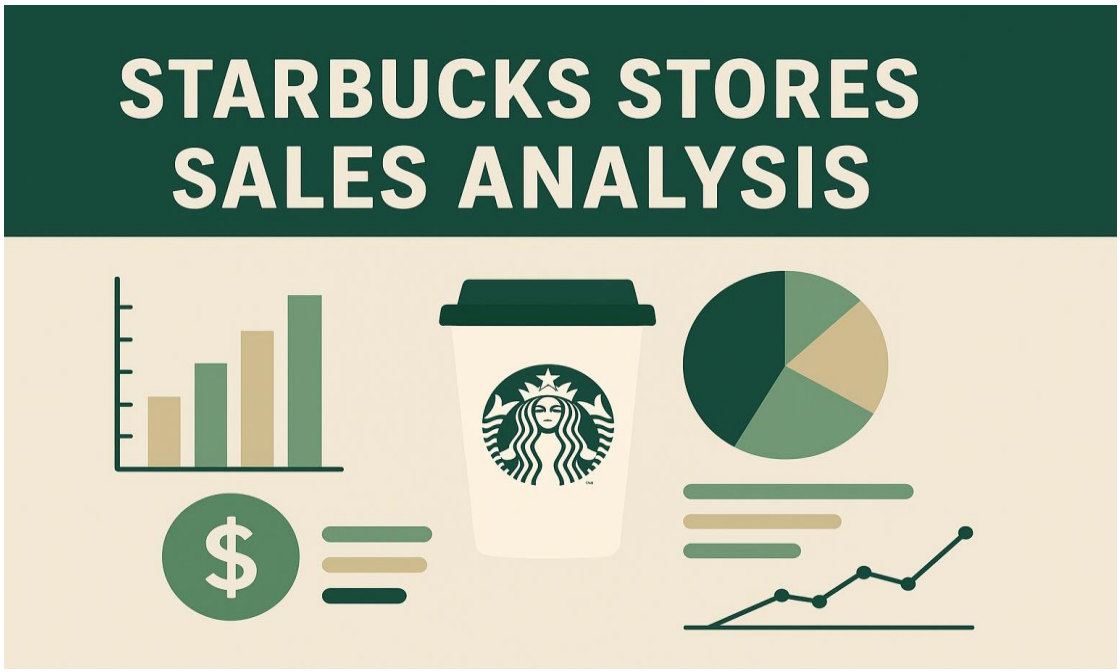
“STARBUCKS SALES REPORT”

STORE SALES ANALYSIS

Empowering Retail Decisions with SQL: Deep Dive into Starbucks Store Sales & Performance

---



OVERVIEW

This project uses SQL-driven analysis (with Power Query as needed) to explore Starbucks store-level sales and operational data. The goal is to transform raw retail data into actionable insights for operations, store managers, regional directors, and the commercial strategy team — enabling better decisions around store performance, staffing, pricing, seasonal planning.

(Power Query + SQL Integrated project)

ASWIN SANTHOSH

PROBLEM STATEMENT

Modern retail chains like Starbucks generate massive volumes of store-level data, yet meaningful insights often remain buried in inconsistent or poorly structured datasets. Essential business decisions—such as evaluating store performance, optimizing manager assignments, improving customer satisfaction, and planning city-level expansion—are frequently made without a clear analytical foundation.

I. What's Going On?

- Starbucks currently has store-level sales data, but it is raw, inconsistent, and difficult to analyse directly.
- Important operational insights such as city performance, manager efficiency, product-demand trends, and store health are not clearly visible.
- Duplicate entries, mismatched city names, and unstructured manager information create barriers to accurate reporting.
- Critical issues such as declining ratings, low-performing managers, or mismanaged store clusters remain hidden inside the unclean dataset.

II. Project Objective

- Clean, normalize, and convert the raw Starbucks dataset into a properly structured data model with dimension and fact tables.
- Apply analytical SQL methods (CTEs, joins, window functions, subqueries) to uncover hidden business patterns.
- Provide answers to essential business questions such as:
    - Which cities generate the highest revenue?
    - Which managers are handling too many low-performing stores?
    - Where are customers high but satisfaction low?
    - Which stores need immediate attention due to declining sales or ratings?
    - Are there store locations that are poorly placed despite good ratings?
- Empower business leaders with data-driven insights for optimizing store performance, improving customer satisfaction, and planning expansion or restructuring.

Data Description

Title: Starbucks sales Dataset

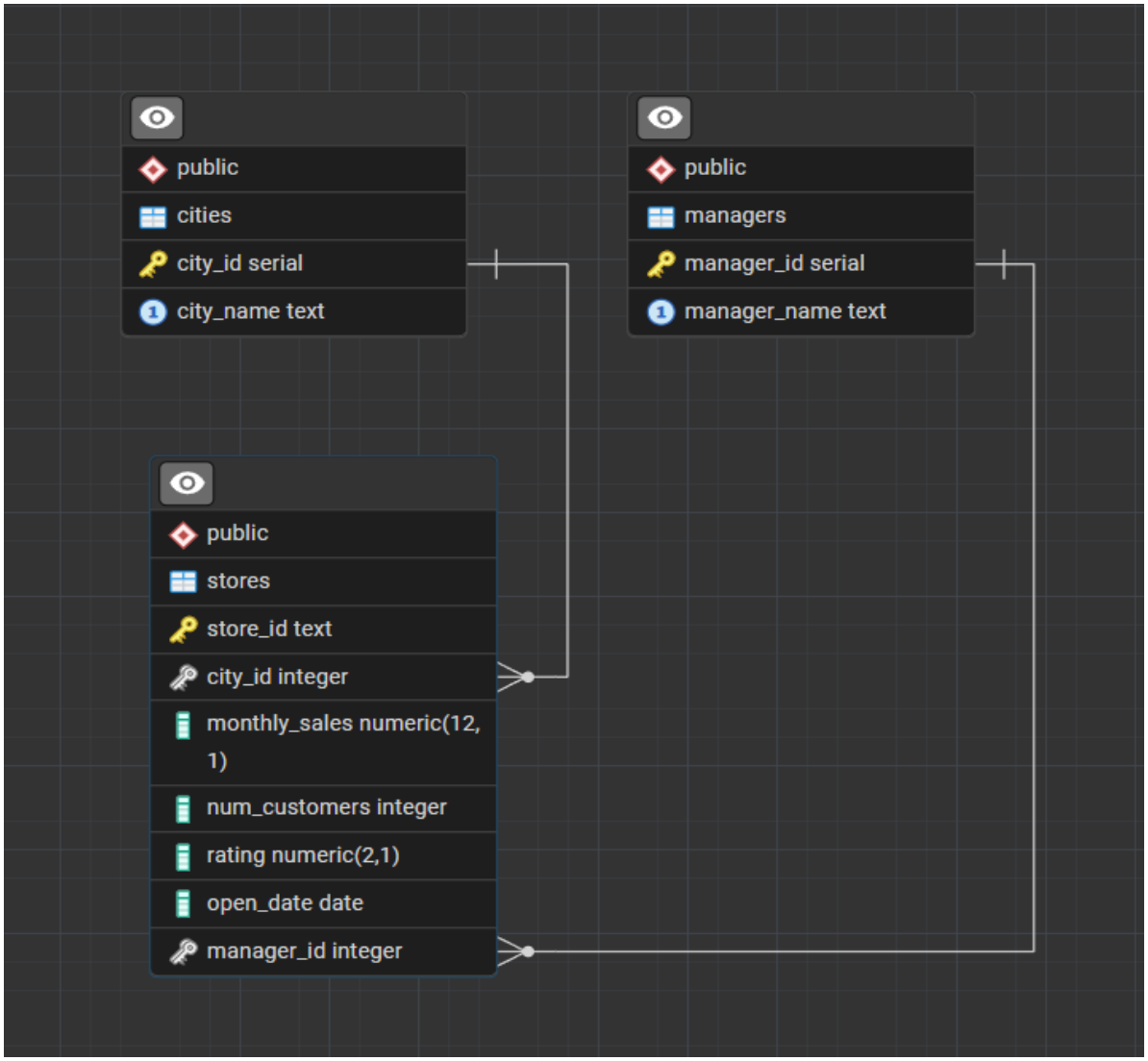
- **Sample Size: ~1498 Rows, 7 Columns**
- **Data Types: Mixed — Numeric, Text, Date**

Key Fields:

- **Store_id, Num_customers**
- **Monthly_sales, City**
- **Rating, Open_date**
- **Manager_name**

Data Cleaning Steps

- Identified and handled missing or null values in key transactional columns such as **Num_customers, Rating**.
- Corrected improper data types (e.g., **Monthly_sales as NUMERIC, Num_customers as INT, Rating as NUMERIC , Open_date as DATE**).
- Replace values (eg. Store_id " " = "").
- Used TRIM and CLEAN to remove empty spaces.
- Normalized the dataset into multiple related SQL tables:
    - **Cities** (unique list of cities and their attributes)
    - **Managers** (manager profiles and assignments)
    - **Stores** (store-level static details)



- The model follows a **1-to-many relationship** pattern : Each dimension table is related to many rows in the fact table. The foreign keys in the fact table reference the primary keys in dimension tables. This structure helps in simplifying complex queries, improving query performance, and ensuring data integrity.

Category 1. Store Performance Issues

1. Underperforming Stores Compared to City Average

```sql
--Underperforming Stores Compared to City Average
WITH city_sales AS (
    SELECT
        city_id,
        ROUND(AVG(monthly_sales),2) AS avg_city_sales
    FROM stores
    GROUP BY city_id
)
SELECT
    s.store_id,
    c.city_name,
    s.monthly_sales,
    cs.avg_city_sales,
    (cs.avg_city_sales - s.monthly_sales) AS difference
FROM stores s
JOIN city_sales cs ON s.city_id = cs.city_id
JOIN cities c ON c.city_id = s.city_id
WHERE s.monthly_sales < cs.avg_city_sales
order by monthly_sales;
```

	store_id text	city_name text	monthly_sales numeric (12,1)	avg_city_sales numeric	difference numeric
1	SB-1791	Phoenix	20155.0	69462.88	49307.88
2	SB-1517	Boston	20162.0	68052.10	47890.10
3	SB-1141	Seattle	20163.0	66533.08	46370.08
4	SB-937	Seattle	20207.0	66533.08	46326.08
5	SB-637	New York	20281.0	69314.70	49033.70
6	SB-1691	Seattle	20282.0	66533.08	46251.08
7	SB-393	Miami	20526.0	71866.98	51340.98
8	SB-1753	Los Angeles	20587.0	69819.31	49232.31
9	SB-627	Phoenix	20619.0	69462.88	48843.88
10	SB-927	Phoenix	20661.0	69462.88	48801.88
11	SB-1636	Boston	20781.0	68052.10	47271.10
12	SB-134	Phoenix	20806.0	69462.88	48656.88

Insight:

- Stores are performing below the average sales levels of their respective cities.
- This indicates weak local marketing strategies or insufficient customer reach.
- Store location may be a major factor, suggesting low visibility or poor accessibility.

2.Older stores might have declining performance

```sql
--Older stores might have declining performance
WITH store_age AS (
    SELECT
        store_id,
        city_id,
        monthly_sales,
        EXTRACT(YEAR FROM AGE(open_date)) AS years_running
    FROM stores
)
SELECT
    sa.store_id,
    c.city_name,
    sa.years_running,
    sa.monthly_sales
FROM store_age sa
JOIN cities c ON sa.city_id = c.city_id
WHERE years_running >
    (SELECT AVG(EXTRACT(YEAR FROM AGE(open_date))) FROM stores)
  AND monthly_sales <
    (SELECT AVG(monthly_sales) FROM stores)
    order by monthly_sales;
```

	store_id text	city_name text	years_running numeric	monthly_sales numeric (12,1)
1	SB-1791	Phoenix	5	20155.0
2	SB-1517	Boston	5	20162.0
3	SB-937	Seattle	5	20207.0
4	SB-637	New York	5	20281.0
5	SB-1691	Seattle	5	20282.0
6	SB-1753	Los Angeles	5	20587.0
7	SB-627	Phoenix	5	20619.0
8	SB-927	Phoenix	5	20661.0
9	SB-1636	Boston	5	20781.0
10	SB-134	Phoenix	5	20806.0
11	SB-531	New York	5	20956.0
12	SB-277	New York	5	21007.0

Insight:

- Stores that opened earlier may show a drop in sales or customer footfall over time.
- Declining performance can indicate aging infrastructure, outdated interiors, or reduced customer appeal.
- These stores may require renovation, marketing boosts, or operational improvements to regain performance.

3.Some stores have very low sales despite high ratings (Poor location)

```sql
--Some stores have very low sales despite high ratings (Poor location)
WITH high_rating_low_sales AS (
    SELECT
        s.store_id,
        s.city_id,
        s.monthly_sales,
        s.rating
    FROM stores s
    WHERE s.rating > (SELECT AVG(rating) FROM stores)
      AND s.monthly_sales < (SELECT AVG(monthly_sales) FROM stores)
)
SELECT
    hr.store_id,
    c.city_name,
    hr.rating,
    hr.monthly_sales
FROM high_rating_low_sales hr
JOIN cities c ON c.city_id = hr.city_id
order by rating desc;
```

	store_id text	city_name text	rating numeric (2,1)	monthly_sales numeric (12,1)
1	SB-545	New York	5.0	51585.0
2	SB-626	Los Angeles	5.0	38309.0
3	SB-290	Miami	5.0	25084.0
4	SB-1625	Seattle	5.0	29283.0
5	SB-292	Chicago	5.0	33507.0
6	SB-1949	Houston	5.0	48803.0
7	SB-165	Los Angeles	5.0	38297.0
8	SB-1223	New York	5.0	67241.0
9	SB-891	Boston	5.0	37625.0
10	SB-753	Los Angeles	5.0	51348.0
11	SB-447	Chicago	5.0	65004.0
12	SB-1101	Chicago	5.0	32411.0
13	SB-1715	Seattle	5.0	62870.0
14	SB-11	Chicago	5.0	21750.0
15	SB-693	Phoenix	4.9	50659.0

Insight:

- High ratings but low sales suggest the store is in a poor or low-traffic location.
- Strong customer satisfaction indicates service quality is good, but footfall is limited.
- These stores may benefit from relocation, better visibility, or targeted local marketing to boost sales.

4.Detect Seasonal Stores (Open Date Effect on Sales)

```sql
--Detect Seasonal Stores (Open Date Effect on Sales)
SELECT
    store_id,c.city_name,
    TO_CHAR(open_date, 'Month') AS open_month,
    monthly_sales,
    (SELECT ROUND(AVG(monthly_sales),2) FROM stores) AS overall_avg,
    monthly_sales - (SELECT ROUND(AVG(monthly_sales),2) FROM stores) AS difference
FROM stores s
join cities c
on s.city_id = c.city_id
order by open_month;
```

	store_id text	city_name text	open_month text	monthly_sales numeric (12,1)	overall_avg numeric	difference numeric
1	SB-1519	Houston	April	39675.0	69790.50	-30115.50
2	SB-580	Los Angeles	April	114588.0	69790.50	44797.50
3	SB-989	Los Angeles	April	24437.0	69790.50	-45353.50
4	SB-1901	Houston	April	117136.0	69790.50	47345.50
5	SB-1125	Seattle	April	26548.0	69790.50	-43242.50
6	SB-1659	New York	April	99025.0	69790.50	29234.50
7	SB-851	Boston	April	101571.0	69790.50	31780.50
8	SB-1900	Los Angeles	April	94813.0	69790.50	25022.50
9	SB-163	Miami	April	96898.0	69790.50	27107.50
10	SB-1899	Miami	April	55292.0	69790.50	-14498.50
11	SB-1321	Miami	April	66486.0	69790.50	-3304.50
12	SB-263	Phoenix	April	53062.0	69790.50	-16728.50

Insight:

- Store performance may vary depending on the season or month in which the store was opened.
- Newly opened stores during off-peak months may show slower initial sales growth.
- Identifying seasonal patterns helps in planning future openings and optimizing early-stage marketing efforts.

2. Customer & Rating Insights

1.High customer footfall but low store ratings

```sql
--High customer footfall but low store ratings
WITH avg_values AS (
    SELECT
        (SELECT AVG(num_customers) FROM stores) AS avg_cust,
        (SELECT AVG(rating) FROM stores) AS avg_rating
)
SELECT
    s.store_id,
    c.city_name,
    s.num_customers,
    s.rating
FROM stores s
JOIN cities c ON c.city_id = s.city_id
JOIN avg_values a ON TRUE
WHERE s.num_customers > a.avg_cust
  AND s.rating < a.avg_rating
  order by num_customers desc;
```

	store_id text	city_name text	num_customers integer	rating numeric (2,1)
1	SB-1030	Phoenix	4998	3.1
2	SB-1435	Houston	4990	3.4
3	SB-1143	Miami	4985	2.7
4	SB-1874	Seattle	4985	3.3
5	SB-1772	Chicago	4982	3.1
6	SB-1540	Phoenix	4978	3.1
7	SB-1079	Seattle	4962	3.0
8	SB-1508	Los Angeles	4960	2.6
9	SB-1157	Los Angeles	4958	2.7
10	SB-1509	Miami	4946	3.6
11	SB-706	Phoenix	4944	2.7
12	SB-1875	Los Angeles	4937	3.0

Insight:

- Stores with high customer traffic but low ratings indicate service bottlenecks or customer dissatisfaction.
- The issue likely arises from slow order processing, long wait times, or inconsistent drink quality.
- These stores risk losing repeat customers unless operational efficiency and service quality are improved.

2.Some cities have high customers but low ratings (City-level service issues)

```sql
--Some cities have high customers but low ratings (City-level service issues)
WITH city_ratings AS (
    SELECT
        c.city_name,
        ROUND(AVG(s.rating),1) AS avg_rating,
        ROUND(AVG(s.num_customers),0) AS avg_customers
    FROM stores s
    JOIN cities c ON s.city_id = c.city_id
    GROUP BY c.city_name
)
SELECT *
FROM city_ratings
WHERE avg_customers >
      (SELECT AVG(num_customers) FROM stores)
  AND avg_rating <
      (SELECT AVG(rating) FROM stores);
```

	city_name text	avg_rating numeric	avg_customers numeric
1	Phoenix	3.7	2888

Insight:

- Cities with high customer volume but low ratings indicate widespread service or operational issues across multiple stores.
- Poor ratings suggest problems like inconsistent quality, understaffing, or inadequate customer experience city-wide.
- These cities may require city-level training, staffing improvements, and operational standardization to improve customer satisfaction.

3. Manager-Level Insights

1.Some managers handle too many low-performing stores

```sql
--Some managers handle too many low-performing stores
WITH m_perf AS (
    SELECT
        m.manager_id,
        m.manager_name,
        COUNT(s.store_id) AS stores_managed,
        AVG(s.monthly_sales) AS avg_sales
    FROM managers m
    LEFT JOIN stores s ON s.manager_id = m.manager_id
    GROUP BY m.manager_id, m.manager_name
),
avg_managed AS (
    SELECT AVG(store_count) AS avg_stores
    FROM (
        SELECT COUNT(store_id) AS store_count
        FROM stores
        GROUP BY manager_id
    ) t
),
avg_sales_all AS (
    SELECT AVG(monthly_sales) AS overall_avg_sales
    FROM stores
)
SELECT
    mp.*
FROM m_perf mp
CROSS JOIN avg_managed am
CROSS JOIN avg_sales_all asa
WHERE mp.stores_managed > am.avg_stores
  AND mp.avg_sales < asa.overall_avg_sales;
```

	manager_id [PK] integer	manager_name text	stores_managed bigint	avg_sales numeric
1	1501	Wei	202	68777.282178217822
2	1507	Priya	205	67784.117073170732
3	1502	Robert	199	69184.085427135678

Insight:

- Identifies managers who handle more stores than average.
- Highlights those managers whose average sales are lower than the company average.
- Helps detect inefficient or overloaded managers who need support, training, or reallocation of stores.

2.Manager Performance Ranking

```sql
--Manager Performance Ranking
WITH manager_rank AS (
    SELECT
        m.manager_name,
        SUM(s.monthly_sales) AS total_sales
    FROM stores s
    JOIN managers m ON m.manager_id = s.manager_id
    GROUP BY m.manager_name
)
SELECT
    manager_name,
    total_sales,
    RANK() OVER (ORDER BY total_sales DESC) AS performance_rank
FROM manager_rank;
```

	manager_name text	total_sales numeric	performance_rank bigint
1	Priya	13895744.0	1
2	Wei	13893011.0	2
3	Robert	13767633.0	3
4	Liu	13331693.0	4
5	John	13154241.0	5
6	Carlos	12497367.0	6
7	Aisha	12217463.0	7
8	Emily	11789015.0	8

Insight:

- Ranking managers by total store sales helps identify top performers and those needing improvement.
- Shows how effectively each manager handles their assigned stores, highlighting strengths and weaknesses.
- Supports data-driven decisions for promotions, training, and store reassignments.

4. City-Level & Revenue Insights

1.City-Level Revenue Contribution

```sql
--City-Level Revenue Contribution
WITH city_sales AS (
    SELECT
        c.city_name,
        SUM(s.monthly_sales) AS total_sales
    FROM stores s
    JOIN cities c ON c.city_id = s.city_id
    GROUP BY c.city_name
)
SELECT
    city_name,
    total_sales,
    ROUND(100 * total_sales / SUM(total_sales) OVER (),2) AS contribution_percentage
FROM city_sales
ORDER BY total_sales DESC;
```

	city_name text	total_sales numeric	contribution_percentage numeric
1	New York	13585682.0	12.99
2	Boston	13474315.0	12.89
3	Houston	13217399.0	12.64
4	Chicago	12999191.0	12.43
5	Los Angeles	12986391.0	12.42
6	Miami	12936057.0	12.37
7	Seattle	12774351.0	12.22
8	Phoenix	12572781.0	12.03

Insight:

- Shows which cities contribute the highest share of total revenue, helping identify key markets.
- Highlights underperforming cities with low revenue contribution, signalling areas needing attention or improvement.
- Helps guide strategic decisions on marketing spend, resource allocation, and future store expansion.

1.Store Performance Issues

Insights:

- Several stores are **underperforming compared to their respective city averages**, indicating gaps in operational efficiency or local competition.
- **Older stores show signs of performance decline**, possibly due to outdated layouts, reduced customer engagement, or lack of renovation.
- A few stores have **high ratings but very low sales**, suggesting issues related to store location, visibility, or accessibility.
- Seasonal patterns are observed where **store performance varies based on the opening date**, indicating that some stores function as seasonal outlets.

Recommendations:

- Conduct **city-wise performance audits** to identify and resolve issues causing stores to fall below average.
- Implement a **renovation or modernization plan** for older stores to improve customer experience and boost sales.
- For high-rated but low-sales stores, evaluate **location suitability**, marketing reach, and consider relocation or targeted promotions.
- Analyse **seasonal trends** and align staffing, inventory, and marketing efforts with peak seasons to maximize revenue.

2.Customer & Rating Insights

Insights:

- Some stores experience **high customer footfall but still maintain low ratings**, indicating issues with service quality, wait times, or in-store experience.
- Certain cities have **consistently high customer volume but low overall ratings**, suggesting broader **city-level service challenges**, such as staffing shortages, training gaps, or inconsistent service standards.

Recommendations:

- Conduct **customer feedback analysis** to identify the root causes of low ratings despite high footfall—focus on service speed, staff behaviour, and product availability.

- Improve **staff training programs** across stores, especially in low-rated high-traffic locations, to ensure consistent service quality.
- Introduce **city-level service improvement plans**, such as standardized SOPs, quality audits, and regular performance reviews.
- Use customer footfall data to **optimize staffing**, ensuring busy stores are adequately staffed during peak hours.
- Implement **real-time feedback systems** (QR codes, quick surveys) to monitor service issues and respond quickly.

3.Manager-Level Insights

Insights:

- Some managers are responsible for **multiple low-performing stores**, which may indicate leadership challenges, ineffective supervision, or workload imbalance.
- The **manager performance ranking** shows clear variations in efficiency, with top managers consistently maintaining high-performing stores while others struggle to improve store outcomes.

Recommendations:

- Reevaluate **manager-to-store allocations** to ensure workloads are balanced and managers are not overwhelmed, especially those handling several underperforming stores.
- Provide **targeted training and mentoring** for low-ranked managers, focusing on leadership skills, performance management, and customer experience strategies.
- Introduce a **performance monitoring system** that tracks store KPIs (sales, ratings, customer footfall) and ties them to managerial evaluations.
- Reward high-performing managers through **recognition, incentives, or expanded responsibilities**, reinforcing positive performance.
- For managers consistently handling low-performing stores, consider **role reassignment or introducing co-managers** to support turnaround efforts.

4.City-Level & Revenue Insights

Insights:

- The analysis of **city-level revenue contribution** shows that certain cities generate a significantly higher share of total sales, while others lag behind despite similar numbers of stores or customer traffic.
- Revenue distribution across cities highlights **regional performance imbalances**, suggesting differences in market size, customer purchasing power, or store operational effectiveness.

Recommendations:

- Prioritize **investment and marketing efforts** in high-revenue cities to maximize returns and strengthen market dominance.
- For low-revenue cities, conduct **root-cause analysis** focusing on customer demographics, competition, store placement, and pricing strategies.
- Implement **city-specific growth plans**, including targeted promotions, store revamps, or localized product offerings to boost sales.
- Regularly track **city-level KPIs**—sales per store, customer footfall, rating averages—to identify emerging trends and respond quickly.
- Consider **reallocating resources** (staffing, inventory, ad spend) based on revenue contribution to optimize overall business performance.

Conclusion

This Starbucks store performance analysis project, built on a carefully normalized SQL data model and advanced analytical queries, has revealed meaningful patterns across sales, customer footfall, store ratings, and managerial efficiency. The insights uncovered—such as underperforming locations, overloaded managers, city-level revenue variations, and mismatches between customer traffic and satisfaction—equip decision-makers with a clear understanding of where operational improvements are most needed.

By converting raw store-level data into actionable business intelligence, this project demonstrates how data-driven analysis can support smarter decisions in store optimization, resource allocation, customer experience enhancement, and future expansion planning. Ultimately, the study highlights the practical value of SQL and analytics in strengthening Starbucks' overall performance and enabling long-term, sustainable growth.