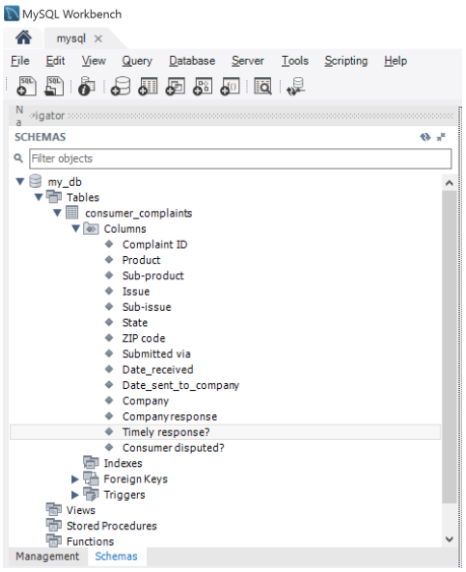
Executive Summary:

The data that was selected for analysis is from a consumer complaint database where customers complain about the financial products and services provided. The data file consist of 5533 record with 14 columns: (complaint ID, product, sub-product, issue, sub-issue, state, zip code, Submitted via, Date received, Date sent to company, Company, Company response, Timely response? And Consumer disputed?) between the years 2011 to 2015. After analyzing the data, there are 11 product categories with a wide verity of sub-products related to them. The top three product having a high number of complaints are (Credit Reporting, Debt collection and mortgage). Additional analysis on the "debt collection" products show that other(phone, health, club, etc..) had the a number of 407 complains and that's more than the credit card that had only 246 complains. Furthermore, analyzing the top ten companies based on the number of complains, Equifax took the first place, where Experian came second, after that TransUnion, Bank of America, Wells Fargo, JPMorgan Chase, Ocwen, Citibank and Nationstar Mortgage was the least. The complaints were filed from 58 states where the highest were from California with 773 complains and Florida with 585 complains. The response time for the complains pretty well. The data showed only 32 from 5533 that did not respond on time.

Analytical Process:

The spreadsheet was uploaded to MySQL Workbench in a table called consumer_complains in the database my_db through the table data input wizard for further analyses. This document will illustrate the queries used to analyze and explore the data, in addition to screen shots that will show the output result of each query. The main queries that were used were SELECT, FROM, WHERE, ORDER BY, GROUP BY, HAVING, LIMIT and CASE. Other cluses where used in the queries to manipulate and show more details from the data as DISTINCT and COUNT.



[redacted]

Appendix:

The first query executed was SELECT * FROM my_db.consumer_complaints. The output showed the data in all columns of the table.

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

Complaint ID	Product	Sub-product	Issue	Sub-issue	State	ZIP code	Submitted via
1431865	Consumer loan	Vehicle loan	Managing the loan or lease		NJ	8736	Web
1431374	Debt collection	Medical	Disclosure verification of debt	Not given enough info to verify debt	WI	54140	Web
1431251	Mortgage	Conventional fixed mortgage	Loan modification,collection,foreclosure		MO	63368	Web
1431743	Debt collection	Medical	Cont'd attempts collect debt not owed	Debt is not mine	WA	98055	Web
1432678	Debt collection	Medical	Cont'd attempts collect debt not owed	Debt was paid	TX	75104	Web
1432104	Debt collection	Other (phone, health club, etc.)	Cont'd attempts collect debt not owed	Debt was paid	CA	95423	Web
1431998	Debt collection	Other (phone, health club, etc.)	Communication tactics	Frequent or repeated calls	TX	75048	Web
1432207	Debt collection	Medical	Disclosure verification of debt	Right to dispute notice not received	TX	75125	Web
1432202	Consumer loan	Vehicle loan	Problems when you are unable to pay		TN	37174	Web
1432334	Debt collection	Other (phone, health club, etc.)	Disclosure verification of debt	Right to dispute notice not received	FL	33830	Web
1431188	Debt collection	Payday loan	Communication tactics	Threatened to take legal action	NE	68134	Web
1431358	Money transfers	Domestic (US) money transfer	Money was not available when promised		AL	35235	Phone
1430973	Debt collection	Other (phone, health club, etc.)	Cont'd attempts collect debt not owed	Debt was paid	OH	44057	Web
1431167	Debt collection		Cont'd attempts collect debt not owed	Debt is not mine	CO	80015	Web
1430909	Credit reporting		Unable to get credit report/credit score	Problem getting report or credit score	CA	95823	Web

Furthermore, to organize the data ascendingly by complaint ID; the following query was used.

```
SELECT *

FROM my_db.consumer_complaints

ORDER BY `Complaint ID` ASC
```

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

Complaint ID	Product	Sub-product	Issue	Sub-issue	State	ZIP code	Submitted via
1399395	Credit reporting		Incorrect information on credit report	Account status	MO	63031	Postal mail
1399397	Bank account or service	Checking account	Account opening, closing, or management		GA	30354	Web
1399398	Mortgage	Conventional adjustable mortgage (ARM)	Loan modification,collection,foreclosure		CA	94505	Web
1399400	Mortgage	FHA mortgage	Loan modification,collection,foreclosure		NV	0	Web
1399401	Debt collection	Other (phone, health club, etc.)	Cont'd attempts collect debt not owed	Debt is not mine	CT	6710	Web
1399407	Student loan	Non-federal student loan	Can't repay my loan		WI	53589	Web
1399409	Credit card		Other		IL	60638	Web
1399410	Mortgage	Conventional adjustable mortgage (ARM)	Loan modification,collection,foreclosure		CO	80124	Web
1399411	Mortgage	Other mortgage	Settlement process and costs		NC	28212	Web
1399418	Mortgage	Other mortgage	Loan modification,collection,foreclosure		NV	89117	Postal mail
1399422	Credit card		Identity theft / Fraud / Embezzlement		MA	1776	Web
1399423	Mortgage	Other mortgage	Loan modification,collection,foreclosure		IN	46229	Web
1399427	Mortgage	Other mortgage	Other		GA	30340	Fax
1399428	Mortgage	Other mortgage	Other		GA	30340	Fax
1399429	Debt collection		Cont'd attempts collect debt not owed	Debt is not mine	VA	22031	Web

[redacted]

After checking the general data, I wanted to see the different types of products in the product column using the DISTINCT query as:

```
SELECT DISTINCT product

FROM my_db.consumer_complaints
```

This showed the distinct products listed in the column without repetition.

Result Grid | Filter R

product
Consumer loan
Debt collection
Mortgage
Money transfers
Credit reporting
Bank account or service
Payday loan
Credit card
Student loan
Prepaid card
Other financial service

Moreover, counting the distinct product by using the count function and naming the column count_product.

```
SELECT COUNT(DISTINCT product) AS count_product

FROM my_db.consumer_complaints
```

count_product
11

Thus, there are 11 different product types in the product column.

Furthermore, using the below query shows the number of complaints in each product from the highest to the lowest.

```
SELECT DISTINCT product , COUNT(`Sub-product`) AS no_complains

FROM my_db.consumer_complaints

GROUP BY product

ORDER BY no_complains DESC
```

Result Grid | Filter Rows:

product	no_complains
Credit reporting	1346
Debt collection	1344
Mortgage	1312
Bank account or service	540
Credit card	500
Consumer loan	216
Student loan	144
Payday loan	51
Money transfers	49
Prepaid card	27
Other financial service	4

[redacted]

The next query was to demonstrate a list of consumer complains that are related to only one product which is the debt collection product and to check the sub-products that are related to the debt collection complains:

```
SELECT `Complaint ID`, Product, `Sub-product`

FROM my_db.consumer_complaints

WHERE product = "debt collection"
```

The result of the query returned 1344 rows which are the total rows that are related to the “debt collection” product and showing next to it is the different sub-product. The screen shot provided gives a sample of the result.

Result Grid | Filter Rows: | Export: | Wr

Complaint ID	Product	Sub-product
1431374	Debt collection	Medical
1431743	Debt collection	Medical
1432678	Debt collection	Medical
1432104	Debt collection	Other (phone, health club, etc.)
1431998	Debt collection	Other (phone, health club, etc.)
1432207	Debt collection	Medical
1432334	Debt collection	Other (phone, health club, etc.)
1431188	Debt collection	Payday loan
1430973	Debt collection	Other (phone, health club, etc.)
1431167	Debt collection	
1430968	Debt collection	
1431161	Debt collection	Other (phone, health club, etc.)
1431020	Debt collection	Other (phone, health club, etc.)
1430875	Debt collection	
1428774	Debt collection	Other (phone, health club, etc.)
1428739	Debt collection	Other (phone, health club, etc.)

consumer_complaints 8

Additionally, to be more precise I wrote a more detailed query to illustration how many times the different sub-products were related to the debt collection complain organizing them form the most frequent to the least.

```
SELECT Product, `Sub-product`, COUNT(`Sub-product`) AS no_complains

FROM my_db.consumer_complaints

WHERE product = "debt collection"

GROUP BY `Sub-product`

ORDER BY no_complains DESC
```

The output showed the number of recurrence of each sub-products related to the debt collection complain.

Result Grid | Filter Rows: | Export:

Product	Sub-product	no_complains
Debt collection	Other (phone, health club, etc.)	407
Debt collection		296
Debt collection	Credit card	246
Debt collection	Medical	235
Debt collection	Payday loan	64
Debt collection	Auto	36
Debt collection	Mortgage	25
Debt collection	Non-federal student loan	18
Debt collection	Federal student loan	17

[redacted]

Another query was implemented to check the complaint ID and the date received for the most recent 50 complains of the credit reporting product.

```
SELECT `Complaint ID`, product, `Date received`

FROM my_db.consumer_complaints

WHERE product = "credit reporting"

LIMIT 50
```

The query did return precisely the most recent 50 rows that include “credit reporting” product. The screen shot provided gives a sample of the result.

Result Grid | Filter Rows:

Complaint ID	product	Date_received
1430909	Credit reporting	6/21/2015
1430879	Credit reporting	6/21/2015
1430955	Credit reporting	6/21/2015
1430945	Credit reporting	6/21/2015
1430916	Credit reporting	6/21/2015
1431060	Credit reporting	6/21/2015
1428766	Credit reporting	6/20/2015
1428738	Credit reporting	6/20/2015
1428665	Credit reporting	6/20/2015
1428786	Credit reporting	6/20/2015
1428835	Credit reporting	6/20/2015
1428850	Credit reporting	6/20/2015
1428632	Credit reporting	6/20/2015
1428719	Credit reporting	6/19/2015
1428715	Credit reporting	6/19/2015

consumer_complaints 14

18 12:56:17 SELECT `Complaint ID`, product, Date_received FROM my_db.consumer_complaints WHERE product = "credit reporting" LIMIT 50 50 row(s) returned 0.000 sec / 0.000 sec

There is a column called submitted via which is the media used by the customer to file the claim of a certain product. By using the query below it distinguishes the different types of products and the media used - with its counts - to submit the claim.

```
SELECT product, `Submitted via`, COUNT(`Submitted via`) AS count_via

FROM my_db.consumer_complaints

GROUP BY product

ORDER BY count_via
```

The output shows the most media type used for each product and the number of times this certain media was used to complain about a specific product.

Result Grid | Filter Rows:

product	Submitted via	count_via
Other financial service	Referral	4
Prepaid card	Web	27
Money transfers	Phone	49
Payday loan	Web	51
Student loan	Web	144
Consumer loan	Web	216
Credit card	Web	500
Bank account or service	Web	540
Mortgage	Web	1312
Debt collection	Web	1344
Credit reporting	Web	1346

[redacted]

Moving on with analyzing different data in the consumer_complaints table. We can use a query to check the number of complains each company got in the period of 2011 to 2015. The output of the query will lists the different companies in the company column with the number of consumer complains each one got from the most to the least.

```
SELECT DISTINCT Company, COUNT(`Complaint ID`) as complains

FROM my_db.consumer_complaints

GROUP BY company

ORDER BY complains DESC
```

The result of the query returned 634 rows which are the total number of companies in the company column and the number of complains each one got .The screen shot provided gives a sample of the result.

Result Grid | Filter Rows:

Company	complains
Equifax	554
Experian	403
TransUnion	353
Bank of America	313
Wells Fargo	293
JPMorgan Chase	259
Ocwen	199
Citibank	174
Nationstar Mortgage	137
Enhanced Recovery Company, LLC	96
GE Capital Retail	96
U.S. Bancorp	86
Capital One	83
Green Tree Servicing, LLC	72
Transworld Systems Inc.	71
Portfolio Recovery Associates, Inc.	59
Navient	57
Encore Capital Group	56
TD Bank	47
Amex	46

Result 25

29 13:14:48 SELECT DISTINCT Company, COUNT(`Complaint ID`) as complains FROM my_db.consumer_complaints GROUP BY company ORDER BY complains DESC LIMIT 0, 50000 634 row(s) returned 0.094 sec / 0.000 sec

To get more specific and show only the companies that have complaints more than a 100 in a descending order.

```
SELECT DISTINCT Company, COUNT(`Complaint ID`) AS complains

FROM my_db.consumer_complaints

GROUP BY company

HAVING complains > 100

ORDER BY complains DESC
```

The result showed 9 rows illustrating only the companies that had more than 100 complains in the whole data set.

Result Grid | Filter Rows:

Company	complains
Equifax	554
Experian	403
TransUnion	353
Bank of America	313
Wells Fargo	293
JPMorgan Chase	259
Ocwen	199
Citibank	174
Nationstar Mortgage	137

[redacted]

Using the product and the state column, a query was executed to show the number of complains in each state.

```
SELECT DISTINCT state, COUNT(product) AS no_complains

FROM my_db.consumer_complaints

WHERE state <> " "

GROUP BY state

ORDER BY state
```

The result of the query returned 58 rows which represents the total state in the data set with the number of complains each one filed. The screen shot provided gives a sample of the result.

Result Grid | Filter

state	no_complains
AE	4
AK	10
AL	70
AP	1
AR	19
AZ	133
CA	773
CO	86
CT	64
DC	30
DE	38
FL	585
FM	1
GA	248
GU	2
HI	10
IA	24
ID	13
IL	216
IN	59

Result 27

38 13:40:07 SELECT DISTINCT state, COUNT(product) as no_complains FROM my_db.consumer_complaints WHERE state <> " " GROUP BY state ORDER BY state LIMIT 0, 50000 58 row(s) retur... 0.094 sec / 0.000 sec

To get deeper in the data, we can filter the previous query to check the states that have more than 100 complains ordered by the most to the least number of complaints.

```
SELECT DISTINCT state, COUNT(product) AS no_complains

FROM my_db.consumer_complaints

WHERE state <> " "

GROUP BY state

HAVING no_complains > 100

ORDER BY  no_complains DESC
```

The output demonstrate 16 rows which are the exact number of state that have more than 100 complains and it also shows the specific number for each.

Result Grid | Filter

state	no_complains
CA	773
FL	585
TX	410
NY	401
GA	248
IL	216
NJ	202
OH	188
PA	175
VA	172
MD	167
NC	160
AZ	133
WA	119
MI	115
MA	106

Result 34

[redacted]

To further understand and analyze the data set more using the case function in determining if the number of complaints per state is good, alertly or bad.

```
SELECT DISTINCT state, COUNT(product) AS no_complains ,
CASE
WHEN COUNT(product) < 100 THEN "Good"
WHEN COUNT(product) BETWEEN 100 AND 500  THEN "Alert"
WHEN COUNT(product) >= 500 THEN "Bad"
END AS status
FROM my_db.consumer_complaints
WHERE state <> " "
GROUP BY state
ORDER BY no_complains DESC
```

Result Grid | Filter Rows:

state	no_complains	status
CA	773	Bad
FL	585	Bad
TX	410	Alert
NY	401	Alert
GA	248	Alert
IL	216	Alert
NJ	202	Alert
OH	188	Alert
PA	175	Alert
VA	172	Alert
MD	167	Alert
NC	160	Alert
AZ	133	Alert
WA	119	Alert
MI	115	Alert
MA	106	Alert
TN	94	Good
CO	86	Good
SC	81	Good
LA	72	Good

Result 36

43 14:13:12 SELECT DISTINCT state, COUNT(product) as no_complains , CASE WHEN COUNT(product) < 100 THEN "Good" WHEN COUNT(product) BETWEEN 100 AND 500 ... 58 row(s) returned 0.078 sec / 0.000 sec

The result displayed 58 rows which are the total states in the data set. It is shown that 2 states are in a “bad” status, 14 are in “alert”, whereas the rest are in a “good” status.

We can also see the top 10 states and the number of complains of a specific sub product which is the vehicle loan and the date received from the most complaint state to the least.

```
SELECT state, `Sub-product`, Date_received, COUNT(`Sub-product`)
AS complain
FROM my_db.consumer_complaints
WHERE `Sub-product`= "Vehicle loan"
GROUP BY state
ORDER BY complain DESC
LIMIT 10
```

Result Grid | Filter Rows:

state	Sub-product	Date_received	complain
CA	Vehicle loan	6/19/2015	14
TX	Vehicle loan	6/15/2015	10
NJ	Vehicle loan	6/22/2015	8
FL	Vehicle loan	6/18/2015	8
NC	Vehicle loan	6/14/2015	7
OH	Vehicle loan	6/8/2015	7
NY	Vehicle loan	6/12/2015	6
IL	Vehicle loan	6/19/2015	5
AR	Vehicle loan	6/10/2015	5
SC	Vehicle loan	6/18/2015	4

The result returned 10 rows that illustrate most recent complains for the "Vehicle loan" product in 10 different states with the number of complain in each in a descending order.

[redacted]

Another detailed query that will display the companies that still have some complains “in progress” in the month of June 2015.

```
SELECT company, product, `Company response`, Date_sent_to_company
FROM my_db.consumer_complaints
WHERE `Company response` = "in progress" and Date_sent_to_company >= "6/01/2015"
```

The query returned 1395 rows, which are the” in progress” complains in the in the month of June and the representative company of each product complain.

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

company	product	Company response	Date_sent_to_company
Nissan Motor Acceptance Corporation	Consumer loan	In progress	6/22/2015
Flagstar Bank	Mortgage	In progress	6/22/2015
Associated Credit Services, Inc.	Debt collection	In progress	6/22/2015
Navy FCU	Consumer loan	In progress	6/22/2015
MoneyGram	Money transfers	In progress	6/22/2015
Southwest Credit Systems, L.P.	Debt collection	In progress	6/22/2015
Arbor Residential Mortgage LLC	Mortgage	In progress	6/21/2015
Transworld Systems Inc.	Debt collection	In progress	6/21/2015
Equifax	Credit reporting	In progress	6/22/2015
Flagstar Bank	Mortgage	In progress	6/21/2015
Rozlin Financial Group, Inc.	Debt collection	In progress	6/20/2015
Transworld Systems Inc.	Debt collection	In progress	6/20/2015
Equifax	Credit reporting	In progress	6/20/2015
Expert Global Solutions, Inc.	Debt collection	In progress	6/20/2015
Credit Corp Solutions Inc.	Debt collection	In progress	6/19/2015
Equifax	Credit reporting	In progress	6/20/2015
Equifax	Credit reporting	In progress	6/19/2015
Equifax	Credit reporting	In progress	6/19/2015
Navy FCU	Consumer loan	In progress	6/22/2015

consumer_complaints 40

52 14:54:38 SELECT company, product, `Company response`, Date_sent_to_company FROM my_db.consumer_complaints WHERE `Company response` = "in progress" and Date_... 1395 row(s) returned 0.016 sec / 0.015 sec

The timely response column indicates if the complaint was solved on time or not. Thus, we can execute a query to see which companies did not respond on time for certain sub-products.

```
SELECT Company, `Sub-product`, `Company response`, `Timely response?`
FROM my_db.consumer_complaints
WHERE `Timely response?` = "no" and `Sub-product` <> " "
ORDER BY `Sub-product`
```

The result returned 32 rows that show the complaints on the sub-product where the companies did not respond on time. The data is order alphabetically by the sub-product.

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

Company	Sub-product	Company response	Timely response?
CCS Financial Services, Inc.	Auto	Untimely response	No
GM Financial	Auto	Untimely response	No
Midpoint Law Group, P.C	Conventional fixed mortgage	Untimely response	No
Livingston Financial, LLC	Credit card	Untimely response	No
CCS Financial Services, Inc.	Credit card	Untimely response	No
Resurgence Legal Group, PC	Credit card	Untimely response	No
OneWest Bank	Home equity loan or line of credit	Closed with explanation	No
West Corporation	Medical	In progress	No
Thunderbird Collection Specialists, Inc.	Medical	Untimely response	No
Credit Collections U.S.A., L.L.C.	Medical	Untimely response	No
ALCHRO INC.	Medical	Untimely response	No
Medical Data Systems, Inc.	Medical	Closed with explanation	No
Overton, Russell, Doerr and Donovan...	Medical	Untimely response	No
Harvard Collections, LLC	Medical	Closed with explanation	No
Professiona Professional Accounts Service, Inc. (Indiana)		Untimely response	No
Broward Adjustment Services, Inc.	Medical	Untimely response	No
Balanced Healthcare Receivables LLC	Medical	Untimely response	No
ALCHRO INC.	Medical	Untimely response	No
ALCHRO INC.	Medical	Untimely response	No
Nelnet	Non-federal student loan	Closed with explanation	No

consumer_complaints 52

By implementing all the queries created above, we can manipulate the data in the consumer_complaints table by choosing different columns and finding the relationships between them in order to get meaningful information that would help improve decision making.