



---

# PPT on Monday Coffee Expansion Project

---

By : Harsh Sharma



# Objective

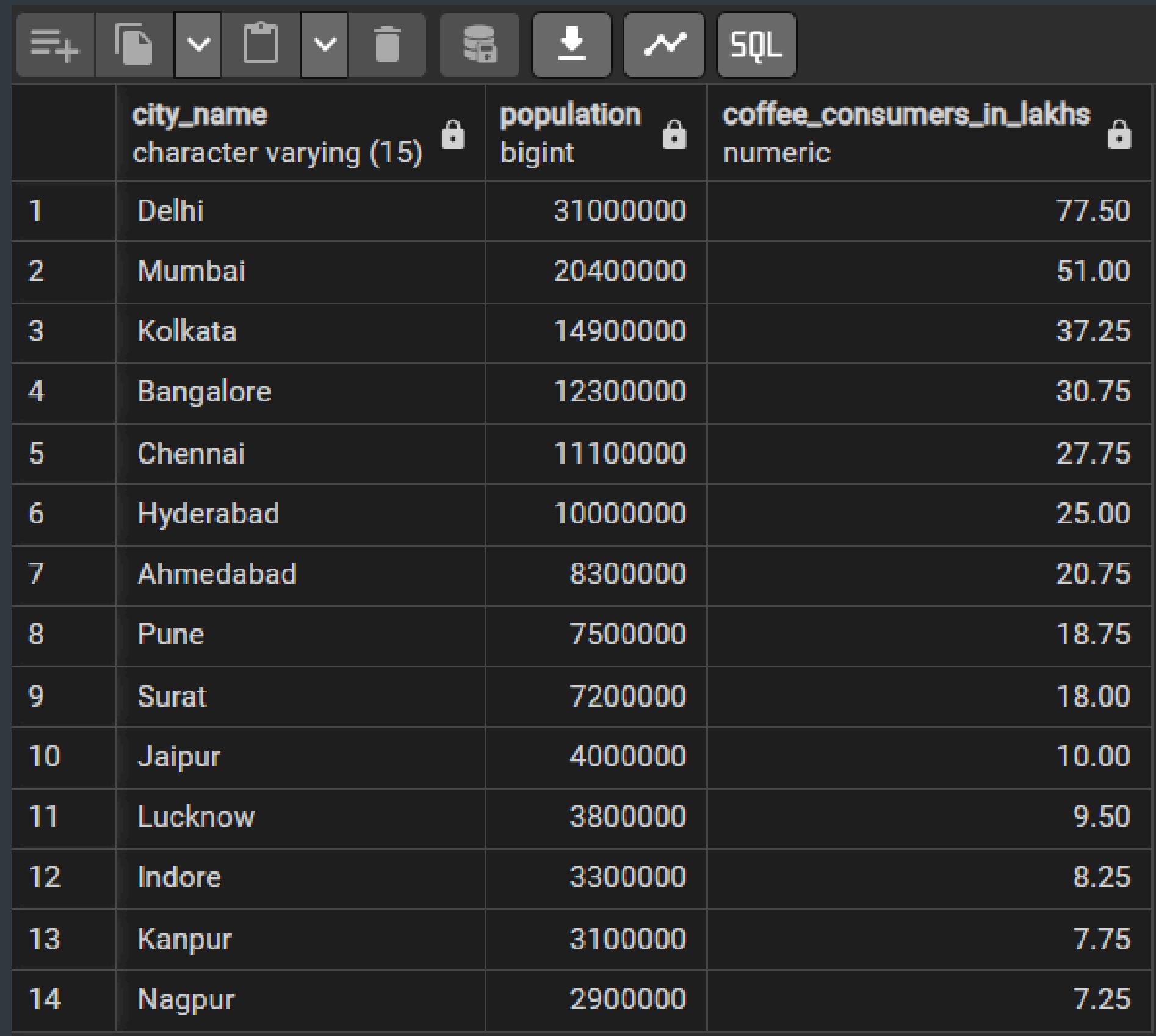
The goal of this project is to analyze the sales data of Monday Coffee, a company that has been selling its products online since January 2023, and to recommend the top three major cities in India for opening new coffee shop locations based on consumer demand and sales performance.

## QUESTION 1 :

How many people in each city are estimated to consume coffee, given that 25% of the population does?

```
SELECT  
CITY.CITY_NAME,  
CITY.POPULATION ,  
ROUND(CITY.POPULATION * 0.25 / 100000,2) AS COFFEE_CONSUMERS_IN_LAKHS  
FROM SALES  
JOIN  
CUSTOMERS  
ON CUSTOMERS.CUSTOMER_ID = SALES.CUSTOMER_ID  
JOIN CITY  
ON CITY.CITY_ID = CUSTOMERS.CITY_ID  
GROUP BY 1,2  
ORDER BY COFFEE_CONSUMERS_IN_LAKHS DESC
```

# SOLUTION : 1



The screenshot shows a database interface with a toolbar at the top containing various icons for file operations and SQL execution. Below the toolbar is a table with 14 rows of data. The table has three columns: 'city\_name' (character varying (15)), 'population' (bigint), and 'coffee\_consumers\_in\_lakhs' (numeric). The data represents 14 Indian cities and their respective populations and coffee consumer counts.

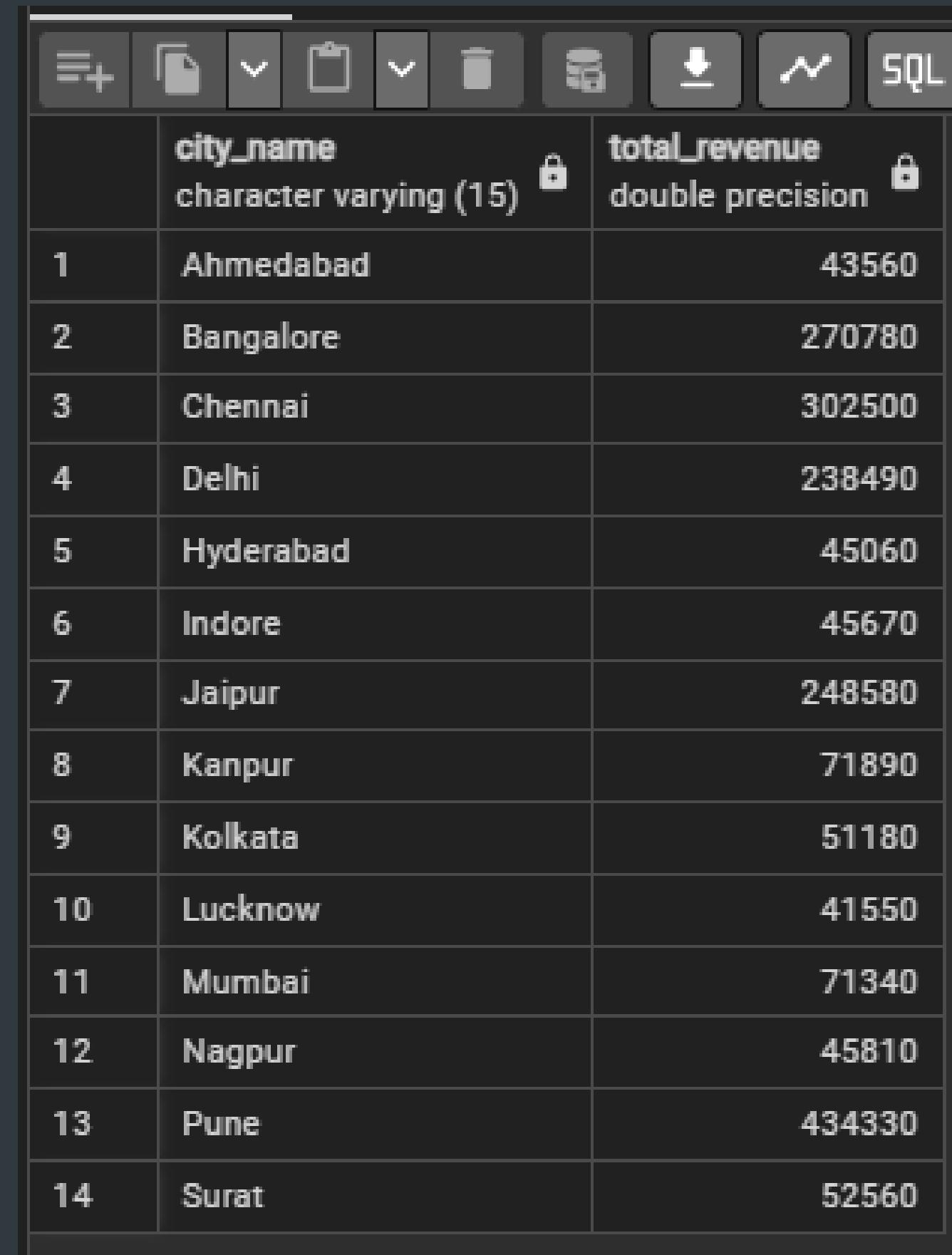
	city_name	population	coffee_consumers_in_lakhs
1	Delhi	31000000	77.50
2	Mumbai	20400000	51.00
3	Kolkata	14900000	37.25
4	Bangalore	12300000	30.75
5	Chennai	11100000	27.75
6	Hyderabad	10000000	25.00
7	Ahmedabad	8300000	20.75
8	Pune	7500000	18.75
9	Surat	7200000	18.00
10	Jaipur	4000000	10.00
11	Lucknow	3800000	9.50
12	Indore	3300000	8.25
13	Kanpur	3100000	7.75
14	Nagpur	2900000	7.25

## QUESTION 2 :

What is the total revenue generated from coffee sales across all cities in the last quarter of 2023?

```
with ct
as
(
    SELECT CITY.CITY_NAME,
    SUM(SALES.TOTAL) AS TOTAL_REVENUE,
    EXTRACT (QUARTER FROM SALES.SALE_DATE) AS QUARTER_NUMBER,
    EXTRACT (YEAR FROM SALES.SALE_DATE) AS YEAR
    FROM SALES
    JOIN CUSTOMERS
    ON CUSTOMERS.CUSTOMER_ID = SALES.CUSTOMER_ID
    JOIN CITY
    ON CITY.CITY_ID = CUSTOMERS.CITY_ID
    GROUP BY 1 , 3 ,4
    ORDER BY 4 , 3
)
select
city_name,
TOTAL_REVENUE
from ct
where QUARTER_NUMBER = 4
AND YEAR = 2023
```

## SOLUTION : 2



The screenshot shows a database interface with a toolbar at the top containing various icons for file operations and SQL queries. Below the toolbar is a table with two columns: 'city\_name' and 'total\_revenue'. The table lists 14 cities with their respective total revenues.

	city_name character varying (15)	total_revenue double precision
1	Ahmedabad	43560
2	Bangalore	270780
3	Chennai	302500
4	Delhi	238490
5	Hyderabad	45060
6	Indore	45670
7	Jaipur	248580
8	Kanpur	71890
9	Kolkata	51180
10	Lucknow	41550
11	Mumbai	71340
12	Nagpur	45810
13	Pune	434330
14	Surat	52560

### QUESTION 3 :

How many units of each coffee product have been sold:

```
SELECT PRODUCTS.PRODUCT_NAME ,  
COUNT(SALES.SALE_ID) AS SALES_COUNT  
FROM SALES  
JOIN  
PRODUCTS  
ON SALES.PRODUCT_ID = PRODUCTS.PRODUCT_ID  
GROUP BY 1  
ORDER BY 2 DESC
```

	product_name	sales_count
1	Cold Brew Coffee Pack (6 Bottles)	1326
2	Ground Espresso Coffee (250g)	1271
3	Instant Coffee Powder (100g)	1226
4	Coffee Beans (500g)	1218
5	Tote Bag with Coffee Design	776
6	Vanilla Coffee Syrup (250ml)	762
7	Cold Brew Concentrate (500ml)	312
8	Organic Green Coffee Beans (500...	307
9	Coffee Art Print	296
10	Flavored Coffee Pods (Pack of 10)	295
11	Coffee Drip Bags (10 Bags)	289
12	Insulated Travel Mug	273
13	Coffee Gift Hamper	270
14	Specialty Coffee Subscription	258
15	Customizable Coffee Coaster Set	258
16	French Press Coffee Set	257
17	Caramel Syrup (250ml)	96
18	Coffee Plant Kit (DIY)	91
19	Coffee Bean Storage Canister	89
20	Coffee Recipe Book	88
21	Mocha Flavored Coffee Mix (200g)	86
22	Personalized Coffee Spoon	83
23	Coffee-Themed T-Shirt	82
24	Reusable Coffee Cup (Eco-friendly)	78
25	Glass Coffee Jar (500ml)	77
26	Coffee-Themed Notebook	76
27	Stainless Steel Tumbler	75
28	Coffee Mug (Ceramic)	73

## QUESTION 4 :

### Average Sales Amount per City

```
|  
|  
|  
|  
SELECT CITY.CITY_NAME,  
ROUND(AVG(SALES.TOTAL) ::NUMERIC,2) AS AVG_SALES_AMOUNT_PER_CITY  
FROM SALES  
LEFT JOIN CUSTOMERS  
ON CUSTOMERS.CUSTOMER_ID =SALES.CUSTOMER_ID  
RIGHT JOIN CITY  
ON CITY.CITY_ID = CUSTOMERS.CITY_ID  
GROUP BY 1  
ORDER BY 2 DESC
```

## SOLUTION : 4



The screenshot shows a database interface with a toolbar at the top containing various icons for file operations, search, and navigation. Below the toolbar is a table with two columns: 'city\_name' and 'avg\_sales\_amount\_per\_city'. The table lists 14 cities and their corresponding average sales amounts.

	city_name character varying (15)	avg_sales_amount_per_city numeric
1	Ahmedabad	611.96
2	Indore	599.96
3	Kolkata	599.51
4	Hyderabad	595.11
5	Surat	594.41
6	Chennai	589.71
7	Pune	589.36
8	Bangalore	587.51
9	Jaipur	583.48
10	Delhi	575.03
11	Kanpur	574.06
12	Mumbai	555.56
13	Nagpur	553.56
14	Lucknow	552.53

## QUESTION 5 :

What is the average sales amount per customer in each city?

```
SELECT CITY.CITY_NAME,  
ROUND(  
    SUM(SALES.TOTAL) ::NUMERIC /  
        COUNT(DISTINCT CUSTOMERS.CUSTOMER_ID) ::NUMERIC ,2  
    ) AS AVG_SALES_AMOUNT_PER_CUSTOMER  
FROM SALES  
JOIN CUSTOMERS  
ON SALES.CUSTOMER_ID = CUSTOMERS.CUSTOMER_ID  
JOIN  
CITY  
ON CITY.CITY_ID = CUSTOMERS.CITY_ID  
GROUP BY 1  
ORDER BY 2 DESC
```

## SOLUTION : 5



The screenshot shows a database interface with a toolbar at the top containing various icons for file operations, search, and navigation. Below the toolbar is a table with two columns: 'city\_name' and 'avg\_sales\_amount\_per\_customer'. The table lists 14 cities with their respective average sales amounts.

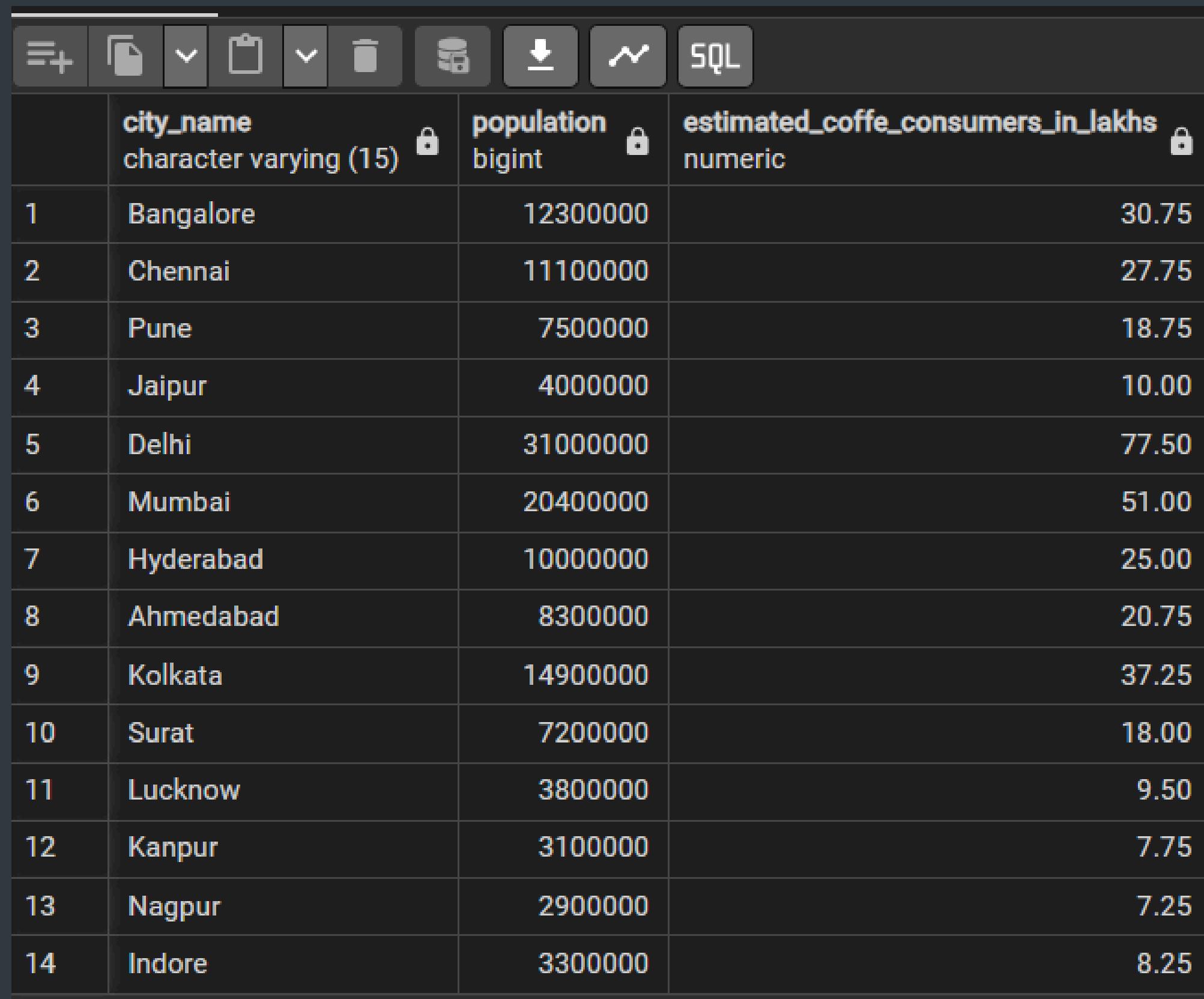
	city_name character varying (15)	avg_sales_amount_per_customer numeric
1	Pune	24197.88
2	Chennai	22479.05
3	Bangalore	22054.10
4	Jaipur	11644.20
5	Delhi	11035.59
6	Mumbai	8703.70
7	Indore	6599.52
8	Surat	6538.52
9	Hyderabad	6262.86
10	Kolkata	6123.57
11	Kanpur	6101.43
12	Ahmedabad	5986.52
13	Nagpur	5835.42
14	Lucknow	5209.52

-- QUESTION 6 :

Provide a list of cities along with their populations and estimated coffee consumers  
{ 25% OF POPULATION }

```
SELECT
CITY_NAME ,
POPULATION,
ROUND( (POPULATION * 0.25 / 100000),2) AS ESTIMATED_COFFE_CONSUMERS_IN_LAKHS
FROM
CITY
```

## SOLUTION : 6



The screenshot shows a database interface with a toolbar at the top containing various icons for file operations, search, and navigation. Below the toolbar is a table with 14 rows of data. The table has four columns: a primary key column (labeled 1 through 14), a city name column, a population column, and an estimated coffee consumer count column.

	city_name	population	estimated_coffee_consumers_in_Jakhs
1	Bangalore	12300000	30.75
2	Chennai	11100000	27.75
3	Pune	7500000	18.75
4	Jaipur	4000000	10.00
5	Delhi	31000000	77.50
6	Mumbai	20400000	51.00
7	Hyderabad	10000000	25.00
8	Ahmedabad	8300000	20.75
9	Kolkata	14900000	37.25
10	Surat	7200000	18.00
11	Lucknow	3800000	9.50
12	Kanpur	3100000	7.75
13	Nagpur	2900000	7.25
14	Indore	3300000	8.25

--QUESTION 7 :

What are the top 3 selling products in each city based on sales volume?

```
WITH CT
AS
(
SELECT
CITY.CITY_NAME,
PRODUCTS.PRODUCT_NAME,
COUNT(PRODUCTS.PRODUCT_ID) AS PRODUCT_COUNT,
ROW_NUMBER() OVER (PARTITION BY CITY.CITY_NAME ORDER BY COUNT(PRODUCTS.PRODUCT_ID)DESC) AS RANK
FROM
SALES
JOIN CUSTOMERS
ON SALES.CUSTOMER_ID = CUSTOMERS.CUSTOMER_ID
JOIN CITY
ON CITY.CITY_ID = CUSTOMERS.CITY_ID
JOIN PRODUCTS
ON
PRODUCTS.PRODUCT_ID = SALES.PRODUCT_ID
GROUP BY 1,2
)
SELECT *
FROM CT
WHERE RANK < 4
```

## SOLUTION : 7

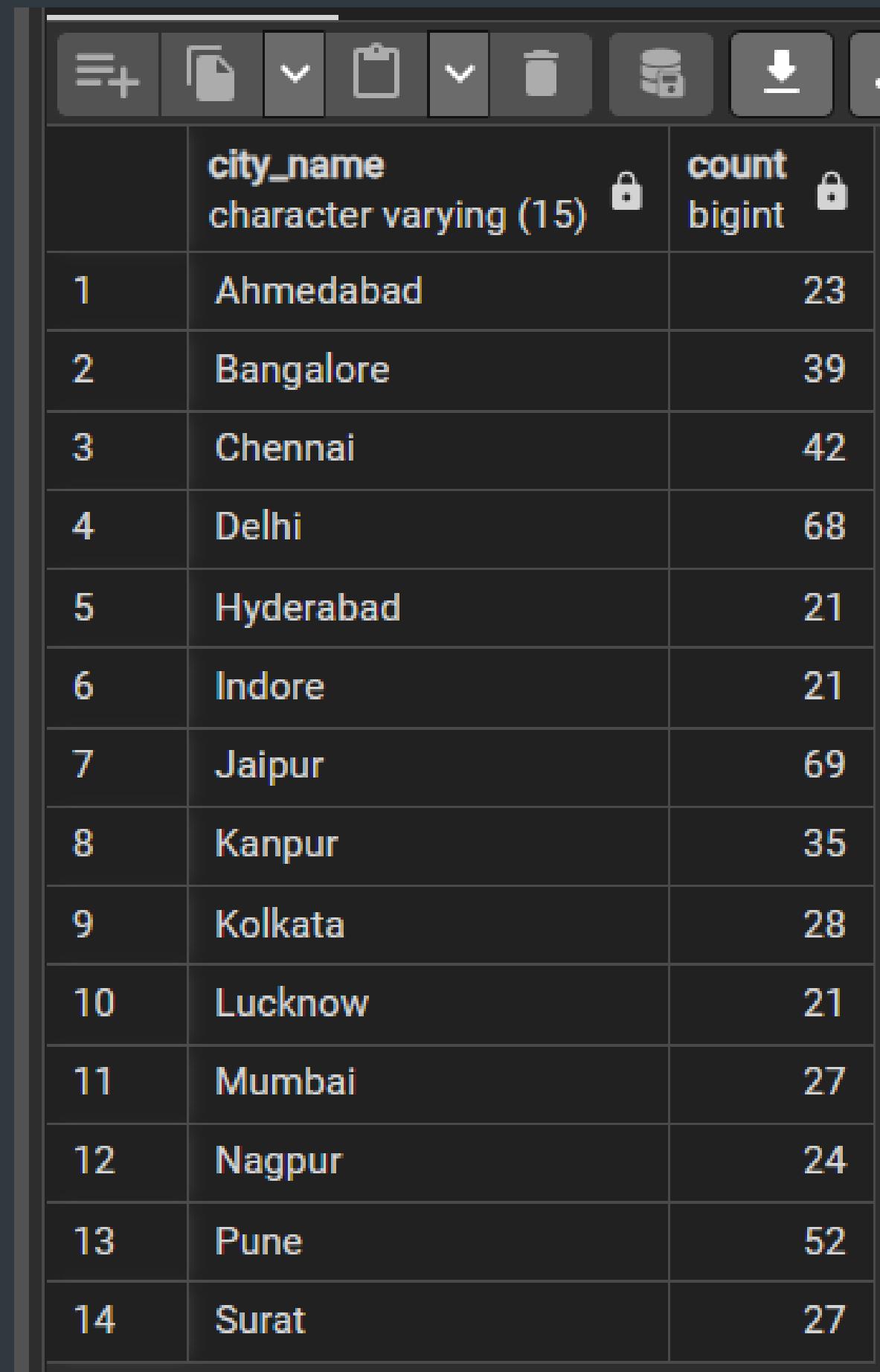
	city_name character varying (15)	product_name character varying (35)	product_count bigint	rank bigint
1	Ahmedabad	Cold Brew Coffee Pack (6 Bottles)	40	1
2	Ahmedabad	Coffee Beans (500g)	35	2
3	Ahmedabad	Instant Coffee Powder (100g)	26	3
4	Bangalore	Cold Brew Coffee Pack (6 Bottles)	197	1
5	Bangalore	Ground Espresso Coffee (250g)	167	2
6	Bangalore	Instant Coffee Powder (100g)	150	3
7	Chennai	Cold Brew Coffee Pack (6 Bottles)	192	1
8	Chennai	Coffee Beans (500g)	181	2
9	Chennai	Instant Coffee Powder (100g)	172	3
10	Delhi	Ground Espresso Coffee (250g)	183	1
11	Delhi	Instant Coffee Powder (100g)	170	2
12	Delhi	Coffee Beans (500g)	161	3
13	Hyderabad	Instant Coffee Powder (100g)	36	1
14	Hyderabad	Cold Brew Coffee Pack (6 Bottles)	28	2
15	Hyderabad	Ground Espresso Coffee (250g)	27	3
16	Indore	Instant Coffee Powder (100g)	33	1
17	Indore	Cold Brew Coffee Pack (6 Bottles)	26	2
18	Indore	Ground Espresso Coffee (250g)	26	3
19	Jaipur	Cold Brew Coffee Pack (6 Bottles)	178	1
20	Jaipur	Coffee Beans (500g)	175	2
21	Jaipur	Instant Coffee Powder (100g)	170	3
22	Kanpur	Cold Brew Coffee Pack (6 Bottles)	57	1
23	Kanpur	Ground Espresso Coffee (250g)	55	2
24	Kanpur	Coffee Beans (500g)	50	3
25	Kolkata	Ground Espresso Coffee (250g)	45	1
26	Kolkata	Cold Brew Coffee Pack (6 Bottles)	44	2
27	Kolkata	Coffee Beans (500g)	38	3
28	Lucknow	Instant Coffee Powder (100g)	28	1
29	Lucknow	Coffee Beans (500g)	25	2
30	Lucknow	Cold Brew Coffee Pack (6 Bottles)	23	3
31	Mumbai	Ground Espresso Coffee (250g)	62	1
32	Mumbai	Instant Coffee Powder (100g)	60	2
33	Mumbai	Cold Brew Coffee Pack (6 Bottles)	53	3
34	Nagpur	Ground Espresso Coffee (250g)	39	1
35	Nagpur	Instant Coffee Powder (100g)	29	2
36	Nagpur	Coffee Beans (500g)	28	3
37	Pune	Cold Brew Coffee Pack (6 Bottles)	259	1
38	Pune	Ground Espresso Coffee (250g)	254	2
39	Pune	Instant Coffee Powder (100g)	245	3
40	Surat	Coffee Beans (500g)	48	1
41	Surat	Cold Brew Coffee Pack (6 Bottles)	45	2
42	Surat	Ground Espresso Coffee (250g)	41	3

--QUESTION 8:

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

```
SELECT CITY.CITY_NAME , COUNT(DISTINCT CUSTOMERS.CUSTOMER_ID)
FROM SALES
JOIN CUSTOMERS
ON SALES.CUSTOMER_ID = CUSTOMERS.CUSTOMER_ID
JOIN PRODUCTS
ON PRODUCTS.PRODUCT_ID = SALES.PRODUCT_ID
JOIN CITY
ON CITY.CITY_ID = CUSTOMERS.CITY_ID
WHERE PRODUCTS.PRODUCT_ID < 15 GROUP BY 1
```

## SOLUTION : 8



A screenshot of a PostgreSQL database table titled 'SOLUTION : 8'. The table has two columns: 'city\_name' (character varying (15)) and 'count' (bigint). The data is sorted by count in descending order. The table shows 14 rows of data.

	city_name	count
1	Ahmedabad	23
2	Bangalore	39
3	Chennai	42
4	Delhi	68
5	Hyderabad	21
6	Indore	21
7	Jaipur	69
8	Kanpur	35
9	Kolkata	28
10	Lucknow	21
11	Mumbai	27
12	Nagpur	24
13	Pune	52
14	Surat	27

-- QUESTION 9 :

Find each city and their average sale per customer and avg rent per customer

```
SELECT
CITY.CITY_NAME ,
CITY.ESTIMATED_RENT ,
ROUND(
    SUM(SALES.TOTAL) ::NUMERIC / COUNT(DISTINCT SALES.CUSTOMER_ID) ::NUMERIC ,2
) AS AVG_SALES_AMOUNT_PER_CUSTOMER,
ROUND(
    CITY.ESTIMATED_RENT ::NUMERIC / COUNT(DISTINCT SALES.CUSTOMER_ID)::NUMERIC,2
) AS AVG_ESTIMATED_RENT
FROM SALES
JOIN
CUSTOMERS
ON SALES.CUSTOMER_ID = CUSTOMERS.CUSTOMER_ID
JOIN
PRODUCTS
ON PRODUCTS.PRODUCT_ID = SALES.PRODUCT_ID
JOIN
CITY
ON CITY.CITY_ID = CUSTOMERS.CITY_ID
GROUP BY 1,2
```

## SOLUTION : 9

The screenshot shows a database interface with a toolbar at the top containing various icons for file operations and SQL. Below the toolbar is a table with four columns and 14 rows of data. The columns are labeled: city\_name, estimated\_rent, avg\_sales\_amount\_per\_customer, and avg\_estimated\_rent. The data rows are as follows:

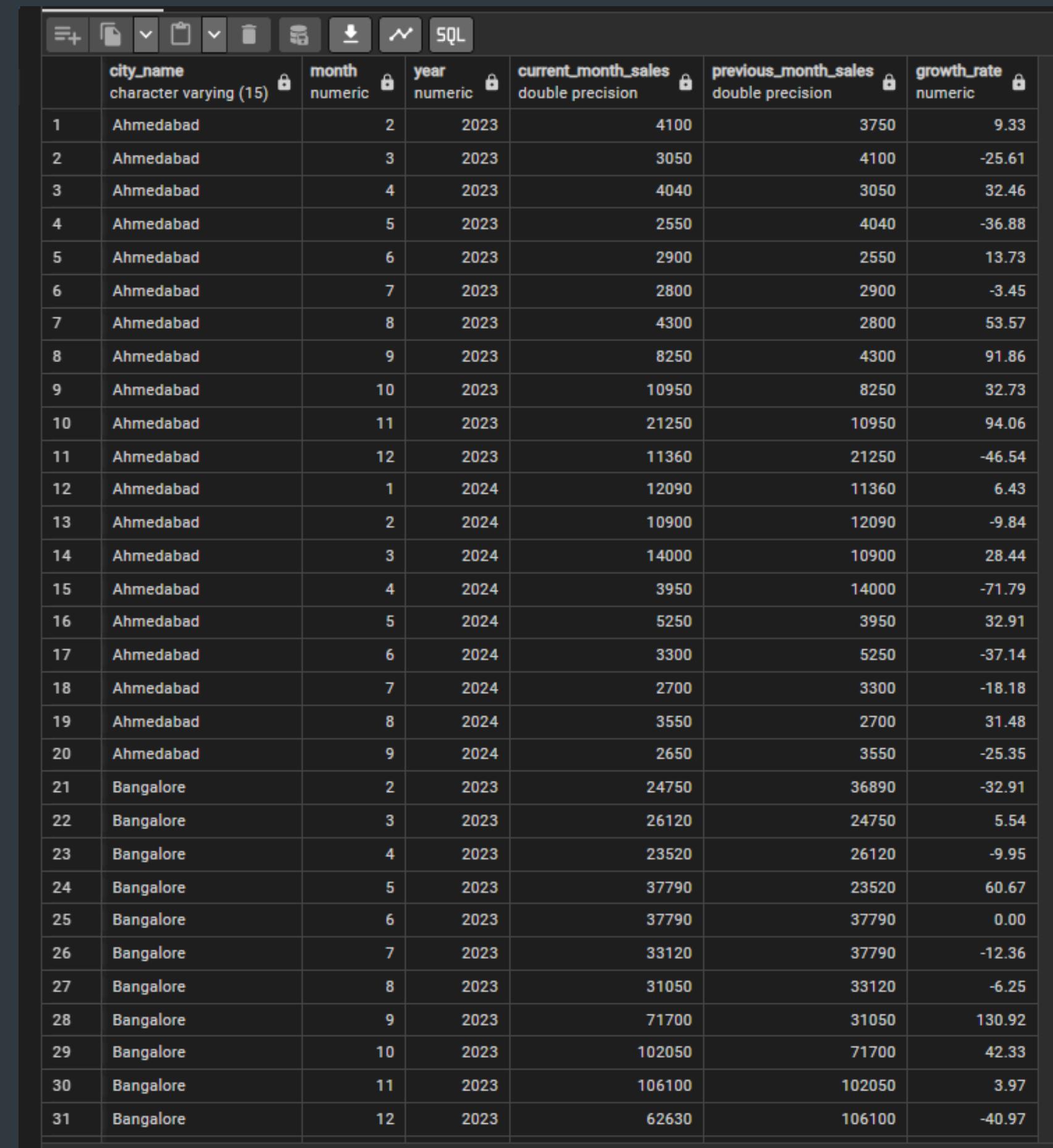
	city_name	estimated_rent	avg_sales_amount_per_customer	avg_estimated_rent
1	Ahmedabad	14400	5986.52	626.09
2	Bangalore	29700	22054.10	761.54
3	Chennai	17100	22479.05	407.14
4	Delhi	22500	11035.59	330.88
5	Hyderabad	22500	6262.86	1071.43
6	Indore	6300	6599.52	300.00
7	Jaipur	10800	11644.20	156.52
8	Kanpur	8100	6101.43	231.43
9	Kolkata	16200	6123.57	578.57
10	Lucknow	9000	5209.52	428.57
11	Mumbai	31500	8703.70	1166.67
12	Nagpur	7200	5835.42	300.00
13	Pune	15300	24197.88	294.23
14	Surat	13500	6538.52	500.00

--QUESTION 10 :

Calculate the percentage growth (or decline) in sales over different time periods (monthly) BY EACH CITY

```
:
WITH LAST_MONTH_SALES
AS
(
SELECT
CITY.CITY_NAME ,
EXTRACT(MONTH FROM SALES.SALE_DATE) AS MONTH,
EXTRACT(YEAR FROM SALES.SALE_DATE) AS YEAR,
SUM(SALES.TOTAL) AS CURRENT_MONTH_SALES
FROM
SALES
JOIN CUSTOMERS
ON SALES.CUSTOMER_ID = CUSTOMERS.CUSTOMER_ID
JOIN
PRODUCTS
ON PRODUCTS.PRODUCT_ID = SALES.PRODUCT_ID
JOIN
CITY
ON CITY.CITY_ID = CUSTOMERS.CITY_ID
GROUP BY 1,2 ,3
ORDER BY 1,3 ,2
),
CT AS
(
SELECT
CITY_NAME,
MONTH,
YEAR,
CURRENT_MONTH_SALES,
LAG(CURRENT_MONTH_SALES,1) OVER (PARTITION BY CITY_NAME) AS PREVIOUS_MONTH_SALES
FROM LAST_MONTH_SALES
)
SELECT
CITY_NAME,
MONTH,
YEAR,
CURRENT_MONTH_SALES,
PREVIOUS_MONTH_SALES ,
ROUND((CURRENT_MONTH_SALES - PREVIOUS_MONTH_SALES) ::NUMERIC / PREVIOUS_MONTH_SALES ::NUMERIC*100,2) AS GROWTH_RATE
FROM CT
WHERE PREVIOUS_MONTH_SALES IS NOT NULL
:
```

## SOLUTION : 10



The screenshot shows a database interface with a toolbar at the top containing various icons. The main area displays a table with the following columns:

	city_name character varying (15)	month numeric	year numeric	current_month_sales double precision	previous_month_sales double precision	growth_rate numeric
1	Ahmedabad	2	2023	4100	3750	9.33
2	Ahmedabad	3	2023	3050	4100	-25.61
3	Ahmedabad	4	2023	4040	3050	32.46
4	Ahmedabad	5	2023	2550	4040	-36.88
5	Ahmedabad	6	2023	2900	2550	13.73
6	Ahmedabad	7	2023	2800	2900	-3.45
7	Ahmedabad	8	2023	4300	2800	53.57
8	Ahmedabad	9	2023	8250	4300	91.86
9	Ahmedabad	10	2023	10950	8250	32.73
10	Ahmedabad	11	2023	21250	10950	94.06
11	Ahmedabad	12	2023	11360	21250	-46.54
12	Ahmedabad	1	2024	12090	11360	6.43
13	Ahmedabad	2	2024	10900	12090	-9.84
14	Ahmedabad	3	2024	14000	10900	28.44
15	Ahmedabad	4	2024	3950	14000	-71.79
16	Ahmedabad	5	2024	5250	3950	32.91
17	Ahmedabad	6	2024	3300	5250	-37.14
18	Ahmedabad	7	2024	2700	3300	-18.18
19	Ahmedabad	8	2024	3550	2700	31.48
20	Ahmedabad	9	2024	2650	3550	-25.35
21	Bangalore	2	2023	24750	36890	-32.91
22	Bangalore	3	2023	26120	24750	5.54
23	Bangalore	4	2023	23520	26120	-9.95
24	Bangalore	5	2023	37790	23520	60.67
25	Bangalore	6	2023	37790	37790	0.00
26	Bangalore	7	2023	33120	37790	-12.36
27	Bangalore	8	2023	31050	33120	-6.25
28	Bangalore	9	2023	71700	31050	130.92
29	Bangalore	10	2023	102050	71700	42.33
30	Bangalore	11	2023	106100	102050	3.97
31	Bangalore	12	2023	62630	106100	-40.97

It contain total 285 rows  
that can't be shown here\*

--QUESTION 11 :

Identify top 3 city based on highest sales, return city name, total sale, total rent, total customers, estimated coffee consumer

```
WITH CT
AS
(
SELECT
    CITY.CITY_NAME ,
    SUM(SALES.TOTAL) AS TOTAL_SALES,
    COUNT(SALES.SALE_ID) AS SALES_COUNT,
    CITY.ESTIMATED_RENT,
    COUNT(DISTINCT SALES.CUSTOMER_ID) AS CUSTOMER_COUNT,
    CITY.POPULATION AS TOTAL_POPULATION,
    ROUND(SUM(SALES.TOTAL) ::NUMERIC/ COUNT(DISTINCT SALES.CUSTOMER_ID)::NUMERIC,2) AS AVG_SALES_PER_CUSTOMER,
    ROUND(CITY.ESTIMATED_RENT ::NUMERIC / COUNT(DISTINCT SALES.CUSTOMER_ID)::NUMERIC,2) AS AVG_RENT_PER_CUSTOMER
FROM SALES
JOIN CUSTOMERS
ON SALES.CUSTOMER_ID = CUSTOMERS.CUSTOMER_ID
JOIN PRODUCTS
ON PRODUCTS.PRODUCT_ID = SALES.PRODUCT_ID
JOIN CITY
ON CITY.CITY_ID = CUSTOMERS.CITY_ID
GROUP BY 1,4,6
)
SELECT
    CITY_NAME,
    TOTAL_SALES,
    SALES_COUNT,
    ESTIMATED_RENT,
    CUSTOMER_COUNT,
    ROUND((TOTAL_POPULATION * 0.25)/1000000,2) AS ESTIMATED_COFFEE_CONSUMERS_IN_MILLIONS,
    AVG_SALES_PER_CUSTOMER,
    AVG_RENT_PER_CUSTOMER
FROM CT
ORDER BY TOTAL_SALES DESC
```

# SOLUTION : 11

	city_name character varying (15)	total_sales double precision	sales_count bigint	estimated_rent double precision	customer_count bigint	estimated_coffee_consumers_in_millions numeric	avg_sales_per_customer numeric	avg_rent_per_customer numeric
1	Pune	1258290	2135	15300	52	1.88	24197.88	294.23
2	Chennai	944120	1601	17100	42	2.78	22479.05	407.14
3	Bangalore	860110	1464	29700	39	3.08	22054.10	761.54
4	Jaipur	803450	1377	10800	69	1.00	11644.20	156.52
5	Delhi	750420	1305	22500	68	7.75	11035.59	330.88
6	Mumbai	235000	423	31500	27	5.10	8703.70	1166.67
7	Kanpur	213550	372	8100	35	0.78	6101.43	231.43
8	Surat	176540	297	13500	27	1.80	6538.52	500.00
9	Kolkata	171460	286	16200	28	3.73	6123.57	578.57
10	Nagpur	140050	253	7200	24	0.73	5835.42	300.00
11	Indore	138590	231	6300	21	0.83	6599.52	300.00
12	Ahmedabad	137690	225	14400	23	2.08	5986.52	626.09
13	Hyderabad	131520	221	22500	21	2.50	6262.86	1071.43
14	Lucknow	109400	198	9000	21	0.95	5209.52	428.57



# Recommendation

City 1: Pune

Average rent per customer is very low.

Highest total revenue.

Average sales per customer is also high.

City 2: Chennai

Estimated coffee consumers at 2.78 million.

Total number of customers are 42 , which is a high number

Average rent per customer is 407.14 (still under 500).

City 3: Jaipur

Highest number of customers, which is 69.

Average rent per customer is very low at 156.

Average sales per customer is better at 11.6k.



---

End of  
Presentation  
Thank You

---

