BANKING FINANCIAL

INTRODUCTION:

This project focuses on analyzing banking financial data to understand customer behaviour, transaction patterns, loan activities and overall financial performance .The raw dataset was first cleaned and transformed using python to ensure accuracy and consistency. After pre-processing, the refined data was stored and analysed using MySQL to extract meaningful insights through structured queries. Finally, an interactive POWER BI dashboard was created to visually present the key trends, metrics, and patterns, enabling better decision-making for banking operations.

OBJECTIVES:

1. Data Cleaning in Python
    - Removing missing, duplicates and inconsistent records.
    - Standardize formats for date, currency, and categorical values.
    - Engineer useful features for deeper financial analysis.
    - Ensure the dataset is accurate, reliable, and analysis ready.
2. Data analyses using Mysql
    - Store the cleaned dataset in a structured database.
    - Perform Mysql queries for customer segmentation, loan performance, and transaction analysis.
    - Extract summary statistics and trends to support decision making.
    - Identify patterns such as high-value customers, transaction frequencies, and risk indicators.
3. Power BI dashboard Creation
    - Visualize key banking metrics using interactive charts.
    - Build dashboards that highlight customer trends, deposits, loans, revenue, and risk.
    - Help stakeholders quickly understand performance and take action.
    - Create drill-down reports for deeper insights into customer and financial activities.

PYTHON CLEANING

🔍 DATA CLEANING IN PYTHON:

- Imported all the libraries for cleaning and visualization.

```
[2]: import pandas as pd
     import numpy as np
     import matplotlib.pyplot as mat
     %matplotlib inline
     import numpy as np
     import seaborn as sns
```

Uploaded a CSV file.

- In data analysis and data pre-processing workflows, reading input data correctly is the first and most important step. CSV commonly used for storing tabular data.

This documentation explains how to read a csv file using python in the pandas library.

```
file=pd.read_csv("C:\\Users\\Kalaivani S\\Downloads\\banking.csv",nrows=100000)
file
```

	ComplaintID	datereceived	Product	Subproduct	Issue	Subissue	Company	State	ZIPcode	consent	Submitted	Datesent
0	2738619	27-11-2017	Mortgage	Conventional home mortgage	Trouble during payment process	NaN	NATIONSTAR MORTGAGE LLC	PA	189XX	Consent provided	Web	27-11-2017
1	2933849	12-06-2018	Mortgage	Conventional home mortgage	Trouble during payment process	NaN	NATIONSTAR MORTGAGE LLC	CO	80132	Consent provided	Web	12-06-2018
2	1165653	21-12-2014	Mortgage	Conventional fixed mortgage	Loan modification,collection,foreclosure	NaN	NATIONSTAR MORTGAGE LLC	CA	91701	Unknown	Web	21-12-2014
3	3316943	24-07-2019	Credit reporting, credit repair services, or o...	Credit reporting	Incorrect information on your report	Information belongs to someone else	Experian Information Solutions Inc.	CA	90221	Consent not provided	Web	24-07-2019
			Payday loan, title	Installment			Santander			Consent		20-02-

🔍 Getting the information about the dataset:

- Using the info() method in python during data cleaning.
  - ✓ Total number of rows and columns –To Understand dataset size.
  - ✓ Column names –To see all fields available.
  - ✓ Number of null values-used to detect missing data.
  - ✓ Data types of each columns.

```
file.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 15 columns):
 #   Column            Non-Null Count   Dtype 
---  ------            --------------   ----- 
 0   ComplaintID       100000 non-null  int64 
 1   datereceived      100000 non-null  object
 2   Product           100000 non-null  object
 3   Subproduct        94460 non-null   object
 4   Issue             100000 non-null  object
 5   Subissue          72438 non-null   object
 6   Company           100000 non-null  object
 7   State             97899 non-null   object
 8   ZIPcode           98153 non-null   object
 9   consent           100000 non-null  object
 10  Submitted         100000 non-null  object
 11  Datesent          100000 non-null  object
 12  response          100000 non-null  object
 13  consumerdispute   100000 non-null  object
 14  Companyresponse   100000 non-null  object
dtypes: int64(1), object(14)
memory usage: 11.4+ MB
```

🔍 Getting the summary of the dataset :

- It provides statistical measures for numerical columns like(count, std, min, max).
  - ✓ Count-num of non missing values.
  - ✓ Mean- average value.
  - ✓ Std- standard Deviation.
  - ✓ Min- minimum values.

To quickly understand the data quality.

To identify outliers.

```
file.describe()
```

	ComplaintID
count	1.000000e+05
mean	2.713482e+06
std	9.514194e+05
min	2.200000e+01
25%	2.690643e+06
50%	3.143367e+06
75%	3.395229e+06
max	3.487002e+06

🔍 To identify and count missing values:

- ✓ Checks every values in dataframe.
- ✓ Returns the count if the value is missing(NaN)
- ✓ Sum() adds up all the True values in each column.

```
file.isnull().sum()
```

```
ComplaintID             0
datereceived            0
Product                 0
Subproduct           5540
Issue                   0
Subissue                0
Company                 0
State                2101
ZIPcode              1847
consent                 0
Submitted               0
Datesent                0
response                0
consumerdispute         0
Companyresponse         0
dtype: int64
```

🔍 Filling Null values in a column using related column information:

- Used the information from the issue column to fill the sub-issue.
- ✓ Missing values in Sub-issue create incomplete records.
- ✓ The nulls affect analysis, grouping, and reporting.
- ✓ Since each issue has a correct and corresponding sub-issue, it is reliable to fill missing entries using the mapping.

```
issue_to_subissue={'Trouble during payment process':'follow up issue related to payment failure',
                   'Incorrect information on your report':'the issue is related to report correction',
                    'Loan servicing, payments, escrow account':'Additional issue linked to your payment account',
                   'Loan modification,collection,foreclosure':'the issue raised for loan adjustment matter',
                   'Attempts to collect debt not owed':'issue raised for incorrect dept collection',
                   "Problem with a credit reporting company's investigation into an existing problem":"problem with a credit reporting companys investme
                  'Written notification about debt':'written notification regarding the department',
                   'Struggling to pay your bill':'difficulty in paying the bill'
                  }

file['Subissue']=file.apply(lambda row:issue_to_subissue.get(row['Issue'],
                                                             row['Subissue']),axis=1)
```

🔍 Missing Sub-issue entries lead to incomplete records:

```
file['Subissue']=file['Subissue'].fillna('Incorrect or incompltete information')
file
```

	ComplaintID	datereceived	Product	Subproduct	Issue	Subissue	Company	State	ZIPcode	consent	Submitted	Datesent
0	2738619	27-11-2017	Mortgage	Conventional home mortgage	Trouble during payment process	follow up issue related to payment	NATIONSTAR MORTGAGE LLC	PA	189XX	Consent provided	Web	27-11-2017

🔍 Filling Null values in Sub-product using forward fill:

- Forward fill take the last known valid value from the column and fills it downward into the next null row.
    - ✓ All missing Sub-product values are filled.

```
file['Subproduct']=file['Subproduct'].fillna(method='ffill')
```

🔍 Filling Null values in State using Backward fill:

- Backward fill take the previous known valid value from the column and fills it upward into the next null row.
    - ✓ All missing State values are filled

```
file['State']=file['State'].fillna(method='bfill')
```

Replaces the values in ZIP-code:

- The Zip-code was missing
- The Zip-code was incorrectly formatted.
- This replacement ensures that the dataset maintains valid geographical information, improves data accuracy, and supports better analysis in later stages.

```
file['ZIPcode']=file['ZIPcode'].replace(['','X','nan'],np.nan)
```

Replaces the –values into NaN:

- The Zip-code have the negative values so, I replaced the values as nan

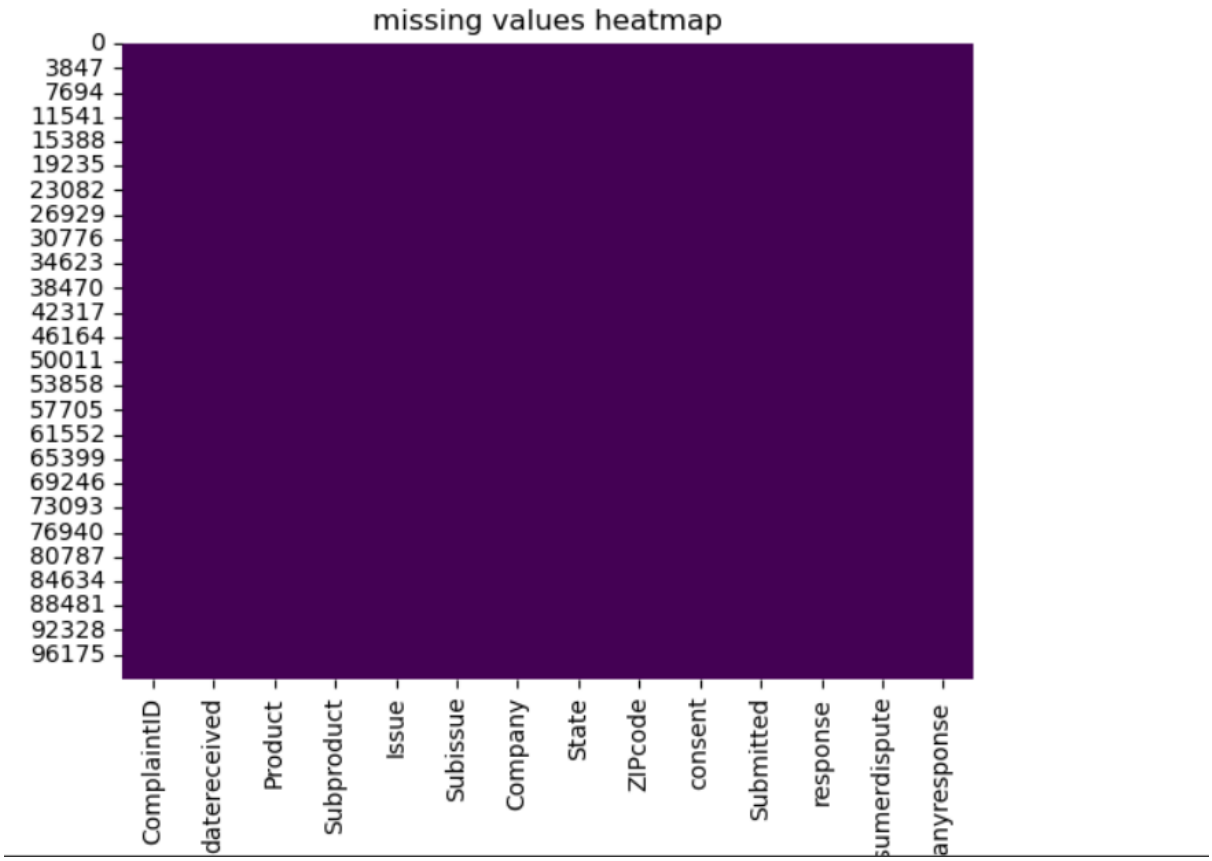
```
file['ZIPcode'].replace([np.inf, -np.inf], np.nan, inplace=True)
```

```
file['ZIPcode']=file['ZIPcode'].str.replace(r'[^0-9]','',regex=True)
```

VISUALIZATION IN PYTHON:

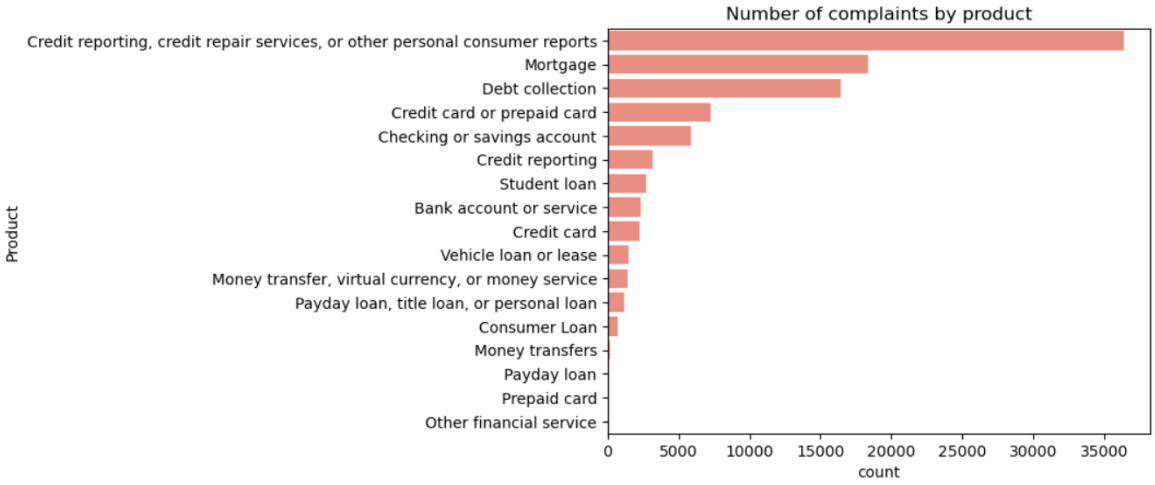
- Using Sea-born and Matplotlib

```
sns.heatmap(file.isnull(),cbar=False,cmap='viridis')
mat.title('missing values heatmap')
mat.show()
```



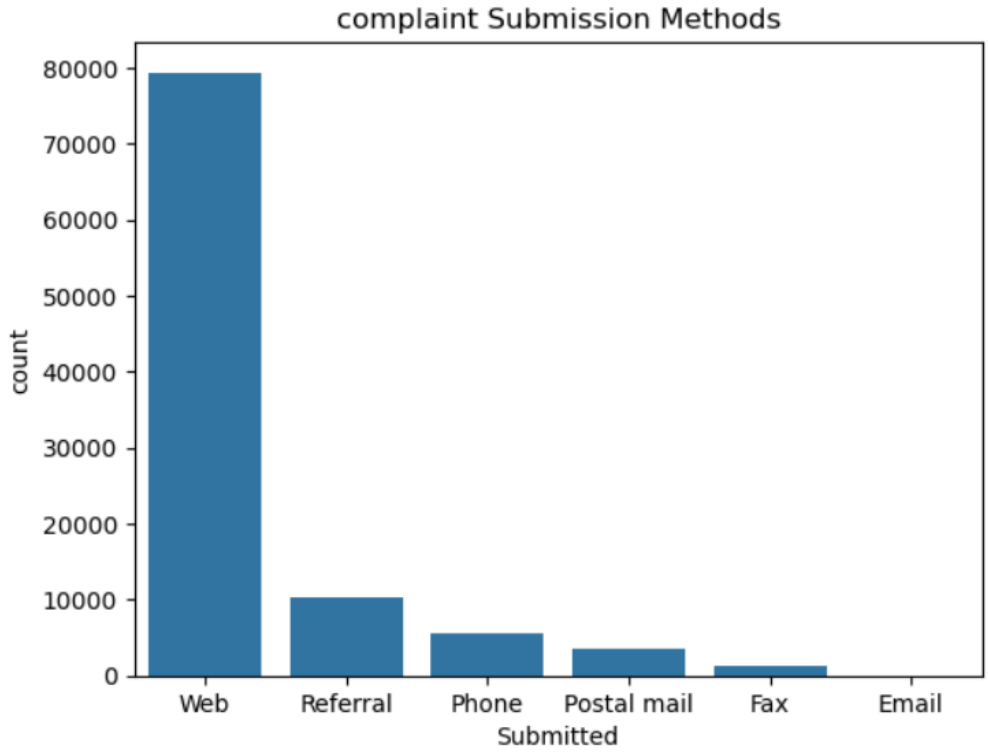
- Heat-map for missing values

```
sns.countplot(y='Product',data=file,
              order=file['Product'].value_counts().index,color='salmon')
mat.title('Number of complaints by product')
mat.xlabel('count')
mat.ylabel('Product')
mat.show()
```

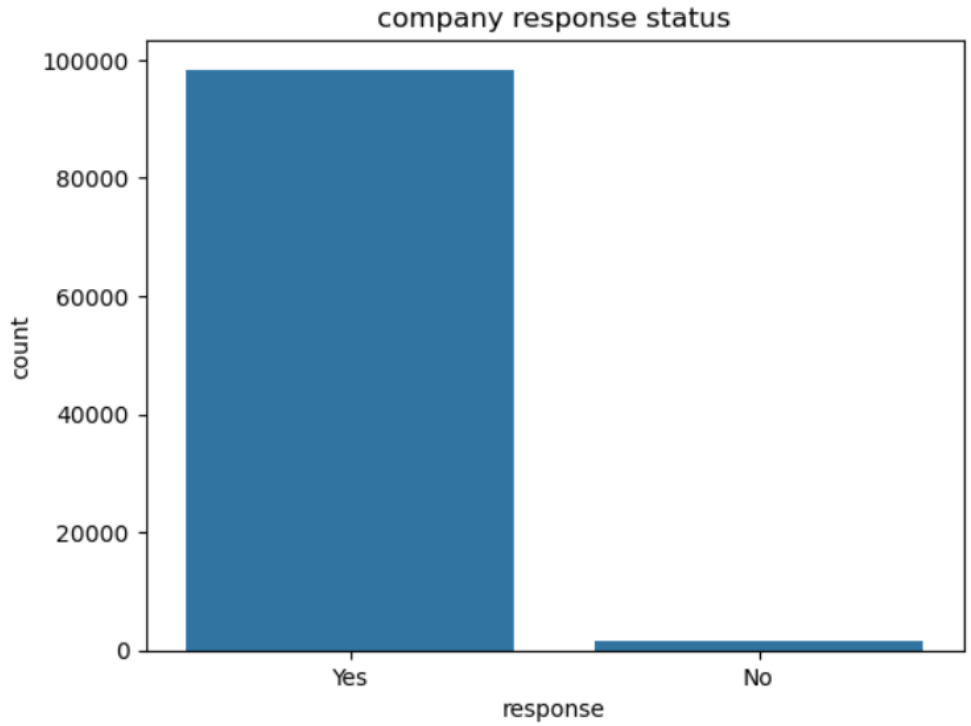


Bar-chat chart for complaints by product

```
sns.countplot(x='Submitted',data=file)
mat.title('complaint Submission Methods')
mat.show()
```

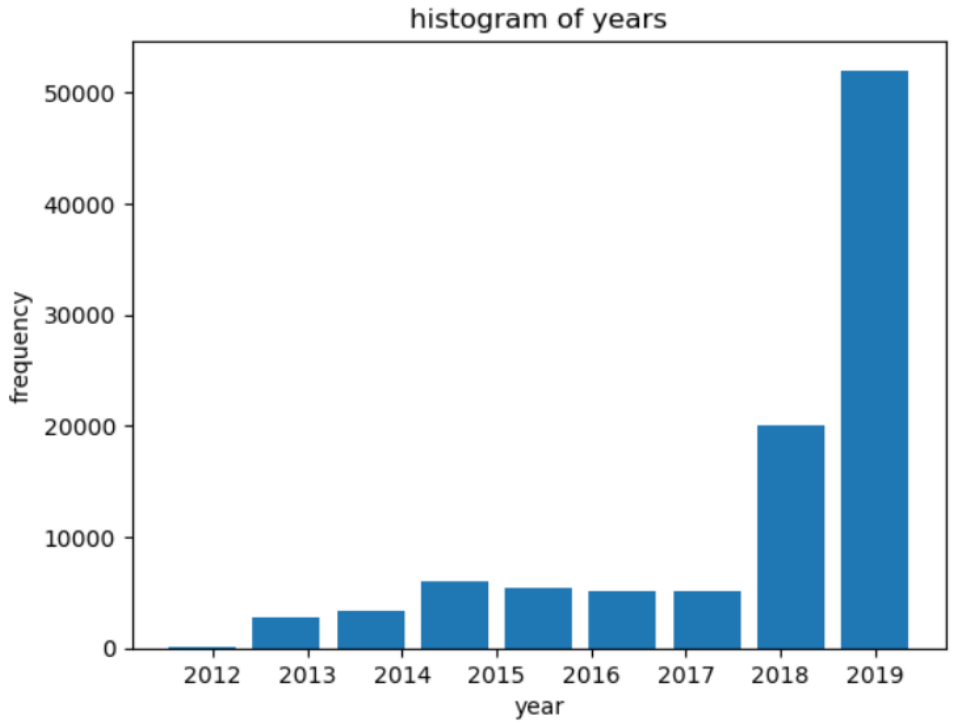


```
sns.countplot(x='response',data=file)
mat.title('company response status')
mat.show()
```



- Company response status counts .

```
mat.hist(file['year'],bins=len(file['year'].unique()),align='right',rwidth=0.8)
mat.xlabel('year')
mat.ylabel('frequency')
mat.title('histogram of years')
mat.show()
```



- Year wise complaint counts.

MYSQL QUERIES

TABE DESIGN:

Field	Type	Null	Key	Default	Extra
ComplaintID	int	YES		NULL	
datereceived	text	YES		NULL	
Product	text	YES		NULL	
Subproduct	text	YES		NULL	
Issue	text	YES		NULL	
Subissue	text	YES		NULL	
Company	text	YES		NULL	
State	text	YES		NULL	
ZIPcode	text	YES		NULL	
consent	text	YES		NULL	
Submitted	text	YES		NULL	
Datesent	text	YES		NULL	
response	text	YES		NULL	
consumerdis...	text	YES		NULL	

SQL QUERIES:

SELECT- It fetches the data in the table using the select query.

```
Select*from banking;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

ComplaintID	datereceived	Product	Subproduct	Issue	Subissue
2738619	27-11-2017	Mortgage	Conventional home mortgage	Trouble during payment process	
2933849	12-06-2018	Mortgage	Conventional home mortgage	Trouble during payment process	
1165653	21-12-2014	Mortgage	Conventional fixed mortgage	Loan modification,collection,foreclosure	
3316943	24-07-2019	Credit reporting, credit repair services, or other...	Credit reporting	Incorrect information on your report	Information belongs to someone els
3157550	20-02-2019	Payday loan, title loan, or personal loan	Installment loan	Struggling to pay your loan	
3068834	07-11-2018	Mortgage	Conventional home mortgage	Trouble during payment process	
1236640	10-02-2015	Mortgage	Other mortgage	Loan modification,collection,foreclosure	
1249267	20-02-2015	Mortgage	Conventional fixed mortgage	Loan servicing, payments, escrow account	
538930	23-09-2013	Mortgage	Conventional fixed mortgage	Loan servicing, payments, escrow account	
625914	09-12-2013	Mortgage	Conventional fixed mortgage	Loan modification,collection,foreclosure	

- It fetches all the columns from banking table.

WHERE- return the row where the data is.

- Select * from banking where product =’mortgage’;

```
select*from banking where product ='mortgage';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

ComplaintID	datereceived	Product	Subproduct	Issue	Subissue	Company
2738619	27-11-2017	Mortgage	Conventional home mortgage	Trouble during payment process		NATIONSTAR MORTGAGE LLC
2933849	12-06-2018	Mortgage	Conventional home mortgage	Trouble during payment process		NATIONSTAR MORTGAGE LLC
1165653	21-12-2014	Mortgage	Conventional fixed mortgage	Loan modification,collection,foreclosure		NATIONSTAR MORTGAGE LLC
3068834	07-11-2018	Mortgage	Conventional home mortgage	Trouble during payment process		Community Loan Servicing, LLC (formerly known...
1236640	10-02-2015	Mortgage	Other mortgage	Loan modification,collection,foreclosure		NATIONSTAR MORTGAGE LLC
1249267	20-02-2015	Mortgage	Conventional fixed mortgage	Loan servicing, payments, escrow account		NATIONSTAR MORTGAGE LLC
538930	23-09-2013	Mortgage	Conventional fixed mortgage	Loan servicing, payments, escrow account		NATIONSTAR MORTGAGE LLC
625914	09-12-2013	Mortgage	Conventional fixed mortgage	Loan modification,collection,foreclosure		NATIONSTAR MORTGAGE LLC
563802	19-10-2013	Mortgage	Conventional adjustable mortgage (ARM)	Loan modification,collection,foreclosure		NATIONSTAR MORTGAGE LLC
164306	01-10-2012	Mortgage	FHA mortgage	Loan servicing, payments, escrow account		NATIONSTAR MORTGAGE LLC

- Returns the rows where product is mortgage

DISTINCT-To get the unique value.

Select distinct product from banking;

- It retrieves the unique product

COUNT- The count is get the count of each row

- Select count (product) from banking;

Result Grid | Filter Rows:

product
Mortgage
Credit reporting, credit repair services, or other...
Payday loan, title loan, or personal loan
Debt collection
Checking or savings account
Bank account or service
Student loan
Vehicle loan or lease
Consumer Loan
Credit card or prepaid card
Credit reporting

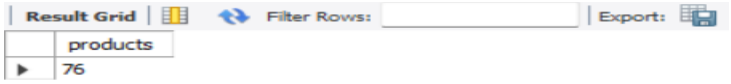
Result Grid | Filter Rows: | Export:

count(product)
102495

- It returns the count of the dataset.

COUNT &DISTINCT- I used for unique values count.

🔍 Select count (distinct subproduct) as products from banking;



products
76

❖ It retrieves unique sub-products count.

ORDER BY- To get the values in the respective order.

🔍 Select * from banking order by datereceived;

```
24 •  select * from banking order by datereceived; --  ascending order of date
```

ComplaintID	datereceived	Product	Subproduct	Issue	Subissue	Company
9027	01-01-2012	Mortgage	FHA mortgage	Application, originator, mortgage broker		BANK OF AMERICA, NATIONAL ASSOCIATI
9002	01-01-2012	Credit card		Payoff process		CITIBANK, N.A.
219469	01-01-2013	Bank account or service	Checking account	Making/receiving payments, sending money		BANK OF AMERICA, NATIONAL ASSOCIATI
221163	01-01-2013	Mortgage	Reverse mortgage	Loan modification,collection,foreclosure		WELLS FARGO & COMPANY
219470	01-01-2013	Mortgage	Other mortgage	Loan servicing, payments, escrow account		REGIONS FINANCIAL CORPORATION
216178	01-01-2013	Bank account or service	Checking account	Problems caused by my funds being low		BANK OF AMERICA, NATIONAL ASSOCIATI
221179	01-01-2013	Consumer Loan	Vehicle loan	Managing the loan or lease		TOYOTA MOTOR CREDIT CORPORATION
215978	01-01-2013	Credit reporting		Incorrect information on credit report	Information is not mine	Experian Information Solutions Inc.
219453	01-01-2013	Mortgage	Other mortgage	Loan modification,collection,foreclosure		WELLS FARGO & COMPANY
219466	01-01-2013	Mortgage	Other mortgage	Loan modification,collection,foreclosure		NATIONSTAR MORTGAGE LLC

❖ It retrieves the datereceived columns in ascending order.

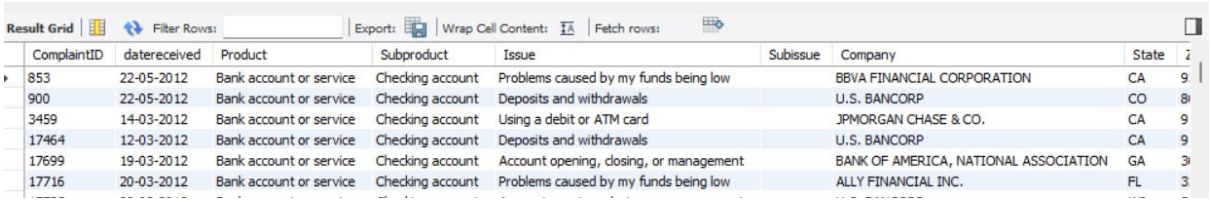
🔍 Select * from banking order by datereceived desc;

```
25 •  select * from banking order by datereceived desc; -- descending order
```

ComplaintID	datereceived	Product	Subproduct	Issue	Subissue	Company
9027	01-01-2012	Mortgage	FHA mortgage	Application, originator, mortgage broker		BANK OF AMERICA, NATIONAL ASSOCIATI
9002	01-01-2012	Credit card		Payoff process		CITIBANK, N.A.
219469	01-01-2013	Bank account or service	Checking account	Making/receiving payments, sending money		BANK OF AMERICA, NATIONAL ASSOCIATI
221163	01-01-2013	Mortgage	Reverse mortgage	Loan modification,collection,foreclosure		WELLS FARGO & COMPANY
219470	01-01-2013	Mortgage	Other mortgage	Loan servicing, payments, escrow account		REGIONS FINANCIAL CORPORATION
216178	01-01-2013	Bank account or service	Checking account	Problems caused by my funds being low		BANK OF AMERICA, NATIONAL ASSOCIATI
221179	01-01-2013	Consumer Loan	Vehicle loan	Managing the loan or lease		TOYOTA MOTOR CREDIT CORPORATION
215978	01-01-2013	Credit reporting		Incorrect information on credit report	Information is not mine	Experian Information Solutions Inc.
219453	01-01-2013	Mortgage	Other mortgage	Loan modification,collection,foreclosure		WELLS FARGO & COMPANY
219466	01-01-2013	Mortgage	Other mortgage	Loan modification,collection,foreclosure		NATIONSTAR MORTGAGE LLC

❖ It retrieves the datereceived columns in descending order.

🔍 Select*from where subproduct='checking account' order by complaintID;



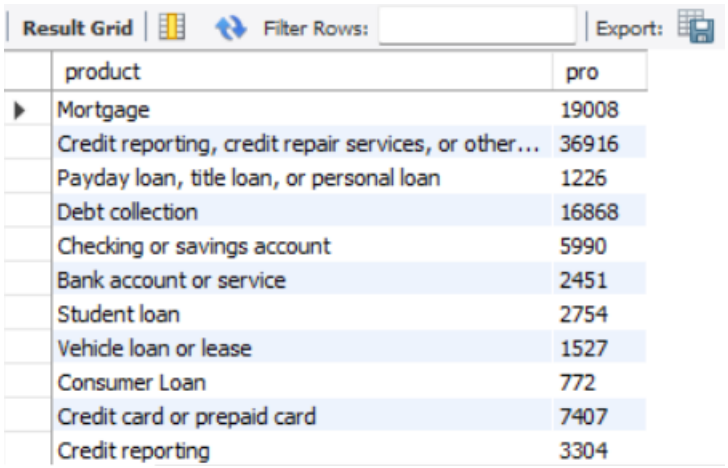
Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

ComplaintID	datereceived	Product	Subproduct	Issue	Subissue	Company	State
853	22-05-2012	Bank account or service	Checking account	Problems caused by my funds being low		BBVA FINANCIAL CORPORATION	CA
900	22-05-2012	Bank account or service	Checking account	Deposits and withdrawals		U.S. BANCORP	CO
3459	14-03-2012	Bank account or service	Checking account	Using a debit or ATM card		JPMORGAN CHASE & CO.	CA
17464	12-03-2012	Bank account or service	Checking account	Deposits and withdrawals		U.S. BANCORP	CA
17699	19-03-2012	Bank account or service	Checking account	Account opening, closing, or management		BANK OF AMERICA, NATIONAL ASSOCIATION	GA
17716	20-03-2012	Bank account or service	Checking account	Problems caused by my funds being low		ALLY FINANCIAL INC.	FL

- It retrieves the sub-product is checking account based on the order of complain-ID.

GROUP BY- The group by is counts the sub-products in each products

- Select product ,count(sub-product) as pro from banking group by product;



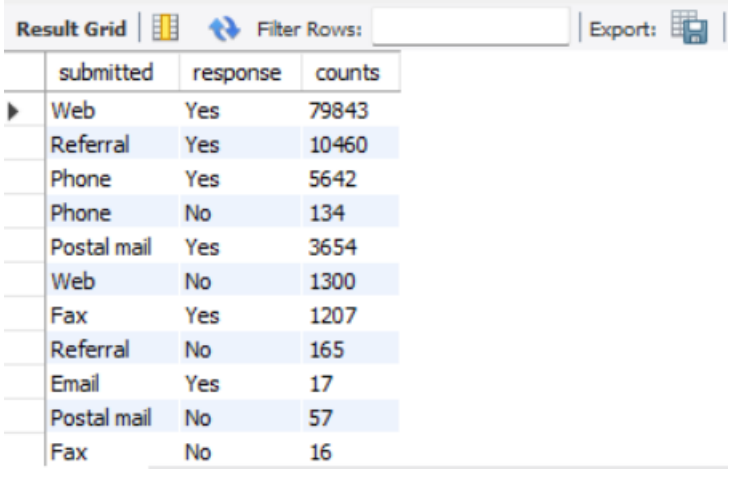
Result Grid | Filter Rows: | Export:

product	pro
Mortgage	19008
Credit reporting, credit repair services, or other...	36916
Payday loan, title loan, or personal loan	1226
Debt collection	16868
Checking or savings account	5990
Bank account or service	2451
Student loan	2754
Vehicle loan or lease	1527
Consumer Loan	772
Credit card or prepaid card	7407
Credit reporting	3304

- It retrieves the counts Sub-products in each product.

MULTIPLE COLUMNS IN GROUP BY:

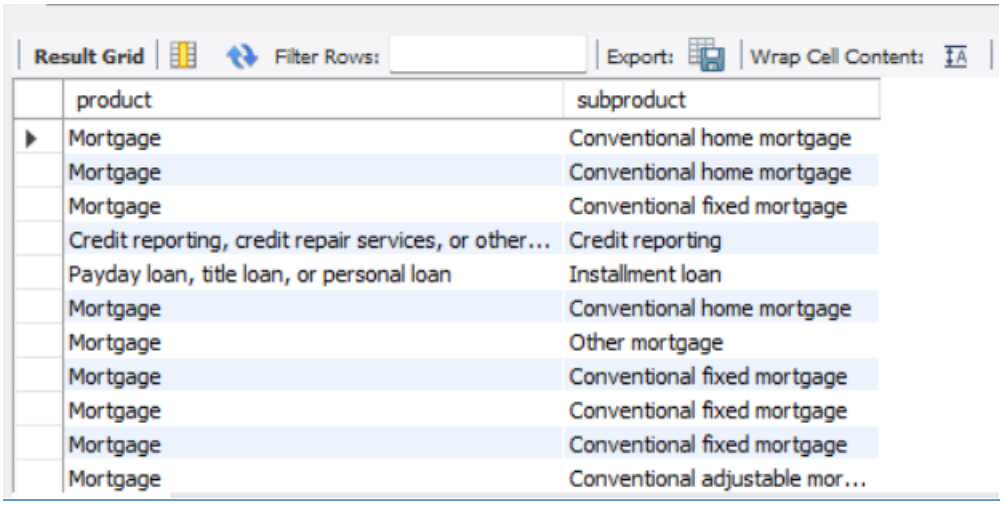
- Select submitted, response, count(*) as counts from banking group by submitted, response;



Result Grid | Filter Rows: | Export:

submitted	response	counts
Web	Yes	79843
Referral	Yes	10460
Phone	Yes	5642
Phone	No	134
Postal mail	Yes	3654
Web	No	1300
Fax	Yes	1207
Referral	No	165
Email	Yes	17
Postal mail	No	57
Fax	No	16

- It retrieves the count of the responses based on submitted.



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

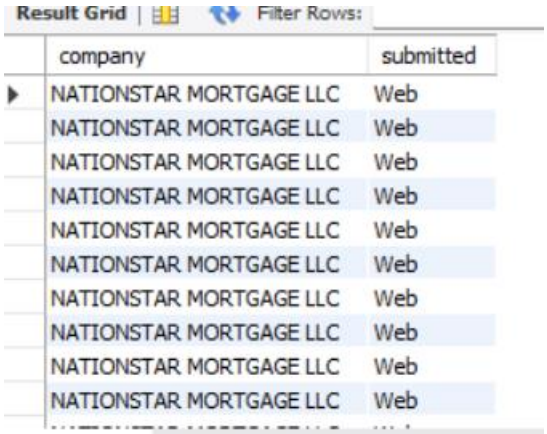
product	subproduct
Mortgage	Conventional home mortgage
Mortgage	Conventional home mortgage
Mortgage	Conventional fixed mortgage
Credit reporting, credit repair services, or other...	Credit reporting
Payday loan, title loan, or personal loan	Installment loan
Mortgage	Conventional home mortgage
Mortgage	Other mortgage
Mortgage	Conventional fixed mortgage
Mortgage	Conventional fixed mortgage
Mortgage	Conventional fixed mortgage
Mortgage	Conventional adjustable mor...

- It retrieves the values in Sub-product values.

LOGICAL OPERATORS:

AND-it filters rows where both conditions are true.

- Select company, submitted from banking where company='NATIONSTAR MORTGAGE LLC' and submitted='Web' ;



Result Grid | Filter Rows:

company	submitted
NATIONSTAR MORTGAGE LLC	Web
NATIONSTAR MORTGAGE LLC	Web
NATIONSTAR MORTGAGE LLC	Web
NATIONSTAR MORTGAGE LLC	Web
NATIONSTAR MORTGAGE LLC	Web
NATIONSTAR MORTGAGE LLC	Web
NATIONSTAR MORTGAGE LLC	Web
NATIONSTAR MORTGAGE LLC	Web
NATIONSTAR MORTGAGE LLC	Web
NATIONSTAR MORTGAGE LLC	Web

- It retrieves the company name and the complaint is submitted on web.

OR- it returns rows where the one condition is true.

- Select consumerdispute, companyresponse from banking where consumerdispute="yes" or companyresponse='closed with explanation';

Result Grid | Filter Rows:

consumerdispute	companyresponse
Unknown	Closed with explanation
Unknown	Closed with explanation
No	Closed with explanation
Unknown	Closed with explanation
Unknown	Closed with explanation
Unknown	Closed with explanation
Yes	Closed with explanation
Yes	Closed with explanation
No	Closed with explanation
No	Closed with explanation

- It retrieves the consumerdispute=yes or companyresponse=closed with explanation.

View

A view is like a virtual table.

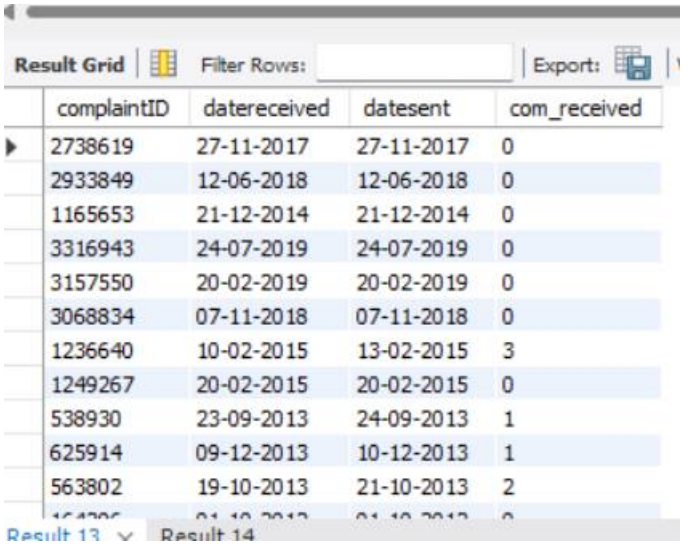
create view responses as select issue,subissue,company,companyresponse from banking where issue='Loan modification,collection,foreclosure';

🔍 select * from responses;

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

issue	subissue	company	companyresponse
Loan modification,collection,foreclosure		NATIONSTAR MORTGAGE LLC	Closed with explanation
Loan modification,collection,foreclosure		NATIONSTAR MORTGAGE LLC	Closed with explanation
Loan modification,collection,foreclosure		NATIONSTAR MORTGAGE LLC	Closed with explanation
Loan modification,collection,foreclosure		NATIONSTAR MORTGAGE LLC	Closed with explanation
Loan modification,collection,foreclosure		NATIONSTAR MORTGAGE LLC	Closed with explanation
Loan modification,collection,foreclosure		NATIONSTAR MORTGAGE LLC	Closed with explanation
Loan modification,collection,foreclosure		NATIONSTAR MORTGAGE LLC	Closed without relief
Loan modification,collection,foreclosure		NATIONSTAR MORTGAGE LLC	Closed with explanation
Loan modification,collection,foreclosure		NATIONSTAR MORTGAGE LLC	Closed with explanation
Loan modification,collection,foreclosure		NATIONSTAR MORTGAGE LLC	Closed with explanation
Loan modification,collection,foreclosure		NATIONSTAR MORTGAGE LLC	Closed with explanation

- Created responses table in banking which holds the issue,subissue , company and companyresponse.

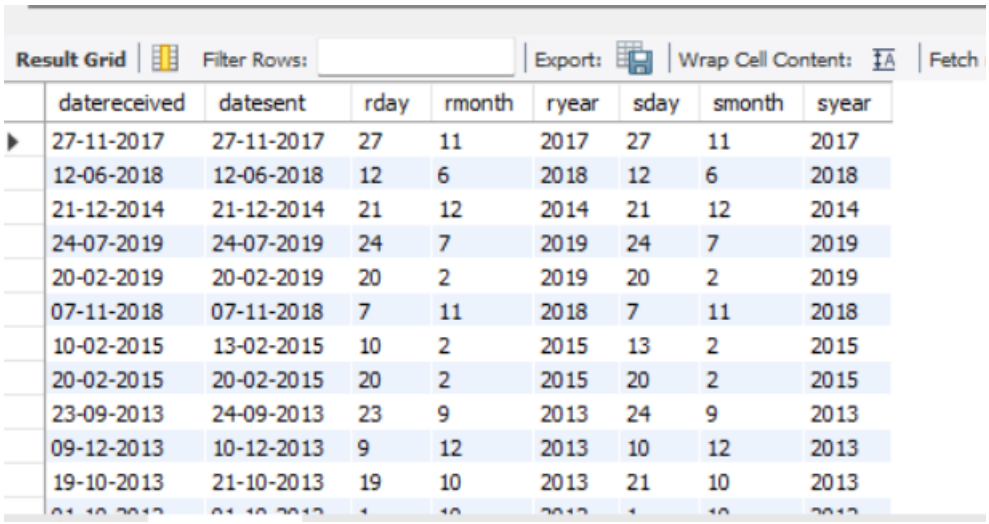


Result Grid | Filter Rows: | Export:

complaintID	datereceived	datesent	com_received
2738619	27-11-2017	27-11-2017	0
2933849	12-06-2018	12-06-2018	0
1165653	21-12-2014	21-12-2014	0
3316943	24-07-2019	24-07-2019	0
3157550	20-02-2019	20-02-2019	0
3068834	07-11-2018	07-11-2018	0
1236640	10-02-2015	13-02-2015	3
1249267	20-02-2015	20-02-2015	0
538930	23-09-2013	24-09-2013	1
625914	09-12-2013	10-12-2013	1
563802	19-10-2013	21-10-2013	2

Result 13 | Result 14

- Duration of the received and sentdate using the datedifference.



Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch

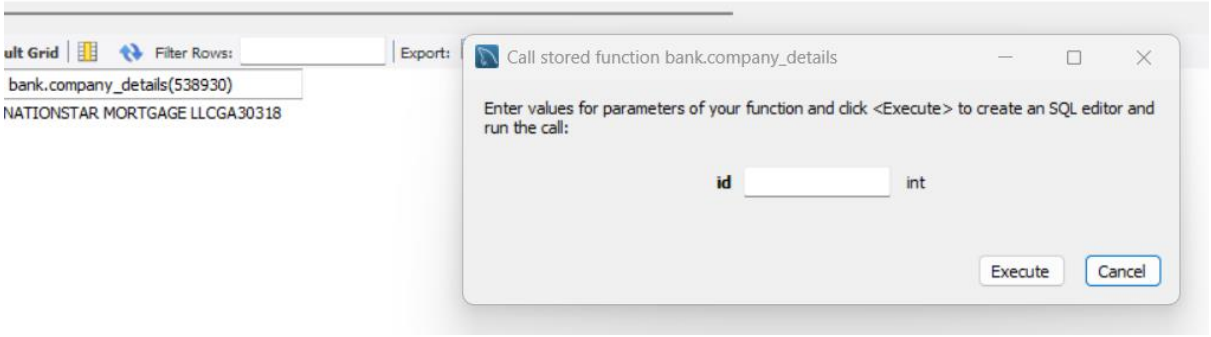
datereceived	datesent	rday	rmonth	ryear	sday	smonth	syear
27-11-2017	27-11-2017	27	11	2017	27	11	2017
12-06-2018	12-06-2018	12	6	2018	12	6	2018
21-12-2014	21-12-2014	21	12	2014	21	12	2014
24-07-2019	24-07-2019	24	7	2019	24	7	2019
20-02-2019	20-02-2019	20	2	2019	20	2	2019
07-11-2018	07-11-2018	7	11	2018	7	11	2018
10-02-2015	13-02-2015	10	2	2015	13	2	2015
20-02-2015	20-02-2015	20	2	2015	20	2	2015
23-09-2013	24-09-2013	23	9	2013	24	9	2013
09-12-2013	10-12-2013	9	12	2013	10	12	2013
19-10-2013	21-10-2013	19	10	2013	21	10	2013

FUCTION:

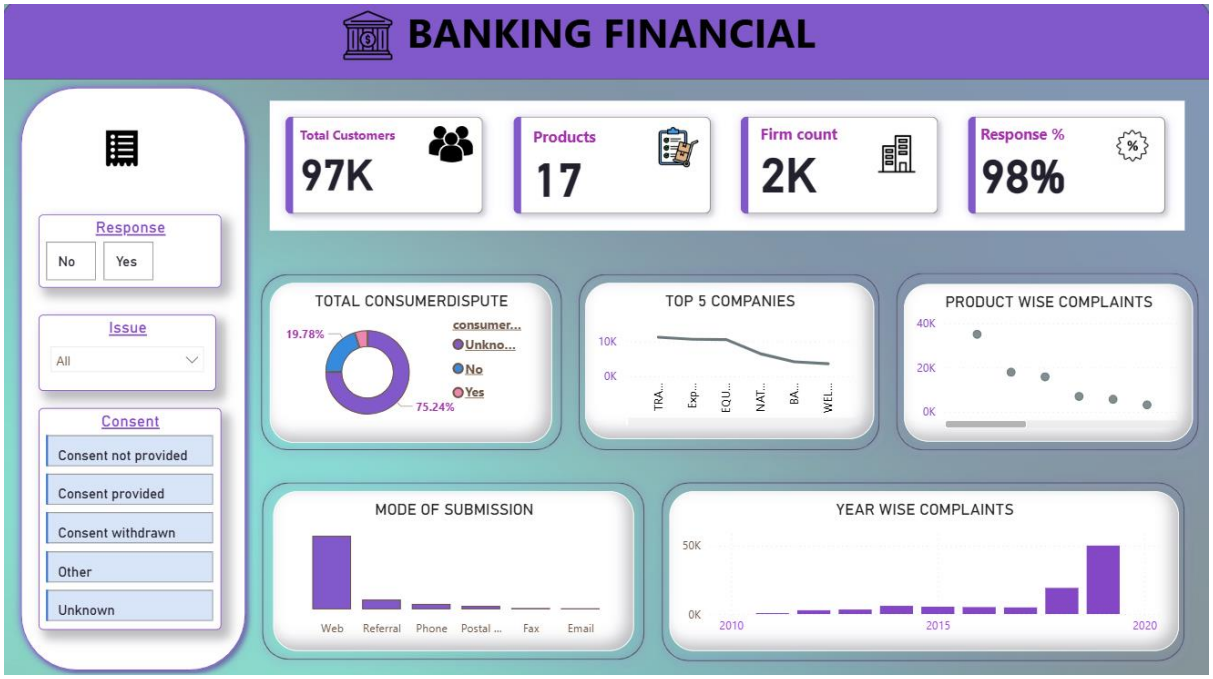
```
delimiter $$

create function company_details(id int)

returns varchar(50)

deterministic

begin

 declare details varchar(50);
```

```
select concat(company,state,ZIPcode) into details
from banking
where complaintID=id;
return details;
end $$
delimiter ;
```

```
select bank.company_details(538930);
```



POWER BI DASHBOARD



1. Overview

The **Banking Financial Dashboard** provides a comprehensive visual summary of customer data, product details, firm information, consumer disputes, and complaint trends. It is designed to help analysts and stakeholders monitor performance, identify patterns, and make data-driven decisions in the banking domain.

The dashboard uses **KPIs, charts, and filters** to offer an intuitive and interactive user experience.

2. Key Performance Indicators (KPIs)

Total Customers – 97K

- Shows the total number of customers associated with the financial institution. This helps measure customer base size and business reach.

Products – 17

- Indicates the number of financial products offered. It highlights product diversity and service range.

Firm Count – 2K

- Represents the number of firms or branches involved. Useful for understanding operational scale.

Response % – 98%

- Shows the percentage of customer responses or complaint resolutions.
  A high value indicates strong customer engagement and resolution efficiency.

3. Filter Panel (Left Side)

Response Filter

- Options: **Yes / No**
  Allows the user to segment complaints based on whether a company responded.

Issue Filter

- Dropdown filter to select specific issues for deeper analysis.

Consent Filter

- Includes categories like:

- Consent provided

- Consent not provided
- Consent withdrawn
- Other
- Unknown

These filters help analyze disputes based on customer consent status.

4. Visual Insights and Charts

A. Total Consumer Dispute (Donut Chart)

- Shows the distribution of dispute outcomes:

- Yes
- No
- Unknown
- Consumer disputes

Helps identify the proportion of resolved vs unresolved complaints.

B. Top 5 Companies (Line Chart)

- showed complaint data for top companies:

- **Line chart** shows trend across companies.
- **Dot plot** shows the complaint volume comparatively.

- Useful to understand which companies receive higher disputes and patterns.

C. Product wise complaints(DOT chart)

- **Dot plot** shows the product wise complaints volume comparatively.

D. Mode of Submission (Bar Chart)

- Displays how customers submitted complaints:

- Web
- Referral
- Phone
- Postal
- Fax
- Email

- Highlights the most commonly used channels (e.g., Web).

E. Year-Wise Complaints (Bar Chart)

- Shows the complaint volume across different years (2010 to 2020).
  Helps identify growth trends, spikes, or reductions in complaints over time.

5. Dashboard Design Features

- **Modern gradient theme** based on purple and teal shades.
- Minimalist icons for clarity.
- Rounded card layouts for readability.
- Interactive filters for quick segmentation.
- Well-spaced visualizations for better user experience.

6. Purpose of the Dashboard

- This dashboard supports:

- Monitoring customer complaints.s
- Identifying top problematic companies.
- Understanding communication channels.
- Reviewing yearly performance.

- Supporting compliance and service improvement strategies.

CONCLUSION:

The Banking Financial Analysis project successfully integrates data cleaning database management, and interactive visualization to provide a comprehensive understanding of customer complaints and financial operations. Through Python, the raw dataset was transformed into a clean, structured, and analysis-ready format. MySQL enabled efficient storage and retrieval of insights, allowing deeper exploration of trends, product behaviours, customer interactions, and operational patterns. The Power BI dashboard brought these insights to life with intuitive visuals and KPIs, helping stakeholders monitor performance, identify issues, and make informed decisions.

Overall, the project demonstrates strong capability in end-to-end data handling – from pre-processing and SQL analytics to professional dashboard creation. This solution can support banking teams in improving customer service, resolving disputes efficiently, and identifying opportunities for operational enhancement. It reflects a practical, industry-relevant data analytics workflow suitable for real-world banking environments.