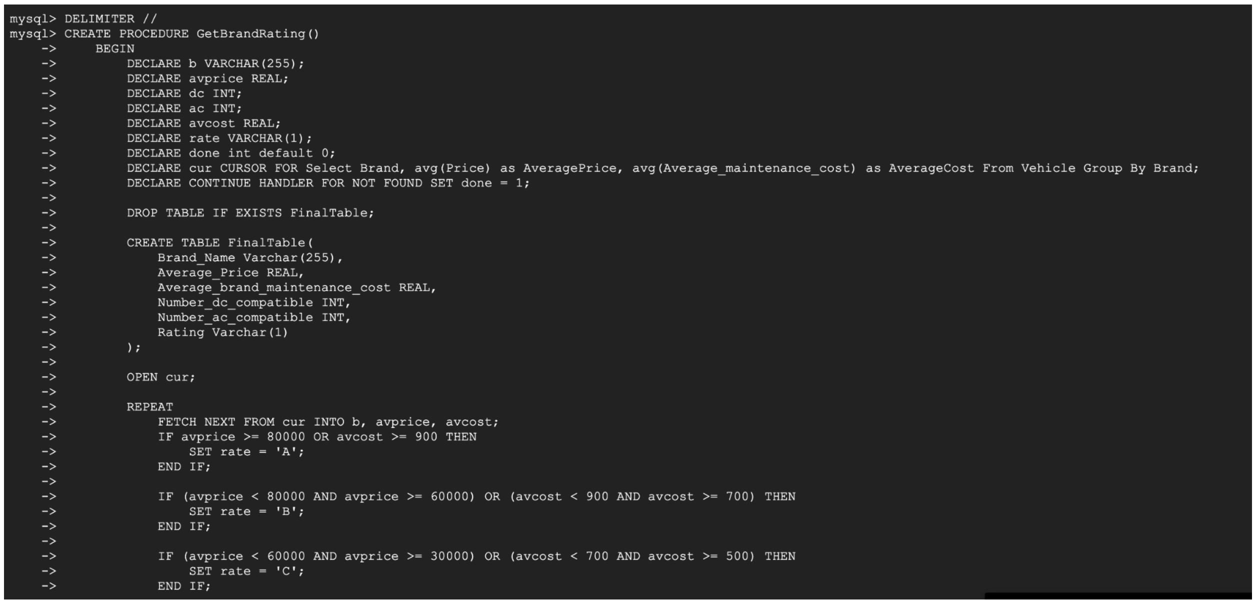
STAGE 5 FINAL DEMO DOCUMENTATION

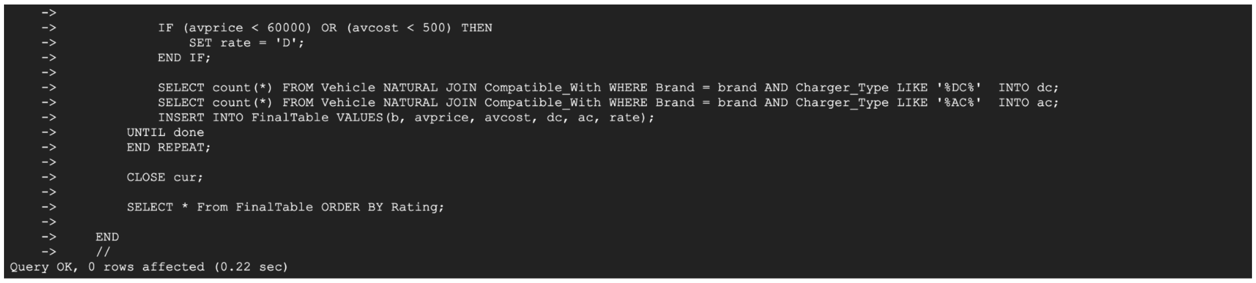
1. **BASIC CRUD INTERFACE**

   Presented during the demo

2. **STORED PROCEDURE**
   We chose the stored procedure + trigger option for this stage.
   We created a stored procedure that calculates the Average Price, Average Maintenance Cost, Number of DC compatible vehicles, Number of AC compatible vehicles, and Rating according to a specified rubric for each brand.
   We believe this stored procedure is useful for rating each brand by average price and average maintenance cost, and for analyzing the number of dc and ac compatible vehicles with each brand.
   The stored procedure was defined as follows:

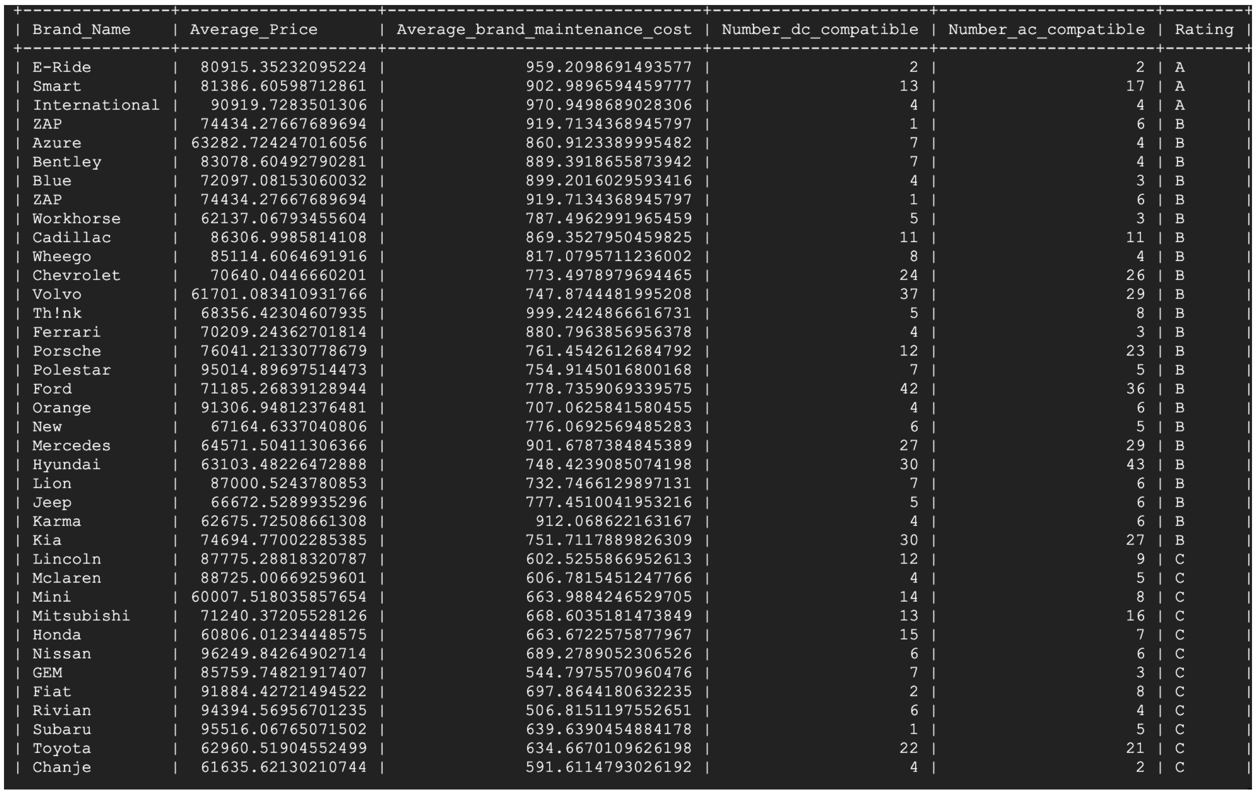


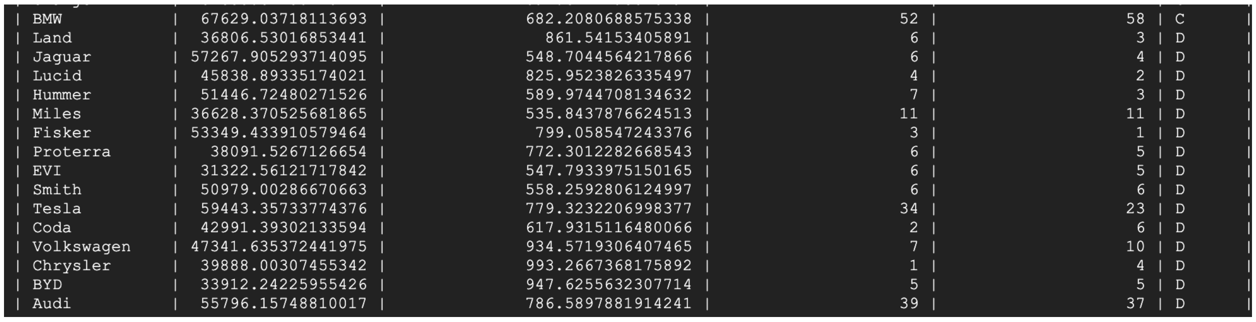
```
mysql> DELIMITER //
mysql> CREATE PROCEDURE GetBrandRating()
    ->     BEGIN
    ->         DECLARE b VARCHAR(255);
    ->         DECLARE avprice REAL;
    ->         DECLARE dc INT;
    ->         DECLARE ac INT;
    ->         DECLARE avcost REAL;
    ->         DECLARE rate VARCHAR(1);
    ->         DECLARE done int default 0;
    ->         DECLARE cur CURSOR FOR Select Brand, avg(Price) as AveragePrice, avg(Average_maintenance_cost) as AverageCost From Vehicle Group By Brand;
    ->         DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;
    ->
    ->         DROP TABLE IF EXISTS FinalTable;
    ->
    ->         CREATE TABLE FinalTable(
    ->             Brand_Name Varchar(255),
    ->             Average_Price REAL,
    ->             Average_brand_maintenance_cost REAL,
    ->             Number_dc_compatible INT,
    ->             Number_ac_compatible INT,
    ->             Rating Varchar(1)
    ->         );
    ->
    ->         OPEN cur;
    ->
    ->         REPEAT
    ->             FETCH NEXT FROM cur INTO b, avprice, avcost;
    ->             IF avprice >= 80000 OR avcost >= 900 THEN
    ->                 SET rate = 'A';
    ->             END IF;
    ->
    ->             IF (avprice < 80000 AND avprice >= 60000) OR (avcost < 900 AND avcost >= 700) THEN
    ->                 SET rate = 'B';
    ->             END IF;
    ->
    ->             IF (avprice < 60000 AND avprice >= 30000) OR (avcost < 700 AND avcost >= 500) THEN
    ->                 SET rate = 'C';
    ->             END IF;
```



```
    ->
    ->              IF (avprice < 60000) OR (avcost < 500) THEN
    ->                  SET rate = 'D';
    ->              END IF;
    ->
    ->              SELECT count(*) FROM Vehicle NATURAL JOIN Compatible_With WHERE Brand = brand AND Charger_Type LIKE '%DC%'  INTO dc;
    ->              SELECT count(*) FROM Vehicle NATURAL JOIN Compatible_With WHERE Brand = brand AND Charger_Type LIKE '%AC%'  INTO ac;
    ->              INSERT INTO FinalTable VALUES(b, avprice, avcost, dc, ac, rate);
    ->          UNTIL done
    ->          END REPEAT;
    ->
    ->          CLOSE cur;
    ->
    ->          SELECT * From FinalTable ORDER BY Rating;
    ->
    ->      END
    ->      //
Query OK, 0 rows affected (0.22 sec)
```

Calling the stored procedure displays the following results:





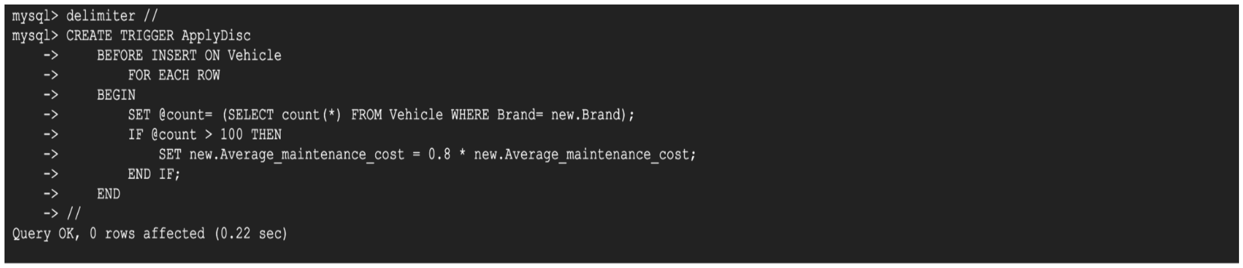
```
+---------------+--------------------+--------------------------------+----------------------+----------------------+--------+
| Brand_Name    | Average_Price      | Average_brand_maintenance_cost | Number_dc_compatible | Number_ac_compatible | Rating |
+---------------+--------------------+--------------------------------+----------------------+----------------------+--------+
| E-Ride        |  80915.35232095224 |              959.2098691493577 |                    2 |                    2 | A      |
| Smart         |  81386.60598712861 |              902.9896594459777 |                   13 |                   17 | A      |
| International |   90919.7283501306 |              970.9498689028306 |                    4 |                    4 | A      |
| ZAP           |  74434.27667689694 |              919.7134368945797 |                    1 |                    6 | B      |
| Azure         | 63282.724247016056 |              860.9123389995482 |                    7 |                    4 | B      |
| Bentley       |  83078.60492790281 |              889.3918655873942 |                    7 |                    4 | B      |
| Blue          |  72097.08153060032 |              899.2016029593416 |                    4 |                    3 | B      |
| ZAP           |  74434.27667689694 |              919.7134368945797 |                    1 |                    6 | B      |
| Workhorse     |  62137.06793455604 |              787.4962991965459 |                    5 |                    3 | B      |
| Cadillac      |   86306.9985814108 |              869.3527950459825 |                   11 |                   11 | B      |
| Wheego        |   85114.6064691916 |              817.0795711236002 |                    8 |                    4 | B      |
| Chevrolet     |   70640.0446660201 |              773.4978979694465 |                   24 |                   26 | B      |
| Volvo         | 61701.083410931766 |              747.8744481995208 |                   37 |                   29 | B      |
| Th!nk         |  68356.42304607935 |              999.2424866616731 |                    5 |                    8 | B      |
| Ferrari       |  70209.24362701814 |              880.7963856956378 |                    4 |                    3 | B      |
| Porsche       |  76041.21330778679 |              761.4542612684792 |                   12 |                   23 | B      |
| Polestar      |  95014.89697514473 |              754.9145016800168 |                    7 |                    5 | B      |
| Ford          |  71185.26839128944 |              778.7359069339575 |                   42 |                   36 | B      |
| Orange        |  91306.94812376481 |              707.0625841580455 |                    4 |                    6 | B      |
| New           |   67164.6337040806 |              776.0692569485283 |                    6 |                    5 | B      |
| Mercedes      |  64571.50411306366 |              901.6787384845389 |                   27 |                   29 | B      |
| Hyundai       |  63103.48226472888 |              748.4239085074198 |                   30 |                   43 | B      |
| Lion          |   87000.5243780853 |              732.7466129897131 |                    7 |                    6 | B      |
| Jeep          |   66672.5289935296 |              777.4510041953216 |                    5 |                    6 | B      |
| Karma         |  62675.72508661308 |               912.068622163167 |                    4 |                    6 | B      |
| Kia           |  74694.77002285385 |              751.7117889826309 |                   30 |                   27 | B      |
| Lincoln       |  87775.28818320787 |              602.5255866952613 |                   12 |                    9 | C      |
| Mclaren       |  88725.00669259601 |              606.7815451247766 |                    4 |                    5 | C      |
| Mini          | 60007.518035857654 |              663.9884246529705 |                   14 |                    8 | C      |
| Mitsubishi    |  71240.37205528126 |              668.6035181473849 |                   13 |                   16 | C      |
| Honda         |  60806.01234448575 |              663.6722575877967 |                   15 |                    7 | C      |
| Nissan        |  96249.84264902714 |              689.2789052306526 |                    6 |                    6 | C      |
| GEM           |  85759.74821917407 |              544.7975570960476 |                    7 |                    3 | C      |
| Fiat          |  91884.42721494522 |              697.8644180632235 |                    2 |                    8 | C      |
| Rivian        |  94394.56956701235 |              506.8151197552651 |                    6 |                    4 | C      |
| Subaru        |  95516.06765071502 |              639.6390454884178 |                    1 |                    5 | C      |
| Toyota        |  62960.51904552499 |              634.6670109626198 |                   22 |                   21 | C      |
| Chanje        |  61635.62130210744 |              591.6114793026192 |                    4 |                    2 | C      |

| BMW           |  67629.03718113693 |              682.2080688575338 |                   52 |                   58 | C      |
| Land          |  36806.53016853441 |                861.54153405891 |                    6 |                    3 | D      |
| Jaguar        | 57267.905293714095 |              548.7044564217866 |                    6 |                    4 | D      |
| Lucid         |  45838.89335174021 |              825.9523826335497 |                    4 |                    2 | D      |
| Hummer        |  51446.72480271526 |              589.9744708134632 |                    7 |                    3 | D      |
| Miles         | 36628.370525681865 |              535.8437876624513 |                   11 |                   11 | D      |
| Fisker        | 53349.433910579464 |               799.058547243376 |                    3 |                    1 | D      |
| Proterra      |   38091.5267126654 |              772.3012282668543 |                    6 |                    5 | D      |
| EVI           |  31322.56121717842 |              547.7933975150165 |                    6 |                    5 | D      |
| Smith         |  50979.00286670663 |              558.2592806124997 |                    6 |                    6 | D      |
| Tesla         |  59443.35733774376 |              779.3232206998377 |                   34 |                   23 | D      |
| Coda          |  42991.39302133594 |              617.9315116480066 |                    2 |                    6 | D      |
| Volkswagen    | 47341.635372441975 |              934.5719306407465 |                    7 |                   10 | D      |
| Chrysler      |  39888.00307455342 |              993.2667368175892 |                    1 |                    4 | D      |
| BYD           |  33912.24225955426 |              947.6255632307714 |                    5 |                    5 | D      |
| Audi          |  55796.15748810017 |              786.5897881914241 |                   39 |                   37 | D      |
```

The results will be corroborated during the demo. A .SQL file with the query has been submitted to the project repo. The stored procedure can be called from the front-end using a button to get Brand ratings.

3. <u>TRIGGER</u>

Our trigger for this change represents a potential scheme that could be launched. We assume that hypothetically the government has created subsidies for companies with more than 100 cars, and hence the average maintenance cost has become 80% of what it actually is. Hence we created the following trigger to reduce the average maintenance cost by 20% for all cars belonging to a brand with more than 100 vehicles.

The code of the trigger is as follows:



```
mysql> delimiter //
mysql> CREATE TRIGGER ApplyDisc
    ->     BEFORE INSERT ON Vehicle
    ->         FOR EACH ROW
    ->     BEGIN
    ->         SET @count= (SELECT count(*) FROM Vehicle WHERE Brand= new.Brand);
    ->         IF @count > 100 THEN
    ->             SET new.Average_maintenance_cost = 0.8 * new.Average_maintenance_cost;
    ->         END IF;
    ->     END
    -> //
Query OK, 0 rows affected (0.22 sec)
```

This trigger interacts with the front-end using the insert function available to the employees.

The results will be corroborated during the demo.A .SQL file with the query has been submitted to the project repo.