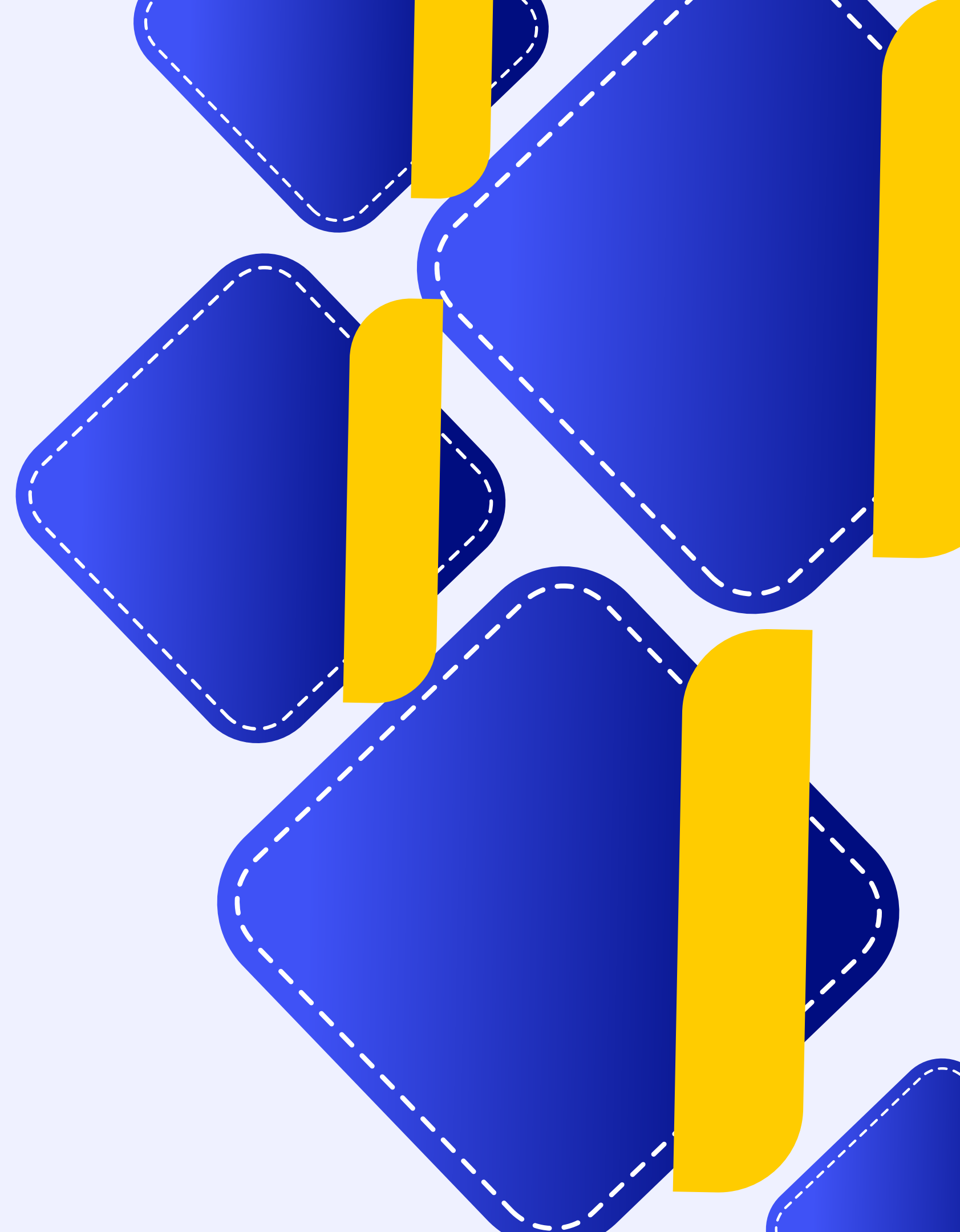




Walmart Sales Data Analysis

**Unveiling Key Sales Insights
Through Data-Driven Analysis**



A decorative blue swirl graphic located above the title.

Project Overview

- End-to-end data analysis solution for extracting key business insights from Walmart sales data.
- Utilized Python for data processing and SQL for querying.
- Focus on structured problem-solving techniques to answer business questions.
- Ideal for data analysts seeking hands-on experience in data manipulation, SQL querying, and pipeline creation.



Tools & Technologies Used

- **Programming Languages:** Python, SQL (MySQL & PostgreSQL)
- **Development Environment:** Visual Studio Code (VS Code)
- **Libraries & Packages:** Pandas, NumPy, SQLAlchemy, MySQL-Connector-Python, Psycopg2
- **API:** Kaggle API for dataset extraction
- **Databases:** MySQL and PostgreSQL

Project Steps - Setup & Data Extraction

1. Set Up Environment:

- Organized workspace in VS Code for structured data handling.

2. Set Up Kaggle API:

- Downloaded Walmart Sales Dataset via Kaggle API.
- Stored dataset in a dedicated folder.

3. Install Required Libraries:

- Installed essential libraries using pip (pandas, numpy, sqlalchemy, etc.).

4. Load Data:

- Read dataset into Pandas DataFrame for initial analysis.



Data Exploration & Cleaning

- Exploration:

- Used `.info()`, `.describe()`, and `.head()` to inspect data structure.
- Checked for data inconsistencies.

- Cleaning Process:

- Removed Duplicates to avoid skewed results.
- Handled Missing Values by filling or dropping as necessary.
- Standardized Data Types for consistency.
- Formatted Currency Columns for accuracy in financial analysis.



Feature Engineering & Database Loading

- Feature Engineering:

- Created a new Total Amount column ($\text{unit_price} * \text{quantity}$).

- Database Loading:




- Established MySQL and PostgreSQL connections using SQLAlchemy.
- Created and loaded cleaned data tables into both databases.
- Verified data integrity with SQL queries.

SQL-Based Business Analysis

1. Analyze Payment Methods and Sales:

- **Question:** What are the different payment methods, and how many transactions and items were sold with each method?
- **Purpose:** This helps understand customer preferences for payment methods, aiding in payment optimization strategies.

```
SELECT
  DISTINCT payment_method,
  COUNT(*) AS no_payments,
  SUM(quantity) AS no_qty_sold
FROM walmart
GROUP BY payment_method;
```

	payment_method 	no_payments 	no_qty_sold 
	text	bigint	double precision
1	Cash	1832	4984
2	Credit card	4256	9567
3	Ewallet	3881	8932

SQL-Based Business Analysis

2. Identify the Highest-Rated Category in Each Branch:

- **Question:** Which category received the highest average rating in each branch?
- **Purpose:** This allows Walmart to recognize and promote popular categories in specific branches, enhancing customer satisfaction and branch-specific marketing.

```
SELECT *  
FROM  
(  
  SELECT  
    branch,  
    category,  
    AVG(rating) AS avg_rating,  
    RANK() OVER(PARTITION BY branch ORDER BY AVG(rating) DESC) AS rank  
  FROM walmart  
  GROUP BY 1,2  
)  
WHERE rank = 1  
LIMIT 5;
```

	branch text	category text	avg_rating double precision	rank bigint
1	WALM001	Electronic accessories	7.45	1
2	WALM002	Food and beverages	8.25	1
3	WALM003	Sports and travel	7.5	1
4	WALM004	Food and beverages	9.3	1
5	WALM005	Health and beauty	8.366666666666667	1

SQL-Based Business Analysis

3. Determine the Busiest Day for Each Branch:

- **Question:** What is the busiest day of the week for each branch based on transaction volume?
- **Purpose:** This insight helps in optimizing staffing and inventory management to accommodate peak days.

```
SELECT *  
FROM  
    (SELECT  
        branch,  
        TO_CHAR(TO_DATE(date, 'DD/MM/YY'), 'Day') AS day_name,  
        COUNT(*) AS no_transactions,  
        RANK() OVER(PARTITION BY branch ORDER BY COUNT(*) DESC) AS rank  
    FROM walmart  
    GROUP BY 1, 2  
    ) subquery  
WHERE rank = 1  
LIMIT 5;
```

	branch text	day_name text	no_transactions bigint	rank bigint
1	WALM001	Thursday	16	1
2	WALM002	Thursday	15	1
3	WALM003	Tuesday	33	1
4	WALM004	Sunday	14	1
5	WALM005	Wednesday	19	1

SQL-Based Business Analysis

4. Calculate Total Quantity Sold by Payment Method:

- **Question:** How many items were sold through each payment method?
- **Purpose:** This helps Walmart track sales volume by payment type, providing insights into customer purchasing habits.

```
SELECT  
    DISTINCT payment_method,  
    SUM(quantity) AS no_qty_sold  
FROM walmart  
GROUP BY payment_method;
```

	payment_method text	no_qty_sold double precision
1	Cash	4984
2	Credit card	9567
3	Ewallet	8932

SQL-Based Business Analysis

5. Analyze Category Ratings by City:

● **Question:** What are the average, minimum, and maximum ratings for each category in each city?

● **Purpose:** This data can guide city-level promotions, allowing Walmart to address regional preferences and improve customer experiences.

```
SELECT
  city, category,
  MIN(rating) AS min_rating,
  MAX(rating) AS max_rating,
  AVG(rating) AS avg_rating
FROM walmart
GROUP BY 1, 2
LIMIT 5;
```

	city text	category text	min_rating double precision	max_rating double precision	avg_rating double precision
1	Little Elm	Fashion accessories	4	9.6	6.118181818181818
2	Mesquite	Sports and travel	7.8	7.8	7.8
3	Canyon	Health and beauty	5.8	8.9	6.9000000000000001
4	McKinn...	Home and lifestyle	3	9	5.9270270270270276
5	Brown...	Food and beverages	6.4	9.2	7.8

SQL-Based Business Analysis

6. Calculate Total Profit by Category:

- **Question:** What is the total profit for each category, ranked from highest to lowest?
- **Purpose:** Identifying high-profit categories helps focus efforts on expanding these products or managing pricing strategies effectively.

```
SELECT
    category,
    SUM(total) AS total_revenue,
    SUM(total * profit_margin) AS profit
FROM walmart
GROUP BY 1;
```

	category text	total_revenue double precision	profit double precision
1	Fashion accessories	489480.89999999997	192314.89320000037
2	Electronic accessories	78175.02999999998	30772.489499999978
3	Health and beauty	46851.17999999998	18671.7345
4	Food and beverages	53471.280000000006	21552.862200000003
5	Sports and travel	52497.930000000002	20613.808199999996
6	Home and lifestyle	489250.06	192213.63809999999

SQL-Based Business Analysis

7. Determine the Most Common Payment Method per Branch:

- **Question:** What is the most frequently used payment method in each branch?
- **Purpose:** This information aids in understanding branch-specific payment preferences, potentially allowing branches to streamline their payment processing systems.

```
WITH cte
AS
(SELECT
  branch,
  payment_method,
  COUNT(*) AS total_trans,
  RANK() OVER(PARTITION BY branch ORDER BY COUNT(*) DESC) AS rank
FROM walmart
GROUP BY 1, 2
)
SELECT *
FROM cte
WHERE rank =1
LIMIT 5;
```

	branch text	payment_method text	total_trans bigint	rank bigint
1	WALM001	Ewallet	45	1
2	WALM002	Ewallet	37	1
3	WALM003	Credit card	115	1
4	WALM004	Ewallet	44	1
5	WALM005	Ewallet	56	1

SQL-Based Business Analysis

8. Analyze Sales Shifts Throughout the Day:

- **Question:** How many transactions occur in each shift (Morning, Afternoon, Evening) across branches?
- **Purpose:** This insight helps in managing staff shifts and stock replenishment schedules, especially during high-sales periods.

```
SELECT
    branch,
CASE
    WHEN EXTRACT (HOUR FROM(time::time)) < 12 THEN 'Morning'
    WHEN EXTRACT (HOUR FROM(time::time)) BETWEEN 12 AND 17 THEN 'Afternoon'
    ELSE 'Evening'
END day_time,
COUNT(*)
FROM walmart
GROUP BY 1, 2
ORDER BY 1, 3 DESC;
```

	branch text	day_time text	count bigint
1	WALM001	Afternoon	36
2	WALM001	Evening	30
3	WALM001	Morning	8
4	WALM002	Afternoon	29
5	WALM002	Evening	21

SQL-Based Business Analysis



9. Identify Branches with Highest Revenue Decline Year-Over-Year :

- **Question:** Which branches experienced the largest decrease in revenue compared to the previous year?
- **Purpose:** Detecting branches with declining revenue is crucial for understanding possible local issues and creating strategies to boost sales or mitigate losses.

```
SELECT *,  
EXTRACT(YEAR FROM TO_DATE(date, 'DD/MM/YY')) as formatted_date  
FROM walmart  
--  
WITH revenue_2022  
AS  
(  
    SELECT  
        branch,  
        SUM(total) as revenue  
    FROM walmart  
    WHERE EXTRACT(YEAR FROM TO_DATE(date, 'DD/MM/YY')) = 2022  
    GROUP BY 1  
)  
--  
revenue_2023  
AS  
(  
    SELECT  
        branch,  
        SUM(total) as revenue  
    FROM walmart  
    WHERE EXTRACT(YEAR FROM TO_DATE(date, 'DD/MM/YY')) = 2023  
    GROUP BY 1  
)
```

```
SELECT  
    ls.branch,  
    ls.revenue as last_year_revenue,  
    cs.revenue as cr_year_revenue,  
    ROUND(  
        (ls.revenue - cs.revenue)::numeric/  
        ls.revenue::numeric * 100,  
        2) as rev_dec_ratio  
FROM revenue_2022 as ls  
JOIN  
    revenue_2023 as cs  
ON ls.branch = cs.branch  
WHERE  
    ls.revenue > cs.revenue  
ORDER BY 4 DESC  
LIMIT 5;
```

	branch text	last_year_revenue double precision	cr_year_revenue double precision	rev_dec_ratio numeric
1	WALM045	1731	647	62.62
2	WALM047	2581	1069	58.58
3	WALM098	2446	1030	57.89
4	WALM033	2099	931	55.65
5	WALM081	1723	850	50.67



Business Insights & Key Findings

- Sales Insights:

- Identified highest revenue-generating categories & branches.
- Analyzed customer payment preferences.

- Profitability:

- Determined most profitable product categories.

- Profitability:

- Identified peak shopping hours and sales shifts throughout the day.
- Analyzed branch-specific customer trends.

Conclusion & Acknowledgments

- Successfully developed an end-to-end Walmart Sales Data Analysis project.
- Utilized Python, SQL, and Kaggle API for data extraction, transformation, and querying.
- Derived actionable business insights to optimize Walmart's sales strategies.
- Inspiration: Walmart's business case studies on sales and supply chain optimization

THANK YOU



bhaskar.official@gmail.com