Abstract:

The goal of this project is to provide a realistic experience in the conceptual design, logical design, implementation, operation, and maintenance of a relational database and associated applications.

Problem Description:

The application is an automobile company, such as General Motors, Ford, Toyota, or Volkswagen (or maybe a company from yesteryear like Studebaker, Hudson, Nash, or Packard).

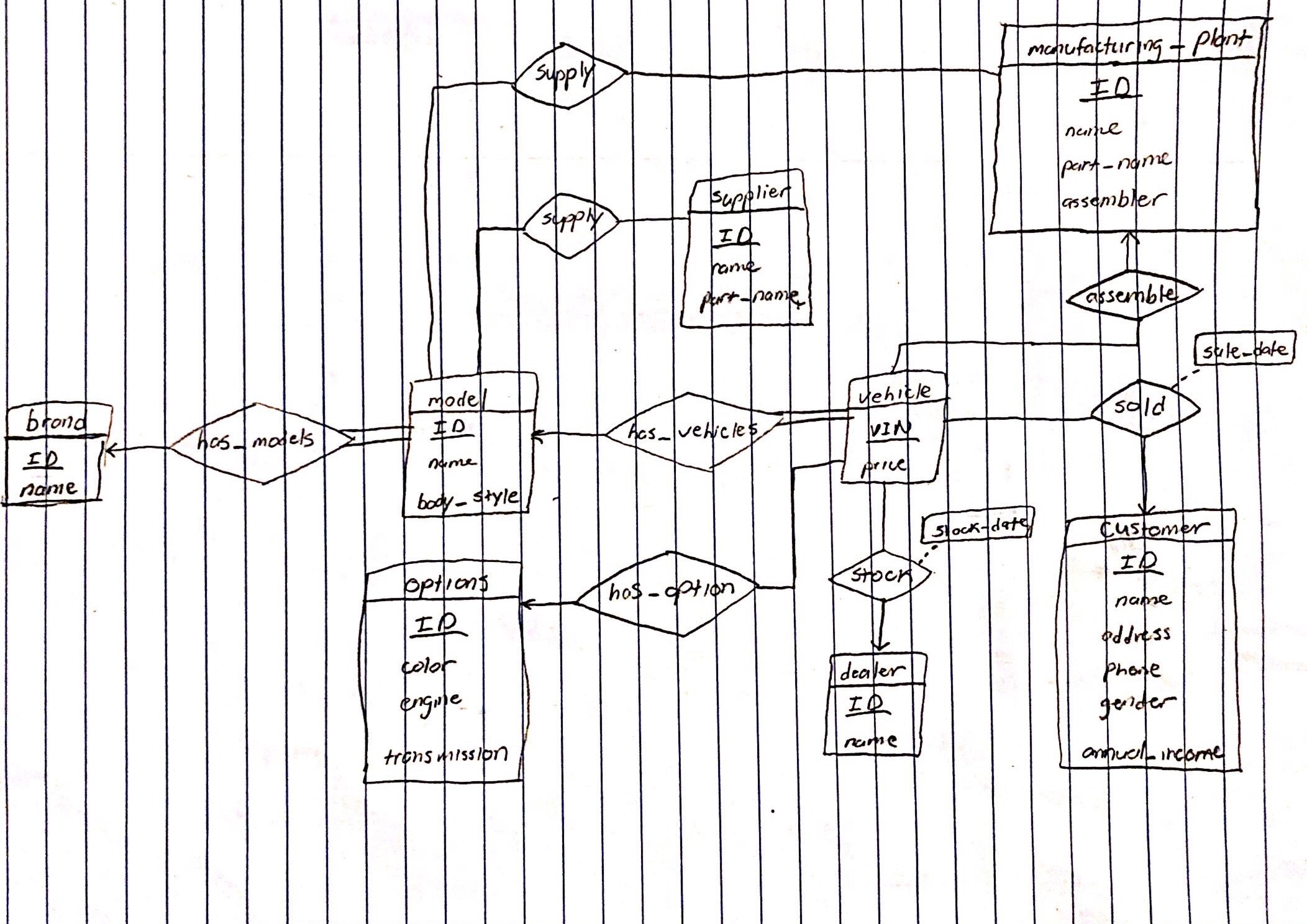
In our hypothetical company, it has been decided to redesign a major part of the database that

underlies company operations. Unfortunately, the manager assigned to solicit database design

proposals is not very computer literate and is unable to provide a very detailed specification at

the technical level. Fortunately, you are able to do that.

E-R Diagram:



Relational Schema:

*brand(brand\_id, brand\_name)*

*model(model\_id, brand\_id, model\_name, body\_style)*

*vehicle(VIN, model\_id, option\_id, plant\_id, dealer\_id, price, stock\_date)*

*option(option\_id, color, engine, transmission)*

*customer(customer\_id, customer\_name, address, phone\_no, gender, annual\_income)*

*dealer(dealer\_id, dealer\_name)*

*supplier(supplier\_id, supplier\_name, part\_name)*

*manufacturing\_plant(plant\_id, plant\_name, part\_name, assembler)*

*supply(supplier\_id, model\_id,*

**foreign key** *supplier\_id* **references** *supplier,*

**foreign key** *model\_id* **references** *model)*

*sold(VIN, customer\_id, sale\_date,*

**foreign key** *VIN* **references** *vehicle*,

**foreign key** *customer\_id* **references** *customer)*

*stock(VIN, dealer\_id, stock\_date,*

**foreign key** *VIN* **references** *vehicle,*

**foreign key** *dealer\_id* **references** *dealer)*

*assemble(VIN, plant\_id,*

**foreign key** *VIN* **references** *vehicle,*

**foreign key** *plant\_id* **references** *manufacturing\_plant)*

*has\_models(brand\_name, model\_id,*

**foreign key** *brand\_name* **references** *brand,*

**foreign key** *model\_id* **references** *model)*

*has\_vehicles(model\_id, VIN,*

**foreign key** *VIN* **references** *vehicle,*

**foreign key** *model\_id* **references** *model)*

*has\_option(VIN, model\_id, option\_id,*

**foreign key** *VIN* **references** *vehicle,*

**foreign key** *model\_id* **references** *model,*

**foreign key** *option\_id* **references** *option)*

Implementation Details:

* MySQL database created using the SQL language in MySQL Workbench 8.0 CE
* MySQL Command-Line Client used as the interface to implement any queries, including the sample queries listed below
  + For the administrator interface, commands can be entered in the command line after connecting the database to the terminal.
  + The marketing department can also enter any OLAP query statements through the command line, just like the administrator.

Running Results and Analysis:

Query images and their code (Images also provided in “Results” folder under “Query Screenshots” if the images here are too small to read):

* Show sales trends for various brands over the past 3 years, by year, month, week. Then break these data out by gender of the buyer and then by income range.
  + By year:

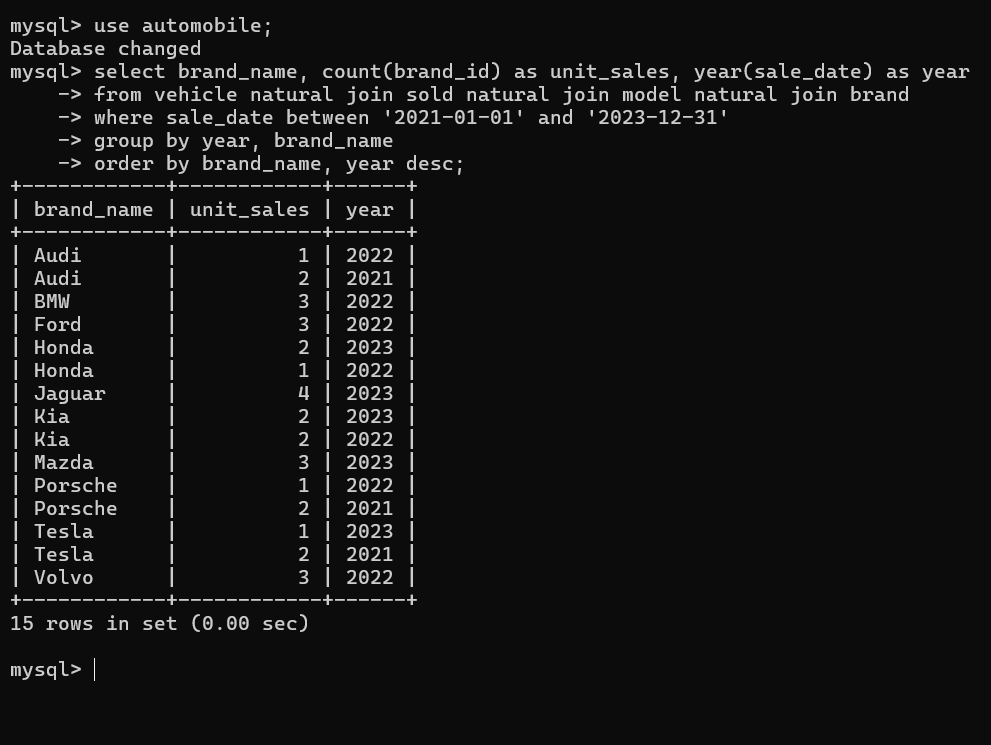
select brand\_name, count(brand\_id) as unit\_sales, year(sale\_date) as year

from vehicle natural join sold natural join model natural join brand

where sale\_date between '2021-01-01' and '2023-12-31'

group by year, brand\_name

order by brand\_name, year desc;



* + By month:

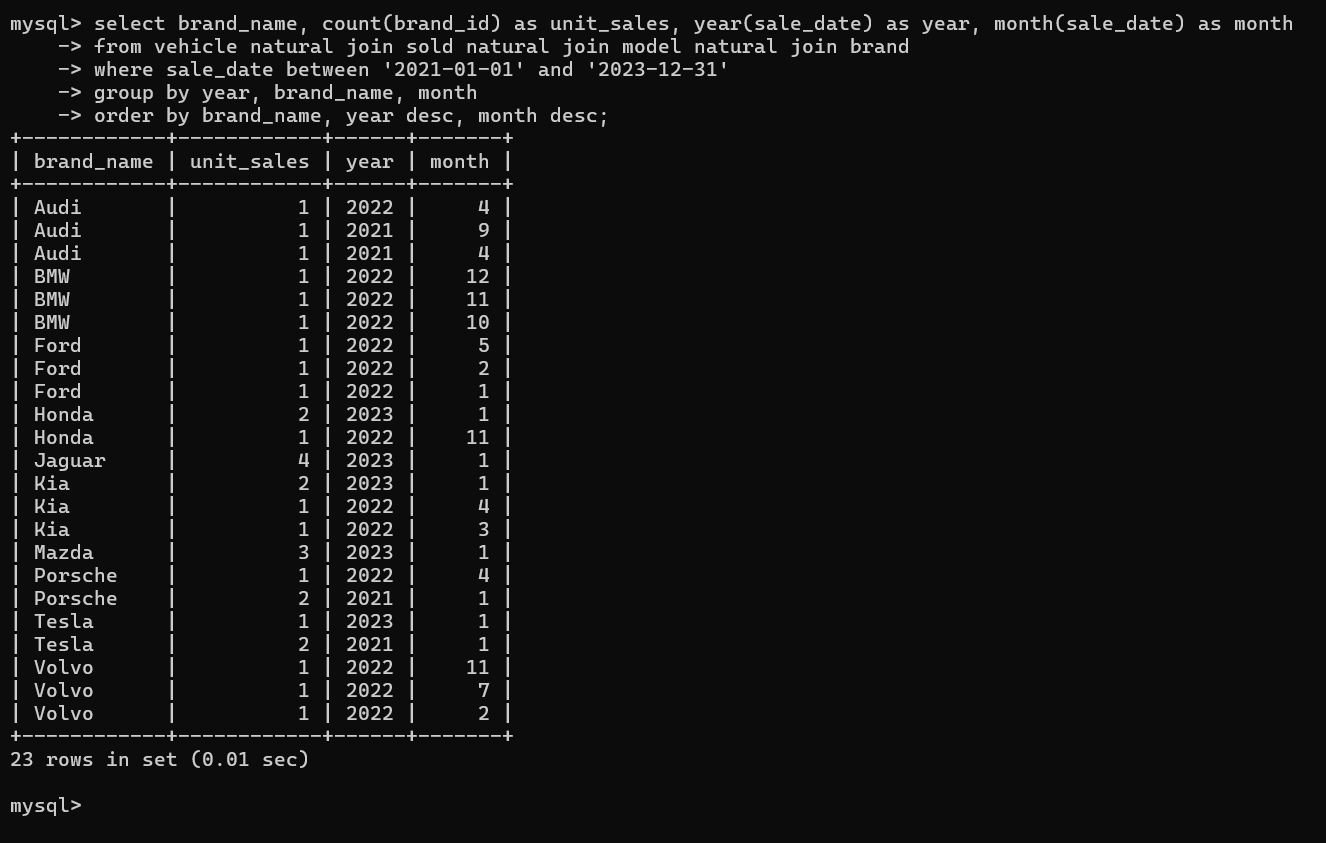
select brand\_name, count(brand\_id) as unit\_sales, year(sale\_date) as year, month(sale\_date) as month

from vehicle natural join sold natural join model natural join brand

where sale\_date between '2021-01-01' and '2023-12-31'

group by year, brand\_name, month

order by brand\_name, year desc, month desc;



* + By week:

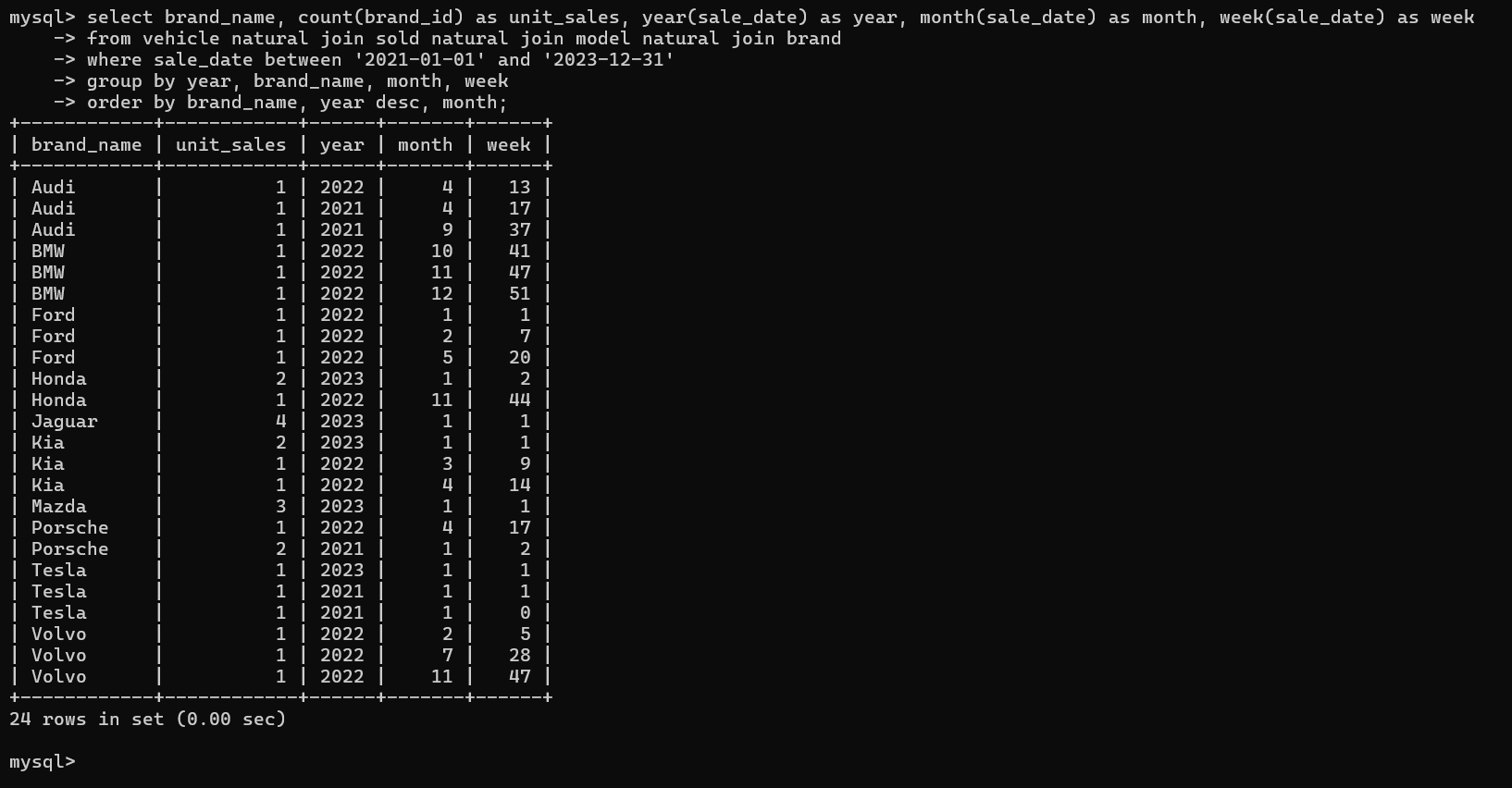
select brand\_name, count(brand\_id) as unit\_sales, year(sale\_date) as year, month(sale\_date) as month, week(sale\_date) as week

from vehicle natural join sold natural join model natural join brand

where sale\_date between '2021-01-01' and '2023-12-31'

group by year, brand\_name, month, week

order by brand\_name, year desc, month;



* + By gender:
    - Both genders:

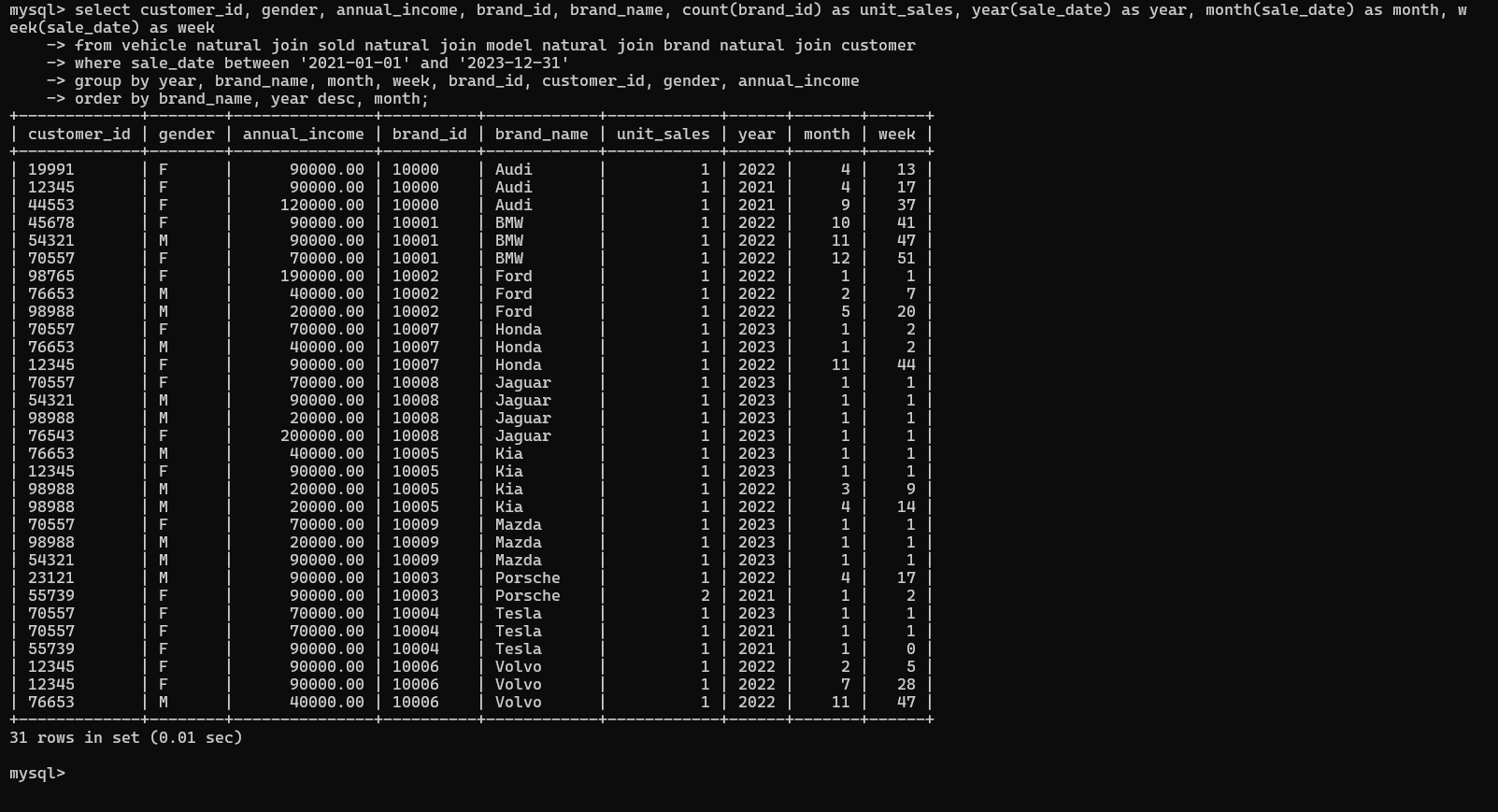
select customer\_id, gender, annual\_income, brand\_id, brand\_name, count(brand\_id) as unit\_sales, year(sale\_date) as year, month(sale\_date) as month, week(sale\_date) as week

from vehicle natural join sold natural join model natural join brand natural join customer

where sale\_date between '2021-01-01' and '2023-12-31'

group by year, brand\_name, month, week, brand\_id, customer\_id, gender, annual\_income

order by brand\_name, year desc, month;



* + - Male:

select customer\_id, gender, annual\_income, brand\_id, brand\_name, count(brand\_id) as

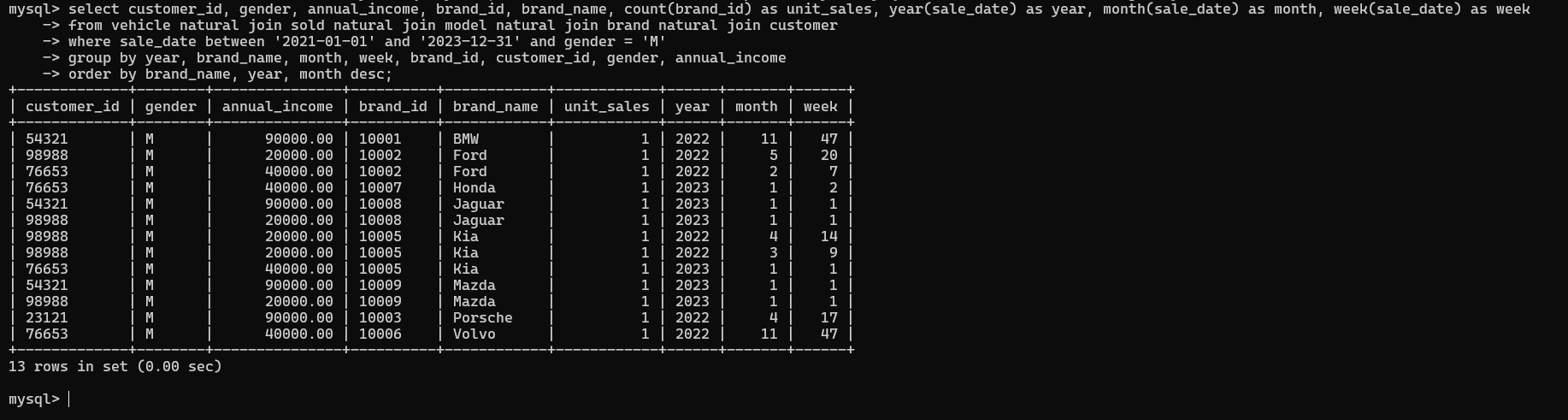
unit\_sales, year(sale\_date) as year, month(sale\_date) as month, week(sale\_date) as week

from vehicle natural join sold natural join model natural join brand natural join customer

where sale\_date between '2021-01-01' and '2023-12-31' and gender = 'M'

group by year, brand\_name, month, week, brand\_id, customer\_id, gender, annual\_income

order by brand\_name, year, month desc;



* + - Female:

select customer\_id, gender, annual\_income, brand\_id, brand\_name, count(brand\_id) as

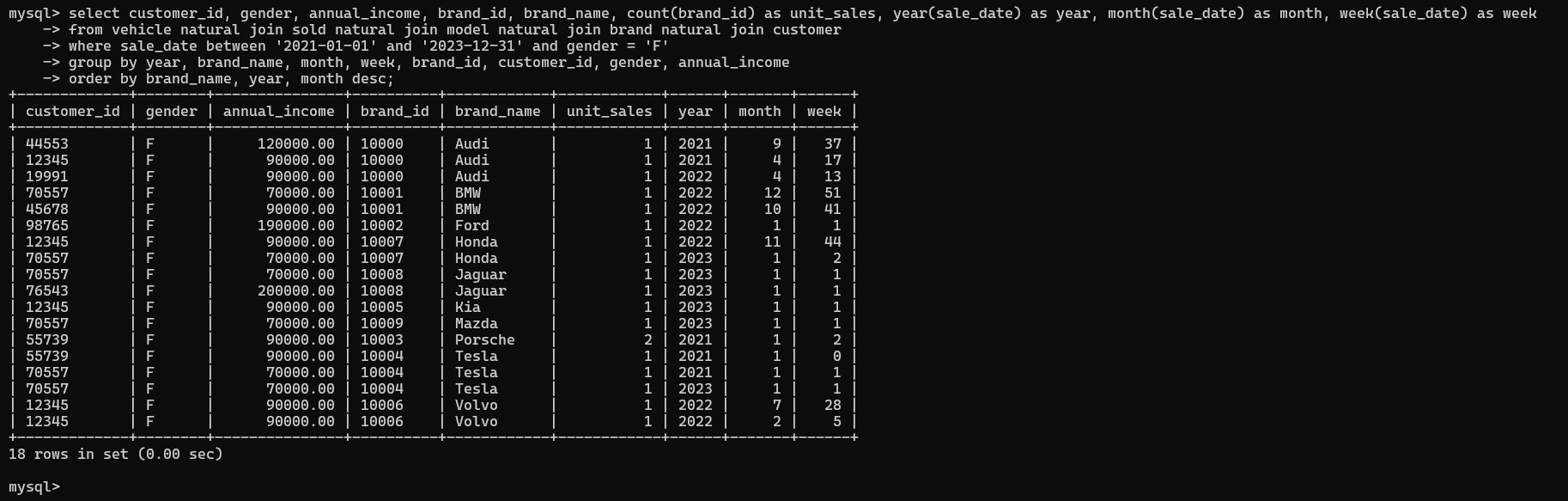
unit\_sales, year(sale\_date) as year, month(sale\_date) as month, week(sale\_date) as week

from vehicle natural join sold natural join model natural join brand natural join customer

where sale\_date between '2021-01-01' and '2023-12-31' and gender = 'F'

group by year, brand\_name, month, week, brand\_id, customer\_id, gender, annual\_income

order by brand\_name, year, month desc;



* + Income > 70,000

select customer\_id, gender, annual\_income, brand\_id, brand\_name, count(brand\_id) as

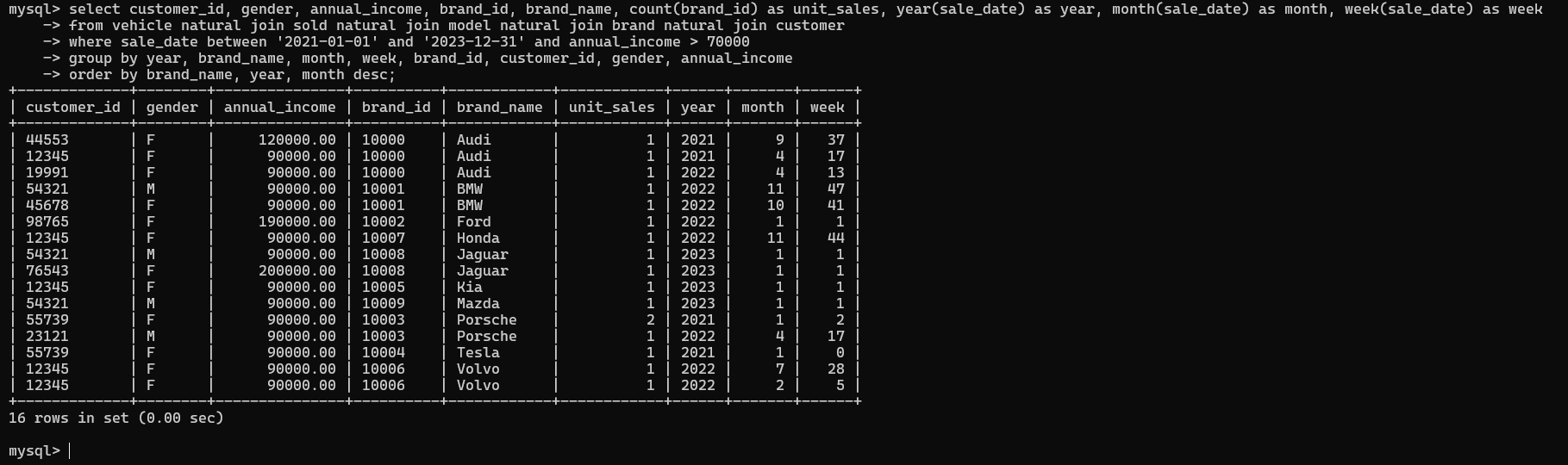
unit\_sales, year(sale\_date) as year, month(sale\_date) as month, week(sale\_date) as week

from vehicle natural join sold natural join model natural join brand natural join customer

where sale\_date between '2021-01-01' and '2023-12-31' and annual\_income > 70000

group by year, brand\_name, month, week, brand\_id, customer\_id, gender, annual\_income

order by brand\_name, year, month desc;



* + Income <= 70,000

select customer\_id, gender, annual\_income, brand\_id, brand\_name, count(brand\_id) as

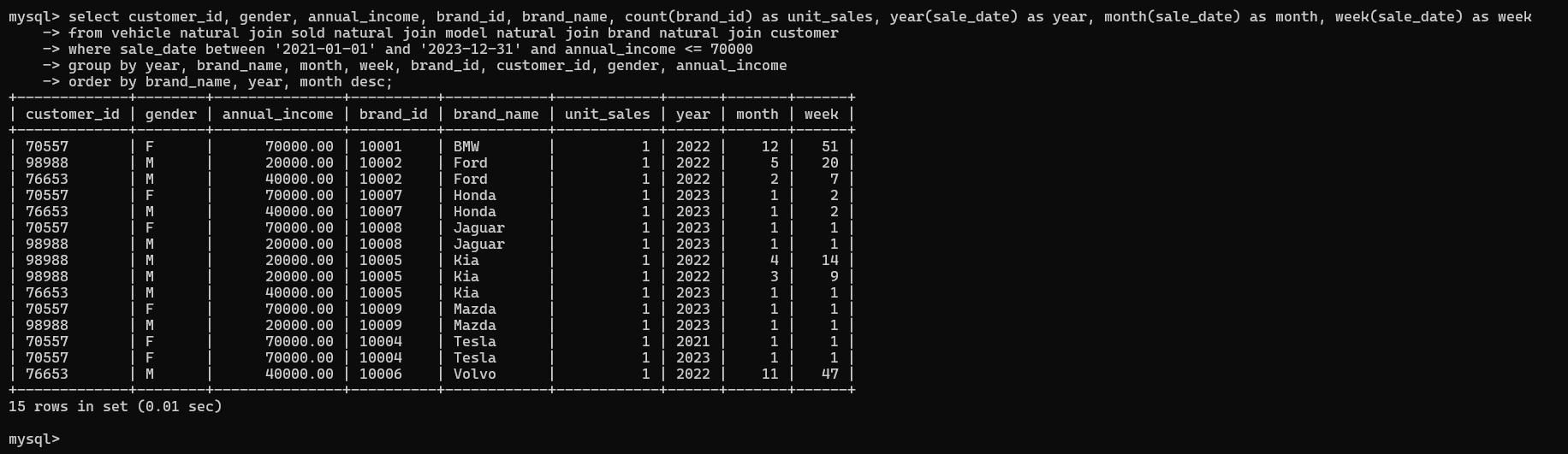
unit\_sales, year(sale\_date) as year, month(sale\_date) as month, week(sale\_date) as week

from vehicle natural join sold natural join model natural join brand natural join customer

where sale\_date between '2021-01-01' and '2023-12-31' and annual\_income <= 70000

group by year, brand\_name, month, week, brand\_id, customer\_id, gender, annual\_income

order by brand\_name, year, month desc;



* Suppose that it is found that transmissions made by supplier Getrag between two given dates are defective. Find the VIN of each car containing such a transmission and the customer to which it was sold. If your design allows, suppose the defective transmissions all come from only one of Getrag’s plants.

create view vehicle\_supplier\_sold(VID, supplier\_name, customer\_id, sale\_date) as

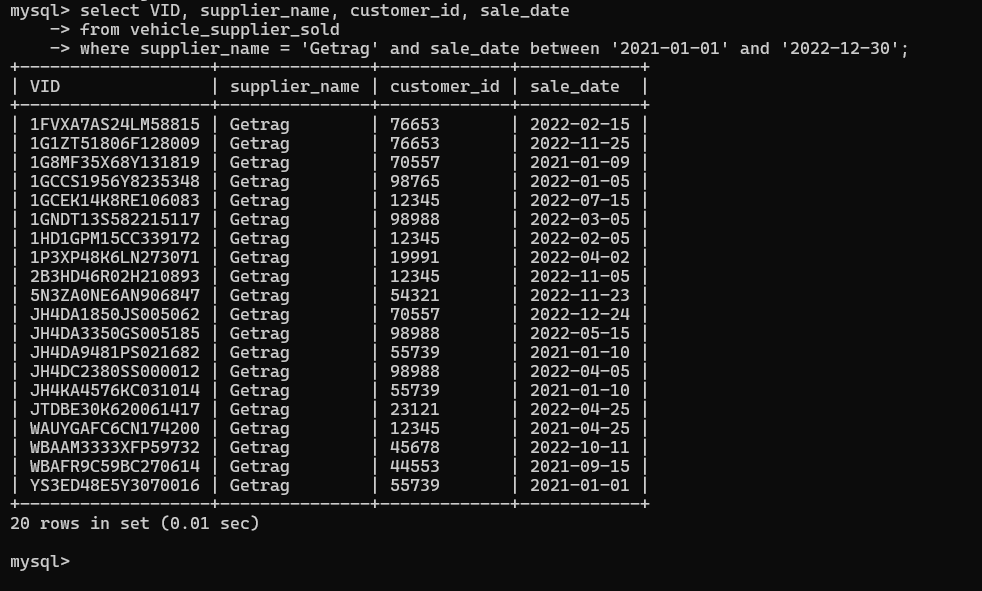
select VIN, supplier\_name, customer\_id, sale\_date

from vehicle natural join model natural join supplier natural join sold;

select VID, supplier\_name, customer\_id, sale\_date

from vehicle\_supplier\_sold

where supplier\_name = 'Getrag' and sale\_date between '2021-01-01' and '2022-12-30';



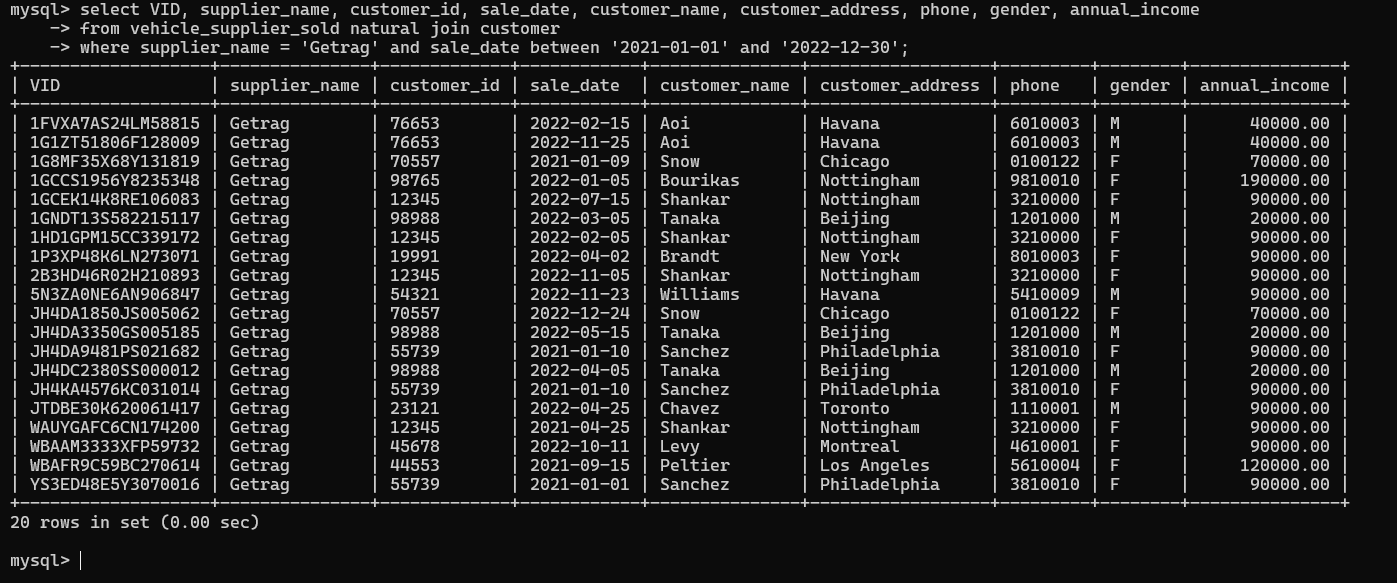
* + Same query but with more customer information:

select VID, supplier\_name, customer\_id, sale\_date, customer\_name, customer\_address,

phone, gender, annual\_income

from vehicle\_supplier\_sold natural join customer

where supplier\_name = 'Getrag' and sale\_date between '2021-01-01' and '2022-12-30';



* Find the top 2 brands by dollar-amount sold in the past year.

with past\_year\_sold(brand\_id, sale\_date, price) as

(select brand\_id, sale\_date, price

from vehicle natural join model natural join sold

where year(sale\_date) = 2022)

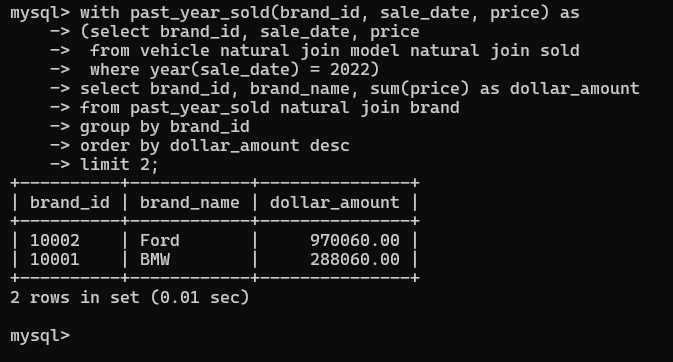
select brand\_id, brand\_name, sum(price) as dollar\_amount

from past\_year\_sold natural join brand

group by brand\_id

order by dollar\_amount desc

limit 2;



* + OLAP results for the query:

with past\_year\_sold(brand\_id, past\_year, dollar\_amount) as

(select brand\_id, year(sale\_date) as past\_year, sum(price) as dollar\_amount

from vehicle natural join model natural join sold

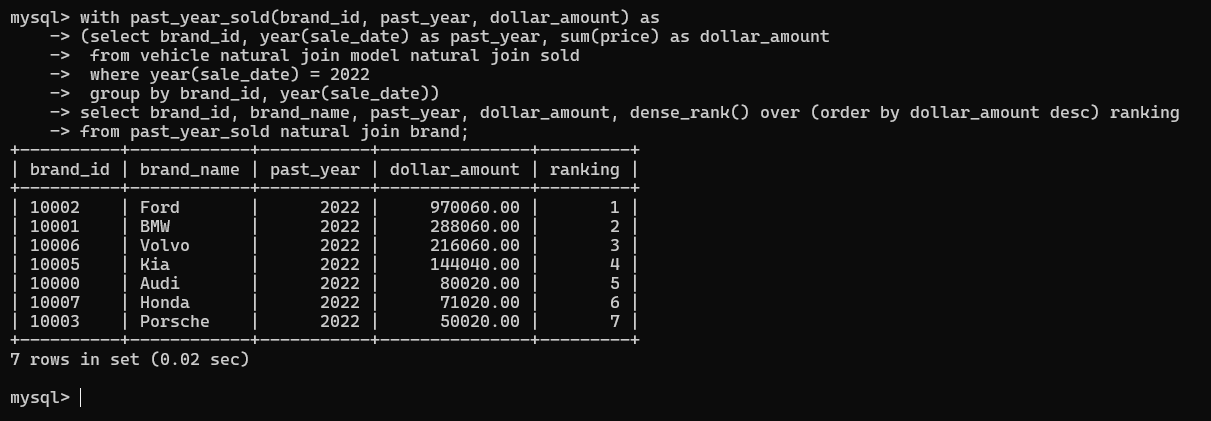
where year(sale\_date) = 2022

group by brand\_id, year(sale\_date))

select brand\_id, brand\_name, past\_year, dollar\_amount,

dense\_rank() over (order by dollar\_amount desc) ranking

from past\_year\_sold natural join brand;



with past\_year\_sold(brand\_id, past\_year, dollar\_amount) as

(select brand\_id, year(sale\_date) as past\_year, sum(price) as dollar\_amount

from vehicle natural join model natural join sold

where year(sale\_date) = 2022

group by brand\_id, year(sale\_date))

select \* from

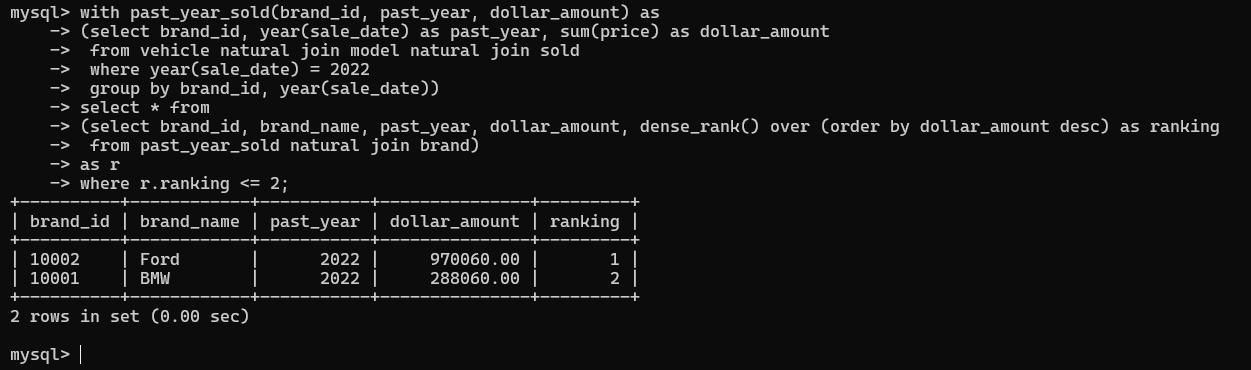
(select brand\_id, brand\_name, past\_year, dollar\_amount,

dense\_rank() over (order by dollar\_amount desc) as ranking

from past\_year\_sold natural join brand)

as r

where r.ranking <= 2;



* Find the top 2 brands by unit sales in the past year.

with past\_year\_sold(brand\_id, sale\_date) as

(select brand\_id, sale\_date

from vehicle natural join model natural join sold

where year(sale\_date) = 2022)

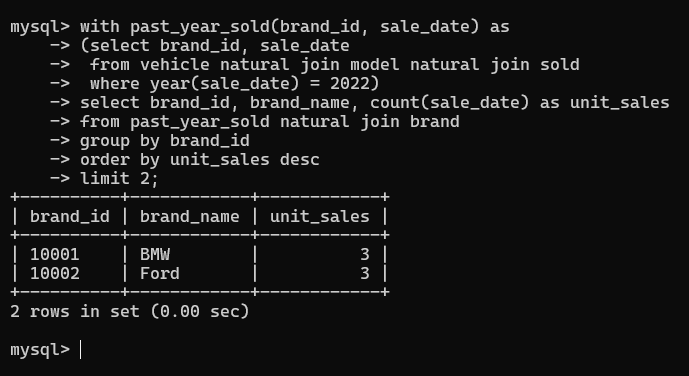
select brand\_id, brand\_name, count(sale\_date) as unit\_sales

from past\_year\_sold natural join brand

group by brand\_id

order by unit\_sales desc

limit 2;



* + OLAP results for the query:

with past\_year\_sold(brand\_id, past\_year, unit\_sales) as

(select brand\_id, year(sale\_date) as past\_year, count(price) as unit\_sales

from vehicle natural join model natural join sold

where year(sale\_date) = 2022

group by brand\_id, year(sale\_date))

select \* from

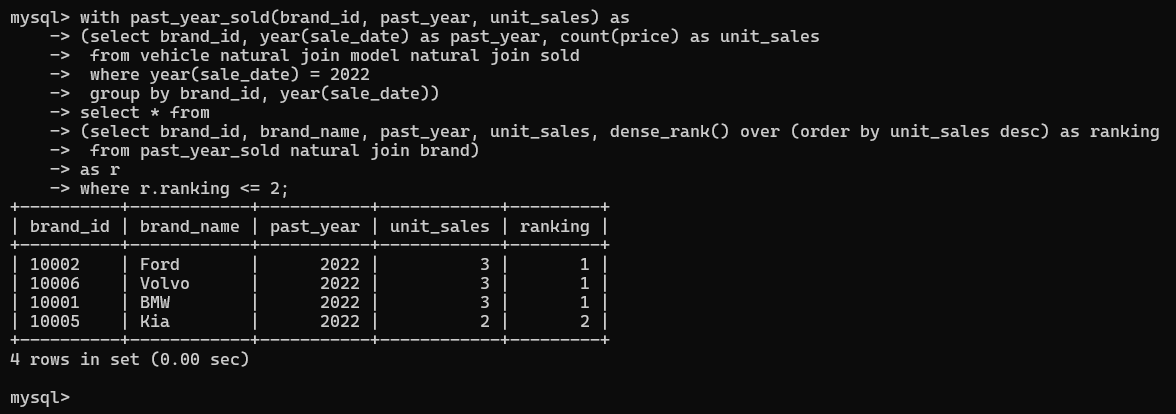
(select brand\_id, brand\_name, past\_year, unit\_sales,

dense\_rank() over (order by unit\_sales desc) as ranking

from past\_year\_sold natural join brand)

as r

where r.ranking <= 2;



* In what month(s) do convertibles sell best?

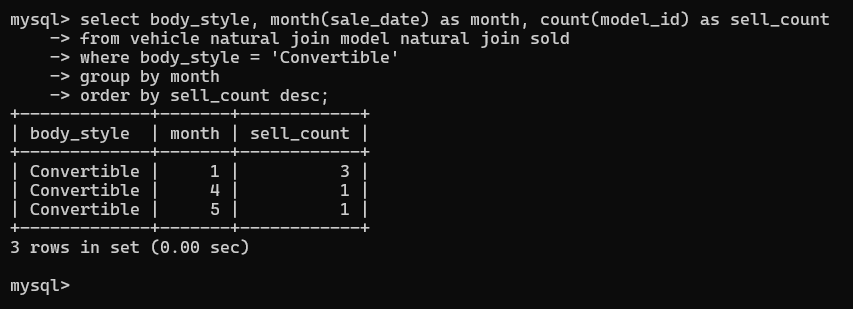
select body\_style, month(sale\_date) as month, count(model\_id) as sell\_count

from vehicle natural join model natural join sold

where body\_style = 'Convertible'

group by month

order by sell\_count desc;

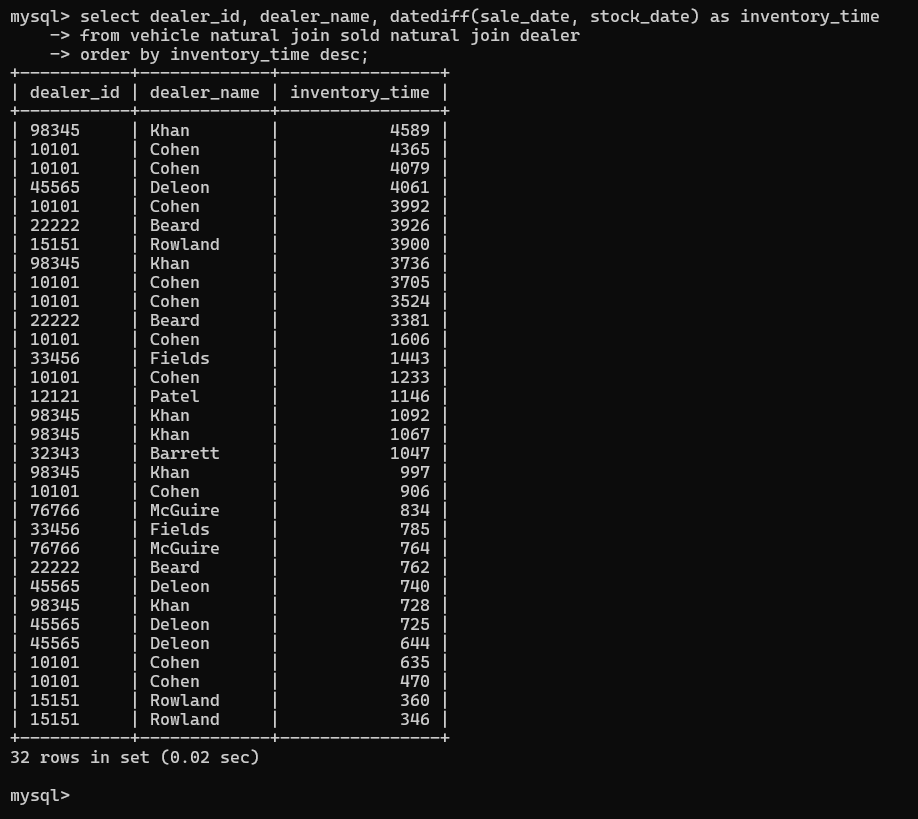


* Find those dealers who keep a vehicle in inventory for the longest average time.

select dealer\_id, dealer\_name, datediff(sale\_date, stock\_date) as inventory\_time

from vehicle natural join sold natural join dealer

order by inventory\_time desc;



Conclusions:

From this project I have learned:

* How to create a database in MySQL using the SQL query language.
* How to populate the database with sufficient and interesting data.
* How to connect the database to an interface (CLI) and run proper queries to display information that is required.