This is an sql file i made holding data on price model and make of 1000 vehicles. Looking at the data the prices are clearly wrong or people were greatly overcharged. The average price of Aston martins sold wouldn't be less than that of volkswagens.

In [1]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import mysql.connector
from mysql.connector import Error

try:
    connection = mysql.connector.connect(host='localhost',
                                         database='cars_owned',
                                         user='root',
                                         password='corvette')
    if connection.is_connected():
        db_Info = connection.get_server_info()
        print("Connected to MySQL Server version", db_Info)
        cursor = connection.cursor()
        cursor.execute("select database();")
        record = cursor.fetchone()
        print("You're connected to database: ", record)
#exception error if problem connecting to the database
except Error as e:
    print("Error while connecting to MySQL", e)
#if connection is successful then it will close connection after
finally:
    if (connection.is_connected()):
        cursor.close()
        connection.close()
        print("MySQL connection is closed")
#tested if it was able to connect to database
```

```
Connected to MySQL Server version 8.0.21
You're connected to database:  ('cars_owned',)
MySQL connection is closed
```

In [2]:

```
try:
    connection = mysql.connector.connect(host='localhost',
                                         database='cars_owned',
                                         user='root',
                                         password='corvette')
    if connection.is_connected():
        db_Info = connection.get_server_info()
        print("Connected to MySQL Server version", db_Info)
        #returns a cursor object. Using a cursor object, we can execute SQL quer
        cursor = connection.cursor()
        #table and columns being worked on here sql query input
        cursor.execute("select database();")
        record = cursor.fetchone()
        print("You're connected to database: ", record)

except Error as e:
    print("Error while connecting to MySQL", e)
#connect to database without closing so i can work with it
```

```
Connected to MySQL Server version 8.0.21
You're connected to database:  ('cars_owned',)
```

In [3]:

```
cursor = connection.cursor(buffered=True)
query="select * from car"
cursor.execute(query)
fetching_size = 2
#numbers of rows pulled will be equal to fetching_size
rows_to_get= cursor.fetchmany(fetching_size)

print("Total number of rows is: ", cursor.rowcount)
print("Printing ", fetching_size, " car record using cursor.fetchmany")
for row in rows_to_get:
    print(row)
#working with 1000 rows of car data
```

```
Total number of rows is:  1000
Printing  2  car record using cursor.fetchmany
(1, 'Jaguar', 'S-Type', Decimal('74064.94'))
(2, 'Oldsmobile', 'Bravada', Decimal('15957.92'))
```

In [4]:

```
cars= pd.read_sql(query, connection)
#set index = to id instead of standard panda index starting at 0
cars=cars.set_index('id')
```

In [5]:

```
cars
```

Out[5]:

```
            make        model     price
  id
   1      Jaguar       S-Type  74064.94
   2  Oldsmobile      Bravada  15957.92
   3       Lexus           RX  91013.67
   4       Dodge  Dakota Club  59080.52
   5       Honda        Civic  59582.79
 ...         ...          ...       ...
 996       Dodge     Sprinter  81171.40
 997     Rambler      Classic  39244.33
 998      Jaguar           XF  44212.85
 999      Nissan       Altima  79086.75
1000   Chevrolet          HHR  50829.11
```

1000 rows × 3 columns

In [6]:

```
#cars with price over $80,000. prices are inaccurate civic overpriced
cars.loc[cars['price']> 80000]
```

Out[6]:

```
           make                model     price
 id
  3       Lexus                   RX  91013.67
  6        Ford              Mustang  82861.26
 12       Honda                Civic  97075.02
 15      Nissan                Rogue  93827.30
 20       Dodge             Ram 2500  94610.72
...         ...                  ...       ...
981       Acura                   RL  94353.21
982  Land Rover  Range Rover Classic  99340.81
985   Chevrolet               Impala  93138.12
994     Mercury             Monterey  84291.59
996       Dodge             Sprinter  81171.40
```

210 rows × 3 columns

Range Rover could be correct price

In [7]:

```
#Group all cars of the same brand together and find average price
cars2=cars.groupby(['make']).mean().round(2)
cars2
```

Out[7]:

	price
make	
Acura	57781.12
Aptera	72048.15
Aston Martin	57779.05
Audi	52213.16
Austin	58452.51
...	...
Suzuki	63587.32
Tesla	30800.84
Toyota	50598.33
Volkswagen	58441.04
Volvo	60620.75

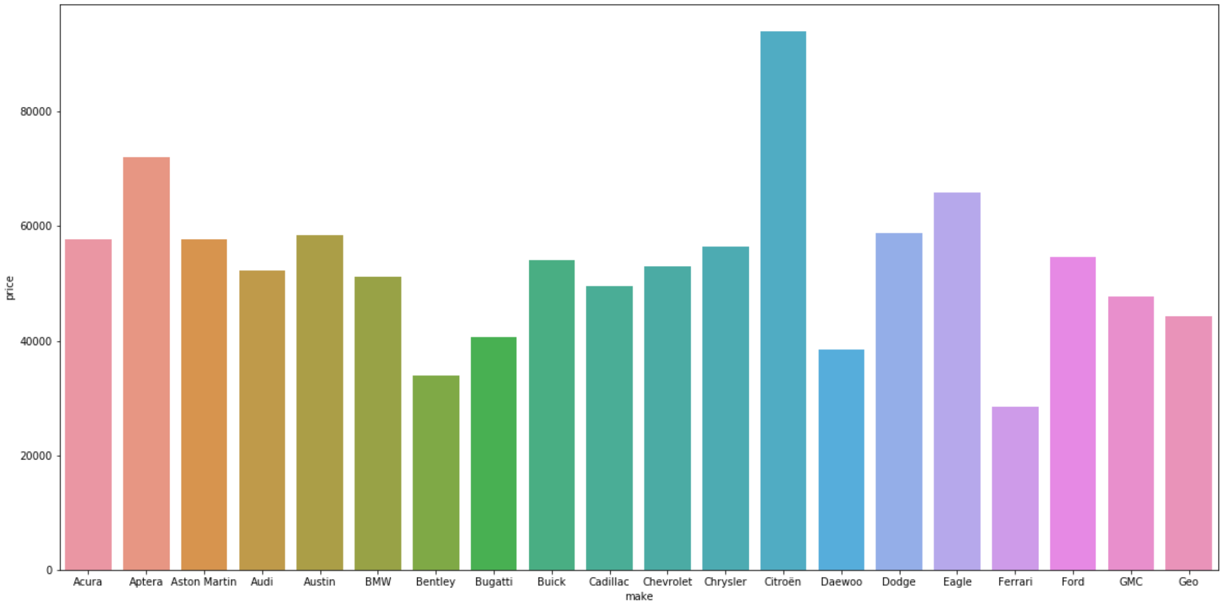
62 rows × 1 columns

In [8]:

```
cars3= cars2.head(20) # just first 20 results from dataframe
plt.figure(figsize=(20,10))
sns.barplot(x=cars3.index, y=cars3['price'])
```

Out[8]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2c5d67e57c8>
```



Ferrari, Bentley, and Bugatti brands all under 40k