

## Project Data Layer And Scaffolding

### Class Project Overview

Your class project will consist of the design, development, testing and debugging of a Rails web application of your choice. Here you get to solve a problem of interest to you and you get to be creative. The project does not have to be anything fancy, but I do want it to involve some kind of database table. It can be most anything that includes Ruby and Rails. I hope you'll pick something that will be fun. There will be multiple assignments that deal with your class project as the semester progresses.

### Project Data Layer

At this time, I would like you to identify and describe the information your class project will be storing in its database. For example, if my class project were about my favorite songs and singers, I would create the following for this assignment:

Field Name	Description	Kind
singerName	The person who sang the song	string
songTitle	The name of the song they sang	string
releaseYear	The year when the song got released	integer
albumTitle	The name of the album which first released this song	integer
favorite	Is this song really one of my favorites??	boolean
songLength	The time length of the song, in seconds	boolean

In addition, I would like you to share the scaffolds command you used to initially build this project. For example, if my class project were about my favorite songs and singers and based on the table I created above, I would enter the following command for this assignment:

```
rails generate scaffold  
song singername:string songtitle:string releaseyear:integer albumtitle:string  
  favorite:boolean songlength:integer
```

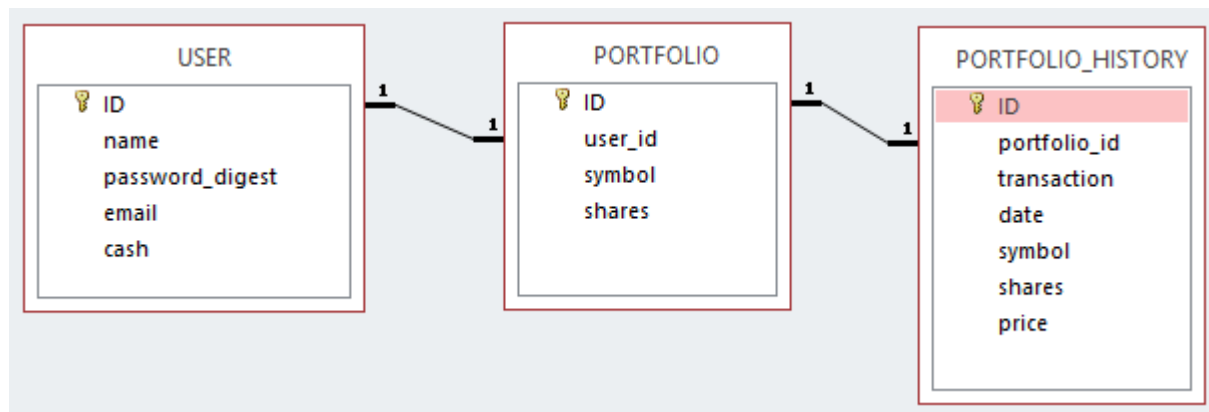
The page here will give you the ability to enter this information right onto this assignment page below.

## Project Data Layer (FinanceApp)

USER		
Field Name	Description	Type
ID (PK)		integer
name	The user's username	string
password_digest	The user's password	string
email	The user's email	string
cash	The amount of cash	decimal
rails generate model user name:string password_digest:string email:string cash:decimal		
\FinanceApp\app\models\user.rb		
<pre>class User &lt; ActiveRecord::Base   # A has_one association sets up a one-to-one connection with   # another model.   # Each user has only one portfolio.   has_one :portfolio    # A has_one :through association sets up a one-to-one connection   # with   # another model by proceeding through a third model. Each user   # has one   # portfolio, and each portfolio is associated with one portfolio   # history.   has_one :portfolio_history, :through =&gt; :portfolio end</pre>		

PORTFOLIO ( the collection of stocks that user own)		
Field Name	Description	Type
ID (PK)		integer
user_id (FK)	The user's ID.	integer
symbol	The stock's symbol.	string
shares	How many shares a user owns of a particular stock.	integer
rails generate model portfolio user:references symbol:string shares:integer		
\FinanceApp\app\models\portfolio.rb		
<pre>class Portfolio &lt; ActiveRecord::Base   # Sets up a one-to-one connection with another model.   # Each portfolio can be assigned to exactly one user.   belongs_to :user    # A has_one association sets up a one-to-one connection with   # another model.   # Each portfolio has only one history (portfolio history).   has_one :portfolio_history end</pre>		

PORTFOLIO_HISTORY		
Field Name	Description	Type
ID (PK)		integer
portfolio_id (FK)	The portfolio's ID	integer
transaction	Whether a stock was bought or sold	string
date	The price of a share at the time of transaction.	date
symbol	The symbol bought or sold.	string
shares	The number of shares bought or sold.	integer
price	The date and time of the transaction.	decimal
rails generate model portfolio_history portfolio:references transaction:string date:timestamps symbol:string shares:integer price:decimal		
\FinanceApp\app\models\portfolio_history.rb		
<pre> class PortfolioHistory &lt; ActiveRecord::Base   # Sets up a one-to-one connection with another model.   # Each (portfolio) history can be assigned to exactly one   # portfolio.   belongs_to :portfolio_history end </pre>		



**Relational schema** (All relations are of ONE-TO-ONE type).