**BANKING MANAGEMENT SYSTEM**

Submitted by: Akashdeep Singh

Harleen Gulati

Mohammed Abdul Akram Khan

Mukhasir Shah Syed

**Introduction:**

The project is built on Banking and its management. In today’s scenarios, we all understand that banking, especially online banking, has a very major role to play in our lives. That is the reason and motivation behind this project.

Banks hold a large amount of data that is sensitive in nature and highly confidential. Banking also involves huge amount of money, therefore it is very important for data to be non-redundant.

Basic functionalities of our system are:

* + - Account creation
    - Deposit
    - Withdrawal
    - Loan

The technologies used in the system are:

* RDBMS using MySQL
* IDE used is MySQL workbench
* Web Interface built over .net Framework

The concepts implanted in the project are:

* The Concept of keys
* Basic SQL queries
* Aggregate functions
* Views
* Stored procedures
* Data representation and analysis

The project was implemented with the following specifications and details, enlisted by various sub-sections and snapshots.

**Queries and View**

1. All account whose balance is smaller than 500.

**A**: select account\_number from account where balance < 500;

1. all name of customers whose city is in Brooklyn

**A**: select customer\_name from customer where customer\_city='Brooklyn';

1. Calculate the average salary of all employees and show the average salary as “avg\_salary”

**A**: select avg(salary) as avg\_salary from employee ;

1. Calculate the number of customer for each account

**A**: select account\_number, count(distinct customer\_name) from depositor group by account\_number;

1. Show all account\_number, branch\_name and corresponding branch\_city

**A**: select account\_number, branch.branch\_name, branch\_city from account, branch

where account.branch\_name=branch.branch\_name;

6. Select sum(assets) ,branch\_city from branch group by branch\_city;

*7. View*

Create

VIEW my\_report AS

select d.customer\_name,branch\_city, b.branch\_name, a.account\_number, balance, bb.loan\_number, l.amount as 'loan amount' from branch b

left join account a

on b.branch\_name = a.branch\_name

left join depositor d

on a.account\_number = d.account\_number

left join borrower bb

on d.customer\_name = bb.customer\_name

left join loan l

on bb.loan\_number = l.loan\_number

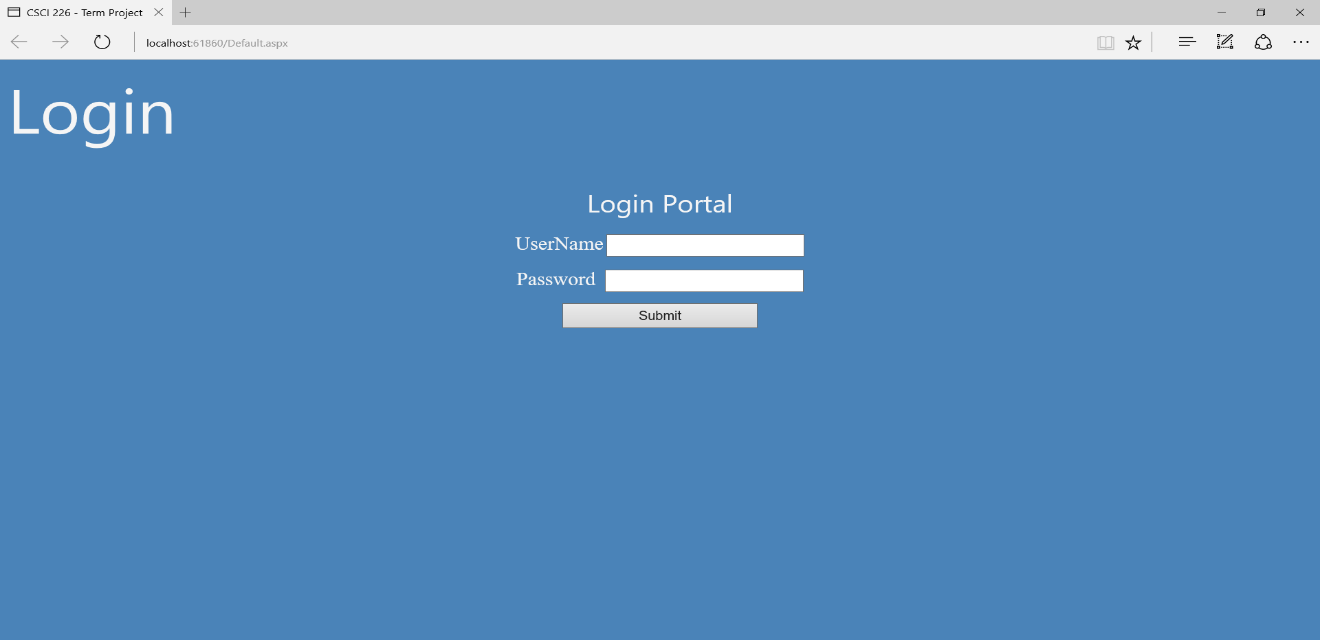
order by d.customer\_name;

**Web Application:**

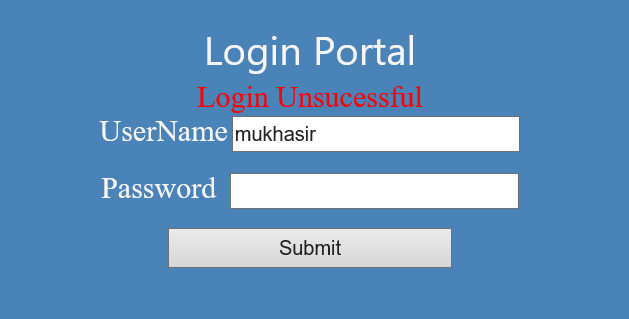
We have created a web application to show how the changes are made on data from the front end as the end user will have no idea on how to write queries on SQL database. The web application is built on .NET Framework 4 using ASP.NET, and we have used C# language to write code.

In this banking website, user can login and add account, remove account, add branch, make payments into personal and loan accounts and, withdraw from personal accounts and clean loan amount. In home page the admin can view all information from all the tables related to Accounts, Customer, Loans.

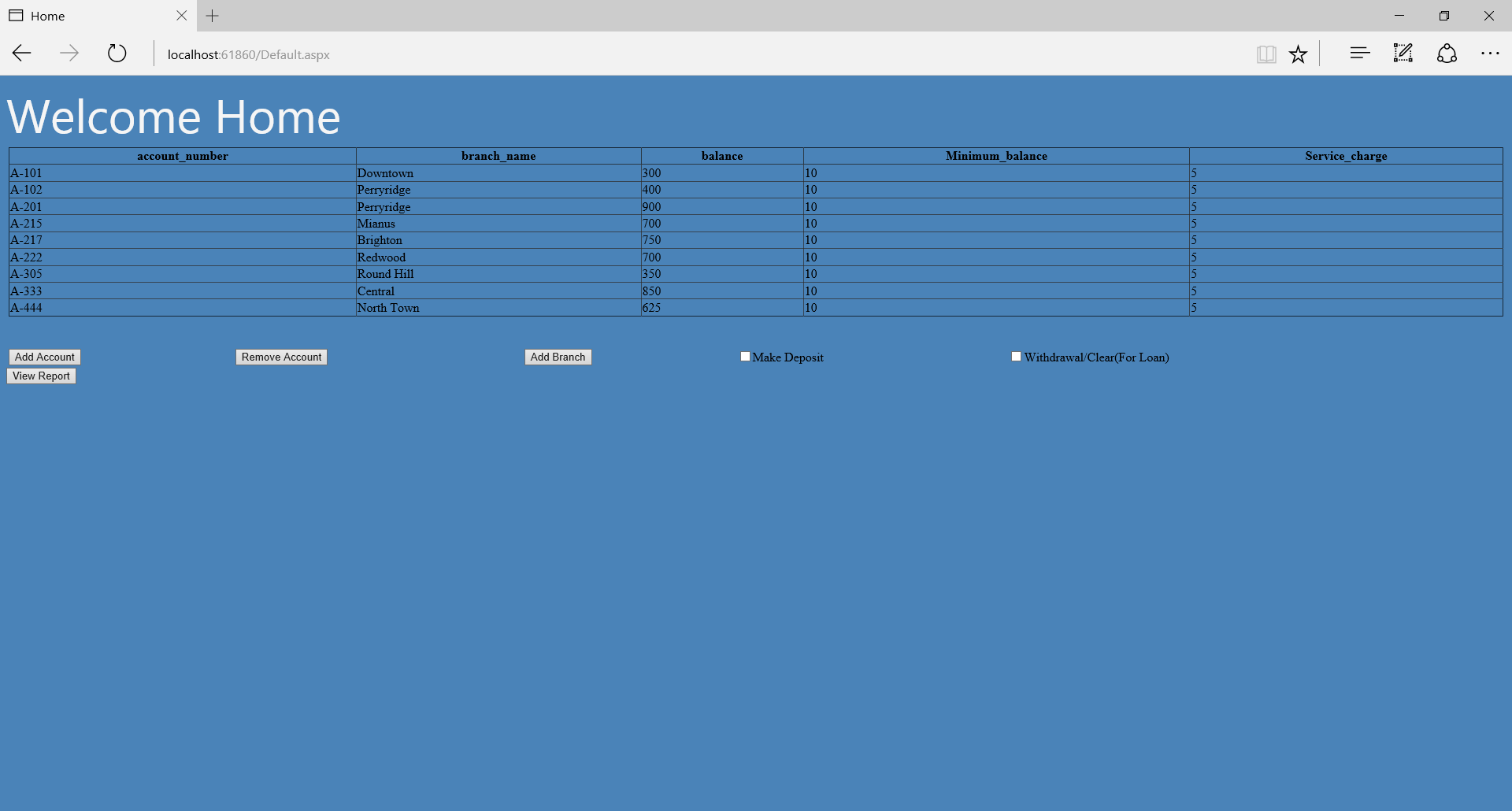
We will go through the website with some screenshots of webpage

1. **Login Page:**

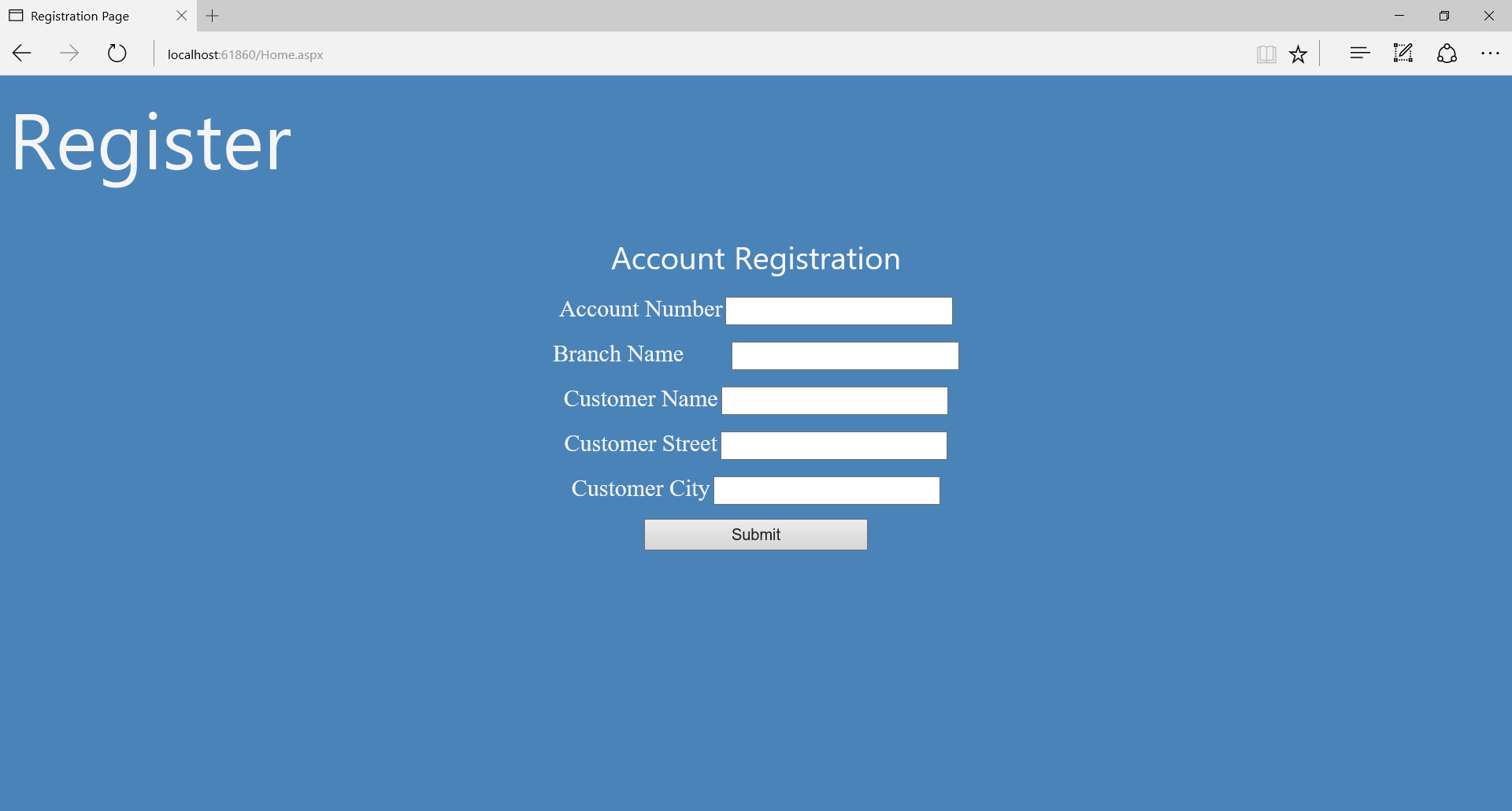
1. There is a check on if admin account exists or not.



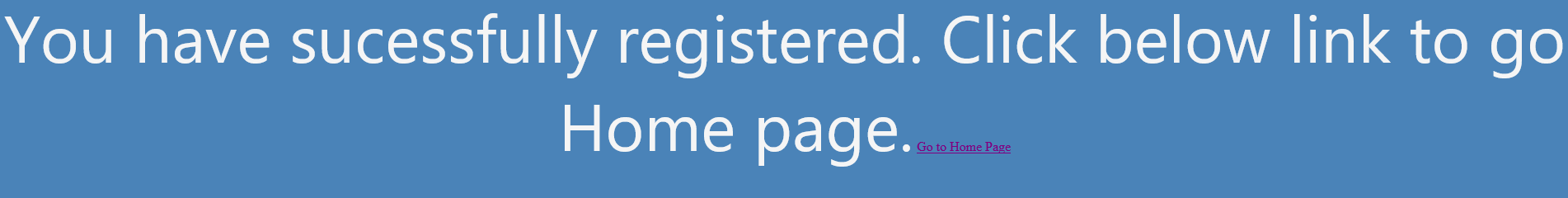
1. If we successfully login then we are redirected to home page which shows all accounts and buttons which on click can perform operations such as add account, remove account, add branch, deposit in personal or loan account, withdraw from personal or clear loan.



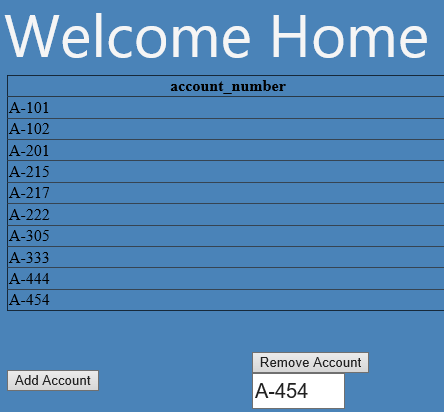
1. Now If we click on Add Account button then we will be redirected to Account Registration page where general information is requested.



1. Once we fill the data and click on submit we will connect to MySql Database and execute stored procedure “” which adds account into the respective tables. If the account creation is successful then we redirect to successful registration page from where we can go to home page.



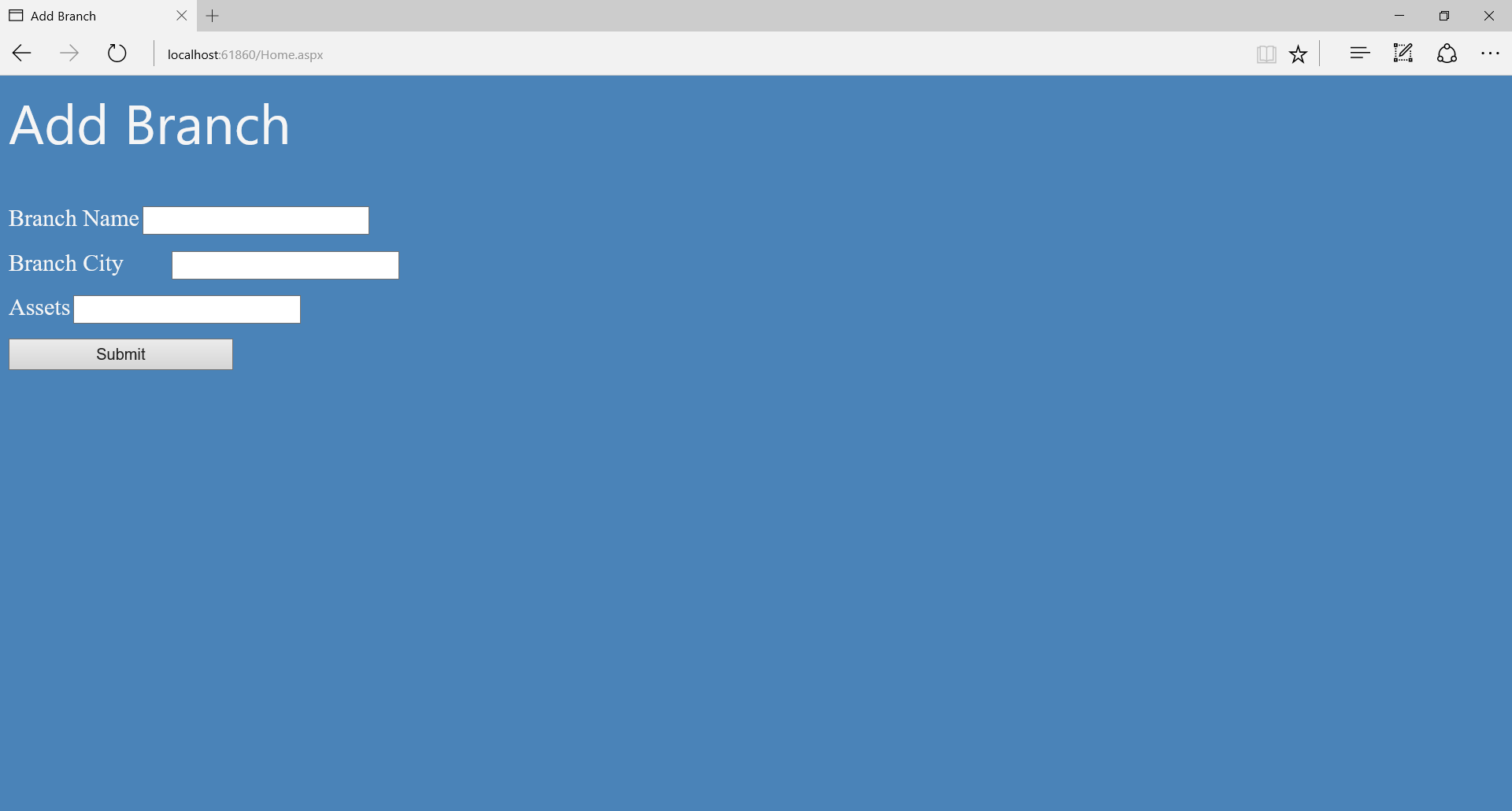
1. After successful registration, we go to Home page. Now if we click Remove account here then a textbox will come up, if we enter Account number and click on remove account button then the account will be removed from database.



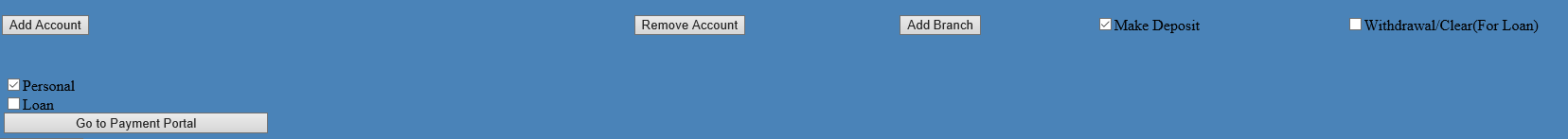
When we account is refreshed, then the table will be refreshed.



1. Now we have functionality to Add branch, when we click on Add branch button it will be redirected to page below.



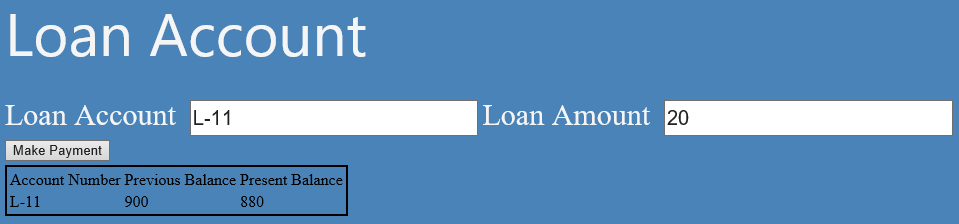
1. Fill in the values in the field and by clicking in the submit button we can add the branch.
2. We also have functionality to make a deposit into Personal Account or Loan Account by selecting the check box “Make Deposit”



Go to Payment Portal and type respective account number (Personal or Loan) and amount and click in “Make Payment” button to make payment and display the current and previous balance.



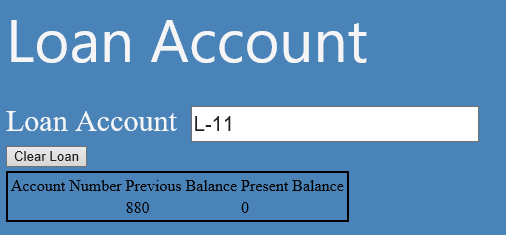
We can have similar page for loan



1. Similar to make payment we have withdrawal from personal account and clear loan account. When we select withdraw/clear then if we select personal we go to the page mentioned below:



If loan is selected then we will have only Loan Account Info where we will make remaining loan amount as “Zero”



1. Finally, we display all report by joining all the tables when you click on “View Report” button.



**Data Analysis and Representation – Tableau**

Tableau is a visualization software that we used in our project for data representation. It is a widely accepted and useful software which makes the data relations much easier to understand and comprehend.

In addition to visualization, tableau also helps learn basic concepts in database.

One good example is difference between how various “join” work.

Why we chose Tableau :

1. Direct connection with MySQL:

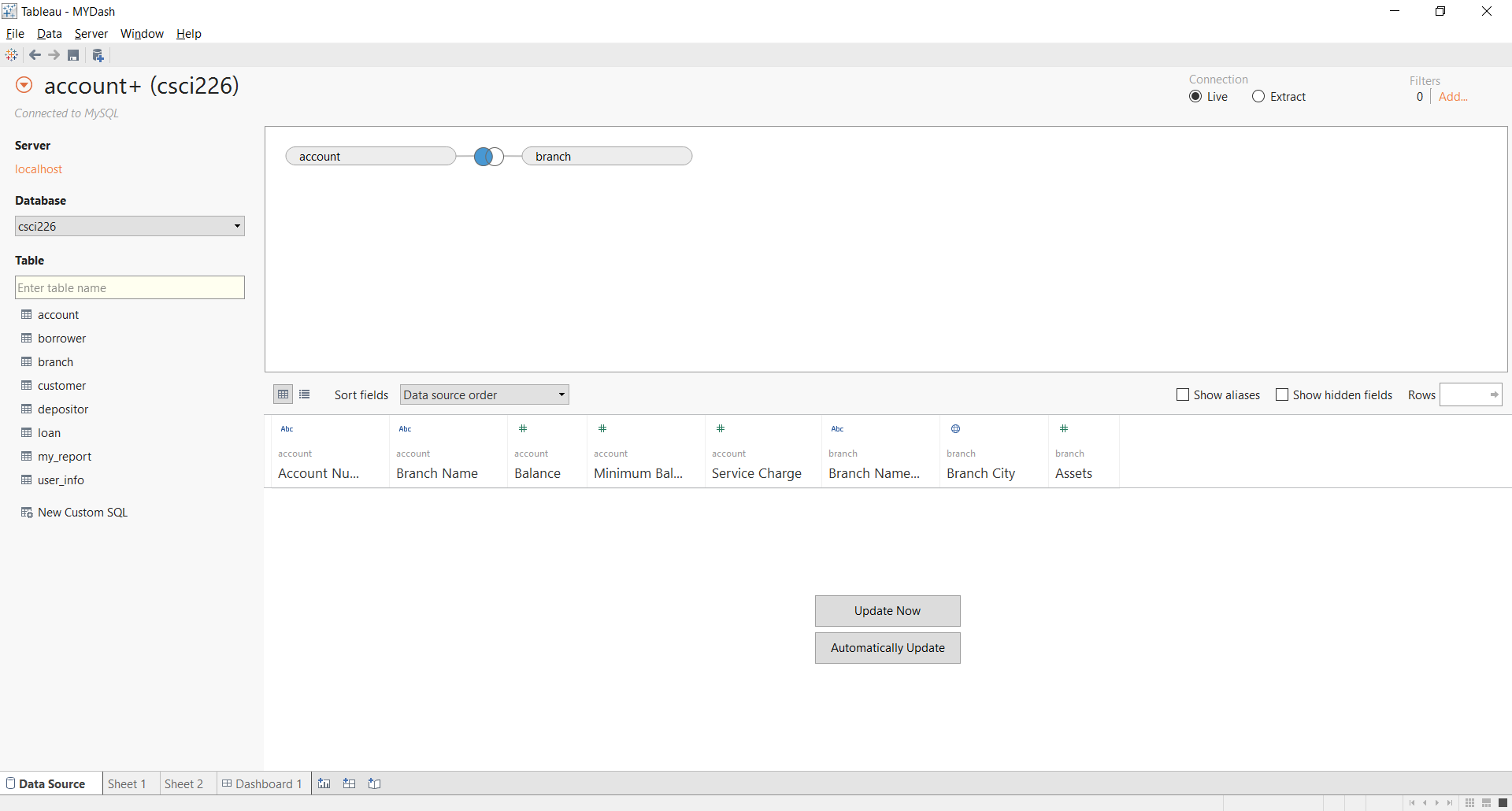
Tableau gives us an option of attaching our MySQL database with it in a simple fashion. The database can directly be linked and shown in a better visualization method.

1. Dynamic in Nature: On selection of different column the visualization of data change place without even writing any code in backend.

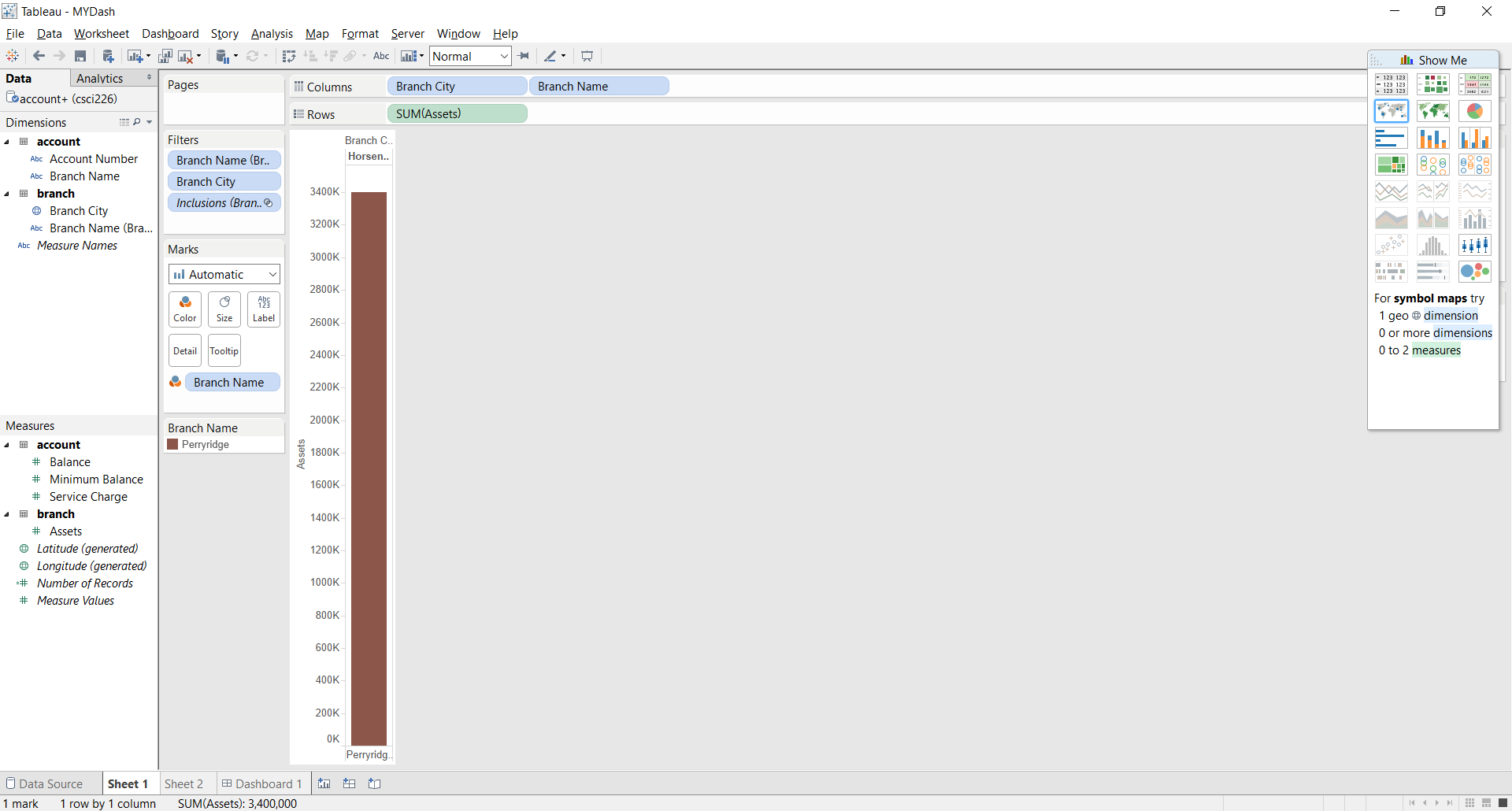
Tableau is dynamic in nature and best data visualization tool in market currently, Tableau understands data – As in it dynamically allow user to have dimensions and measures in provided data.

Measures are the column for which data can be pulled. For Example – Country can be a measure and sales in that country can be measure, this provides user with aggregations, like we can have a visualization for Sum of sales.

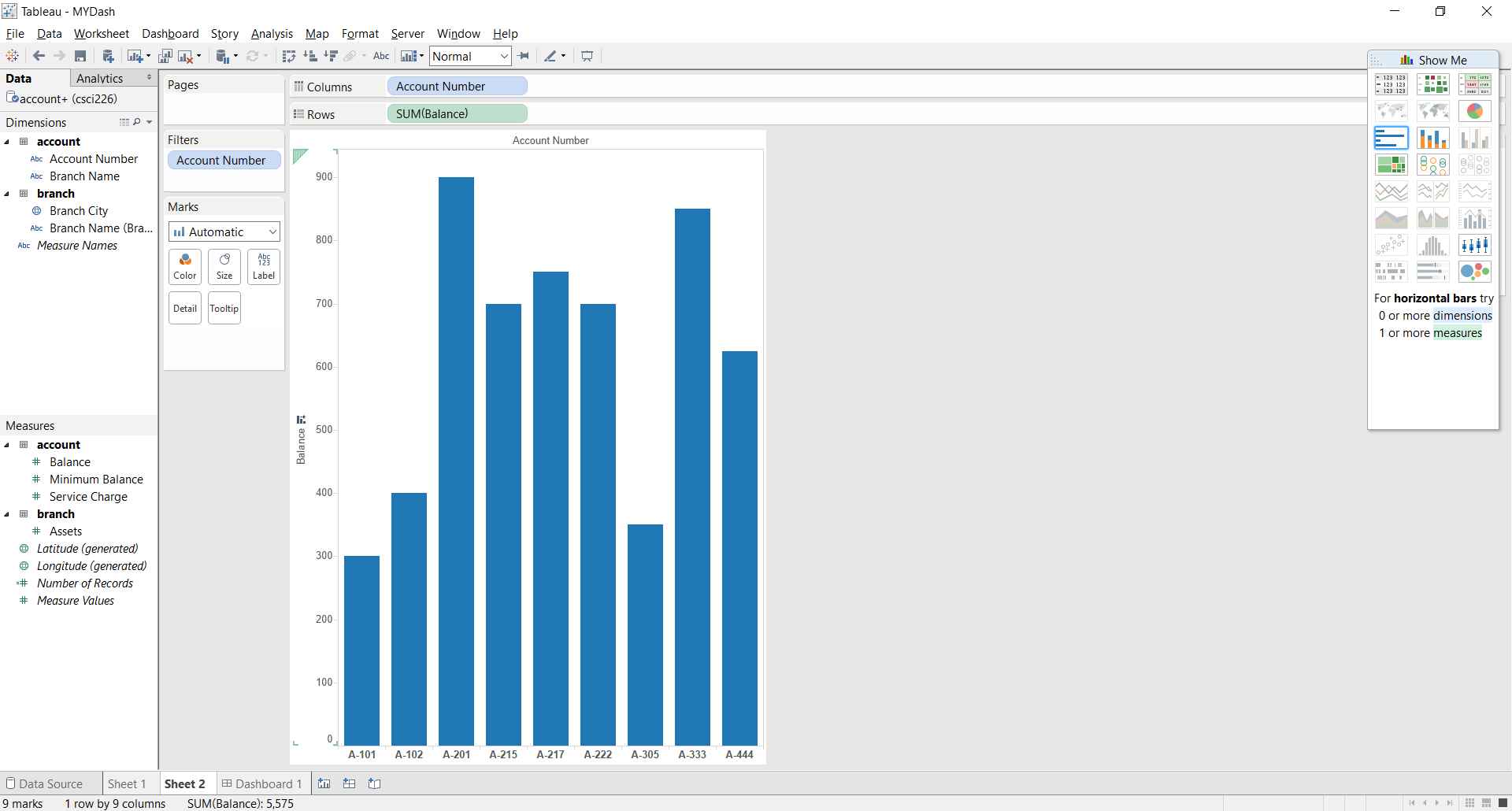
Tableau accepts wide data acceptation such as – MySQL, Excel, Access DB, CSV and many more. Also, changes in the database are immediately reflected in the software which makes it very helpful for use in real time. Further, it can be better explained with the following screenshots:



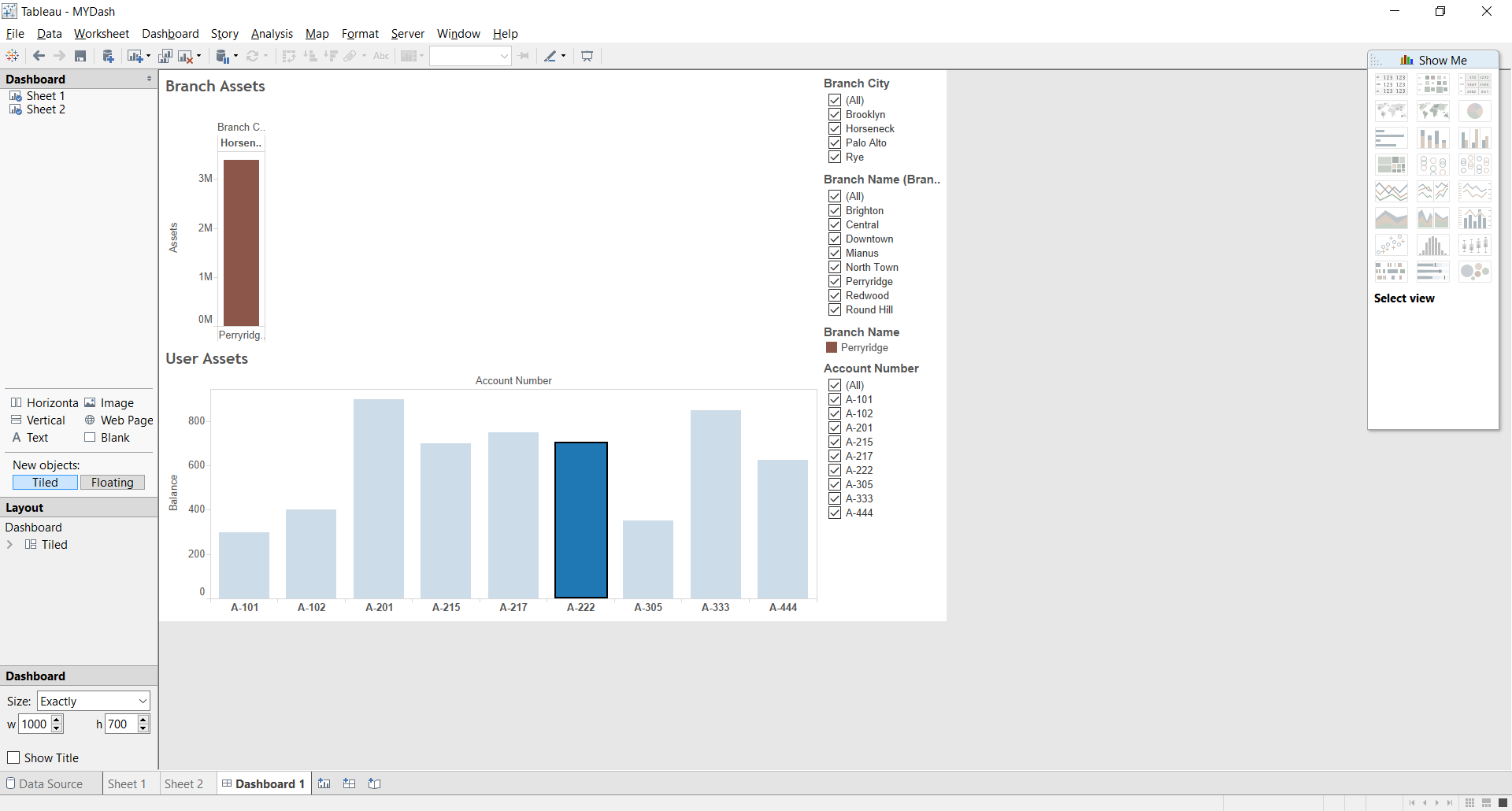
This screenshot shows simple left join that can be easily created and visualized in Tableau. Here it is done between account table and branch table.



This is a screen shot while creatimg a simple sheet1 with branch city,branch name and sum deposited by the customer. The show me tab on the right gives an idea of various options of graphs that Tableau has to offer.



This is another sheet created between account number and sum deposited and shows in form of bar graphs.



This screenshot shows a dashboard that compares data in various sheets and helps better understanding of how they are related/

Additional color coding policies and hide and show filter make Tableasu a very effective tool for visualization of data.