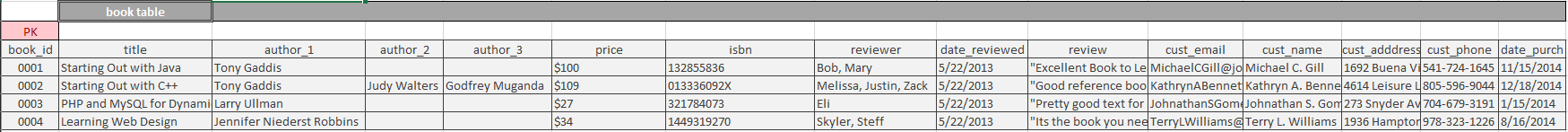
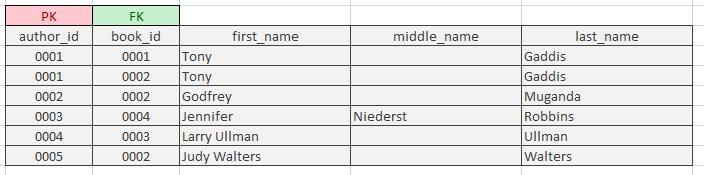
**Final Project**

*This assignment will be completed using a spreadsheet or word document table. Perform the necessary steps to normalize a database to Third Normal Form*

1. *Use the data file as a starting point for your database design.  
   The file contains some sample data from the Fake online bookstore.  
   Analyze the data to determine the meaning of the data in order to determine dependencies.  
   Also use your judgment and past experience in the process of identifying data meaning.*
2. *Normalize the database to 3NF. Create a spreadsheet showing your 3NF tables including the data.  
   Under your normalized tables, itemize the steps you took during the process of normalization.  
   For each step describe briefly the process.  
   The items for 2NF and 3NF should identify the dependencies involved.*
3. *Once the database in correctly normalized to 3NF submit the document for grading.*
4. Put the “Fake Online Bookstore” dataset in RDBMS format style with lower case table named “book” with arbitrary (no alphabetization) primary key “book\_id” and column names with words separated by underscores.
   1. fkdsjhg



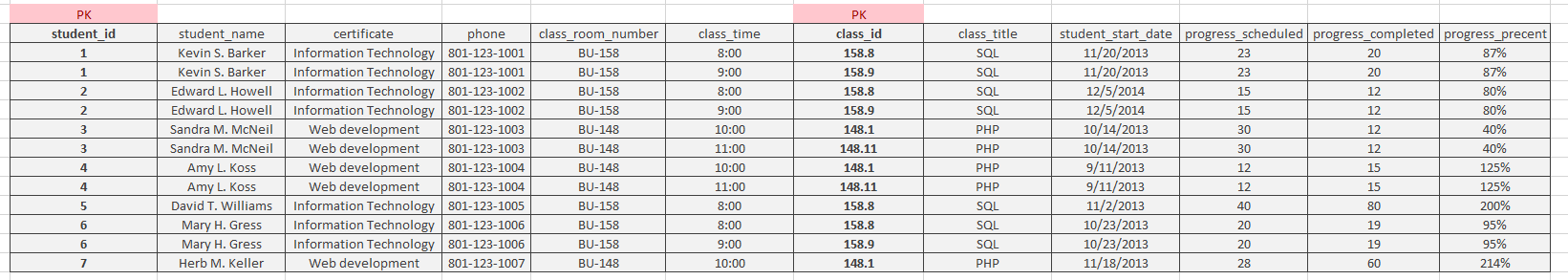
1. *All repeating groups must be removed and placed in new tables* *(1NF)*
   1. Create an “author” table from original author columns (“Author1,” “Author2,” “Author3”) with primary key column “author\_id;” “author\_id” column enumerated when original “Author” column separated into “first\_name,” “middle\_name,” and “last\_name” columns (so that *each column only has only single value (rule 4)*) and alphabetized by “last\_name.”
   2. Gave each author a “book\_id” foreign key column (coming from the initial enumeration of the book table) corresponding to the books they have published or co-published.



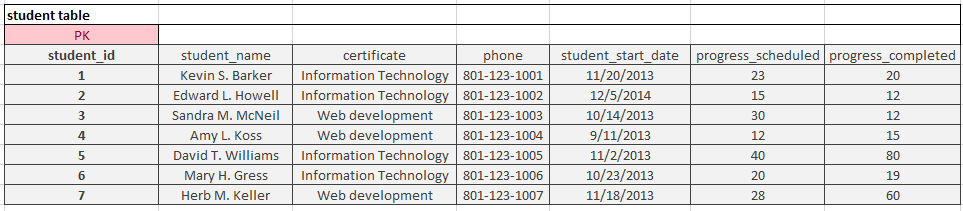
1. Separated out book table from author, reviewer and customer information.
   1. Created “book” table with “book\_id” primary key, and “author\_id” and “reviewer\_id” foreign keys.
   2. Created “reviewer” table with primary key column “reviewer\_id” enumerated when original “reviewer” column (with potentially several reviewers) separated into “first\_name” column (so that *each column only has only single value (rule 4)*) and alphabetized.
   3. Create an “customer” table from original “cust\_name” column with primary key column “customer\_id” enumerated when original “cust\_name” column separated into “first\_name”, “middle\_initial,” and “last\_name” columns (so that *each column only has only single value (rule 4)*) and alphabetized by “last\_name.”

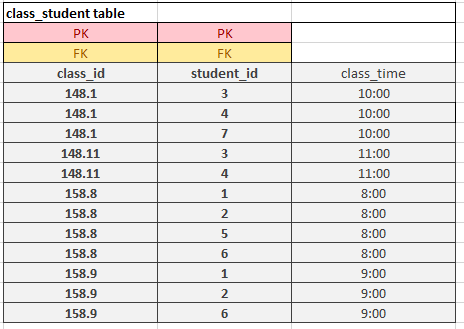
**A:01 Database Design and SQL**

1. Put the data in RDBMS format style with lower case table named “student” and column names with words separated by underscores.
2. Label the two primary keys that make up the concatenated, or composite key.



1. Determine if columns may exist without one or the other part of the concatenated keys (partial dependencies).
2. Discard “progress\_percentage” as a derived column of progress\_completed divided by progress\_scheduled.
3. Create another table as several of the columns depend on only one part of our concatenated key.





1. And create an intersection table to resolve the many to many relationships of students to classes and classes to students.

