# Introduction

For this project, a musical instrument company requested that the way data is handled be changed. Currently, they are storing their data within a Microsoft Excel spreadsheet, and would like to change to a more professional database. This can be achieved by creating a database, along with a program that will pull information from the database and print it to the console. In order to design this database, the data given within the excel spreadsheet must be normalized. From this normalization, an ERD can be created. When this ERD is created, a database is able to be created using the relationships formed in the ERD. Finally, making a java program that can read information from the database and print it into a console is required.

(blue = filled field by author, yellow = pre-filled)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** |  | **Functional Requirement** |  | **Value** | **Stakeholder** |
| FID001 | **I want to** | Normalize data given in Excel Spreadsheet | **so that** | The relationships make sense while being as separated as possible | Student |
| FID002 | **I want to** | Create an ERD | **so that** | The relationships are shown visually, using crows foot notation to show cardinality | student |
| FID003 | **I want to** | Import data from the spreadsheet to the database | **so that** | There will be data in the database | student |
| FID004 | **I want to** | Create a java program that can connect to the database | **so that** | Direct access to the database isn’t required for people to see what’s in the database | Client |
| FID005 | **I want to** | Print the data to console | **so that** | Employees can see the order history | Client |

# Analysis

In this project, normalization was the first step. The data was able to be consolidated into five tables in order to reach 3NF. These five tables were CUSTOMER, ORDER, PAYMENT, PRODUCT, and LINEITEM. LINEITEM is an associative table between PRODUCT and ORDER to get rid of the one:many to one:many relationship that was originally there. From this, the ERD was basic to set up, all that needed to be added were the data types that are associated to each attribute. From here, the next step was creating the database. This is rather simple, but without any way to import the data, it was time consuming. Finally, creating the java program would be a way to fulfil the last two functional requirements. Using the data access management layer (DAM), a connection from the java program to the database was able to be established. Within each DAM file, each entity had a SQL query that would pull the items in the respective entity. In the class folder, there are constructors, getters, and setters for each entity, allowing the respective DAM file to pull attributes from each entity and call functions to print the attributes to the console.

# Conclusion

This program, when working, will use the OrderNumber to print out all of the associated attributes to the order. It is possible to change what’s printed out by simply changing the order number. It is also possible to change the login destination of the program, and if by some miracle, another database has the same table names, the user can print out the data for those orders as well.

This project was a ton of fun, and was probably the least-frustrating one that I’ve had to date. That’s progress, right? I ran into a glaring issue that causes the program to not run correctly, so I can’t really see that the program prints out what it’s supposed to. The login to the server seems to be wrong, but I’ve triple checked, and the information is correct. Other than that issue, the program compiles, and tries to connect but fails the connection. Overall, this project was really satisfying, because I think everything clicked (and I would be able to double-check this if the program actually ran). I’m sure there are mistakes in the code, but it was an actual enjoyable experience. I really learned a lot about databases this semester, and I hope that this project shows that. Thank you for teaching me these last two semesters, especially given such short notice that you would be in the position that you are in. You really are an excellent instructor, and have taught me more than I would think possible in such a short amount of time.