**Documentation**

# Database

A database is a collection of information organized in such a way that it can be easily accessed, managed and updated. Data is organized into rows, columns and tables, and is indexed to render it easier to find information.

## Member functions

* (constructor)
* (destructor)
* open
* query
* close

## Database::database(char\* filename)

1. Constructs a database and sets it equal to NULL
2. opens the database with name filename

## Database::~database()

1. Nothing really happens

## bool Database::open(char\* filename)

1. Checks to see if the database constructed is good; returns true if successful, else false
   1. if(sqlite3\_open(filename,&database) == SQLITE\_OK)

## vector<vector<std::string>> Database::query(char\* query)

What the fuck is going on here/

## Void Database::close()

1. Closes the database

# StockStallion

StockStallion is the premier choice in checking current prices and creating personal private portfolios for your personal investment choices. Unlike other services that require you to sign up with a bank account StockStallion is free and anonymous so do whatever you like with it without fearing the consequences of blowing your money before the market decides to dive lower than manic depression.

## Member functions

* (constructor)
* View functions
  + commandLineLoginRegisterView
  + portfolioView
* Prompts
  + loginRegisterPrompt
  + InitializeDB
  + addUserToDB
  + verifyLogin
  + authorizeLogin
* Input handling functions
  + verifyUsername
  + verifyPassword
  + verifyChoiceInRange
  + registerNewUser

## Constructor

1. Initializes a database by calling the InitializeDB method
2. Returns a prompt

## Void StockStallion::commandLineLoginRegisterView

1. Creates a boolean value initialized to false
2. While said value is false will run a while loop by checking an integer value inputted by the user using the loginRegisterPrompt method
3. If 1 then will call the authorizeLogin method and break out of the loop afterwards.
4. If 2 then will call the registerNewUser method and break out of the loop afterwards.
5. If 3 will return a prompt and end with exit code 0

## Int StockStallion::loginRegisterPrompt

1. Starts by returning a prompt asking the user to Register (1), Login (2) or Exit (3)
2. Creates an integer value and a boolean
3. Checks to see if the value is valid and an integer. If not will loop until a proper value is given
4. Returns the value the user has input

## Void StockStallion::initializeDB

1. Constructs a Database object with the string literal "stockstallion.db"
2. Creates another character pointer q0
3. Calls the query method from the current Database object using this new pointer
4. Closes the database

## Void StockStallion::addUserToDB(std::string username, std::string password)

1. Constructs a Database object
2. Constructs a string with some random value
3. Constructs a string with an empty space
4. Constructs another string using the previous two strings
5. Creates a character pointer
6. Calls the query method from the current Database object using this new pointer
7. Closes the database

## Void StockStallion::verifyLogin(std::string username, std::string password)

1. Constructs a Database object
2. Creates a string
3. Creates a character pointer
4. Makes a vector of a vector of strings that holds query results (result)from current database object
5. If the vector (result) is empty then will return a prompt saying incorrect username and password combination, closes the database and ends the method
6. If the vector contains a value then an iterator is constructed and will iterate through the vector
7. Then checks if the values at row(0) and row(1) match the username and password and then creates a static object before ending the method

## Bool StockStallion::verifyUsername(std::string &username)

1. Creates two counters for characters and integers
2. Loops through the string counting every occurence of integers and characters
3. If any other character occurs will automatically return false
4. Checks to see if string meets size requirements
5. Checks to see if it has one character and one integer value
6. If passes checks returns true, else returns false

## Bool StockStallion::verifyPassword(std::string &password)

1. See StockStallion::verifyPassword

## Bool StockStallion::verifyChoiceinRange(int choice, int max)

1. If choice is less than 0 returns false \*\*won't this method fail if user inputs 0?
2. Else returns value of expression choice <= max

## Void StockStallion::registerNewUser()

1. Prints a string welcoming the user to the program
2. Creates two string variables named username and password
3. Asks the user to enter a username and then calls the verifyUsername method
4. Asks the user to enter a password and then calls the verifyPassword method
5. Calls addUserToDB method
6. Prints a string congratulating user for registering

## Bool StockStallion::authorizeLogin()

1. Prints a string welcoming user to login prompt
2. Creates two string variables
3. Asks user to enter username and password
4. Verifies user inputs using verifyUsername and verifyPassword methods
5. Calls the verifyLogin method