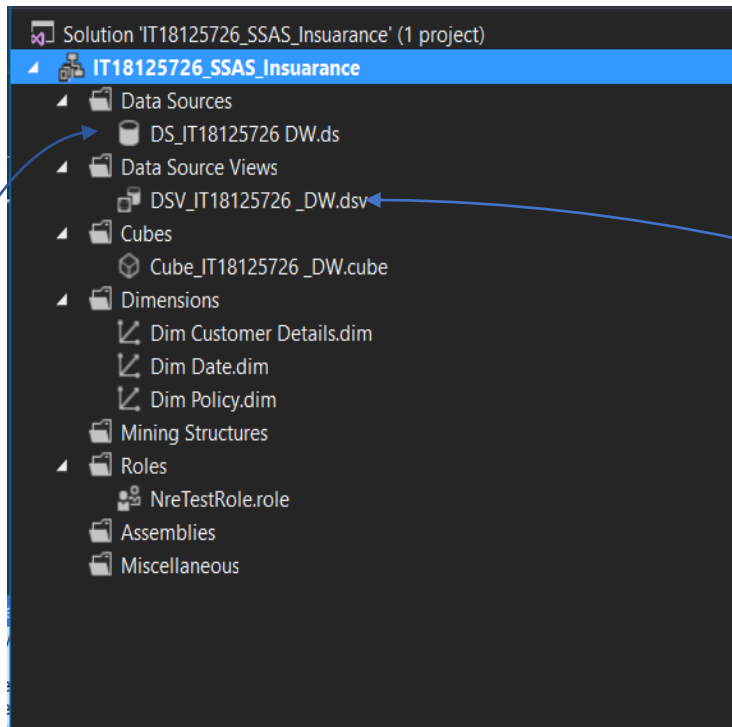




## **Step 2: SSAS Cube implementation**

Created a new Analysis Service Multidimensional and Data Mining Project called IT18125726\_SSAS\_Insurance.

Then I have configured each option from top to bottom as shown below to create a data cube



I created a data source based on new connection.

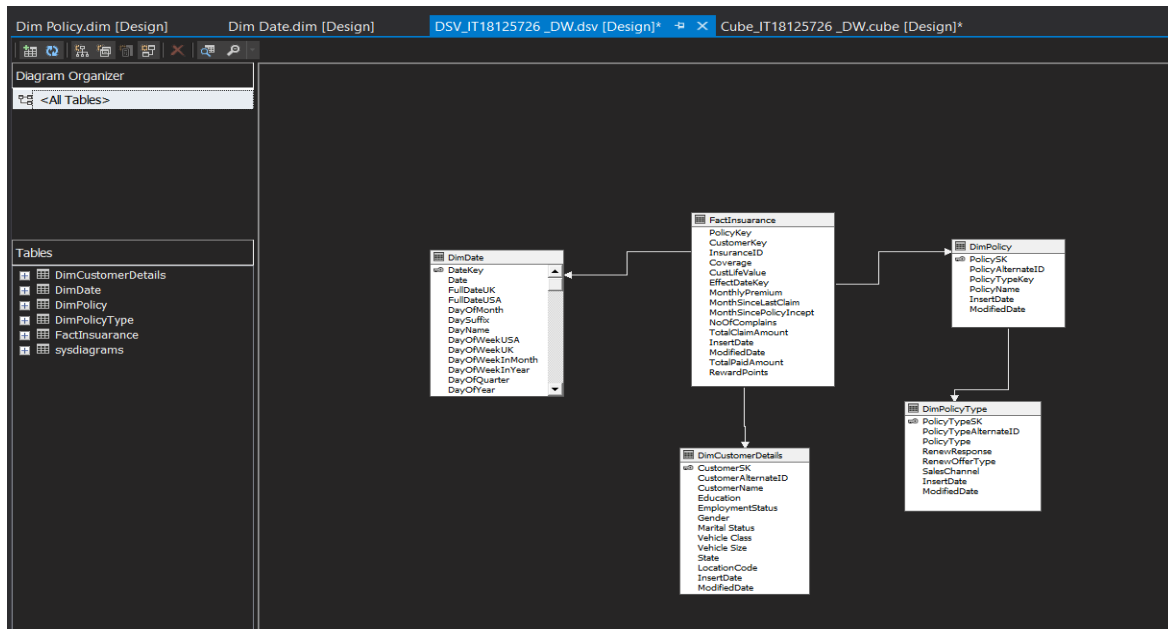
Once I completed the data source configuration, it created a new data source in solution explorer under data source.

- Creating data source view

Then I right clicked on data source views and select “New Data Source View”. Then in the welcome screen of the wizard, clicked “Next” to continue the configuration.

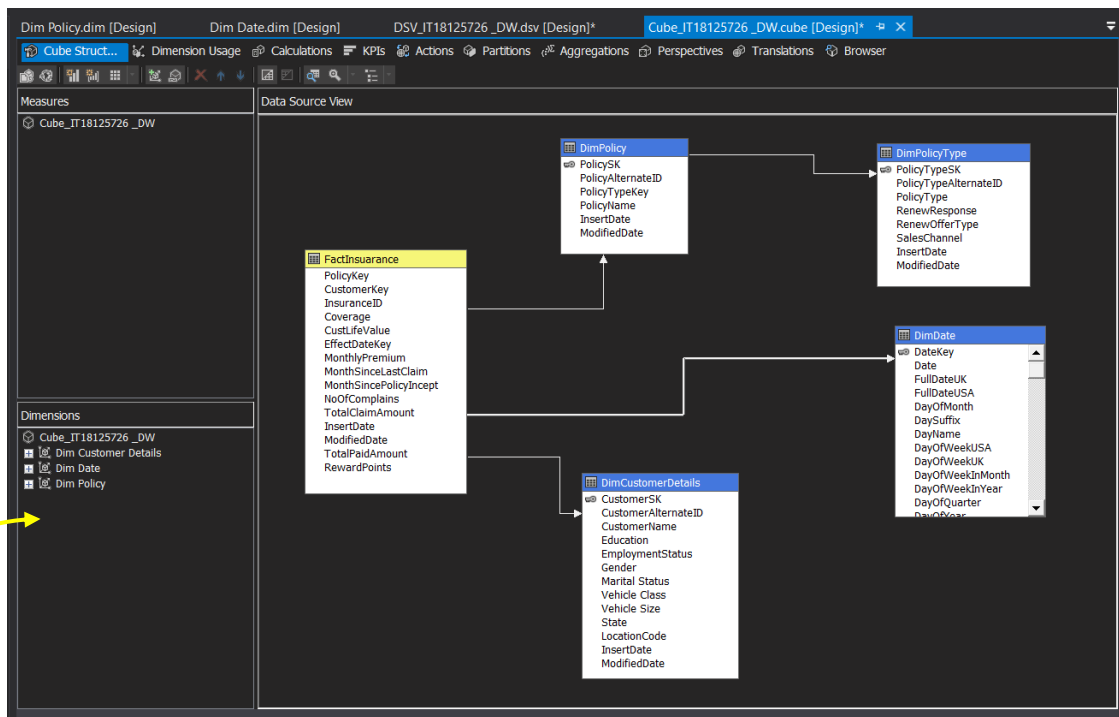
After that, I clicked on the “Add Related Tables” to automatically select and add the related dimension tables. It should automatically add FactInsurance, DimCustomerDetails, DimPolicy, DimDate dimensions to the list. However, it did not automatically add the DimpolicyType. So, I manually selected them and added to the list.

Once completed creating data source view, the newly created data source view as shown in the below screenshot.



- Creating a Data Cube

Once cube wizard configurations completed, it created a data cube as shown in the screenshot



Then I have configured each dimension are shown in the table view.

First, I configured the Dim Policy dimension

The policy, Policy Type all have keys as the displayed values in this cube.

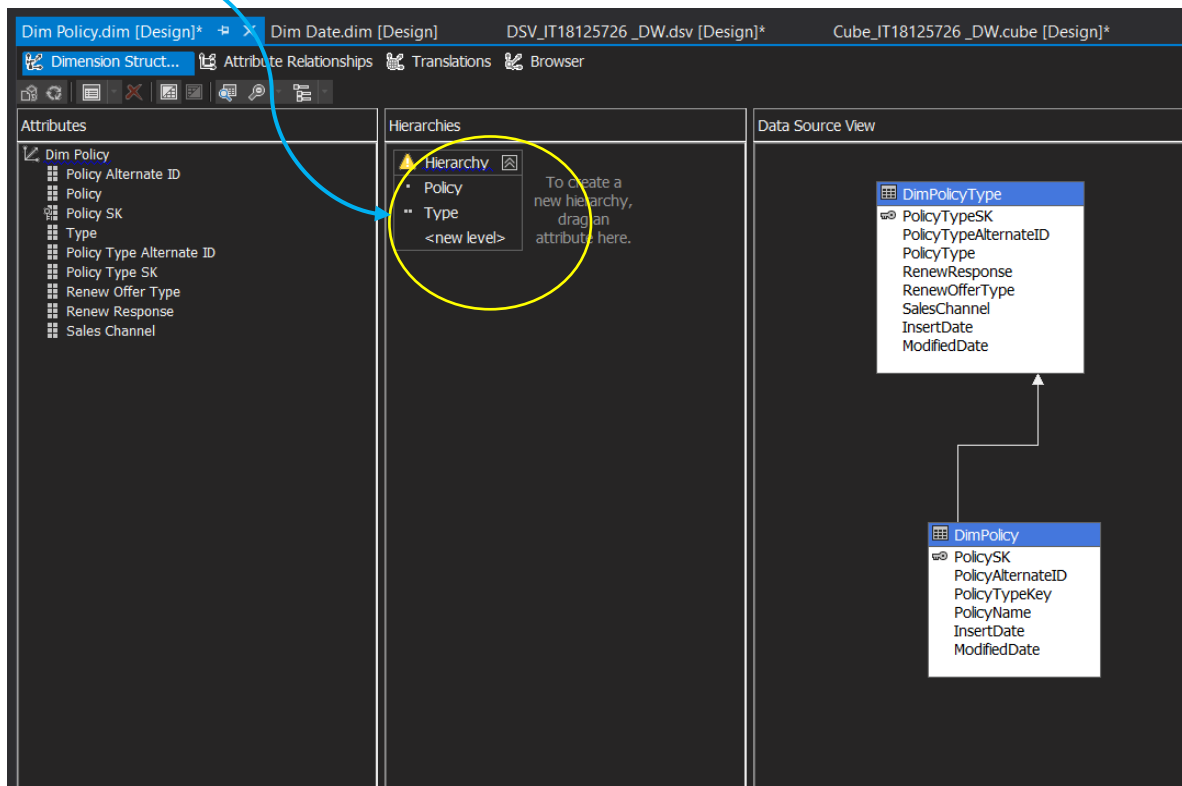
Changing the Name Column property will enable a different column value to be used. The end user does not want to see the primary key for policy and, they would want to see the policy Name.

So, I have changed the name NameColumn to Type.

As shown in the below I have created a policy in the cube.

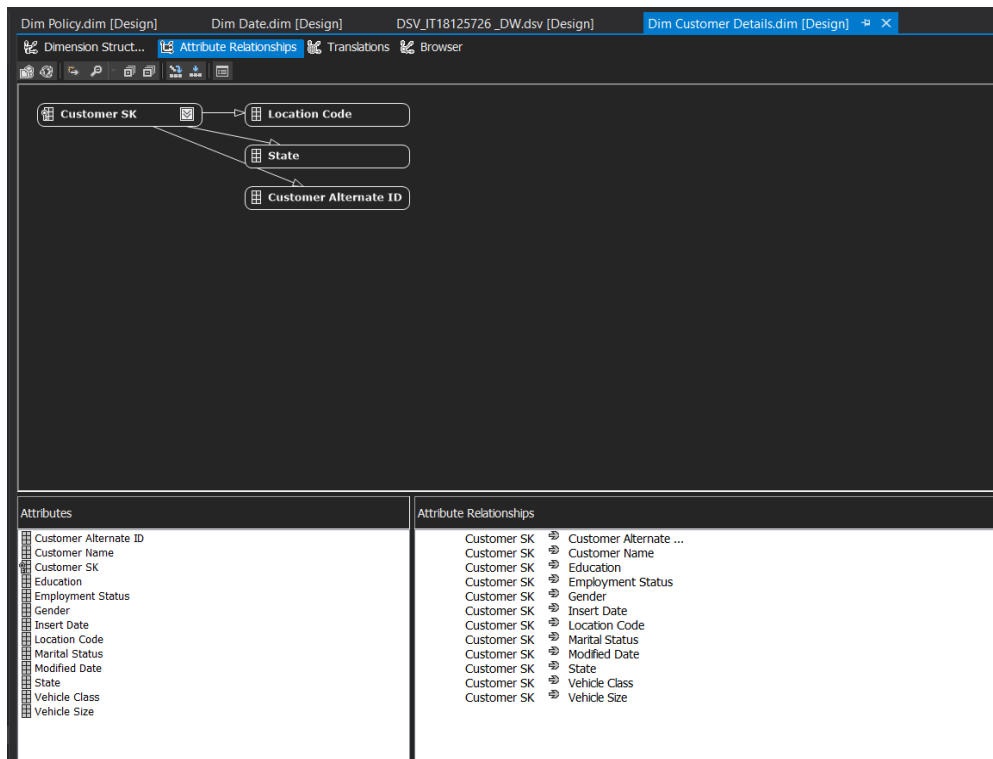
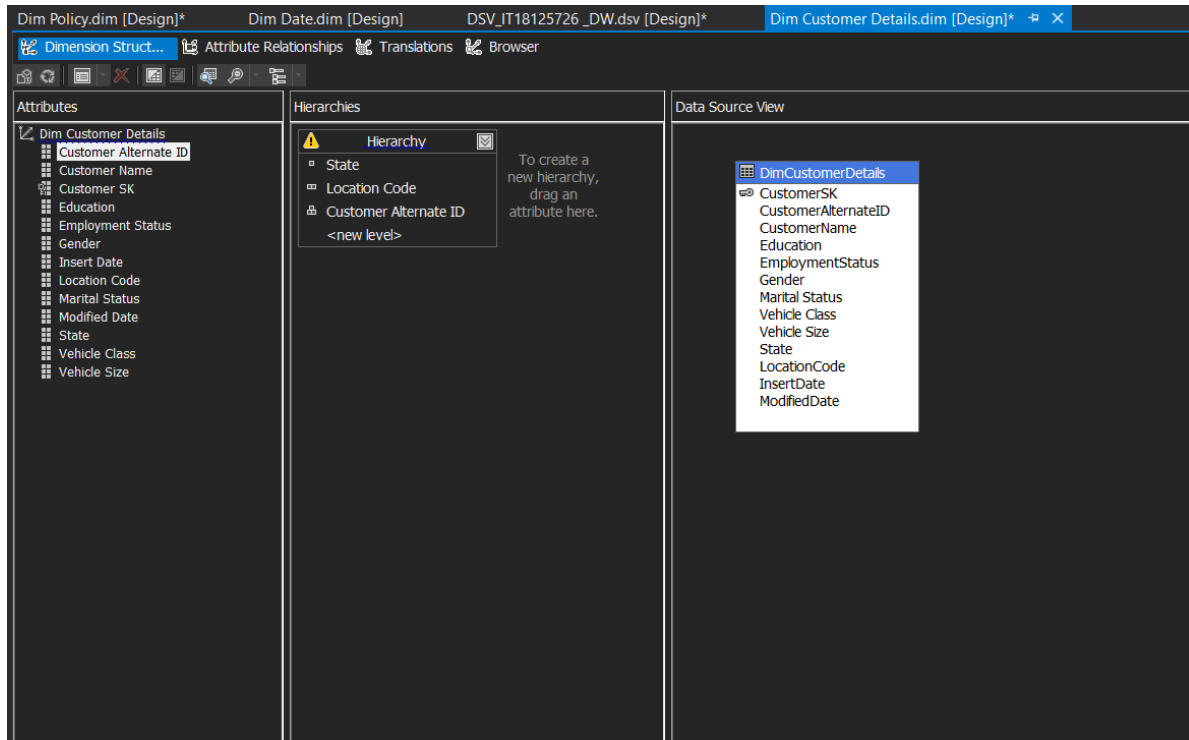


This indicator is only a warning and will not prevent the cube from being deployed. It is only a saying no relationship can cause performance issues when using large dimensions with this hierarchy type.



(Policy Hierarchy)

Hierarchies are useful in visual reporting tools to show the parent/child relationship between attributes. So, I have Created another hierarchy called Location hierarchy.



(Location Hierarchy)

## Creating a KPI's

Then I have created 4 KPI's based on my business requirement.

The screenshot shows the 'KPI Organizer' on the left with 'KPI Total Paid Amount' selected. The main configuration area on the right has the following fields:

- Name: KPI Total Paid Amount
- Associated measure group: Fact Insurance
- Value Expression: [Measures].[Total Paid Amount]
- Goal Expression: [Measures].[Total Paid Amount] > 300

**KPI Total  
Paid Amount**

The screenshot shows the 'KPI Organizer' on the left with 'KPI Customer Life Time Value' selected. The main configuration area on the right has the following fields:

- Name: KPI Customer Life Time Value
- Associated measure group: Fact Insurance
- Value Expression: [Measures].[Cust Life Value]
- Goal Expression: [Measures].[Cust Life Value] > 7000

**KPI  
Customer  
Lifetime  
Value**

The screenshot shows the 'KPI Organizer' on the left with 'KPI Rewards Points' selected. The main configuration area on the right has the following fields:

- Name: KPI Rewards Points
- Associated measure group: <All>
- Value Expression: [Measures].[Reward Points]
- Goal Expression: [Measures].[Reward Points] > 50

**KPI Rewards  
Points**

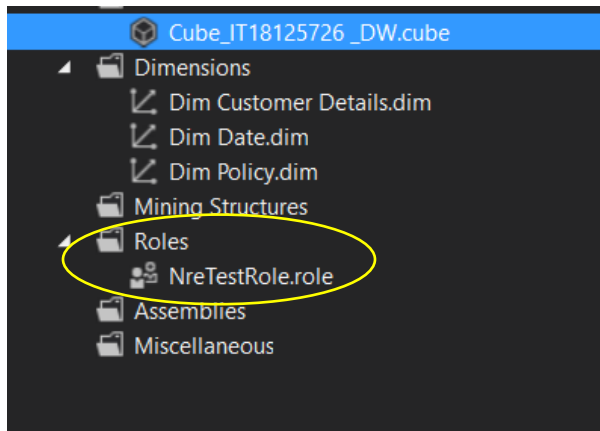
The screenshot shows the 'KPI Organizer' on the left with 'KPI Total Claim amount' selected. The main configuration area on the right has the following fields:

- Name: KPI Total Claim amount
- Associated measure group: Fact Insurance
- Value Expression: [Measures].[Total Claim Amount]
- Goal Expression: [Measures].[Total Claim Amount] > 300

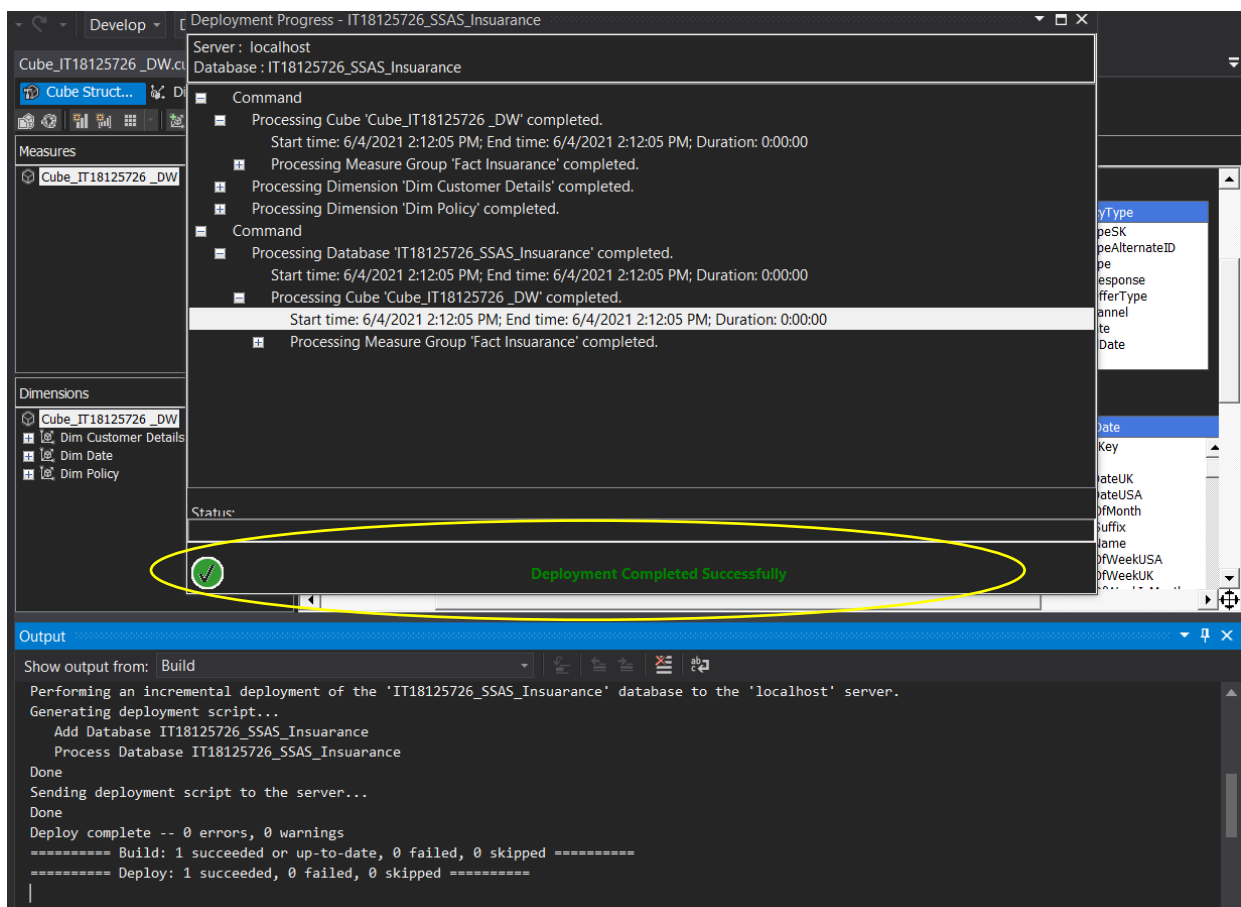
**KPI Total  
Claim  
Amount**



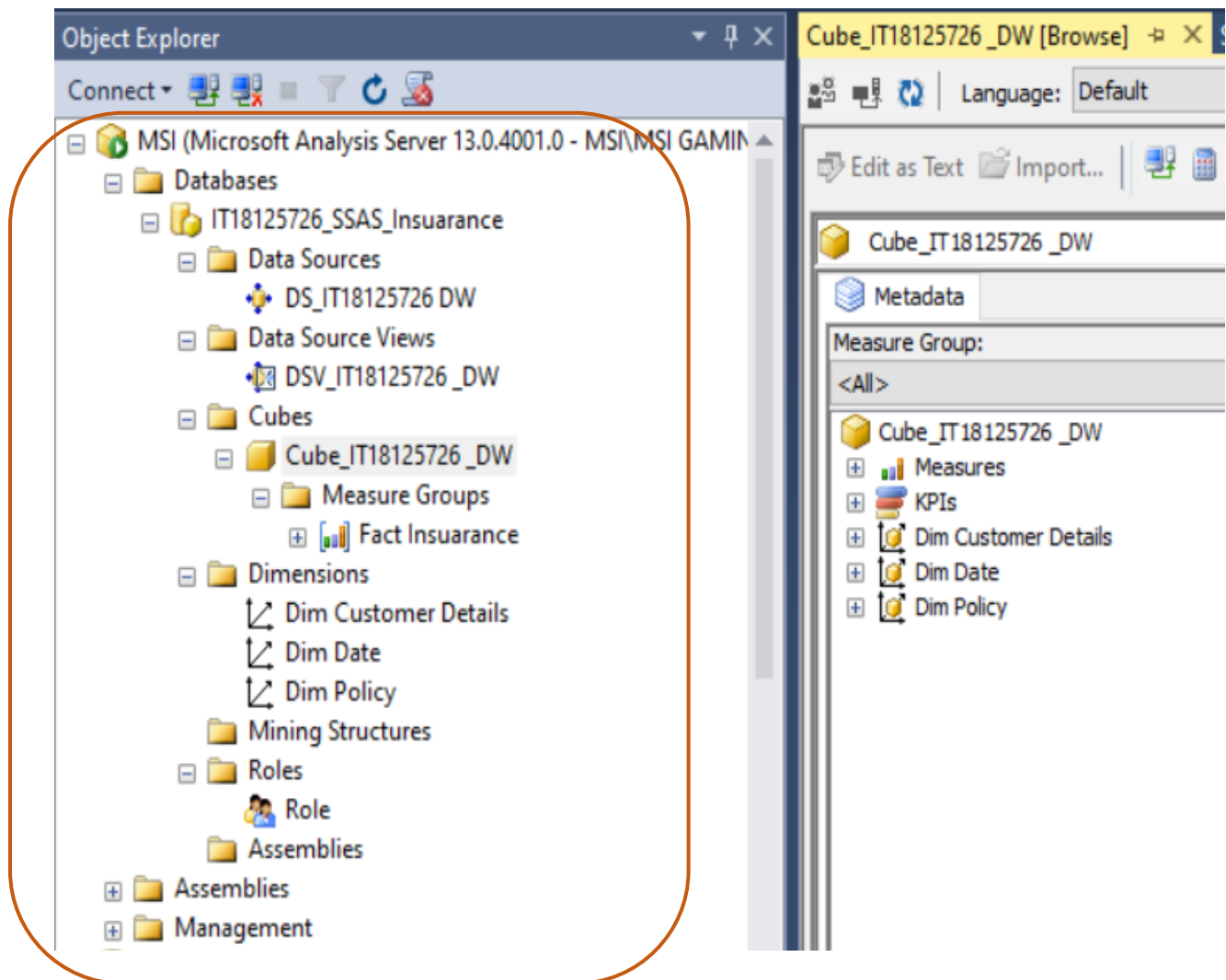
Then I have created one **user Role** and provide permissions to access to the data cube



Finally, I have Deployed the project, I got the deployment is successful message as shown below.



## Browsing the data cube



### Step 3: Demonstration of OLAP operations

Create an Excel report using MDX query

Using the power pivot, Power Query, Power view in excel allow us to create a semantic layer inside excel.

To connect the excel workbook and to get the data to the semantic layer we use MDX query.

I dragged and dropped state from location hierarchy in customer details dimension, policy type name from policy hierarchy in policy type dimension, Total claim Amount from fact Insurance under measures, Customer Lifetime Value **Value** KPIs and Customer Lifetime Value **Goal** KPIs. Additionally, I had added 'State' from 'Dim Customer details as a filter into the section above the data grid as shown below

After that the data grid as displayed as below.

Filter

Dimension	Hierarchy	Operator	Filter Expression
Dim Customer Details	State	Equal	
<Select dimension>			

State	PolicyType	Total Claim Amount	KPI Customer Life Time Value ...	KPI Customer Life Time Value ...
Ariz...	Corporat...	981789.939102	17232079.440961	True
Ariz...	Personal ...	3844885.322908	71962783.5342849	True
Ariz...	Special Auto	244058.539998	4520188.920009	True
Ariz...	Unknown	(null)	(null)	False
Calif...	Corporat...	2221007.862164	41061178.653145	True
Calif...	Personal ...	7086200.551647	128741950.813077	True
Calif...	Special Auto	346706.44141	6677303.605651	True
Calif...	Unknown	(null)	(null)	False
Nev...	Corporat...	608234.332972	11148452.660779	True
Nev...	Personal ...	1995455.854616	36500592.517032	True
Nev...	Special Auto	105155.809773	2093062.848849	True
Nev...	Unknown	(null)	(null)	False
Ore...	Corporat...	1640878.190672	29489438.931239	True
Ore...	Personal ...	5852431.751691	110062178.803948	True
Ore...	Special Auto	390549.408305	7522729.245269	True
Ore...	Unknown	(null)	(null)	False
Was...	Corporat...	476070.381584	8720156.306309	True
Was...	Personal ...	1882324.195115	34160700.057561	True
Was...	Special Auto	79020.748821	1927087.75161	True
Was...	Unknown	(null)	(null)	False

Then I clicked on the Design Mode button to view the MDX query of the configuration

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the cube structure for 'Cube\_IT18125726\_DW', including measures like 'KPI Total Claim amount' and dimensions like 'Dim Customer Details'. The right pane shows the MDX query in Design Mode. The query is highlighted with an orange oval, and an arrow points from the text below to it.

```
SELECT NON EMPTY { [Measures].[Total Claim Amount], KPIValue("KPI Customer Life Time Value"), KPIGoal("KPI Customer Life Time Value") } ON COLUMNS, NON  
EMPTY { ([Dim Customer Details].[State].[State].ALLMEMBERS * [Dim Policy].[PolicyType].[PolicyType].ALLMEMBERS) } DIMENSION PROPERTIES MEMBER_CAPTION,  
MEMBER_UNIQUE_NAME ON ROWS FROM [Cube_IT18125726_DW] CELL PROPERTIES VALUE, BACK_COLOR, FORE_COLOR, FORMATTED_VALUE, FORMAT_STRING,  
FONT_NAME, FONT_SIZE, FONT_FLAGS
```

State	PolicyType	Total Claim Amount	Cust Life Value	KPI Customer Life Time Value ...
Ariz...	Corporat...	981789.939102	17232079.44...	True
Ariz...	Personal ...	3844885.322908	71962783.53...	True
Ariz...	Special Auto	244058.539998	4520188.920...	True
Ariz...	Unknown	(null)	(null)	False

Then I have used this query in an Excel sheet to generate a report through Excel.

## Connecting Excel to SSAS Cube using a MDX Query

After pasting the generated query, clicked on validate to make sure the query does not contain any errors.

Table Import Wizard

**Specify a MDX Query**  
Type or paste a MDX query to select data to import from the source database.

Friendly Query Name:

MDX Statement:

```
SELECT NON EMPTY { [Measures].[Total Claim Amount], KPIValue("KPI Customer Life Time Value"), KPIGoal("KPI Customer Life Time Value") } ON COLUMNS, NON EMPTY { ([Dim Customer Details].[State].[State].ALLMEMBERS * [Dim Policy].[PolicyType].[PolicyType].ALLMEMBERS ) } DIMENSION PROPERTIES MEMBER_CAPTION, MEMBER_UNIQUE_NAME ON ROWS FROM [Cube_IT18125726_DW] CELL PROPERTIES VALUE, BACK_COLOR, FORE_COLOR, FORMATTED_VALUE, FORMAT_STRING, FONT_NAME, FONT_SIZE, FONT_FLAGS
```


☐ Import measures as text

The MDX statement is valid.

Then I have clicked on finish to extract the data the cube.


Table Import Wizard

**Importing**  
The import operation might take several minutes to complete. To stop the import operation, click the Stop Import button.


**Success**

Total: 1   Cancelled: 0  
Success: 1   Error: 0

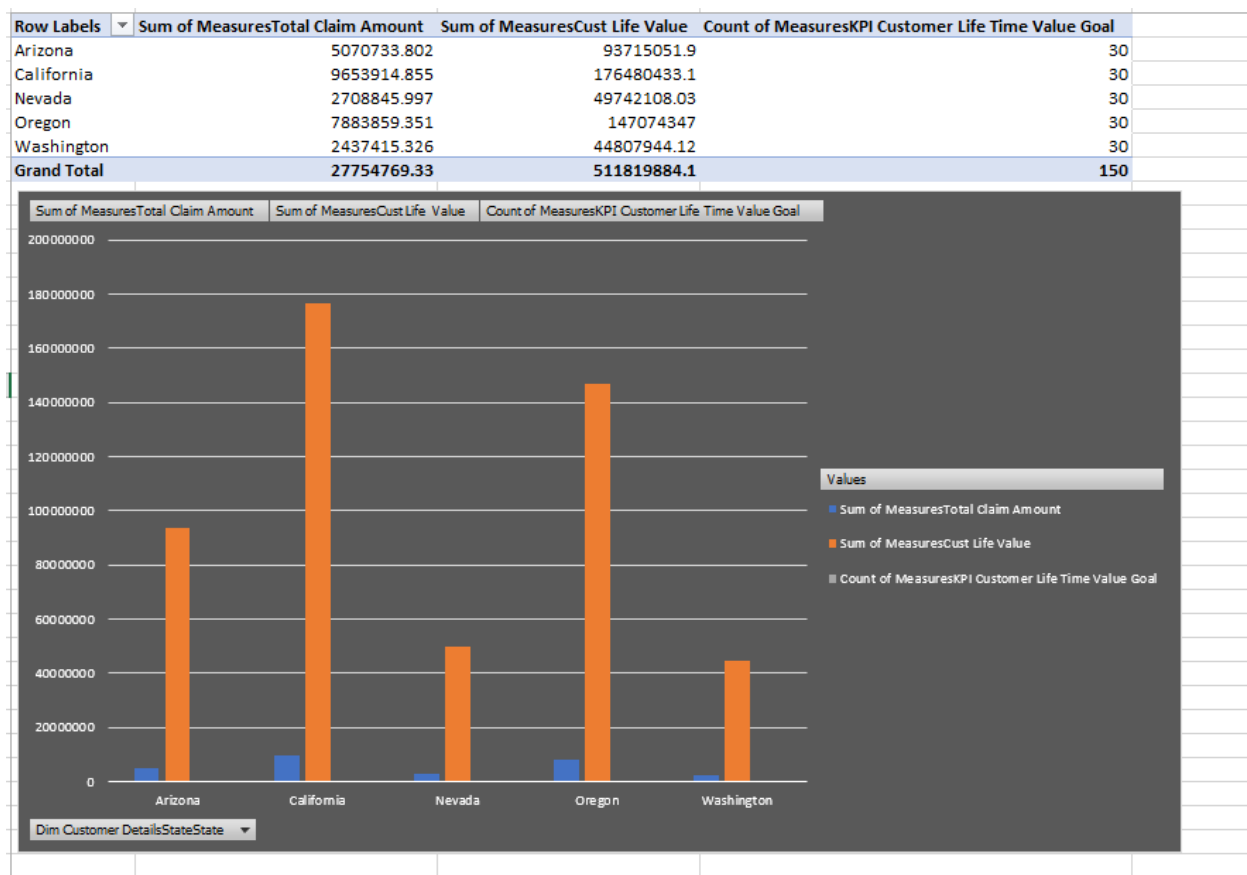
Details:

	Work Item	Status	Message
	Query	Success. 20 rows transferred.	

## Excel report 01

- **Pivot**

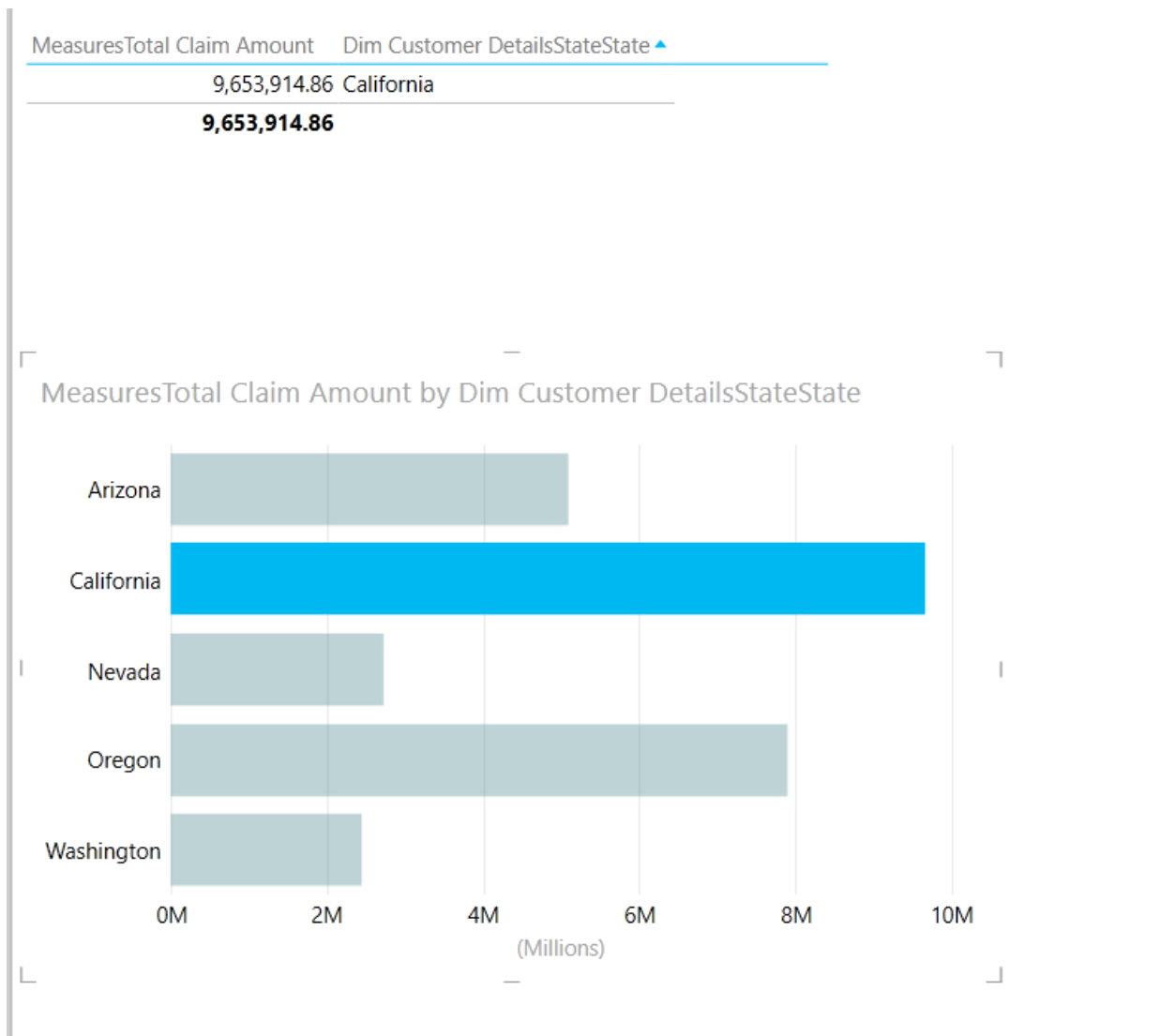
In the below pivot table, I have statically summarized the data of a more extensive insurance table. This summary includes sum of total claim Amount, sum of customer lifetime value and count of customer lifetime value, which the pivot table groups together in a descriptive manner in a state wise. And using this pivot table we can visualize our data by giving them a different perspective and view. We can rotate the axis of the dimension and see different pattern of the same data.



- **Slice**

Slice is a rectangular subset of a cube, by choosing a single value for one of its dimensions.

So here I have used a slicer to filter data in table and graph by State wise. So, this blue colored highlighted area displays the Total claim amount in California state. Likewise, we can view total claim amount of each state (Arizona, California, Nevada, Oregon, Washington)

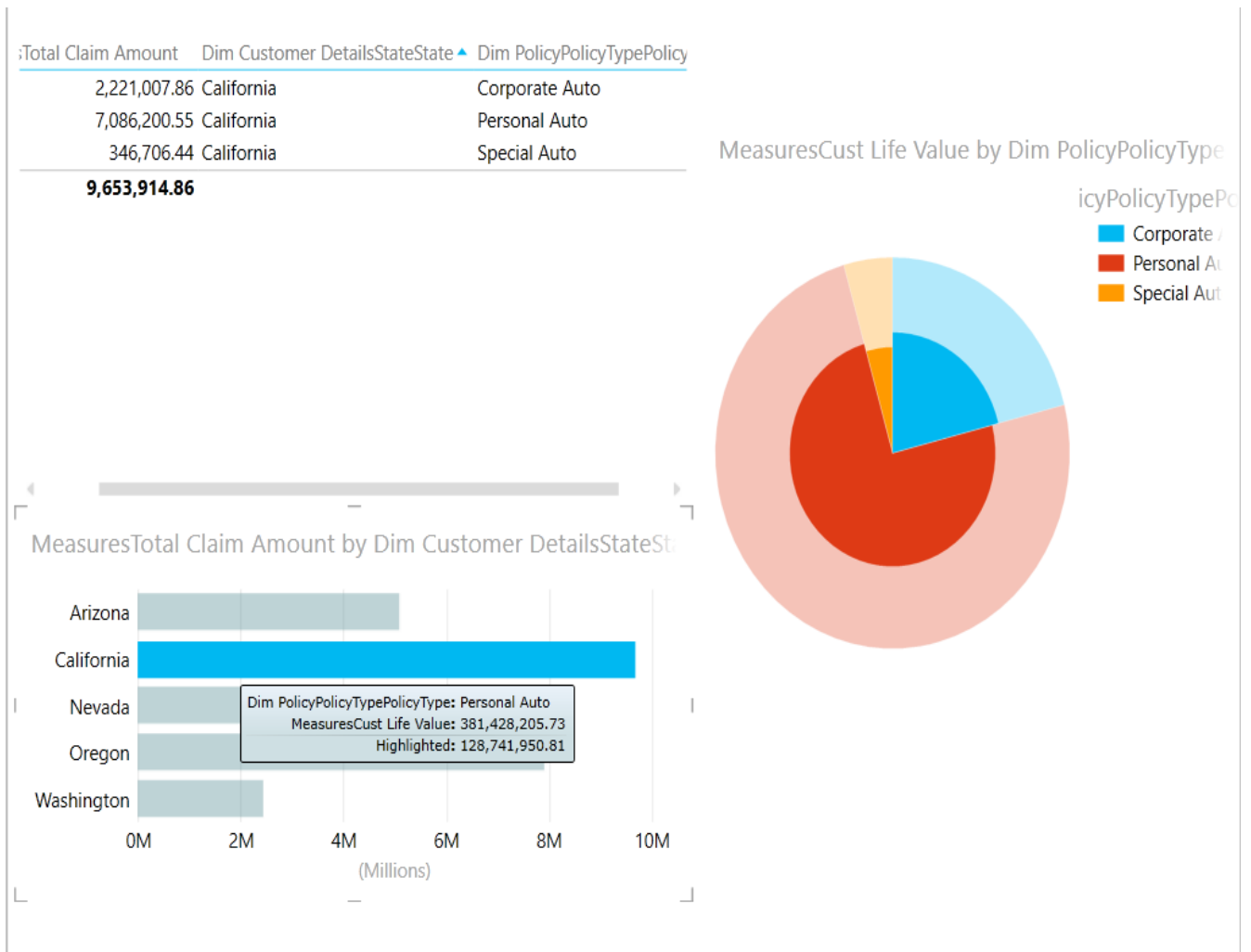


- **Dice**

Selects two or more dimensions from a given cube and provides new sub-cube by selecting specific values on those selected dimensions.

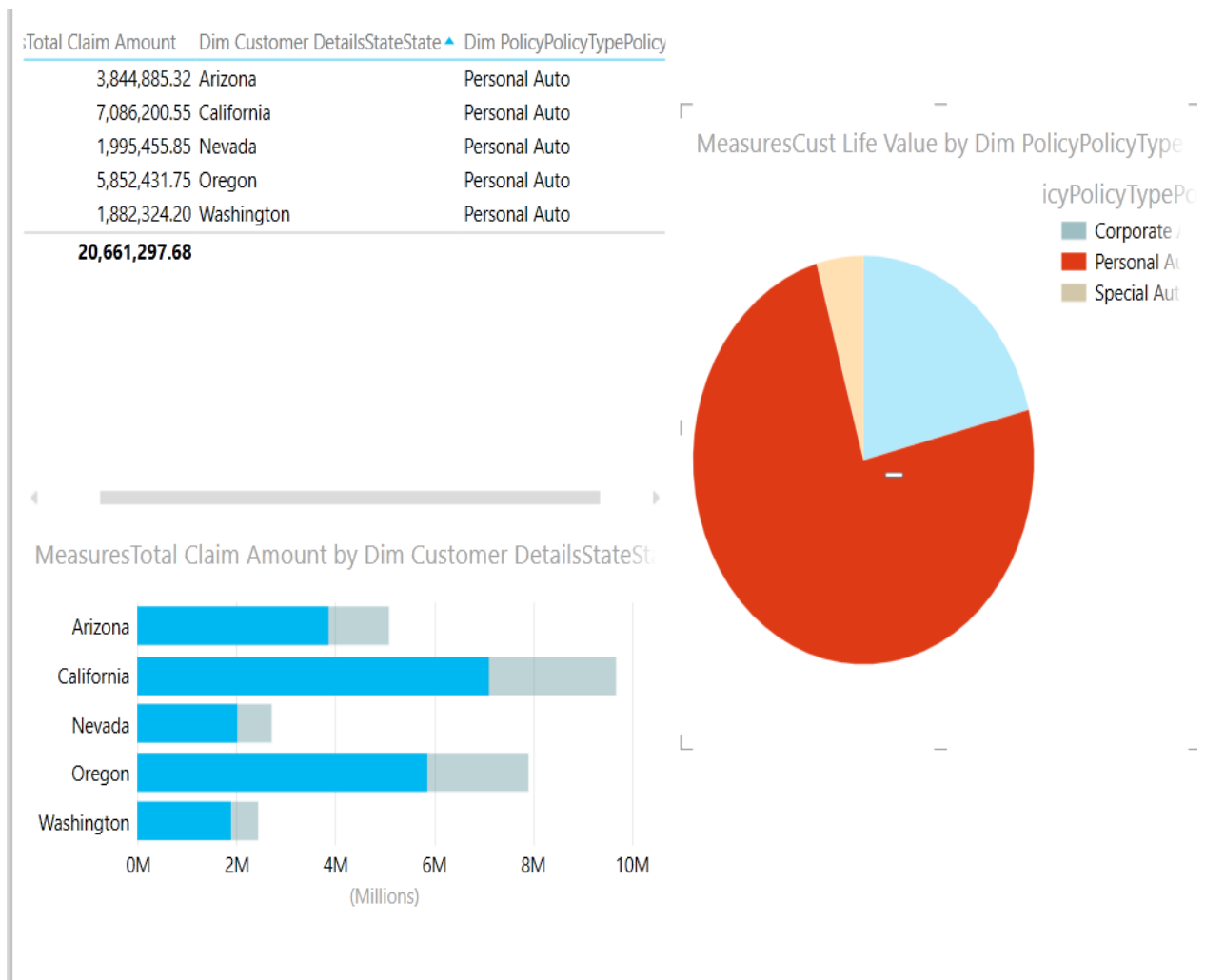
Therefore, this report shows total claim amount in Arizona, California, Nevada, Oregon, Washington by Cooperate auto, Personal auto and Special auto policy type

As an instance in below figure, pie chart's highlighted area emphasizes all values of 3 different types of policies belongs to state of California.





And, below figure pie chart's highlighted area illustrate the personal auto policy type values which are belong to all states (Arizona, California, Nevada, Oregon, Washington) include in dataset.



- Personal Auto, Corporate Auto and Special Auto are policy type of policy.
- Arizona, California, Nevada, Oregon, and Washington are state of customer details

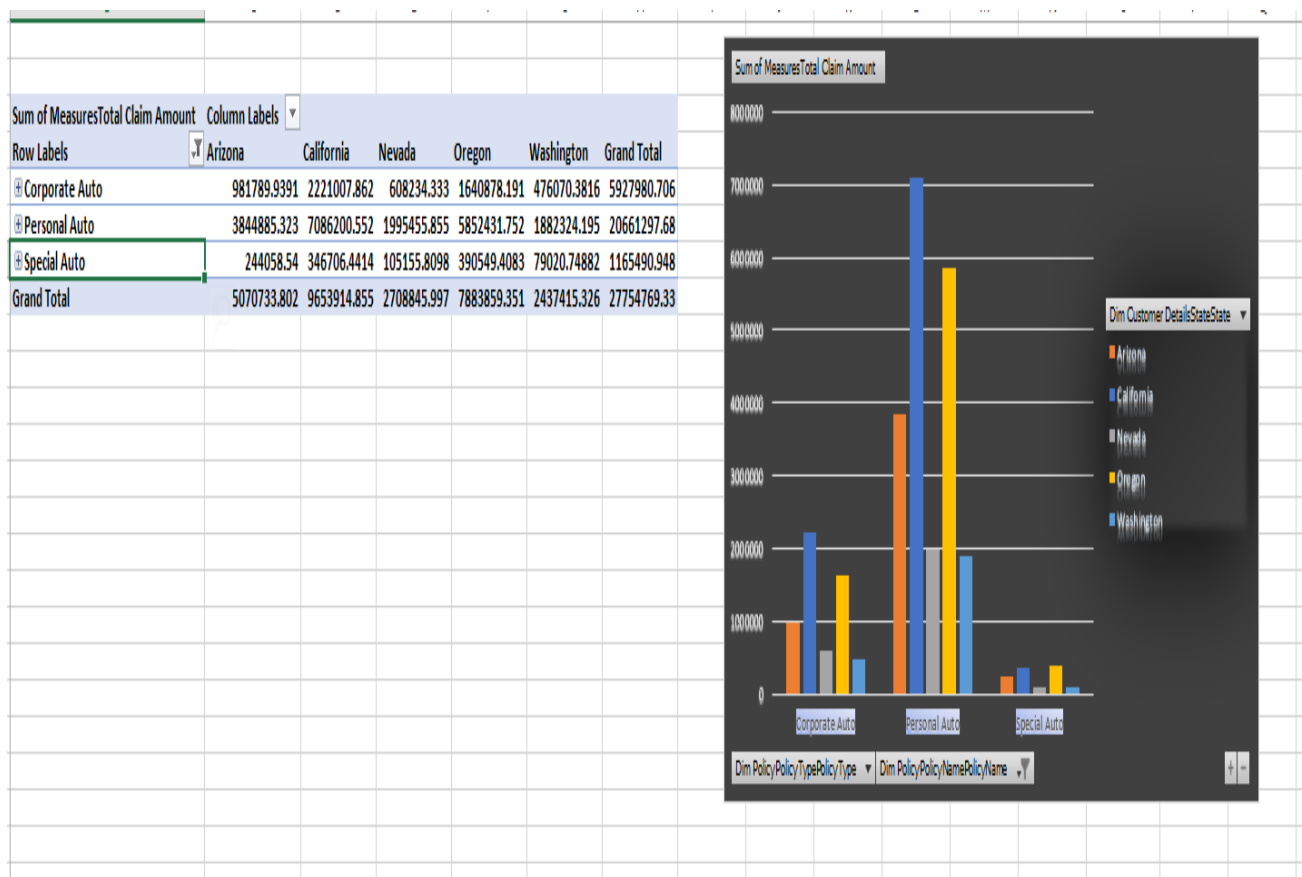
- **Roll-up and drill-down**

In this excel sheet it shows the sum of measures total amount. In the columns I have included a state attribute in customer details value and for the rows I have used policy hierarchy (PolicyType → Policy). So, in this case report can be view by the state wise location wise

In this roll up and drill down report rows can drill down policy type to policy. So that report can view the policy claim amount .and the report can roll up from policy-to-policy type so we can view policy type total claim amount.

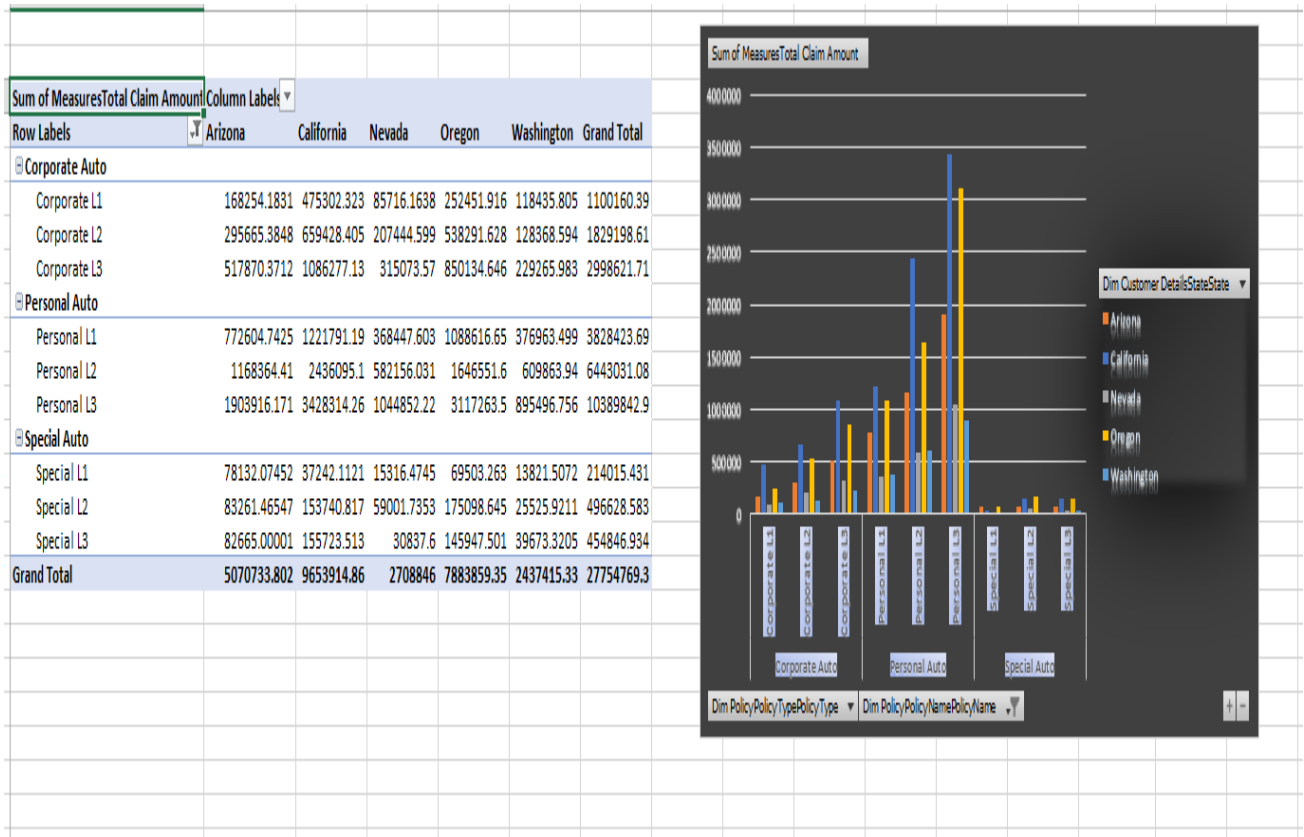
- **Roll up**

Climbing up a hierarchy of the dimension to aggregate data means the roll up OLAP operation in cubes



- **Drill down**

Stepping down a hierarchy of the dimension allowing navigation through details means the drill down OLAP operation in cube

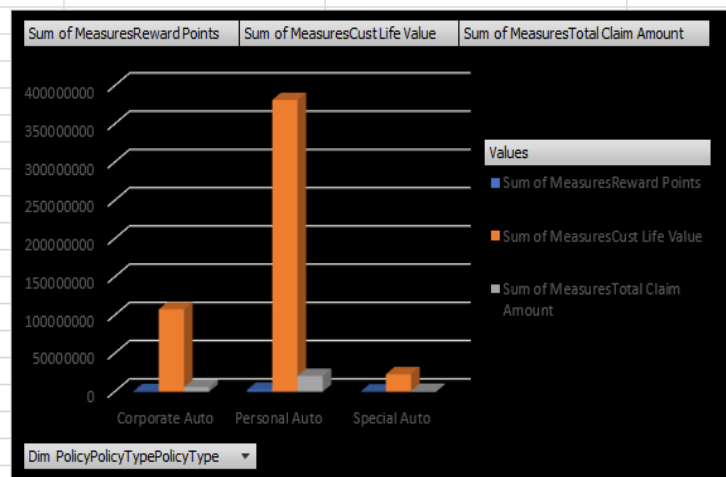


## Excel Report 02

- **Pivot**

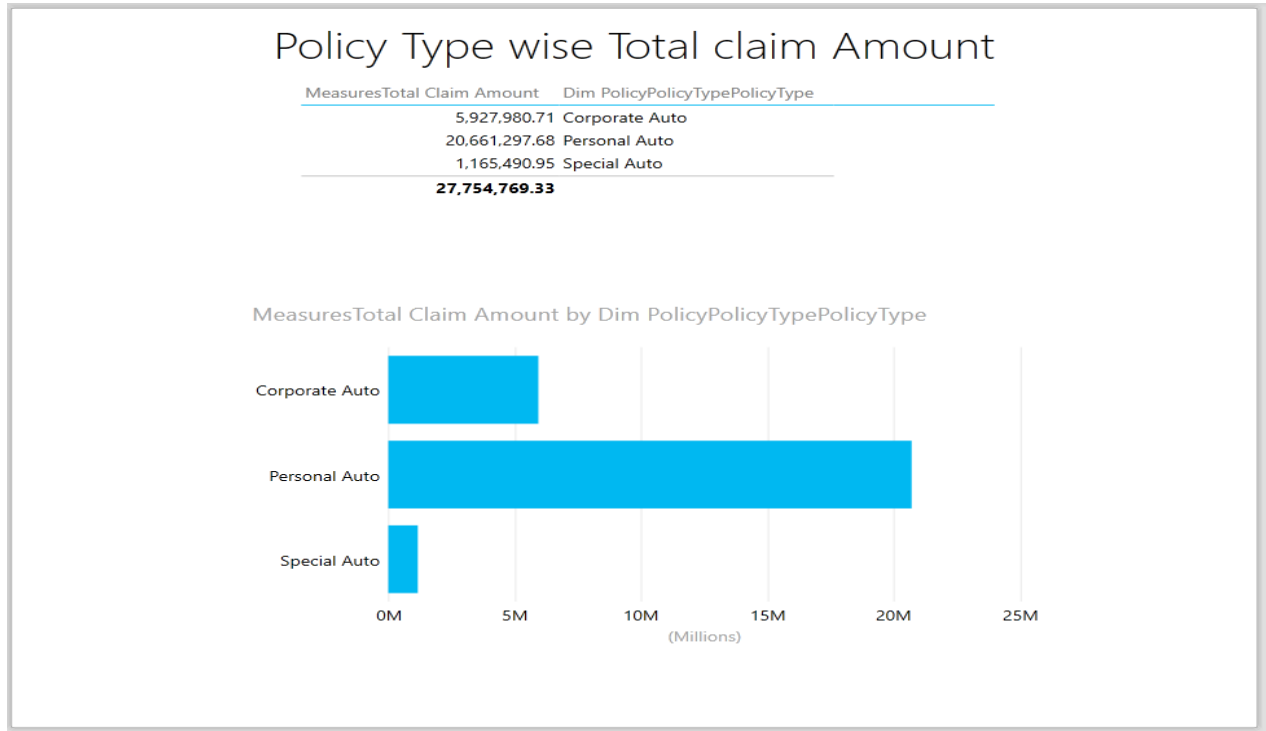
In the below pivot table, I have statistically summarized the data to the more extensive Insurance table. This summary includes sum of Measures Reward points, sum of measures customer lifetime value and sum of measures total claim amount which the pivot table groups together in a meaningful way in Policy Type wise.

A	B	C	D	E	F	G
	Row Labels	Sum of MeasuresReward Points	Sum of MeasuresCust Life Value	Sum of MeasuresTotal Claim Amount		
	Corporate Auto	645907.836	107651306	5927980.706		
	Personal Auto	2288569.234	381428205.7	20661297.68		
	Special Auto	136442.2342	22740372.37	1165490.948		
	Grand Total	3070919.305	511819884.1	27754769.33		



- **Slice**

So here I have used a slicer to filter data in table and graph by Policy Type wise. So, this blue colored highlighted area displays the Total claim amount in All three policy types. Likewise, we can view total claim amount of each Policy type (corporate Auto, Personal Auto, Special Auto)

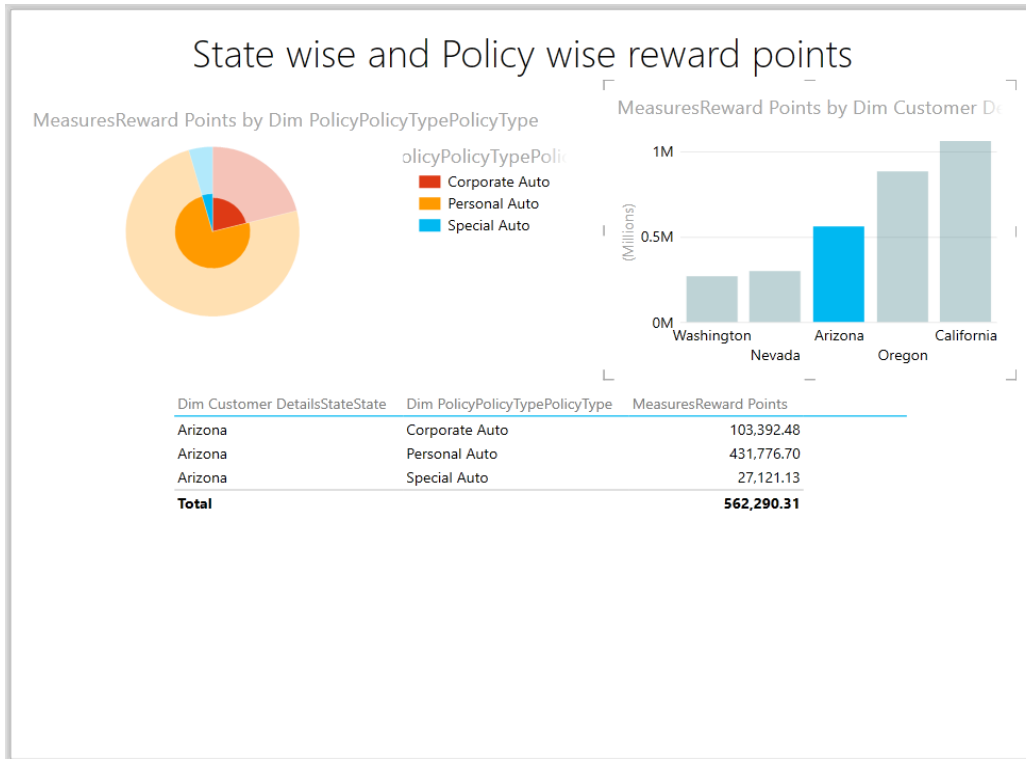


- **Dice**

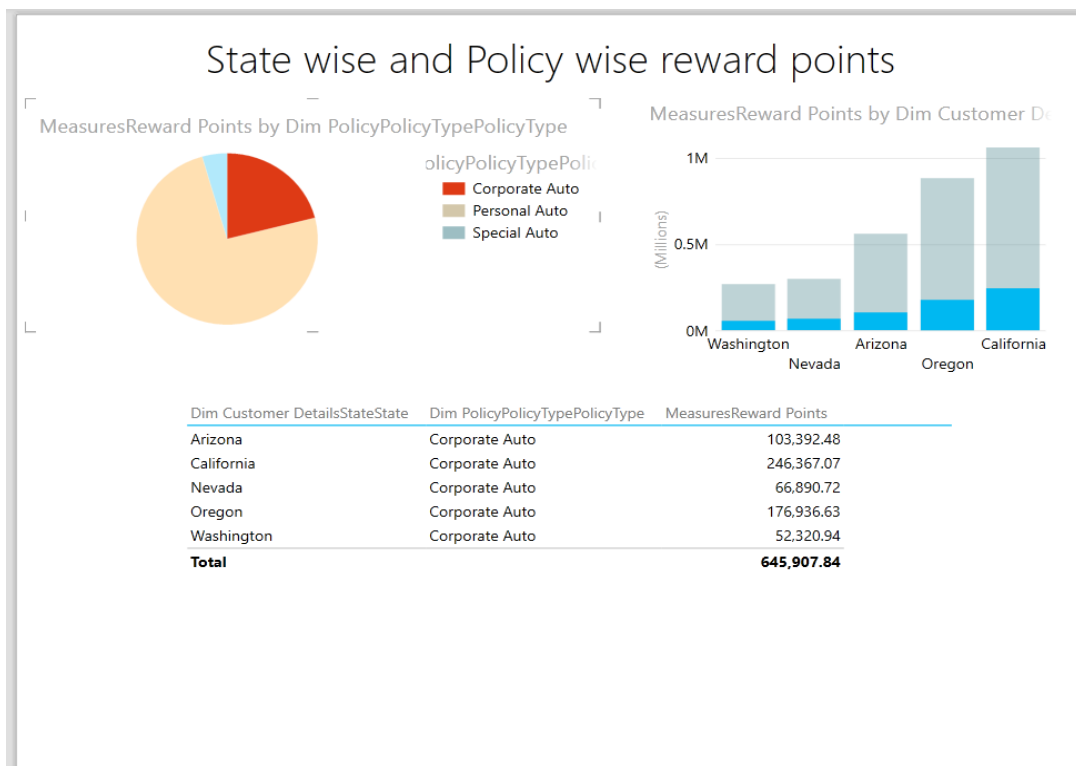
So, this report shows Rewards Points in Arizona, California, Nevada, Oregon, Washington by Cooperate auto, Personal auto and Special auto policy type

As an instance in below **figure 01**, pie chart's highlighted area emphasizes all values belongs to the reward points of 3 different types of policies belongs to state of Arizona.

And below **figure 02** pie chart's highlighted area illustrate the corporate auto policy type values which are belong to all states (Arizona, California, Nevada, Oregon, Washington) include in dataset.



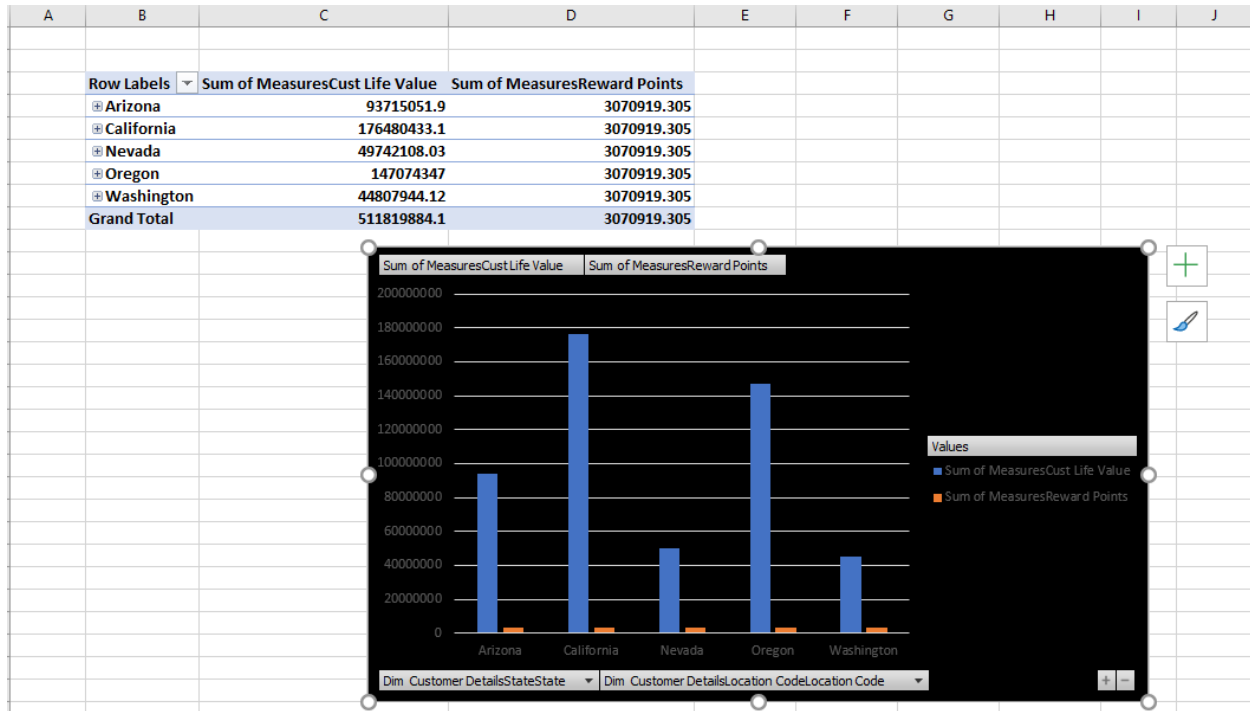
**Figure 01**



**Figure 02**

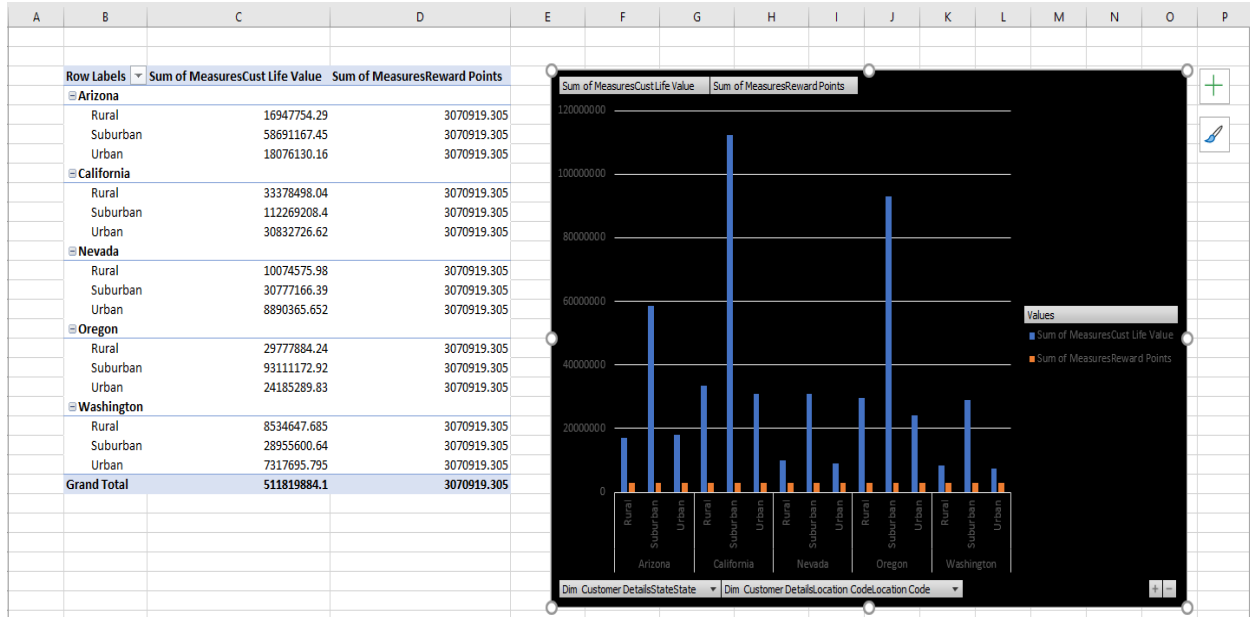
- **Roll up**

Aggregate the values of the customer lifetime value and the reward points values from location wise to state wise. From this report User can go through more specific view of data to general view of data



- **Drill Down**

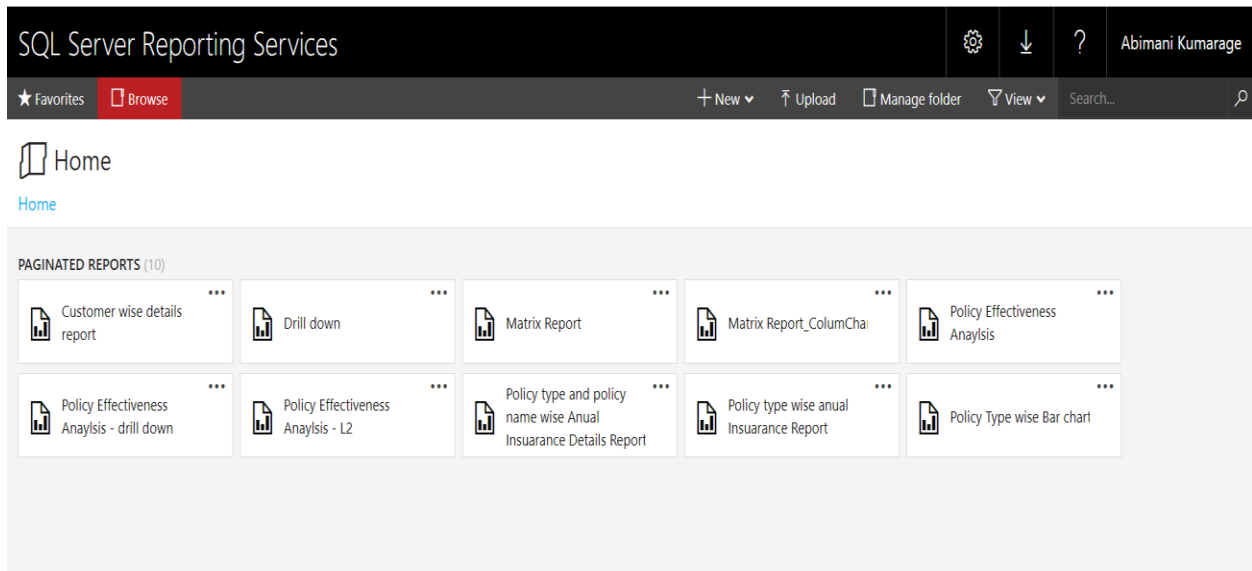
Split the values of the Customer lifetime value and reward point value from state wise to location code wise. Using this report user can go through general view of data to more specific view of data





## Step 4: SSRS Reports

SQL Server Reporting Services (SSRS) is a platform for creating, publishing, and managing mobile and paginated reports/dashboards, then delivering them to the right users in different ways, such as via a web browser, on their mobile device, or via email. Here I have mainly used **Report Builder** to create SSRS Reports because as a standalone application it will provide more freedom for users than using Report Design (SSDT) in visual studio.



## **Report 1: Report with a matrix**

In order retrieve dataset to the report builder, I have created data source and dataset in the report builder respectively and add below query to retrieve specific data (Before executing this query I have executed this in the sql server management studio to prevent the errors occur during the direct execution.

```
select
dp.PolicyName,dpt.PolicyType,dcd.CustomerAlternateID,dcd.Gender,dcd.CustomerName,dcd.State,
dcd.LocationCode,dd.Month,dd.MonthName,dd.Year,fi.CustLifeValue,fi.MonthlyPremium,fi.TotalClaim
Amount,
fi.TotalPaidAmount,fi.RewardPoints,fi.NoOfComplains,fi.MonthSinceLastClaim,fi.MonthSincePolicyInc
ept
from [dbo].[FactInsurance] fi
inner join [dbo].[DimPolicy] dp on fi.[PolicyKey] = dp.[PolicySK]
inner join [dbo].[DimPolicyType] dpt
on dp.[PolicyTypekey] = dpt.PolicyTypeSK
inner join [dbo].[DimCustomerDetails] dcd
on fi.[CustomerKey] = dcd.[CustomerSK]
inner join [dbo].[DimDate] dd
on fi.[EffectDateKey] = dd.[DateKey]
```

Dataset Properties

Query

Fields

Options

Filters

Parameters

Choose a data source and create a query.

Name:

DataSet\_Insurance

☐ Use a shared dataset.

☒ Use a dataset embedded in my report.

Data source:

DS\_IT18125726\_DW New...

Query type:

☒ Text ☐ Table ☐ Stored Procedure

Query:

```
dp.PolicyName,dpt.PolicyType,dcd.CustomerAlternateID,dcd.Gender,dcd.CustomerName,dcd.State,
dcd.LocationCode,dd.Month,dd.MonthName,dd.Year,fi.CustLifeValue,fi.MonthlyPremium,fi.TotalClaim
Amount,
fi.TotalPaidAmount,fi.RewardPoints,fi.NoOfComplains,fi.MonthSinceLastClaim,fi.MonthSincePolicyInc
ept
from [dbo].[FactInsurance] fi
inner join [dbo].[DimPolicy] dp on fi.[PolicyKey] = dp.[PolicySK]
inner join [dbo].[DimPolicyType] dpt
on dp.[PolicyTypekey] = dpt.PolicyTypeSK
inner join [dbo].[DimCustomerDetails] dcd
on fi.[CustomerKey] = dcd.[CustomerSK]
inner join [dbo].[DimDate] dd
on fi.[EffectDateKey] = dd.[DateKey]
where dp.PolicyName = @Policy_Name
```

Query Designer... Import... Refresh Fields

Help OK Cancel

## Design view

Policy Type Wise Annual Insurance Matrix Report					
		[Year]		Total	
Policy Type	Policy Name	Monthly Pre	Total Claim	Monthly Premium	Total Claim Amount
[PolicyType]	[PolicyName]	[Sum(MonthlyP	[Sum(TotalClai	[Sum(MonthlyPremiu	[Sum(TotalClaimAmount)]
	Total	[Sum(MonthlyP	[Sum(TotalClai	[Sum(MonthlyPremium	[Sum(TotalClaimAmount)]
Total		[Sum(Monthly	[Sum(TotalCla	[Sum(MonthlyPremium	[Sum(TotalClaimAmount)]

## Preview

SQL Server Reporting Services

Abimani Kumara

★ Favorites Browse

Home > Matrix Report

1 of 1 100% Find | Next

Policy Type Wise Annual Insurance Matrix Report

Policy Type	Policy Name	2011		2012		Total	
		Monthly Premium	Total Claim Amount	Monthly Premium	Total Claim Amount	Monthly Premium	Total Claim Amount
Corporate Auto	Total	687890	3165811.37784	596561	2762169.32865	1284451	5927980.70649399
Personal Auto	Total	2282777	10587613.018101	2147180	10073684.657876	4429957	20661297.6759774
Special Auto	Total	124208	580264.766248999	121639	585226.182058	245847	1165490.94830699
Total		3094875	14333689.162191	2865380	13421080.1685871	5960255	27754769.3307789

## Final Report View

SQL Server Reporting Services

★ Favorites Browse

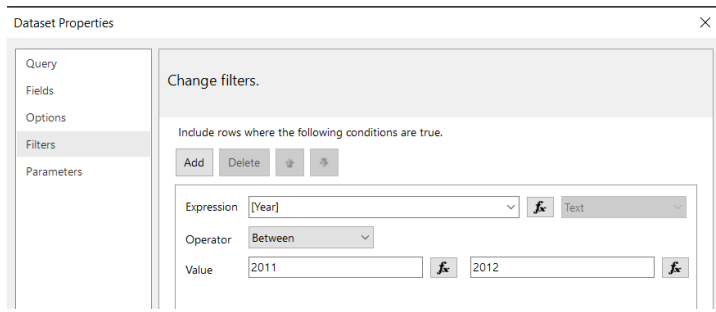
Home > Matrix Report

1 of 1 100% Find | Next

Policy Type Wise Annual Insurance Matrix Report

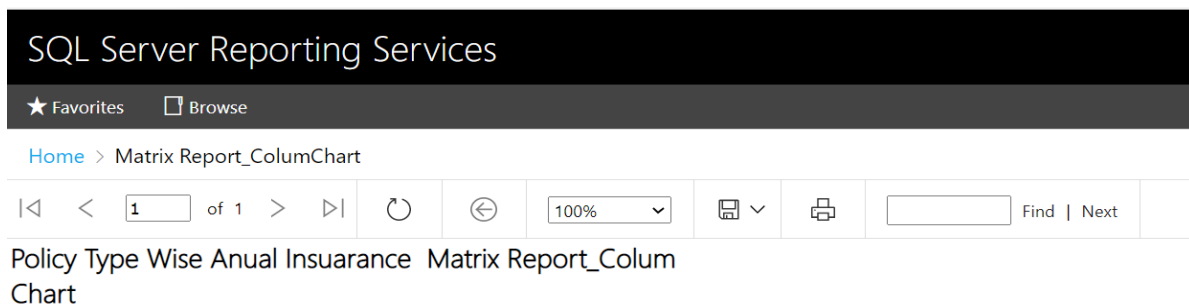
		2011		2012		Total	
Policy Type	Policy Name	Monthly Premium	Total Claim Amount	Monthly Premium	Total Claim Amount	Monthly Premium	Total Claim Amount
Corporate Auto	Corporate L1	128443	592133.667419	107184	308026.722861	235627	1100160.39028
	Corporate L2	207388	947826.588811	187439	881372.023959	394807	1829198.61077101
	Corporate L3	352079	1625851.12361001	301938	1372770.581833	654017	2998621.70544299
	Total	687890	3165811.37784	596561	2762169.328653	1284451	5927980.70649399
			1		3		
Personal Auto	Personal L1	417592	1967253.67214601	391986	1881170.01956	809578	3828423.69171499
	Personal L2	709240	3328557.45393	667919	3116473.624103	1377159	6443031.07804297
	Personal L3	1155945	5293801.89201499	1087275	5096041.01420396	2243320	10388842.9062191
	Total	2382777	10587613.018101	2147180	10073684.657876	4429957	20661297.6759774
			999		76		
Special Auto	Special L1	23898	118860.973159	18256	97154.4582119	42154	214015.431371
	Special L2	51450	226354.092587	55986	270274.490339	107436	496628.582926
	Special L3	48860	237049.70503	47397	217797.233507	96257	454846.934010001
	Total	124208	580264.766248	121639	585226.182058	245847	1165490.94830699
			999				
Total		3094875	14333689.162191	2865380	13421080.1685871	5960255	27754769.3307789

Since I have data for Several years, we can limit (filter) data based on the year to keep the report clean and less crowded. Therefore, I have added filter (Year) to the dataset. I have added expression as 'between' and values between 2011 – 2012. This will sort the dataset values which are belongs only to the 2011 and 2012.



## Column Chart for Matrix report

As a visualization element I here used a column chart to provide an easier way to see and understand pattern in data. These visualization elements turn patterns which are invisible in raw data format into visible patterns, that people can understand intuitively.



## Report 2: Parameter Report

### Multiple Parameterized Report – Policy Type and Policy Wise

In this report I have used 2 parameters which have lists of values. Selection of 1<sup>st</sup> parameter value (Policy Type) changes the list of available values in the policy name parameter values

SQL Server Reporting Services

Home > Policy type and policy name wise Annual Insurance Details Report

Policy Type: Corporate Auto

Policy Name: Corporate L3

100%

Find | Next

View Report

		2011		2012	
Policy Type	Policy Name	Monthly Premium	Total Claim Amount	Monthly Premium	Total Claim Amount
Corporate Auto	Corporate L3	352079	1625851.12361	301938	1372770.581833

Selection of the Policy type (1<sup>st</sup> Parameter changed the listed value of 2<sup>nd</sup> parameter

SQL Server Reporting Services

Home > Policy type and policy name wise Annual Insurance Details Report

Policy Type: Corporate Auto, Personal Auto, Special

Policy Name: Corporate L3

100%

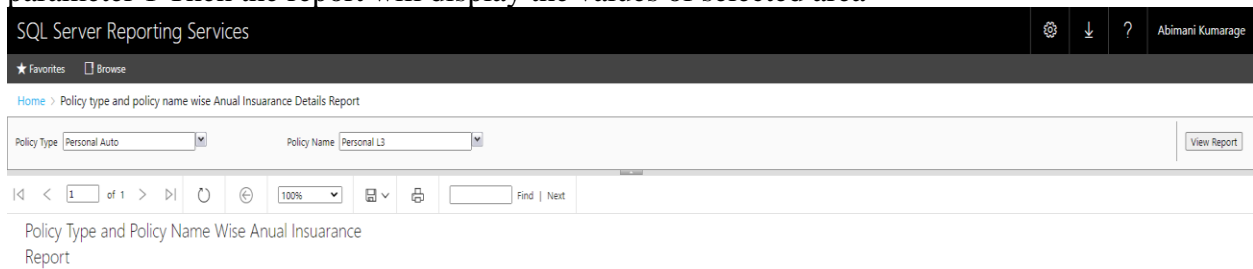
Find | Next

View Report

		2011		2012	
Policy Type	Policy Name	Monthly Premium	Total Claim Amount	Monthly Premium	Total Claim Amount
Corporate Auto	Corporate L3	352079	1625851.12361	301938	1372770.581833

Here I select Corporate Auto and Personal Auto as a first parameter. So according to the query I written, Second parameter will filter the Policy names Related to the Policy types that I have choose in

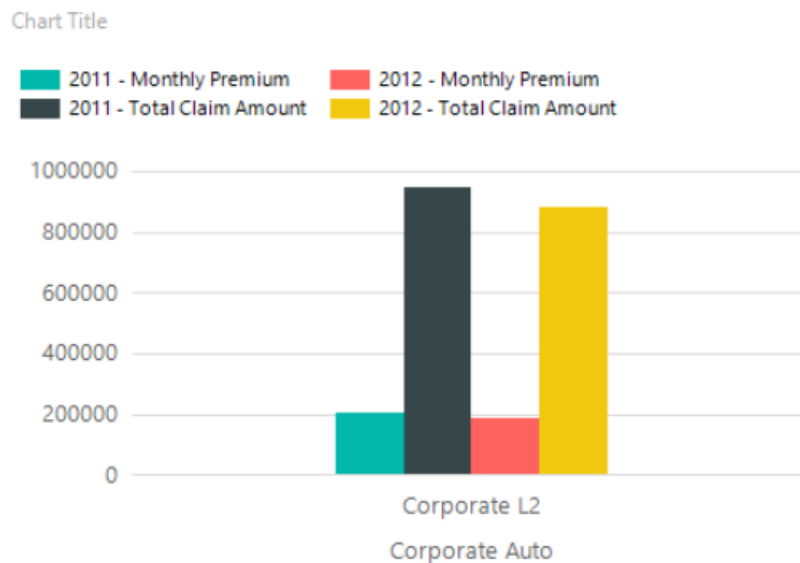
parameter 1 Then the report will display the values of selected area



## Colum Chart for Multiple Parameterized Report

As a visualization element I here used a column chart to provide an easier way to see and understand pattern in data. These visualization elements turn patterns which are invisible in raw data format into visible patterns, that people can understand intuitively. This Colum chart will provide more understandability about the data and compare each policy type and policy wise values with each other.

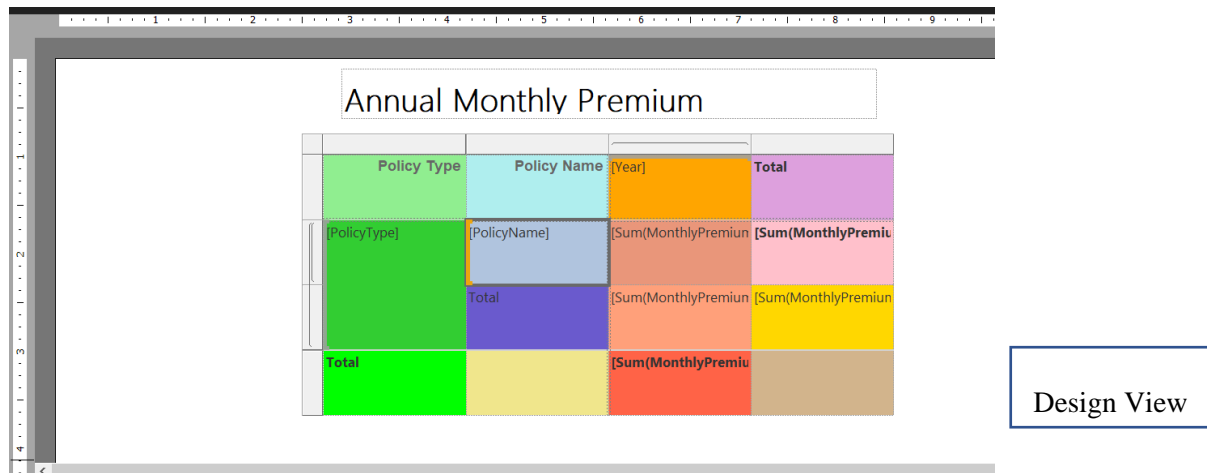
## Policy Type and Policy wise Bar chart



### Report 3: SSRS Drill Down Report:

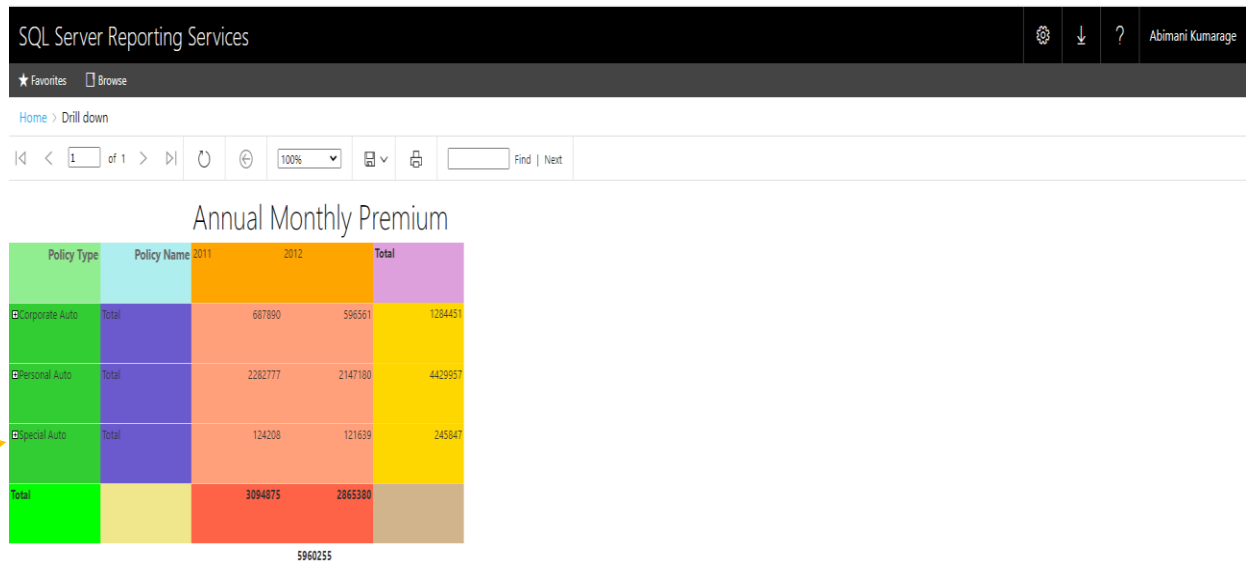
Drill down report will take users from general view of the data to a more specific view, enabling them to dig deeper into the layers in a hierarchy.

#### Annual Monthly Premium Report



The image shows the design view of an SSRS report titled "Annual Monthly Premium". The report is displayed within a window with a ruler at the top and a scroll bar on the left. The report content is a table with four columns: "Policy Type", "Policy Name", "[Year]", and "Total". The table has four rows: a header row, a row for "Corporate Auto", a row for "Personal Auto", and a row for "Special Auto". Each row has a "Total" column. The table is styled with a color-coded grid. A "Design View" label is present in the bottom right corner.

Policy Type	Policy Name	[Year]	Total
[PolicyType]	[PolicyName]	[Sum(MonthlyPremium]	[Sum(MonthlyPremium]
Total	Total	[Sum(MonthlyPremium]	[Sum(MonthlyPremium]
Total	Total	[Sum(MonthlyPremium]	[Sum(MonthlyPremium]



The image shows the preview view of the SSRS report titled "Annual Monthly Premium". The report is displayed within a window with a title bar "SQL Server Reporting Services" and a user name "Abimani Kumara". The report content is a table with four columns: "Policy Type", "Policy Name", "2011", "2012", and "Total". The table has four rows: a header row, a row for "Corporate Auto", a row for "Personal Auto", and a row for "Special Auto". Each row has a "Total" column. The table is styled with a color-coded grid. A "Preview View" label is present in the bottom right corner. An orange arrow points from the "Preview View" label to the table.

Policy Type	Policy Name	2011	2012	Total
Corporate Auto	Total	687890	596561	1284451
Personal Auto	Total	2282777	2147180	4429957
Special Auto	Total	124208	121639	245847
Total	Total	3094875	2865380	5960255

the user has clicked the plus signs (+) in the report to show detail data

SQL Server Reporting Services

★ Favorites □ Browse

Home > Drill down

1 of 1 100% Find | Next

### Annual Monthly Premium

Policy Type	Policy Name	2011	2012	Total
Corporate Auto	Total	687890	596561	1284451
Personal Auto	Total	2282777	2147180	4429957
Special Auto	Total	124208	121639	245847
Total		3094875	2865380	5960255

In this SSRS Report it shows the Monthly Premium details in the insurance Policy wise and Year wise

For the rows I have included a Policy type hierarchy (PolicyType → Policy) so that report can be view by Policy Type wise and Policy Wise.

SQL Server Reporting Services

★ Favorites □ Browse

Home > Drill down

1 of 1 100% Find | Next

### Annual Monthly Premium

Policy Type	Policy Name	2011	2012	Total
Corporate Auto	Corporate L1	128443	107184	235627
	Corporate L2	207368	187439	394807
	Corporate L3	352079	301938	654017
	Total	687890	596561	1284451
Personal Auto	Personal L1	417592	391986	809578
	Personal L2	709240	667919	1377159
	Personal L3	1155945	1087275	2243220
	Total	2282777	2147180	4429957
Special Auto	Special L1	23898	18256	42154
	Special L2	51450	55988	107436
	Special L3	48860	47397	96257
	Total	124208	121639	245847
Total		3094875	2865380	5960255

Policy Type → Policy



## Report 4: Create an SSRS drill-through report.

Drill-through report allow users to pass from one report to another while analyzing the same set of data

### Policy Effectiveness Analysis

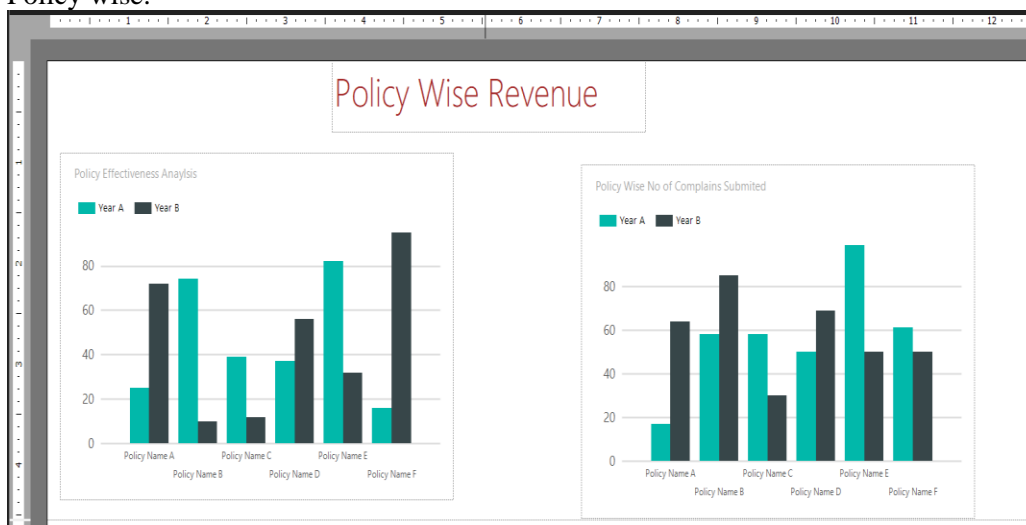
Here I had compared and analysis connection between most famous policy type and policy type among clients between a year and the no of complain got from the users for the particular policy type and policy

```
select
dp.PolicyName,dpt.PolicyType,dcd.CustomerAlternateID,dcd.Gender,dcd.CustomerName,dcd.State,
dcd.LocationCode,dd.Month,dd.MonthName,dd.Year,fi.CustLifeValue,fi.MonthlyPremium,fi.TotalClaim
Amount,
fi.TotalPaidAmount,fi.RewardPoints,fi.NoOfComplains,fi.MonthSinceLastClaim,fi.MonthSincePolicyInc
ept
from [dbo].[FactInsuarance] fi
inner join [dbo].[DimPolicy] dp
on fi.[PolicyKey] = dp.[PolicySK]
inner join [dbo].[DimPolicyType] dpt
on dp.[PolicyTypekey] = dpt.PolicyTypeSK
inner join [dbo].[DimCustomerDetails] dcd
on fi.[CustomerKey] = dcd.[CustomerSK]
inner join [dbo].[DimDate] dd
on fi.[EffectDateKey] = dd.[DateKey]

where dpt.PolicyType = @Type
```

**SQL query for the Detail Report**

Here I have created a **Parameter** called Type. So that when I clicked any kind of policy type it will show the detailed report about Customer Lifetime value and no. of complains from Policy Type Wise and Policy wise.



**Design view of summary Graph**

## Summary Report

SQL Server Reporting Services

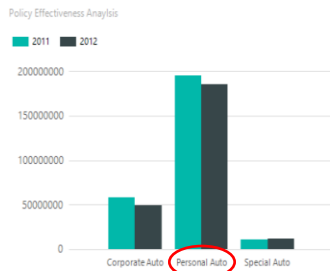
Home > Policy Effectiveness Analysis

1 of 1

100%

Find | Next

### Policy Type Wise Revenue



In this report customer lifetime value is listed according to the policy type. When user click one of the policies type it will show the detailed report of the policies related to the clicked policy type. (Here I have clicked a personal Auto policy type in summary report and it will direct me to the detail report which have the details related to the **personal Auto** policy type's policies

## Detailed Report

SQL Server Reporting Services

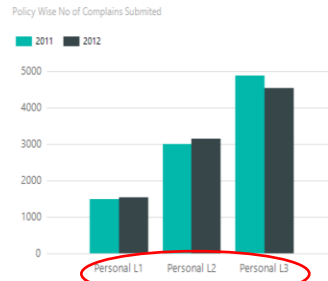
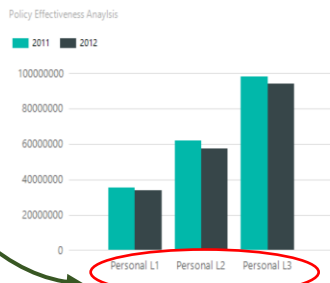
Home > Policy Effectiveness Analysis

1 of 1

100%

Find | Next

### Policy Wise Revenue



As above query shows I have created a '**type**' parameter using where clause. And it is not necessary to fill this parameter by the user. instead, need a value for this parameter to be send by the first level report 'Policy Effectiveness Analysis'.

Right clicked on the parameter 'Type' and select Parameter Properties to open the Parameter Properties window. Set the Select parameter visibility as Hidden.

Report Parameter Properties

General

Available Values

Default Values

Advanced

Change name, data type, and other options.

Name:

Type

Prompt:

Type

Data type:

Text

☐ Allow blank value ("")

☐ Allow null value

☐ Allow multiple values

Select parameter visibility:

☐ Visible

☒ Hidden

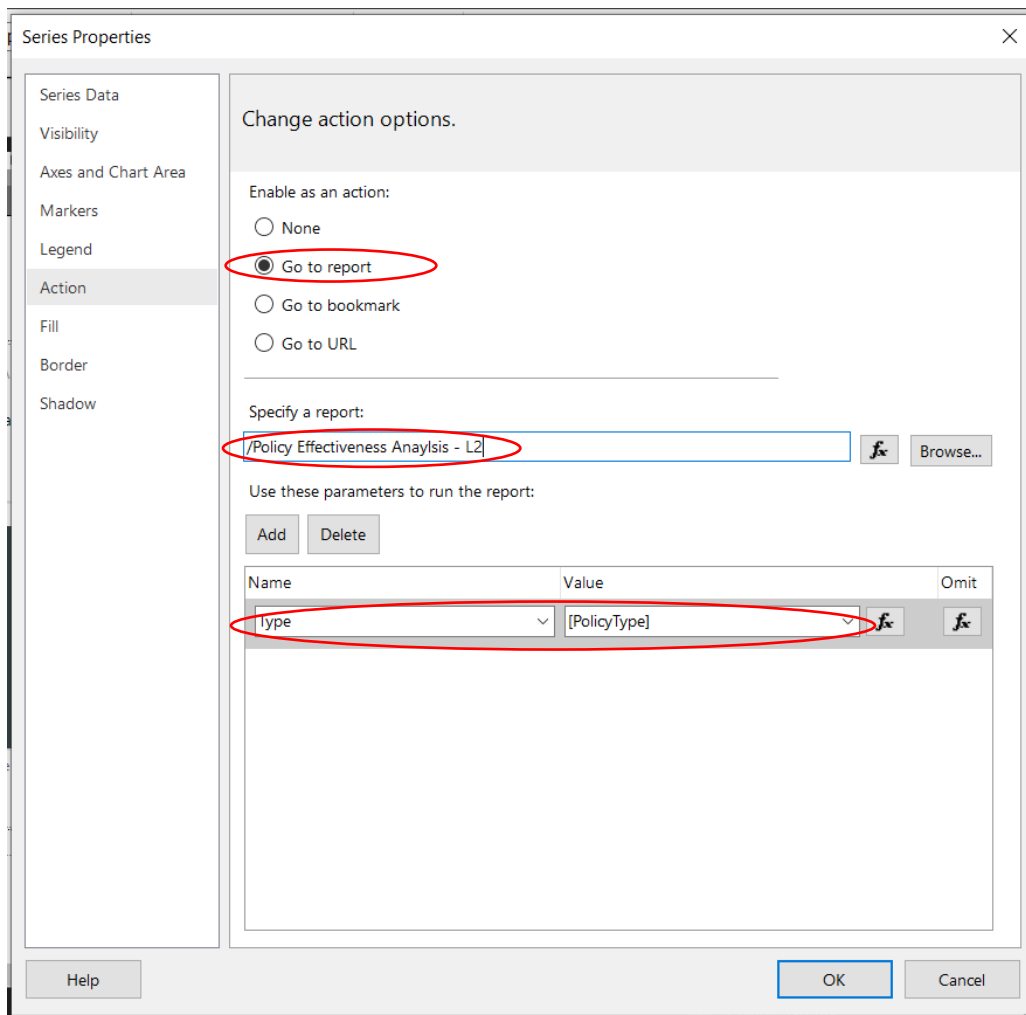
☐ Internal

Help OK Cancel

To edit the first level report 'Policy Effectiveness Analysis' and enable a click event to allow the user to click on a Policy Type (a column in the column chart) to drill-through to the second level report I created. To do so, open the first level report 'Policy Effectiveness Analysis' in Repot Builder. Right clicked on any column of the 'Policy-wise Revenue' chart and select Series Properties... to open Series Properties window.

In the **Series Properties** window, clicked to **Action** section and select **Go to report**.

Under, specify a report, select my second level report 'Policy Effectiveness Analysis – L2' using the Browse... button. Click on Add button and select 'Type' for Name (this is defined in the 'Policy Effectiveness Analysis – L2' report). Select '[PolicyType]' for Value and then I clicked OK.



I have repeated the same step for the **no of complain submitted** graph.

## References: -

- <https://courseweb.sliit.lk/course/view.php?id=4696>
- <https://docs.microsoft.com/en-us/sql/reporting-services/create-deploy-and-manage-mobile-and-paginated-reports?view=sql-server-ver15>
- <https://www.contextures.com/excel-pivot-table-report-filters.html>
- <https://support.microsoft.com/en-us/office/create-a-pivottable-to-analyze-worksheet-data-a9a84538-bfe9-40a9-a8e9-f99134456576>
- <https://www.youtube.com/watch?v=8gGcPDNXwo4&t=914s>
- <https://www.youtube.com/watch?v= 3wf2Cg8WWho&feature=youtu.be>

**END**