Alec Schneider

July 27th, 2020

Database Project

## Next Steps:

1. Continue to Refine the conceptual and logical models (boldface updates made to the required fields in the logical model on draw.io) -done
2. Create fake data based on the business rules -done
3. Insert the data via Pythons -done
4. Create Views, Procedures,
5. Insert Data

## Narrative:

People are struggling to understand their financial health now more than ever due to COVID-19. To better understand how we can improve our future financial well beings we need to understand how our debts, bank, and retirement funds change over time. Spending behaviors can be changed by seeing the effects on balances they have, leading to improved financial standing, better decision making, and long-term wealth accumulation to assist future generations. It is time people log their account balances, incomes, and expenses to receive quick and simple to understand feedback about their financial health in order to create a better world for themselves and others.

## Business Rules

* A person is considered a User the moment they sign up with their information.
* A User can have zero or more than one Account. An Account can only belong to one User.
* A User can have zero or many Cash Flow patterns and a Cash Flow pattern can only belong to one and only one user.
* An Account has one and only one Account Type and an Account type belongs to one and only one Category.
* A Category can have many Account Types and an Account Type can belong to zero or many Accounts.

## Glossary

* **User** – A person who has entered their personal information, and will b
* **Discretionary monthly Income** – Income after taxes
* **Average Spending** – average estimated monthly spending
* **Account Category** – categorization of the account that describes the general characteristics of the account
  + **Debt** – an obligation to pay money to the creditor who loaned the money
  + **Investment** – an account in which you hold stocks, bonds, mutual funds, or other investment vehicles
  + **Bank** – accounts in which can be easily accessed and withdrawn from.

## What Would I Want To Know?

* What is my debt to income ratio?
  + View of all debt\_to\_incomes created dbo.debt\_to\_income\_ratios
  + Function created dbo.user\_debt\_to\_income
* What percent of my monthly income am I spending?
  + Created function spend\_ratio
* How has spending changed over time?
  + Done for user 650 in python
* How has my debt changed over time?
* What is my debt to bank account balance ratio?
* How much interest am I paying on my debt every year?

## Conceptual Model

A screenshot of a cell phone

Description automatically generated

## Logical Model

A screenshot of a cell phone

Description automatically generated

## Physical Database Design

### What is my debt to income ratio?

Two ways to achieve

create view debt\_to\_income\_ratios as

select

debt.users\_id,

round(debt.current\_total\_debt/(inc.discretionary\_monthly\_income \*12),2) as debt\_to\_income\_ratio

from

dbo.user\_debt debt

join

dbo.user\_incomes inc on debt.users\_id = inc.users\_id

go

if object\_id('dbo.user\_debt\_to\_income') is not null

drop function dbo.user\_debt\_to\_income

go

/\*

Find the debt to income ratio of every user

with debt accounts and a cash flow in the

database.

\*/

create function dbo.user\_debt\_to\_income(@userID int)

returns decimal(6,2) as

begin

declare @returnVal decimal(6,2)

/\*

return a view that calculates the debt to income

ratio for each user. Find only ratio given for the

passed userID.

\*/

select

@returnVal = debt\_to\_income\_ratio

from

dbo.debt\_to\_income\_ratios

where users\_id = @userID

return @returnVal

end

go

-- Run the function for all users

select

users\_id,

first\_name,

last\_name,

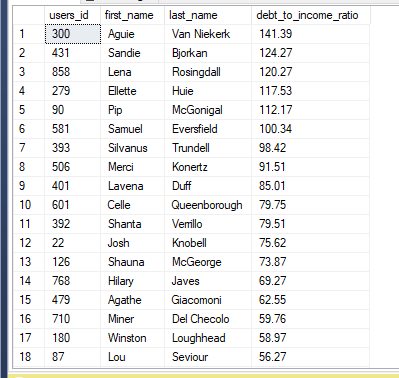
dbo.user\_debt\_to\_income(users\_id) as debt\_to\_income\_ratio

from

users

order by debt\_to\_income\_ratio desc

go



### What percent of my monthly income am I spending?

### How has spending changed over time?

Used the below query in python and looked at user 650’s spending behavior over time.

select

\*,

row\_number() over (partition by cash\_flow\_updates.cash\_flow\_id order by

cash\_flow\_updates.cash\_flow\_update\_date desc) as RowNum

from

cash\_flow\_updates

join

cash\_flow on cash\_flow.cash\_flow\_id = cash\_flow\_updates.cash\_flow\_id

A close up of a map

Description automatically generated

### How has my debt changed over time?

### What is my debt to bank account balance ratio?

### How much interest am I paying on my debt every year?