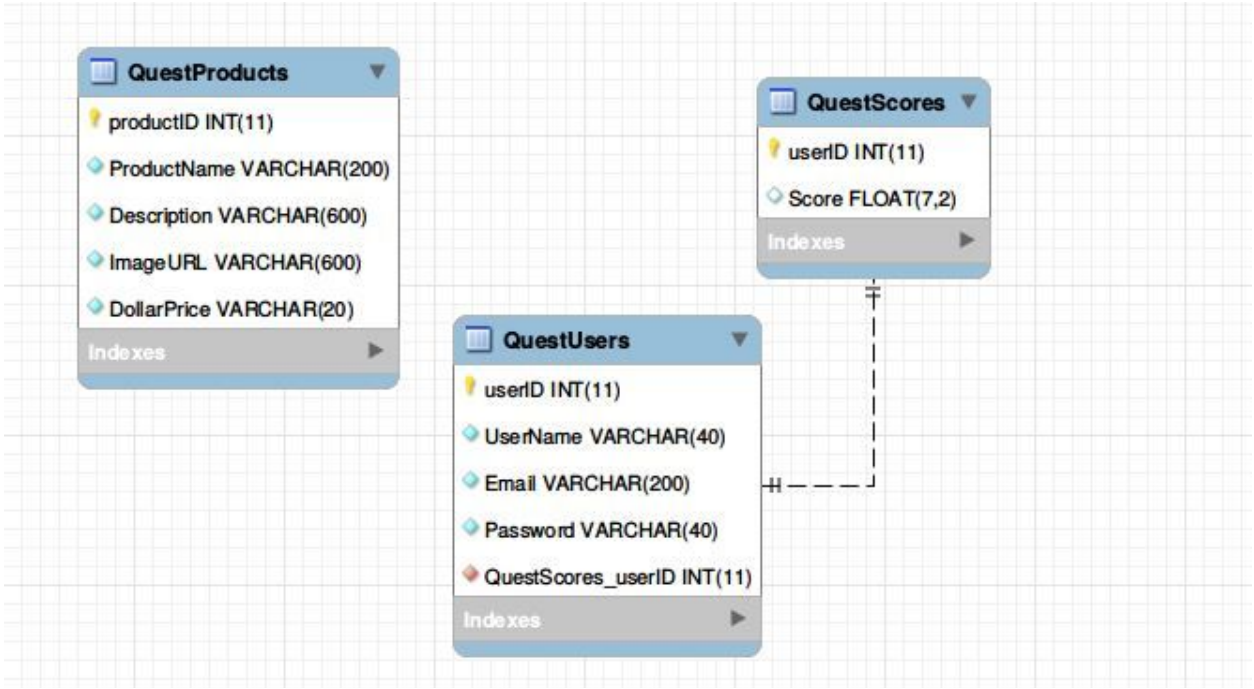
CostQuest Database Design

We will be using MySQL as our DBMS software. Our project requires three tables. The first table, called “QuestProducts”, contains the data about all of the products that will be included in our price guessing game. The columns of this database hold the product ID, the product name, the product description, an image URL for the product, and finally, the actual price of the object. All of the objects that are placed in the QuestProducts database were found on Amazon and manually entered into the MySQL table.

The second table is called “QuestUsers”. This table is comprised of a user ID, a username, an email, and a password. Whenever a user wants to log into our CostQuest game, we will use this table to make sure that the user can be authenticated. If the user is not found in this table, they will not be able to sign into the game. The QuestUsers table is related to a third table, called “QuestScores”, that tracks the current user and updates their score according to how well they do in the game. The QuestScores table is comprised of two columns. The first column is the user ID, and the second is that users score.

The image below is a visual representation of our database design. Notice that the QuestUsers table and the QuestScores table have a one to one relationship. This means that one user can have a single score that is stored, and one score corresponds to a singular user.



The sql scripts for our database can be found in the Cost_Quest github repository under the Database branch. Please use the following link:
https://github.com/Zasmussen/Cost_Quest/tree/Database

Path for QuestProducts table: Cost_Quest/Code/src/QuestProducts.sql
Path for QuestUsers table: Cost_Quest/Code/src/QuestUsers.sql
Path for QuestScores table: Cost_Quest/Code/src/QuestScores.sql

Each of the database tables has been completely filled out. The data inside QuestProducts and QuestUsers will not be subject to change. However, the data inside QuestScores will change as a user plays the game and their score is updated. Since we are still in the developmental phase, the QuestScores table was pre-populated and the data inside this table will be used for testing purposes.

Below, we have included screenshots of each of the database tables in MySQL. Since the QuestProducts table is so long, we have only included the first few rows. We also broke it into two so that the text would be large enough to read.

QuestProducts
(first 3 rows)

```
mysql> SELECT * FROM QuestProducts LIMIT 3;
+-----------+---------------------+---------------------------------+----------
| productID | ProductName         | Description                     | ImageURL
+-----------+---------------------+---------------------------------+----------
|         1 | Acer Aspire Laptop  | 15.6" Full HD Widescreen Display | https://
|         2 | Acer Aspire Notebook | 15.6" Full HD Widescreen Display | https://
|         3 | ASUS Android Tablet | 8" Touch Screen                 | https://
+-----------+---------------------+---------------------------------+----------
3 rows in set (0.00 sec)
```

```
+-----------------------------------------------------------------------------+-------------+
| ImageURL                                                                    | DollarPrice |
+-----------------------------------------------------------------------------+-------------+
| https://images-na.ssl-images-amazon.com/images/I/71Tl%2BD5RcNL._SL1500_.jpg | $799.99     |
| https://images-na.ssl-images-amazon.com/images/I/61Yeir0uhIL._SL1322_.jpg   | $349.99     |
| https://images-na.ssl-images-amazon.com/images/I/41JYmgGBhRL.jpg            | $126.00     |
+-----------------------------------------------------------------------------+-------------+
```

QuestUsers

```
mysql> SELECT * FROM QuestUsers;
+--------+----------+---------------------------------+----------+
| userID | UserName | Email                           | Password |
+--------+----------+---------------------------------+----------+
|      1 | Bekah    | reha8209@colorado.edu           | Bekah    |
|      2 | Jorge    | jorge.pulidolopez@colorado.edu  | Jorge    |
|      3 | Zach     | zachary.asmussen@colorado.edu   | Zach     |
|      4 | Haotian  | haotian.zheng@colorado.edu      | Haotian  |
|      5 | Theo     | theodore.margoles@colorado.edu  | Theo     |
|      6 | Binpeng  | binpeng.wu@colorado.edu         | Binpeng  |
+--------+----------+---------------------------------+----------+
6 rows in set (0.00 sec)
```

QuestScores

```
mysql> SELECT * FROM QuestScores;
+--------+--------+
| userID | Score  |
+--------+--------+
|      1 | 123.40 |
|      2 |   5.67 |
|      3 |  89.00 |
|      4 |  12.30 |
|      5 |  45.60 |
|      6 |  78.90 |
+--------+--------+
6 rows in set (0.00 sec)
```

*** Please note that all of our code for the login and security measures is on the “Interface” branch of our github CostQuest repository.