DROP TABLE IF EXISTS sales;

DROP TABLE IF EXISTS customers;

DROP TABLE IF EXISTS products;

DROP TABLE IF EXISTS city;

-- Import Rules

-- 1st import to city

COPY city

FROM 'C:\\Program Files\\PostgreSQL\\16\\data\\data\_copy\\monday\_coffee\\city.csv'

DELIMITER ','

CSV HEADER;

-- 2nd import to products

COPY products

FROM 'C:\\Program Files\\PostgreSQL\\16\\data\\data\_copy\\monday\_coffee\\products.csv'

DELIMITER ','

CSV HEADER;

-- 3rd import to customers

COPY customers

FROM 'C:\\Program Files\\PostgreSQL\\16\\data\\data\_copy\\monday\_coffee\\customers.csv'

DELIMITER ','

CSV HEADER;

-- 4th import to sales

COPY sales

FROM 'C:\\Program Files\\PostgreSQL\\16\\data\\data\_copy\\monday\_coffee\\sales.csv'

DELIMITER ','

CSV HEADER;

CREATE TABLE city

(

city\_id INT PRIMARY KEY,

city\_name VARCHAR(15),

population BIGINT,

estimated\_rent FLOAT,

city\_rank INT

);

CREATE TABLE customers

(

customer\_id INT PRIMARY KEY,

customer\_name VARCHAR(25),

city\_id INT,

CONSTRAINT fk\_city FOREIGN KEY (city\_id) REFERENCES city(city\_id)

);

CREATE TABLE products

(

product\_id INT PRIMARY KEY,

product\_name VARCHAR(35),

Price float

);

CREATE TABLE sales

(

sale\_id INT PRIMARY KEY,

sale\_date date,

product\_id INT,

customer\_id INT,

total FLOAT,

rating INT,

CONSTRAINT fk\_products FOREIGN KEY (product\_id) REFERENCES products(product\_id),

CONSTRAINT fk\_customers FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id)

);

--Monday Coffee -- Data Analysis

select \* from city;

select \* from customers;

## --Q1. How many peoples in each city consume coffee, given that 25% of the population does?

select

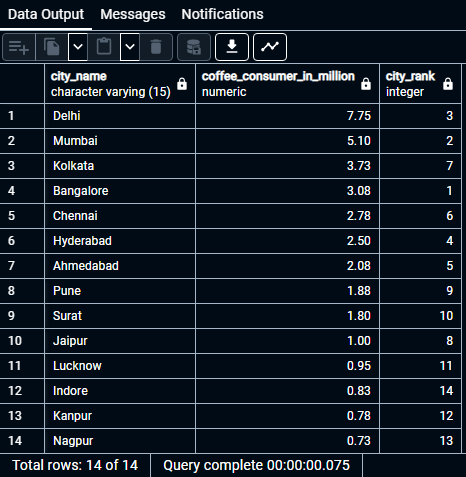
city\_name,

round((population \*0.25)/1000000, 2) as coffee\_consumer\_in\_million,

city\_rank

from city

order by 2 desc;



## --Q2 what is the total revenue generated from coffee sales across all citites in the last quarter of 2023?

select ci.city\_name,

sum(s.total)

from sales as s

join customers as c

on s.customer\_id = c.customer\_id

join city as ci

on ci.city\_id= c.city\_id

where

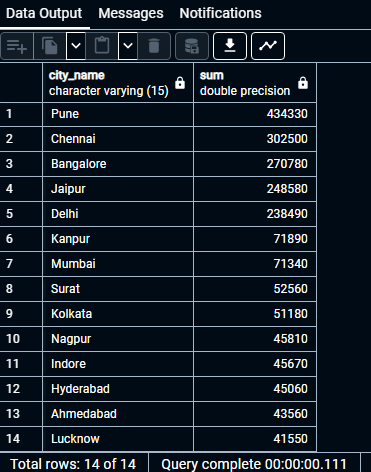
extract(year from s.sale\_date) = 2023

and

extract(quarter from s.sale\_date) =4

group by 1

order by 2 desc



## --Q3 How many units of each coffee products have been sold?

SELECT

p.product\_name,

count(s.sale\_id) as total\_orders

from products as p

left join

sales as s

on p.product\_id = s.product\_id

group by 1

order by 2 desc

A screenshot of a computer

Description automatically generated

## --Q4. What is the average sales amount per customer in each city?

select

ci.city\_name,

sum(s.total) as total\_revenue,

count( distinct s.customer\_id) as total\_cx,

round(

sum(s.total)::numeric /count(distinct s.customer\_id)::numeric

,2) as avg\_sale\_per\_customers

from sales as s

join customers as c

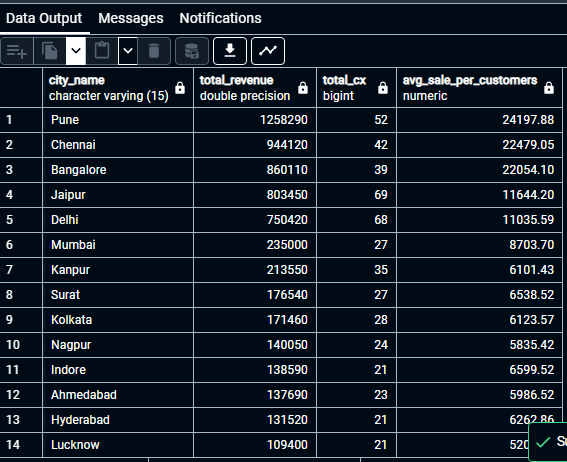
on s.customer\_id = c.customer\_id

join city as ci

on ci.city\_id = c.city\_id

group by 1

order by 2 desc



## Q5-- city population and coffee consumers

## --Provide a list of cities along with their populations and estimated coffee consumers

## -- re turn city name, total current cx, estimated coffee consumers (25%)

WITH

cit\_table AS (

SELECT

ci.city\_name,

ROUND((ci.population \* 0.25) / 1000000, 2) AS coffee\_consumers

FROM city AS ci

),

customers\_table AS (

SELECT

ci.city\_name,

COUNT(DISTINCT cu.customer\_id) AS unique\_cx

FROM sales AS s

JOIN customers AS cu ON cu.customer\_id = s.customer\_id

JOIN city AS ci ON cu.city\_id = ci.city\_id

GROUP BY ci.city\_name

)

SELECT

ct.city\_name,

ct.coffee\_consumers as coffee\_consumers\_in\_million,

cut.unique\_cx

FROM

cit\_table AS ct

JOIN

customers\_table AS cut ON ct.city\_name = cut.city\_name;

A screenshot of a computer

Description automatically generated

## --Q6 top selling products by city

## --what are the top 3 selling products in each cit based on sales volume?

SELECT \*

FROM (

SELECT

ci.city\_name,

p.product\_name,

COUNT(s.sale\_id) AS total\_orders,

DENSE\_RANK() OVER (PARTITION BY ci.city\_name ORDER BY COUNT(s.sale\_id) DESC) AS rank

FROM products AS p

JOIN sales AS s ON p.product\_id = s.product\_id

JOIN customers AS c ON c.customer\_id = s.customer\_id

JOIN city AS ci ON ci.city\_id = c.city\_id

GROUP BY ci.city\_name, p.product\_name

) AS t1

WHERE rank <= 3;

A screenshot of a computer

Description automatically generated

## --Q7 How many unique customers are there in each city who have purchased coffee products?

select

ci.city\_name,

count(distinct c.customer\_id) as unique\_cx

from city as ci

left join

customers as c

on c.city\_id = ci.city\_id

join sales as s

on s.customer\_id = c.customer\_id

where

s.product\_id in (1,2,3,4,5,6,7,8,9,10,11,12,13,14)

group by 1

A screenshot of a computer

Description automatically generated

## --Q8 find each city and their average sale per customer and avg rent per customer

with

city\_table

as

(

select

ci.city\_name,

sum(s.total) as total\_revenue,

count( distinct s.customer\_id) as total\_cx,

round(

sum(s.total)::numeric /count(distinct s.customer\_id)::numeric

,2) as avg\_sale\_per\_customer

from sales as s

join customers as c

on s.customer\_id = c.customer\_id

join city as ci

on ci.city\_id = c.city\_id

group by 1

order by 2 desc

),

city\_rent

as

(

select city\_name, estimated\_rent

from city)

select

cr.city\_name,

cr.estimated\_rent,

ct.total\_cx,

ct.avg\_sale\_per\_customer,

round(cr.estimated\_rent::numeric/total\_cx::numeric,2) as avg\_rent

from city\_rent as cr

join city\_table as ct

on cr.city\_name = ct.city\_name

A screenshot of a computer screen

Description automatically generated

## --Q9 sales growth rate: calculate the percentage growth or decline in sales over different time periods (monthly)

-- by each city

with

monthly\_sales

as

(

select ci.city\_name,

extract(month from s.sale\_date) as months,

extract (year from s.sale\_date) as years,

sum(s.total) as total\_sale

from sales as s

join customers as c

on c.customer\_id = s.customer\_id

join city as ci

on ci.city\_id = c.city\_id

group by 1,2,3

order by 1,3, 2

),

growth\_ratio

as

(

select city\_name,

months,

years,

total\_sale as current\_month\_sale,

lag(total\_sale ,1) over(partition by city\_name order by years, months) as last\_month\_sale

from monthly\_sales

)

select city\_name,

months,

years,

current\_month\_sale,

last\_month\_sale,

round((current\_month\_sale-last\_month\_sale)::numeric/last\_month\_sale::numeric \*100, 2)

as growth\_rattio

from growth\_ratio

A screenshot of a computer screen

Description automatically generated

## --Q10 identify top 3 city based on highest sales, return city name, total sale, totran rent, total customers, estimatedcoffee consumers

with

city\_table

as

(

select

ci.city\_name,

sum(s.total) as total\_revenue,

count( distinct s.customer\_id) as total\_cx,

round(

sum(s.total)::numeric /count(distinct s.customer\_id)::numeric

,2) as avg\_sale\_per\_customer

from sales as s

join customers as c

on s.customer\_id = c.customer\_id

join city as ci

on ci.city\_id = c.city\_id

group by 1

order by 2 desc

),

city\_rent

as

(

select city\_name, estimated\_rent,

population \*0.25 as estimated\_coffee\_consumers

from city)

select

cr.city\_name,

total\_revenue,

cr.estimated\_rent as total\_rent,

ct.total\_cx,

estimated\_coffee\_consumers,

ct.avg\_sale\_per\_customer,

round(cr.estimated\_rent::numeric/total\_cx::numeric,2) as avg\_rent

from city\_rent as cr

join city\_table as ct

on cr.city\_name = ct.city\_name

A screenshot of a computer

Description automatically generated