



Modern Data Management & Business Intelligence

Assignment #2: IOWA Liquor Sales



Despotis Spyridon: p2822111

Papailiou Thanasis: p2822128

Contents

1. Goals of Study & Dataset Exploration	3
1.1 Goals of the Study	3
1.2 Data Exploration.....	3
1.3 Data Preparation Methodology & Challenges	4
2. Designing our Data Flow	11
2.2 ETL Architecture	11
3. Fact Table & Dimension Tables.....	17
3.2 Create & Update Fact Table.....	27
4. SSAS Architecture	32
4.2 Calculated Members	34
4.3 Hierarchies.....	35
5. Visualizations with Power BI.....	40
5.1 Main Dashboard & Insights.....	40
5.2 Bottle Price per Category Dashboard	46
5.3 Bottle Price by County Dashboard	46
5.4 Sale Dollars By Liquor Category in Linn County Dashboard	47

1. Goals of Study & Dataset Exploration

1.1 Goals of the Study

In this assignment we are a Liquor Company and we want to expand our business activity in Iowa State in order to generate more profit.

Researching and understanding the new market is critical to success. So, our main goal is to analyse the total spirits sales from every competitive store that sells alcohol in bottled form for off-the-premises consumption.

Our guiding questions to our analysis are:

- What's the best geographic location to build our new premises
- Which are the top 5 stores by sales
- What are the top counties with sales
- How many categories of liquor do companies stock
- What is the most and least expensive category
- Which year had the most and lowest sales
- Is there a relationship between county sales and bottle price

1.2 Data Exploration

The “Iowa Liquid Sales” dataset contains transactions from all Iowa’s liquor stores holding a Class E liquor license. It is provided by the Iowa Department of Commerce, in which all state stores are registered. This dataset contains every wholesale purchase of liquor in the State of Iowa by retailers for sale to individuals since January 1, 2012. The State of Iowa controls the wholesale distribution of liquor intended for retail sale, which means this dataset offers a complete view of retail liquor sales in the entire state. The dataset contains every wholesale order of liquor by all grocery stores, liquor stores, convenience stores, etc., with details about the store and location, the exact liquor brand and size, and the number of bottles ordered.

We are going to explore the data from Iowa Retail Liquor Sales which is available to access on Google [BigQuery](#) public datasets.



Iowa Liquor Retail Sales

Iowa Department of Commerce

Liquor sales in Iowa since 2012, by store, by item, and by day

[VIEW DATASET](#)

iowa_liquor_sales

[CREATE TABLE](#) [SHARING](#) [COPY](#) [DELETE](#)

Dataset info

[EDIT DETAILS](#)

Dataset ID	bigquery-public-data:iowa_liquor_sales
Created	May 8, 2019, 10:20:57 PM UTC+3
Default table expiration	Never
Last modified	Oct 14, 2021, 9:39:05 PM UTC+3
Data location	US
Description	"This dataset contains every wholesale purchase of liquor in the State of Iowa by retailers for sale to individuals since January 1, 2012. The State of Iowa controls the wholesale distribution of liquor intended for retail sale (off-premises consumption), which means this dataset offers a complete view of retail liquor consumption in the entire state. The dataset contains every wholesale order of liquor by all grocery stores, liquor stores, convenience stores, etc., with details about the store and location, the exact liquor brand and size, and the number of bottles ordered. You can find more details, as well as sample queries, in the GCP Marketplace here: https://console.cloud.google.com/marketplace/details/iowa-department-of-commerce/iowa-liquor-sales "

[https://console.cloud.google.com/bigquery\(cameo:product/iowa-department-of-commerce/iowa-liquor-sales\)?authuser=1&project=iowa-liquor-sales-334411](https://console.cloud.google.com/bigquery(cameo:product/iowa-department-of-commerce/iowa-liquor-sales)?authuser=1&project=iowa-liquor-sales-334411)

1.3 Data Preparation Methodology & Challenges

Having consistent, usable and correct data files imported to data source, is crucial to avoid inaccuracy and operating risks in the future. So, we began importing our dataset in R environment to choose the columns that were useful for are database.

This is the query that we used to retrieve the dataset of Iowa Retail Liquor Sales 2018–2021 from Google BigQuery [public datasets](#).

```
# Authorization in Google Big Query
bigquery::bq_auth()

# Store the project id
projectid = "iowa-liquor-sales-334411"

# Set your query
sql <- "SELECT * FROM bigquery-public-data.iowa_liquor_sales.sales WHERE date >= '2018-01-01'"

DT <- bq_table_download(
  bq_project_query(
    projectid,
    query=sql
  )
)
```

Below is the starting dataset, that has 9717461 rows and 24 columns.

	invoice_and_item_number	date	store_number	store_name	address	city	zip_code	store_location	county_number
1	INV-30832000021	2020-10-07	4463	Casey's General Store #3031 / Garner	145 Us Hwy 18 W	Garner	50438	POINT (-93.603007 43.105833)	41
2	INV-10221100182	2018-02-06	2619	Hy-Vee Wine and Spirits / WDM	1725 74th St	West Des Moines	50266.0	POINT (-93.808855 41.598515)	77
3	INV-41518100030	2021-11-01	2508	Hy-Vee Food Store #1 / Cedar Rapids	1843 Johnson Avenue, N.W.	Cedar Rapids	52405.0	POINT (-91.697941 41.97447)	57
4	INV-37946400010	2021-06-30	4507	Casey's General Store #2902 / Spencer	411 W 18th	Spencer	51301	POINT (-95.150966 43.155628)	21
5	INV-27941500002	2020-06-12	5252	Freeman Foods of North English	119, Main St	North English	52316	POINT (-92.166582 41.575084)	48
6	INV-40893000032	2021-10-12	3723	J D Spirits Liquor	1023 9th St	Onawa	51040.0	POINT (-96.095845 42.025841)	67
7	INV-38169600004	2021-07-08	5428	Casey's General Store # 2789/ Cedar Rapids	888 Vernon Valley Drive SE	Cedar Rapids	52403	POINT (-91.553516 41.977177)	57

county	category	category_name	vendor_number	vendor_name	item_number	item_description	pack	bottle_volume_ml	state_bottle_cost	state_bottle_retail
HANCOCK	1081600.0	Whiskey Liqueur	421	SAZERAC COMPANY INC	86888	Southern Comfort PET	6	1750	20.26	30.39
POLK	1081600.0	Whiskey Liqueur	421	SAZERAC COMPANY INC	64868	Fireball Cinnamon Whiskey	6	1750	15.33	23.00
LINN	1081600.0	Whiskey Liqueur	421.0	SAZERAC COMPANY INC	64904	Fireball Cinnamon Whiskey PET	6	1750	15.33	23.00
CLAY	1081600	Whiskey Liqueur	421	SAZERAC COMPANY INC	64904	Fireball Cinnamon Whiskey PET	6	1750	15.33	23.00
IOWA	1081600.0	Whiskey Liqueur	421	SAZERAC COMPANY INC	64904	Fireball Cinnamon Whiskey PET	6	1750	15.33	23.00
MONONA	1081600.0	Whiskey Liqueur	421	SAZERAC COMPANY INC	64904	Fireball Cinnamon Whiskey PET	6	1750	15.33	23.00
LINN	1081600	Whiskey Liqueur	421	SAZERAC COMPANY INC	64868	Fireball Cinnamon Whiskey	6	1750	15.33	23.00

bottles_sold	sale_dollars	volume_sold_liters	volume_sold_gallons
6	182.34	10.5	2.77
6	138.00	10.5	2.77
6	138.00	10.5	2.77
6	138.00	10.5	2.77
6	138.00	10.5	2.77
6	138.00	10.5	2.77
6	138.00	10.5	2.77

We decided choosing a subset of 16 columns from 24 columns of the full dataset to succeed the highest possible performance in our database. Specifically in our columns we made the following:

We dropped the following columns:

- 1) invoice_and_item_number
- 2) store_number
- 3) county_number
- 4) category
- 5) vendor_number
- 6) item_number
- 7) bottles_sold
- 8) volume_sold_liters
- 9) volume_sold_gallons

Since there are redundant and do not offer us extra information from what we already have.

The columns that we chose to build our Data Warehouse on are the following:

- 1) **Date:** Date of order
- 2) **Store_Name:** Name of store who ordered the liquor.
- 3) **County:** County where the store who ordered the liquor is located
- 4) **City:** City where the store who ordered the liquor is located
- 5) **Address:** Address of store who ordered the liquor.
- 6) **Zip_code:** Zip code where the store who ordered the liquor is located
- 7) **Store Location:** Location of store who ordered the liquor
- 8) **Vendor Name:** The vendor name of the company for the brand of liquor ordered
- 9) **Category Name:** Category of the liquor ordered.
- 10) **Item Description:** Description of the individual liquor product ordered.
- 11) **Pack:** The number of bottles in a case for the liquor ordered
- 12) **Bottle Volume ml:** Volume of each liquor bottle ordered in milliliters.
- 13) **State Bottle Cost:** The amount that Alcoholic Beverages Division paid for each bottle of liquor ordered
- 14) **State Bottle Retail:**

The amount the store paid for each bottle of liquor ordered

- 15) **Bottles Sold:** The number of bottles of liquor ordered by the store
- 16) **Sale Dollars:** Total cost of liquor order (number of bottles multiplied by the state bottle retail)

```
#Filter columns
df <- DT %>% select("date", "store_name", "county", "city", "address", "zip_code", "store_location", "vendor_name", "category_name", "item_description",
                    "pack", "bottle_volume_ml", "state_bottle_cost", "state_bottle_retail",
                    "bottles_sold", "sale_dollars")
```

The next step was to proceed with data cleansing. We made the following actions:

- Renaming columns to be more descriptive

```
# Renaming Columns
colnames(df)[which(names(df) == "city")] <- "City"
colnames(df)[which(names(df) == "pack")] <- "Pack"
colnames(df)[which(names(df) == "county")] <- "County"
colnames(df)[which(names(df) == "address")] <- "Address"
colnames(df)[which(names(df) == "category_name")] <- "Category_Name"
colnames(df)[which(names(df) == "date")] <- "Date"
colnames(df)[which(names(df) == "item_description")] <- "Item_Description"
colnames(df)[which(names(df) == "store_name")] <- "Store_Name"
colnames(df)[which(names(df) == "store_location")] <- "Store_Location"
colnames(df)[which(names(df) == "zip_code")] <- "Zip_Code"
colnames(df)[which(names(df) == "state_bottle_cost")] <- "State_Bottle_Cost"
colnames(df)[which(names(df) == "state_bottle_retail")] <- "State_Bottle_Retail"
colnames(df)[which(names(df) == "bottles_sold")] <- "Bottles_Sold"
colnames(df)[which(names(df) == "sale_dollars")] <- "Sale_dollars"
colnames(df)[which(names(df) == "bottle_volume_ml")] <- "Bottle_Volume_ml"
colnames(df)[which(names(df) == "vendor_name")] <- "Vendor_Name"
```

- Removing the string after '/' from the Store_Name Column, since it is the City that the store is located

store_name	county	city	address
Casey's General Store #3031 / Garner	HANCOCK	Garner	145 Us Hwy 18 W
Hy-Vee Wine and Spirits / WDM	POLK	West Des Moines	1725 74th St
Hy-Vee Food Store #1 / Cedar Rapids	LINN	Cedar Rapids	1843 Johnson Avenue, N.W.
Casey's General Store #2902 / Spencer	CLAY	Spencer	411 W 18th

```
#Removing the string after / in Store_Name column
df$Store_Name <- sub("/.*", "", df$Store_Name)
df$Store_Name <-trimws(df$Store_Name)
```

Store_Name
Casey's General Store #3031
Hy-Vee Wine and Spirits
Hy-Vee Food Store #1
Casey's General Store #2902

- Filter Store Location column in order to extract x and y coordinates and save them in 2 separate columns

store_location
POINT (-93.603007 43.105833)
POINT (-93.808855 41.598515)
POINT (-91.697941 41.97447)
POINT (-95.150966 43.155628)
POINT (-92.166582 41.575084)
POINT (-96.095845 42.025841)

```
#Filter Store_Location in order to keep only x,y coordinates in new columns, |
#we replace cells with no coordinate information with NA
df$Store_Location<- gsub(".*\\(", "(", df$Store_Location)
df$Store_Location<- gsub("\\).*", ")", df$Store_Location)
is.na(df$Store_Location) <- !startsWith(df$Store_Location, "(")
df$Store_Location<- gsub(".*\\(", "", df$Store_Location)
df$Store_Location<- gsub("\\).*", "", df$Store_Location)
df$x_coordinate <- as.numeric(sapply(strsplit(as.character(df$Store_Location), ' '), "[", 1))
df$x_coordinate <- as.character(df$x_coordinate)
df$x_coordinate <- substr(df$x_coordinate, 0, 8)
df$y_coordinate <- as.numeric(sapply(strsplit(as.character(df$Store_Location), ' '), "[", 2))
df$y_coordinate <- as.character(df$y_coordinate)
df$y_coordinate <- substr(df$y_coordinate, 0, 7)
df$Store_Location <- NULL
```

x_coordinate	y_coordinate
-93.6030	43.1058
-93.8088	41.5985
-91.6979	41.9744
-95.1509	43.1556
-92.1665	41.5750
-96.0958	42.0258

- We eliminated the discrepancies in zip code column

zip_code
50438
50266.0
52405.0
51301
52316
51040.0
52403
52241.0

```
df$Zip_Code = as.character(df$Zip_Code)
df$Zip_Code <- gsub("(.*)\\. (.*)", "\\1", df$Zip_Code)
```

Zip_Code
50438
50266
52405
51301
52316
51040
52403
52241

- We convert all string to lowercase in order to avoid inconsistencies

```
#convert all attributes that are characters to lowercase in order to remove inconsistencies
df <- df %>% mutate(City = tolower(City))
df <- df %>% mutate(Store_Name = tolower(Store_Name))
df <- df %>% mutate(County = tolower(County))
df <- df %>% mutate(Vendor_Name = tolower(Vendor_Name))
df <- df %>% mutate(Item_Description = tolower(Item_Description))
df <- df %>% mutate(Address = tolower(Address))
df <- df %>% mutate(Category_Name = tolower(Category_Name))
```

- Dealing with empty values by placing the string “other” in categorical columns and “NULL” in metric columns to be futureproof for our data warehouse design later

```
#We check for NA cells and replace them with ""
sapply(df, function(x) sum(is.na(x)))
df[is.na(df)] <- ""

#We check for empty cells in dimension columns and replace the Dimensions with "Other"
sapply(df, function(x) sum(x==""))

df$City[df$City == ""] <- "Other"
df$County[df$County == ""] <- "Other"
df$Vendor_Name[df$Vendor_Name== ""] <- "Other"
df$Address[df$Address== ""] <- "Other"
df$Zip_Code[df$Zip_Code== ""] <- "Other"
df$Category_Name[df$Category_Name== ""] <- "Other"
```

After the cleaning we have ended up with the following 17 columns :

"Date"	"Store_Name"	"County"	"City"	"Address"
"Zip_Code"	"Vendor_Name"	"Category_Name"	"Item_Description"	"Pack"
"Bottle_Volume_ml"	"State_Bottle_Cost"	"State_Bottle_Retail"	"Bottles_Sold"	"Sale_dollars"
"x_coordinate"	"y_coordinate"			

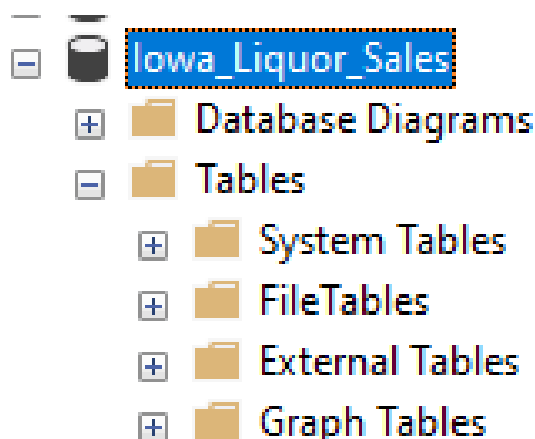
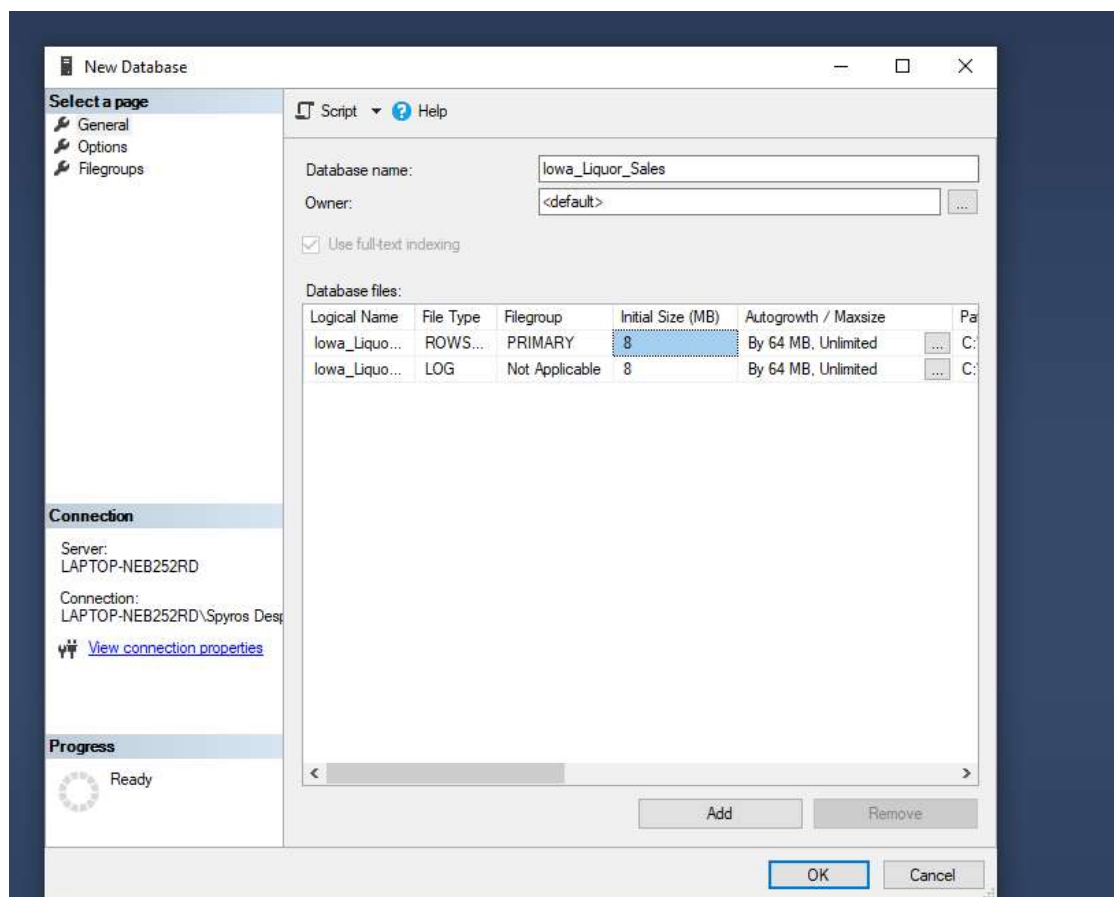
Finally our data were cleaned and ready to imported in our database.
Then we were ready to proceed our ETL process.

2. Designing our Data Flow

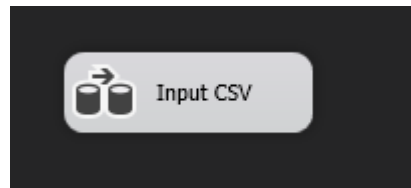
2.2 ETL Architecture

For our organisation we decided to develop an in-house data warehouse populated by SSIS packages. The ETL process in SSIS environment can handle a variety of data movement and transformation tasks and feed successfully our DWH with well-structured data.

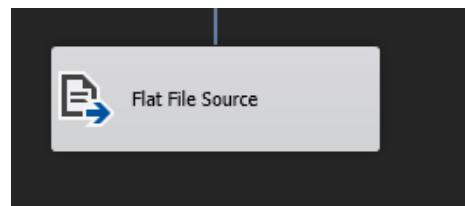
Before, we started implementing our ETL process, we had to create our database in SQL Server Management Studio (SSMS) named “Iowa_Liquor_Sales”.



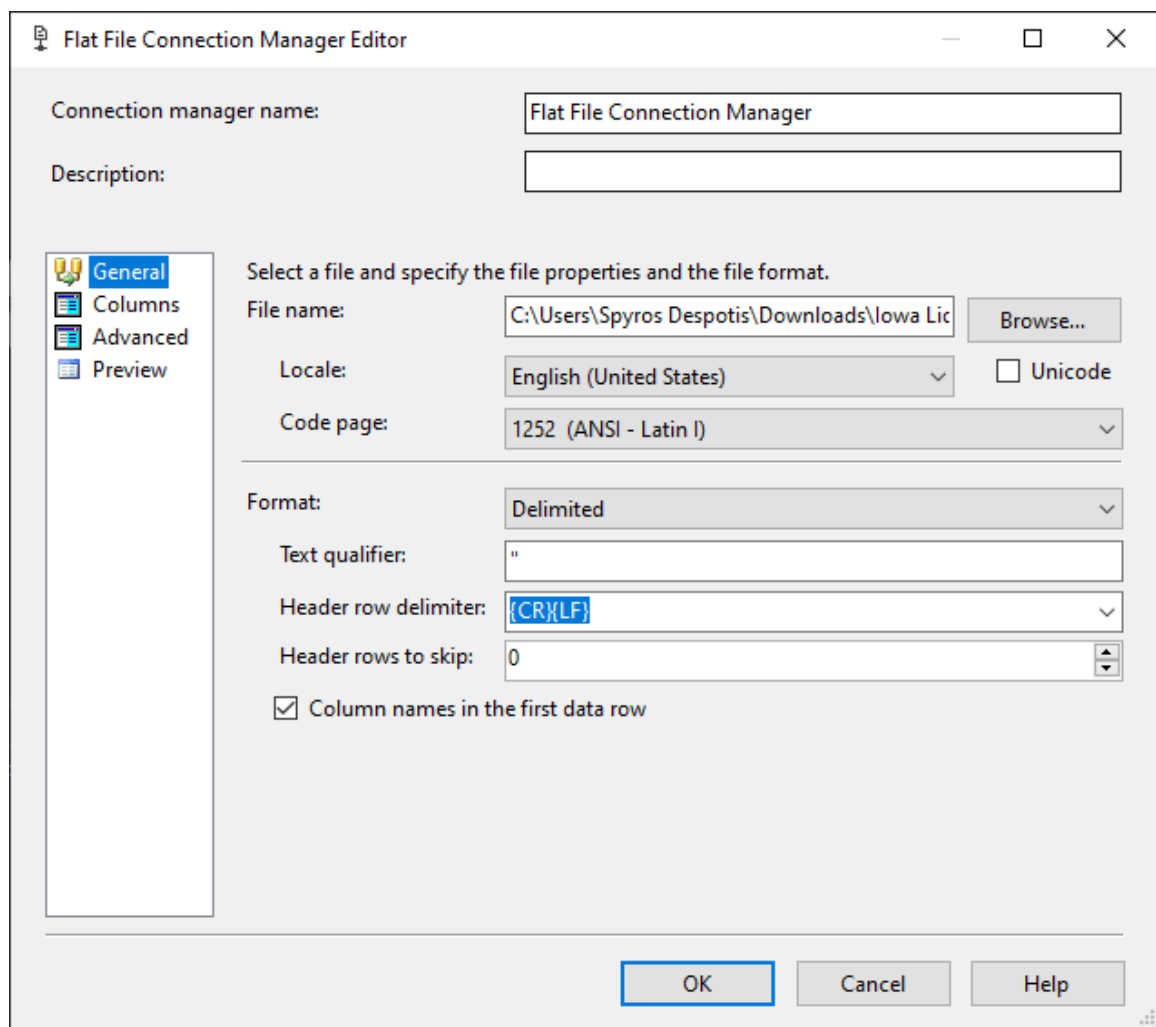
Then we proceeded with the **extraction** step. Specifically, we created in SQL Server Integration Services (SSIS) environment an integration Service Project called “Input CSV” in order to control our data feeding flow with Data Flow Tasks.



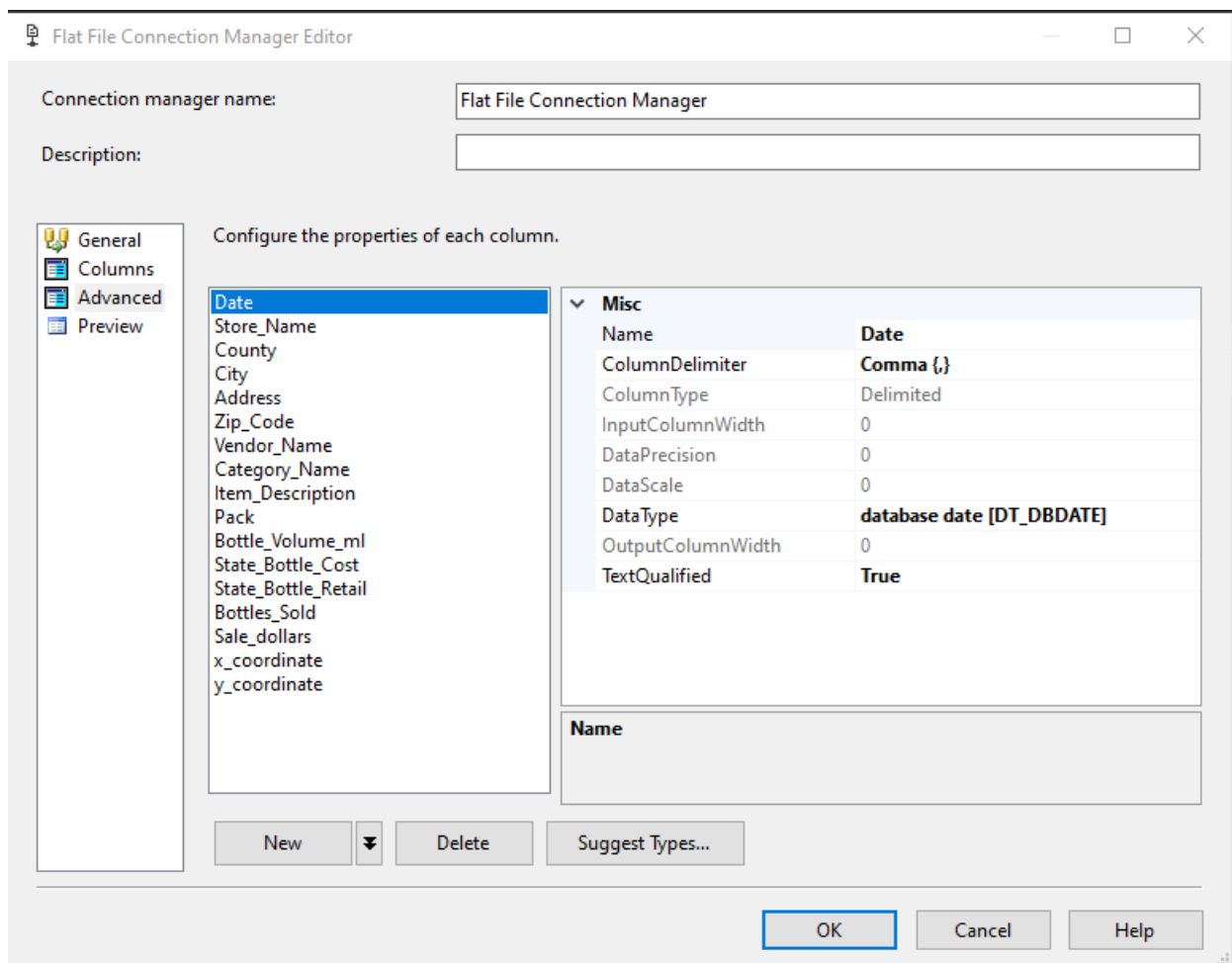
As data source we imported the CSV file (Flat File Source) that we cleaned in data preparation step.



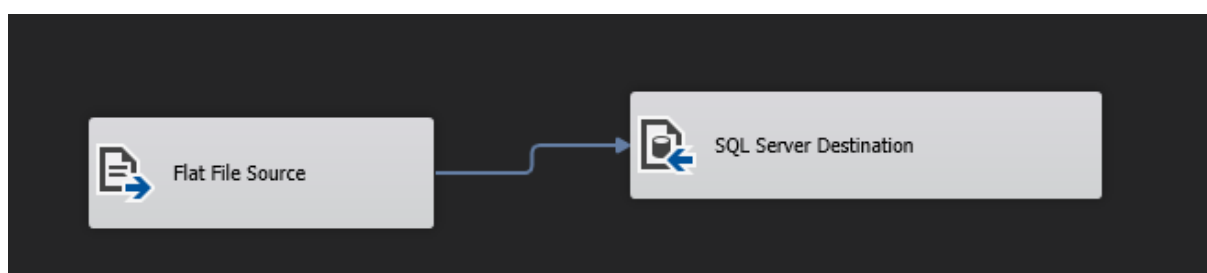
Then we continued with the **transformation** step. In this step we applied sets of rules to the extracted data to convert them into a standard format to meet the schema requirements of the target database.



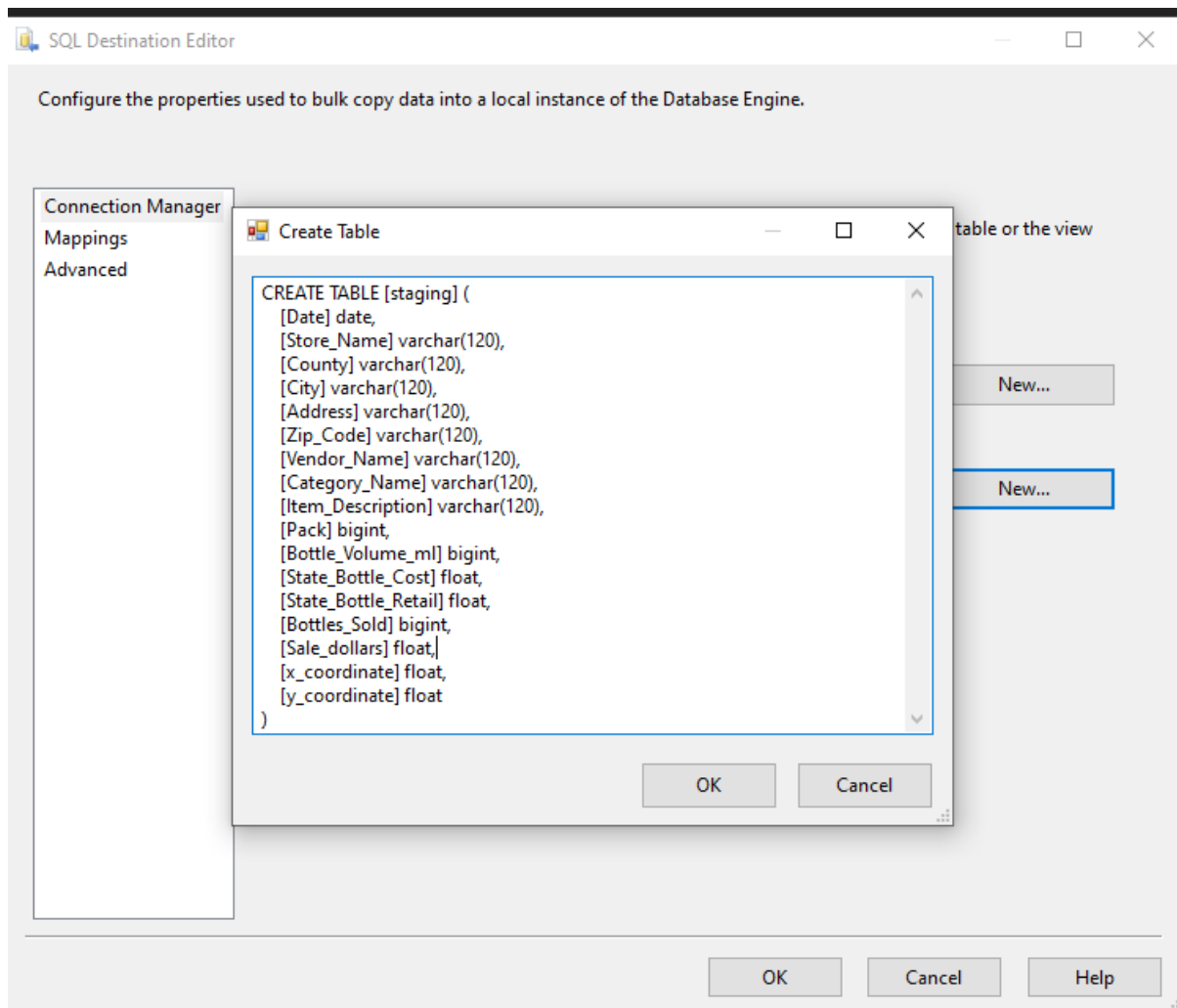
Inside the Flat File Source Editor, we set the csv format as delimited text, as qualifier the double quotes and as column delimiter the comma. Then we configured the properties of each column with the assistance of auto suggest data types from SSIS. This step was crucial for later in order the SQL server automatically to recognise correctly columns types from SSIS and also suggest correct data types. Also we improved the OutputColumnWidth of each column to be futureproof for bigger widths of columns later.



Afterwards we connected our Data Task Flow with the destination SQL Database Server.



We connected in our database “ Iowa _Liquor_Sales “ and then we created a new table named “staging” in order to extract temporary our data. Staging area or landing zone gives an opportunity to validate extracted data before it moves into the final Data warehouse.



Finally our data flow task was ready. Then we had to test if our data flow wrote all the necessary rows in Destination Sql Server and was running successfully .



Date	Store_Name	County	City	Address	Zip_Code	Vendor_Name	Item_Category	Item_Description	Pack	Bottle_Volume_ml	State_Bottle_Cost	State_Bottle_Retail	Bottles_So
2021-03-11	discount liquor	linn	cedar rapids	2933 1st ave se	52402	pemod ricard usa	imported vodkas	absolut swedish vodka 80prf mini	10	50	7.92	11.88	10
2020-07-31	casey's general store #1997	allamakee	waukon	516 rossville rd	52172	pemod ricard usa	imported vodkas	absolut swedish vodka 80prf mini	10	50	7.92	11.88	10
2021-05-07	tequila's liquor store	polk	des moines	1434 des moines st ste 5	50316	pemod ricard usa	imported vodkas	absolut swedish vodka 80prf mini	10	50	7.92	11.88	10
2020-07-21	kum & go #52	johnson	iowa city	25 w burlington st	52240	pemod ricard usa	imported vodkas	absolut swedish vodka 80prf mini	10	50	7.92	11.88	10
2021-07-28	kum & go #521	johnson	coralville	205 2nd st	52241	pemod ricard usa	imported vodkas	absolut swedish vodka 80prf mini	10	50	7.92	11.88	10
2021-08-05	kum & go #50	polk	west des moines	745 s 51st st	50265	pemod ricard usa	imported vodkas	absolut swedish vodka 80prf mini	10	50	7.92	11.88	10
2020-02-06	yesway store # 10026	cerro gord	mason city	1303 4th st sw	50401	pemod ricard usa	imported vodkas	absolut swedish vodka 80prf mini	10	50	7.92	11.88	10
2020-05-28	smokin joe's # 6 tobacco and liquor outlet	johnson	coralville	2411 2nd st ste 4	52241	pemod ricard usa	imported vodkas	absolut swedish vodka 80prf mini	10	50	7.92	11.88	10
2021-05-25	keystone liquor	johnson	coralville	517 2nd st	52241	pemod ricard usa	imported vodkas	absolut swedish vodka 80prf mini	10	50	7.92	11.88	10
2020-09-17	yesway store # 10026	cerro gord	mason city	1303 4th st sw	50401	pemod ricard usa	imported vodkas	absolut swedish vodka 80prf mini	10	50	7.92	11.88	10
2020-12-18	gasland #102	des moines	burlington	1703 mt pleasant st	52601	pemod ricard usa	imported vodkas	absolut swedish vodka 80prf mini	10	50	7.92	11.88	10
2020-11-24	marshall beer wine spirits	marshall	marshalltown	11 n 3rd ave	50158	pemod ricard usa	imported vodkas	absolut swedish vodka 80prf mini	10	50	7.92	11.88	10
2019-11-07	kwik stop liquor & groceries ames	story	ames	125 6th st	50010	pemod ricard usa	imported vodkas	absolut swedish vodka 80prf mini	10	50	7.92	11.88	10
2021-05-03	casey's general store #2913	stovr	colo	702 us hwy 65	50056	pemod ricard usa	imported vodkas	absolut swedish vodka 80prf mini	10	50	7.92	11.88	10

```

/***** Script for SelectTopNRows command from SSMS *****/
SELECT COUNT(*)
FROM [Iowa_Liquor_Sales].[dbo].[staging]

```

Despotis)

100 %

Results Messages

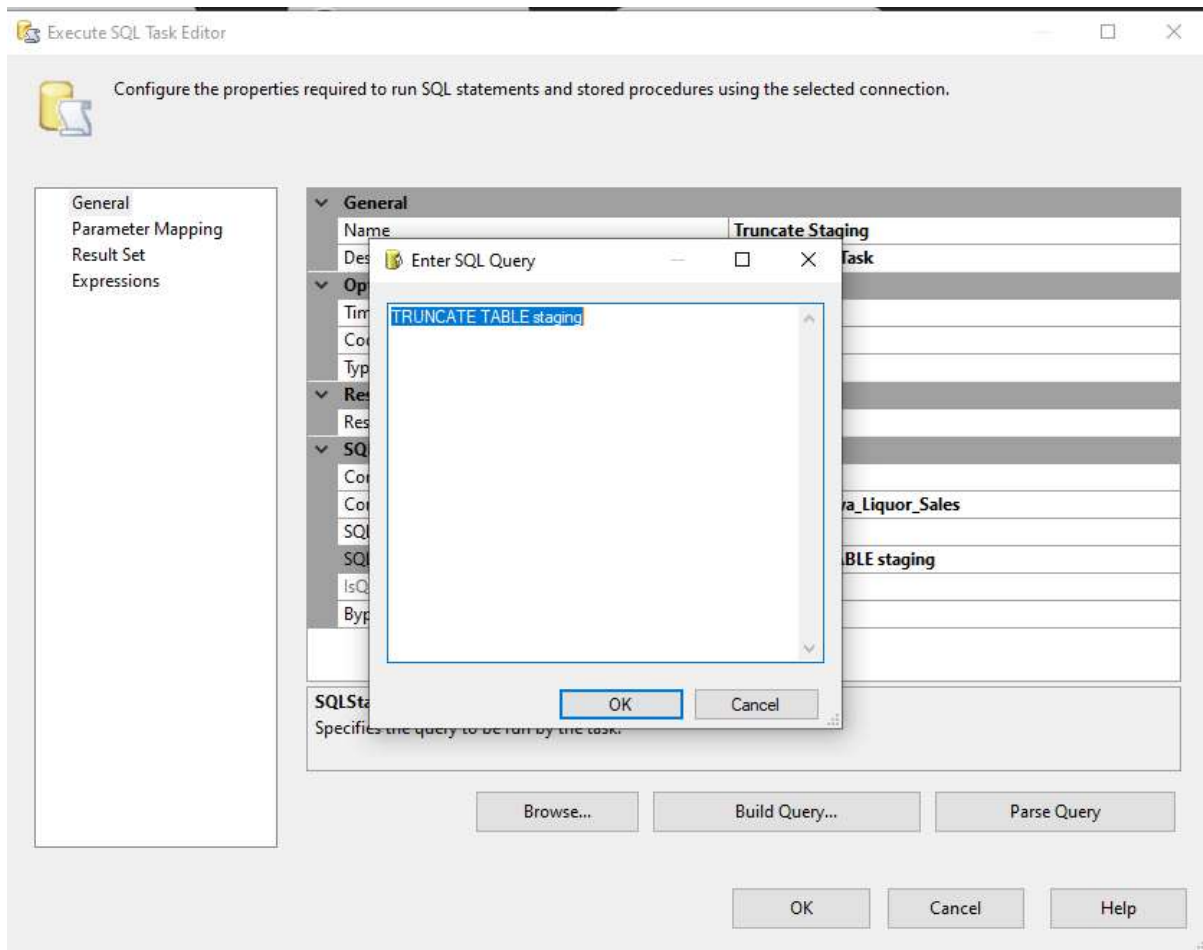
	(No column name)
1	9717461

Last but not least, we created a truncate pattern that will delete all rows in staging table and insert again from the data source all afresh data with unique ids. So we created a new “Execute SQL Task”, named “Truncate Staging” connected to our SQL database and we inserted the appropriate SQL statement. Then we tested again if the new flow was running successfully.

Execute SQL Task Editor

Configure the properties required to run SQL statements and stored procedures using the selected connection.

General	Parameter Mapping	Result Set	Expressions
<div>General</div> <div>Name</div> <div>Description</div> <div>Options</div> <div>Result Set</div> <div>SQL Statement</div> <div>Name</div>			
Name		Truncate Staging	
Description		Execute SQL Task	
Timeout		0	
CodePage		1252	
TypeConversionMode		Allowed	
ResultSet		None	
ConnectionType		OLE DB	
Connection		LocalHost.Iowa_Liquor_Sales	
SQLSourceType		Direct input	
SQLStatement		TRUNCATE TABLE staging	
IsQueryStoredProcedure		False	
BypassPrepare		True	
Name			
Specifies the name of the task.			
Browse...		Build Query...	
Parse Query			



3. Fact Table & Dimension Tables

3.1 Create & Update of Dimension Tables

Our database contains seven major dimensions with relationships:

1. Category Dimension:

- Fact/ Measurement: the category name of the liquor ordered
- Columns: id_category (primary key) and label_category

Column Name	Data Type	Allow Nulls
id_category	int	<input type="checkbox"/>
label_category	varchar(120)	<input type="checkbox"/>

Column Properties	
(General)	
(Name)	id_category
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	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

2. Item Description Dimension

- Fact/ Measurement: the description of the individual liquor product ordered.
- Columns: id_item (primary key), id_category_fk (foreign key from table category and column id_category) and label_item

Column Name	Data Type	Allow Nulls
id_item	int	<input type="checkbox"/>
label_item	varchar(120)	<input type="checkbox"/>
id_category_fk	int	<input type="checkbox"/>

Column Properties	
(General)	
(Name)	id_item
Allow Nulls	No
Data Type	int
Default Value or Binding	

Tables and Columns

Relationship name:
FK_item_dimension_item_dimension

Primary key table:
category_dimension

Foreign key table:
item_dimension

id_category

id_category_fk

OK Cancel

3. Date Dimension

- Fact/ Measurement: Date, Year, Quarter, Month of order
- Columns: id_date (primary key), label_date, label_year, label_month, label_day, label_quarter, label_monthname, label_dayname, label_daynum

	Column Name	Data Type	Allow Nulls
🔑	id_date	int	<input type="checkbox"/>
	label_date	date	<input type="checkbox"/>
	label_year	int	<input type="checkbox"/>
	label_month	int	<input type="checkbox"/>
	label_day	int	<input type="checkbox"/>
	label_quarter	int	<input type="checkbox"/>
	label_monthname	varchar(50)	<input type="checkbox"/>
	label_dayname	varchar(120)	<input type="checkbox"/>
▶	label_daynum	int	<input type="checkbox"/>
			<input type="checkbox"/>

Column Properties

(General)

(Name) label_daynum

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 No

Has Non-SQL Server Subscriber No

Identity Specification No

Indestructible Yes

4. Vendor Dimension

- Fact/ Measurement: The vendor name of the company for the brand of liquor ordered
- Columns: id_vendor (primary key), label_vendor

Column Name	Data Type	Allow Nulls
id_vendor	int	<input type="checkbox"/>
label_vendor	varchar(120)	<input type="checkbox"/>

Column Properties	
(General)	
(Name)	id_vendor
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	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

5. Store Dimension

- Fact/ Measurement: Name and address of store who ordered the liquor.
- Columns: id_store (primary key), label_store, label_address, label_zipcode, label_longitude, label_latitude and id_cityfk (foreign key from City table and id_city column)

Column Name	Data Type	Allow Nulls
id_store	int	<input type="checkbox"/>
label_store	varchar(120)	<input type="checkbox"/>
label_address	varchar(120)	<input type="checkbox"/>
label_zipcode	varchar(120)	<input type="checkbox"/>
label_longitude	float	<input checked="" type="checkbox"/>
label_latitude	float	<input checked="" type="checkbox"/>
id_cityfk	int	<input type="checkbox"/>

Column Properties	
(General)	
(Name)	id_cityfk
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	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

6. City Dimension

- Fact/ Measurement: City where the store who ordered the liquor is located
- Columns: id_city (primary key), label_city and id_county_fk (foreign key from table County and column column_id)

Column Name	Data Type	Allow Nulls
id_city	int	<input type="checkbox"/>
label_city	varchar(120)	<input type="checkbox"/>
id_county_fk	int	<input type="checkbox"/>


Column Properties	
▼ (General)	
(Name)	id_county_fk
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	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	4

Foreign Key Relationships		?	×
Selected Relationship:		Editing properties for existing relationship.	
city_dimension_county_dimension			
store_dimension_city_dimension			
		▼ (General)	
		Check Existing Data On Creat Yes	
		> Tables And Columns Specific	
		▼ Identity	
		(Name)	FK_city_dimension_county_dimension
		Description	
		▼ Table Designer	
		Enforce For Replication	Yes
		Enforce Foreign Key Constrai	Yes
		> INSERT And UPDATE Specifica	

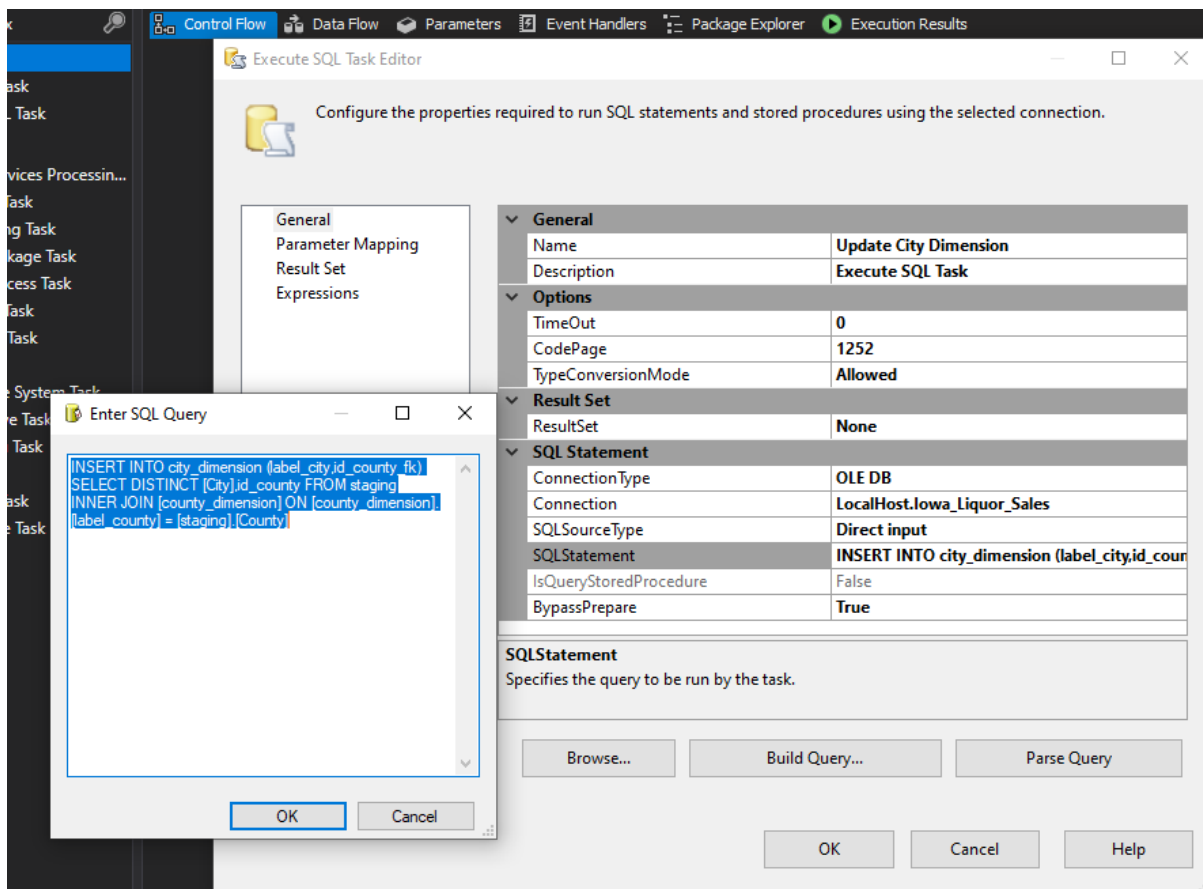
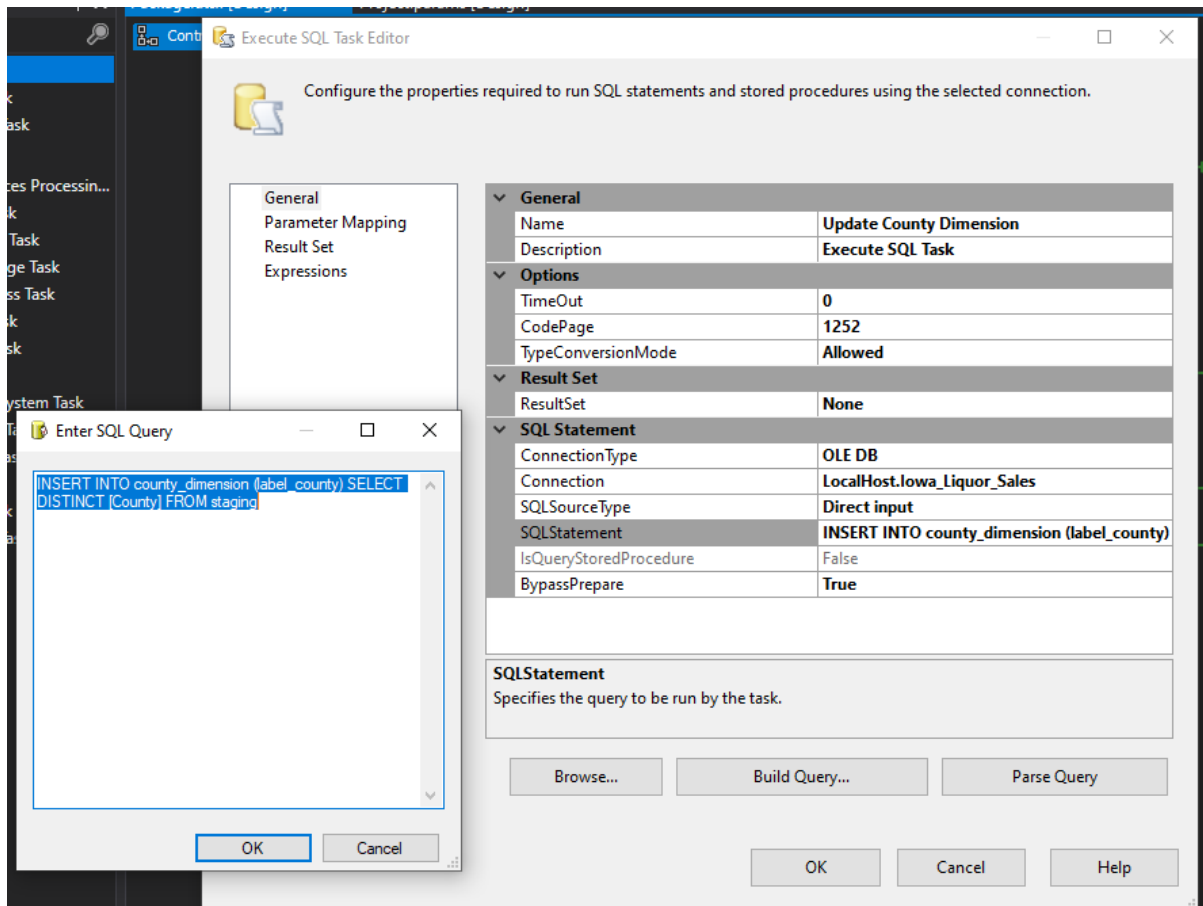
7. County Dimension

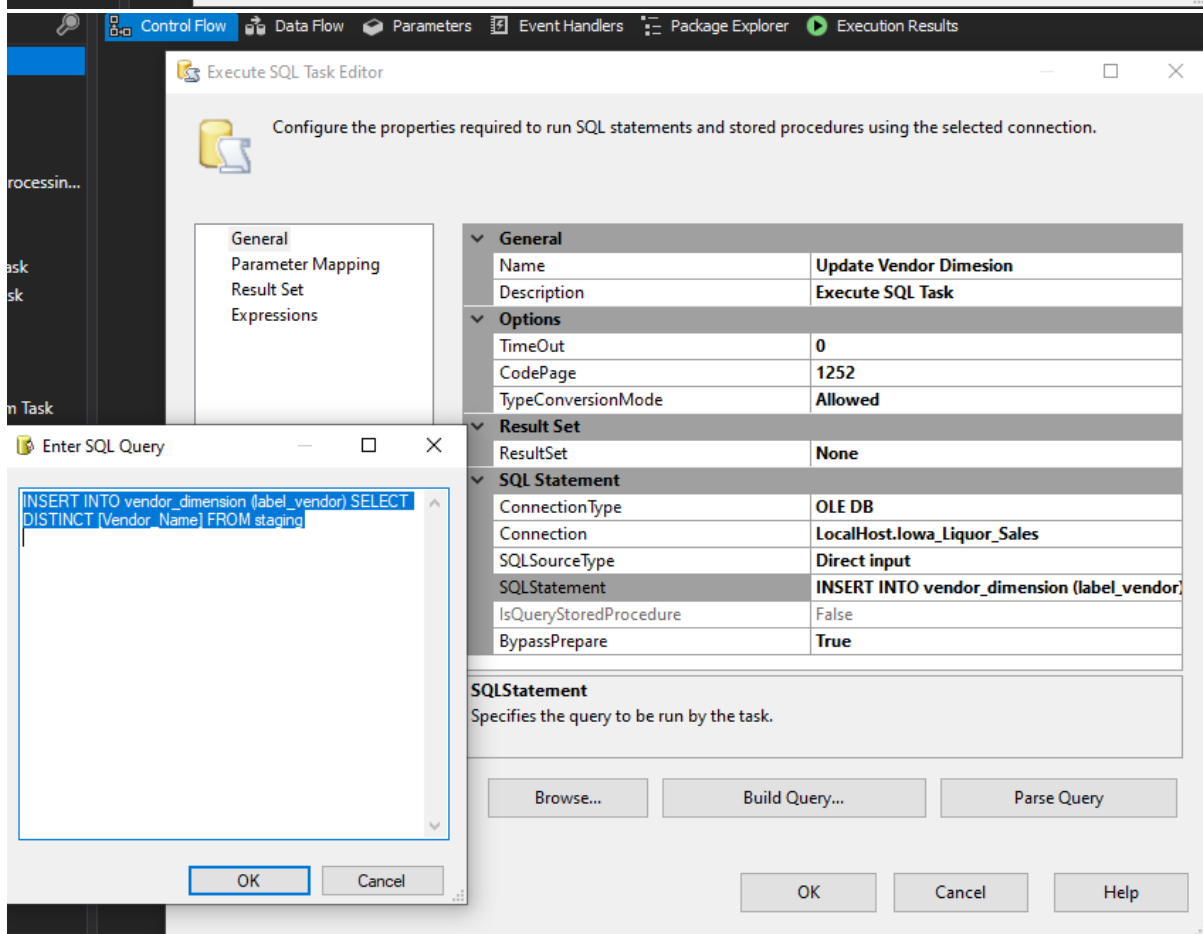
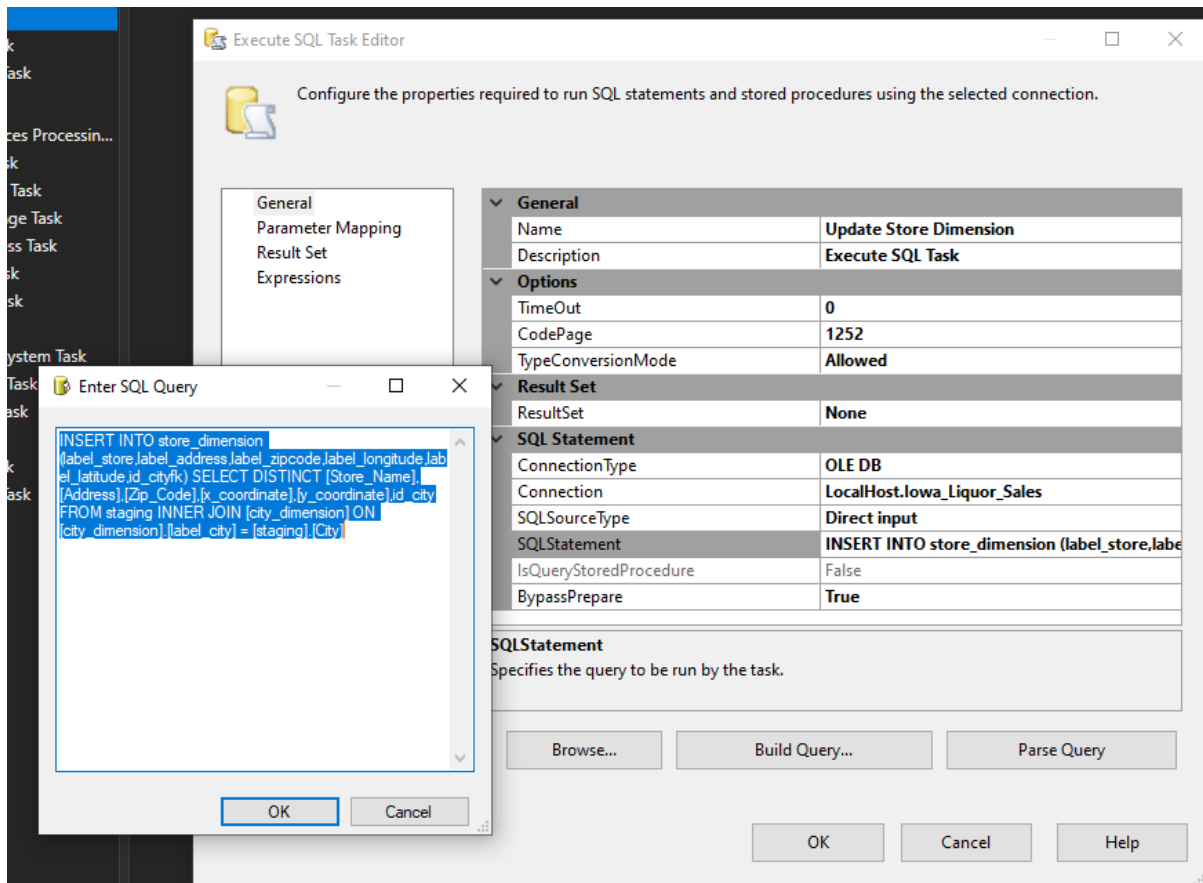
- Fact/ Measurement: County where the store who ordered the liquor is located
- Columns: id_county (primary key), label_county

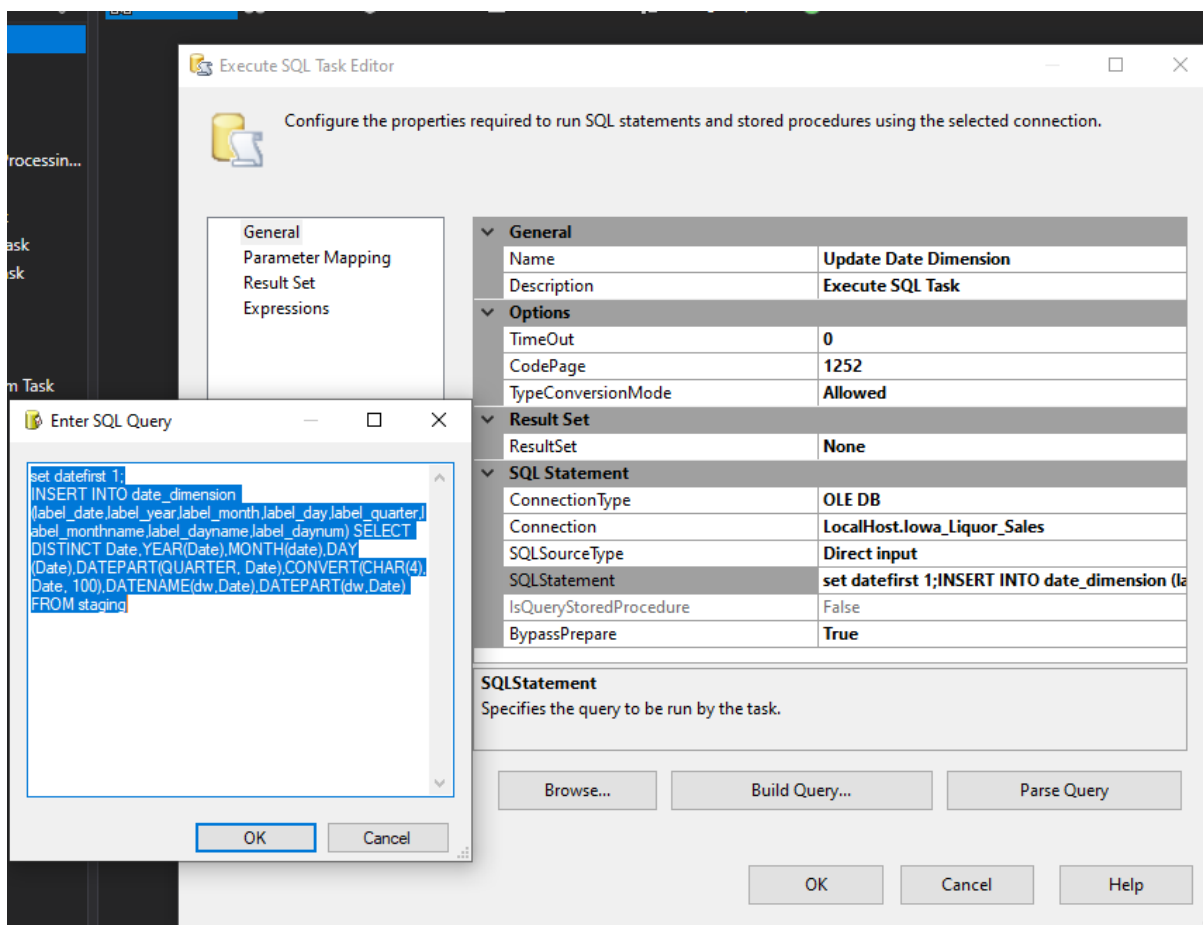
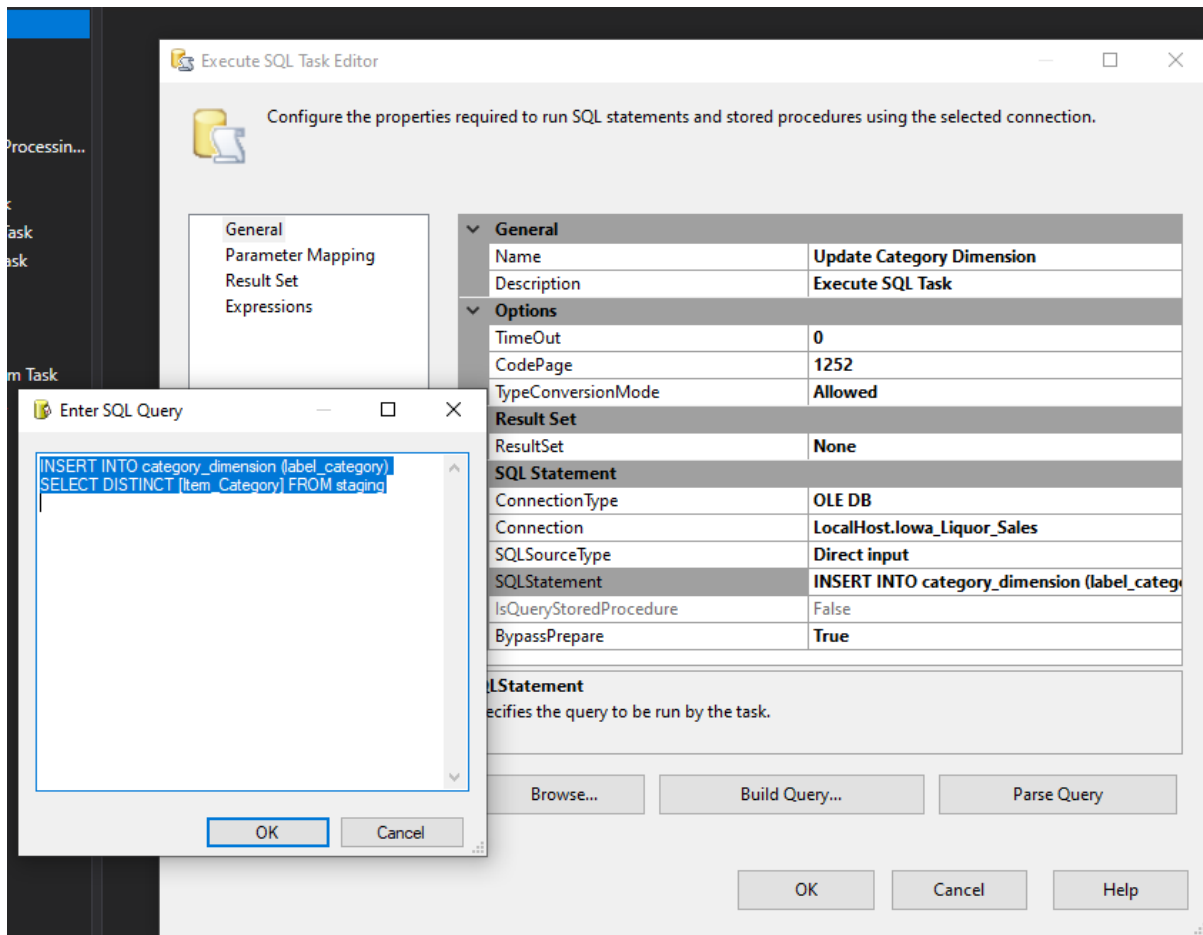
Column Name	Data Type	Allow Nulls
id_county	int	<input type="checkbox"/>
label_county	varchar(120)	<input type="checkbox"/>
		<input type="checkbox"/>

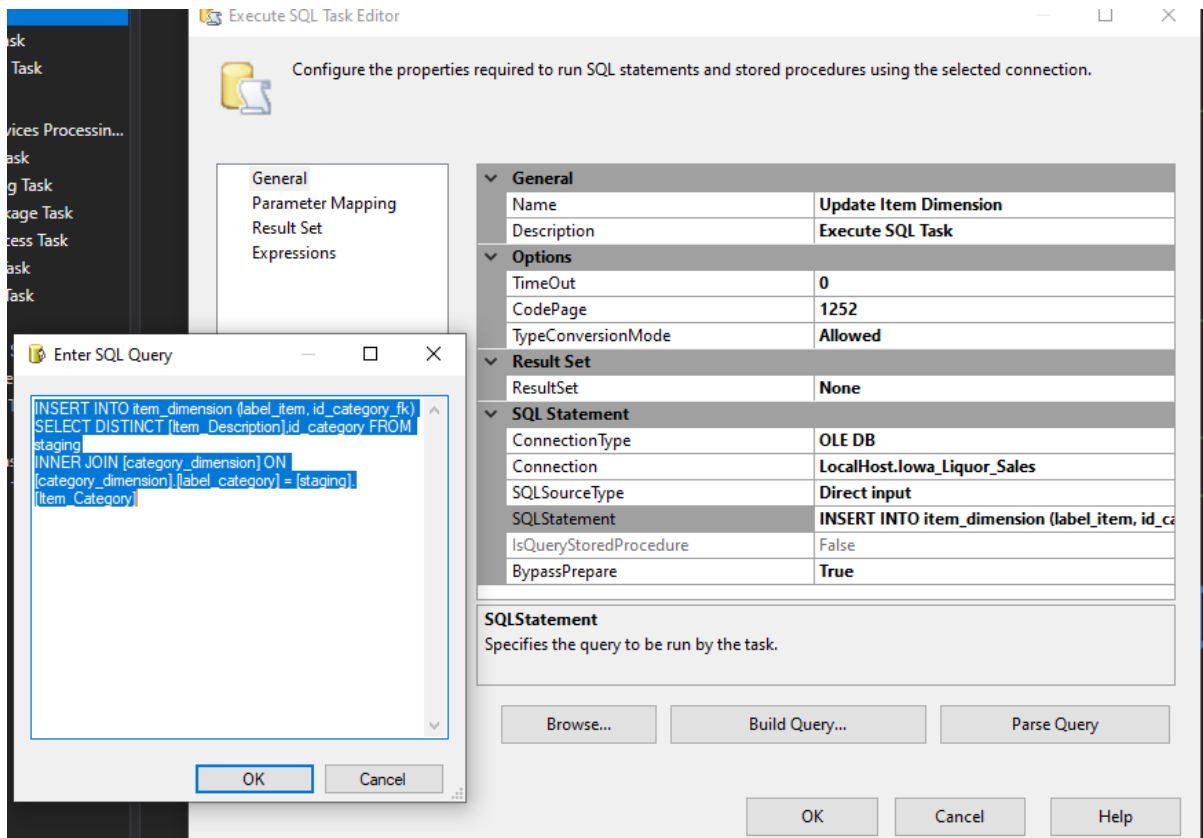
Column Properties	
	
▼ (General)	
(Name)	id_county
Allow Nulls	No
Data Type	int
Default Value or Binding	
▼ Table Designer	
Collation	< database default >
> Computed Column Specification	
Condensed Data Type	int
Description	
Deterministic	Yes
DTS-published	No
> Full-text Specification	No
Has Non-SQL 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

The next step was to update our data flow with new SQL tasks, in order to feed with data all the new tables - dimensions. As Connection for SQL tasks we used the “LocalHost.Iowa_Liquor_Sales” and for every new dimensions we inserted a SQL query based on the relationships between the tables and the distinct values.

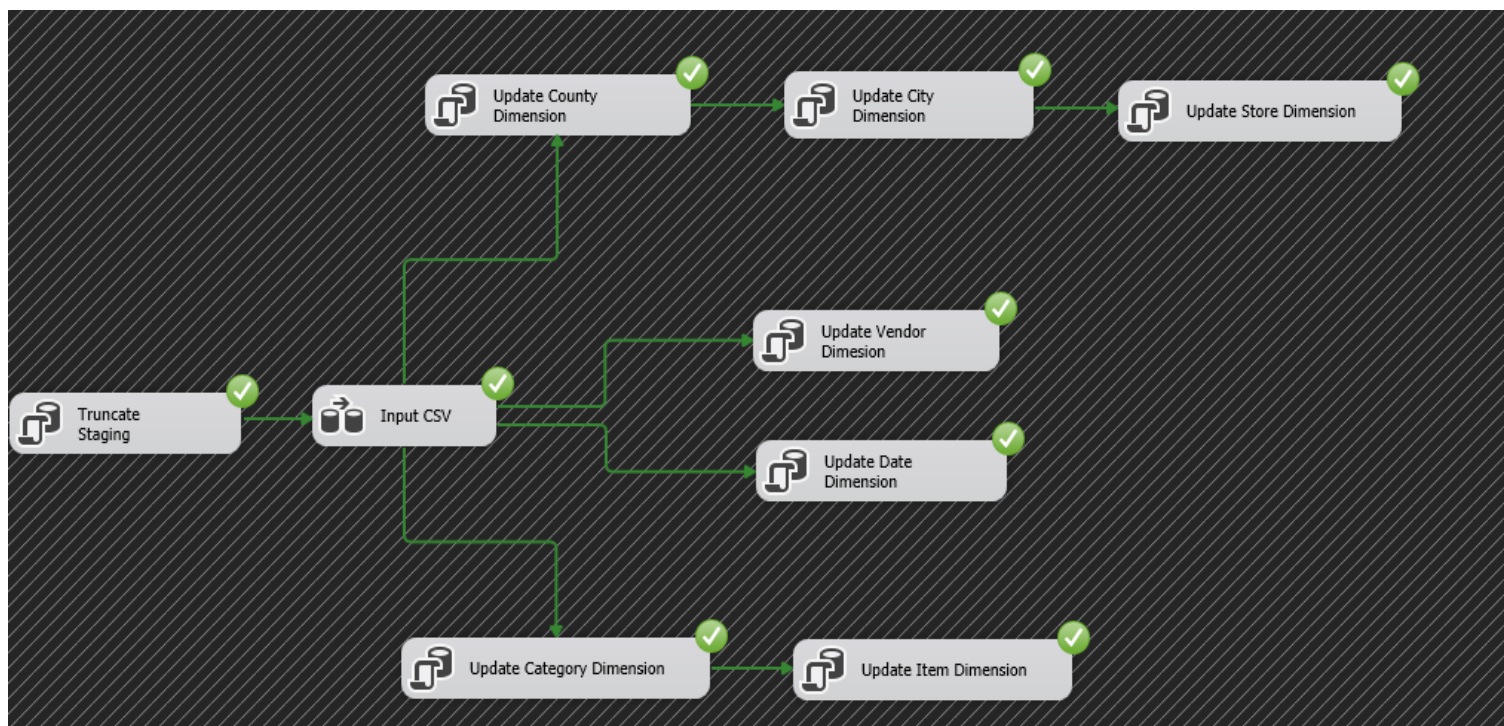








Our data flow with the dimensions was ready.



[SSIS.Pipeline] Information: Cleanup phase is beginning.

- Progress: Cleanup - 0 percent complete
- Progress: Cleanup - 50 percent complete
- Progress: Cleanup - 100 percent complete
- Finished, 8:16:01 PM, Elapsed time: 00:00:35.063

Task Truncate Staging

- Validation has started (2)
- Validation is completed (2)
- Start, 8:15:26 PM
- Progress: Executing query "TRUNCATE TABLE staging". - 100 percent complete
- Finished, 8:15:26 PM, Elapsed time: 00:00:00.031

Task Update Category Dimension

- Validation has started (2)
- Validation is completed (2)
- Start, 8:16:01 PM
- Progress: Executing query "INSERT INTO category_dimension (label_category) SE...". - 100 percent complete
- Finished, 8:16:04 PM, Elapsed time: 00:00:03.172

Task Update City Dimension

- Validation has started (2)
- Validation is completed (2)
- Start, 8:16:03 PM
- Progress: Executing query "INSERT INTO city_dimension (label_city,id_county_f...". - 100 percent complete
- Finished, 8:16:06 PM, Elapsed time: 00:00:03.031

Task Update County Dimension

- Validation has started (2)
- Validation is completed (2)
- Start, 8:16:01 PM
- Progress: Executing query "INSERT INTO county_dimension (label_county) SELECT...". - 100 percent complete
- Finished, 8:16:03 PM, Elapsed time: 00:00:02.625

Task Update Date Dimension

- Validation has started (2)
- Validation is completed (2)
- Start, 8:16:01 PM
- Progress: Executing query "set datefirst 1; INSERT INTO date_dimension (label...". - 100 percent complete
- Finished, 8:16:06 PM, Elapsed time: 00:00:05.297

Task Update Item Dimension

- Validation has started (2)
- Validation is completed (2)
- Start, 8:16:04 PM
- Progress: Executing query "INSERT INTO item_dimension (label_item, id_categor...". - 100 percent complete
- Finished, 8:16:06 PM, Elapsed time: 00:00:02.515

Task Update Store Dimension

- Validation has started (2)
- Validation is completed (2)
- Start, 8:16:06 PM
- Progress: Executing query "INSERT INTO store_dimension (label_store,label_add...". - 100 percent complete
- Finished, 8:16:09 PM, Elapsed time: 00:00:02.422

Task Update Vendor Dimension

- Validation has started (2)
- Validation is completed (2)
- Start, 8:16:01 PM
- Progress: Executing query "INSERT INTO vendor_dimension (label_vendor) SELECT...". - 100 percent complete
- Finished, 8:16:03 PM, Elapsed time: 00:00:02.625

Validation is completed

Start, 8:15:26 PM

Finished, 8:16:09 PM, Elapsed time: 00:00:43.266

```
SELECT TOP (1000) [id_item]
, [label_item]
, [id_category_fk]
FROM [Iowa_Liquor_Sales].[dbo].[item_dimension]
```

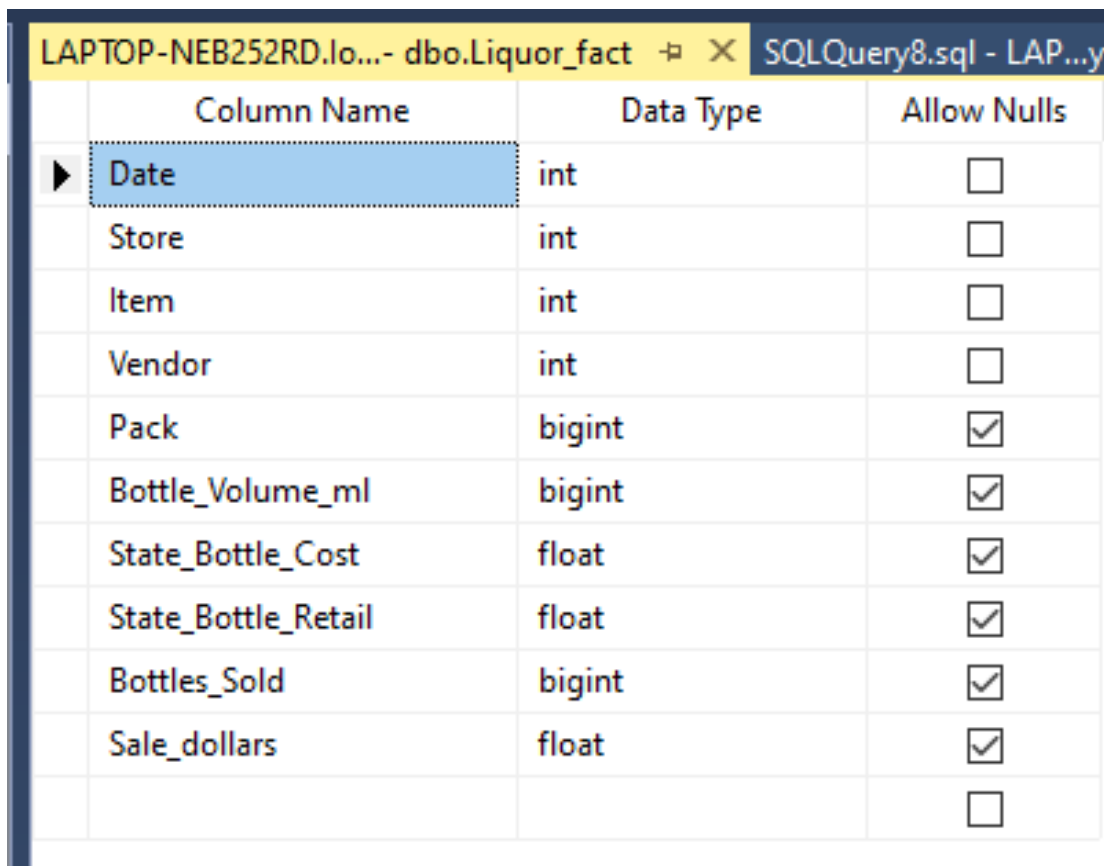
100 %

Results Messages

	id_item	label_item	id_category_fk
1	1	"rumchata ""minichatas"" creamer cups"	22
2	2	135A° east hyogo japanese dry gin	50
3	3	135i?½east hyogo japanese dry gin	50
4	4	1792 12yr old bourbon	38
5	5	1792 bottle in bond bourbon	20
6	6	1792 bottled in bond bourbon	14
7	8	1792 bottled in bond bourbon barrel	14
8	9	1792 full proof	38
9	10	1792 full proof buy the barrel	41
10	11	1792 single barrel	41

3.2 Create & Update Fact Table

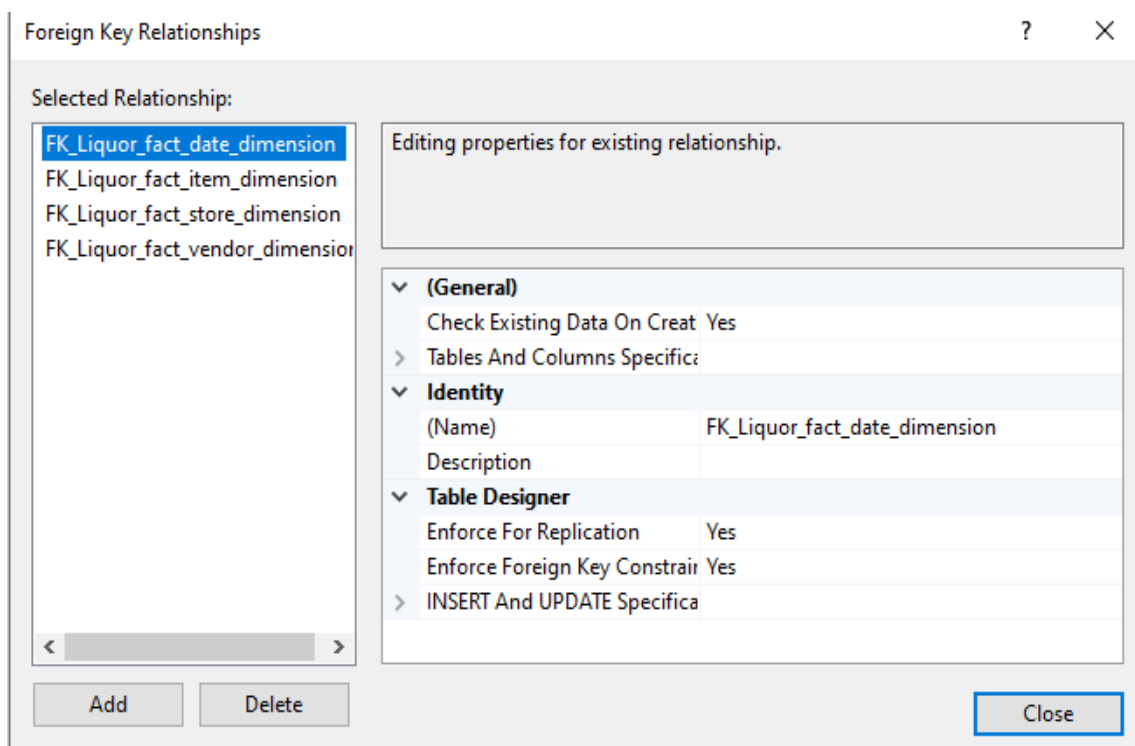
Then we continued with designing the fact table which will be the center of our snowflake shema. The fact table contains the content of the data warehouse and store different types of measures like additive, non additive, and semi additive measures.



The screenshot shows a SQL Server Enterprise Designer window with the title 'LAPTOP-NEB252RD.lo... - dbo.Liquor_fact' and 'SQLQuery8.sql - LAP...y'. The table structure is displayed in a grid with columns: Column Name, Data Type, and Allow Nulls. The 'Date' column is selected.

Column Name	Data Type	Allow Nulls
Date	int	<input type="checkbox"/>
Store	int	<input type="checkbox"/>
Item	int	<input type="checkbox"/>
Vendor	int	<input type="checkbox"/>
Pack	bigint	<input checked="" type="checkbox"/>
Bottle_Volume_ml	bigint	<input checked="" type="checkbox"/>
State_Bottle_Cost	float	<input checked="" type="checkbox"/>
State_Bottle_Retail	float	<input checked="" type="checkbox"/>
Bottles_Sold	bigint	<input checked="" type="checkbox"/>
Sale_dollars	float	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

The foreign keys allow joins with dimension tables:



The next step was to insert the appropriate data to fact table with the following query:

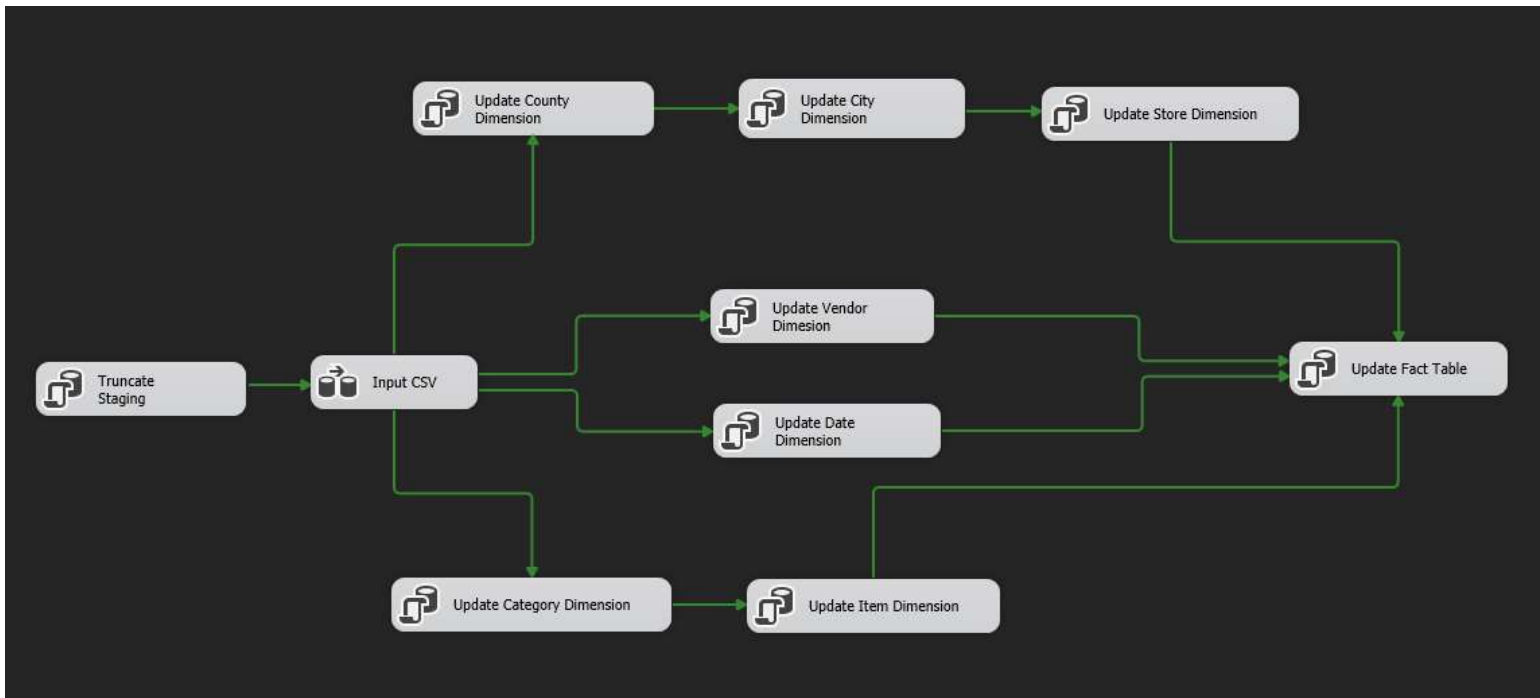
```
INSERT INTO liquor_fact (Date, Store, Item, Vendor, Pack, Bottle_Volume_ml, State_Bottle_Cost, State_Bottle_Retail, Bottles_Sold, Sale_dollars)
SELECT
[date_dimension].[id_date] AS [Date],
[store_dimension].[id_store] AS [Store],
[item_dimension].[id_item] AS [Item],
[vendor_dimension].[id_vendor] AS [Vendor],
[staging].[Pack] AS [Pack],
[staging].[Bottle_Volume_ml] AS [Bottle_Volume_ml],
[staging].[State_Bottle_Cost] AS [State_Bottle_Cost],
[staging].[State_Bottle_Retail] AS [State_Bottle_Retail],
[staging].[Bottles_Sold] AS [Bottles_Sold],
[staging].[Sale_dollars] AS [Sale_dollars]
FROM staging
INNER JOIN [date_dimension] ON [staging].[Date] = [date_dimension].[label_date]
INNER JOIN [store_dimension] ON [staging].[Store_Name] = [store_dimension].[label_store]
INNER JOIN [vendor_dimension] ON [staging].[Vendor_Name] = [vendor_dimension].[label_vendor]
INNER JOIN [item_dimension] ON [staging].[Item_Description] = [item_dimension].[label_item]
```

Script for SelectTopNRows command from SSMS

```
SELECT TOP (1000) [Date]
, [Store]
, [Item]
, [Vendor]
, [Pack]
, [Bottle_Volume_ml]
, [State_Bottle_Cost]
, [State_Bottle_Retail]
, [Bottles_Sold]
, [Sale_dollars]
FROM [Iowa_Liquor_Sales].[dbo].[Liquor_fact]
```

	Date	Store	Item	Vendor	Pack	Bottle_Volume_ml	State_Bottle_Cost	State_Bottle_Retail	Bottles_Sold	Sale_dollars
1	391	1485	798	98	6	750	10	15	6	90
2	391	1520	7218	254	6	1750	9.5	14.25	6	85.5
3	391	1484	2650	281	12	750	9	13.5	24	324
4	391	1485	2650	281	12	750	9	13.5	24	324
5	391	1486	2650	281	12	750	9	13.5	24	324
6	391	1487	2650	281	12	750	9	13.5	24	324
7	391	1934	5780	83	12	500	13.5	20.25	1	20.25
8	391	1488	2650	281	12	750	9	13.5	24	324
9	391	1533	2650	281	12	750	9	13.5	24	324
10	391	1534	2650	281	12	750	9	13.5	24	324
11	391	1535	2650	281	12	750	9	13.5	24	324
12	391	1536	2650	281	12	750	9	13.5	24	324
13	391	1537	2650	281	12	750	9	13.5	24	324
14	391	1519	7218	254	6	1750	9.5	14.25	6	85.5
15	391	1518	7218	254	6	1750	9.5	14.25	6	85.5
16	391	1517	7218	254	6	1750	9.5	14.25	6	85.5
17	391	1516	7218	254	6	1750	9.5	14.25	6	85.5
18	391	1515	7218	254	6	1750	9.5	14.25	6	85.5
19	391	1514	7218	254	6	1750	9.5	14.25	6	85.5
20	391	1094	261	280	6	1750	11.55	17.33	48	831.84
21	391	1513	7218	254	6	1750	9.5	14.25	6	85.5
22	391	1512	7218	254	6	1750	9.5	14.25	6	85.5
23	391	1511	7218	254	6	1750	9.5	14.25	6	85.5
24	391	1510	7218	254	6	1750	9.5	14.25	6	85.5
25	391	1670	851	255	12	1000	6.63	9.95	24	238.8
26	391	1509	7218	254	6	1750	9.5	14.25	6	85.5
27	391	1508	7218	254	6	1750	9.5	14.25	6	85.5
28	391	2330	5781	83	12	1000	15.84	23.76	24	570.24
29	391	1507	7218	254	6	1750	9.5	14.25	6	85.5

The next step was to insert our new table, inside the ETL process. So we created a new SQL Task called “Update Fact Table”, with SQL Statement the query we tested before and we executed the flow again.



The last step was to schedule our package deploy in SSIS environment. But firstly we had to create a new catalogue in Integration Services Catalogs:

The screenshot shows the 'Catalog Creation Wizard' window. The 'General' tab is selected, showing options to 'Enable CLR Integration' (checked) and 'Enable automatic execution of Integration Services stored procedure at SQL Server startup' (unchecked). The 'Name of the catalog database' is set to 'SSISDB'. A password field is present for encryption. The 'Connection' section shows the connection to 'LAPTOP-NEB252RD [LAPTOP-NEB252RD\Spyros Despotis]'. The 'Progress' section shows 'Ready'. The 'Lift & Shift Your ETL Workload with SSIS in ADF' section provides instructions on creating an SSIS Integration Runtime (IR) in Azure Data Factory (ADF) and deploying SSIS projects into SSISDB. A 'Create SSIS IR' button is visible at the bottom.



Select Destination

Introduction
Select Source
Select Deployment Target
Select Destination
Review
Results

Help

Enter the destination server name and where the project will be located in the Integration Services

Browse for Project

Select a destination project for package deployment. Empty project could be created under selected folder.

SSISDB
liquor_etl
Update_dwh

New folder... New project... OK Cancel

Browse...
Connect
Browse...

Enter the name of the server instance that contains the Integration Services catalog and the path that specifies the location in the catalog where the packages should be deployed.

< Previous Next > Deploy Cancel



Results

Introduction
Select Source
Select Deployment Target
Select Destination
Review
Results

Help

Results

Action	Result
✓ Loading packages	Passed
✓ Connecting to destination server	Passed
✓ Changing packages protection level	Passed
✓ Deploying packages	Passed

Save Report...

< Previous Next > Close Cancel

Therefore we continued with enabling SQL Server Agent and create a new job named: Daily_Update, with step the execution of SSIS package, step name “Run DW ETL”. Afterwards we created a new job schedule.



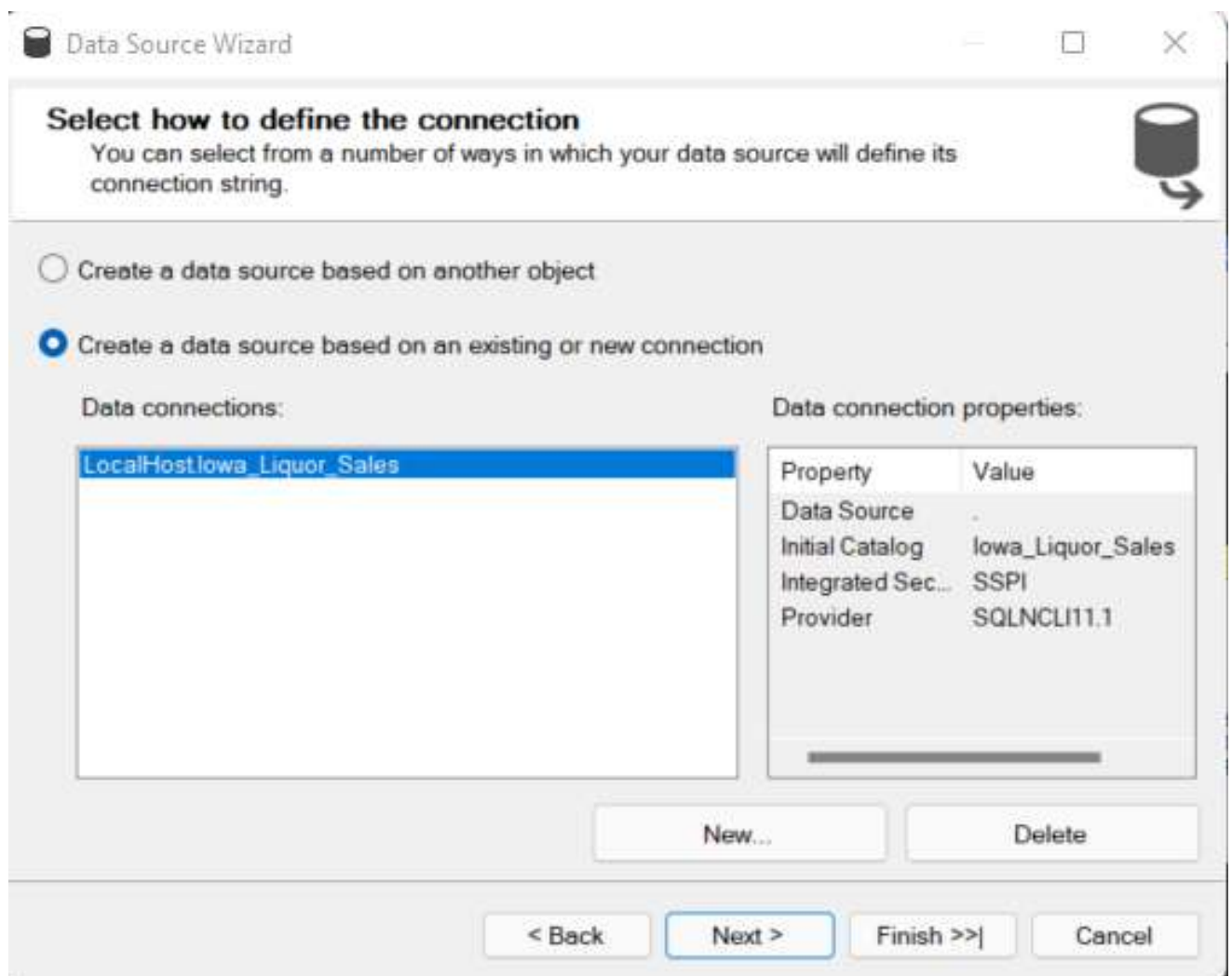
4. SSAS Architecture

4.1 SSAS Architecture

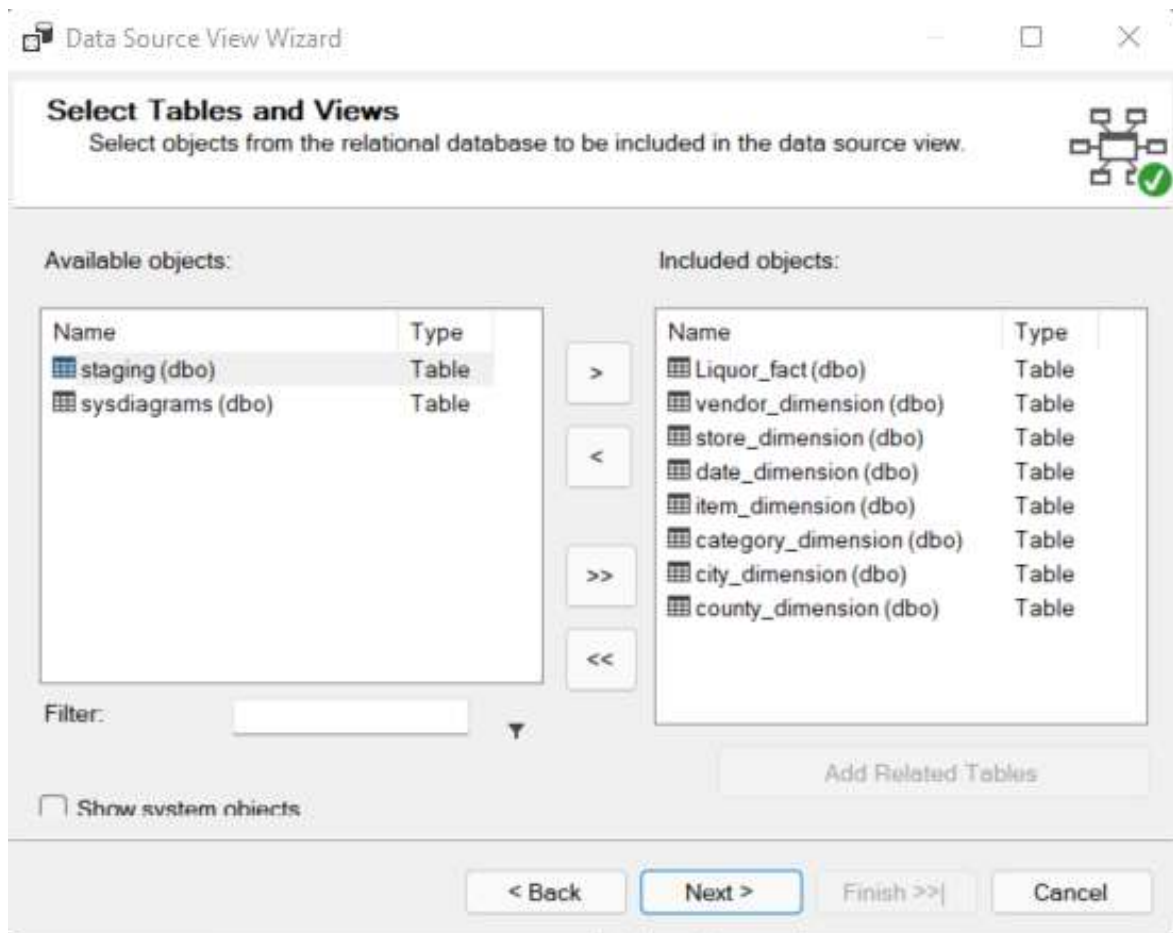
After creating our data warehouse and building the ETL data flow to update our dimension and fact tables, we are ready to define an OLAP Server cube which will be used for the reporting of our organization.

We have created a new Analysis Services Multidimensional Project named “Iowa Liquor Sales Multi”.

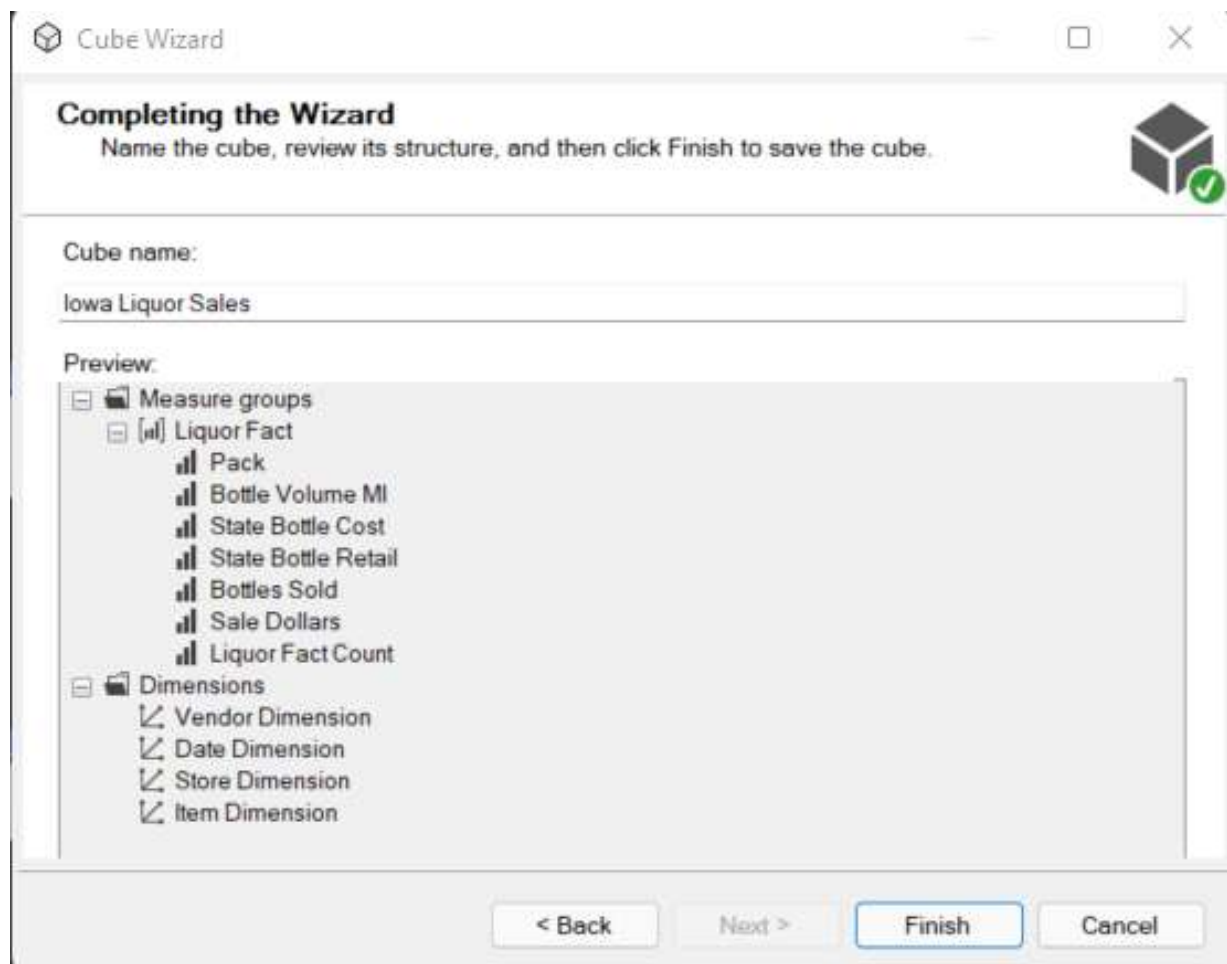
First, we define the “Data Source” to be the SQL-Server that we created locally.



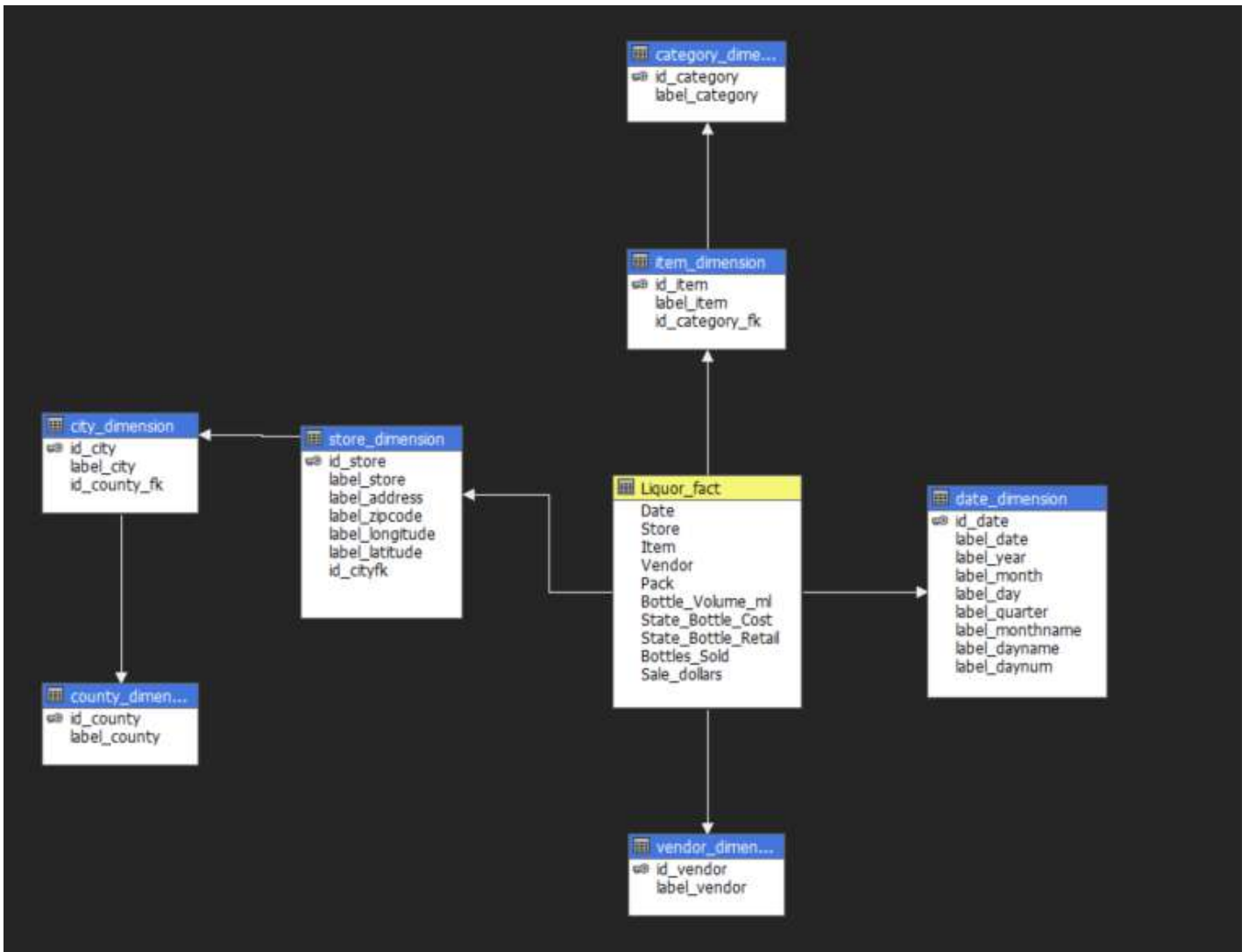
In the Data Source Views field we chose the tables that are needed in order to create the cube.



Finally in “Cubes and Dimensions” fields we determine the Fact and Dimension tables, as the picture above.



As a result we have the following **Cube**:



4.2 Calculated Members

We have built the following calculated members, using the above formulas:

- The price of each Liquor Item: $\text{Bottle Price} = \text{Sale Dollars} / \text{Bottles Sold}$

Name: [Bottle Price]

Parent Properties

Parent hierarchy: Measures

Parent member: [] Change

Expression

`ROUND([Measures].[Sale Dollars]/ [Measures].[Bottles Sold],2)`

Additional Properties

Format string: "Currency"

Visible: True

Non-empty behavior: []

Associated measure group: Liquor Fact

Display folder: []

Color Expressions

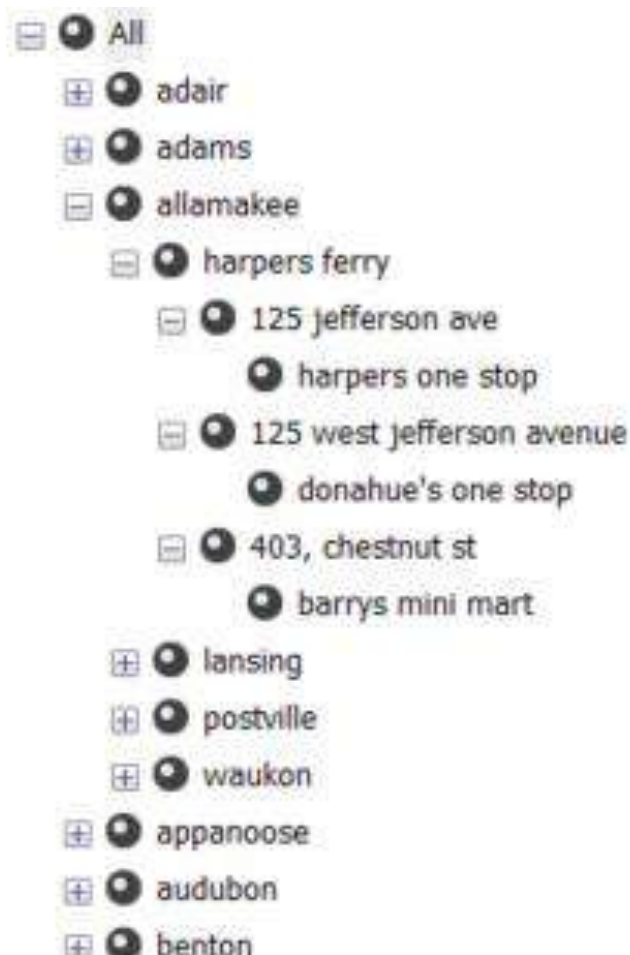
Font Expressions

4.3 Hierarchies

In our schema we have parent-child relationships, the next step is to define the Hierarchies. Starting from the Store Dimension we have defined the following hierarchy:



As we can see from the following output, in the first level we have all the Counties, in the second level the Cities that belong to the selected County, and finally the store addresses and names that are located in the selected County and City combination.



Following the Item Dimension we created the following hierarchy



Each Liquor Item is displayed under the Liquor Category that it belongs:

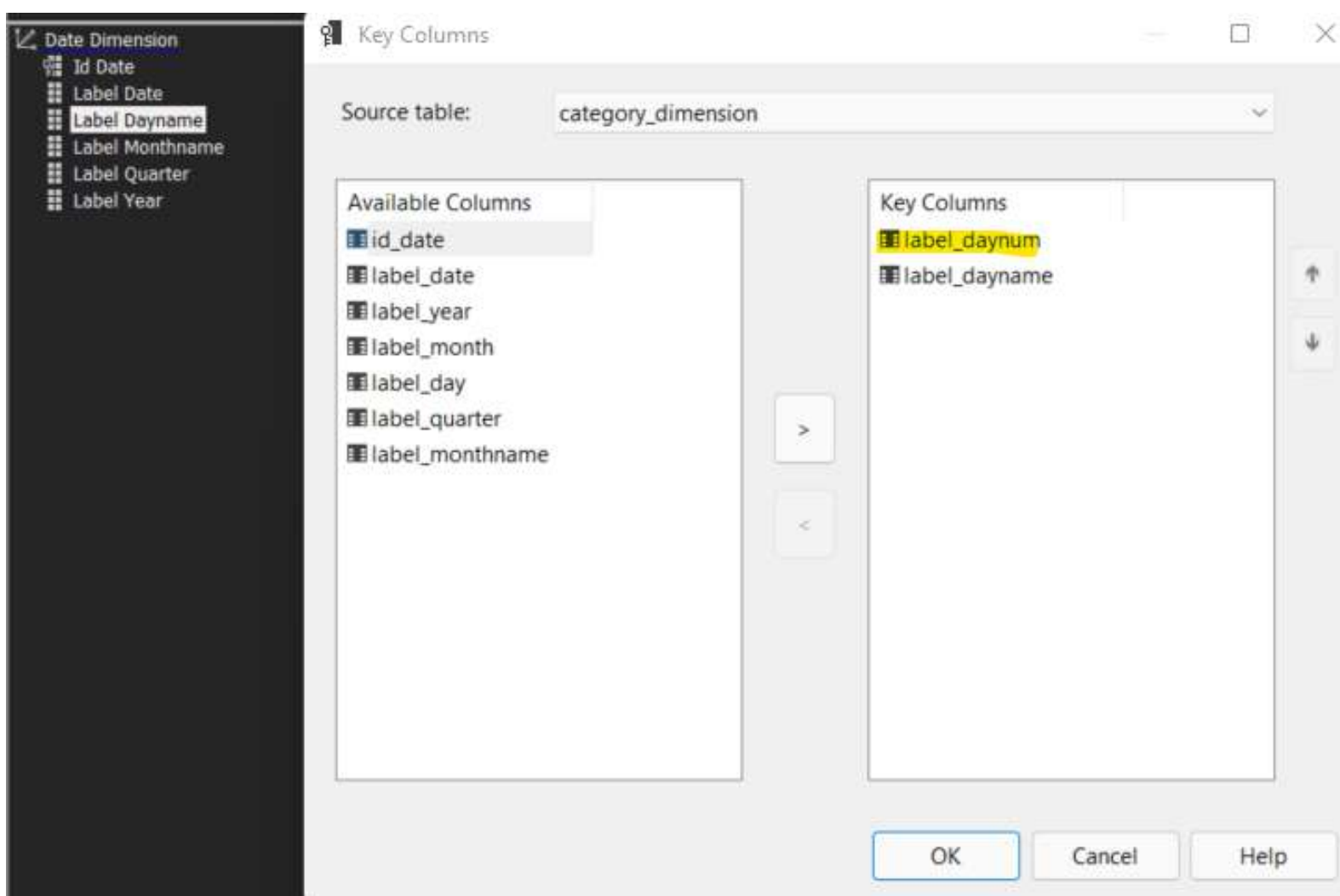


Finally, the Hierarchy for the date dimension is the following :



Before processing our cube we had to make some modifications, in order for the Label Monthname and the Label Dayname to be sorted correctly.

We added the label_daynum (a number from 1-7 ,based on the day of week) as a Key Column in the Label Dayname attribute. We then make it the first key column so that it is sorted properly.



Now that we have 2 key values, we should first define the 'NameColumn' field as the Label Dayname and the 'OrderBy' field as the Key.

Source	
CustomRollupColumn	(none)
CustomRollupPropertiesColumn	(none)
KeyColumns	(Collection)
NameColumn	date_dimension.label_dayname (WChar)
ValueColumn	(none)
IsAggregatable	True
OrderBy	Key
OrderByAttribute	

The output is the following:

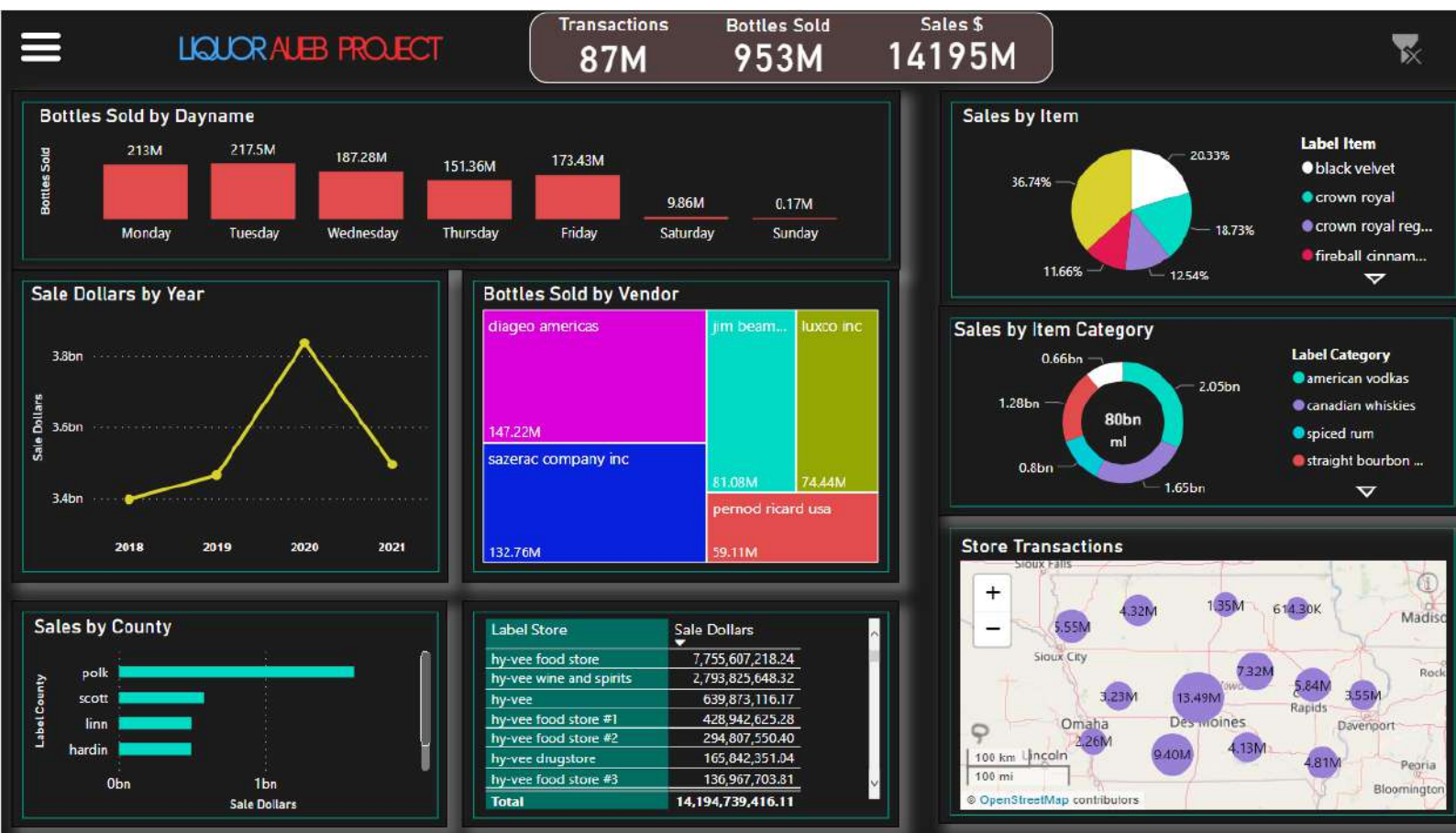


We followed the same process to order the month names correctly, and we used the `label_month` as the extra key column.

5. Visualizations with Power BI

5.1 Main Dashboard & Insights

The visualization are important to help us find answers in our guiding questions. So, we created a main dashboard in Power BI software, with 9 visualizations.





Specifically, we created:

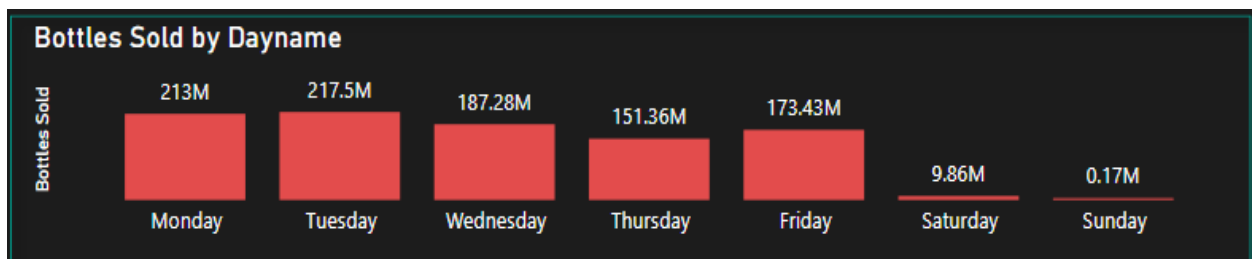
- An Advanced Card Visual with three metrics “**Transactions**”, “**Bottles Sold**” and “**Sales**” with the appropriate data fields
- A Stacked Column Chart visual named “**Bottles sold by Dayname**” which contains in Axis Label Dayname and as values bottles sold
- A line chart visual named “**Sale Dollars by Year**” with data in axis the hierarchy of date (Label Year, Label Monthname, Label Date) and data in values Sale dollars
- A clustered bar chart named “**Sales by County**” with data in axis the hierarchy of location (label county. Label city) and data in values the Sale Dollars. Also with filtered the top 5 counties-city with most sales by descending order
- A Teemap visual named “**Bottles Sold by Vendor**” with data in Group filed the label vendor and values the bottles sold. Also with filtered the top 5 vendors by Bottles Sold.
- A Table visual with data values the label store and sale dollars by descending order
- A Pie Chart named “**Sales by Item**” with values the Sale Dollars column and legent the label item. Also we filtered the top 5 items by sales
- A Donut Chart named “**Sales by Item Category**” with data the hierarchy of Category (Label Category, Label Item) and values the

Sale Dollars column. We filtered the top 5 item categories by sales. Also we placed an Advanced Card visual, inside the Donut Chart with the bottle volume ml metric

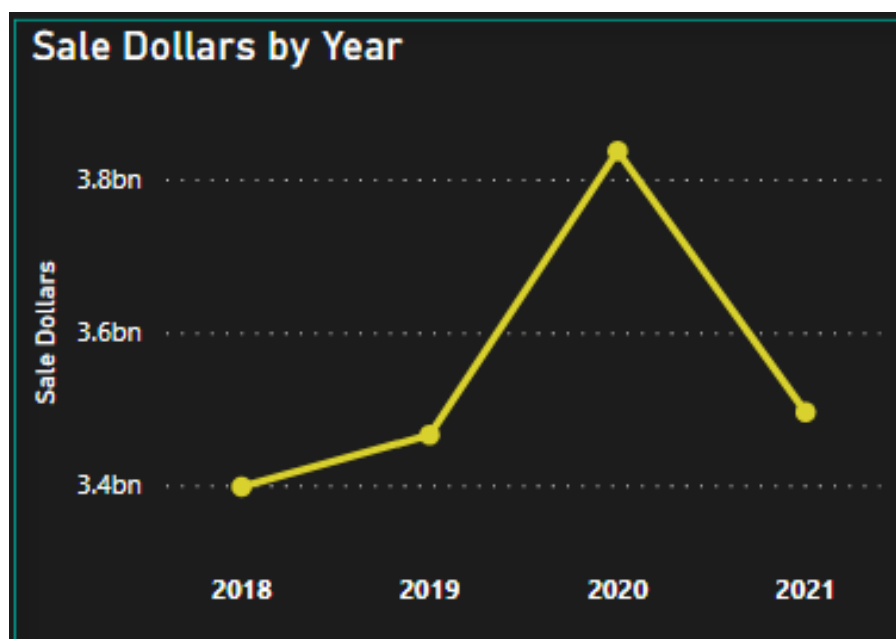
- A Drill Down Map Pro by ZoomCharts named “**Store Transactions**” with the appropriate latitude and longitude and as value the liquor fact count

Insights:

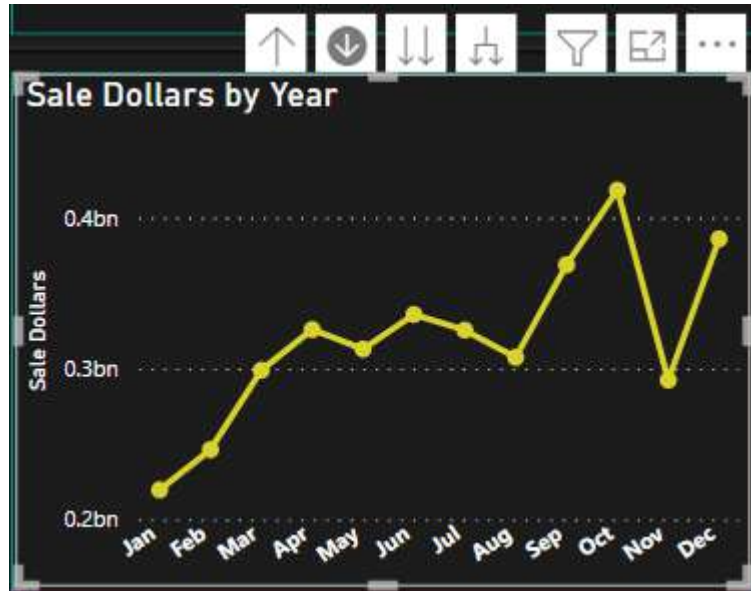
Due to local restrictions and legislation liquor stores on weekends are not able to place orders to their suppliers - vendors. So there is only a small amount of bottles sold in the weekends, in contrast with the days Monday and Tuesday in which we have the most sales because stores are in need of stock. Based on that, our company must be prepared to buy stock at the beginning of each week and not rely on the weekends.



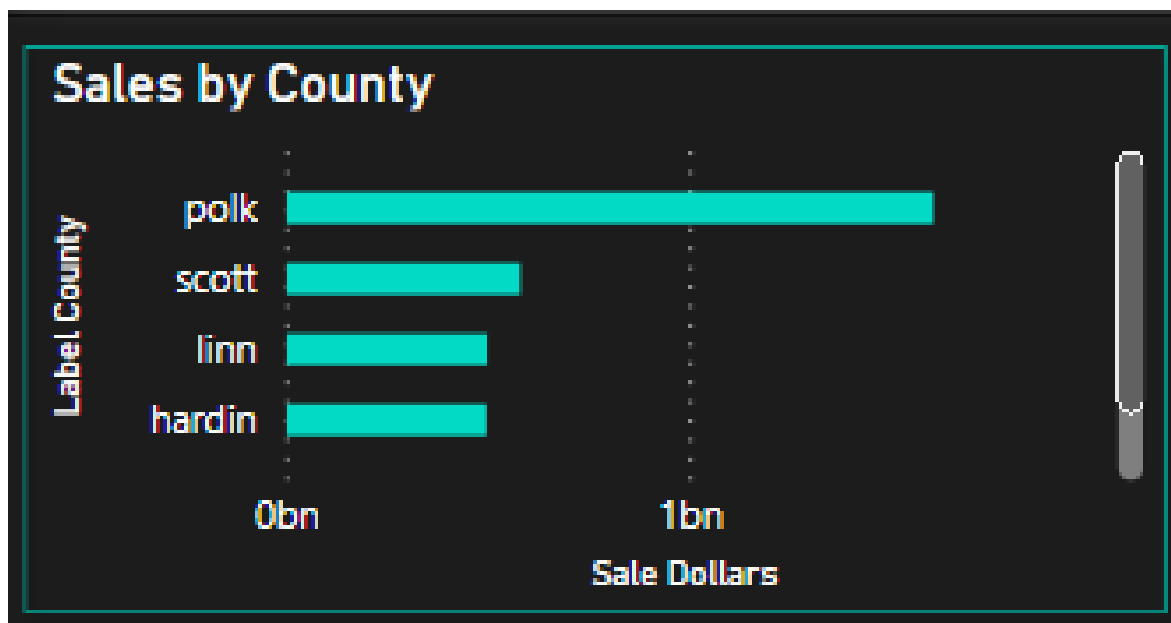
- From the years 2018 to 2019 liquor sales were in an uptrend. The year with the most sales was 2020 and after 2020 due to Covid-19 pandemic there was a drop in liquor sales.



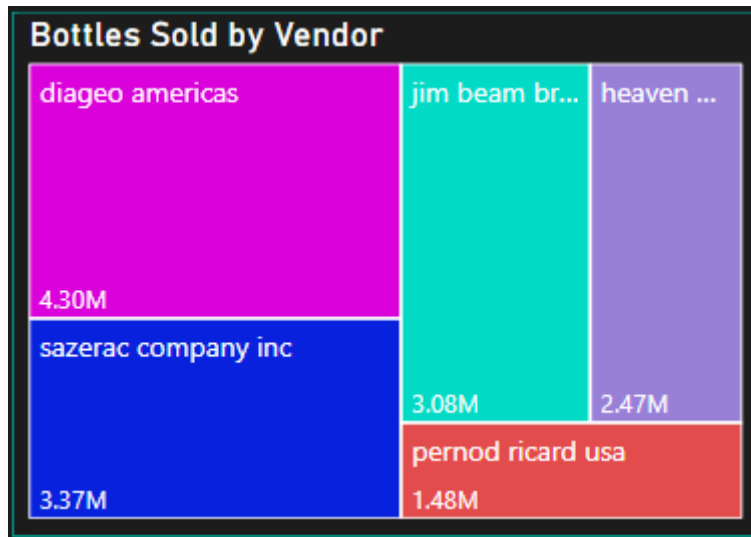
- The months with the most store sales are October and December. In October usually there are organised many festivals and in December due to Christmas Eve the liquor stores are selling more. So in order our company to improve its sales we have to run more campaigns and promotions especially that months.



- As we expected the biggest populated Counties have the biggest sales. For example Polk County has 214,133 population and it is the biggest populated County in the State of Iowa.



- The visual of Bottles Sold by Vendor helps us to understand which vendors supply the most the liquor in Iowa state. Therefore we can know the competitiveness between them and close deals in our advantage.



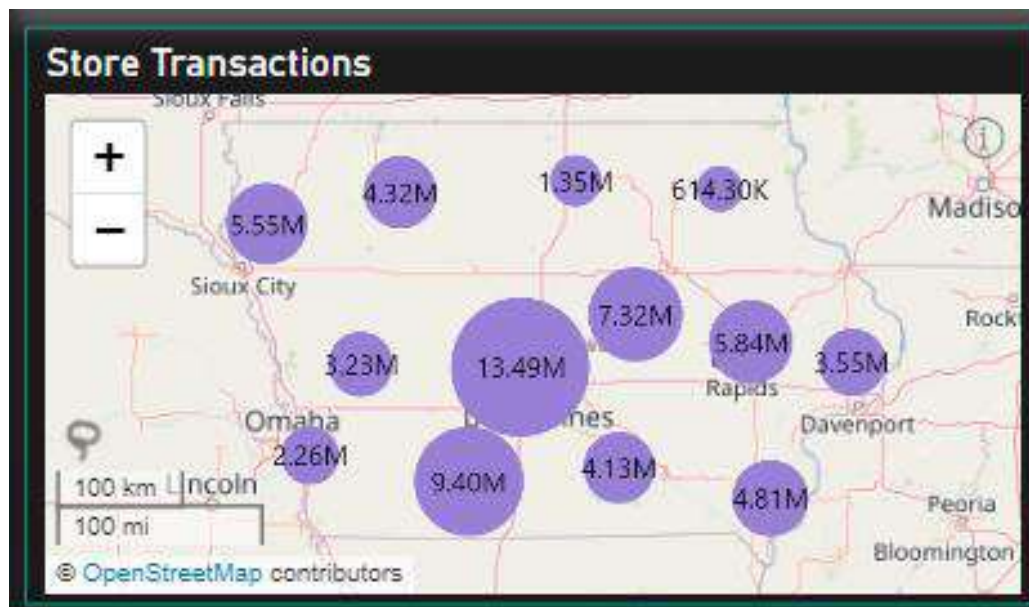
- Through Label Store table it clear that the Hy-vee company will be our biggest competitor since it has many stores with many sales around the Counties.

Label Store	Sale Dollars
hy-vee food store	7,755,607,218.24
hy-vee wine and spirits	2,793,825,648.32
hy-vee	639,873,116.17
hy-vee food store #1	428,942,625.28
hy-vee food store #2	294,807,550.40
hy-vee drugstore	165,842,351.04
hy-vee food store #3	136,967,703.81
Total	14,194,739,416.11

- Knowing the best-selling items and sales per Category, we can figure out which items and categories are the most popular and create our product range accordingly. For example from category American vodkas, the product titos handmade is the most preferable in Iowa State.

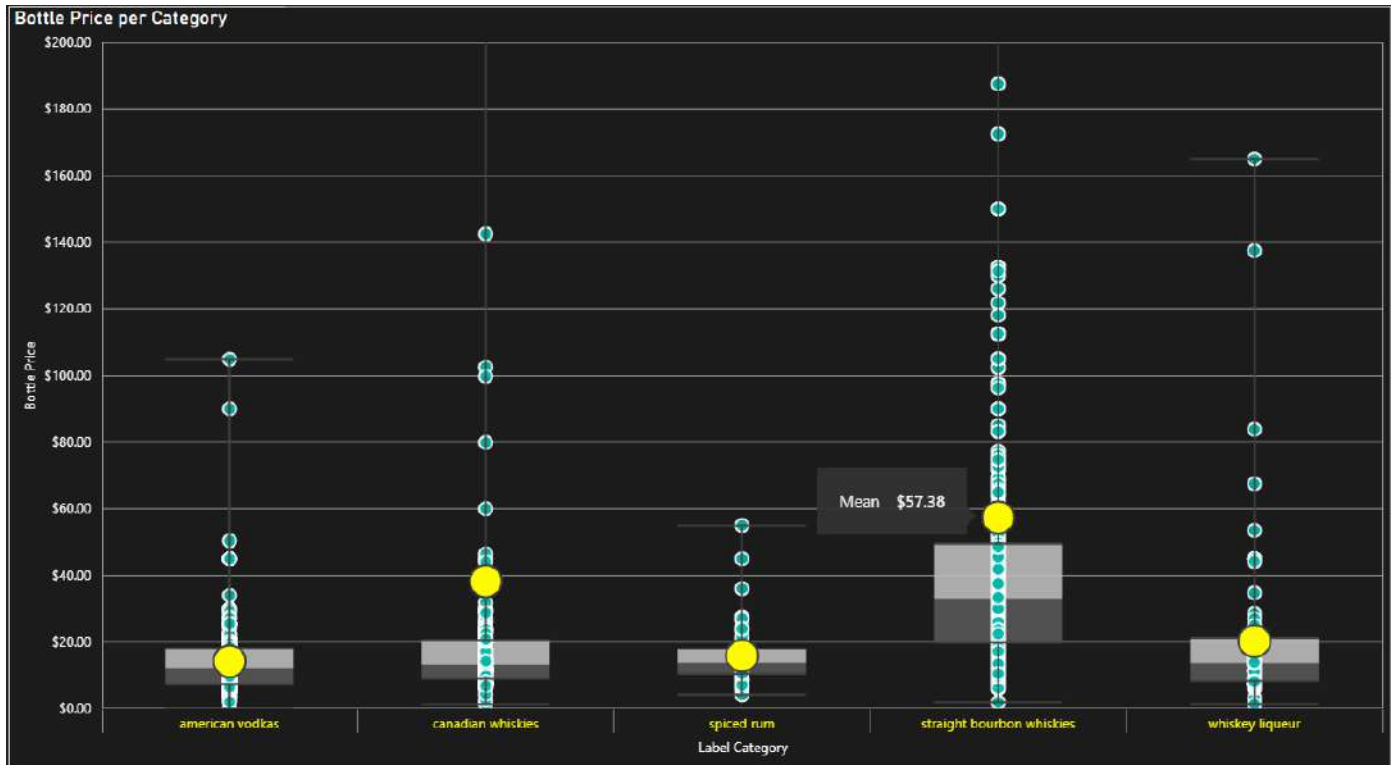


- From the visual Store Transactions, we have a mapping of the number of transactions of each store. As we expected the stores located to the most populated cities and counties have the highest number of transactions.



5.2 Bottle Price per Category Dashboard

It is clear that the mean bottle price of straight bourbon whiskies and Canadian whiskies is higher than the rest of the categories. Also the more top shelf a brand is, the more variation we see in the pricing.



5.3 Bottle Price by County Dashboard

In the below graph, firstly we can observe that across all counties, on average, bottles of liquor will be sold for around \$15, with most liquor bottle prices ranging from \$14 - \$16. Secondly we can see that the top 3 counties by sales have the least median bottle price.

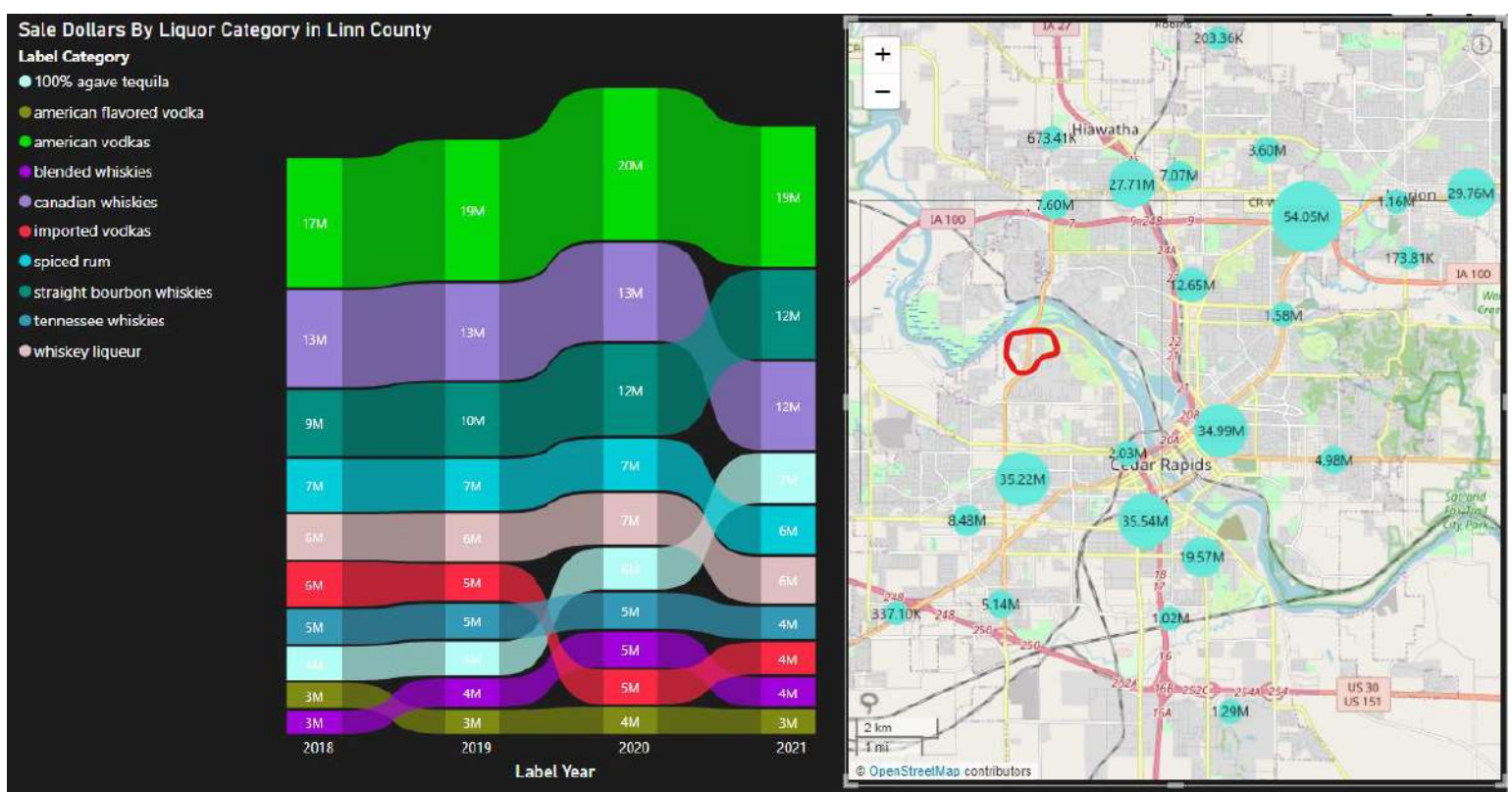


Last but not least, we can see that the Linn county is the most ideal place to locate our company for the following reasons:

- it has one of the lowest median bottle price
- it belongs to the top 3 counties with the most sales
- it is the second most populated county in Iowa State

5.4 Sale Dollars By Liquor Category in Linn County Dashboard

With the below visualizations we can specify our company location inside the Linn city. We believe that the red circle in the map is the ideal location for our shop due to the fact that it is in the center area of the city and there are not many competitors very close.



Moreover, it is important for our company to be well stocked with the American vodkas and straight bourbon whiskies in order to respond to the demand of the Christmas season.

