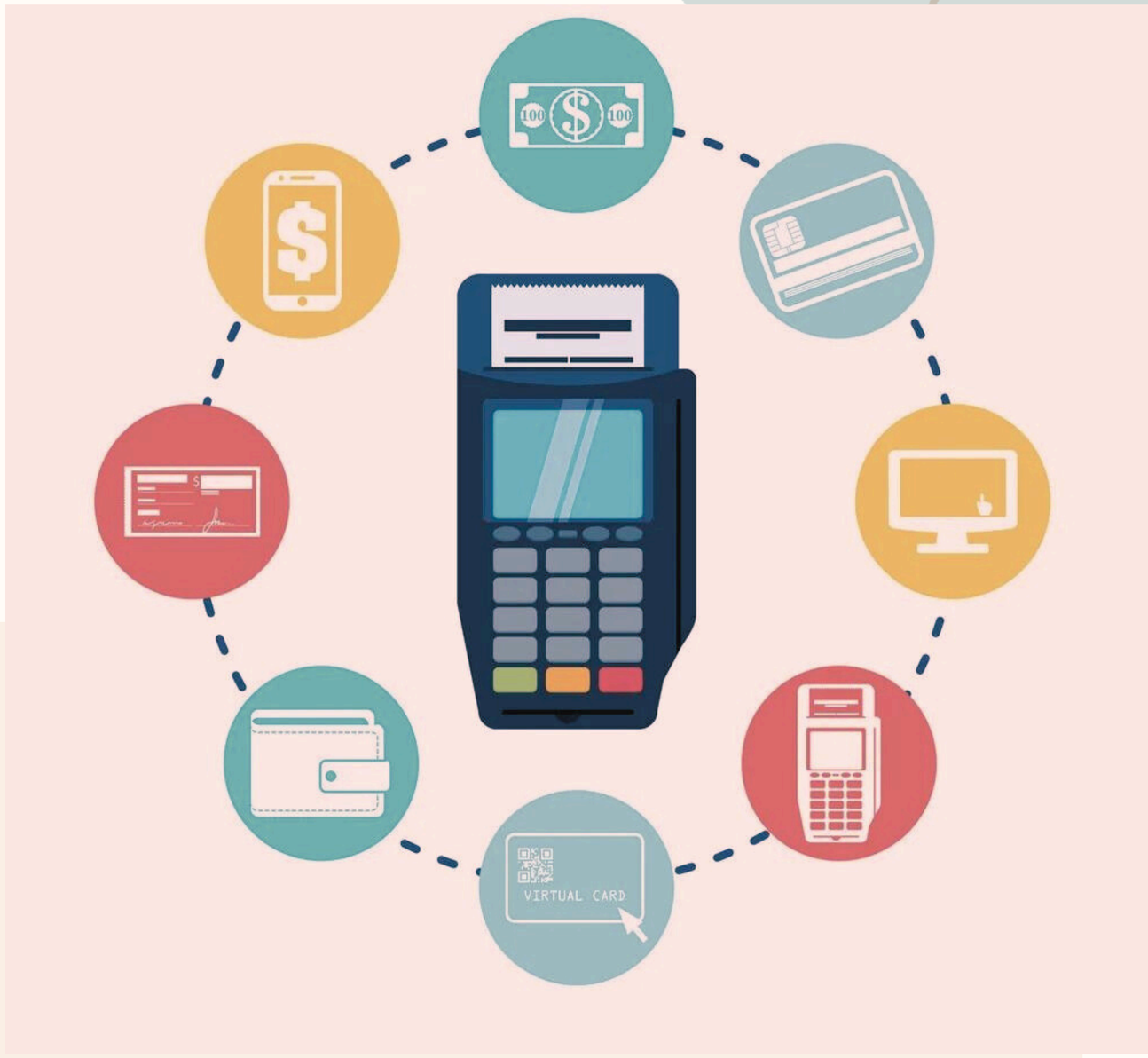
Digital Payment Transaction Analysis Using SQL

Presented by Abhay Rawat

OBJECTIVE:-

The goal of this project is to use SQL to examine a dataset of 5,000 records of digital wallet transactions. By analysing this data, we aim to reveal important insights about transaction habits, how different payment methods used for transactions, and get to know more details about transaction amount, transaction status, payment category, merchant details, device information, location, and user behaviours. This will help us to identify trends and improve digital wallet services.



KEY ANALYSIS GOALS:-

1- Transaction Patterns by Payment Methods:

- **Maximum Transactions:** Find the payment method with the maximum number of transactions.
- **Total Transactions:** Calculate the total number of transactions for each payment method.
- **Highest Charge:** Identify the payment method with the highest charge.
- **Highest Cashback:** Determine the payment method offering the highest cashback.
- **Loyalty Points:** Find Top 5 Users based on Highest Loyalty Points
- **Loyalty points by each payment method:** Highest Loyalty points by each payment method

2- Device and Location Analysis:

- **Transactions by Device:** Identify the device with the maximum number of transactions.
- **Transactions by Location:** Find the location with the highest number of transactions.

3- Product and Merchant Performance:

- **Top Products:** List the top 5 products based on transaction amounts.
- **Top Merchants by Transaction Counts:** Identify the merchant with the highest number of transactions.
- **Top Merchants by Transaction Amounts:** Find the merchant with the highest transaction amount.

4- Category and User Insights :-

- **Top Product Category:** Discover the top 5 product categories with the highest transaction volumes.
- **Top Users by Transactions Amounts:** List the top 5 users with the highest transaction amounts.

5- Payment Transactions Status:-

- **Status of Failed Payment:** Total Count of Failed Payment Transactions.
- **Status of Failed Payment by each Payment Method:** Number of Failed Payment Transactions by Each Payment Method.
- **Status of Pending Payment:** Total Count of Pending Payment Transactions.
- **Status of Pending Payment by each Payment Method:** Number of Pending Payment Transactions by each Payment Method.
- **Status of Successful Payment:** Total Count of Successful Payment Transactions.
- **Status of Successful Payment by each Payment Method:** Number of Pending Payment Transactions by each Payment Method.

SQL QUERIES FOR ANALYSIS

Q1-Find the maximum number of digital transactions recorded for each payment method:-

QUERY-1

```sql
SELECT
    payment_method, COUNT(payment_method) AS Payment_count
FROM
    digital_wallet_transactions
GROUP BY payment_method
ORDER BY payment_count DESC;
```

OUTPUT

payment_method	Payment_count
Bank Transfer	1045
Debit Card	1022
UPI	999
Credit Card	992
Wallet Balance	942

"BANK TRANSFER has the maximum number of transactions, with a total of 1045, followed by DEBIT CARD with 1022 transactions"

Q2- Calculate the total number of transactions for each payment method:-

```sql
SELECT
    payment_method,
    ROUND(SUM(product_amount), 2) AS Total_Transaction
FROM
    digital_wallet_transactions
GROUP BY payment_method;
```

QUERY

OUTPUT

payment_method	Total_Transaction
Debit Card	5160183.4
UPI	4898628.78
Wallet Balance	4623590.42
Credit Card	4886969.02
Bank Transfer	5218141.99

"BANK TRANSFER has the highest amount of transactions, followed by DEBIT CARD"

Q3-Identify the payment method with the highest charge:-

QUERY

```sql
SELECT
    payment_method,
    ROUND(SUM(transaction_fee), 2) AS Total_Charge
FROM
    digital_wallet_transactions
GROUP BY payment_method
ORDER BY Total_Charge DESC;
```

payment_method	Total_Charge
UPI	25801.92
Bank Transfer	25780.24
Debit Card	25695.74
Credit Card	25206.79
Wallet Balance	23459.68

OUTPUT

"UPI takes the highest charges, followed by BANK TRANSFER"

Q4-Determine the payment method offering the highest cashback:-

QUERY

```sql
SELECT
    payment_method, ROUND(SUM(cashback), 2) AS Total_Cashback
FROM
    digital_wallet_transactions
GROUP BY payment_method
ORDER BY Total_Cashback DESC;
```

payment_method	Total_Cashback
Bank Transfer	52385.87
Debit Card	52269.06
UPI	52117.45
Credit Card	48612.78
Wallet Balance	47908.75

OUTPUT

“BANK TRANSFER has offered the highest cashback,followed by DEBIT CARD”

Q5- Find Top 5 Users based on Highest Loyalty Points:-

QUERY

```sql
select user_id,round(sum(loyalty_points),2) AS Loyalty_Points
FROM
    digital_wallet_transactions
GROUP BY user_id
ORDER BY Loyalty_Points DESC
LIMIT 5;
```

user_id	Loyalty_Points
USER_08584	3109
USER_08946	2760
USER_05836	2572
USER_07771	2450
USER_04319	2438

OUTPUT

“USER_08584 has the Highest Loyalty Points with 2760, followed by USER_08946 ,with 3109 points”

Q6- Highest Loyalty points by each payment method:-

QUERY

```
SELECT
    payment_method, SUM(loyalty_points) AS Loyalty_Points
FROM
    digital_wallet_transactions
GROUP BY payment_method
ORDER BY loyalty_points DESC;
```

payment_method	Loyalty_Points
Debit Card	529586
Bank Transfer	513417
Credit Card	491335
UPI	485284
Wallet Balance	474330

OUTPUT

“DEBIT CARD has the highest number of Loyalty Points, with total 529586 points, followed by BANK TRANSFER, with total of 513417 points”

Q7-Identify the device with the maximum number of transactions:-

QUERY

```
SELECT
    device_type, COUNT(device_type) AS Transaction_Count
FROM
    digital_wallet_transactions
GROUP BY device_type
ORDER BY Transaction_Count DESC;
```

device_type	Transaction_Count
Android	2995
iOS	1526
Web	479

OUTPUT

“ANDROID has the maximum number of Transaction, with 2995 Transactions, followed by IOS with 1526 Transactions”

Q8- Find the location with the highest number of transactions:-

QUERY

```sql
SELECT
    location, ROUND(SUM(product_amount), 2) AS Total_Transaction
FROM
    digital_wallet_transactions
GROUP BY location
ORDER BY Total_Transaction DESC;
```

OUTPUT

location	Total_Transaction
Urban	17249266.19
Suburban	5013900.17
Rural	2524347.25

“URBAN area has the highest number of Transaction”

Q9-Name of list of top 5 products based on transaction amounts:-

QUERY

```sql
SELECT
    product_name,
    ROUND(SUM(product_amount), 2) AS Total_Transaction
FROM
    digital_wallet_transactions
GROUP BY product_name
ORDER BY product_name desc limit 5;
```

product_name	Total_Transaction
Wireless Earbuds	269831.42
Whole Wheat Atta	231284.5
Volvo Multi-Axle	210782.4
Villa	265406.57
Vegetable Biryani	302472.94

OUTPUT

"The WIRELESS EARBUDS has the highest transaction amount ,followed by the WHOLE WHEAT ATTA"

Q10- Identify the merchant with the highest number of transactions:-

```sql
SELECT
    merchant_name, COUNT(merchant_name) AS Transaction_Count
FROM
    digital_wallet_transactions
GROUP BY merchant_name
ORDER BY Transaction_count DESC limit 5;
```

QUERY

OUTPUT

merchant_name	Transaction_Count
Airbnb	106
Flipkart	102
MakeMyTrip	85
Netflix	72
Unacademy	69

“AIRBNB has the highest number of transactions, with 106 counts, followed by FLIPKART with 102”

Q11- Find the merchant with the highest transaction amount:-

QUERY

```sql
SELECT
    merchant_name, round(sum(product_amount),2) AS Total_Transaction
FROM
    digital_wallet_transactions
GROUP BY merchant_name
ORDER BY ToTal_Transaction DESC limit 5;
```

merchant_name	Total_Transaction
Airbnb	550248.4
Flipkart	458472.89
MakeMyTrip	438496.77
Bangalore Water Supply	337482.55
Netflix	335355.47

OUTPUT

“AIRBNB has the highest transaction amount, followed by FLIPKART”

Q12- Discover the top 5 product categories with the highest transaction volumes:-

QUERY

```sql
SELECT
    product_category,
    ROUND(SUM(product_amount), 2) AS Total_Transaction
FROM
    digital_wallet_transactions
GROUP BY product_category
ORDER BY Total_Transaction DESC
LIMIT 5;
```

product_category	Total_Transaction
Streaming Service	1462461.66
Water Bill	1400668.85
Gas Bill	1361519.8
Education Fee	1349321.77
Movie Ticket	1337399.89

“The STREAMING SERVICES has the highest amount of transactions, followed by WATER BILL”

Q13-Name of the list the top 5 users with the highest transaction amounts:-

QUERY

```sql
SELECT
    user_id,ROUND(SUM(product_amount), 2) AS Total_Transaction
FROM
    digital_wallet_transactions
GROUP BY user_id
ORDER BY Total_Transaction DESC
LIMIT 5;
```

user_id	Total_Transaction
USER_04724	29315.93
USER_08946	28799.29
USER_05836	25189.19
USER_03945	25063.95
USER_05744	24949.89

OUTPUT

“USER_04724 has the highest amount transaction, followed by USER_08946”

Q14- Total Count of Failed Payment Transactions:-

```sql
SELECT transaction_status,
    COUNT(Transaction_status) AS Failed_Transaction
FROM
    digital_wallet_transactions
WHERE
    transaction_status = 'failed';
```

QUERY

OUTPUT

transaction_status	Failed_Transaction
Failed	146

“Total Count of Failed payment is 146”

Q15-Number of Failed Payment Transactions by Each Payment Method:-

QUERY

```sql
SELECT
    payment_method, COUNT(transaction_status) AS Failed_Transaction
FROM
    digital_wallet_transactions
WHERE
    transaction_status = 'failed'
GROUP BY payment_method
ORDER BY Failed_Transaction DESC;
```

payment_method	Failed_Transaction
Credit Card	35
Wallet Balance	32
Bank Transfer	29
Debit Card	25
UPI	25

OUTPUT

“The CREDIT CARD has the highest number of failed payment status, with 35 transactions, followed by the WALLET BALANCE ,with 32 transactions”

Q16- Total Count of Pending Payment Transactions:-

```sql
SELECT transaction_status,
    COUNT(Transaction_status) AS Pending_Transaction
FROM
    digital_wallet_transactions
WHERE
    transaction_status = 'Pending';
```

QUERY

transaction_status	Pending_Transaction
Pending	99

OUTPUT

“Total Count of Pending payment is 99”

Q17-Number of Pending Payment Transactions by each Payment Method:-

QUERY

```
SELECT payment_method,
    COUNT(Transaction_status) AS Pending_Transaction
FROM
    digital_wallet_transactions
WHERE
    transaction_status = 'Pending' group by payment_method;
```

OUTPUT

payment_method	Pending_Transaction
Bank Transfer	18
UPI	24
Credit Card	24
Wallet Balance	24
Debit Card	9

“UPI, CREDIT CARD and WALLET BALANCE has the same number of pending Transactions with a count of 24”

Q18- Total Count of Successful Payment Transactions:-

QUERY

```sql
SELECT transaction_status,
    COUNT(Transaction_status) AS Successful_Transaction
FROM
    digital_wallet_transactions
WHERE
    transaction_status = 'successful';
```

transaction_status	Successful_Transaction
Successful	4755

OUTPUT

“Total Count of Successful Payment Transaction is 4755”

Q19- Number of Pending Payment Transactions by each Payment Method:-

```sql
SELECT payment_method,
    COUNT(Transaction_status) AS Successful_Transaction
FROM
    digital_wallet_transactions
WHERE
    transaction_status = 'successful' group by payment_method;
```

QUERY

OUTPUT

payment_method	Successful_Transaction
Debit Card	988
UPI	950
Wallet Balance	886
Credit Card	933
Bank Transfer	998

“As we can see in the output section, BANK TRANSFER has 998 successful transactions, followed by DEBIT CARD 988 transactions”

Conclusions:-

The analysis reveals that bank transfers and debit cards are the most popular payment methods, primarily because they offer higher cash-back rewards and lower fees compared to UPI, which is associated with higher costs. This preference is further supported by the frequent issues of failed or pending transactions with Credit Cards and UPI, prompting users to choose more dependable options. Transactions are more prevalent among residents of urban and suburban areas, with these regions showing a higher volume of activity compared to rural areas. Airbnb and Flipkart emerge as the top merchants, indicating their strong customer base and trust. Most transactions are conducted using Android devices, highlighting their widespread adoption. Additionally, streaming services and water bills account for the highest transaction volumes. To enhance user satisfaction and competitiveness, it is crucial to address the transaction issues associated with UPI and credit cards, or to improve cash-back offers to make these payment methods more attractive.

Thank you!