**SQL Code |Profile Formulation**



**Source**: Combined Car Data *(double click)* 🡪 

For SQL utilization, we used Oracle APEX SQL. Cetain syntax for code may vary across different software, but the code provided in the document runs on Oracle. The steps of how we achieved our vehicle selections and how to *achieve future results* are shown below. You can copy and paste every step with the code provided.

Every profile has more than 3 cars that fit the parameters and this code allows us to see all our options for each vehicle profile.

SQL Steps:

1. **Cleaning MSRP Column**

Our first step is to generate a code that deletes all the commas and money signs within our MSRP column. When searching and sorting, SQL does not work well with other characters when using numerical values.

We inserted our table as “**CARS**” into Oracle and used the code down below to remove these extra characters from the MSRP column:

UPDATE CARS

SET MSRP\_PRICE = REGEXP\_REPLACE(MSRP\_PRICE, '[$,]', '');

1. **Luxury Enthusiast**

With the code below, we can look at cars within the selected price range, while the code also displays the varying MPG ranges. With these parameters in place, we selected three models that fit the features of the Luxury Enthusiast profile well, but there are other vehicle options that fit the criteria as well from our data.

SELECT MAKE\_MODEL, MSRP\_PRICE, CITY\_MPG, HIGHWAY\_MPG FROM CARS

WHERE MSRP\_PRICE > 70000 AND MSRP\_PRICE <80000 AND HIGHWAY\_MPG > 15

ORDER BY HIGHWAY\_MPG ASC;

1. **Economical Driver**

We added the “Engine\_Type” to our code to look at the cars that fit our profile parameters while also having a smaller engine. These smaller engines, in most cases, offer a better fuel-efficient vehicle, which is a big differentiator for this vehicle profile.

SELECT MAKE\_MODEL, MSRP\_PRICE, CITY\_MPG, HIGHWAY\_MPG, ENGINE\_TYPE FROM CARS

WHERE MSRP\_PRICE > 25000 AND MSRP\_PRICE <35000 AND HIGHWAY\_MPG > 25 AND HIGHWAY\_MPG < 43

ORDER BY HIGHWAY\_MPG DESC;

1. **Dynamic Driver**

Following the same process, we limited our selection using a different MSRP price range and looking at the vehicles that have similar MPG performance as our profile.

SELECT MAKE\_MODEL, MSRP\_PRICE, CITY\_MPG, HIGHWAY\_MPG, ENGINE\_TYPE FROM CARS

WHERE MSRP\_PRICE > 40000 AND MSRP\_PRICE <50000 AND HIGHWAY\_MPG > 12 AND HIGHWAY\_MPG < 30

ORDER BY HIGHWAY\_MPG ASC;

1. **High-End Motorists**

Lastly, we used very similar code as previous steps. We added another condition where we only look at vehicles above 30 MPG on the highway, making it easier to see which cars are going to match our profile’s performance.

SELECT MAKE\_MODEL, MSRP\_PRICE, CITY\_MPG, HIGHWAY\_MPG, ENGINE\_TYPE FROM CARS

WHERE MSRP\_PRICE > 40000 AND MSRP\_PRICE <50000 AND HIGHWAY\_MPG >= 30

ORDER BY HIGHWAY\_MPG ASC;