My Data Analytics Journey at Cox Automotive

BA Basu, Abhi (CAI - Austin)
Principal Technical Architect

Published 2/26/2020

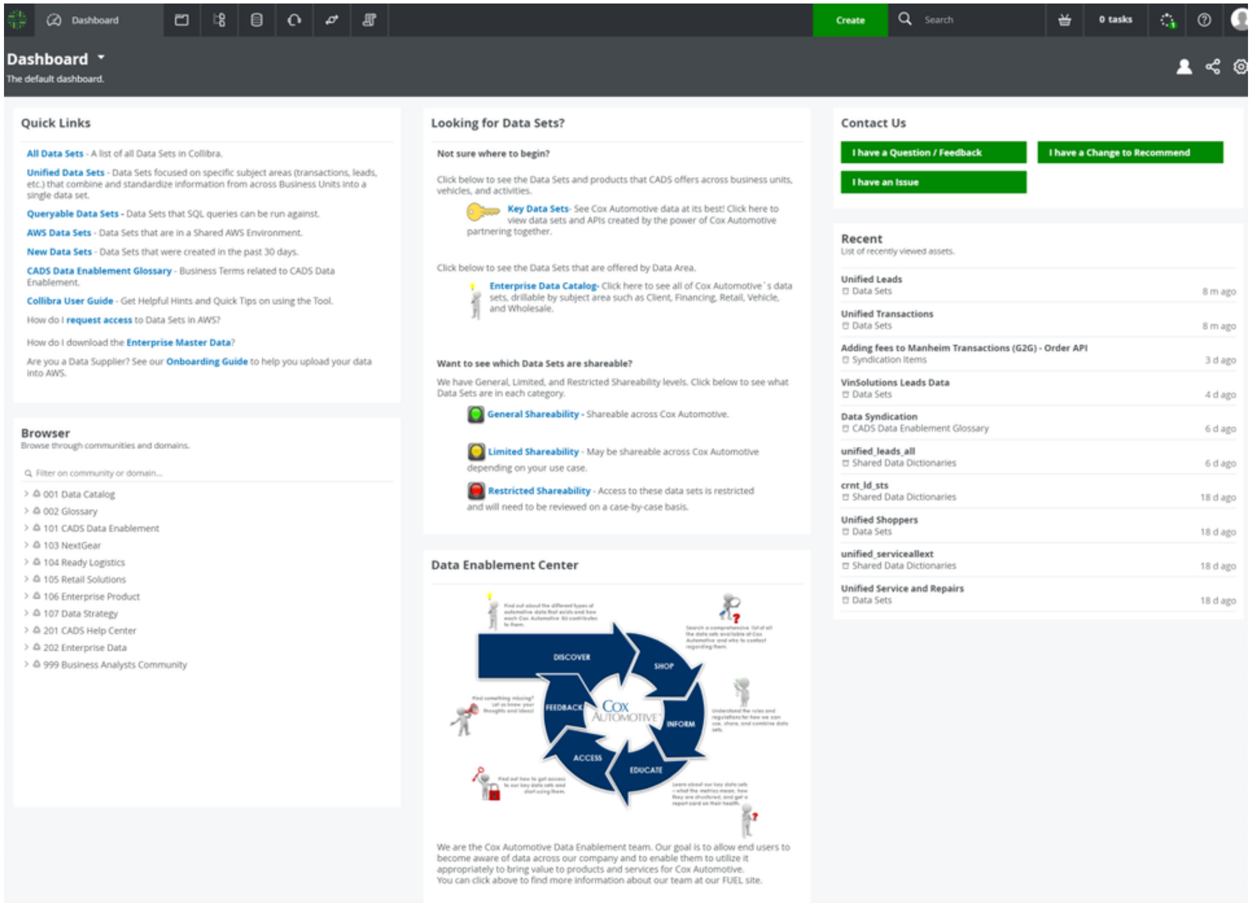
I joined Cox Automotive on September, 2019, as a Principal Technical Architect for the Retail Analytics Release Train, within the Common Retail Services Delivery Stream. Mouthful, right? 😊 Over the last 8 years of my career, I have focused on (Big) Data Architecture, Data Analytics and Data Science areas. Given that my primary charter is to work on Cross-BU analytics, I thought the best place to start would be to familiarize myself with the various Business Unit delivered solutions and data sets currently available through the Enterprise Data Platform (or the Corporate Data Lake).

I will attempt to describe the steps required to be able to find, request and access data sets that can be used for analysis. I will also describe, using a specific example, on how to access a Unified Dataset (so that you may follow my journey easily) and how to use various tools and techniques to perform data exploration and analysis.

Step 1 – Access to Corporate Data Catalog (Collibra)

The first place to start would be Collibra, which is the Corporate Data Catalog and Data Governance tool. Even before that, you would need to go to the Service Portal and request access to the Enterprise Data Catalog (Collibra) here: https://coxauto.service-now.com/sp_technology/?id=sc_cat_item&sys_id=b6f9505fdbb9bbc8afe7e33648961954. Collibra datasets come from the various BUs and there are some Unified (aggregated) data sets that are very valuable as the Data Scientists in the Data Enablement Team have worked hard to provide useful transformations and aggregations of the raw datasets published by various Business Units. Once you have obtained Collibra access, you can explore from here: (https://coxautoinc.sharepoint.com/sites/service-collibra). The Dashboard looks like the screen shot below.

Skip to main content



Dashboard
The default dashboard.

Quick Links

All Data Sets - A list of all Data Sets in Collibra.

Unified Data Sets - Data Sets focused on specific subject areas (transactions, leads, etc.) that combine and standardize information from across Business Units into a single data set.

Queryable Data Sets - Data Sets that SQL queries can be run against.

AWS Data Sets - Data Sets that are in a Shared AWS Environment.

New Data Sets - Data Sets that were created in the past 30 days.

CADS Data Enablement Glossary - Business Terms related to CADS Data Enablement.

Collibra User Guide - Get Helpful Hints and Quick Tips on using the Tool.

How do I **request access** to Data Sets in AWS?

How do I download the **Enterprise Master Data**?

Are you a Data Supplier? See our **Onboarding Guide** to help you upload your data into AWS.

Browser
Browse through communities and domains.

Filter on community or domain...

- 001 Data Catalog
- 002 Glossary
- 101 CADS Data Enablement
- 103 NextGear
- 104 Ready Logistics
- 105 Retail Solutions
- 106 Enterprise Product
- 107 Data Strategy
- 201 CADS Help Center
- 202 Enterprise Data
- 999 Business Analysts Community

Looking for Data Sets?

Not sure where to begin?

Click below to see the Data Sets and products that CADS offers across business units, vehicles, and activities.

Key Data Sets- See Cox Automotive data at its best! Click here to view data sets and APIs created by the power of Cox Automotive partnering together.

Click below to see the Data Sets that are offered by Data Area.

Enterprise Data Catalog- Click here to see all of Cox Automotive`s data sets, drillable by subject area such as Client, Financing, Retail, Vehicle, and Wholesale.

Want to see which Data Sets are shareable?

We have General, Limited, and Restricted Shareability levels. Click below to see what Data Sets are in each category.

🟢 **General Shareability** - Shareable across Cox Automotive.

🟡 **Limited Shareability** - May be shareable across Cox Automotive depending on your use case.

🔴 **Restricted Shareability** - Access to these data sets is restricted and will need to be reviewed on a case-by-case basis.

Data Enablement Center

Find out about the different types of automotive data that exists and how each Cox Automotive BU contributes to them.
Search a comprehensive list of all the data sets available at Cox Automotive and who to contact regarding them.
DISCOVER
SHOP
FEEDBACK
COX AUTOMOTIVE
INFORM
ACCESS
EDUCATE
Find something missing? Let us know your thoughts and ideas!
Understand the rules and regulations for how we can use, share, and combine data sets.
Find out how to get access to our key data sets and start using them.
Learn about our key data sets - what the metrics mean, how they are structured, and get a report card on their health.

We are the Cox Automotive Data Enablement team. Our goal is to allow end users to become aware of data across our company and to enable them to utilize it appropriately to bring value to products and services for Cox Automotive.
You can click above to find more information about our team at our FUEL site.

Contact Us

I have a Question / Feedback | I have a Change to Recommend | I have an Issue

Recent
List of recently viewed assets.

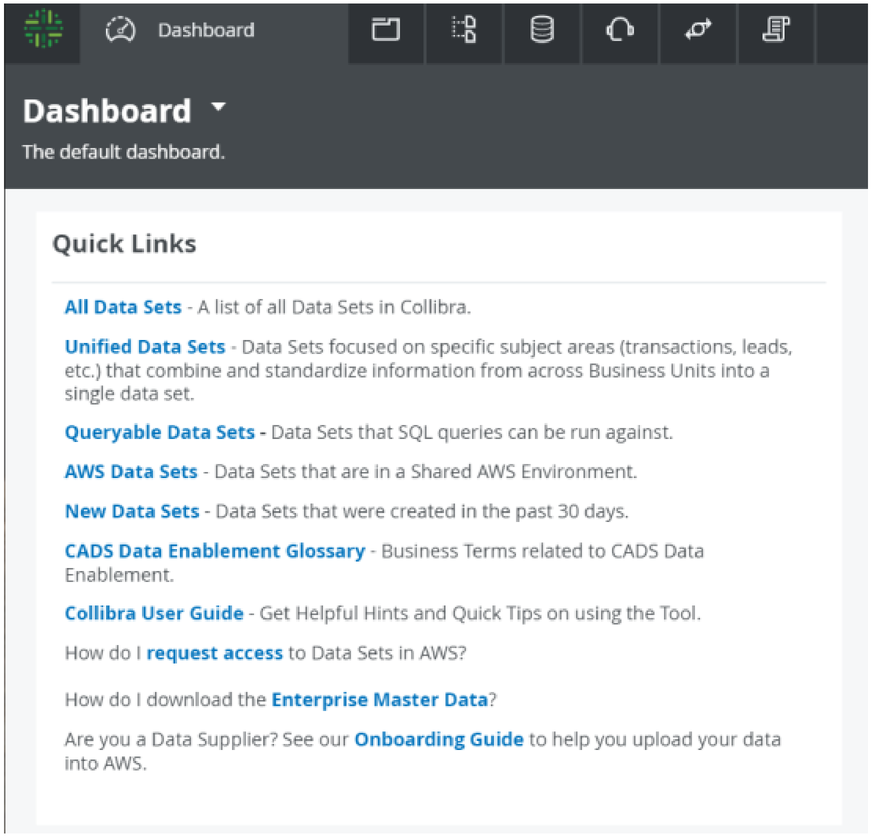
Unified Leads Data Sets	8 m ago
Unified Transactions Data Sets	8 m ago
Adding fees to Manheim Transactions (G2G) - Order API Syndication Items	3 d ago
VinSolutions Leads Data Data Sets	4 d ago
Data Syndication CADS Data Enablement Glossary	6 d ago
unified_leads_all Shared Data Dictionaries	6 d ago
crnt_ld_sts Shared Data Dictionaries	18 d ago
Unified Shoppers Data Sets	18 d ago
unified_serviceallext Shared Data Dictionaries	18 d ago
Unified Service and Repairs Data Sets	18 d ago

It is advisable to go through the Quick User Guide here at this time : https://coxautoinc.sharepoint.com/:w:/r/sites/service-collibra/_layouts/15/Doc.aspx?sourcedoc=%7BDF9D77E5-8335-4DA6-B8EF-BB69C2266CD0%7D&file=Collibra%20Data%20Enablement%20Center%20Quick%20User%20Guide.docx&action=default&mobileredirect=true&DefaultItemOpen=1.

Step 2 – Finding and Requesting Data from Collibra

Collibra provides various ways of searching for datasets and examining their data catalog and other details (Quick User guide will explain usage). For my use case, I went to the “Unified Datasets” section from the Collibra Dashboard and selected the Unified Transactions dataset (https://cai-prod.collibra.com/asset/2e7ffcca-354c-4bc2-a8da-204a368cfb2b).

Skip to main content



Dashboard

The default dashboard.

Quick Links

All Data Sets - A list of all Data Sets in Collibra.

Unified Data Sets - Data Sets focused on specific subject areas (transactions, leads, etc.) that combine and standardize information from across Business Units into a single data set.

Queryable Data Sets - Data Sets that SQL queries can be run against.

AWS Data Sets - Data Sets that are in a Shared AWS Environment.

New Data Sets - Data Sets that were created in the past 30 days.

CADS Data Enablement Glossary - Business Terms related to CADS Data Enablement.

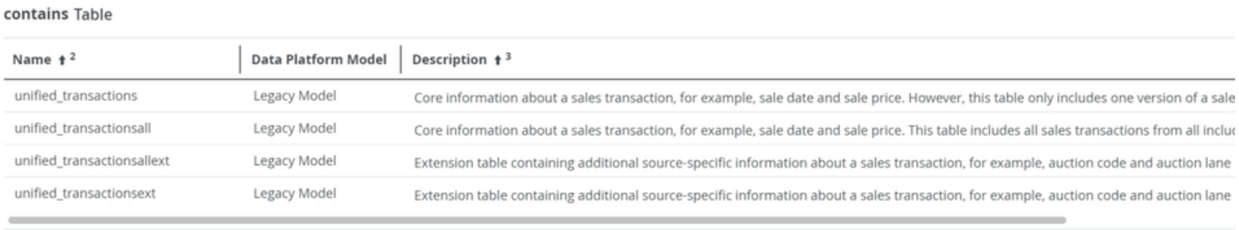
Collibra User Guide - Get Helpful Hints and Quick Tips on using the Tool.

How do I **request access** to Data Sets in AWS?

How do I download the **Enterprise Master Data**?

Are you a Data Supplier? See our **Onboarding Guide** to help you upload your data into AWS.

The Unified Transactions page will have a lot of information, but notable items are the Business and Technical Points of Contact, Table names and various Data Policies attached to the dataset. The Technical Points of Contact are our colleagues from the Data Enablement Team (under Data Solution Team) that crawl the data provided by the source systems so that they can produce the data dictionaries for people to know what tables and columns exist, and spend a lot of time researching and aggregating various element to provide the Unified datasets. Any questions about these datasets should be directed to the Technical Contact first and if needed they would be forwarded to the Business Contact (source data experts).

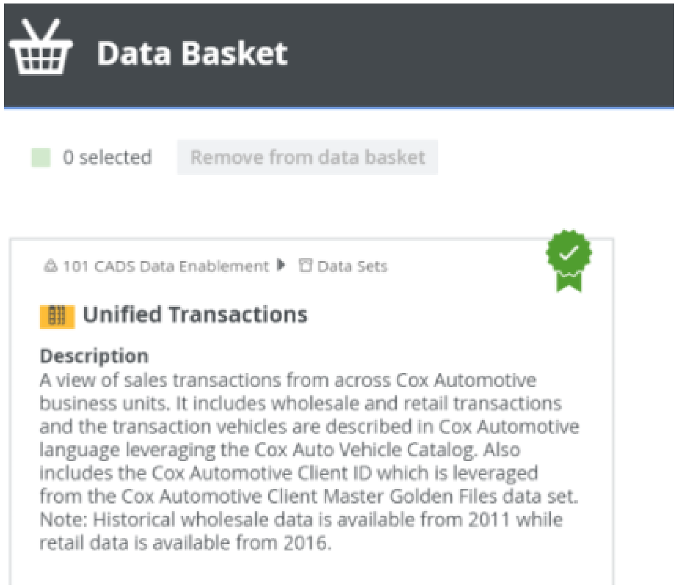


contains Table

Name ↑ 2	Data Platform Model	Description ↑ 3
unified_transactions	Legacy Model	Core information about a sales transaction, for example, sale date and sale price. However, this table only includes one version of a sale
unified_transactionsall	Legacy Model	Core information about a sales transaction, for example, sale date and sale price. This table includes all sales transactions from all inclu
unified_transactionsallext	Legacy Model	Extension table containing additional source-specific information about a sales transaction, for example, auction code and auction lane
unified_transactionsext	Legacy Model	Extension table containing additional source-specific information about a sales transaction, for example, auction code and auction lane

After consulting with our Data Enablement team, I wanted to request access to the Unified Transaction dataset. Click on the "Add to Data Basket" button on top section of the Collibra site to add this dataset to the Cart. You will find the dataset available in the Data Basket to Checkout and start the request process.

Skip to main content



Unified Transactions

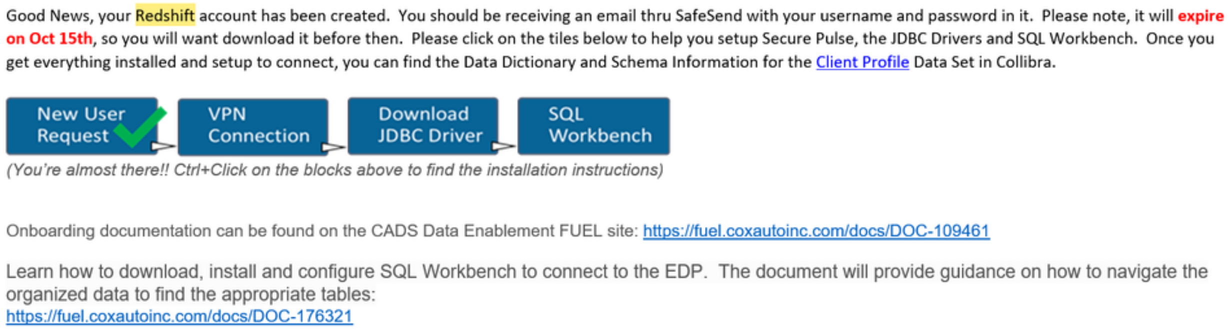
Description

A view of sales transactions from across Cox Automotive business units. It includes wholesale and retail transactions and the transaction vehicles are described in Cox Automotive language leveraging the Cox Auto Vehicle Catalog. Also includes the Cox Automotive Client ID which is leveraged from the Cox Automotive Client Master Golden Files data set. Note: Historical wholesale data is available from 2011 while retail data is available from 2016.

Clicking on the above item in the Data Basket will initiate a set of pop-up windows that will provide information about data access guidelines and mechanism to access the data, i.e. AWS S3 bucket of Redshift DB (ability to query). After consulting with my Data Enablement Team contact and understanding the size of the dataset, I selected Queryable access, so I would be able to filter the data as required.

Step 3 – Accessing the Data Set(s)

Once the data access is approved, you will receive an email with directions. You must follow the outlined steps to be able to log into Redshift DB to look at the approved data set(s).


Good News, your Redshift account has been created. You should be receiving an email thru SafeSend with your username and password in it. Please note, it will **expire on Oct 15th**, so you will want download it before then. Please click on the tiles below to help you setup Secure Pulse, the JDBC Drivers and SQL Workbench. Once you get everything installed and setup to connect, you can find the Data Dictionary and Schema Information for the Client Profile Data Set in Collibra.

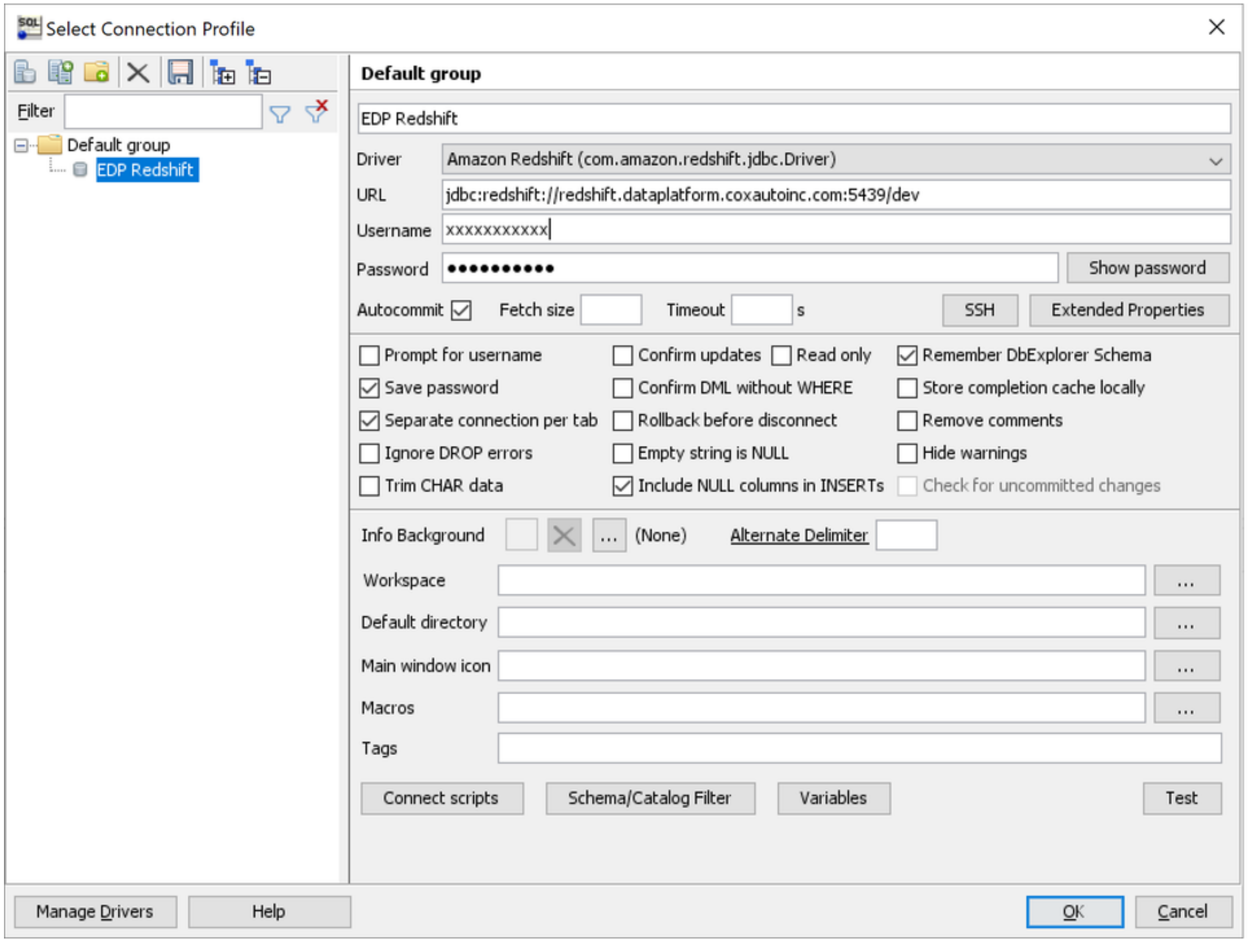
(You're almost there!! Ctrl+Click on the blocks above to find the installation instructions)

Onboarding documentation can be found on the CADS Data Enablement FUEL site: https://fuel.coxautoinc.com/docs/DOC-109461

Learn how to download, install and configure SQL Workbench to connect to the EDP. The document will provide guidance on how to navigate the organized data to find the appropriate tables:
https://fuel.coxautoinc.com/docs/DOC-176321

Redshift DB needs VPN to be able to connect. The above information will describe how to install and configure Pulse Secure VPN on your computer. You can use the suggested GUI tool like SQL Workbench to access and query the datasets. SQL Workbench requires a JDBC connection to be able to connect with Redshift DB. The

Skip to main content



Default group

EDP Redshift

Driver: Amazon Redshift (com.amazon.redshift.jdbc.Driver)

URL: jdbc:redshift://redshift.dataplatform.coxautoinc.com:5439/dev

Username: xxxxxxxxxxx

Password: ●●●●●●●●●●

Autocommit ☑ Fetch size Timeout s

- ☐ Prompt for username
- ☑ Save password
- ☑ Separate connection per tab
- ☐ Ignore DROP errors
- ☐ Trim CHAR data
- ☐ Confirm updates
- ☐ Read only
- ☐ Confirm DML without WHERE
- ☐ Rollback before disconnect
- ☐ Empty string is NULL
- ☑ Include NULL columns in INSERTs
- ☑ Remember DbExplorer Schema
- ☐ Store completion cache locally
- ☐ Remove comments
- ☐ Hide warnings
- ☐ Check for uncommitted changes

Info Background (None)

Workspace

Default directory

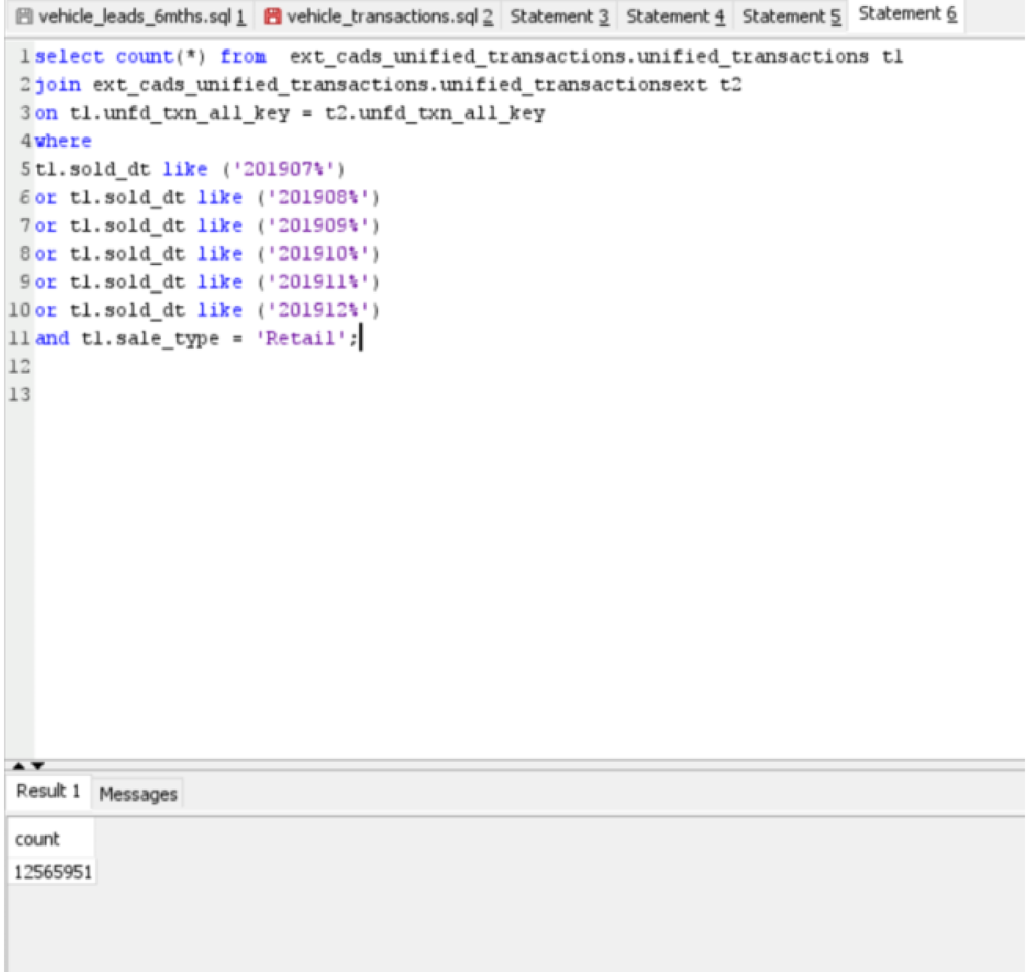
Main window icon

Macros

Tags

Below is an example query joining the two Unified Transactions tables and filtering by the last 6 months of 2019 and only Retail sales. This form of query capability would allow me to estimate the size of data I may need to work with. NOTE: Redshift provided Read-Only access now, so there is no easy way to create stage tables of data for easier access using a remote client for ease of data analysis. In subsequent steps, I will describe how I was able to extract the data using Python locally for data analysis and visualization work.

Skip to main content



```
select count(*) from  ext_cads_unified_transactions.unified_transactions t1
join ext_cads_unified_transactions.unified_transactionsext t2
on t1.unfd_txn_all_key = t2.unfd_txn_all_key
where
t1.sold_dt like ('201907%')
or t1.sold_dt like ('201908%')
or t1.sold_dt like ('201909%')
or t1.sold_dt like ('201910%')
or t1.sold_dt like ('201911%')
or t1.sold_dt like ('201912%')
and t1.sale_type = 'Retail';
```

count
12565951

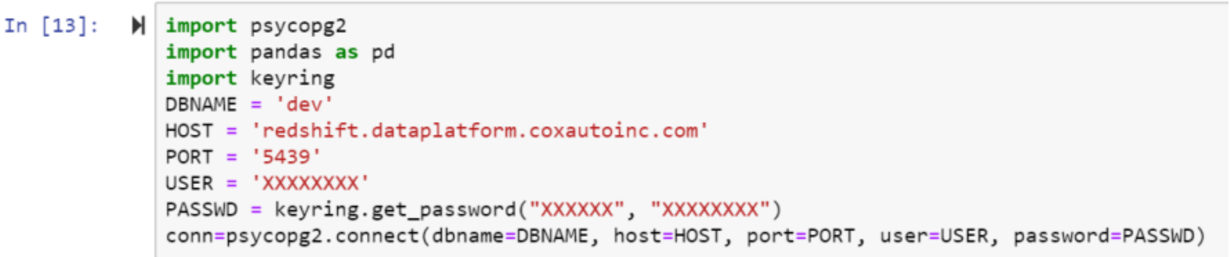
Step 4 – Extracting Data Sets (Locally or onto alternate Data Warehouse)

In the future, Snowflake Data Warehouse will be available to us with read-write access, so we would be able to create temporary tables (and views) for easier access and manipulation. Since the Data Platform team is working on this at present, the most feasible way to work with data sets currently is to extract to local computer and perform ETL and data transformations there. I have installed Anaconda Python distribution (https://www.anaconda.com/distribution/) which provides base Python (3.x) bundled with numerous useful scientific and data analyses libraries. Anaconda also bundles Jupyter Notebook (https://jupyter.org/) which can be run easily from command line once Anaconda is installed on your system. One big advantage of using Jupyter Notebook is the ability to share the code and transformations with other team members and collaborate in-line. Python3 provides libraries to connect to all types of data storage systems.

Here is an example of code I used to extract a dataset onto my local system using Jupyter Notebook and Python 3. This adapter is needed to connect with Redshift DB (https://pypi.org/project/psycopg2/).

Making Connection to Redshift DB

Skip to main content



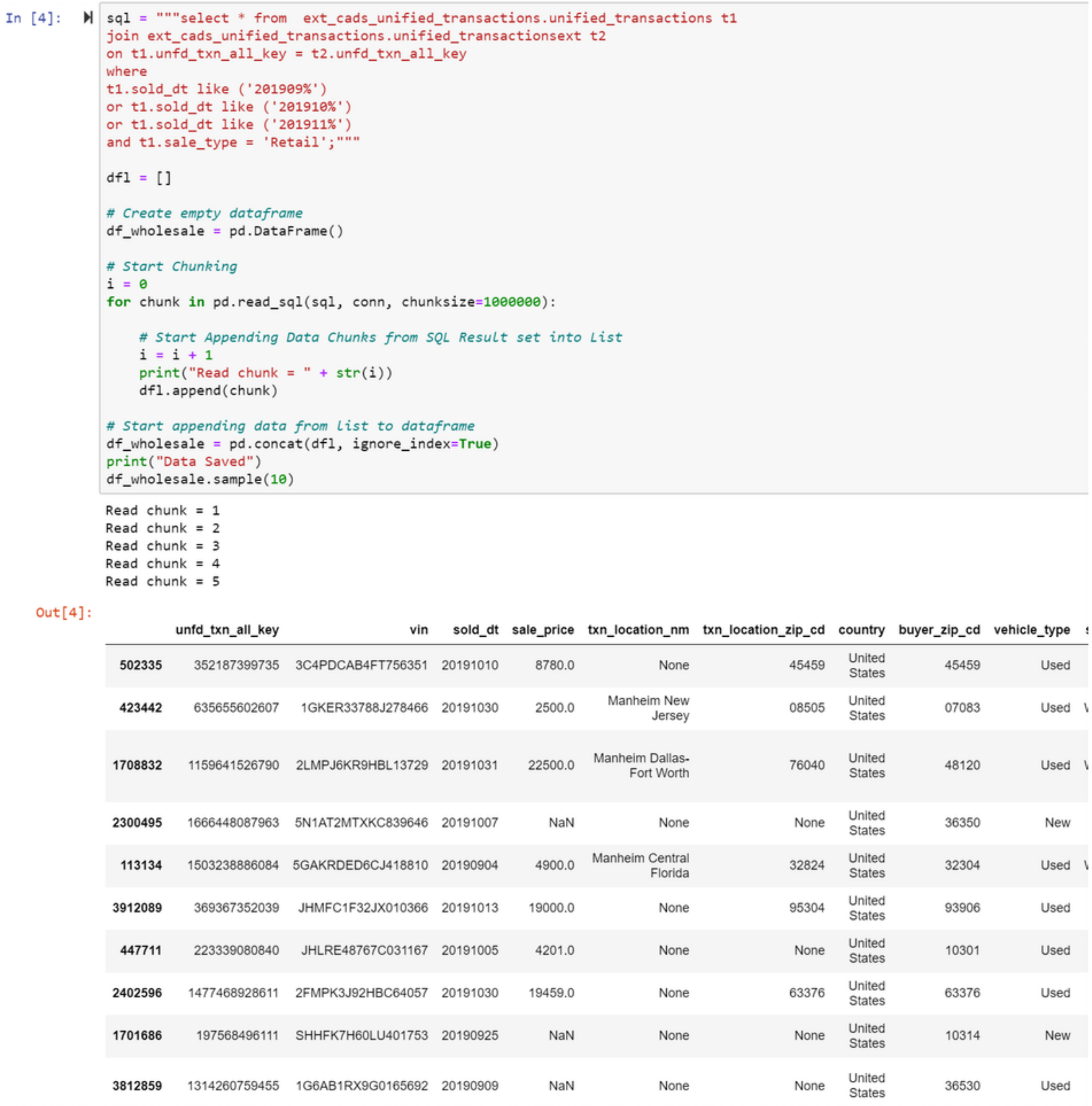
```
import psycopg2
import pandas as pd
import keyring
DBNAME = 'dev'
HOST = 'redshift.dataplatform.coxautoinc.com'
PORT = '5439'
USER = 'XXXXXXXX'
PASSWD = keyring.get_password("XXXXXX", "XXXXXXXX")
conn=psycopg2.connect(dbname=DBNAME, host=HOST, port=PORT, user=USER, password=PASSWD)
```

Note: The Keyring python library is used above as an safe way to store passwords (https://pypi.org/project/keyring/).

Executing Query Against Redshift DB

Querying the Redshift DB and using numpy and pandas libraries to chunk through the dataset and write to a pandas dataframe (https://www.datacamp.com/community/tutorials/pandas-tutorial-dataframe-python).

Skip to main content



In [4]:

```
sql = """select * from  ext_cads_unified_transactions.unified_transactions t1
join ext_cads_unified_transactions.unified_transactionsext t2
on t1.unfd_txn_all_key = t2.unfd_txn_all_key
where
t1.sold_dt like ('201909%')
or t1.sold_dt like ('201910%')
or t1.sold_dt like ('201911%')
and t1.sale_type = 'Retail';"""

dfl = []

# Create empty dataframe
df_wholesale = pd.DataFrame()

# Start Chunking
i = 0
for chunk in pd.read_sql(sql, conn, chunksize=1000000):

    # Start Appending Data Chunks from SQL Result set into List
    i = i + 1
    print("Read chunk = " + str(i))
    dfl.append(chunk)

# Start appending data from list to dataframe
df_wholesale = pd.concat(dfl, ignore_index=True)
print("Data Saved")
df_wholesale.sample(10)
```

```
Read chunk = 1
Read chunk = 2
Read chunk = 3
Read chunk = 4
Read chunk = 5
```

Out[4]:

	unfd_txn_all_key	vin	sold_dt	sale_price	txn_location_nm	txn_location_zip_cd	country	buyer_zip_cd	vehicle_type	s
502335	352187399735	3C4PDCAB4FT756351	20191010	8780.0	None	45459	United States	45459	Used	
423442	635655602607	1GKER33788J278466	20191030	2500.0	Manheim New Jersey	08505	United States	07083	Used	V
1708832	1159641526790	2LMPJ6KR9HBL13729	20191031	22500.0	Manheim Dallas-Fort Worth	76040	United States	48120	Used	V
2300495	1666448087963	5N1AT2MTXKC839646	20191007	NaN	None	None	United States	36350	New	
113134	1503238886084	5GAKRDED6CJ418810	20190904	4900.0	Manheim Central Florida	32824	United States	32304	Used	V
3912089	369367352039	JHMFC1F32JX010366	20191013	19000.0	None	95304	United States	93906	Used	
447711	223339080840	JHLRE48767C031167	20191005	4201.0	None	None	United States	10301	Used	
2402596	1477468928611	2FMPK3J92HBC64057	20191030	19459.0	None	63376	United States	63376	Used	
1701686	197568496111	SHHFK7H60LU401753	20190925	NaN	None	None	United States	10314	New	
3812859	1314260759455	1G6AB1RX9G0165692	20190909	NaN	None	None	United States	36530	Used	

Saving Dataset Locally

Once the data is filtered to our needs, we can easily save it as a csv file locally. This allows us to cut off connection to RedShift DB (which can be slow at times) and work on this dataset locally.

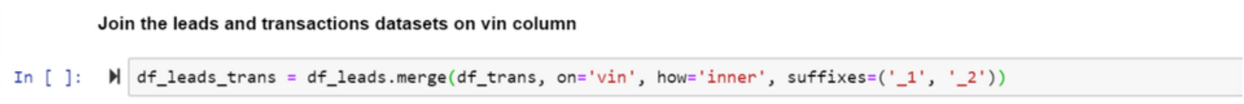
Skip to main content

Step 5 – Performing Data Exploration, ETL and Analysis

Now you have a free hand at cleaning up, categorizing, supplementing and transforming the dataset as per your requirements.

Below I show some examples of this type of work (while leaving out some details) like joining two datasets (Unified Leads and Unified Transactions) and categorizing some fields for easy of visualization.

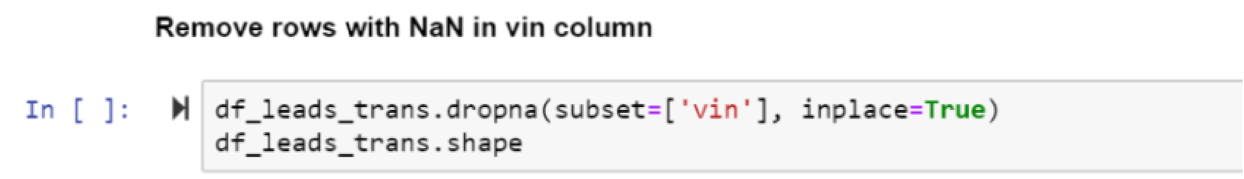
Joining Datasets



Join the leads and transactions datasets on vin column

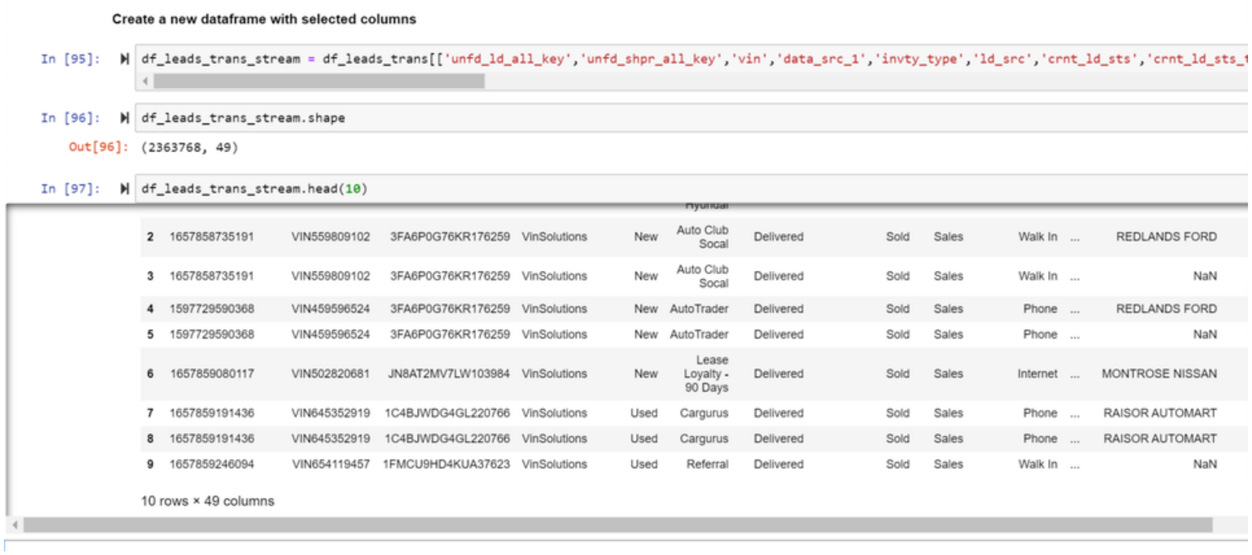
```
In [ ]:  df_leads_trans = df_leads.merge(df_trans, on='vin', how='inner', suffixes=('_1', '_2'))
```

Some Cleanup



Remove rows with NaN in vin column

```
In [ ]:  df_leads_trans.dropna(subset=['vin'], inplace=True)
         df_leads_trans.shape
```



Create a new dataframe with selected columns

```
In [95]:  df_leads_trans_stream = df_leads_trans[['unfd_ld_all_key','unfd_shpr_all_key','vin','data_src_1','invty_type','ld_src','crnt_ld_sts','crnt_ld_sts_t

In [96]:  df_leads_trans_stream.shape
Out[96]:  (2363768, 49)

In [97]:  df_leads_trans_stream.head(10)
```

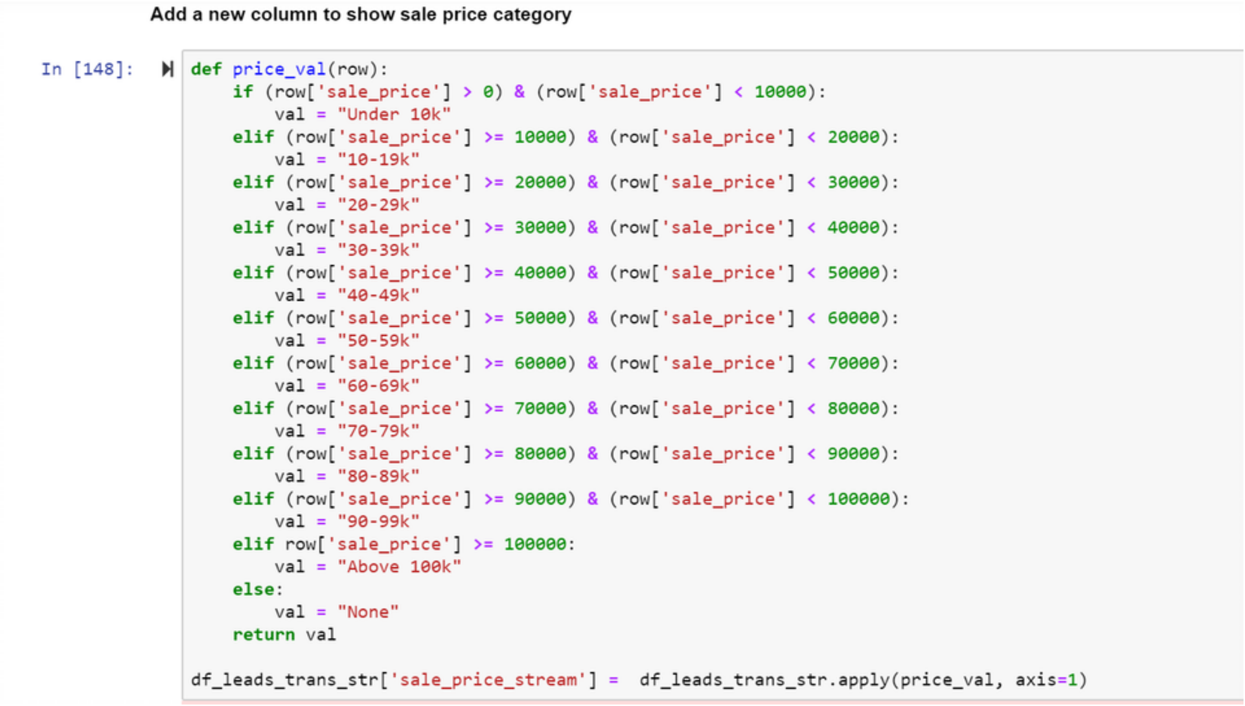
						Hyundai						
2	1657858735191	VIN559809102	3FA6P0G76KR176259	VinSolutions	New	Auto Club Socal	Delivered	Sold	Sales	Walk In	...	REDLANDS FORD
3	1657858735191	VIN559809102	3FA6P0G76KR176259	VinSolutions	New	Auto Club Socal	Delivered	Sold	Sales	Walk In	...	NaN
4	1597729590368	VIN459596524	3FA6P0G76KR176259	VinSolutions	New	AutoTrader	Delivered	Sold	Sales	Phone	...	REDLANDS FORD
5	1597729590368	VIN459596524	3FA6P0G76KR176259	VinSolutions	New	AutoTrader	Delivered	Sold	Sales	Phone	...	NaN
6	1657859080117	VIN502820681	JN8AT2MV7LW103984	VinSolutions	New	Lease Loyalty - 90 Days	Delivered	Sold	Sales	Internet	...	MONTROSE NISSAN
7	1657859191436	VIN645352919	1C4BJWDG4GL220766	VinSolutions	Used	Cargurus	Delivered	Sold	Sales	Phone	...	RAISOR AUTOMART
8	1657859191436	VIN645352919	1C4BJWDG4GL220766	VinSolutions	Used	Cargurus	Delivered	Sold	Sales	Phone	...	RAISOR AUTOMART
9	1657859246094	VIN654119457	1FMCU9HD4KUA37623	VinSolutions	Used	Referral	Delivered	Sold	Sales	Walk In	...	NaN

10 rows × 49 columns

Categorizing Column

Here we are taking a "sale_price" column and creating a new categorized column that will help us when Visualizing the data.

Skip to main content



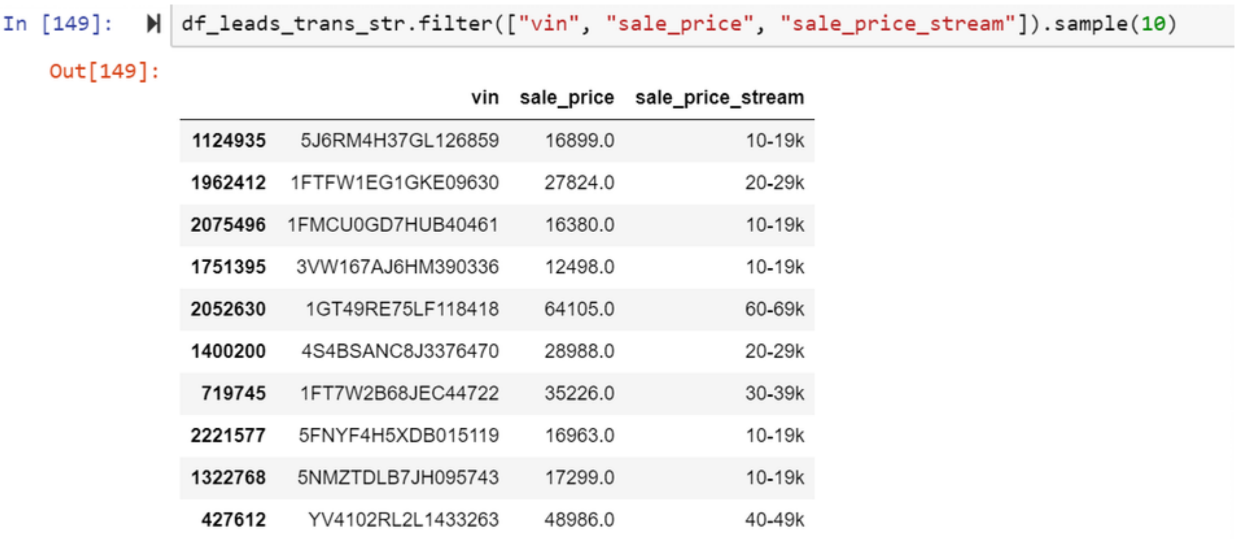
Add a new column to show sale price category

In [148]:

```
def price_val(row):
    if (row['sale_price'] > 0) & (row['sale_price'] < 10000):
        val = "Under 10k"
    elif (row['sale_price'] >= 10000) & (row['sale_price'] < 20000):
        val = "10-19k"
    elif (row['sale_price'] >= 20000) & (row['sale_price'] < 30000):
        val = "20-29k"
    elif (row['sale_price'] >= 30000) & (row['sale_price'] < 40000):
        val = "30-39k"
    elif (row['sale_price'] >= 40000) & (row['sale_price'] < 50000):
        val = "40-49k"
    elif (row['sale_price'] >= 50000) & (row['sale_price'] < 60000):
        val = "50-59k"
    elif (row['sale_price'] >= 60000) & (row['sale_price'] < 70000):
        val = "60-69k"
    elif (row['sale_price'] >= 70000) & (row['sale_price'] < 80000):
        val = "70-79k"
    elif (row['sale_price'] >= 80000) & (row['sale_price'] < 90000):
        val = "80-89k"
    elif (row['sale_price'] >= 90000) & (row['sale_price'] < 100000):
        val = "90-99k"
    elif row['sale_price'] >= 100000:
        val = "Above 100k"
    else:
        val = "None"
    return val

df_leads_trans_str['sale_price_stream'] =  df_leads_trans_str.apply(price_val, axis=1)
```

Here we are inspecting the source and target columns side-by-side. Any columns not needed may be dropped from dataframe, thereby saving on the memory foot-print.



In [149]:

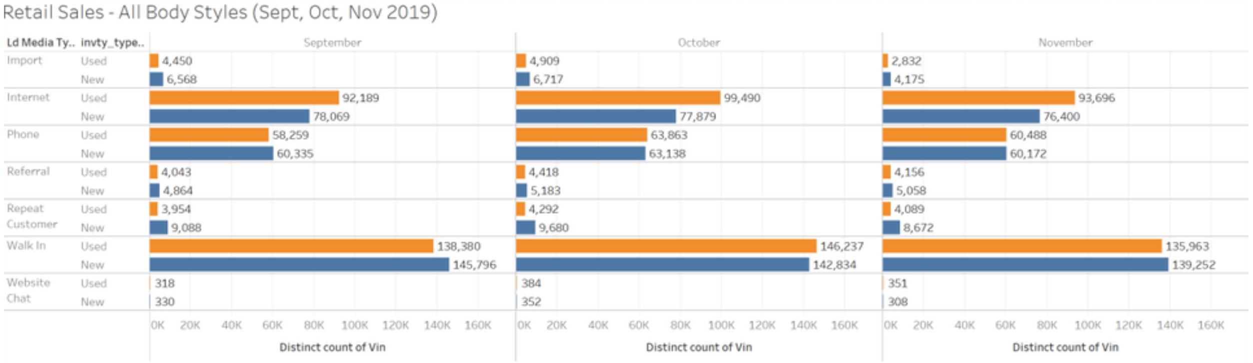
```
df_leads_trans_str.filter(["vin", "sale_price", "sale_price_stream"]).sample(10)
```

Out[149]:

	vin	sale_price	sale_price_stream
1124935	5J6RM4H37GL126859	16899.0	10-19k
1962412	1FTFW1EG1GKE09630	27824.0	20-29k
2075496	1FMCU0GD7HUB40461	16380.0	10-19k
1751395	3VW167AJ6HM390336	12498.0	10-19k
2052630	1GT49RE75LF118418	64105.0	60-69k
1400200	4S4BSANC8J3376470	28988.0	20-29k
719745	1FT7W2B68JEC44722	35226.0	30-39k
2221577	5FNYF4H5XDB015119	16963.0	10-19k
1322768	5NMZTDLB7JH095743	17299.0	10-19k
427612	YV4102RL2L1433263	48986.0	40-49k

Skip to main content

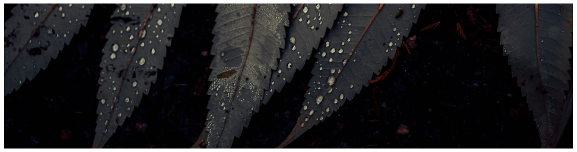
Since all data was local to my computer and I was able to obtain a Tableau Desktop license, I used Tableau to do all visualizations. When working with data, there are always occasions when you may have to return to the source dataset to tweak it and then return to viz tool to observe the changes. There is an alternative to do data transformations in Tableau, but that becomes disjoint from the source dataset (csv file) so I prefer to keep my source accurate. Examples of a charts that was generated using Tableau Desktop 10.5 (https://www.tableau.com/).



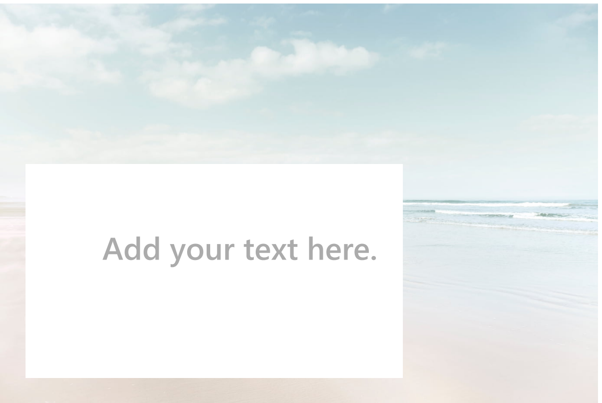
(Optional) Step 7 – Publishing Dataset back to Corporate Data Platform (and Collibra)

I am personally investigating the process required to be able to publish useful data sets back to the Corporate Data Platform. This would involve interfacing with the Data Enablement and Data Platform teams to decide on the value of the data and the best way and format to publish it.

I hope this blog has been helpful to others. Please reach out to me if you have questions, comments or need further information. I will be writing a future blog with more details on usage of some of the tools mentioned here. Thank You for reading and happy data wrangling .....



Skip to main content



Add your text here.

Above is an image with a text overlay. Click inside the image to replace the text with your own.

Add your text here.

Above is an image with a text overlay. Click inside the image to replace the text with your own.

Add your text here.

Above is an image with a text overlay. Click inside the image to replace the text with your own.