OVERVIEW AND INTRODUCTION

In today's modern and advancing era technology is advancing day by day and has curbed most of the industries. One such majorly benefited industry or institution is Banking and Finance. Nowadays operation of banking is shifting on a large scale from offline mode to online mode. The new generation are already well versed with the advanced technology but the people using traditional methods are also getting acquainted with the newer methods. The essential thing which plays an important role in the success of any institution be it related to banking, finance, IT etc. is the customer satisfaction and relationship between a customer and a bank in case of banking. The feedback is the most valuable asset since it on constant basis keeps the companies, policies, services and profit in check. With increasing market competition nearly every entity in the banking sector is offering the same services thus providing less room for competition. So therefore, the customer experience pertaining to a bank gives that bank an advantage over another bank. There are several means of communication between a customer and an employee in the bank be it offline face to face, online, mobile banking and ATM. So, providing a smooth service becomes a priority for the bank since a customer expects the same quality of service over each channel of communication.

Our banking services, performance, and feedback MySQL database is a valuable resource for gaining insights into the effectiveness of our banking services and identifying areas for improvement. The database contains information on the various services we offer, as well as performance metrics and customer feedback. By analyzing this data, we are able to gain a better understanding of how our services are being used, how well they are performing, and how satisfied our customers are with their experiences.

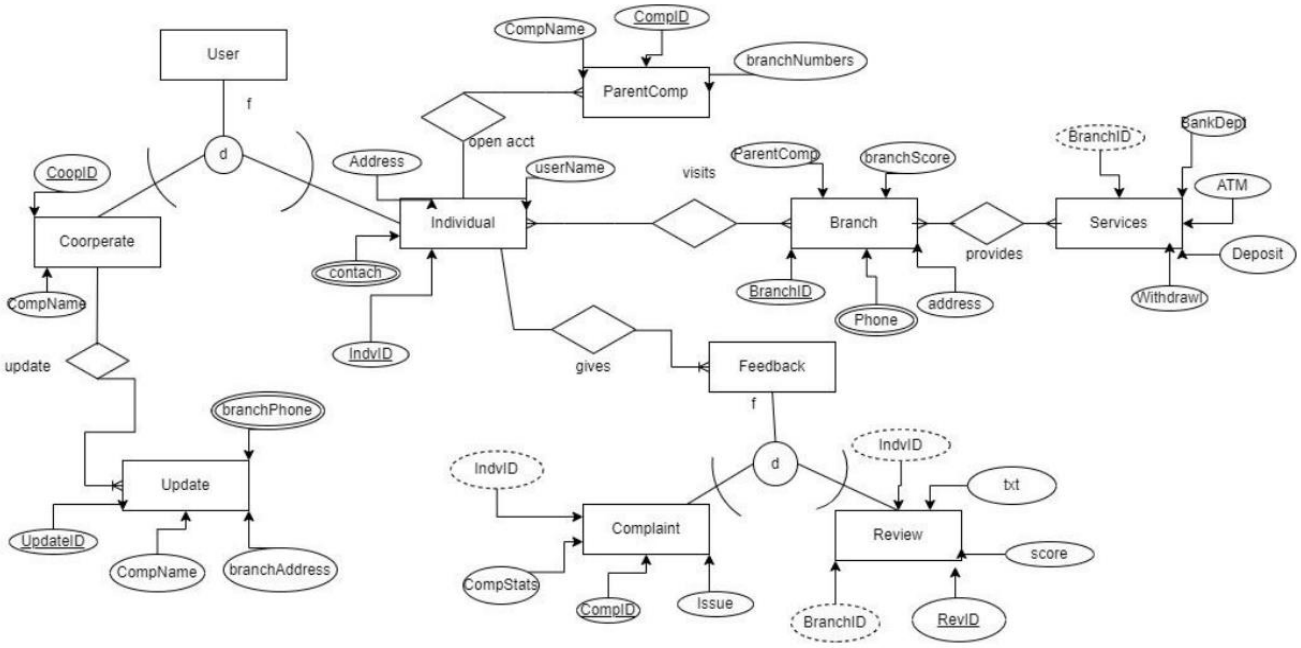
In this report, we present the findings of our analysis of the data contained in the database. We have looked at trends and patterns in the data in order to identify key areas of strength and weakness in our services. We have also compared our performance to that of other banks in order to see how we stack up against the competition.However, our analysis also revealed some areas for improvement. For example, our customer satisfaction scores for in-person services, such as teller transactions and loan applications, are lower than those for our online services. This indicates that we may need to focus on improving the quality of these services in order to better meet the needs of our customers.

Overall, the data contained in our banking services, performance, and feedback MySQL database provides valuable insights into the effectiveness of our services. By analyzing this data, we are able to identify strengths and weaknesses, as well as areas for improvement. We hope that the findings of this analysis will be helpful in informing future decisions about our services and enhancing the customer experience.

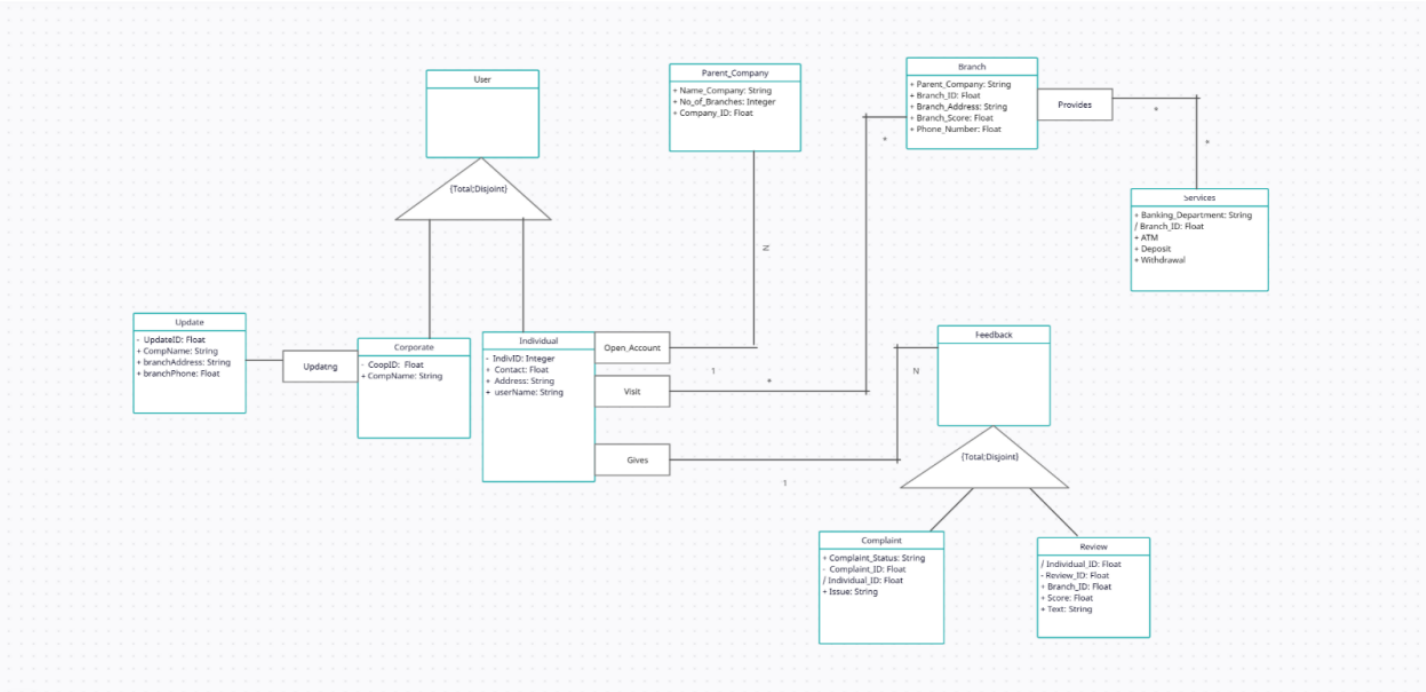
By the means of our project, we focus to build and analyze database through which an individual can access the reviews and performance of a financial institution in which they are interested. The user can have the access to the database before visiting the branch of a financial institution for purposes like opening and account, using ATM, general enquiry etc. and can see the ratings and reviews about the performance accordingly. The user after visiting the bank and completing there required work can upload their experience in the database via a feedback form. On creation of this database, we can provide services and end to end analysis to the parent financial institution/companies regarding the performances of branches, comparison amongst other banks in the market, if a company wants to register themselves in the database and add a branch to their existing network so they will be able to do that by filling an update branch form.

CONCEPTUAL MODELLING

1) ERR Diagram



2) UML Diagram



MAPPING CONCEPTUAL MODEL TO RELATIONAL MODEL:

Primary key: UNDERLINED Foreign Key: *Italicized*

1) Cooperate(CoopID, CompName)

2) ParentComp(CompanyID, CompName, BranchNumbers)

3) Individual(IndvID, UserName, Address, Contact,FirstName, LastName)

4) OpenAcct(*IndvID, CompanyID*, Date)

5) Branch(BranchID, ParentComp, Address, Phone, Score)

6) Services(BranchID, ParentComp, Address, Phone, BranchScore)

7) Complaint(CompID, *IndvID*, CompStatus,Issue)

8) Review(RevID, *BranchID, IndvID*, Text, Score)

9) Update(UpdateID, CompName, Branch_Add, Branch_phone)

IMPLEMENTATION WITH MYSQL

1. **SQL Implementation: The database was created in MySQL and the following queries were performed:**

i) To find the review score of customers, branch, company from Massachusetts having individual id between 5000-7000?

```
with cte as
(select i.individual_id, i.region, r.review_score, r.Branch_ID
from individual_info_2 as i
inner join
review as r on i.Individual_ID = r.Individual_ID
),

cte2 as
( select b.branch_id, b.company_id
from branch as b
inner join
review as r1 on b.Branch_ID = r1.Branch_ID)

select c1.individual_id, c1.region, c1.review_score, c2.branch_id, c2.company_id
from
cte as c1 inner join cte2 as c2 on c1.branch_id = c2.branch_id
where c1.region = "Massachusetts" AND c1.Individual_ID between "5000" AND
"7000";
```

```
#1) to find the review score of customers, branch, company from massachusetts having individual id between 5000-7000?
with cte as
(select i.individual_id, i.region, r.review_score, r.Branch_ID
from individual_info_2 as i
inner join
review as r on i.Individual_ID = r.Individual_ID
),

cte2 as
( select b.branch_id, b.company_id
from branch as b
inner join
review as r1 on b.Branch_ID = r1.Branch_ID)

select c1.individual_id, c1.region, c1.review_score, c2.branch_id, c2.company_id from
cte as c1 inner join cte2 as c2 on c1.branch_id = c2.branch_id
where c1.region = "Massachusetts" AND c1.Individual_ID between "5000" AND "7000";
```

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

individual_id	region	review_score	branch_id	company_id
6360	Massachusetts	5	6668	2378

ii) Finding the number of customers who have opened the account in a specific time frame in a company?

```
with cte2 as
( select o.individual_id, o.company_id, o.Account_Opening_date from
open_account as o),

cte3 as
(select p.Comapny_name, p.company_id from parent_comapny_2 as p)

select count(c2.individual_id) as no_of_customers, (c2.account_opening_date),
c3.comapny_name from
cte2 as c2
inner join
cte3 as c3 on c2.company_id = c3.company_id
group by c2.Account_Opening_date;
```

```
#2) finding the number of customers who have opened the account in a specific time frame in a company?
with cte2 as
( select o.individual_id, o.company_id, o.Account_Opening_date from
open_account as o),

cte3 as
(select p.Comapny_name, p.company_id from parent_comapny_2 as p)

select count(c2.individual_id) as no_of_customers, (c2.account_opening_date), c3.comapny_name from
cte2 as c2
inner join
cte3 as c3 on c2.company_id = c3.company_id
group by c2.Account_Opening_date;
```

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

no_of_customers	Account_Opening_date	Comapny_name
7	1978	Molestie Tortor Nibh Foundation
3	2005	Molestie Tortor Nibh Foundation
5	1971	Egestas Ligula PC
4	2020	Porttitor Tellus Associates
9	2014	Porttitor Tellus Associates
4	1993	Nunc Corp.
6	1983	Nunc Corp.
4	2015	Enim Non Corporation
5	1975	At Auctor Corp.
11	2011	At Auctor Corp.
11	1977	At Auctor Corp.
4	2010	Lacinia Orci Limited
8	1987	Luctus Sit Amet Corporation
4	2007	Luctus Sit Amet Corporation
10	2006	Luctus Sit Amet Corporation
7	1989	Pharetra Quisque Institute
2	1998	Pharetra Quisque Institute
3	2016	Suspendisse Ac Associates
8	1992	At Velit LLP
3	1996	Vulputate Mauris Industries

iii) To find the number of branches in each region alongwith there branch_id?

with cte4 as

(select r.individual_id, r.branch_id

from review as r

left join

branch as b on r.branch_id = b.branch_id)

select i.region, count(c4.branch_id) as total_branches, c4.branch_id

from individual_info_2 as i

inner join

cte4 as c4 on i.Individual_ID = c4.Individual_ID

group by i.region, c4.branch_id;

```
#3) to find the number of branches in each region alongwith there branch_id?
with cte4 as
( select r.individual_id, r.branch_id
from review as r
left join
branch as b on r.branch_id = b.branch_id)

select i.region, count(c4.branch_id) as total_branches, c4.branch_id
from individual_info_2 as i
inner join
cte4 as c4 on i.Individual_ID = c4.Individual_ID
group by i.region, c4.branch_id;
```

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

no_of_customers	Account_Opening_date	Comapny_name
7	1978	Molestie Tortor Nibh Foundation
3	2005	Molestie Tortor Nibh Foundation
5	1971	Egestas Ligula PC
4	2020	Porttitor Tellus Associates
9	2014	Porttitor Tellus Associates
4	1993	Nunc Corp.
6	1983	Nunc Corp.
4	2015	Enim Non Corporation
5	1975	At Auctor Corp.
11	2011	At Auctor Corp.
11	1977	At Auctor Corp.
4	2010	Lacinia Orci Limited
8	1987	Luctus Sit Amet Corporation
4	2007	Luctus Sit Amet Corporation
10	2006	Luctus Sit Amet Corporation
7	1989	Pharetra Quisque Institute
2	1998	Pharetra Quisque Institute
3	2016	Suspendisse Ac Associates
8	1992	At Velit LLP
3	1996	Vulputate Mauris Industries
2	2019	Mus Proin Corp.

iv) If the rating is less than 1 then give title bad reviews, if it is between 1 and 3 give title moderate review and greater than 3 then good?

```
with cte6 as (
select review_score,
( case when r.review_score < 1 then "Bad Reviews"
when r.review_score between 1 and 3 then "Moderate Reviews"
when r.review_score > 3 then "Good Reviews"
else "No review is mentioned"
end ) as review_bifurcation, r.branch_id
from review as r),

cte7 as
(
Select b.branch_id, c6.review_bifurcation,c6.review_score
from branch as b
inner join cte6 as c6
on b.branch_id = c6.branch_id)

select * from cte7
order by cte7.review_score desc;
```

```
#4)if the rating is less than 1 then give title bad reviews, if it is between 1 and 3 give title moderate review and greater than 3 then good?
with cte6 as (
select review_score,
( case when r.review_score < 1 then "Bad Reviews"
      when r.review_score between 1 and 3 then "Moderate Reviews"
      when r.review_score > 3 then "Good Reviews"
      else "No review is mentioned"
      end ) as review_bifurcation, r.branch_id
   from review as r),

   cte7 as
   (
   select b.branch_id, b.company_id, c6.review_bifurcation,c6.review_score
   from branch as b
   inner join cte6 as c6
   on b.branch_id = c6.branch_id)

   select * from cte7
   order by cte7.review_score desc;
```

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

branch_id	company_id	review_bifurcation	review_score
181	6434	Good Reviews	5
3256	2919	Good Reviews	5
5606	2484	Good Reviews	5
9653	2316	Good Reviews	5
7936	4228	Good Reviews	5
7759	2285	Good Reviews	5
7332	9289	Good Reviews	5
5159	8331	Good Reviews	5
2098	9722	Good Reviews	5
3592	900	Good Reviews	5
2137	9549	Good Reviews	5
2233	4716	Good Reviews	5

v) Finding customers whose complaint status is still in progress?

with cte8 as

(select (r.branch_id) as total_branch, c.complaint_status, c.individual_id

from review as r

left join

complaint as c on r.individual_id = c.individual_id

where c.complaint_status = "In progress")

#group by c.complaint_status)

select * from cte8;

```
#5) finding customers whose complaint status is still in progress?
with cte8 as
(select (r.branch_id) as total_branch, c.complaint_status, c.individual_id
from review as r
left join
complaint as c on r.individual_id = c.individual_id
where c.complaint_status = "In progress")
#group by c.complaint_status)

select * from cte8;
```

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

total_branch	complaint_status	individual_id
181	In progress	317
456	In progress	443
519	In progress	539
532	In progress	571
568	In progress	631
733	In progress	693
748	In progress	974
797	In progress	1111
1043	In progress	1161
1343	In progress	1415
1618	In progress	2058
1623	In progress	2251
1636	In progress	2254
1648	In progress	2363
1652	In progress	2370
1817	In progress	2484
1908	In progress	2898
2098	In progress	3273
2233	In progress	3359
2611	In progress	3723
2780	In progress	3922

vi) To find the complaint_status which are in progress and Conversion rate for complaints still in progress via different modes of communication.

```
with cte10 as (
select (c.complaint_status), (c.Complaint_ID)
from complaint as c where c.complaint_status = "In progress")

select * from cte10;

# to find how much complaints were submitted via web mode.
with cte11 as (
select (c1.submitted_via) as mode_of_submission, c1.Complaint_ID
from complaint as c1 where c1.submitted_via = "web")

select * from cte11;
```

#)Conversion rate for complaints still in progress via different modes of communication.

```
select
count(case when c.submitted_via = "web" AND c.complaint_status = "in progress" then c.submitted_via else null end) /
count(case when c.complaint_status ='in progress' then c.complaint_status else null end) as web_to_in_progress_rate,
count(case when c.submitted_via = "phone" AND c.complaint_status = "in progress" then c.submitted_via else null end) /
count(case when c.complaint_status ='in progress' then c.complaint_status else null end) as phone_to_in_progress_rate,
count(case when c.submitted_via = "referral" AND c.complaint_status = "in progress" then c.submitted_via else null end) /
count(case when c.complaint_status ='in progress' then c.complaint_status else null end) as referral_to_in_progress_rate
from complaint as c;
```

```
#6) to find the complaint_status which are in progress and Conversion rate for complaints still in progress via different modes of communication.
 with cte10 as (
 select (c.complaint_status), (c.Complaint_ID)
 from complaint as c where c.complaint_status = "In progress")

 select * from cte10;

 # to find how much complaints were submitted via web mode.
 with cte11 as (
 select (c1.submitted_via) as mode_of_submission, c1.Complaint_ID
 from complaint as c1 where c1.submitted_via = "web")

 select * from cte11;

#)Conversion rate for complaints still in progress via different modes of communication.
 select
 count(case when c.submitted_via = "web" AND c.complaint_status = "in progress" then c.submitted_via else null end) /
 count(case when c.complaint_status ='in progress' then c.complaint_status else null end) as web_to_in_progress_rate,
 count(case when c.submitted_via = "phone" AND c.complaint_status = "in progress" then c.submitted_via else null end) /
 count(case when c.complaint_status ='in progress' then c.complaint_status else null end) as phone_to_in_progress_rate,
 count(case when c.submitted_via = "referral" AND c.complaint_status = "in progress" then c.submitted_via else null end) /
 count(case when c.complaint_status ='in progress' then c.complaint_status else null end) as referral_to_in_progress_rate
 from complaint as c;
```

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

web_to_in_progress_rate	phone_to_in_progress_rate	referral_to_in_progress_rate
0.5079	0.1746	0.2222

vii) Specifying a limit for no of atm,deposit and withdrawal and the catgeorizing them in small scale.

with cte12 as(

select

count(case when b1.branch_atm < 15 then b1.Branch_ID else null end) as Small_scale_atm_branch,

count(case when b1.branch_deposit <10 then b1.branch_id else null end) as Small_scale_deposit_branch,

count(case when b1.branch_withdrawal <10 then b1.branch_id else null end) as small_scale_withdrawal_branch

from branch_services as b1)

select * from cte12 as c12;

```
#7) Specifying a limit for no of atm,deposit and withdrawal and the catgeorizing them in small scale.
with cte12 as(
select
count(case when b1.branch_atm < 15 then b1.Branch_ID else null end) as Small_scale_atm_branch,
count(case when b1.branch_deposit <10 then b1.branch_id else null end) as Small_scale_deposit_branch,
count(case when b1.branch_withdrawal <10 then b1.branch_id else null end) as small_scale_withdrawal_branch
from branch_services as b1)

select * from cte12 as c12;
```

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

Small_scale_atm_branch	Small_scale_deposit_branch	small_scale_withdrawal_branch
76	39	47

viii) Correlated Query: To find the top-rated reviews by the customers.

select r.review_score as top_rating, r.individual_id, r.review_id
from review as r where r.Review_Score in
(select max(r2.review_score) from review as r2);

```
#8 Correlated Query: To find the top rated reviews by the cutomers.
select r.review_score as top_rating, r.individual_id, r.review_id
from review as r where r.Review_Score in
( select max(r2.review_score) from review as r2 );
```

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

top_rating	individual_id	review_id
5	317	112
5	3273	2859
5	3309	2889
5	3359	2989
5	4118	4064

ix) **Correlated Query: Finding number of customers from California region**.

select (i.individual_id), i.region from individual_info_2 as i

where i.region = "California" AND

(select sum(i1.individual_id) from individual_info_2 as i1);

```
#9 Correlated Query: Finding number of customers from California region.
select (i.individual_id), i.region from individual_info_2 as i
where i.region = "California" AND
(select sum(i1.individual_id) from individual_info_2 as i1);
```

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

individual_id	region
3486	California
4426	California
5005	California
6624	California
7967	California

x) Correlated subquery using exists: Customers opening account in the year 2005 along with company id.

select o.Individual_ID, o.company_id from open_account as o

where exists

(select o1.account_opening_date from open_account as o1 where o.individual_id= o1.Individual_ID AND o1.account_opening_date = "2005");

```
#10 Correlated subquery using exists: Customers opening account in the year 2005 aling with company id.
select o.Individual_ID, o.company_id from open_account as o
where exists
(select o1.account_opening_date from open_account as o1 where o.individual_id= o1.Individual_ID AND  o1.account_opening_date = "2005");
```

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

Individual_ID	company_id
317	87
1220	23

NOSQL IMPLEMENTATION

For Nosql Implenentaion we used Graph method using Neo4j AuraDB, where two relations were created which are

i) Individual -> Opens_account -> Parent_company

ii) Individual -> Visits -> Branch

i) Return name of individuals, branch score and branch id where individual visits branch with branch score greater than 4

```
MATCH(i:individual)-[v:Visits]→(b:Branch)
WHERE b.Branch_Score > 4
RETURN i.name, b.Branch_Score, b.Branch_ID
```

Output:

i.name	b.Branch_Score	b.Branch_ID
"Ainsley House"	5	456
"Maya Benton"	5	1326
"Hadley Booker"	5	1618
"Dean Booker"	5	2024
"Brendan Wallace"	5	3341
"Rachel Contreras"	5	3582

ii) Return individual name, company name and branch numbers where individual opens their account but sort the output by highest number of branches

```
MATCH(i:individual)-[o:`Opens Account`]→(p:Parent_Company)
RETURN i.name, p.Comapny_name, p.Branch_numbers
order by p.Branch_numbers DESC
```

Output:

i.name	p.Comapny_name	p.Branch_numbers
"Kirsten Washington"	"Nunc Pulvinar Corp."	99
"Dennis Crosby"	"Tellus Lorem Ltd"	98
"Kennan Hickman"	"Tellus Lorem Ltd"	98
"Elmo Lane"	"Eleifend Nunc Risus LLP"	97
"Sean Vance"	"Eleifend Nunc Risus LLP"	97
"Maryam Smith"	"Eleifend Nunc Risus LLP"	97

Showing **1-50** of **100** results

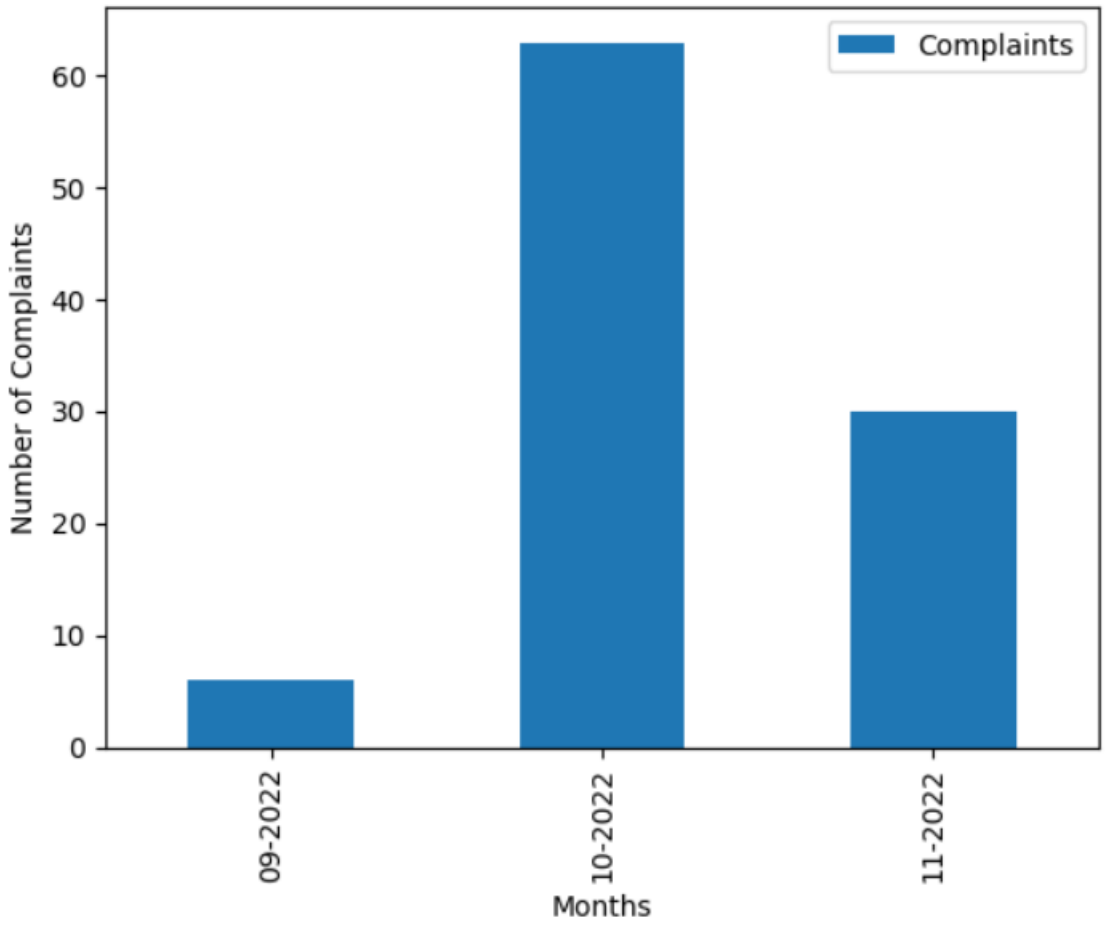
1 2 Show 50

Started streaming 100 records after 12ms and completed after 14ms.

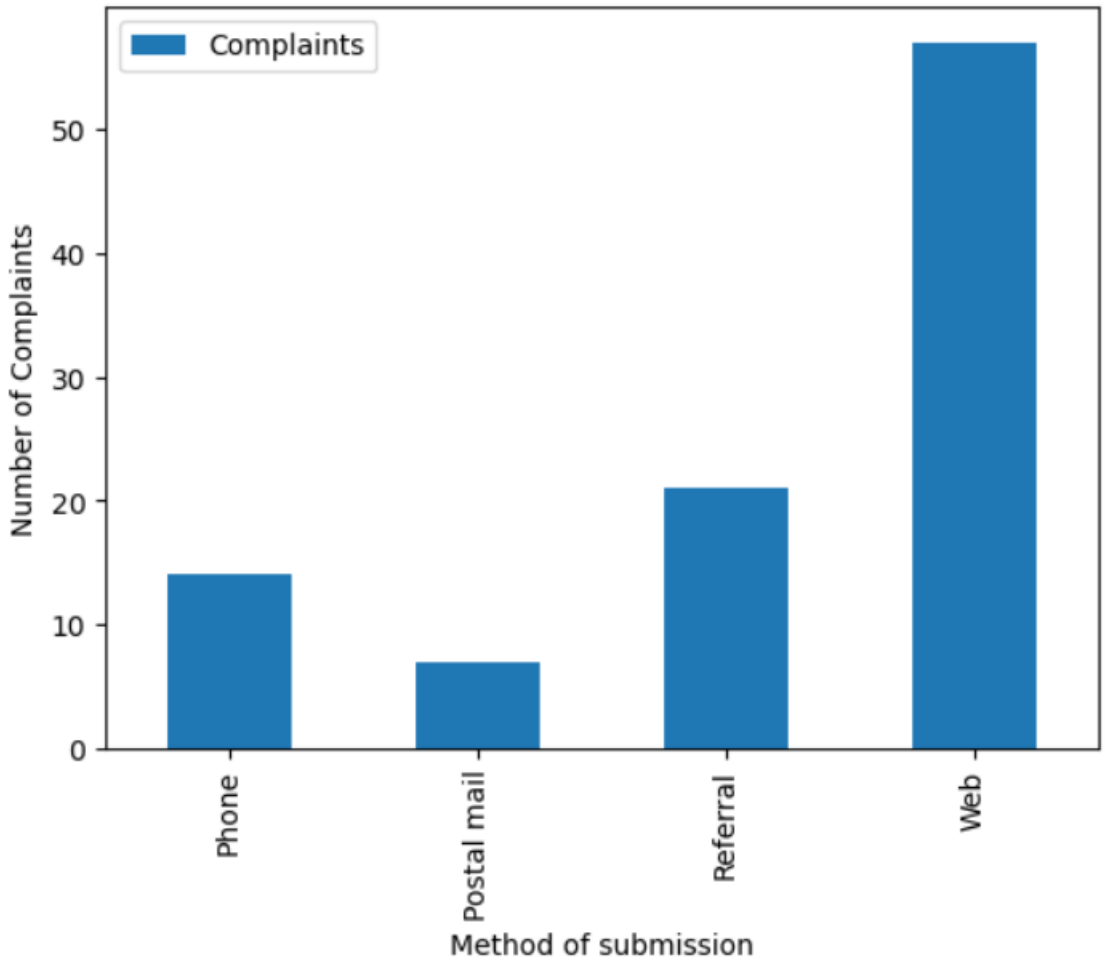
DATABASE ACCESS VIA PYTHON

To access the database, Python and the pymysql.connect library are used. The pd.read_sql_query method is used to run and retrieve the results of a query, which are already converted into a dataframe. Finally, matplotlib is used to visualize the analyzed data.

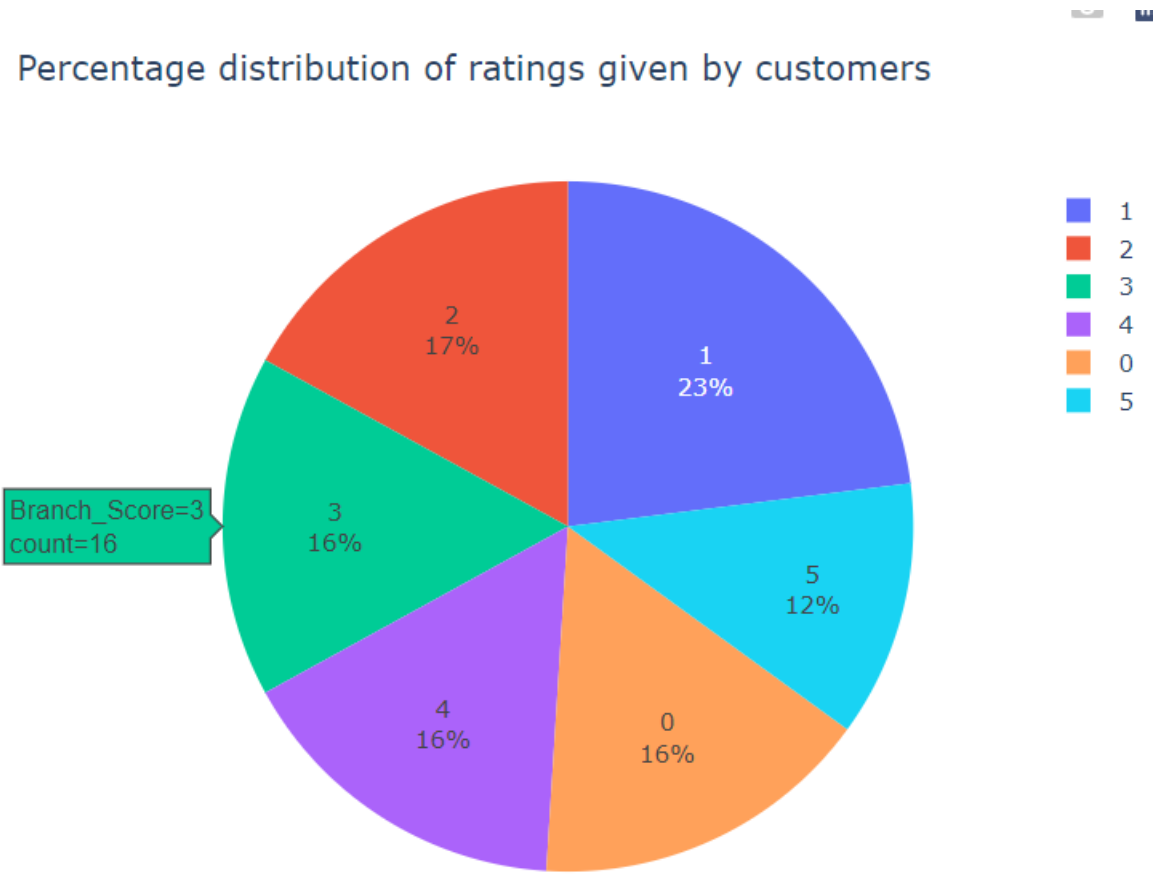
Graph 1: Display Number of complaints received on a month basis.



Graph 2: Display the number of complaints grouped by mode of submission of the complaints.



Graph 3: Percentage distribution of Scores given by customers to branches.

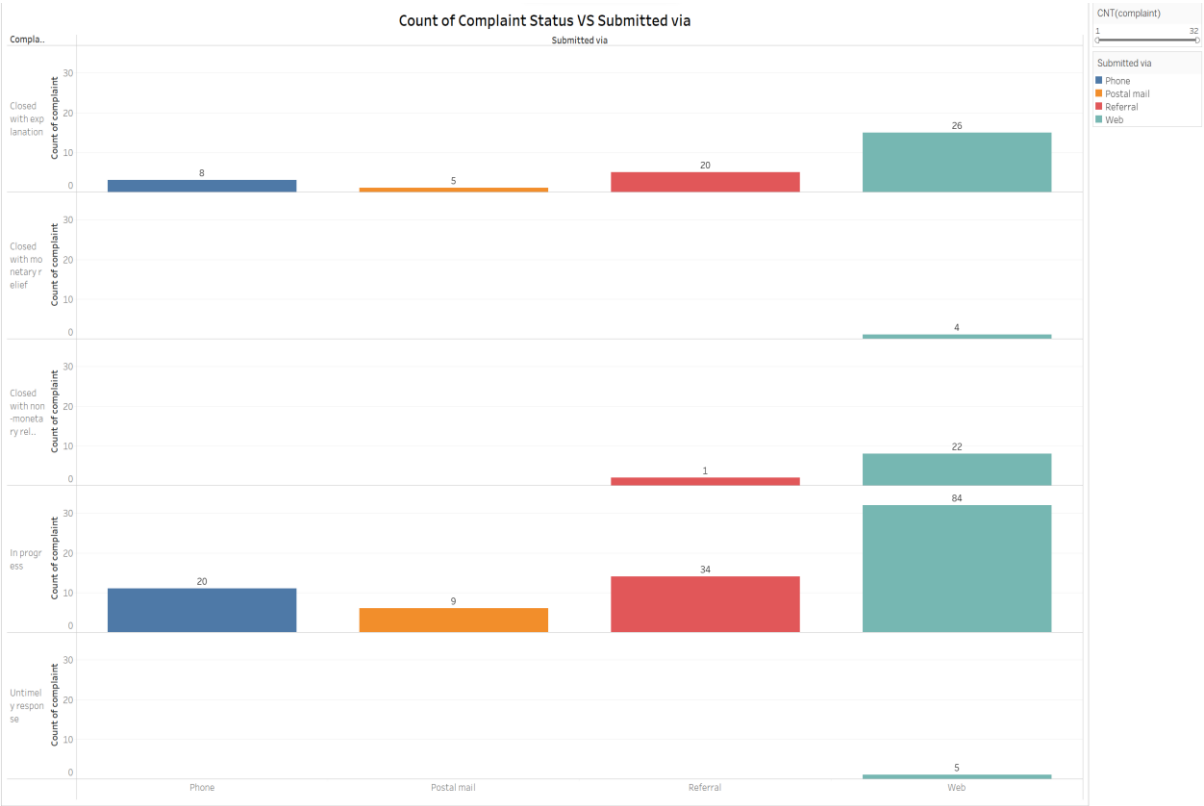


IMPLEMENTATION IN TABLEAU

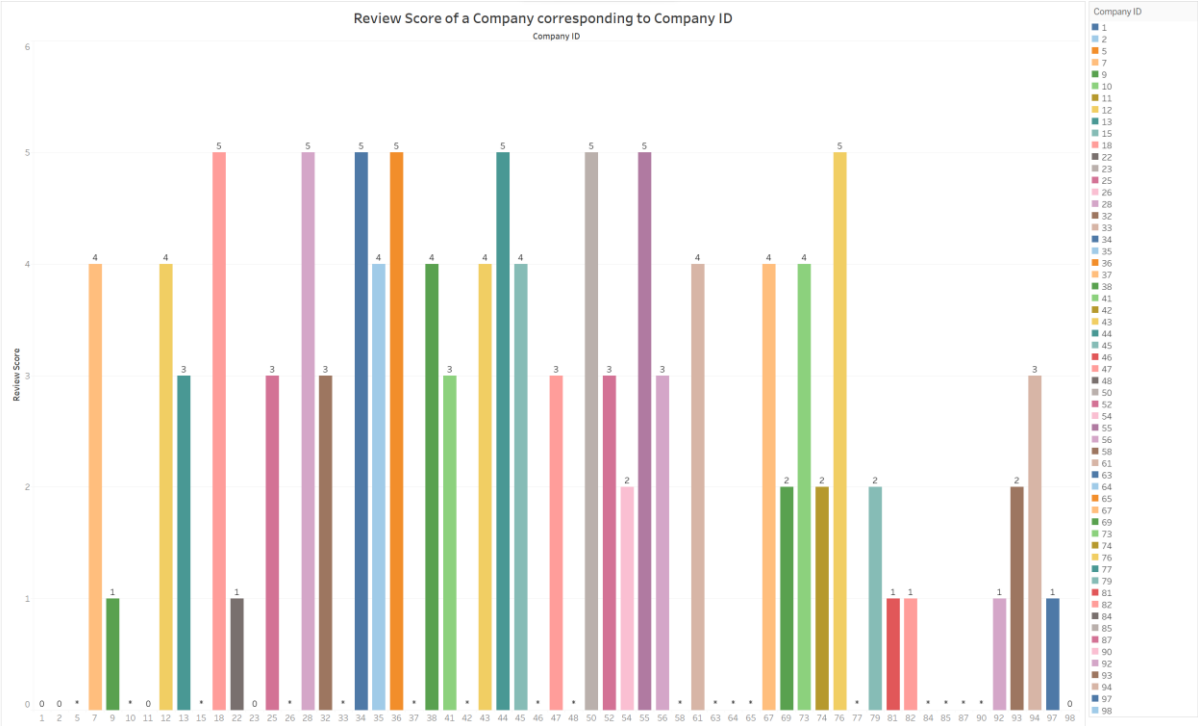
Connected Tableau and MYSQL Workbench by installing the required drivers and then initiated the connection.

Used the join feature in Tableau to join tables and perform visualization.

i) To find how many complaints were received via each mode and what is there status



ii) To find the review score of a company.



FUTURE WORK

- The user can have the access to the database before visiting the branch of a financial institution for purposes like opening and account, using ATM, general enquiry etc. and can see the ratings and reviews about the performance accordingly.
- The user after visiting the bank and completing there required work can upload their experience in the database via a feedback form.
- On creation of this database, we can provide services and end to end analysis to the parent financial institution companies regarding the performances of branches, comparison amongst other banks in the market.