--Create tables and import csv files

create table orders (

row\_id int

, order\_id varchar(50)

, order\_date date

, ship\_date date

, ship\_mode varchar(20)

, customer\_id varchar(50)

, product\_id varchar(50)

, country varchar(20)

, city varchar(20)

, state varchar(20)

, postal\_code int

, region varchar(10)

, sales decimal(10,2)

, constraint "pk\_orders" primary key (

"row\_id")

);

create table customers (

customer\_id varchar(50)

, customer\_name varchar(50)

, segment varchar(20)

, constraint "pk\_customers" primary key (

"customer\_id")

);

create table products (

product\_id varchar(50)

, category varchar(20)

, sub\_category varchar(20)

, constraint "pk\_products" primary key (

"product\_id")

);

/\*

Get a comprehensive view of store revenue. This query gathers

data across various aspects of each sale, such as the total items,

total revenues and combines it with customer information. The aim

is to get a deeper insight into revenue trends in multiple directions,

such as over time, by region, and by customer segment

\*/

select

o.order\_date

, o.order\_id

, c.customer\_name

, c.segment

, o.state

, o.region

, count(p.sub\_category) as total\_items -- Counts the number of items per order

, sum(o.sales) as total\_revenues -- Sum of sales per order

from

orders o

left join customers c

on o.customer\_id = c.customer\_id

left join products p

on o.product\_id = p.product\_id

group by

o.order\_date

, o.order\_id

, c.customer\_name

, c.segment

, o.state

, o.region

order by

o.order\_date

;

A screenshot of a computer

Description automatically generated

/\*

The query focuses on sales data by product category. This provides the

performance of product categories on daily, monthly, and yearly basics,

allowing us to track which categories perform best on a day-to-day basis

and vice versa. Tt is necessary for identifying market trends and making

informed decisions about marketing strategies, and sales forecasting.

\*/

select

o.order\_date

, p.category

, count(p.product\_id) as total\_item -- Counts the number of items per category

, sum(o.sales) as total\_sale -- Sum of sales per category

from

orders o

left join products p

on o.product\_id = p.product\_id

group by

o.order\_date

, p.category

order by

o.order\_date

;

A screenshot of a list of items

Description automatically generated