Query 1:

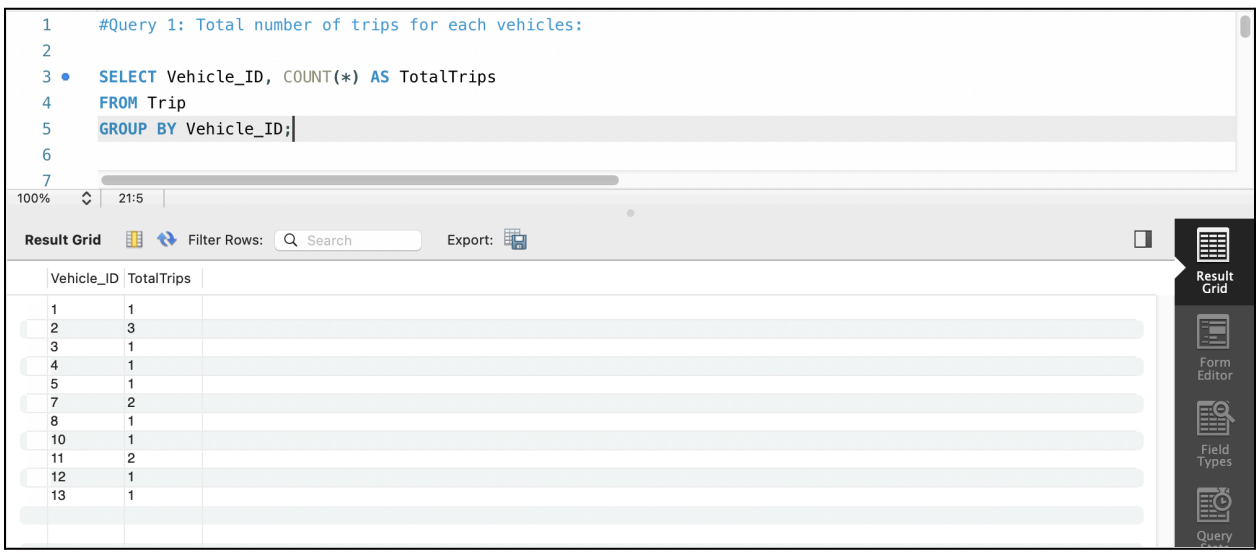
Query statement:

Total number of trips for each vehicle:

SQL command:

SELECT Vehicle_ID, COUNT(*) AS TotalTrips
FROM Trip
GROUP BY Vehicle_ID;

Output:



Query 2:

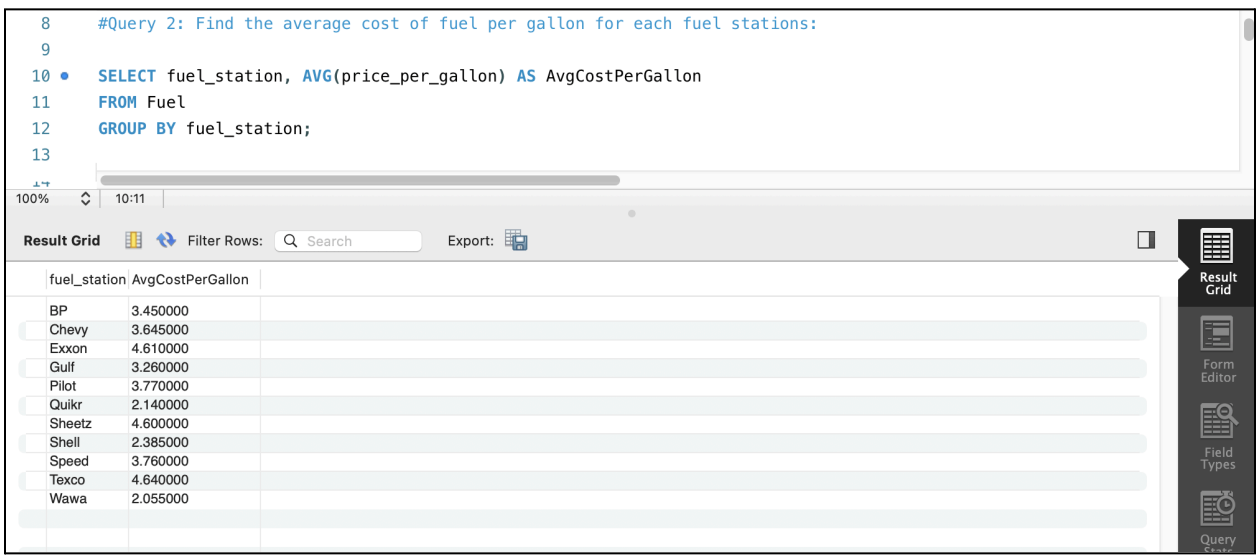
Query statement:

Find the average cost of fuel per gallon for each fuel station:

SQL command:

```
SELECT fuel_station, AVG(price_per_gallon) AS AvgCostPerGallon
FROM Fuel
GROUP BY fuel_station;
```

Output:



fuel_station	AvgCostPerGallon
BP	3.450000
Chevy	3.645000
Exxon	4.610000
Gulf	3.260000
Pilot	3.770000
Quikr	2.140000
Sheetz	4.600000
Shell	2.385000
Speed	3.760000
Texco	4.640000
Wawa	2.055000

Query 3:

Query statement:

Find the total fuel cost for each vehicle:

SQL command:

```
SELECT vehicle_ID, SUM(total_cost) AS TotalFuelCost
FROM Fuel
GROUP BY vehicle_ID;
```

Output:

```
#Query 3: Find the total fuel cost for each vehicle:

SELECT vehicle_ID, SUM(total_cost) AS TotalFuelCost
FROM Fuel
GROUP BY vehicle_ID;
```

Result Grid

vehicle_ID	TotalFuelCost
1	59.823
2	114.673
4	17.815
5	55.481
6	116.150
7	11.799
8	26.445
9	115.062
12	94.573
13	91.105
14	51.411

Query 4:

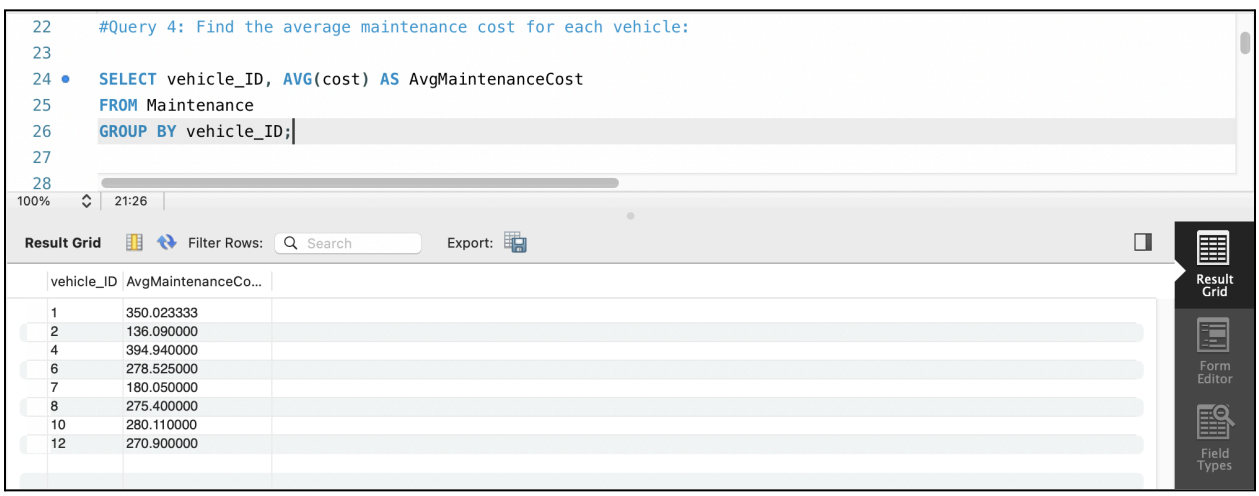
Query statement:

Find the average maintenance cost for each vehicle:

SQL command:

```
SELECT vehicle_ID, AVG(cost) AS AvgMaintenanceCost
FROM Maintenance
GROUP BY vehicle_ID;
```

Output:



Query 5:

Query statement:

Find the total inventory cost per item:

SQL command:

SELECT ID, name, stock_level * unit_price AS TotalInventoryCost
FROM Inventory
group by ID;

Output:

```
#Query 5: Find the total inventory cost per item:

SELECT ID,name, stock_level * unit_price AS TotalInventoryCost
FROM Inventory
group by ID;
```

100% 13:33

Result Grid Filter Rows: Search Export:

ID	name	TotalInventoryC...
1	engine	22505.60
2	tire	5811.30
3	brake pad	48277.90
4	headlight	22584.24
5	bumper	5664.96
6	radiator	18003.18
7	spark plug	34960.80
8	fuel pump	37759.63
9	air filter	47094.93
10	alternator	18671.38
11	steering...	44451.80
12	exhaust...	32252.40
13	shock ab...	6764.76
14	windshie...	8697.24
15	catalytic...	9132.55

Result Grid
Form Editor
Field Types
Query Stats

Query 6:

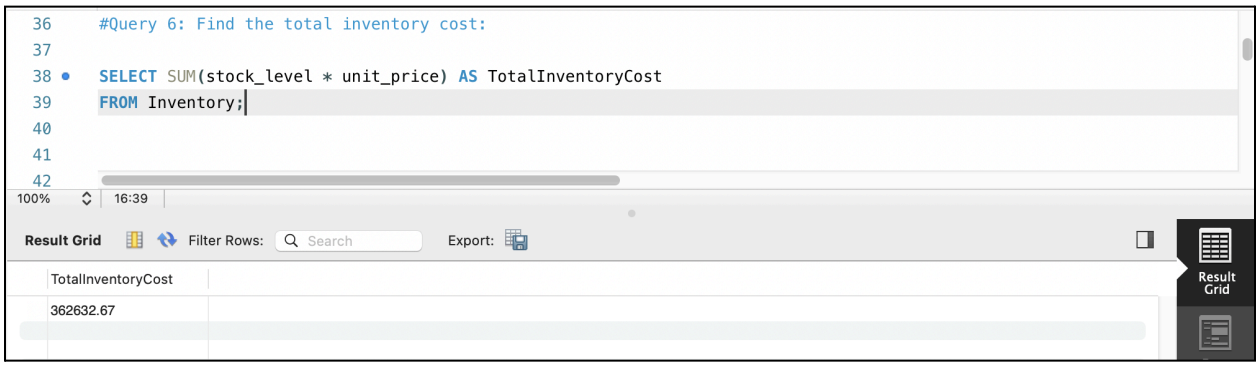
Query statement:

Find the total inventory cost:

SQL command:

SELECT SUM(stock_level * unit_price) AS TotalInventoryCost
FROM Inventory;

Output:



Query 7:

Query statement:

Find the inventory that needs to be ordered. (i.e. the stock level is below the reorder point):

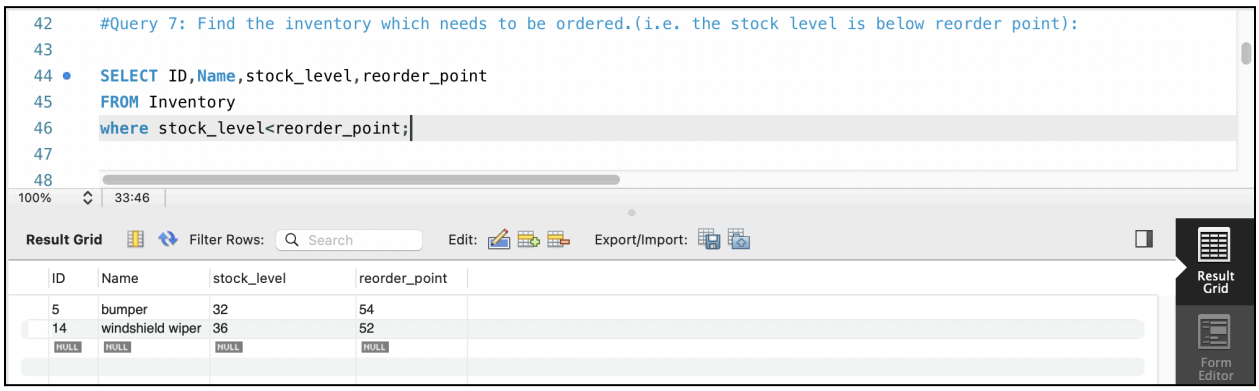
SQL command:

SELECT ID, Name,stock_level,reorder_point

FROM Inventory

where stock_level<reorder_point;

Output:



Query 8:

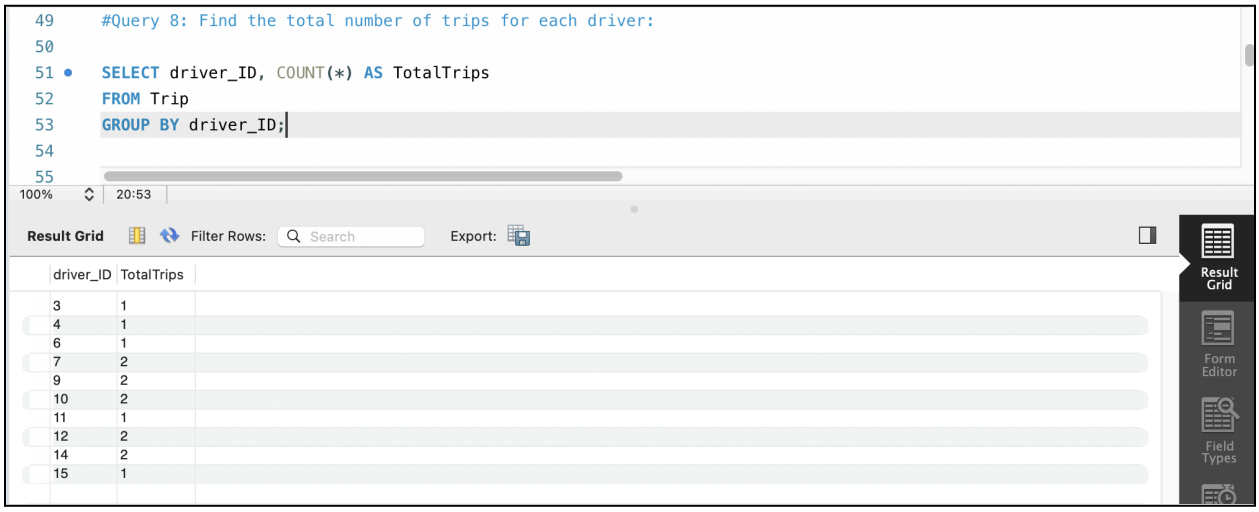
Query statement:

Find the total number of trips for each driver:

SQL command:

```
SELECT driver_ID, COUNT(*) AS TotalTrips
FROM Trip
GROUP BY driver_ID;
```

Output:



Query 9:

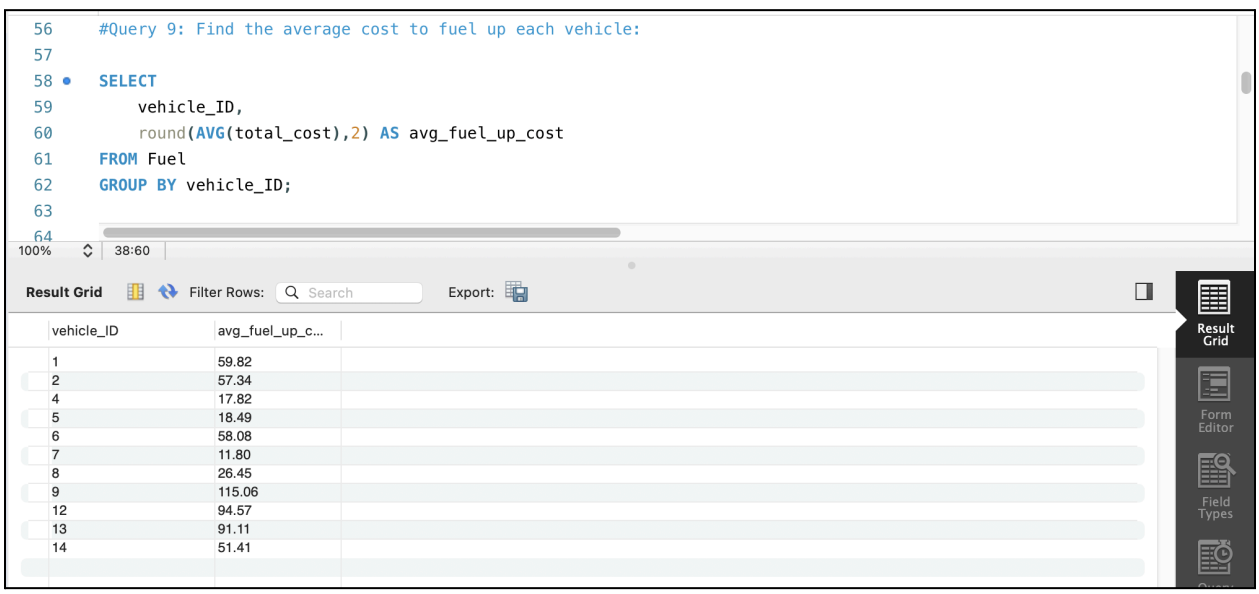
Query statement:

Find the average cost to fuel up each vehicle:

SQL command:

```
SELECT
    vehicle_ID,
    round(AVG(total_cost),2) AS avg_fuel_up_cost
FROM Fuel
GROUP BY vehicle_ID;
```

Output:



vehicle_ID	avg_fuel_up_c...
1	59.82
2	57.34
4	17.82
5	18.49
6	58.08
7	11.80
8	26.45
9	115.06
12	94.57
13	91.11
14	51.41

Query 10: (Duplicate) (Same as no.8)

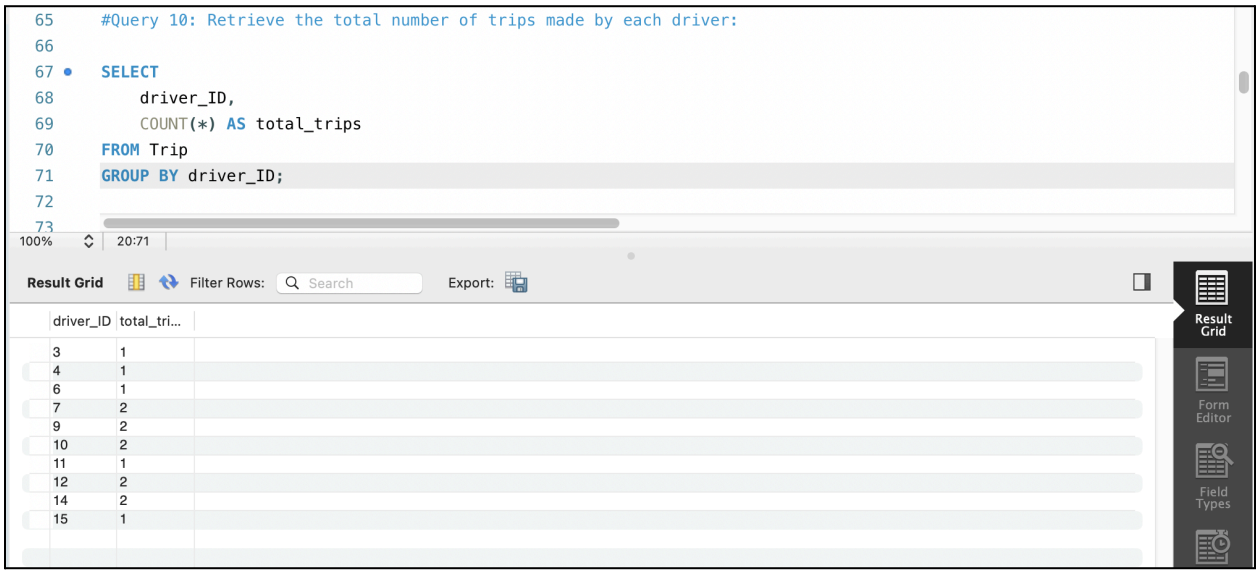
Query statement:

Retrieve the total number of trips made by each driver:

SQL command:

```
SELECT
    driver_ID,
    COUNT(*) AS total_trips
FROM Trip
GROUP BY driver_ID;
```

Output:



Query 11:

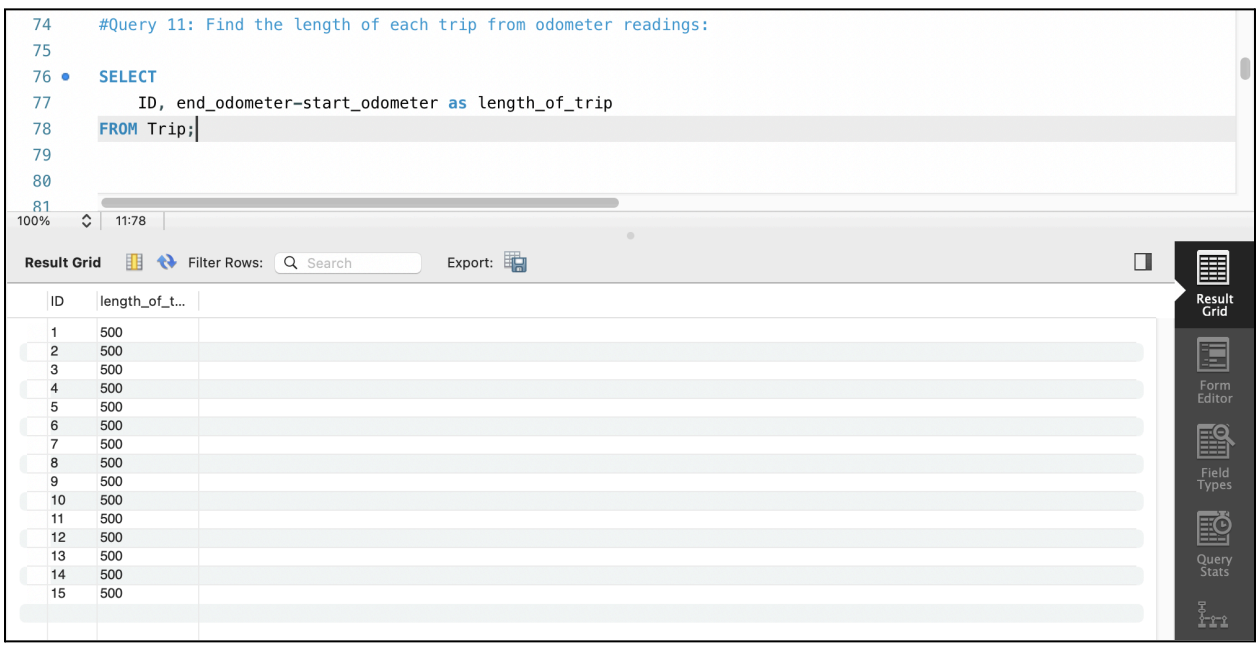
Query statement:

Find the length of each trip from odometer readings:

SQL command:

```
SELECT
    ID, end_odometer-start_odometer as length_of_trip
FROM Trip;
```

Output:



“Unusual result, because our data is in a way that all trips have the same length. Practical/real data can give different results.”

Query 12:

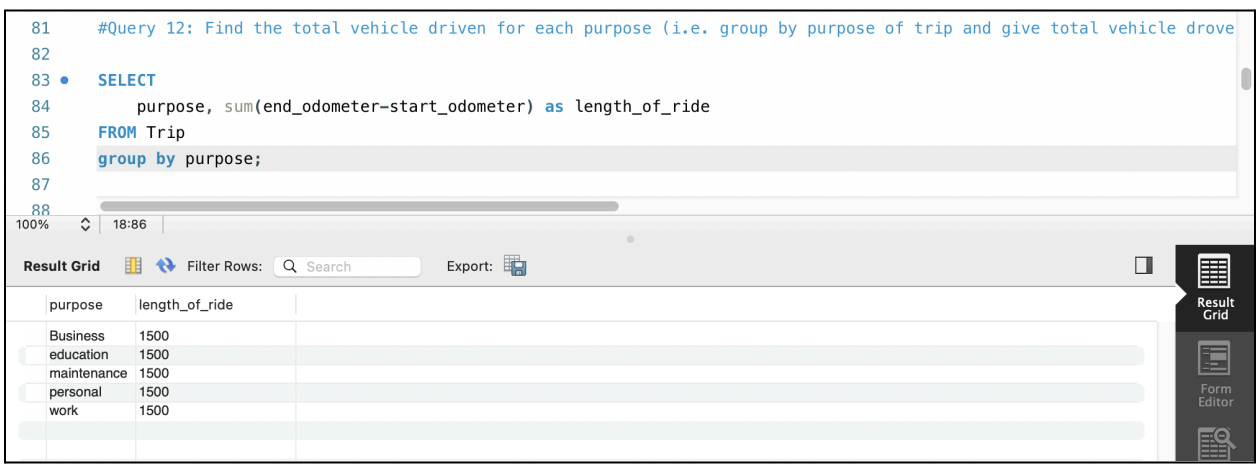
Query statement:

Find the total vehicle driven for each purpose (i.e. group by the purpose of the trip and give the total vehicle driven for that particular task):

SQL command:

```
SELECT
    purpose, sum(end_odometer-start_odometer) as length_of_ride
FROM Trip
group by purpose;
```

Output:



“Unusual result, because our data is in a way that all-purpose have the same number of trips and same number of trip length.

Practical/real data can give different results.”

Query 13:

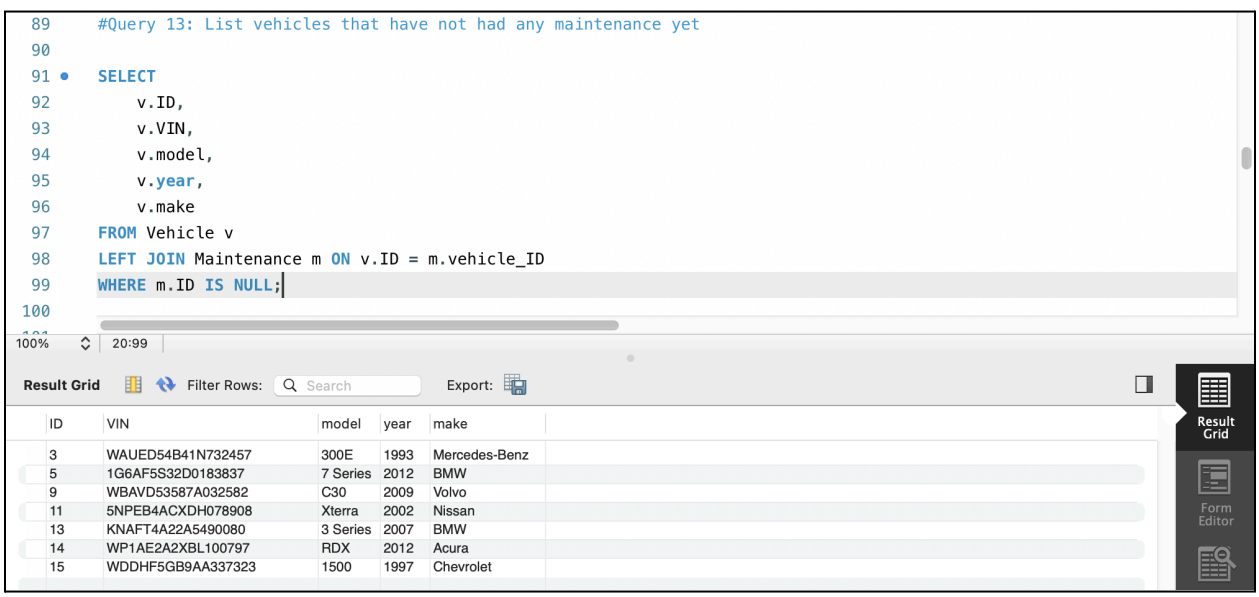
Query statement:

List vehicles that have not had any maintenance yet

SQL command:

```
SELECT
    v.ID,
    v.VIN,
    v.model,
    v.year,
    v.make
FROM Vehicle v
LEFT JOIN Maintenance m ON v.ID = m.vehicle_ID
WHERE m.ID IS NULL;
```

Output:



ID	VIN	model	year	make
3	WAUED54B41N732457	300E	1993	Mercedes-Benz
5	1G6AF5S32D0183837	7 Series	2012	BMW
9	WBAVD53587A032582	C30	2009	Volvo
11	5NPEB4ACXDH078908	Xterra	2002	Nissan
13	KNAFT4A22A5490080	3 Series	2007	BMW
14	WP1AE2A2XBL100797	RDX	2012	Acura
15	WDDHF5GB9AA337323	1500	1997	Chevrolet

Query 14:

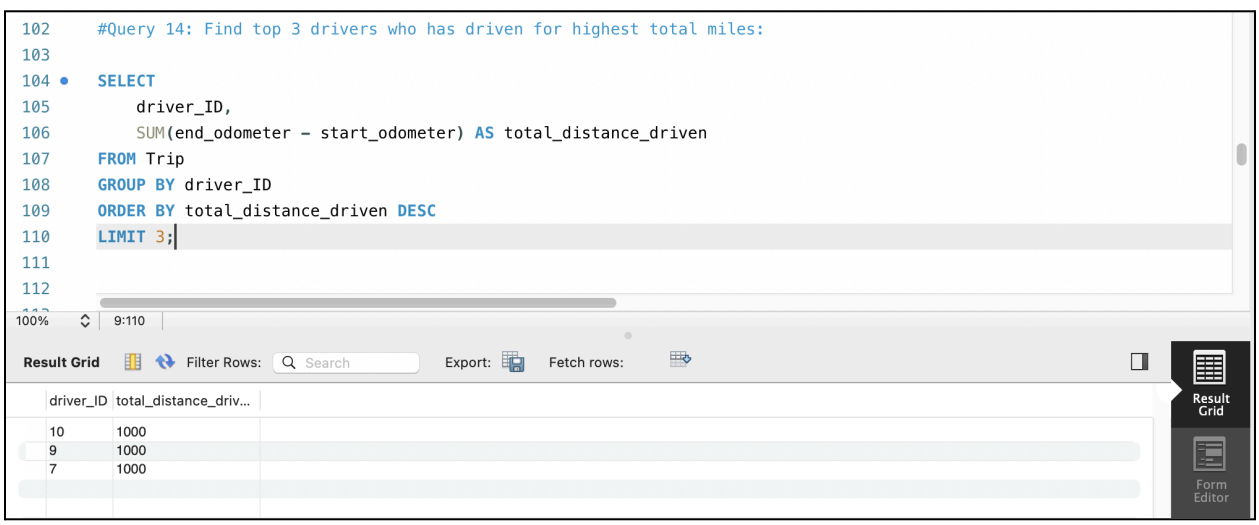
Query statement:

Find the top 3 drivers who have driven for the highest total miles:

SQL command:

```
SELECT
    driver_ID,
    SUM(end_odometer - start_odometer) AS total_distance_driven
FROM Trip
GROUP BY driver_ID
ORDER BY total_distance_driven DESC
LIMIT 3;
```

Output:



Query 15:

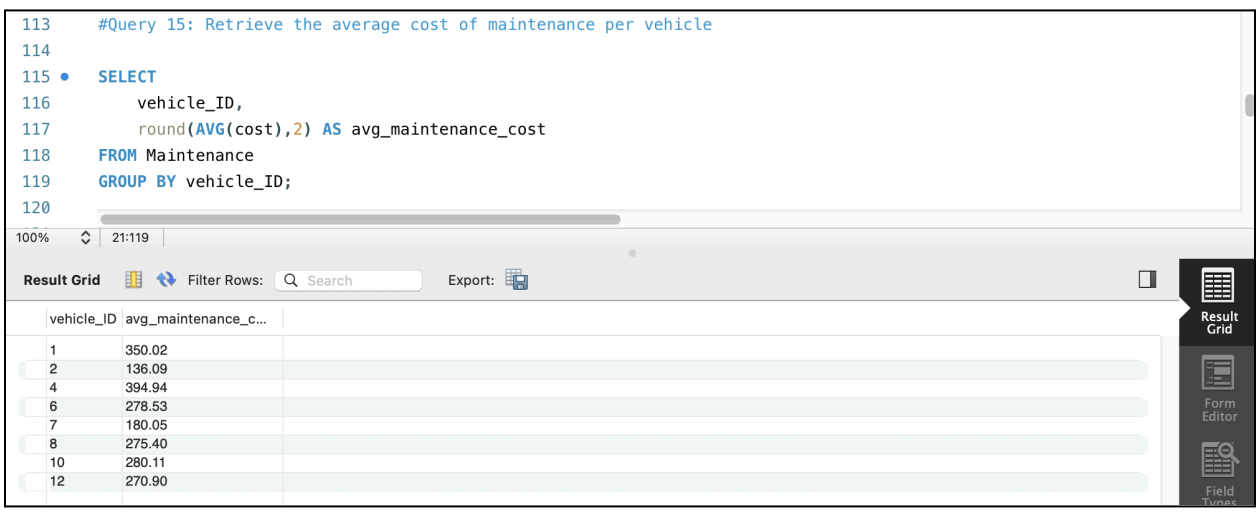
Query statement:

Retrieve the average cost of maintenance per vehicle

SQL command:

```
SELECT
    vehicle_ID,
    round(AVG(cost),2) AS avg_maintenance_cost
FROM Maintenance
GROUP BY vehicle_ID;
```

Output:



Query 16:

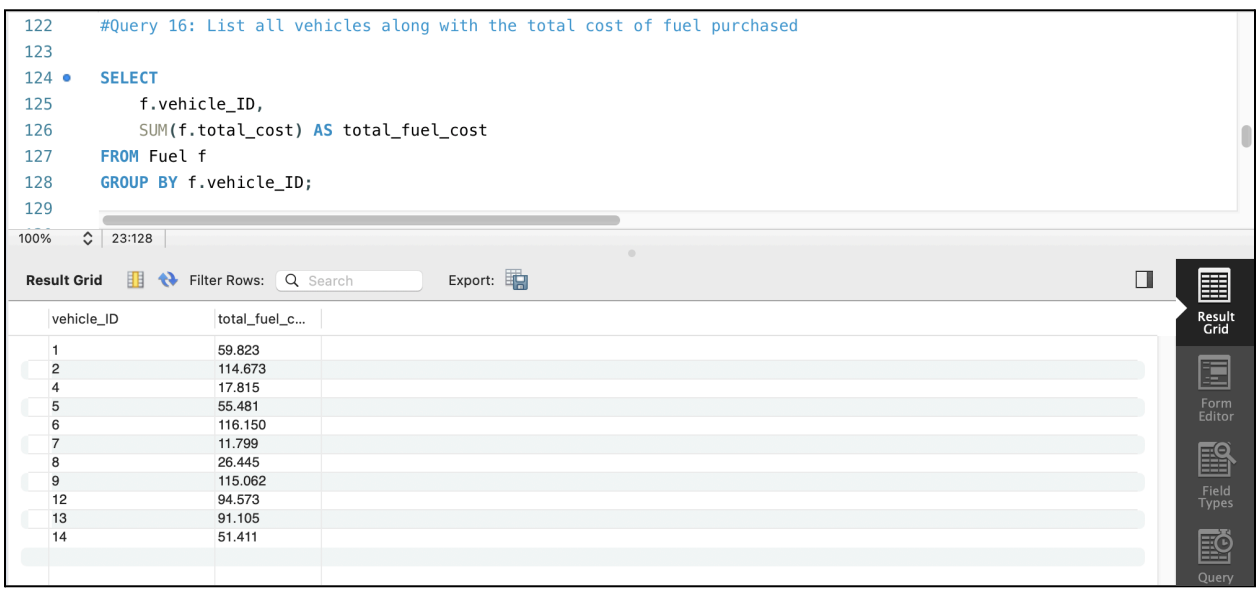
Query statement:

List all vehicles along with the total cost of fuel purchased

SQL command:

```
SELECT
    f.vehicle_ID,
    SUM(f.total_cost) AS total_fuel_cost
FROM Fuel f
GROUP BY f.vehicle_ID;
```

Output:



vehicle_ID	total_fuel_c...
1	59.823
2	114.673
4	17.815
5	55.481
6	116.150
7	11.799
8	26.445
9	115.062
12	94.573
13	91.105
14	51.411

Query 17:

Query statement:

Find details of the top 3 fuel spent:

SQL command:

SELECT *

FROM Fuel

ORDER BY total_cost DESC

LIMIT 3;

Output:

```
#Query 17: Find details of the top 3 fuel spent:

SELECT *
FROM Fuel
ORDER BY total_cost DESC
LIMIT 3;
```

100% 9:136

Result Grid Filter Rows: Search Edit: Export/Import: Fetch rows:

ID	vehicle_ID	date	gallons_purchased	odometer_readi...	price_per_gallon	total_cost	fuel_station
5	9	2023-07-22	25.40	57137	4.53	115.062	Pilot
7	12	2023-08-26	29.01	54209	3.26	94.573	Gulf
12	13	2023-03-08	24.23	53624	3.76	91.105	Speed
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Result Grid

Form Editor

Query 18:

Query statement:

Retrieve the average stock level for each inventory item:

SQL command:

```
SELECT
    name,
    AVG(stock_level) AS avg_stock_level
FROM Inventory
GROUP BY name;
```

Output:

```
#Query 18: Retrieve the average stock level for each inventory item:

SELECT
    name,
    AVG(stock_level) AS avg_stock_level
FROM Inventory
GROUP BY name;
```

100% 15:145

Result Grid Filter Rows: Search Export:

name	avg_stock_le...
air filter	153.0000
alternator	182.0000
brake pad	110.0000
bumper	32.0000
catalytic converter	97.0000
engine	64.0000
exhaust pipe	120.0000
fuel pump	79.0000
headlight	168.0000
radiator	78.0000
shock absorber	92.0000
spark plug	210.0000
steering wheel	148.0000
tire	55.0000
windshield wiper	36.0000

Result Grid Form Editor Field Types Query Stats

Query 19:

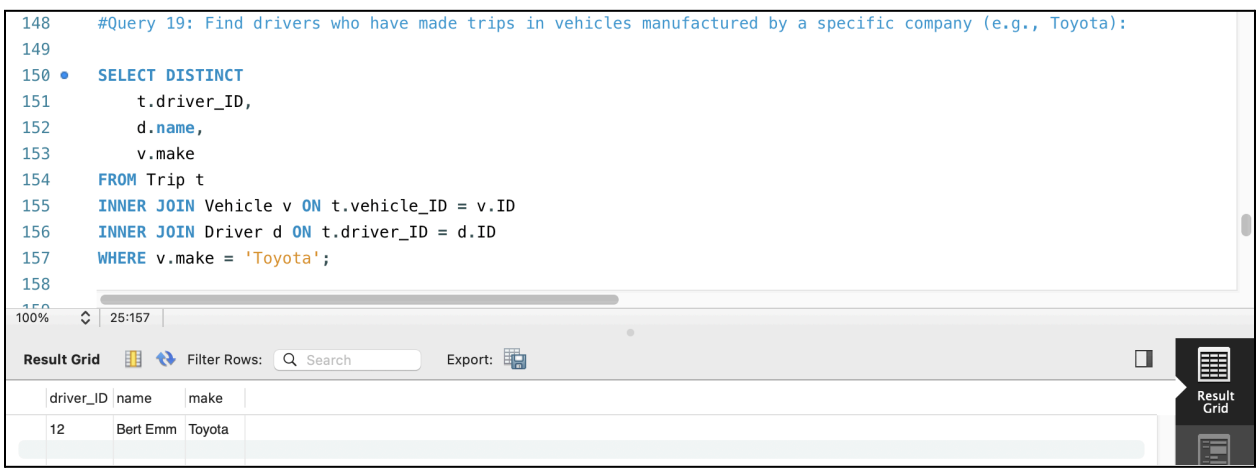
Query statement:

Find drivers who have made trips in vehicles manufactured by a specific company (e.g., Toyota):

SQL command:

```
SELECT DISTINCT
    t.driver_ID,
    d.name,
    v.make
FROM Trip t
INNER JOIN Vehicle v ON t.vehicle_ID = v.ID
INNER JOIN Driver d ON t.driver_ID = d.ID
WHERE v.make = 'Toyota';
```

Output:



Query 20:

Query statement:

Find the total cost of fuel purchased for each vehicle model in the past year, broken down by month:

SQL command:

```
SELECT
    v.model,
    EXTRACT(MONTH FROM f.date) AS month,
    SUM(f.total_cost) AS monthly_fuel_cost
FROM Fuel f
INNER JOIN Vehicle v ON f.vehicle_ID = v.ID
WHERE EXTRACT(YEAR FROM f.date) = 2023
GROUP BY v.model, month
ORDER BY v.model, month;
```

Output:

```
#Query 20: Find the total cost of fuel purchased for each vehicle model in the past year, broken down by month:

SELECT
    v.model,
    EXTRACT(MONTH FROM f.date) AS month,
    SUM(f.total_cost) AS monthly_fuel_cost
FROM Fuel f
INNER JOIN Vehicle v ON f.vehicle_ID = v.ID
WHERE EXTRACT(YEAR FROM f.date) = 2023
GROUP BY v.model, month
ORDER BY v.model, month;
```

100% 25:170

Result Grid Filter Rows: Search Export:

model	month	monthly_fuel_cost
3 Series	3	91.105
7 Series	6	19.039
7 Series	8	4.430
7 Series	9	32.012
Aspire	9	17.815
C30	7	115.062
Cruze	8	94.573
Jetta	11	26.445
Jetta	12	11.799
Odyssey	3	79.780
Odyssey	5	34.893
RDX	11	51.411
Reatta	3	59.823
V12 Va...	3	75.946
V12 Va...	8	40.204

Result Grid Form Editor Field Types Query Stats

Query 21:

Query statement:

Calculate the average maintenance cost per vehicle made for the last two years, grouped by year and make:

SQL command:

```
SELECT
    v.make,
    EXTRACT(YEAR FROM m.date) AS year,
    AVG(m.cost) AS avg_maintenance_cost
FROM Maintenance m
INNER JOIN Vehicle v ON m.vehicle_ID = v.ID
WHERE EXTRACT(YEAR FROM m.date) >= EXTRACT(YEAR FROM
CURRENT_DATE) - 7
GROUP BY v.make, EXTRACT(YEAR FROM m.date)
ORDER BY v.make, year;
```

Output:

```sql
#Query 21: Calculate the average maintenance cost per vehicle make for the last two years, grouped by year and make:

SELECT
    v.make,
    EXTRACT(YEAR FROM m.date) AS year,
    AVG(m.cost) AS avg_maintenance_cost
FROM Maintenance m
INNER JOIN Vehicle v ON m.vehicle_ID = v.ID
WHERE EXTRACT(YEAR FROM m.date) >= EXTRACT(YEAR FROM CURRENT_DATE) - 7
GROUP BY v.make, EXTRACT(YEAR FROM m.date)
ORDER BY v.make, year;
```

100% 23:183

Result Grid Filter Rows: Search Export:

make	year	avg_maintenance_c...
Aston Martin	2018	307.800000
Aston Martin	2020	249.250000
Buick	2018	363.810000
Chevrolet	2020	270.900000
Ford	2017	499.580000
Ford	2018	290.300000
Honda	2018	136.090000
Toyota	2017	280.250000
Toyota	2019	279.970000
Volkswagen	2017	181.040000
Volkswagen	2019	100.470000

Result Grid Form Editor Field Types

Query 22:

Query statement:

Identify locations where a specific item is stocked and its current quantity (example: item name: bumper)

SQL command:

SELECT l.id, l.address, name, i.stock_level

FROM Location l

JOIN Inventory i ON l.id = i.id

WHERE i.name = 'Bumper';

Output: