

Data Warehousing, Visualizations

Student: Chris Vlassis

The Dataset can be found in kaggle:

[https://www.kaggle.com/datasets/mastmustu/insurance-claims-fraud-data?
select=insurance data.csv](https://www.kaggle.com/datasets/mastmustu/insurance-claims-fraud-data?select=insurance_data.csv)

The Dataset can be found in kaggle and contains 3 different tables with 55 columns. The employee data table and vendor data table contain the information of employee(Agents) and vendor respectively. Information such as address lines, the persons names, postal code and date of joining the company are removed from the data warehouse. Regarding the insurance data table (transactional) customer name, addresses of customers, SSN of customers, number of family members, house type and routing number are also removed since they won't be used in the analysis. Each line of the insurance data table represents a transaction. Each transaction has information for the customer, employee and vendor. Also, information about the incident is present.

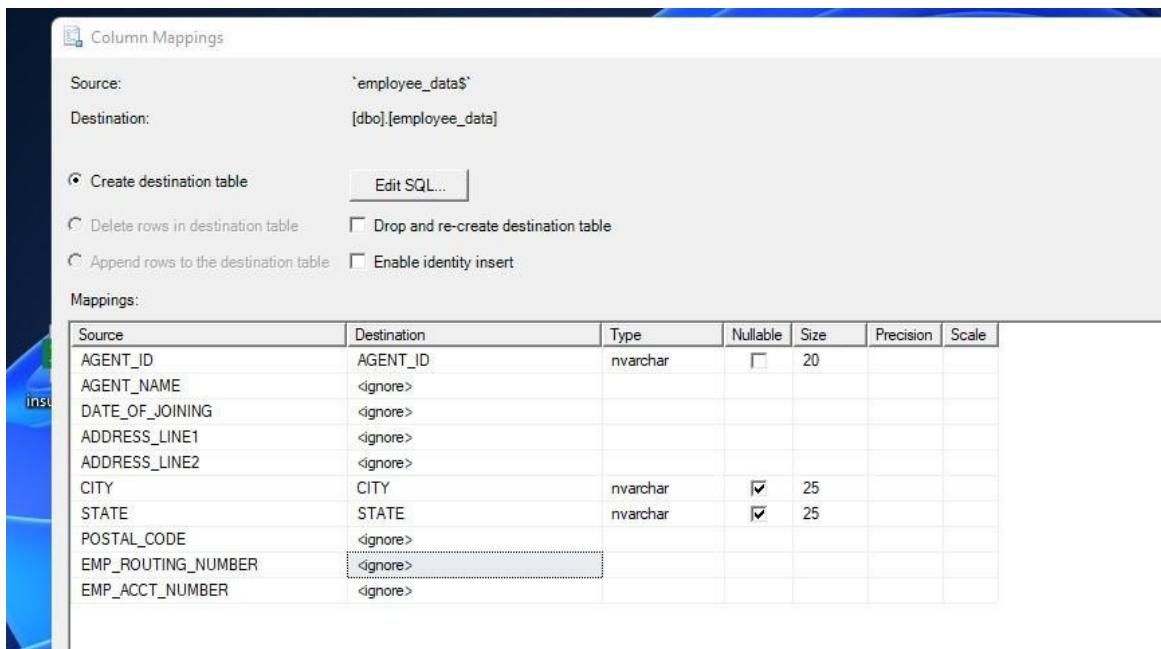
This paper is broken into 6 Sections. In Section one and two the data are inserted and the first necessary manipulations are being done in order to have better quality of data. Then, they are inserted into the staging layer. In Section three the tables of the data warehouse are created and the necessary data are inserted. In Section four the connection to SSAS is being made and in Section five the connection to tableau. Finally, in Section

6, the companies data are analyzed using tableau.

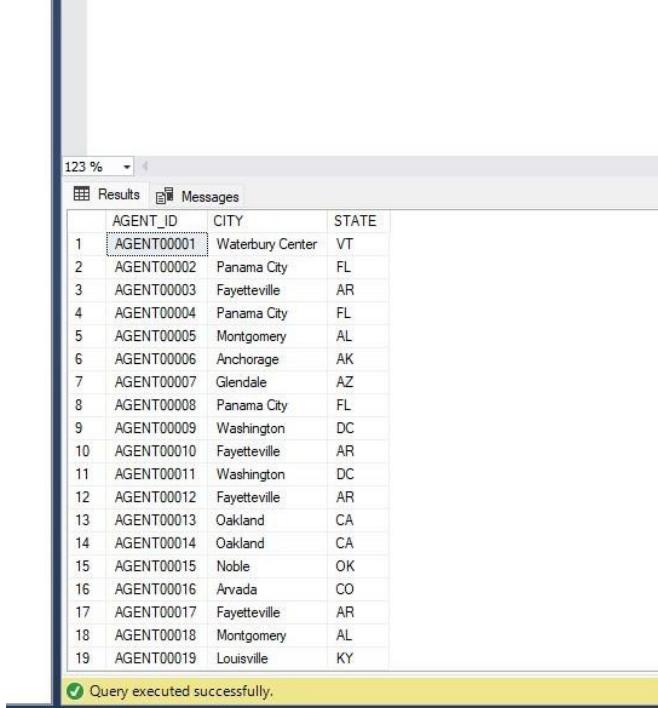
Section 1, Inserting the data:	3
Section 2, The staging layer:	8
Section 3, The data warehouse schema:	12
Section 4, SQL Server Analysis	40
Section 5, Connecting to Tableau	48
Section 6, Analyzing the company	53

Section 1, Inserting the data:

What follows are the screenshots from where the tables were uploaded:
Employee:



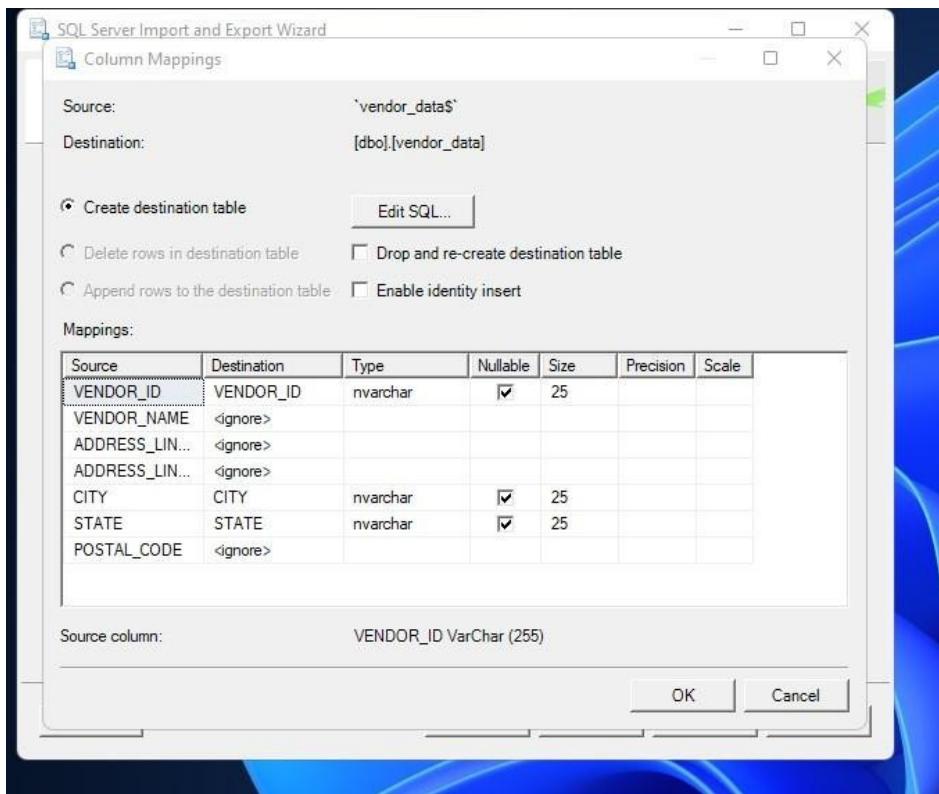
Source	Destination	Type	Nullable	Size	Precision	Scale
AGENT_ID	AGENT_ID	nvarchar	<input type="checkbox"/>	20		
AGENT_NAME	<ignore>					
DATE_OF_JOINING	<ignore>					
ADDRESS_LINE1	<ignore>					
ADDRESS_LINE2	<ignore>					
CITY	CITY	nvarchar	<input checked="" type="checkbox"/>	25		
STATE	STATE	nvarchar	<input checked="" type="checkbox"/>	25		
POSTAL_CODE	<ignore>					
EMP_ROUTING_NUMBER	<ignore>					
EMP_ACCT_NUMBER	<ignore>					



	AGENT_ID	CITY	STATE
1	AGENT00001	Waterbury Center	VT
2	AGENT00002	Panama City	FL
3	AGENT00003	Fayetteville	AR
4	AGENT00004	Panama City	FL
5	AGENT00005	Montgomery	AL
6	AGENT00006	Anchorage	AK
7	AGENT00007	Glendale	AZ
8	AGENT00008	Panama City	FL
9	AGENT00009	Washington	DC
10	AGENT00010	Fayetteville	AR
11	AGENT00011	Washington	DC
12	AGENT00012	Fayetteville	AR
13	AGENT00013	Oakland	CA
14	AGENT00014	Oakland	CA
15	AGENT00015	Noble	OK
16	AGENT00016	Avada	CO
17	AGENT00017	Fayetteville	AR
18	AGENT00018	Montgomery	AL
19	AGENT00019	Louisville	KY

Query executed successfully.

Vendor:



Results		
1	VNDR00001	Fayetteville
2	VNDR00002	Montgomery
3	VNDR00003	Montgomery
4	VNDR00004	Savannah
5	VNDR00005	Nashville
6	VNDR00006	Arvada
7	VNDR00007	Ashland
8	VNDR00008	West Sacramento
9	VNDR00009	Arvada
10	VNDR00010	Pasadena
11	VNDR00011	Panama City Beach
12	VNDR00012	Nashville
13	VNDR00013	Montgomery
14	VNDR00014	Washington
15	VNDR00015	Louisville
16	VNDR00016	Arvada
17	VNDR00017	Nashville
18	VNDR00018	Washington
19	VNDR00019	Glendale

Insurance(Transactional):

Source: `insurance_data\$`
 Destination: [dbo].[insurance_data]

Create destination table

Delete rows in destination table Drop and re-create destination table

Append rows to the destination table Enable identity insert

Mappings:

Source	Destination	Type	Nullable	Size	Precision	Scale
TXN_DATE_TIME	TXN_DATE_TIME	date	<input checked="" type="checkbox"/>			
TRANSACTION_ID	TRANSACTION_ID	nvarchar	<input checked="" type="checkbox"/>	50		
CUSTOMER_ID	CUSTOMER_ID	nvarchar	<input checked="" type="checkbox"/>	50		
POLICY_NUMBER	POLICY_NUMBER	nvarchar	<input checked="" type="checkbox"/>	50		
POLICY_EFF_DT	POLICY_EFF_DT	date	<input checked="" type="checkbox"/>			
LOSS_DT	LOSS_DT	date	<input checked="" type="checkbox"/>			
REPORT_DT	REPORT_DT	datetime	<input checked="" type="checkbox"/>			
INSURANCE_TYPE	INSURANCE_TYPE	nvarchar	<input checked="" type="checkbox"/>	50		
PREMIUM_AMOUNT	PREMIUM_AMOUNT	float	<input checked="" type="checkbox"/>			
CLAIM_AMOUNT	CLAIM_AMOUNT	float	<input checked="" type="checkbox"/>			
CUSTOMER_NAME	<ignore>					
ADDRESS_LINE1	<ignore>					
ADDRESS_LINE2	<ignore>					
CITY	CITY	nvarchar	<input checked="" type="checkbox"/>	50		
STATE	STATE	nvarchar	<input checked="" type="checkbox"/>	50		
POSTAL_CODE	POSTAL_CODE	float	<input checked="" type="checkbox"/>			
SSN	<ignore>					
MARITAL_STATUS	MARITAL_STATUS	nvarchar	<input checked="" type="checkbox"/>	50		
AGE	AGE	float	<input checked="" type="checkbox"/>			
TENURE	TENURE	float	<input checked="" type="checkbox"/>			
EMPLOYMENT_STATUS	EMPLOYMENT_STATUS	nvarchar	<input checked="" type="checkbox"/>	50		
NO_OF_FAMILY_MEMBERS	<ignore>					
RISK_SEGMENTATION	RISK_SEGMENTATION	nvarchar	<input checked="" type="checkbox"/>	50		
HOUSE_TYPE	<ignore>					
SOCIAL_CLASS	SOCIAL_CLASS	nvarchar	<input checked="" type="checkbox"/>	50		
ROUTING_NUMBER	<ignore>					
ACCT_NUMBER	<ignore>					
CUSTOMER_EDUCATION_LEVEL	CUSTOMER_EDUCATION_LEVEL	nvarchar	<input checked="" type="checkbox"/>	50		
CLAIM_STATUS	CLAIM_STATUS	nvarchar	<input checked="" type="checkbox"/>	50		
INCIDENT_SEVERITY	INCIDENT_SEVERITY	nvarchar	<input checked="" type="checkbox"/>	50		
AUTHORITY_CONTACTED	AUTHORITY_CONTACTED	nvarchar	<input checked="" type="checkbox"/>	50		
Source column:	VENDOR_ID VarChar (255)					

CONTACTED	CONTACTED	INJURED	INJURED
ANY_INJURY	ANY_INJURY	float	<input checked="" type="checkbox"/>
POLICE_REPORT_AVAILABLE	POLICE_REPORT_AVAILABLE	float	<input checked="" type="checkbox"/>
INCIDENT_STATE	INCIDENT_STATE	nvarchar	<input checked="" type="checkbox"/> 50
INCIDENT_CITY	INCIDENT_CITY	nvarchar	<input checked="" type="checkbox"/> 50
INCIDENT_HOUR_OF_THE_DAY	INCIDENT_HOUR_OF_THE_DAY	float	<input checked="" type="checkbox"/>
AGENT_ID	AGENT_ID	nvarchar	<input checked="" type="checkbox"/> 50
VENDOR_ID	VENDOR_ID	nvarchar	<input checked="" type="checkbox"/> 50

123 %

Results Messages

	VENDOR_ID	CITY	STATE
1	VNDR00001	Fayetteville	AR
2	VNDR00002	Montgomery	AL
3	VNDR00003	Montgomery	AL
4	VNDR00004	Savannah	GA
5	VNDR00005	Nashville	TN
6	VNDR00006	Arvada	CO
7	VNDR00007	Ashland	MA
8	VNDR00008	West Sacramento	CA
9	VNDR00009	Arvada	CO
10	VNDR00010	Pasadena	MD
11	VNDR00011	Panama City Beach	FL
12	VNDR00012	Nashville	TN
13	VNDR00013	Montgomery	AL
14	VNDR00014	Washington	DC
15	VNDR00015	Louisville	KY
16	VNDR00016	Arvada	CO
17	VNDR00017	Nashville	TN
18	VNDR00018	Washington	DC
19	VNDR00019	Glendale	AZ

Query executed successfully.



Section 2, The staging layer:

In order to have clean and organized data, in the staging layer the data are inserted, cleaned and some basic transformations are being made. Also, other issues are being resolved such as missing values.

With that being said, first, the Staging Table is created which contains the data from the three tables. Columns that don't have any use are being dropped and any null values from the dimension are replaced with N/A.

What follows are the screenshots from the staging layer:

Creating the staging table:

The screenshot shows the SQL Server Management Studio (SSMS) interface. In the center, there is a code editor window displaying a T-SQL script. The script creates a new table named 'Staging_Table' by selecting specific columns from three tables: 'insurance_data' (aliased as 'ins'), 'employee_data' (aliased as 'ed'), and 'vendor_data' (aliased as 'vd'). The columns selected are 'ins.*', 'vd.CITY AS Vendor_City', 'vd.STATE AS Vendor_State', 'ed.CITY AS Employee_City', and 'ed.State AS Employee_State'. The script uses LEFT JOINs to handle cases where a row might not exist in one of the joined tables. The code editor has syntax highlighting and line numbers on the left. Below the code editor, the message pane shows the execution results: '(10000 rows affected)' and 'Completion time: 2022-11-12T19:40:54.1451065+02:00'.

```
1
2
3   SELECT
4     ins.*,
5     vd.CITY AS Vendor_City,
6     vd.STATE AS Vendor_State,
7     ed.CITY AS Employee_City,
8     ed.State AS Employee_State
9
10    INTO Staging_Table
11
12    FROM insurance_data ins
13    LEFT JOIN employee_data ed ON ins.AGENT_ID = ed.AGENT_ID
14    LEFT JOIN vendor_data vd ON vd.VENDOR_ID = ins.VENDOR_ID;
15
```

123 %

Messages

(10000 rows affected)

Completion time: 2022-11-12T19:40:54.1451065+02:00

Removing any unnecessary columns:

The screenshot shows the SQL Server Management Studio interface. A query window titled 'SQLQuery14.sql' is open, containing the following code:

```
1 ALTER TABLE Staging_Table DROP COLUMN POSTAL_CODE;
```

Searching the measurements for null values:

The screenshot shows the SQL Server Management Studio interface. A query window titled 'SQLQuery16.sql' is open, containing the following code:

```
1 /****** Script for SelectTopNRows command from SSMS *****/
2 SELECT PREMIUM_AMOUNT from Staging_Table
3 where PREMIUM_AMOUNT is null
4
5
6 SELECT PREMIUM_AMOUNT from Staging_Table
7 where CLAIM_AMOUNT is null
```

The results pane shows a single row with the value 'PREMIUM_AMOUNT'.

At the bottom of the screen, a status bar displays the message 'Query executed successfully.'

For the dimensions, we replace nulls with N/A:

The screenshot shows a SQL Server Management Studio (SSMS) window. The title bar indicates two open queries: "SQLQuery17.sql - (l...CHRISVL\vllass (55))" and "SQLQuery16.sql - (l...CHRISVL\vllass (52))". The main pane displays a script for updating dimension tables. The script consists of 18 numbered lines, each starting with an UPDATE statement for a specific table (Staging_Table). The updates set values for columns CITY, VENDOR_ID, [Vendor_City], [Vendor_State], [Employee_City], and [Employee_State] to 'N/A' where the corresponding columns are null. The script ends with a message indicating 0 rows affected and a completion time.

```
1 UPDATE Staging_Table
2 SET CITY='N/A'
3 WHERE CITY IS NULL;
4 UPDATE Staging_Table
5 SET VENDOR_ID='N/A'
6 WHERE VENDOR_ID is null
7 UPDATE Staging_Table
8 SET [Vendor_City]='N/A'
9 WHERE [Vendor_City] is null
10 UPDATE Staging_Table
11 SET [Vendor_State]='N/A'
12 WHERE [Vendor_State] is null
13 UPDATE Staging_Table
14 SET [Employee_City]='N/A'
15 WHERE [Employee_City] is null
16 UPDATE Staging_Table
17 SET [Employee_State]='N/A'
18 WHERE [Employee_State] is null
19
```

123 %

Messages

(0 rows affected)

Completion time: 2022-11-13T15:32:40.8357174+02:00

Searching for any other null values in the dimensions:

The screenshot shows a SQL Server Management Studio (SSMS) interface. In the center, there is a results grid with the following columns:

TXN_DATE_TIME	TRANSACTION_ID	CUSTOMER_ID	POLICY_NUMBER	POLICY_EFF_DT	LOSS_DT	REPORT_DT	INSURANCE_TYPE	PREMIUM_AMOUNT	CLAIM_AMOUNT	CITY	STATE	MARITAL_STATUS	AGE	TENURE

On the left, the query window displays the following T-SQL code:

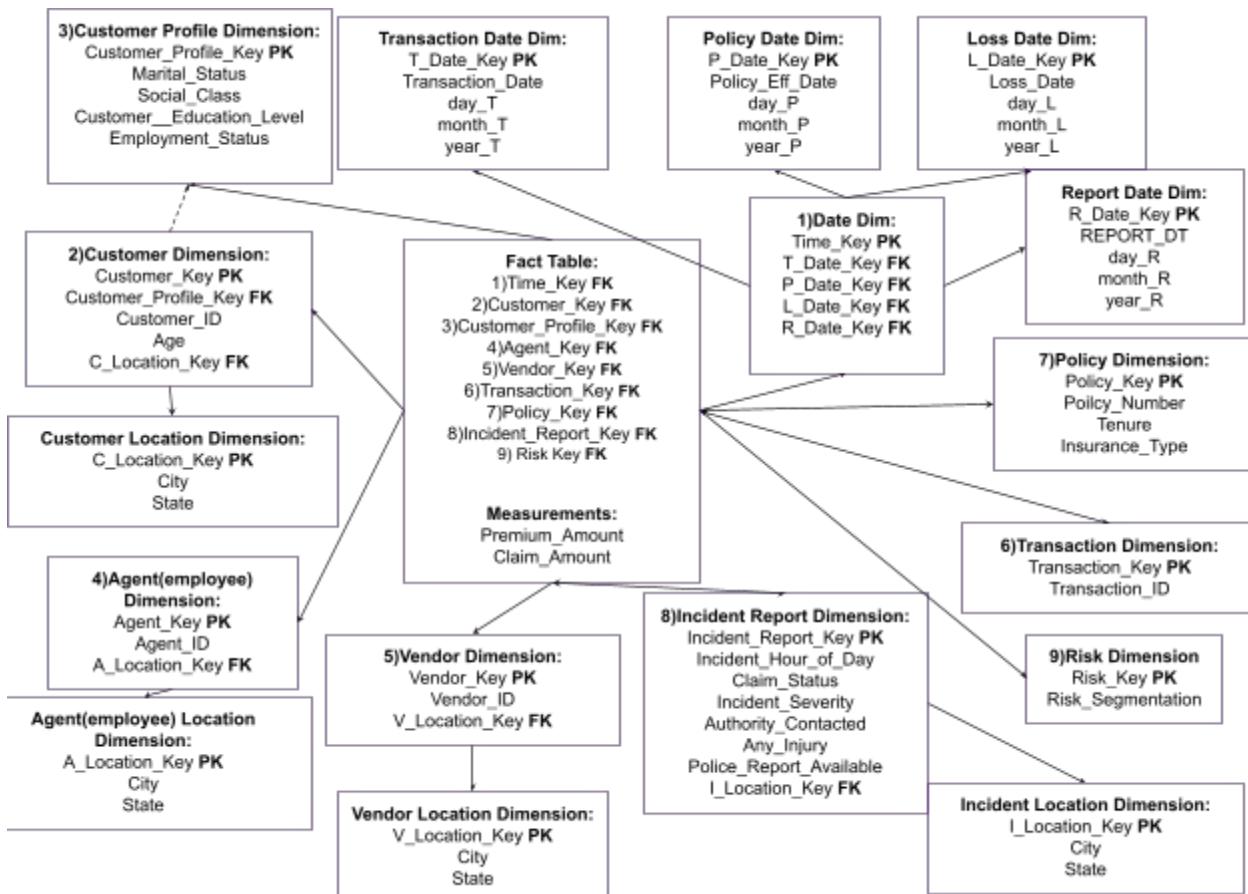
```
1 select * from Staging Table
2 where
3     [TXN_DATE_TIME] IS NULL
4     or [TRANSACTION_ID] IS NULL
5     or [CUSTOMER_ID] IS NULL
6     or [POLICY_NUMBER] IS NULL
7     or [POLICY_EFF_DT] IS NULL
8     or [LOSS_DT] IS NULL
9     or [REPORT_DT] IS NULL
10    or [INSURANCE_TYPE] IS NULL
11    or [PREMIUM_AMOUNT] IS NULL
12    or [CLAIM_AMOUNT] IS NULL
13    or [CITY] IS NULL
14    or [STATE] IS NULL
15    or [MARITAL_STATUS] IS NULL
16    or [AGE] IS NULL
17    or [TENURE] IS NULL
18    or [EMPLOYMENT_STATUS] IS NULL
19    or [RISK_SEGMENTATION] IS NULL
20    or [SOCIAL_CLASS] IS NULL
21    or [CUSTOMER_EDUCATION_LEVEL] IS NULL
22    or [CLAIM_STATUS] IS NULL
23    or [INCIDENT_SEVERITY] IS NULL
24    or [AUTHORITY_CONTACTED] IS NULL
25    or [ANY_INJURY] IS NULL
26    or [POLICE_REPORT_AVAILABLE] IS NULL
27    or [INCIDENT_STATE] IS NULL
28    or [INCIDENT_CITY] IS NULL
29    or [INCIDENT_HOUR_OF_THE_DAY] IS NULL
30    or [AGENT_ID] IS NULL
31    or [VENDOR_ID] IS NULL
```

The right side of the screen shows the properties pane for the current connection.

Section 3, The data warehouse schema:

In order to have the company data in easy reach the data warehouse is created which contains the information that the company needs to run reports, analytics and other models. A snowflake schema is created. In order to create the data warehouse the UI is used to create the tables, and SQL to insert the data. Some tables will be created and later dropped. Also, some later changes were made. The day, month and year columns of each of the **date dim** will be created in the end using SQL. So, they will not be found in the first queries.

An organized and clean data warehouse is created. The customer profile dimension is used in order to have the ‘fast changing attributes’ of the customers all together and also more static and big dimensions are created such as incident report dimension because these are data that do not change. Also, each dimension is created in order to have all the related attributes together.



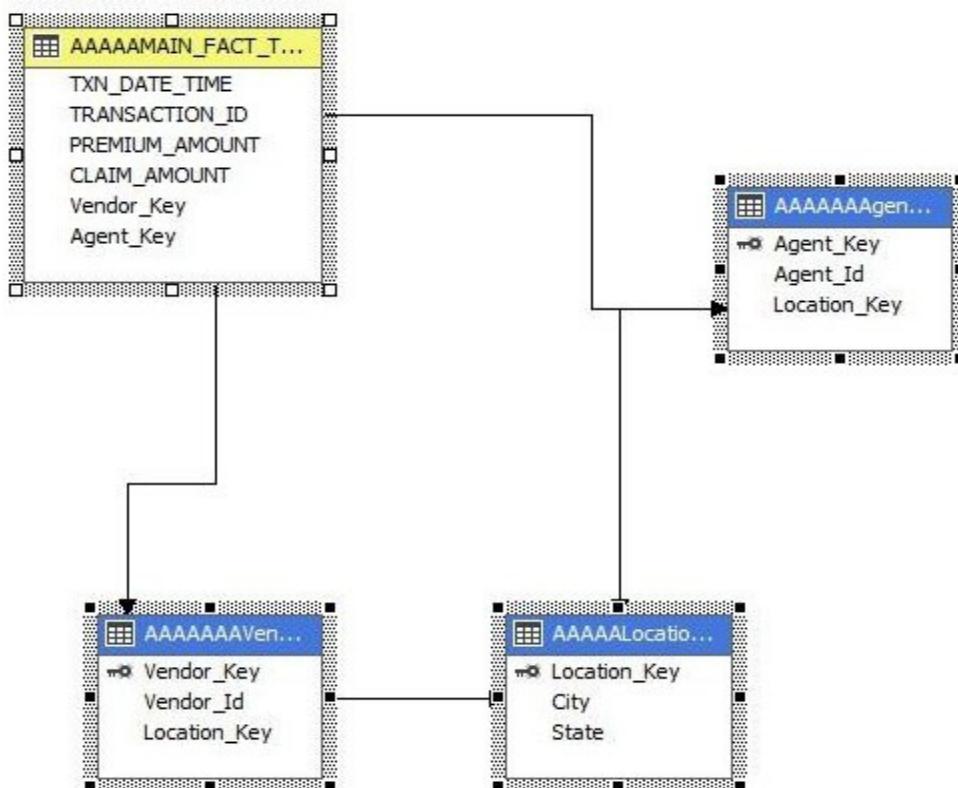
The Date and City-State dimensions were the trickiest mainly because there were many and different options to represent the data format. For the Date

Dimensions that's because there are 4 date columns. We could either use one table with three columns, where the first column will be the date key, the second the Date and the Date type for the third column but that way we would create many unnecessary rows and we would have to always filter our data depending on the date, otherwise we would have duplicate values in our report.

Another way to represent the data for the dates is by using, firstly, four different tables. Each table would have the date key for each of the four dates. And also a table with two columns date keys and date which table is the UNION of all the dates that are present in the data set. But, by using this method the JOINS have to be changed depending on the Date Type that wants to be analyzed.

For the City-State, we could use a table for all the locations that would be the UNION of all the locations. Later these locations would be connected to the agent, vendor, incident report and customer dimensions. That way we could have one location table. But, again, if we don't use the cube to connect to the report software we would have to change the JOINS depending on the city-state we would like to make the analysis for.

Sample screenshot of one location table:



With that being said, a simpler logic was preferred to remove any unnecessary confusion and because the data warehouse is used to connect to the report software.

What follows are the screenshots and queries for the creation of the dimension tables:

Transaction Date Dim:

The screenshot shows the SQL Server Management Studio (SSMS) interface. On the left, the Object Explorer displays a database named 'CHRSVL.master'. In the center, a table named 'transaction_date_dim' is being designed. The table has four columns: 'T_Date_Key' (int, primary key), 'Transaction_Date' (date), 'day_T' (int), 'month_T' (int), and 'year_T' (int). On the right, a 'Indexes/Keys' dialog box is open, showing a primary key named 'PK_Transaction_Date_Dim' and a unique index named 'IX_Transaction_Date_Dim'. The 'General' properties for the primary key are set to 'Transaction_Date (ASC)', 'Is Unique: Yes', and 'Type: Index'. The 'Table Designer' section shows 'Create As Clustered: No' and 'Fill Specification: PRIMARY'. At the bottom of the dialog, there are 'Add' and 'Delete' buttons, and a 'Close' button.

Reported Date Dim

The screenshot shows the 'Report Date_dim' table in the 'CHRISVL.master - dbo' database. The table has four columns: R_Report_Date_Key (int, primary key), Reported_Date (date), day_R (int), month_R (int), and year_R (int). The 'Reported_Date' column is set to allow nulls. The 'Table Designer' properties pane is open, showing various settings like Collation, Computed Column Specification, Full-text Specification, and Identity Specification. The 'Indexes/Keys' dialog box is open, displaying a primary key named 'PK_Report_Date_dim' with a single column 'Reported_Date'. The 'General' tab of the key properties shows 'Columns' as 'Reported_Date (ASC)', 'Is Unique' as 'Yes', and 'Type' as 'Index'. The 'Identity' tab shows '(Name)' as 'IX_Report_Date_dim'. The 'Table Designer' tab shows 'Create As Clustered' as 'No', 'Data Space Specification' as 'PRIMARY', and 'Fill Specification'.

Policy date dim:

The screenshot shows the 'Policy Date_Dim' table in the 'CHRISVL.master - dbo' database. The table has five columns: P_Report_Date_Key (int, primary key), Policy_Eff_Date (date), day_P (int), month_P (int), and year_P (int). The 'Policy_Eff_Date' column is set to allow nulls. The 'Table Designer' properties pane is open, showing various settings like Collation, Computed Column Specification, Full-text Specification, and Identity Specification. The 'Indexes/Keys' dialog box is open, displaying a primary key named 'PK_Policy_Date_Dim' with a single column 'Policy_Eff_Date'. The 'General' tab of the key properties shows 'Columns' as 'Policy_Eff_Date (ASC)', 'Is Unique' as 'Yes', and 'Type' as 'Index'. The 'Identity' tab shows '(Name)' as 'IX_Policy_Report_Date_Dim'. The 'Table Designer' tab shows 'Create As Clustered' as 'No', 'Data Space Specification' as 'PRIMARY', and 'Fill Specification'.

Loss date dim:

The screenshot shows the 'Loss Date Dim' table in the 'CHRISVL.master' database. The table has five columns: 'Loss_Date_Key' (int, primary key, not null), 'Loss_Date' (date), 'day_L' (int), 'month_L' (int), and 'year_L' (int). The 'Allow Nulls' column for 'Loss_Date' and other columns is checked.

Column Name	Data Type	Allow Nulls
Loss_Date_Key	int	<input type="checkbox"/>
Loss_Date	date	<input checked="" type="checkbox"/>
day_L	int	<input checked="" type="checkbox"/>
month_L	int	<input checked="" type="checkbox"/>
year_L	int	<input checked="" type="checkbox"/>

Column Properties

- (General)**
 - (Name)**: Loss_Date_Key
 - Allow Nulls**: No
 - Data Type**: int
 - Default Value or Binding**: <database default>
- Table Designer**
 - Collation**: <database default>
 - Computed Column Specification**: int
 - Condensed Data Type**: No
 - Description**: Description
 - Deterministic**: Yes
 - DTS-published**: No
 - Full-text Specification**: Has Non-SQL Server Subscriber
 - Identity Specification** (Is Identity)
 - Identity Increment: 1
 - Identity Seed: 1
 - Indexable**: Yes
 - Is Columnset**: No
 - Is Sparse**: No
 - Merge-published**: No
 - Not For Replication**: No

First Main date dimension, we will delete it later

The screenshot shows the 'Table_1' table in the 'CHRISVL.master' database. The table has five columns: 'Time_Key' (int, primary key, not null), 'Loss_Date' (date), 'Policy_Date' (date), 'Report_Date' (date), and 'Transaction_Date' (date). The 'Allow Nulls' column for all columns is checked.

Column Name	Data Type	Allow Nulls
Time_Key	int	<input type="checkbox"/>
Loss_Date	date	<input checked="" type="checkbox"/>
Policy_Date	date	<input checked="" type="checkbox"/>
Report_Date	date	<input checked="" type="checkbox"/>
Transaction_Date	date	<input checked="" type="checkbox"/>

Column Properties

- (General)**
 - (Name)**: Time_Key
 - Allow Nulls**: No
 - Data Type**: int
 - Default Value or Binding**: <database default>
- Table Designer**
 - Collation**: <database default>
 - Computed Column Specification**: int
 - Condensed Data Type**: Yes
 - Description**: Description
 - Deterministic**: Yes

The main Date dimension

The screenshot shows the SQL Server Management Studio (SSMS) interface. A table named 'Main_Date_Dim' is being created in the 'CHRISVL.master' database. The table has four columns: 'Time_Key' (int, primary key, not null), 'P_Date_Key' (int), 'R_Date_Key' (int), and 'T_Date_Key' (int). The 'Time_Key' column is defined with properties: Name is 'Time_Key', Allow Nulls is No, Data Type is 'int', and Default Value or Binding is '<database default>'. The 'Table Designer' section shows various properties like Collation, Computed Column Specification, Condensed Data Type, Description, Deterministic, DTS-published, Full-text Specification, Has Non-SQL Server Subscriber, Identity Specification (Is Identity Yes, Identity Increment 1, Identity Seed 1), Indexable, Is Columnset, Is Sparse, Merge-published, Not For Replication, Replicated, and RowGuid.

Queries and comments for the creation of the date dimension

First we have created the 4 Date_Dimension. Then we populate them with the data.

```
insert into Transaction_Date_Dim(Transaction_Date) select distinct  
[TXN_DATE_TIME] from AA_Staging_Table  
insert into Loss_Date_Dim(Loss_Date) select distinct [LOSS_DT] from  
AA_Staging_Table  
insert into Policy_Date_Dim(Policy_Eff_Date) select distinct [POLICY_EFF_DT]  
from AA_Staging_Table  
insert into Report_Date_Dim(Reported_Date) select distinct [REPORT_DT] from  
AA_Staging_Table
```

/ Here we create the date_dim which is used for keeping the date_keys. For now this table is gonna be populated with dates.*

We will later join to bring the keys from the 4 small date table and delete this table/*

```
insert into Date_Dim(Transaction_Date,Policy_Date,Loss_Date,Reported_Date)
```

```
select distinct [TXN_DATE_TIME], [POLICY_EFF_DT], [LOSS_DT],  
[REPORT_DT] from AA_Staging_Table
```

```
/* here i bring the keys from the smaller date tables to the main date table */  
SELECT DISTINCT  
    Time_Key,  
    P_Date_Key,  
    R_Date_Key,  
    T_Date_Key,  
    Loss_Date_Key  
INTO Main_Date_Dim  
FROM Date_Dim DD  
join Policy_Date_Dim PDD on PDD.Policy_Eff_Date = DD.Policy_Date  
join Report_Date_dim RD on RD.Reported_Date = DD.Reported_Date  
join Loss_Date_Dim LS on LS.Loss_Date = DD.Loss_Date  
join Transaction_Date_Dim TD on TD.Transaction_Date =  
DD.Transaction_Date  
  
/* finally we remove the first date_dim table because it doesn't have any use,  
anymore. */  
DROP TABLE Date_Dim
```

The policy dimension:

The screenshot shows the SQL Server Management Studio interface. On the left, the Object Explorer displays a database structure. In the center, the 'Table Designer' is open for the 'dbo.Table_1' table. The 'Policy_Key' column is selected, and its properties are shown in the 'Column Properties' pane. The 'Allow Nulls' checkbox is unchecked. To the right, a 'Indexes/Keys' dialog box is open, showing the 'Selected Primary/Unique Key or Index' section. It lists 'PK_Table_1' with the 'Type' set to 'Identity'. Below it, the 'Table Designer' pane shows the 'Identity Specification' settings, where 'Is Identity' is checked and 'Identity Increment' is set to 1.

Queries for policy dimension

```
/* populating the policy_dim */
insert into Policy_Dim(Policy_Number,Tenure,Insurance_Type)
select distinct
    POLICY_NUMBER,
    TENURE,
    INSURANCE_TYPE
from AA_Staging_Table
```

Transaction dimension

The screenshot shows the SQL Server Management Studio interface. On the left, the Object Explorer displays a database structure. In the center, the Table Designer is open for a table named 'Table_1'. The 'Indexes/Keys' dialog is displayed over the Table Designer, specifically for a primary key named 'PK_Table_1'. The dialog shows the following properties:

Column	Type	Properties
IX_Table_1	Index	Is Unique: Yes, Type: Index, Name: IX_Table_1
PK_Table_1	Primary Key	Is Unique: Yes, Type: Primary Key, Name: PK_Table_1

The Table Designer pane on the left lists various properties for the table, including General, Identity, and Table Designer settings. The General section shows the column 'Transaction_ID' as type int, with 'Allow Nulls' checked.

Queries for transaction dimension

```
/* populating the transaction_dim */
insert into [dbo].[Transaction_dim]([Transaction_ID])
select distinct
    [TRANSACTION_ID],
from AA_Staging_Table
```

Vendor dimension

Table Properties

Column Name	Data Type	Allow Nulls
Vendor_Key	int	<input checked="" type="checkbox"/>
Vendor_ID	nvarchar(50)	<input checked="" type="checkbox"/>
Vendor_Location_Key	int	<input checked="" type="checkbox"/>

Column Properties

(General)

(Name)	Vendor_ID
Allow Nulls	Yes
Data Type	nvarchar
Default Value or Binding	
Length	50
Table Designer	<database default>
Collation	nvarchar(50)
Computed Column Specification	
Condensed Data Type	
Description	
Deterministic	Yes
DTS-published	No
Full-text Specification	No
Has Non-SQL Server Subscriber	No
Identity Specification	No
Indexable	Yes
Is Columnset	No
Is Sparse	No
Merge-published	No
Not For Replication	No
Replicated	No
RowGuid	No
Size	100

Indexes/Keys

Selected Primary/Unique Key or Index:

IX_Table_1*

Editing properties for new unique key or index.

Type	Index
Identity	IX_Table_1
(Name)	
Description	
Table Designer	
Create As Clustered	No
Data Space Specification	PRIMARY
Fill Specification	
Ignore Duplicate Keys	Yes
Included Columns	
Is Disabled	No

Vendor location dimension

Table Properties

Column Name	Data Type	Allow Nulls
Vendor_Location_Key	int	<input checked="" type="checkbox"/>
Vendor_City	nvarchar(50)	<input checked="" type="checkbox"/>
Vendor_State	nvarchar(50)	<input checked="" type="checkbox"/>

Column Properties

(General)

(Name)	Vendor_Location_Key
Allow Nulls	No
Data Type	int
Default Value or Binding	
Table Designer	<database default>
Collation	int
Computed Column Specification	
Condensed Data Type	
Description	
Deterministic	Yes
DTS-published	No
Full-text Specification	No
Has Non-SQL Server Subscriber	No
Identity Specification	Yes
(Is Identity)	Yes
Identity Increment	1
Identity Seed	1
Indexable	Yes
Is Columnset	No

Queries for vendor dimensions:

```
/* populating the 1st vendor dimensions */
```

```
select distinct
```

```
    [VENDOR_ID],  
    [Vendor_City],  
    [Vendor_State]
```

```
INTO Vendors_1
```

```
from [AA_Staging_Table]
```

```
/* populating the vendor location dimension */
```

```
insert into [dbo].[Vendor_Location_Dimension]([Vendor_City],[Vendor_State])
```

```
select distinct Vendor_City, Vendor_State from [dbo].[Vendors_1]
```

```
/* creating the main vendor dimension */
```

```
insert into [dbo].[Vendor_Dimension]([Vendor_ID],[Vendor_Location_Key])
```

```
select distinct Vendor_Id,
```

```
    Vendor_Location_Key
```

```
from [Vendors_1] V
```

```
join [Vendor_Location_Dimension] VLD on VLD.Vendor_City =
```

```
V.Vendor_City and VLD.Vendor_State = V.Vendor_State
```

```
/* deleting the vendor_1, we dont need it anymore */
```

```
DROP TABLE Vendors_1
```

Customer dimension

The screenshot shows the SQL Server Management Studio (SSMS) interface. A table named 'Main_Customer_Dim' is being designed. The 'Customer_ID' column is selected, and a context menu is open, leading to the 'Indexes/Keys' dialog. In this dialog, a new primary key is being created with the name 'PK_Main_Customer_Dim'. The properties for this key are set to use the 'Customer_ID' column in ascending order (ASC). The 'Type' is set to 'Index'. The 'Table Designer' section shows that the key is a clustered index ('Create As Clustered' is checked) and is designated as the primary key ('PRIMARY').

Customer dimension location

The screenshot shows the SQL Server Management Studio (SSMS) interface. A table named 'Customer_Dim_Location' is being designed. The 'C_Location_Key' column is selected, and a context menu is open, leading to the 'Indexes/Keys' dialog. In this dialog, a new primary key is being created with the name 'PK_Customer_Dim_Location'. The properties for this key are set to use the 'C_Location_Key' column. The 'Type' is set to 'Index'. The 'Table Designer' section shows that the key is a clustered index ('Create As Clustered' is checked) and is designated as the primary key ('PRIMARY').

Customer profile dimension

Table Structure:

Column Name	Data Type	Allow Nulls
Customer_Profile_Key	int	<input checked="" type="checkbox"/>
MARITAL_STATUS	nvarchar(50)	<input checked="" type="checkbox"/>
SOCIAL_CLASS	nvarchar(50)	<input checked="" type="checkbox"/>
CUSTOMER_EDUCATION_LEVEL	nvarchar(50)	<input checked="" type="checkbox"/>
EMPLOYMENT_STATUS	nvarchar(50)	<input checked="" type="checkbox"/>

Table Properties (General):

- (Name): Customer_Profile_Key
- Allow Nulls: No
- Data Type: int
- Default Value or Binding: <database default>

Table Designer Properties:

- Collation: <database default>
- Computed Column Specification: None
- Condensed Data Type: None
- Description: None
- Deterministic: No
- DTS-published: No
- Full-text Specification: Has Non-SQL Server Subscriber
- Identity Specification:
 - (Is Identity): Yes
 - Identity Increment: 1
 - Identity Seed: 1
 - Indexable: Yes
 - Is Columnset: No
 - Is Sparse: No
 - Merge-published: No
 - Not For Replication: No
 - Replicated: No
 - RowGuid: No
 - Size: 4

Queries for customer dimensions:

```
/* creating the customer 1st dimension */
SELECT DISTINCT
    [CUSTOMER_ID],
    [AGE],
    [CITY], [STATE],
    [MARITAL_STATUS]
    , [SOCIAL_CLASS],
    [CUSTOMER_EDUCATION_LEVEL],
    [EMPLOYMENT_STATUS]
INTO Customer_Dim1
FROM AA_Staging_Table
```

```
/* populating the customer dimension, we use the UI to create the customer
location key */
SELECT DISTINCT
    [CITY],
```

```

[STATE]
INTO Customer_Dim_Location
FROM [dbo].[Customer_Dim]

/* populating the customer profile dimension, we use the UI to create the
customer key */
SELECT DISTINCT MARITAL_STATUS,
    SOCIAL_CLASS,
    CUSTOMER_EDUCATION_LEVEL,
    EMPLOYMENT_STATUS
INTO Customer_Profile_Dim
FROM [master].[dbo].[Customer_Dim]

/* populating the main customer dimension */
SELECT DISTINCT
    Customer_Key,
    Customer_Profile_Key,
    Customer_ID,
    Age,
    C_Location_Key
INTO Main_Customer_Dim
FROM [master].[dbo].[Customer_Dim_Location] CDL
JOIN [master].[dbo].[Customer_Dim] CD ON CDL.CITY = CD.CITY AND
CDL.STATE = CD.STATE
JOIN [dbo].[Customer_Profile_Dim] CPD
ON CPD.[MARITAL_STATUS] = CD.
    [MARITAL_STATUS] AND CPD.SOCIAL_CLASS =
    CD.SOCIAL_CLASS
    AND CPD.CUSTOMER_EDUCATION_LEVEL =
    CD.CUSTOMER_EDUCATION_LEVEL
    AND CPD.EMPLOYMENT_STATUS = CD.EMPLOYMENT_STATUS

/* deleting the 1st customer dimension */
DROP TABLE Customer_Dim

```

Creating the Risk Segmentation dimension:

The screenshot shows the SQL Server Management Studio (SSMS) interface. On the left, there's a table definition for the 'dbo.Risk_Dimension' table with two columns: 'Risk_Key' (int, primary key, not null) and 'RISK_SEGMENTATION' (nvarchar(10), null). On the right, an 'Indexes/Keys' dialog is open, showing a primary key named 'PK_Risk_Dimension' for the 'Risk_Key' column. The 'General' properties for this key are set to 'Risk_Key (ASC)', 'Is Unique: Yes', and 'Type: Index'. The 'Identity' properties show 'Name: IX_Risk_Dimension', 'Description: ', and 'Create As Clustered: No'. The 'Table Designer' section shows various properties like Collation, Computed Column Specification, Full-text Specification, and Identity Specification.

```
SELECT DISTINCT
    [RISK_SEGMENTATION]
INTO Risk_Dimension
FROM [dbo].[AA_Staging_Table]
```

Incident report dimension

The screenshot shows the 'Incident_Report_Dim' table structure in SQL Server Management Studio. The table has the following columns:

Column Name	Data Type	Allow Nulls
Incident_Report_Key	int	<input type="checkbox"/>
I_Location_Key	int	<input type="checkbox"/>
INCIDENT_HOUR_OF_THE_DAY	int	<input checked="" type="checkbox"/>
CLAIM_STATUS	nvarchar(50)	<input checked="" type="checkbox"/>
INCIDENT_SEVERITY	nvarchar(50)	<input checked="" type="checkbox"/>
AUTHORITY_CONTACTED	nvarchar(50)	<input checked="" type="checkbox"/>
ANY_INJURY	int	<input checked="" type="checkbox"/>
POLICE_REPORT_AVAILABILITY	int	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Below the table structure, the 'Column Properties' pane is open for the 'Incident_Report_Key' column. The properties are as follows:

Property	Value
Data Type	int
Default Value or Binding	
Table Designer	
Collation	<database default>
Computed Column Specification	
Condensed Data Type	int
Description	
Deterministic	Yes
DTS-published	No
Full-text Specification	
Has Non-SQL Server Subscriber	No
Identity Specification	
(Is Identity)	Yes
Identity Increment	1
Identity Seed	1
Indexable	Yes
Is Columnset	No
Is Sparse	No
Merge-published	No
Not For Replication	No
Replicated	No
RowGuid	No
Size	4
Identity Specification	

Incident location dimension

The screenshot shows the 'Ident_Location_Dim' table structure in SQL Server Management Studio. The table has the following columns:

Column Name	Data Type	Allow Nulls
I_Location_Key	int	<input type="checkbox"/>
INCIDENT_CITY	nvarchar(50)	<input checked="" type="checkbox"/>
INCIDENT_STATE	nvarchar(50)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Below the table structure, the 'Column Properties' pane is open for the 'I_Location_Key' column. The properties are as follows:

Property	Value
(General)	
(Name)	I_Location_Key
Allow Nulls	No
Data Type	int
Default Value or Binding	
Table Designer	
Collation	<database default>
Computed Column Specification	
Condensed Data Type	int
Description	
Deterministic	Yes
DTS-published	No
Full-text Specification	
Has Non-SQL Server Subscriber	No
Identity Specification	
(Is Identity)	Yes
Identity Increment	1
Identity Seed	1
Indexable	Yes
Is Columnset	No
Is Sparse	No
Merge-published	No
Not For Replication	No
Replicated	No
RowGuid	No
Size	4
(Is Identity)	

Queries for incident report dimensions:

```
/* Creating the first incident report dimension */
SELECT DISTINCT
    [INCIDENT_HOUR_OF_THE_DAY],
    [CLAIM_STATUS],
    [INCIDENT_SEVERITY],
    [AUTHORITY_CONTACTED],
    [ANY_INJURY],
    [POLICE_REPORT_AVAILABLE],
    [INCIDENT_CITY],
    [INCIDENT_STATE]
INTO Incident_Report_Dim111111111
FROM AA_Staging_Table

/* creating the incident report dimension, we use the UI to create the incident
location key */
SELECT DISTINCT
    [INCIDENT_CITY],
    [INCIDENT_STATE]
INTO Incident_Location_Dim
FROM [dbo].[Incident_Report_Dim111111111]

/* creting the main report dimension */
SELECT DISTINCT
    I_Location_Key
    ,[INCIDENT_HOUR_OF_THE_DAY]
    ,[CLAIM_STATUS]
    ,[INCIDENT_SEVERITY]
    ,[AUTHORITY_CONTACTED]
    ,[ANY_INJURY]
    ,[POLICE_REPORT_AVAILABLE]
INTO Main_Incident_Report_Dim
FROM Incident_Location_Dim ILD
JOIN Incident_Report_Dim111111111 IRD
ON ILD.[INCIDENT_CITY] =IRD.[INCIDENT_CITY]
AND ILD.[INCIDENT_STATE] = IRD.[INCIDENT_STATE]
```

```
/* deleting the first incident report dimension, we dont need it anymore*/
DROP TABLE [dbo].[Incident_Report_Dim11111111]
```

Agent-employee Dimension

The screenshot shows the SSMS interface with the following details:

- Table Designer:** The main window displays the structure of the `in_Agent_Dimension` table. It has three columns:
 - Agent_Key**: Data type int, Allow Nulls checked.
 - Agent_ID**: Data type nvarchar(50), Allow Nulls checked.
 - A_Location_Key**: Data type int, Allow Nulls checked.
- Column Properties:** A sidebar panel provides detailed properties for each column, such as `Name`, `Data Type`, `Allow Nulls`, and `Default Value or Binding`.
- Indexes/Keys Dialog:** A modal dialog is open, titled "Selected Primary/Unique Key or Index". It lists two keys:
 - IX_Main_Agent_Dimension***
 - PK_Main_Agent_Dimension**
 The dialog is set to edit the **PK_Main_Agent_Dimension** key, which is defined as a primary key (Type: Identity, Name: IX_Main_Agent_Dimension). Other properties shown include `Description`, `Table Designer` settings (Create As Clustered: No, Data Space Specification: PRIMARY, Fill Specification: Yes, Included Columns: Yes, Is Disabled: No), and `Index` settings (Name: IX_Main_Agent_Dimension, Description: PRIMARY).

Agent-employee location dimension

The screenshot shows the SQL Server Management Studio (SSMS) interface. A new table named 'Agent_Employee_Location_Dimension' is being created in the 'master' database. The table has three columns: 'A_Location_Key' (int, primary key, not null), 'Employee_City' (nvarchar(25)), and 'Employee_State' (nvarchar(25)). The 'Table Designer' properties pane is open, showing detailed settings for each column. The 'General' tab is selected, displaying properties like Name, Data Type, Allow Nulls, and Default Value or Binding.

Column Name	Data Type	Allow Nulls
A_Location_Key	int	<input checked="" type="checkbox"/>
Employee_City	nvarchar(25)	<input checked="" type="checkbox"/>
Employee_State	nvarchar(25)	<input checked="" type="checkbox"/>

Column Properties	
(Name)	A_Location_Key
Allow Nulls	No
Data Type	int
Default Value or Binding	
Table Designer	
Collation	<database default>
Computed Column Specification	
Condensed Data Type	
Description	
Deterministic	Yes
DTS-published	No
Full-text Specification	No
Has Non-SQL Server Subscriber	No
Identity Specification	Yes
Indexable	Yes
Is Columnset	No
Is Sparse	No
Merge-published	No
Not For Replication	No
Replicated	No
RowGuid	No
Size	4

Queries for agent-employee dimensions:

```
/* Creating the first agent(employee) dimension */
SELECT DISTINCT
    [AGENT_ID],
    [Employee_City],
    [Employee_State]
INTO Agent_Dimension_111111111
FROM [dbo].[AA_Staging_Table]

/* Populating the agent(employee) location dimension, i use the UI to create the
A_Location_Key */
SELECT DISTINCT
    [Employee_City],
    [Employee_State]
INTO Agent_Location_Dimension
FROM [master].[dbo].[Agent_Dimension_111111111]

/* Populating the main agent(employee) dimension */
SELECT DISTINCT
    Agent_Key,
```

```

    Agent_ID,
    A_Location_Key
INTO Main_Agent_Dimension
FROM Agent_Dimension_111111111 AD
JOIN Agent_Location_Dimension ADL ON AD.[Employee_City] = ADL.
[Employee_City] AND AD.[Employee_State] = ADL.[Employee_State]

/* deleting the first agent(employee) dimension, we do not need it anymore */
DROP TABLE Agent_Dimension_111111111

```

we are making some changes to the date dimensions. I am adding the day, month and year columns:

```
alter table Transaction_Date_Dim add day_T int, month_T int, year_T int
```

```
update Transaction_Date_Dim
```

```
    set day_T = day(Transaction_Date), month_T =
MONTH(Transaction_Date), year_T = year(Transaction_Date)
```

```
alter table Report_Date_dim add day_R int, month_R int, year_R int
```

```
update Report_Date_dim
```

```
    set day_R = day(Reported_Date), month_R = MONTH(Reported_Date),
year_R = year(Reported_Date)
```

```
alter table Policy_Date_Dim add day_P int, month_P int, year_P int
```

```
update Policy_Date_Dim
```

```
    set day_P = day(Policy_Eff_Date), month_P = MONTH(Policy_Eff_Date),
year_P = year(Policy_Eff_Date)
```

```
alter table Loss_Date_Dim add day_L int, month_L int, year_L int
```

```
update Loss_Date_Dim
```

```
    set day_L = day(Loss_Date), month_L = MONTH(Loss_Date), year_L =
year(Loss_Date)
```

Finally we create the Fact Table

Queries for the fact table

```

SELECT Time_Key,
       Customer_Key,
       CPD.Customer_Profile_Key,
       Agent_Key,
       Vendor_Key,
       Transaction_Key,
       Policy_Key,
       Incident_Report_Key,
       CLAIM_AMOUNT,
       PREMIUM_AMOUNT
INTO Fact_Table
FROM Main_Date_Dim MDD
JOIN Transaction_Date_Dim TDD ON MDD.T_Date_Key = TDD.T_Date_Key
JOIN Policy_Date_Dim PDD ON MDD.P_Date_Key = PDD.P_Date_Key
JOIN Loss_Date_Dim LDD ON MDD.Loss_Date_Key = LDD.Loss_Date_Key
JOIN Report_Date_dim RDD ON MDD.R_Date_Key = RDD.R_Date_Key
    
```

```
JOIN AA_Staging_Table ST
ON ST.TXN_DATE_TIME =
TDD.Transaction_Date AND ST.POLICY_EFF_DT
= PDD.Policy_Eff_Date AND ST.LOSS_DT =
LDD.Loss_Date
AND ST.REPORT_DT = RDD.Reported_Date

JOIN Risk_Dimension rd ON rd.RISK_SEGMENTATION =
ST.RISK_SEGMENTATION

JOIN Policy_Dim PD
ON ST.POLICY_NUMBER = PD.Policy_Number
AND ST.TENURE = PD.Tenure
AND ST.INSURANCE_TYPE = PD.Insurance_Type

JOIN Transaction_dim TD
ON TD.Transaction_ID = ST.TRANSACTION_ID

JOIN Main_Incident_Report_Dim MIRD
ON MIRD.INCIDENT_HOUR_OF_THE_DAY =
ST.INCIDENT_HOUR_OF_THE_DAY
AND MIRD.CLAIM_STATUS = ST.CLAIM_STATUS
AND MIRD.INCIDENT_SEVERITY = ST.INCIDENT_SEVERITY
AND MIRD.AUTHORITY_CONTACTED =
ST.AUTHORITY_CONTACTED AND MIRD.ANY_INJURY =
ST.ANY_INJURY
AND MIRD.POLICE_REPORT_AVAILABLE
= ST.POLICE_REPORT_AVAILABLE

JOIN Incident_Location_Dim ILD
ON ILD.I_Location_Key = MIRD.I_Location_Key
AND ILD.INCIDENT_CITY = ST.INCIDENT_CITY
AND ILD.INCIDENT_STATE =
ST.INCIDENT_STATE

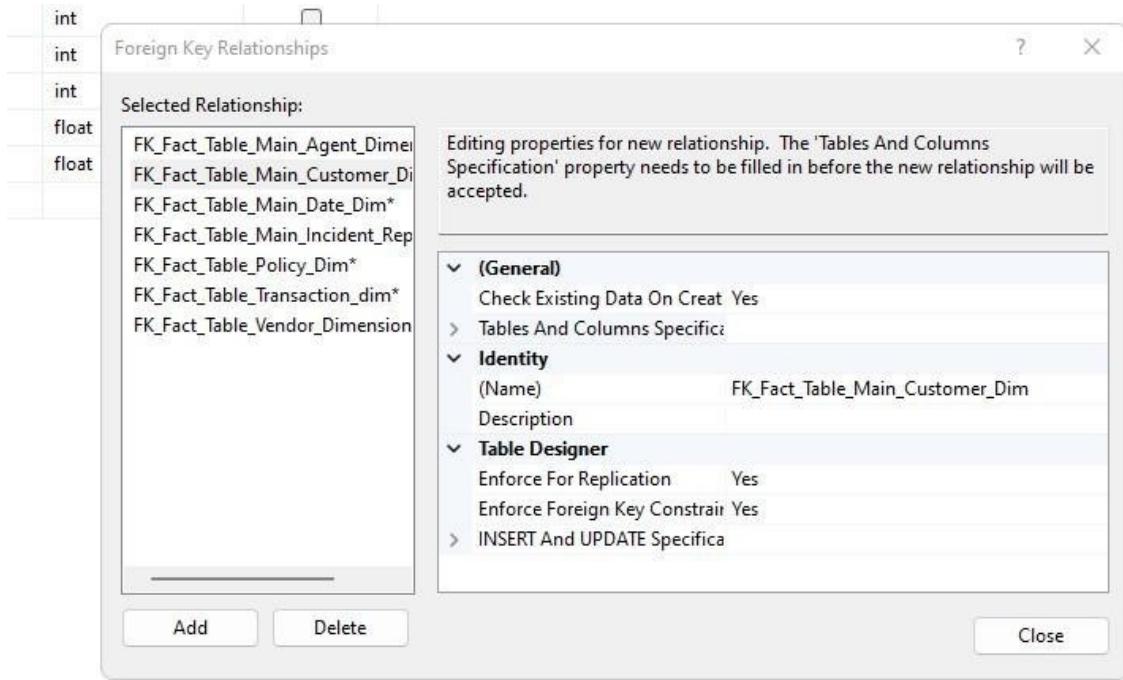
JOIN Vendor_Dimension VD ON ST.VENDOR_ID =
VD.Vendor_ID JOIN Vendor_Location_Dimension VLD
ON VLD.Vendor_Location_Key = VD.Vendor_Location_Key
```

```
AND VLD.Vendor_City = ST.Vendor_City  
AND VLD.Vendor_State = ST.Vendor_State  
  
JOIN Main_Agent_Dimension MAD ON MAD.Agent_ID = ST.AGENT_ID  
  
JOIN Agent_Location_Dimension ALD  
ON ALD.A_Location_Key = MAD.A_Location_Key  
AND ALD.Employee_City = ST.Employee_City  
AND ALD.Employee_City = ST.Employee_City  
  
JOIN Customer_Profile_Dim CPD  
ON CPD.MARITAL_STATUS =  
ST.MARITAL_STATUS AND CPD.SOCIAL_CLASS =  
ST.SOCIAL_CLASS AND  
CPD.CUSTOMER_EDUCATION_LEVEL =  
ST.CUSTOMER_EDUCATION_LEVEL  
AND CPD.EMPLOYMENT_STATUS = ST.EMPLOYMENT_STATUS  
  
JOIN Main_Customer_Dim MCD  
ON MCD.Customer_ID = ST.CUSTOMER_ID  
AND MCD.Age = ST.AGE
```

Here we see all the tables that we have

System Databases	
master	
Tables	
System Tables	
External Tables	
Graph Tables	
dbo-AA_Staging_Table	
dbo.AB_employee_data	
dbo.AB_insurance_data	
dbo.AB_vendor_data	
dbo.Agent_Location_Dimension	
dbo.Customer_Dim_Location	
dbo.Customer_Profile_Dim	
dbo.Fact_Table	
dbo.Incident_Location_Dim	
dbo.Loss_Date_Dim	
dbo.Main_Agent_Dimension	
dbo.Main_Customer_Dim	
dbo.Main_Date_Dim	
dbo.Main_Incident_Report_Dim	
dbo.Policy_Date_Dim	
dbo.Policy_Dim	
dbo.Report_Date_dim	
dbo.Transaction_Date_Dim	
dbo.Transaction_dim	
dbo.Vendor_Dimension	
dbo.Vendor_Location_Dimension	
Views	
Synonyms	
Programmability	
Service Broker	

Finally we set the necessary relationships with the foreign key and primary key

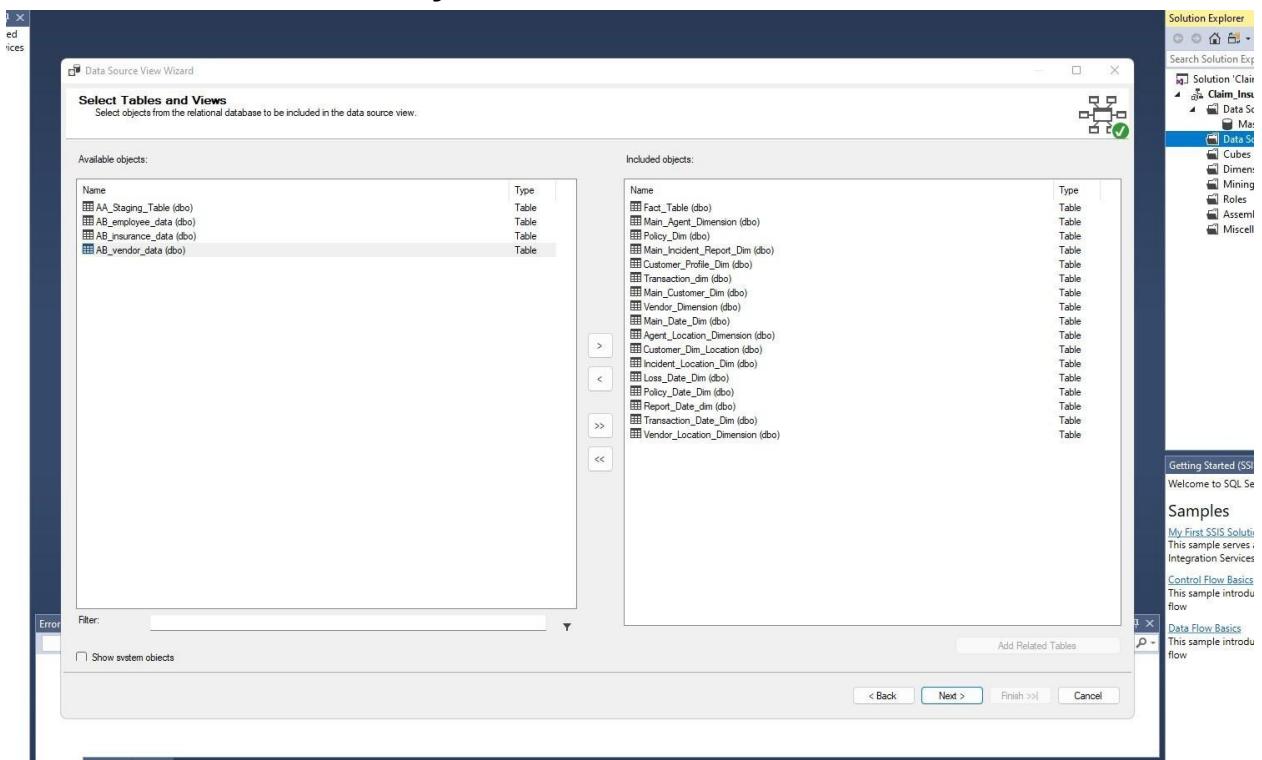


We also set the relationship for the date dimensions with the Main dimension and for the city-state dimensions with their regarding dimension tables

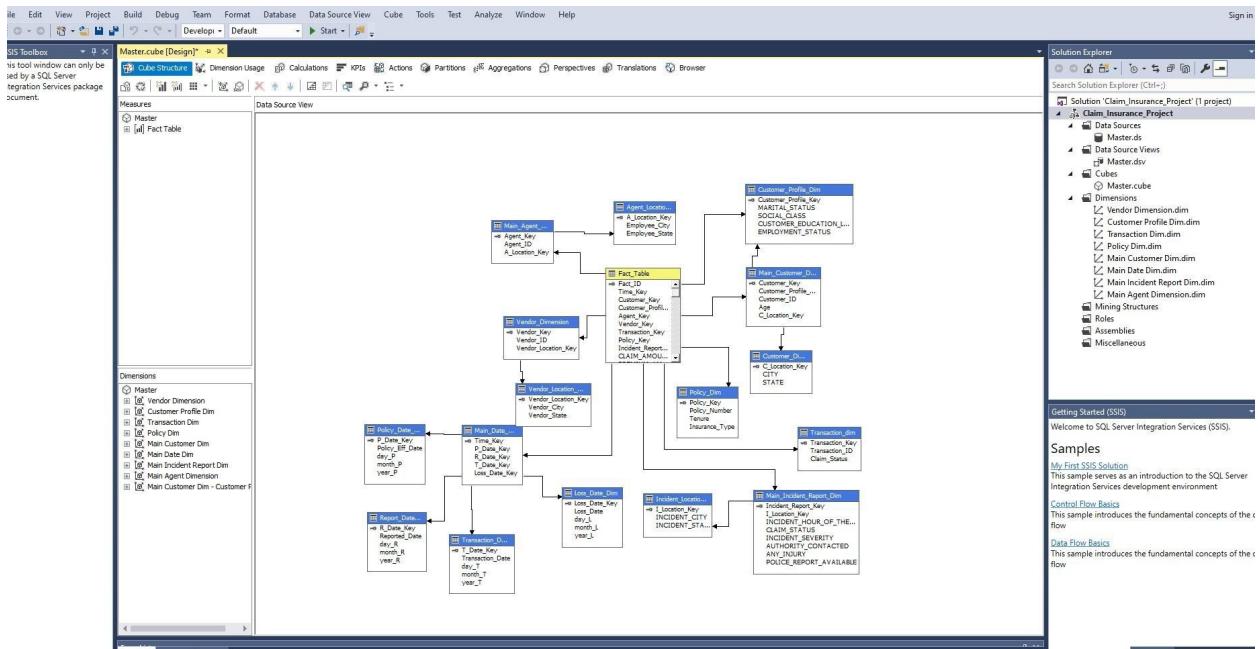
Section 4, SQL Server Analysis

Now that the data warehouse has been created and populated, the SSAS is used for the creation of the cubes. First a new project is created and later connected to the database using the windows credentials. The fact table and the dimension tables are selected. Also, two different hierarchies are created. State-City and Year-Month-Day. Finally the attributes are taken from the small dimensions to the main, in order to make meaningful aggregations.

We select all the necessary tables:

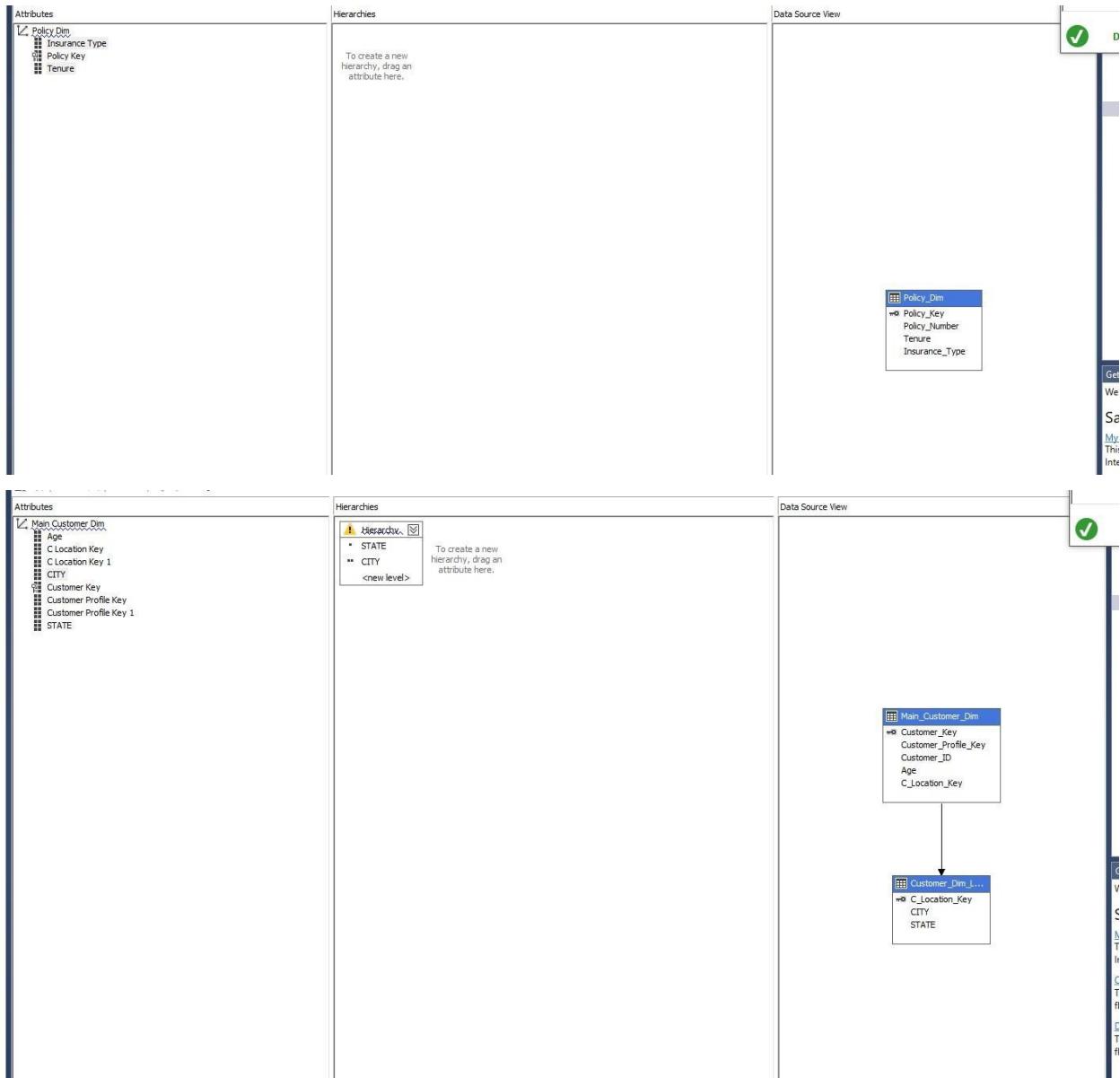


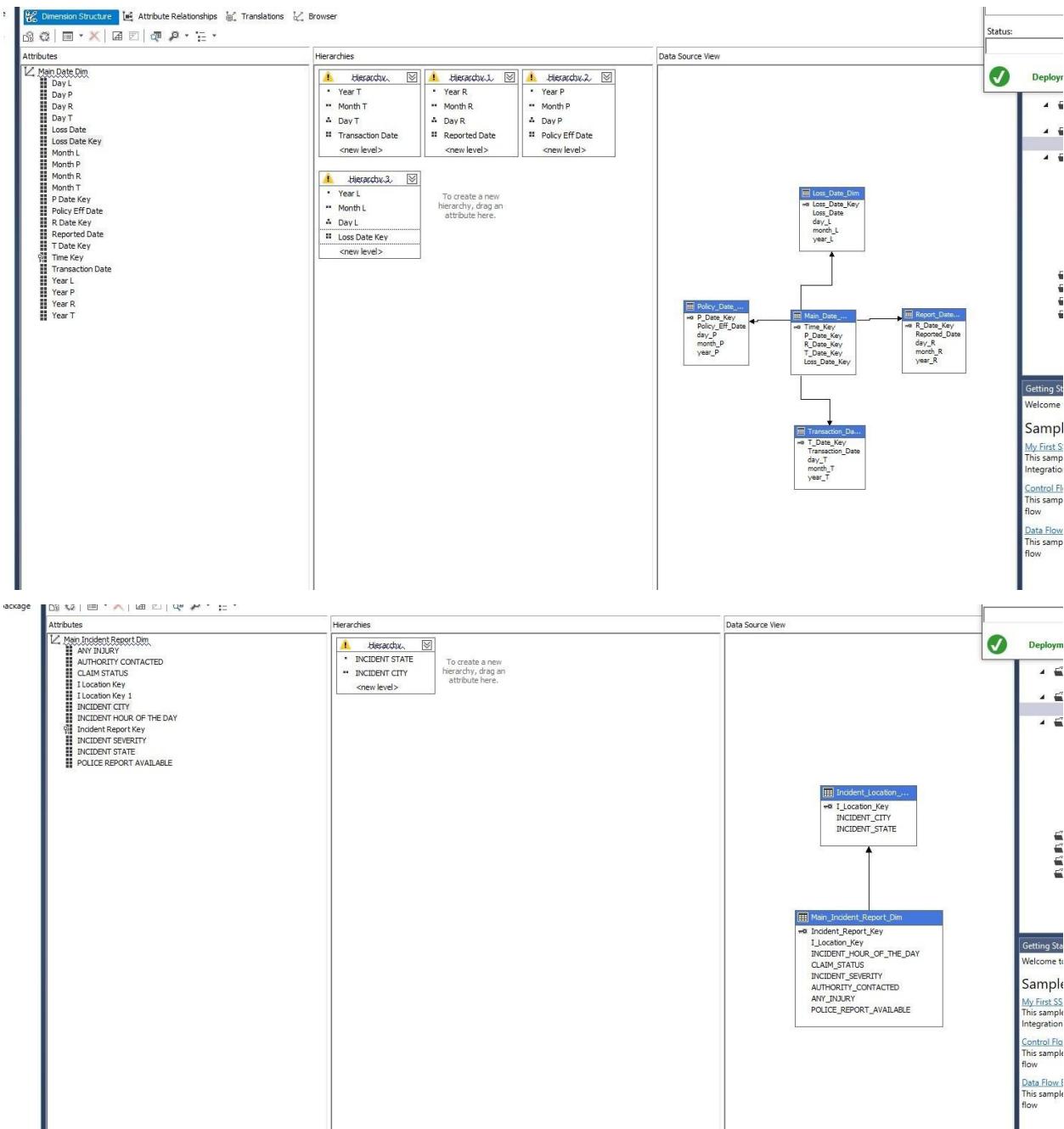
And we create the following schema:

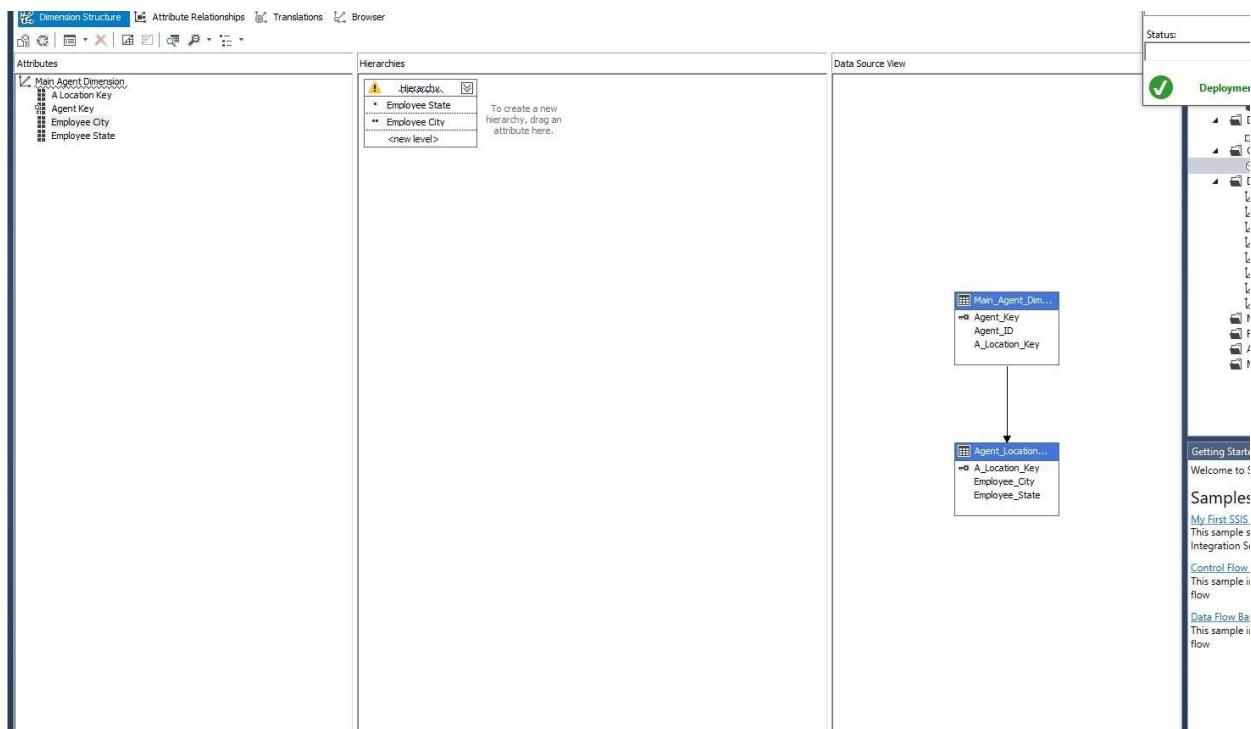


Finally we create the hierarchies that we want. We also, bring the descriptions to have more aggregation capabilities and more meaningful dimensions:

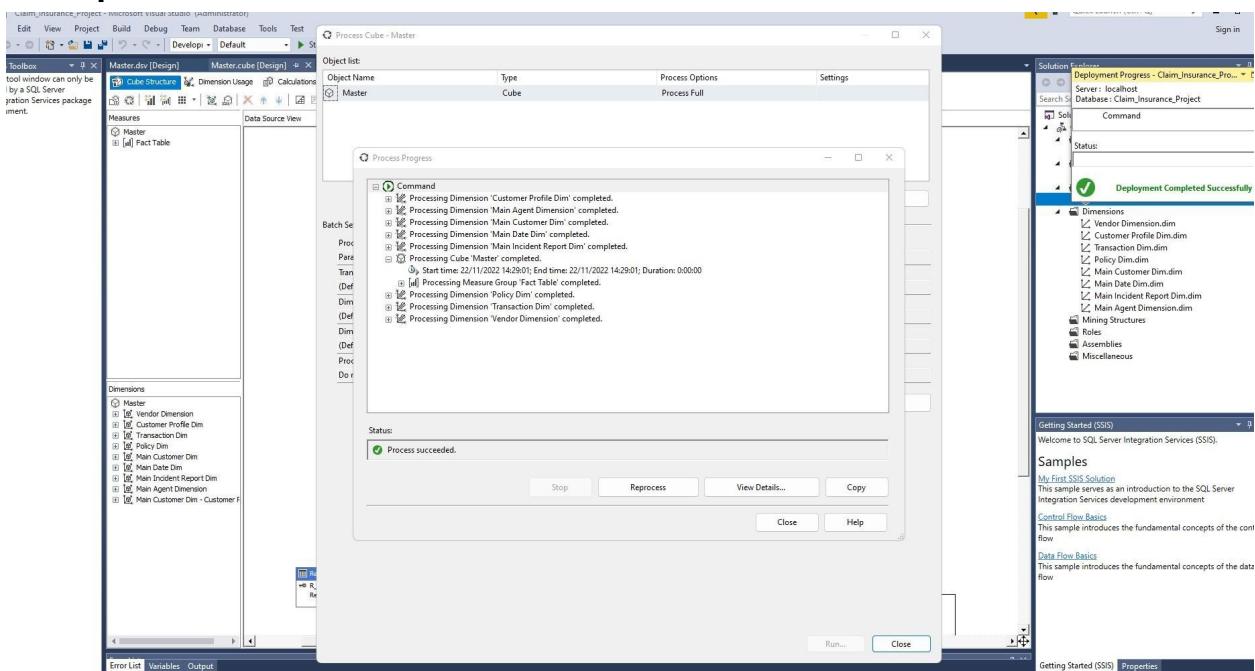
<p>Attributes</p> <ul style="list-style-type: none"><input checked="" type="checkbox"/> Customer_Profile_Dim<ul style="list-style-type: none">Customer Profile KeyCUSTOMER EDUCATION LEVELEMPLOYMENT STATUSMARITAL STATUSSOCIAL CLASS	<p>Hierarchies</p> <p>To create a new hierarchy, drag an attribute here.</p>	<p>Data Source View</p> <table border="1"><tr><td>Customer_Profile_Dim</td></tr><tr><td>Customer_Profile_Key</td></tr><tr><td>MARITAL_STATUS</td></tr><tr><td>SOCIAL_CLASS</td></tr><tr><td>CUSTOMER_EDUCATION_LEVEL</td></tr><tr><td>EMPLOYMENT_STATUS</td></tr></table>	Customer_Profile_Dim	Customer_Profile_Key	MARITAL_STATUS	SOCIAL_CLASS	CUSTOMER_EDUCATION_LEVEL	EMPLOYMENT_STATUS
Customer_Profile_Dim								
Customer_Profile_Key								
MARITAL_STATUS								
SOCIAL_CLASS								
CUSTOMER_EDUCATION_LEVEL								
EMPLOYMENT_STATUS								
<p>Attributes</p> <ul style="list-style-type: none"><input checked="" type="checkbox"/> Transaction_Dim<ul style="list-style-type: none">Claim StatusTransaction Key	<p>Hierarchies</p> <p>To create a new hierarchy, drag an attribute here.</p>	<p>Data Source View</p> <table border="1"><tr><td>Transaction_dim</td></tr><tr><td>Transaction_Key</td></tr><tr><td>Transaction_ID</td></tr><tr><td>Claim_Status</td></tr></table>	Transaction_dim	Transaction_Key	Transaction_ID	Claim_Status		
Transaction_dim								
Transaction_Key								
Transaction_ID								
Claim_Status								







We reprocess the Cube



Browsing some of the dimensions-measurements:

The screenshot shows the Analysis Services Management Studio interface with the following details:

- Left pane:** The 'Dimensions' node is selected in the tree view under the 'Master' cube.
- Right pane:** The 'Dimensions' node is expanded, showing its child dimensions: Customer Profile Dim, Main Agent Dimension, Main Customer Dim, Main Date Dim, Main Incident Report Dim, Policy Dim, Transaction Dim, and Vendor Dimension.

In this screenshot a small OLAP report was made as a preview. The Tableau will be used as the main reporting tool.

The screenshot shows a user interface for an OLAP report. At the top, there is a navigation bar with icons for Calculations, KPIs, Actions, Partitions, Aggregations, Perspectives, Translations, and a Browser button. Below the navigation bar is a toolbar with various icons for report management. The main area contains a table with three columns: Insurance Type, ANY INJURY, and CLAIM AMOUNT. The table data is as follows:

Insurance Type	ANY INJURY	CLAIM AMOUNT
Health	0	5172000
Health	1	13082000
Life	0	26942000
Life	1	64536000
Mobile	0	198300
Mobile	1	490000
Motor	0	2775000
Motor	1	5888000
Property	0	12023000
Property	1	29556000
Travel	0	1547000
Travel	1	3429000

Section 5, Connecting to Tableau

Now that the cube has been created, a reporting software can be connected to the data warehouse or directly to the cube. In this case, Tableau is used to connect to the data warehouse and any necessary changes, like creating the hierarchies will be made in Tableau. Tableau gives three actual ways to join the tables of the data warehouse. Merging, joining and using a custom SQL query. For our convenience only we will use custom SQL query and we will hide any columns that wont be used in the Report.

Connecting to the cube:

The screenshot shows the Microsoft Analysis Services Data Source configuration interface. It consists of two main sections: 'Step 1: Select a Database' on the left and 'Step 2: Select a Cube:' on the right.

Step 1: Select a Database

Connected to Microsoft Analysis Services .

Enter search text

Name
Claim_Insurance_Project
DemoMulti
Exercise_22
MultidimensionalProject3
MultidimensionalProject4
MultidimensionalProject5
MultidimensionalProject6

Step 2: Select a Cube:

Enter search text

Name
Master

Fields

Type	Field Name	Physical Table	Remote Field Name
Customer Profile Dim	Customer Profile Dim		Customer Profile Dim
Main Agent Dimension	Main Agent Dimension		Main Agent Dimension
Main Customer Dim - Customer Profile	Main Customer Dim - Customer Profile		Main Customer Dim - Customer Profile
Main Date Dim	Main Date Dim		Main Date Dim
Main Incident Report Dim	Main Incident Report Dim		Main Incident Report Dim
Policy Dim	Policy Dim		Policy Dim
Transaction Dim	Transaction Dim		Transaction Dim
Vendor Dimension	Vendor Dimension		Vendor Dimension
CLAIM AMOUNT	CLAIM AMOUNT	Fact Table	CLAIM AMOUNT
PREMIUM AMOUNT	PREMIUM AMOUNT	Fact Table	PREMIUM AMOUNT
Fact Table Count	Fact Table Count	Fact Table	Fact Table Count

Data Source Status: 1

Custom SQL Query:

```

SELECT [Main_Incident_Report_Dim].[ANY_INJURY] AS [ANY_INJURY],
       [Main_Incident_Report_Dim].[AUTHORITY_CONTACTED] AS [AUTHORITY_CONTACTED],
       [Agent_Location_Dimension].[A_Location_Key] AS [A_Location_Key (Agent_Location_Dimension)],
       [Main_Customer_Dim].[Age] AS [Age],
       [Main_Agent_Dimension].[Agent_ID] AS [Agent_ID],
       [Main_Agent_Dimension].[Agent_Key] AS [Agent_Key (Main_Agent_Dimension)],
       [Fact_Table].[Agent_Key] AS [Agent_Key],
       [Customer_Dim_Location].[CITY] AS [CITY],
       [Fact_Table].[CLAIM_AMOUNT] AS [CLAIM_AMOUNT],
       [Main_Incident_Report_Dim].[CLAIM_STATUS] AS [CLAIM_STATUS1],
       [Customer_Profile_Dim].[CUSTOMER_EDUCATION_LEVEL] AS [CUSTOMER_EDUCATION_LEVEL],
       [Customer_Dim_Location].[C_Location_Key] AS [C_Location_Key (Customer_Dim_Location)],
       [Main_Customer_Dim].[C_Location_Key] AS [C_Location_Key],
       [Transaction_dim].[Claim_Status] AS [Claim_Status],
       [Main_Customer_Dim].[Customer_ID] AS [Customer_ID],
       [Main_Customer_Dim].[Customer_Key] AS [Customer_Key (Main_Customer_Dim)],
       [Fact_Table].[Customer_Key] AS [Customer_Key],
       [Customer_Profile_Dim].[Customer_Profile_Key] AS [Customer_Profile_Key (Customer_Profile_Dim)],
       [Main_Customer_Dim].[Customer_Profile_Key] AS [Customer_Profile_Key (Main_Customer_Dim)],
       [Fact_Table].[Customer_Profile_Key] AS [Customer_Profile_Key],
       [Customer_Profile_Dim].[EMPLOYMENT_STATUS] AS [EMPLOYMENT_STATUS],
       [Agent_Location_Dimension].[Employee_City] AS [Employee_City],
       [Agent_Location_Dimension].[Employee_State] AS [Employee_State],
       [Fact_Table].[Fact_ID] AS [Fact_ID],
       [Incident_Location_Dim].[INCIDENT_CITY] AS [INCIDENT_CITY],
       [Main_Incident_Report_Dim].[INCIDENT_HOUR_OF_THE_DAY] AS [INCIDENT_HOUR_OF_THE_DAY],
       [Main_Incident_Report_Dim].[INCIDENT_SEVERITY] AS [INCIDENT_SEVERITY],
       [Incident_Location_Dim].[INCIDENT_STATE] AS [INCIDENT_STATE],
       [Incident_Location_Dim].[I_Location_Key] AS [I_Location_Key (Incident_Location_Dim)],
       [Main_Incident_Report_Dim].[I_Location_Key] AS [I_Location_Key],
       [Main_Incident_Report_Dim].[Incident_Report_Key] AS [Incident_Report_Key (Main_Incident_Report_Dim)],
       [Fact_Table].[Incident_Report_Key] AS [Incident_Report_Key],
       [Policy_Dim].[Insurance_Type] AS [Insurance_Type],
       [Loss_Date_Dim].[Loss_Date] AS [Loss_Date],
       [Loss_Date_Dim].[Loss_Date_Key] AS [Loss_Date_Key (Loss_Date_Dim)],
       [Main_Date_Dim].[Loss_Date_Key] AS [Loss_Date_Key],
       [Customer_Profile_Dim].[MARITAL_STATUS] AS [MARITAL_STATUS],
       [Main_Incident_Report_Dim].[POLICE_REPORT_AVAILABLE] AS [POLICE_REPORT_AVAILABLE],
       [Fact_Table].[PREMIUM_AMOUNT] AS [PREMIUM_AMOUNT],
       [Policy_Date_Dim].[P_Date_Key] AS [P_Date_Key (Policy_Date_Dim)],
       [Main_Date_Dim].[P_Date_Key] AS [P_Date_Key],
       [Policy_Date_Dim].[Policy_Eff_Date] AS [Policy_Eff_Date],
       [Policy_Dim].[Policy_Key] AS [Policy_Key (Policy_Dim)],
       [Fact_Table].[Policy_Key] AS [Policy_Key],
       [Policy_Dim].[Policy_Number] AS [Policy_Number],
       [Report_Date_dim].[R_Date_Key] AS [R_Date_Key (Report_Date_dim)],
       [Main_Date_Dim].[R_Date_Key] AS [R_Date_Key],
       [Report_Date_dim].[Reported_Date] AS [Reported_Date],
       [Customer_Profile_Dim].[SOCIAL_CLASS] AS [SOCIAL_CLASS],
       [Customer_Dim_Location].[STATE] AS [STATE],
       [Transaction_Date_Dim].[T_Date_Key] AS [T_Date_Key (Transaction_Date_Dim)],
       [Main_Date_Dim].[T_Date_Key] AS [T_Date_Key],
       [Preview Results...] [Insert Parameter ▾]

```

Previews the results of the query.

```

[Customer_Dim_Location].[STATE] AS [STATE],
[Transaction_Date_Dim].[T_Date_Key] AS [T_Date_Key (Transaction_Date_Dim)],
[Main_Date_Dim].[T_Date_Key] AS [T_Date_Key],
[Policy_Dim].[Tenure] AS [Tenure],
[Main_Date_Dim].[Time_Key] AS [Time_Key (Main_Date_Dim)],
[Fact_Table].[Time_Key] AS [Time_Key],
[Transaction_Date_Dim].[Transaction_Date] AS [Transaction_Date],
[Transaction_dim].[Transaction_ID] AS [Transaction_ID],
[Fact_Table].[Transaction_Key] AS [Transaction_Key (Transaction_dim)],
[Vendor_Dim].[Vendor_Dim] AS [Vendor_City] AS [Vendor_City],
[Vendor_Dim].[Vendor_ID] AS [Vendor_ID],
[Vendor_Dim].[Vendor_Key] AS [Vendor_Key (Vendor_Dimension)],
[Fact_Table].[Vendor_Key] AS [Vendor_Key],
[Vendor_Location_Dimension].[Vendor_Location_Key] AS [Vendor_Location_Key (Vendor_Location_Dimension)],
[Vendor_Dimension].[Vendor_Location_Key] AS [Vendor_Location_Key],
[Vendor_Location_Dimension].[Vendor_State] AS [Vendor_State],
[Vendor_Location_Dimension].[Vendor_Location_Key] AS [Vendor_Location_Key],
[Loss_Date_Dim].[day_L] AS [day_L],
[Policy_Date_Dim].[day_F] AS [day_F],
[Report_Date_dim].[day_R] AS [day_R],
[Report_Date_dim].[day_M] AS [day_M],
[Loss_Date_Dim].[month_L] AS [month_L],
[Loss_Date_Dim].[month_F] AS [month_F],
[Policy_Date_Dim].[month_B] AS [month_B],
[Report_Date_dim].[month_B] AS [month_B],
[Transaction_Date_Dim].[month_T] AS [month_T],
[Loss_Date_Dim].[year_L] AS [year_L],
[Policy_Date_Dim].[year_F] AS [year_F],
[Report_Date_dim].[year_R] AS [year_R],
[Transaction_Date_Dim].[year_T] AS [year_T]
FROM [dbo].[Fact_Table] [Fact_Table]
INNER JOIN [dbo].[Main_Incident_Report_Dim] ON ([Fact_Table].[Time_Key] = [Main_Date_Dim].[Time_Key])
INNER JOIN [dbo].[Loss_Date_Dim] [Loss_Date_Dim] ON ([Main_Pats_Dim].[Loss_Date_Key] = [Loss_Date_Dim].[Loss_Date_Key])
INNER JOIN [dbo].[Policy_Date_Dim] [Policy_Date_Dim] ON ([Main_Date_Dim].[P_Date_Key] = [Policy_Date_Dim].[P_Date_Key])
INNER JOIN [dbo].[Report_Date_dim] [Report_Date_dim] ON ([Main_Date_Dim].[R_Date_Key] = [Report_Date_dim].[R_Date_Key])
INNER JOIN [dbo].[Transaction_Date_Dim] [Transaction_Date_Dim] ON ([Main_Date_Dim].[T_Date_Key] = [Transaction_Date_Dim].[T_Date_Key])
INNER JOIN [dbo].[Policy_Dim] [Policy_Dim] ON ([Fact_Table].[Policy_Key] = [Policy_Dim].[Policy_Key])
INNER JOIN [dbo].[Transaction_dim] [Transaction_dim] ON ([Fact_Table].[Transaction_Key] = [Transaction_dim].[Transaction_Key])
INNER JOIN [dbo].[Main_Incident_Report_Dim] [Main_Incident_Report_Dim] ON ([Fact_Table].[Incident_Report_Key] = [Main_Incident_Report_Dim].[Incident_Report_Key])
INNER JOIN [dbo].[Incident_Location_Dim] [Incident_Location_Dim] ON ([Main_Incident_Report_Dim].[I_Location_Key] = [Incident_Location_Dim].[I_Location_Key])
INNER JOIN [dbo].[Vendor_Dimension] [Vendor_Dimension] ON ([Fact_Table].[Vendor_Key] = [Vendor_Dimension].[Vendor_Key])
INNER JOIN [dbo].[Vendor_Location_Dimension] [Vendor_Location_Dimension] ON ([Vendor_Dimension].[Vendor_Location_Key] = [Vendor_Location_Dimension].[Vendor_Location_Key])
INNER JOIN [dbo].[Main_Agent_Dimension] [Main_Agent_Dimension] ON ([Fact_Table].[Agent_Key] = [Main_Agent_Dimension].[Agent_Key])
INNER JOIN [dbo].[Main_Customer_Dim] [Main_Customer_Dim] ON ([Fact_Table].[Customer_Key] = [Main_Customer_Dim].[Customer_Key])
INNER JOIN [dbo].[Customer_Profile_Dim] [Customer_Profile_Dim] ON ([Fact_Table].[Customer_Profile_Key] = [Customer_Profile_Dim].[Customer_Profile_Key])
INNER JOIN [dbo].[Customer_Dim_Location] [Customer_Dim_Location] ON ([Main_Customer_Dim].[C_Location_Key] = [Customer_Dim_Location].[C_Location_Key])

```

Previews the results of the query.

Preview of the Dimensions and measurements from Tableau

The screenshot shows the Tableau Data Catalog interface with the following details:

- Search Bar:** Data Warehousing using joins
- Folders:**
 - # Age
 - Abc Agent ID
 - # Any Injury
 - Abc Authority Contacted
 - Abc Claim Status
 - Abc Customer Education Level
 - Abc Customer ID
 - Abc Employment Status
 - # Incident Hour Of The Day
 - Abc Incident Severity
 - > INCIDENT_STATE, INCIDENT_CITY
 - Abc Insurance Type
 - Abc Loss Date
 - Abc Marital Status
 - # Police Report Available
 - Abc Policy Eff Date
 - Abc Policy Number
 - Abc Reported Date
 - Abc Social Class
 - > State Employee(Agent), City Employee(Agent)
 - > State Vendor, City Vendor
 - > State(Customer), City(Customer)
 - Abc Transaction Date
 - Abc Transaction ID
 - Abc Vendor ID
 - > year_L, month_L
 - > year_P, month_P
 - > year_R, month_R
 - > year_T, month_T
- Measure Names:**
 - # Claim Amount
 - # Premium Amount
 - # Tenure
 - # Fact_Table (Count)
 - ④ Latitude (generated)
 - ④ Longitude (generated)
 - # Measure Values

The right side of the interface shows the Tableau shelf with sections for Filters, Marks, Colour, and Detail.

Section 6, Analyzing the company

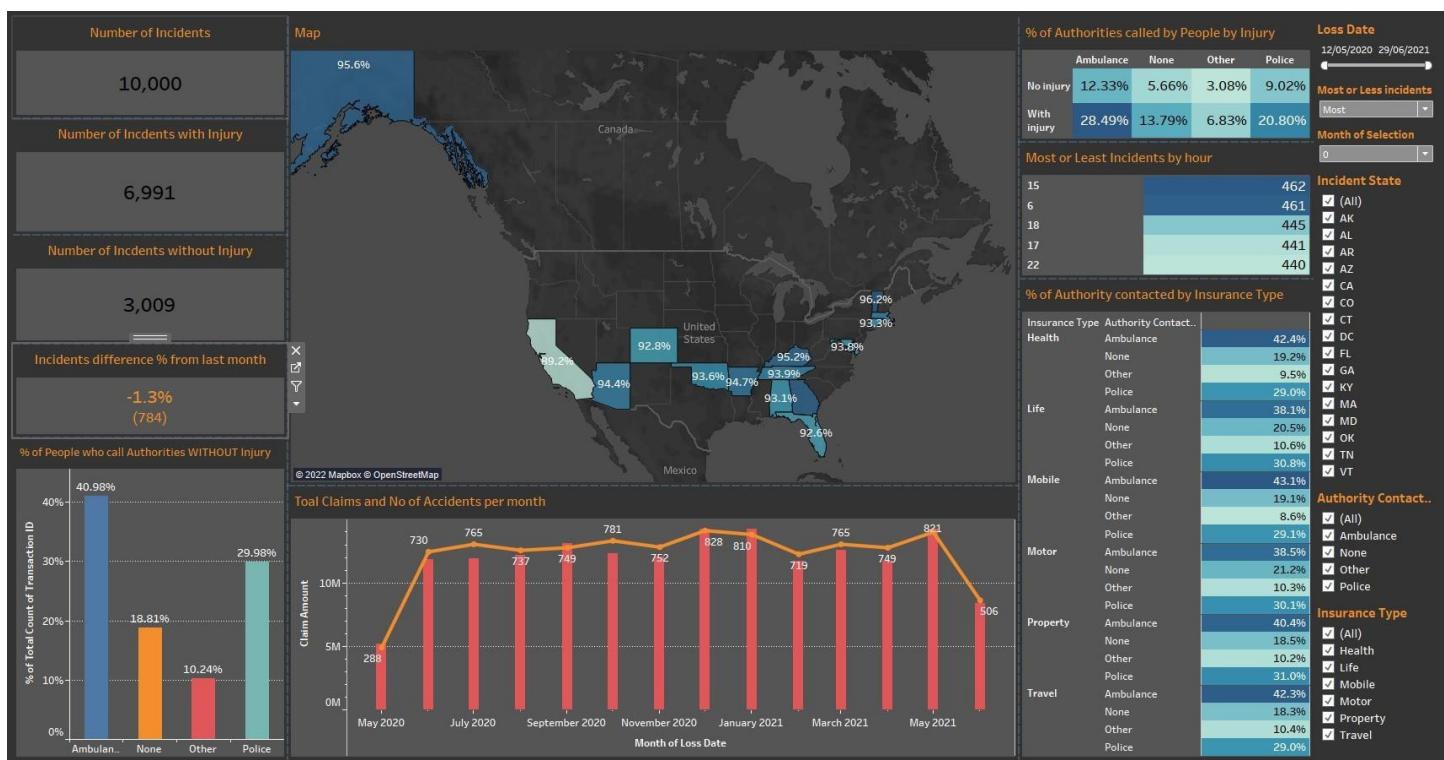
We are unable to make a Revenue Analysis, as our data concerns only the incidents themselves. Thus, trying to make any assumptions about the revenue would be incorrect, as we don't have the necessary data.

So, we will perform an Incident Analysis and a Claim Analysis.

In each segment we will present a screenshot of the dashboard and a list of the metrics used along with their perceived usefulness for the Insurance company.

A. Incident Analysis

This dashboard contains metrics regarding the incidents that have taken place.

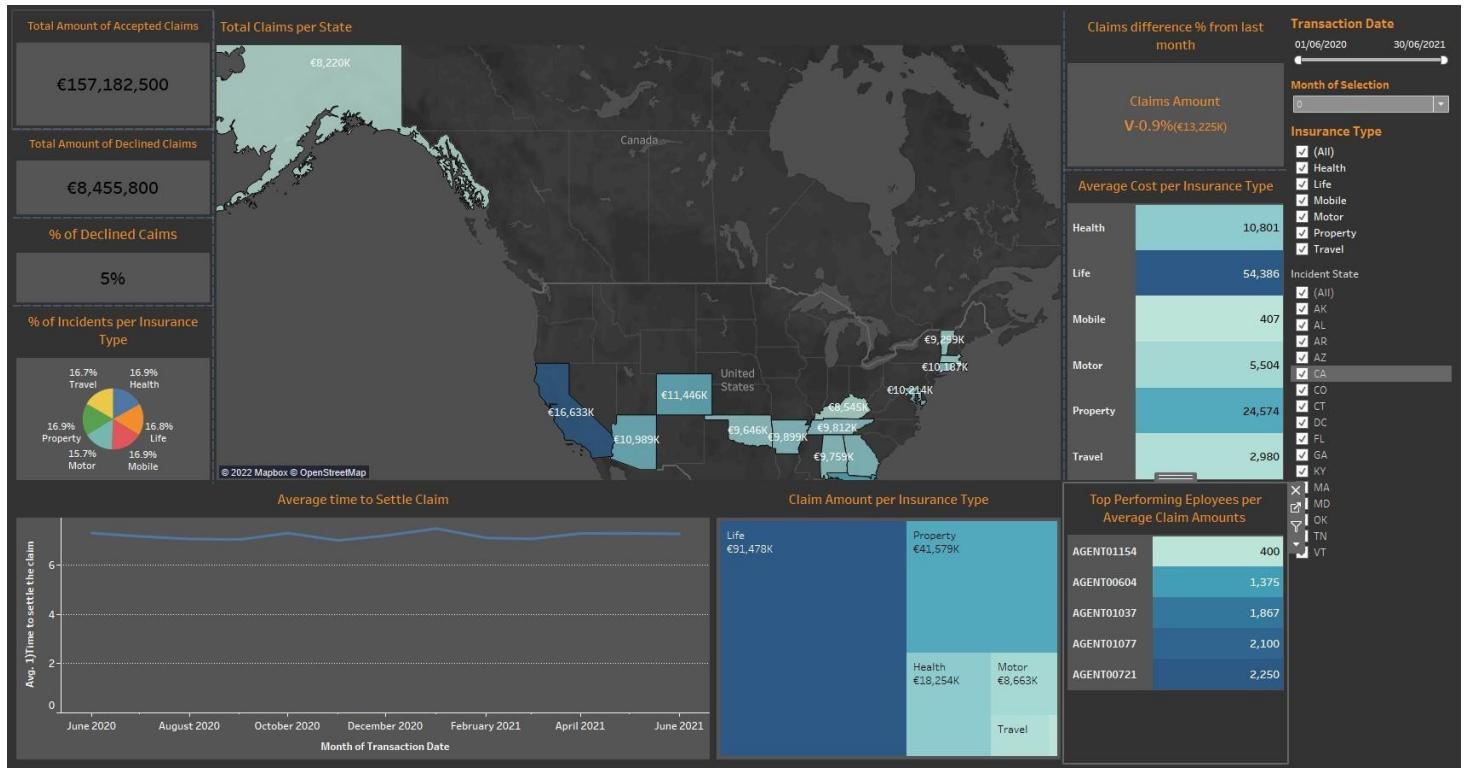


- **Number of Incidents:**
The total number of incidents that occurred in the selected time period
- **Number of Incidents with Injury:**
The total number of incidents where the involved person got injured, that occurred in the selected time period

- Number of Incidents without Injury:
The total number of incidents where the involved person did not get injured, that occurred in the selected time period
- Incidents difference % from last month:
The difference percentage of incidents from the month selected and the previous one
- % of People who call Authorities WITHOUT Injury:
The percentage of people who call the authorities (and the type of Authority) even though they have not sustained any injuries
- Total Claims and No of Accidents per month:
The total amount of claims and number of accidents per month for the selected time period
- % of Authorities called by People by Injury:
The percentages of authority types that were called with or without sustained injury by the involved persons
- Most or Least Accidents by hour:
The hours with most or least accidents occurring
- % of Authority contacted by Insurance Type:
The percentages of authority types called by the insurance type of the people
- Map:
A map portraying

B. Claim Analysis

This dashboard contains metrics regarding the claims made by the customers.



- **Total Amount of Accepted Claims:**
The total amount of accepted claims that were paid for in the selected time period
- **Total Amount of Declined Claims:**
The total amount of declined claims that were not paid for in the selected time period
- **% of Declined Claims:**
The percentage of claims that were declined against the total number of claims
- **% of Incidents per Insurance Type:**
The percentage of incidents of a specific insurance type against the total number of incidents
- **Average time to Settle Claim:**
The average time spent to settle the claims, per month
- **Claim amount per Insurance Type:**
The total claim amounts per insurance type
- **Claims difference % from last month:**

The difference percentage of claim amounts from the month selected and the previous one

- Average Cost per Insurance Type:

The average cost of claims paid by the insurance company, per insurance type

- Top Performing Employees per Average Claim Amounts:

The top performing employees whose cases generated the least average amount of claims to be paid

- Total Claims per State:

A map portraying the total amounts of claims made in each state independently