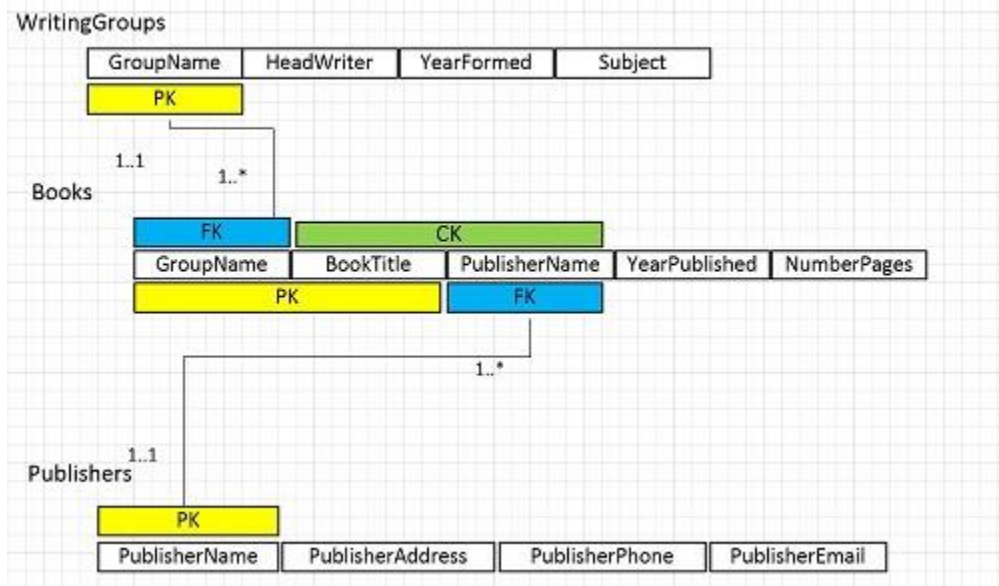
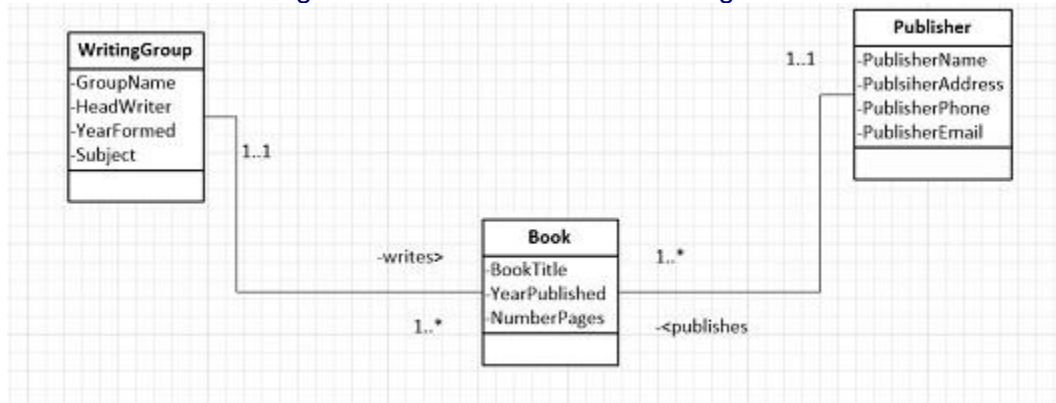


JDBC Project

- With the ease of communications that we enjoy today, it is much easier to collaborate on writing a book than ever. It is so easy in fact, that many authors have formed "writing groups". Each writing group has a unique name and a head writer.
- You will form groups of two for this project. Your project should allow for interaction with the user so the user may specified the required data. You need to create a database for the following class diagram/relation scheme.



- Create a .sql file for the DDL and run this script to create the database, tables, etc.
- Write a JDBC program to support the following functions:
 - List all writing groups
 - List all the data for a group specified by the user
 - List all publishers
 - List all the data for a publisher specified by the user
 - List all book titles
 - List all the data for a book specified by the user
 - Insert a new book

- Insert a new publisher and update all **books** published by one publisher to be published by the new publisher. Leave the old publisher alone, just modify the books that they have published.
 - Remove a book specified by the user
- For all queries involving user input, you must use prepared statements
- You must be able to prove your results after each query
- Make sure to validate the data either through the java code or database constraints
 - Don't worry about bullet-proofing the application itself. If you prompt the user for an integer menu choice, assume that they will enter an integer.
 - But you **do** want to handle any input from the user that might cause an integrity violation in the database.
- Make sure you handle any SQLExceptions that are thrown
- You must use your NetBeans/Derby database for this project
- Make sure you have enough sample data to properly demonstrate your project but don't go overboard. A half dozen rows in each table should be adequate.
- The data can be fictional but it must be meaningful. I do **not** want to see Book One, Book Two, etc.

Deliverables

- Java code written to support this project
- DDL used to create the tables, keys and other constraints
- DML for your sample data
- Examples of the output for each query stated above.
 - Please do this as a .txt file. Simply run through a test of your JDBC application, use ctrl-a to select all of the text in your NetBeans console window, and then paste it into a .txt file. Screenshots are often hard to read, and I would prefer to have just one file rather than nine of them to review.
 - I want all of the dialog between the user and the application shown in the sample output.
- Live demonstration of all project functionality.

Quality Checks

- If something goes wrong in any of these operations, it is your job to tell the user exactly what went wrong.
 - For instance, when inserting a new book, you have to tell them specifically whether it is the publisher, the writing group, or both that is missing if it is a foreign key violation.
 - If the user attempts to insert a duplicate in any of your tables, you need to tell the user that there is already a row in the table with those key column values.
- If your application **prevents** errors from occurring in the first place, then try/catch blocks are not required.
- For each report that you produce, be sure to order the rows by some column(s) that will make it easy for your user to find a specific row.

Latest Update: 2/28/2018 4:27 PM

David.brown@csulb.edu

