

# WALLMART QUERIES

SALES BUSINESS QUERIES





**WELCOME TO**

# **WALLMART QUERIES**



THIS PROJECT IS AN END-TO-END DATA ANALYSIS SOLUTION DESIGNED TO EXTRACT CRITICAL BUSINESS INSIGHTS FROM WALMART SALES DATA. WE UTILIZE PYTHON FOR DATA PROCESSING AND ANALYSIS, SQL FOR ADVANCED QUERYING, AND STRUCTURED PROBLEM-SOLVING TECHNIQUES TO SOLVE KEY BUSINESS QUESTIONS. THE PROJECT IS IDEAL FOR DATA ANALYSTS LOOKING TO DEVELOP SKILLS IN DATA MANIPULATION, SQL QUERYING, AND DATA PIPELINE CREATION.





# FIND DIFFERENT PAYMENT METHODS, NUMBER

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

Result Grid     Filter Rows: <input type="text"/>			
	payment_method	no_payments	no_qty_sold
▶	Ewallet	3881	8932
	Cash	1832	4984
	Credit card	4256	9567



# IDENTIFY THE HIGHEST-RATED CATEGORY IN EACH BRANCH DISPLAY THE BRANCH, CATEGORY, AND AVG RATING

```
SELECT branch, category, avg_rating
FROM (
    SELECT
        branch,
        category,
        AVG(rating) AS avg_rating,
        RANK() OVER(PARTITION BY branch ORDER BY AVG(rating)
        DESC) AS rank_position
    FROM walmart.walmart_clean_data
    GROUP BY branch, category
) AS ranked
WHERE rank_position = 1;
```

Result Grid	Filter Rows:	Export:
branch	category	avg_rating
WALM001	Electronic accessories	7.45
WALM002	Food and beverages	8.25
WALM003	Sports and travel	7.5
WALM004	Food and beverages	9.3
WALM005	Health and beauty	8.366666666666667
WALM006	Fashion accessories	6.797058823529412
WALM007	Food and beverages	7.55
WALM008	Food and beverages	7.4
WALM009	Sports and travel	9.6
WALM010	Electronic accessories	9
WALM011	Food and beverages	7
WALM012	Health and beauty	7.45



# IDENTIFY THE BUSIEST DAY FOR EACH BRANCH BASED ON THE NUMBER OF TRANSACTIONS

```
SELECT branch, day_name, no_transactions
FROM (
  SELECT
    branch,
    DAYNAME(STR_TO_DATE(date, '%d/%m/%Y')) AS day_name,
    COUNT(*) AS no_transactions,
    RANK() OVER(PARTITION BY branch ORDER BY COUNT(*)
    DESC) AS rank_position
  FROM walmart.walmart_clean_data
  GROUP BY branch, day_name
) AS ranked
WHERE rank_position = 1;
```

Result Grid			
Filter Rows:			
	branch	day_name	no_transactions
▶	WALM001	Thursday	16
	WALM002	Thursday	15
	WALM003	Tuesday	33
	WALM004	Sunday	14
	WALM005	Wednesday	19
	WALM006	Thursday	15
	WALM007	Friday	12
	WALM007	Sunday	12
	WALM008	Tuesday	17
	WALM009	Sunday	42
	WALM010	Wednesday	12
	WALM011	Tuesday	10



# CALCULATE THE TOTAL QUANTITY OF ITEMS SOLD PER PAYMENT METHOD

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

Result Grid



Filter Rows:

	payment_method	no_qty_sold
▶	Ewallet	8932
	Cash	4984
	Credit card	9567



# DETERMINE THE AVERAGE, MINIMUM, AND MAXIMUM RATING OF CATEGORIES FOR EACH CITY

```
SELECT
    city,
    category,
    MIN(rating) AS min_rating,
    MAX(rating) AS max_rating,
    AVG(rating) AS avg_rating
FROM
    walmart.walmart_clean_data
GROUP BY city , category;
```

Result Grid

Filter Rows:

Export:



Wrap Cell Content:

	city	category	min_rating	max_rating	avg_rating
▶	San Antonio	Health and beauty	5	9.1	7.05
	Harlingen	Electronic accessories	9.6	9.6	9.6
	Haltom City	Home and lifestyle	3	9.5	6.227777777777778
	Bedford	Health and beauty	6.1	9.3	8.15
	Irving	Sports and travel	5.3	5.3	5.3
	Denton	Electronic accessories	4.1	9	6.7
	Cleburne	Electronic accessories	5.8	7.8	7.25
	Canyon	Home and lifestyle	3	9	6.25
	Grapevine	Health and beauty	7.2	7.2	7.2
	Texas City	Food and beverages	5.5	5.9	5.7
	Irving	Fashion accessories	3	9.8	6.206896551724138
	San Angelo	Electronic accessories	3	7	5.8307692307692305



# CALCULATE THE TOTAL PROFIT FOR EACH CATEGORY

```
SELECT
    category,
    SUM(unit_price * quantity * profit_margin) AS total_profit
FROM
    walmart.walmart_clean_data
GROUP BY category
ORDER BY total_profit DESC;
```

Result Grid     Filter Rows: <input type="text"/>		
	category	total_profit
▶	Health and beauty	0
	Electronic accessories	0
	Home and lifestyle	0
	Sports and travel	0
	Food and beverages	0
	Fashion accessories	0



# DETERMINE THE MOST COMMON PAYMENT METHOD FOR EACH BRANCH

```
WITH cte AS (  
    SELECT  
        branch,  
        payment_method,  
        COUNT(*) AS total_trans,  
        RANK() OVER(PARTITION BY branch ORDER BY COUNT(*)  
        DESC) AS rank_position  
    FROM walmart.walmart_clean_data  
    GROUP BY branch, payment_method  
)  
SELECT branch, payment_method AS preferred_payment_method  
FROM cte  
WHERE rank_position = 1;
```

Result Grid		Filter Rows:
branch	preferred_payment_method	
WALM001	Ewallet	
WALM002	Ewallet	
WALM003	Credit card	
WALM004	Ewallet	
WALM005	Ewallet	
WALM006	Ewallet	
WALM007	Ewallet	
WALM008	Ewallet	
WALM009	Credit card	
WALM010	Ewallet	
WALM011	Ewallet	
WALM012	Ewallet	
WALM013	Ewallet	



# CATEGORIZE SALES INTO MORNING, AFTERNOON, AND EVENING SHIFTS

```
SELECT
  branch,
  CASE
    WHEN HOUR(TIME(time)) < 12 THEN 'Morning'
    WHEN HOUR(TIME(time)) BETWEEN 12 AND 17 THEN 'Afternoon'
    ELSE 'Evening'
  END AS shift,
  COUNT(*) AS num_invoices
FROM
  walmart.walmart_clean_data
GROUP BY branch , shift
ORDER BY branch , num_invoices DESC;
```

Result Grid			
Filter Rows:			
	branch	shift	num_invoices
▶	WALM001	Afternoon	36
	WALM001	Evening	30
	WALM001	Morning	8
	WALM002	Afternoon	29
	WALM002	Evening	21
	WALM002	Morning	15
	WALM003	Afternoon	95
	WALM003	Morning	50
	WALM003	Evening	41
	WALM004	Afternoon	27
	WALM004	Evening	24
	WALM004	Morning	9
	WALM005	Evening	35



**THANK YOU**

