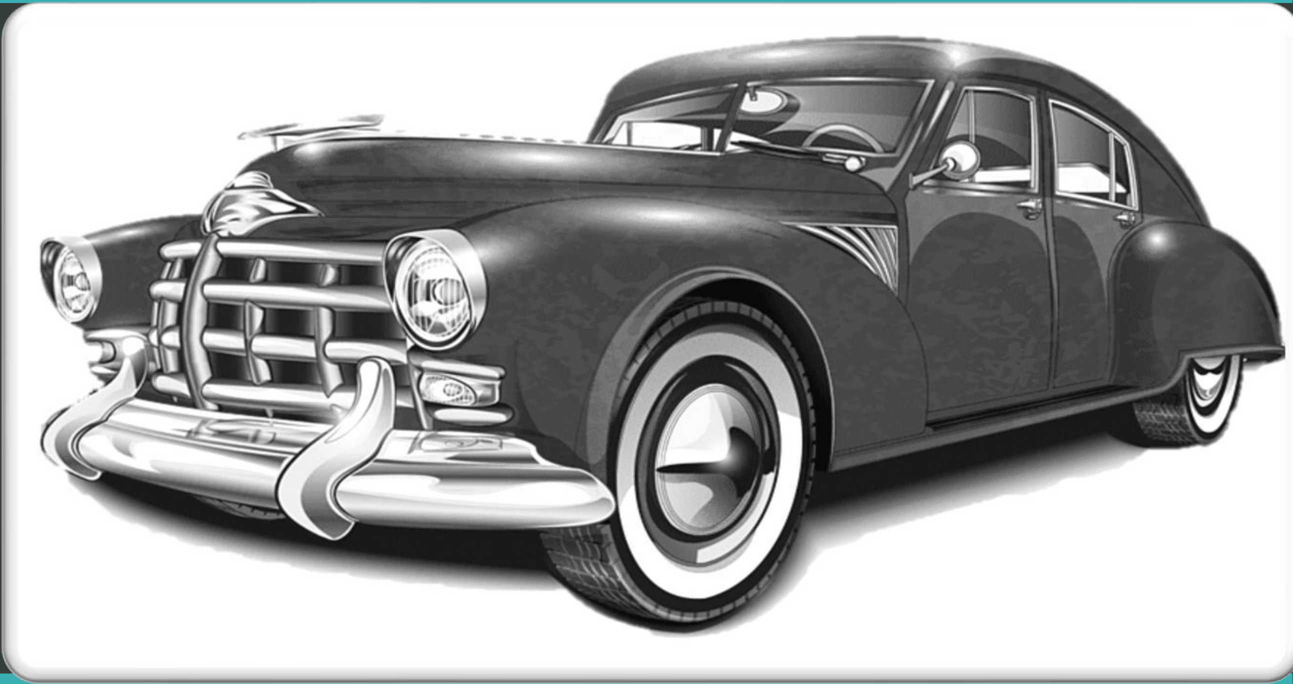
Used Cars: The Data you never knew you needed



BY: REJANE BERINGER, SARA EDGAR AND MICHAEL GOOD

Used Cars market

- Vehicles are the second highest-priced asset consumers purchase, so it’s very important for the economy.

- The used car market in the U.S. is already estimated at 41 million units annually.

Used Cars market

- Consumers can now find greater inventory of used vehicles online.

- More dealers are accepting the move to digital, which is a major behavioral shift for the market.

- A wider net can help you get a better price.

Covid-19 influence

Covid-19 has led to an increase in used car sales

- People avoid mass transportation
- People are more sensitive to auto cost in the recession.

Why used cars?

- Large datasets
    - ❑ There is a lot of available information
- Many different parameters
    - ❑ What is important about the cars that are being sold?
- Location
    - ❑ Where is the best place to find a specific type of car
- Potentially confusing field

Sources:

- Car Sales

https://www.kaggle.com/gagandeep16/car-sales

- US Used Cars Catalog

https://www.kaggle.com/lepchenkov/usedcarscatalog

- Used Cars Dataset

https://www.kaggle.com/doaaalsenani/used-cars-dataets

ETL

Extract

- .csv from Kaggle.

Transform

- We will be using pandas to transform the dataset.
- We will select the important columns from each dataset, remove any irrelevant data or empty rows, and combine the datasets into the relevant tables.

Load

- PostgreSQL Relational database, expected tables to generate:
  - Important subset data from each dataset (x3)
  - Make & model Key table

Extraction

- Kaggle was key
  - List of links
  - Sifting through datasets not as simple as appears
- Data.world was slightly overwhelming
- CSVs reigned supreme

Transformation

- First Step:
  - Downloading the csvs (not as simple as it seems)

- Second Step:
  - Cleaning
    - De duplicating
    - Dropping/adding columns
    - Renaming Columns
    - Eliminating empty or N/A values

- Third Step:
  - Saving as new csv files

```
In [1]: import pandas as pd
        from sqlalchemy import create_engine
        import os
        import csv
```

```
In [2]: #first file
        csv_file = "Car_sales.csv"
        car_sales1_df = pd.read_csv(csv_file, encoding="utf-8")
        car_sales1_df.head()
```

Out[2]:

	Manufacturer	Model	Sales_in_thousands	__year_resale_value	Vehicle_type	Price_in_thousands	Engine_size	Horsepower	Wheelbase	Width	Length
0	Acura	Integra	16.919	16.360	Passenger	21.50	1.8	140.0	101.2	67.3	172.4
1	Acura	TL	39.384	19.875	Passenger	28.40	3.2	225.0	108.1	70.3	192.9
2	Acura	CL	14.114	18.225	Passenger	NaN	3.2	225.0	106.9	70.6	192.0
3	Acura	RL	8.588	29.725	Passenger	42.00	3.5	210.0	114.6	71.4	196.6
4	Audi	A4	20.397	22.255	Passenger	23.99	1.8	150.0	102.6	68.2	178.0

```
In [3]: car_sales1 = car_sales1_df[["Manufacturer", "Model", "Vehicle_type", "Engine_size", "Horsepower","Wheelbase", "Width",
                                    "Length", "Curb_weight", "Fuel_capacity", "Fuel_efficiency", ]]
        car_sales1
```

Out[3]:

	Manufacturer	Model	Vehicle_type	Engine_size	Horsepower	Wheelbase	Width	Length	Curb_weight	Fuel_capacity	Fuel_efficiency
0	Acura	Integra	Passenger	1.8	140.0	101.2	67.3	172.4	2.639	13.2	28.0
1	Acura	TL	Passenger	3.2	225.0	108.1	70.3	192.9	3.517	17.2	25.0
2	Acura	CL	Passenger	3.2	225.0	106.9	70.6	192.0	3.470	17.2	26.0
3	Acura	RL	Passenger	3.5	210.0	114.6	71.4	196.6	3.850	18.0	22.0
4	Audi	A4	Passenger	1.8	150.0	102.6	68.2	178.0	2.998	16.4	27.0
...	...	...	...	...	...	...	...	...	...	...	...
152	Volvo	V40	Passenger	1.9	160.0	100.5	67.6	176.6	3.042	15.8	25.0
153	Volvo	S70	Passenger	2.4	168.0	104.9	69.3	185.9	3.208	17.9	25.0
154	Volvo	V70	Passenger	2.4	168.0	104.9	69.3	186.2	3.259	17.9	25.0
155	Volvo	C70	Passenger	2.3	236.0	104.9	71.5	185.7	3.601	18.5	23.0
156	Volvo	S80	Passenger	2.9	201.0	109.9	72.1	189.8	3.600	21.1	24.0

157 rows × 11 columns

```
In [5]: new_car_sales1_df.insert(0,'id', range(1, 1+ len(new_car_sales1_df)), True)
```

```
In [31]: new_car_sales1_df
```

Out[31]:

	id	Manufacturer	Model	Vehicle_Type	Engine_Size	Horsepower	Wheelbase	Width	Length	Curb_Weight	Fuel_Capacity	Fuel_Efficiency
0	1	Acura	Integra	Passenger	1.8	140.0	101.2	67.3	172.4	2.639	13.2	28.0
1	2	Acura	TL	Passenger	3.2	225.0	108.1	70.3	192.9	3.517	17.2	25.0
2	3	Acura	CL	Passenger	3.2	225.0	106.9	70.6	192.0	3.470	17.2	26.0
3	4	Acura	RL	Passenger	3.5	210.0	114.6	71.4	196.6	3.850	18.0	22.0
4	5	Audi	A4	Passenger	1.8	150.0	102.6	68.2	178.0	2.998	16.4	27.0
...	...	...	...	...	...	...	...	...	...	...	...	...
152	153	Volvo	V40	Passenger	1.9	160.0	100.5	67.6	176.6	3.042	15.8	25.0
153	154	Volvo	S70	Passenger	2.4	168.0	104.9	69.3	185.9	3.208	17.9	25.0
154	155	Volvo	V70	Passenger	2.4	168.0	104.9	69.3	186.2	3.259	17.9	25.0
155	156	Volvo	C70	Passenger	2.3	236.0	104.9	71.5	185.7	3.601	18.5	23.0
156	157	Volvo	S80	Passenger	2.9	201.0	109.9	72.1	189.8	3.600	21.1	24.0

157 rows × 12 columns

```
In [7]: new_car_sales1_df.to_csv("new_car_sales1")
```

In [8]:
```python
#second file
cars_df = pd.read_csv('cars.csv')
```

In [9]:
```python
cars_df.head()
```

Out[9]:

	manufacturer_name	model_name	transmission	color	odometer_value	year_produced	engine_fuel	engine_has_gas	engine_type	engine_capacity	...	f
0	Subaru	Outback	automatic	silver	190000	2010	gasoline	False	gasoline	2.5	...	T
1	Subaru	Outback	automatic	blue	290000	2002	gasoline	False	gasoline	3.0	...	T
2	Subaru	Forester	automatic	red	402000	2001	gasoline	False	gasoline	2.5	...	T
3	Subaru	Impreza	mechanical	blue	10000	1999	gasoline	False	gasoline	3.0	...	F
4	Subaru	Legacy	automatic	black	280000	2001	gasoline	False	gasoline	2.5	...	T

5 rows × 30 columns

In [10]:
```python
count = 0
wanted_columns = []
for val in cars_df.dtypes:
    if val != bool:
        wanted_columns.append(cars_df.columns[count])
    count += 1

wanted_columns
```

Out[10]:
```
['manufacturer_name',
 'model_name',
 'transmission',
 'color',
 'odometer_value',
 'year_produced',
 'engine_fuel',
 'engine_type',
 'engine_capacity',
 'body_type',
 'state',
 'drivetrain',
 'price_usd',
 'location_region',
 'number_of_photos',
 'up_counter',
 'duration_listed']
```

In [11]:
```python
cleaned_cars = cars_df[wanted_columns]
```

In [12]:
```python
cleaned_cars = cleaned_cars.drop('number_of_photos',axis = 1)
cleaned_cars = cleaned_cars.drop('engine_fuel',axis = 1)
cleaned_cars = cleaned_cars.drop('location_region',axis = 1)
```

```
In [13]: cleaned_cars = cleaned_cars.rename( columns = {
             'manufacturer_name': 'Manufacturer',
             'model_name': 'Model',
             'transmission': 'Transmission',
             'color': 'Color',
             'odometer_value': 'Odometer',
             'year_produced': 'Year',
             'engine_type': 'Engine_Type',
             'engine_capacity': 'Engine_Capacity',
             'body_type': 'Body',
             'state': 'State',
             'drivetrain': 'Drivetrain',
             'price_usd': 'Price',
             'up_counter': 'Up_Counter',
             'duration_listed': 'List_Duration'
         }
                                         )
```

```
In [14]: cleaned_cars = cleaned_cars.drop('Up_Counter',axis = 1)
```

```
In [15]: cleaned_cars.insert(0,'id', range(1, 1+ len(cleaned_cars)), True)
```

```
In [16]: cleaned_cars
```

Out[16]:

	id	Manufacturer	Model	Transmission	Color	Odometer	Year	Engine_Type	Engine_Capacity	Body	State	Drivetrain	Price	List_Dura
0	1	Subaru	Outback	automatic	silver	190000	2010	gasoline	2.5	universal	owned	all	10900.00	16
1	2	Subaru	Outback	automatic	blue	290000	2002	gasoline	3.0	universal	owned	all	5000.00	83
2	3	Subaru	Forester	automatic	red	402000	2001	gasoline	2.5	suv	owned	all	2800.00	151
3	4	Subaru	Impreza	mechanical	blue	10000	1999	gasoline	3.0	sedan	owned	all	9999.00	86
4	5	Subaru	Legacy	automatic	black	280000	2001	gasoline	2.5	universal	owned	all	2134.11	7
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
38526	38527	Chrysler	300	automatic	silver	290000	2000	gasoline	3.5	sedan	owned	front	2750.00	301
38527	38528	Chrysler	PT Cruiser	mechanical	blue	321000	2004	diesel	2.2	hatchback	owned	front	4800.00	317
38528	38529	Chrysler	300	automatic	blue	777957	2000	gasoline	3.5	sedan	owned	front	4300.00	369
38529	38530	Chrysler	PT Cruiser	mechanical	black	20000	2001	gasoline	2.0	minivan	owned	front	4000.00	490
38530	38531	Chrysler	Voyager	automatic	silver	297729	2000	gasoline	2.4	minivan	owned	front	3200.00	632

38531 rows × 14 columns

```
In [17]: cleaned_cars.to_csv('cleaned_cars.csv')
```

In [18]:
```
#file 3
csv_file = "Resources/final_cars_datasets.csv"
car_sales2_df = pd.read_csv(csv_file, encoding="utf-8")
car_sales2_df.head()
```

Out[18]:

	Unnamed: 0	price	mark	model	year	mileage	engine_capacity	transmission	drive	hand_drive	fuel
0	0	80	nissan	march	2003	80000	1240	at	2wd	rhd	gasoline
1	1	110	nissan	march	2010	53000	1200	at	2wd	rhd	gasoline
2	2	165	nissan	lafesta	2005	47690	2000	at	2wd	rhd	gasoline
3	3	190	toyota	avensis	2008	130661	1990	at	2wd	rhd	gasoline
4	4	190	daihatsu	mira	2006	66300	660	at	2wd	rhd	gasoline

In [19]:
```
car_sales2_df= car_sales2_df.drop('Unnamed: 0', axis=1)
```

In [20]:
```
car_sales2_df.insert(0,'id', range(1, 1+ len(car_sales2_df)), True)
```

In [21]:
```
car_sales2_df
```

Out[21]:

	id	price	mark	model	year	mileage	engine_capacity	transmission	drive	hand_drive	fuel
0	1	80	nissan	march	2003	80000	1240	at	2wd	rhd	gasoline
1	2	110	nissan	march	2010	53000	1200	at	2wd	rhd	gasoline
2	3	165	nissan	lafesta	2005	47690	2000	at	2wd	rhd	gasoline
3	4	190	toyota	avensis	2008	130661	1990	at	2wd	rhd	gasoline
4	5	190	daihatsu	mira	2006	66300	660	at	2wd	rhd	gasoline
...	...	...	...	...	...	...	...	...	...	...	...
2313	2314	1400	toyota	vitz	2009	121000	996	at	2wd	rhd	gasoline
2314	2315	1400	toyota	estima	2003	101000	3000	at	2wd	rhd	gasoline
2315	2316	1400	subaru	r2	2005	101000	660	cvt	2wd	rhd	gasoline
2316	2317	1400	honda	z	2000	170000	660	at	4wd	rhd	gasoline
2317	2318	1400	toyota	estima t	2005	72320	3000	at	2wd	rhd	gasoline

2318 rows × 11 columns

In [22]:
```
car_sales2_df.to_csv('car_sales2_df.csv')
```

In [23]:
```
#connect to database
connection= "postgres:USrec986!@localhost:5432/cars_db"
engine = create_engine(f'postgresql://{connection}')
```

In [24]:
```
engine.table_names()
```

Out[24]:
```
['cars.cars', 'cars', 'vehicles', 'sara_cars']
```

In [25]:
```
cleaned_cars.to_sql(name='cars', con=engine, if_exists='append', index=False)
```

In [26]:
```
pd.read_sql_query('select * from cars', con=engine).head()
```

Out[26]:

	id	Manufacturer	Model	Transmission	Color	Odometer	Year	Engine_Type	Engine_Capacity	Body	State	Drivetrain	Price	List_Duration
0	1	Subaru	Outback	automatic	silver	190000	2010	gasoline	2.5	universal	owned	all	10900.00	16
1	2	Subaru	Outback	automatic	blue	290000	2002	gasoline	3.0	universal	owned	all	5000.00	83
2	3	Subaru	Forester	automatic	red	402000	2001	gasoline	2.5	suv	owned	all	2800.00	151
3	4	Subaru	Impreza	mechanical	blue	10000	1999	gasoline	3.0	sedan	owned	all	9999.00	86
4	5	Subaru	Legacy	automatic	black	280000	2001	gasoline	2.5	universal	owned	all	2134.11	7

In [27]:
```
new_car_sales1_df.to_sql(name='vehicles', con=engine, if_exists='append', index=False)
```

In [35]:
```
pd.read_sql_query('select * from vehicles', con=engine).head()
```

Out[35]:

	id	Manufacturer	Model	Vehicle_Type	Engine_Size	Horsepower	Wheelbase	Width	Length	Curb_Weight	Fuel_Capacity	Fuel_Efficiency
0	1	Acura	Integra	Passenger	1.8	140.0	101.2	67.3	172.4	2.639	13.2	28.0
1	2	Acura	TL	Passenger	3.2	225.0	108.1	70.3	192.9	3.517	17.2	25.0
2	3	Acura	CL	Passenger	3.2	225.0	106.9	70.6	192.0	3.470	17.2	26.0
3	4	Acura	RL	Passenger	3.5	210.0	114.6	71.4	196.6	3.850	18.0	22.0
4	5	Audi	A4	Passenger	1.8	150.0	102.6	68.2	178.0	2.998	16.4	27.0

```
In [29]: car_sales2_df.to_sql(name='sara_cars', con=engine, if_exists='append', index=False)

In [30]: pd.read_sql_query('select * from sara_cars', con=engine).head()
```

Out[30]:

	id	price	mark	model	year	mileage	engine_capacity	transmission	drive	hand_drive	fuel
0	1	80	nissan	march	2003	80000	1240	at	2wd	rhd	gasoline
1	2	110	nissan	march	2010	53000	1200	at	2wd	rhd	gasoline
2	3	165	nissan	lafesta	2005	47690	2000	at	2wd	rhd	gasoline
3	4	190	toyota	avensis	2008	130661	1990	at	2wd	rhd	gasoline
4	5	190	daihatsu	mira	2006	66300	660	at	2wd	rhd	gasoline

Load

- Creating cars database
- Writing schema for the tables
- Connecting pandas dataframes to SQL
- Creating queries for new tables
  - Using inner join to create new tables

SQL

Query Editor | Query History

```sql
SELECT cars."Manufacturer", cars."Model", vehicles."Manufacturer", vehicles."Horsepower"
FROM cars
INNER JOIN vehicles
ON vehicles."Manufacturer" = cars."Manufacturer"
AND vehicles."Model" = cars."Model";
```

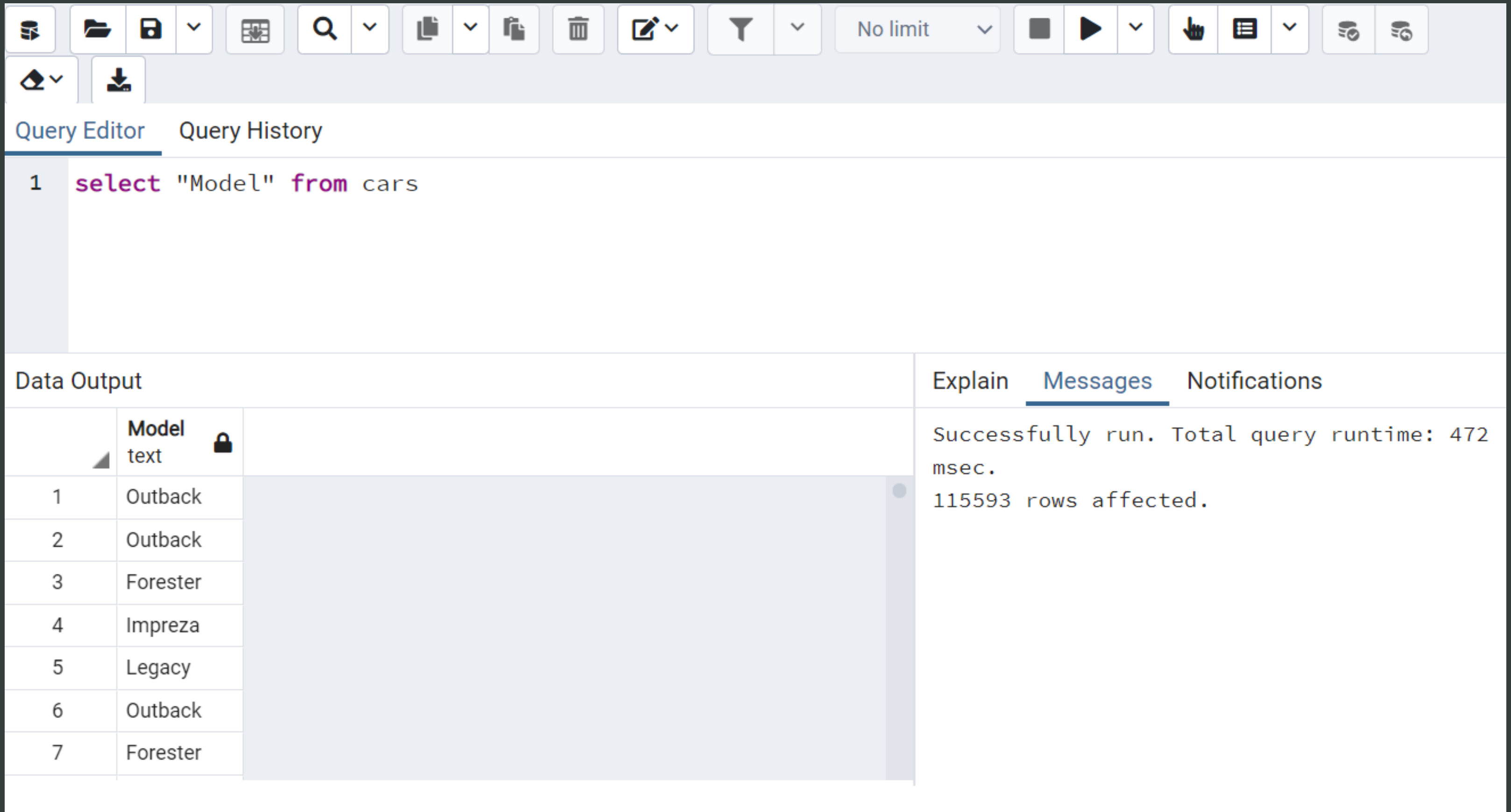
Data Output

	Manufacturer text	Model text	Manufacturer text	Horsepower double precision
1	Subaru	Outback	Subaru	165
2	Subaru	Outback	Subaru	165
3	Subaru	Outback	Subaru	165
4	Subaru	Outback	Subaru	165
5	Subaru	Outback	Subaru	165
6	Subaru	Outback	Subaru	165
7	Subaru	Outback	Subaru	165
8	Subaru	Outback	Subaru	165

Explain | Messages | Notifications

```
Successfully run. Total query runtime: 548 msec.
83580 rows affected.
```

SQL


No limit
Query Editor
Query History
1 select "Model" from cars
Data Output
Model
text
1 Outback
2 Outback
3 Forester
4 Impreza
5 Legacy
6 Outback
7 Forester
Explain
Messages
Notifications
Successfully run. Total query runtime: 472 msec.
115593 rows affected.

SQL

No limit

Query Editor | Query History

```sql
select * from vehicles
where "Horsepower" > 150
```

Data Output

	id integer	Manufacturer text	Model text	Vehicle_Type text	Engine_Size double precision
1	2	Acura	TL	Passenger	
2	3	Acura	CL	Passenger	
3	4	Acura	RL	Passenger	
4	6	Audi	A6	Passenger	
5	7	Audi	A8	Passenger	
6	8	BMW	323i	Passenger	
7	9	BMW	328i	Passenger	

Explain | Messages | Notifications

```
Successfully run. Total query runtime: 359 msec.
432 rows affected.
```

Issues Encountered

- Connecting Data Frames to SQL
- Discovering that inserting the “tables” through pandas and creating the schema from scratch in SQL would cause MAJOR issues
- Combining Tables
- Needing to use “” to emphasize uppercase letters when identifying the columns in the tables within SQL

Questions