The Aerialist Database

Jacqueline Lincroft

Northeastern University, Boston, MA, USA

# Abstract

In writing up your report, follow this template exactly. Your report will be compiled into the database design class proceedings. Future students will learn from your work. In the abstract, provide a summary overview of your project, its goals and accomplishments.

# Introduction

One of the things I personally miss most about the days before quarantine, was my weekly journey on the T to Jamaica Plains, where I loved taking classes at Commonwealth Circus Center. I’ve been doing aerial arts since I was in high school, taking breaks here and there, switching studios, and one of the things I’ve noticed is a lack of consistency or resources for aerialists. Since aerial arts are such a comparatively young discipline, compared to similar arts like gymnastics, dance, or other circus arts, new skills and apparatuses are constantly being invented, renamed, and shared – the collaborative and innovative nature of the practice is one of my favorite things about the circus community.

For my final project, I wanted to create an original database to store everything related to aerial arts – moves/skills, apparatuses, vocabulary, and more. The big picture for this project would be to make the database accessible and open source for aerialists all over to use and contribute too, through a mobile or web application.

The bulk of the data in this project is stored as moves, which are distinct skills performed on an apparatus, and can be things like climbs, drops, or poses. Because skills can build off of each other in an aerial routine, I designed the table to contain a foreign key to the same table, creating a sort of recursive table. Moves can have different attributes, such as difficulty level, whether they’re or not they’re dynamic, and whether they require inverting (a skill that can sometimes take time for beginners to master). All of these features allow for flexible usability in many different cases, such as performers planning a routine, beginners looking for new moves to learn, or teachers planning a class or private lesson.

In addition to the moves stored, there are a few other features of this database. There is a table to store apparatuses, since the umbrella term of aerial arts encompasses many different apparatuses, including silks/tissue, trapeze, or lyra/hoop, to name a few. As shown by just my brief listing of examples, aerial apparatuses can also go by many different names, depending on the region, type of practice, or even just varying from studio to studio. Therefore, I also included functionality to store synonyms for both apparatuses and moves in my database, to make the project as accessible to as many as possible. And finally, I allowed my database to store users of the application as well – although the database could be a useful feature for anonymous guests as well, if the application were to include login and account functionality, users could potentially bookmark favorite moves, or create lists of moves while planning an act.

Because of the time constraints of the course, I did not have time to finish a complete application to interact with the database. Instead, I created an initial skeleton for the Java backend, and used created wireframes for what a potential mobile application could look like, using Figma and Adobe Illustrator. Both of these components are my attempt to show the full extent of what this project could become, but are by no means perfectly polished or complete.

# Database Design

The core of my database schema is the **move** table. This table stores all key information for a move, and references many of the other tables. It references the table **move\_type**, a very simple table that enumerates the different kinds of moves that can be stored, such as climbs, spins, poses, and splits, and it references the table **apparatus**, which stores the different apparatuses in the database, including silks, trapeze, and rope, to name a few. The **move** table also has a foreign key reference to itself, seen in the **builds\_off** column– this is to allow for the recursive nature of these skills, that can build off of each other, and connect together. This also allows for one of my personal favorite use cases, designing sequences or routines by picking a path through the tree of moves that build off of each other.

Some of the other key tables in my database include the **apparatus** table, which stores apparatuses and a few key attributes about them, as well as the **user** table, to store information about users that may download or log into the application. There is also a join table, **user\_apparatus­**, which allows users and apparatuses to have a many-to-many relationship – this is so that a user can mark multiple apparatuses that they train on (as many aerialists do indeed train on several apparatuses), and of course multiple users can also use the same apparatus.

Finally, the last tables I wanted to mention are the **move\_synonym** and **apparatus\_synonym** tables. Because aerial is such a young and constantly evolving discipline, with terminology varying by region or just randomly, I wanted to make sure my database could store multiple names for the same apparatus or move. In order to allow for this while maintaining database normalization, I made separate tables to store the synonyms, which reference the move/attribute they describe as a foreign key.

Present and describe your database design. Include a figure with your ER Diagram / Conceptual Model. Explain the *key* entities and their relationships. Use the MySQL modeling tool to create your model. Make sure the image of the model is fully readable: the entities should be neatly laid out and all relationships should be fully visible. We’ll assess the quality of your design and its effectiveness in addressing your project requirements.

# Data Sources and Methods

To acquire my data for this project, I used my own experience from the past several years learning aerial arts, and consulted a few friends as well, and manually inserted entries into the various tables. Because my experience is mostly limited to silks, trapeze, and lyra, those are the apparatuses I had the most entries for, although I tried to add a few entries for some other apparatuses as well. The data currently stored in the schema could be considered somewhere in between mock and real, since although I didn’t make it up, my memory could certainly be imperfect, and many features of the data are entirely subjective, such as difficulty level or whether or not a move qualifies as dynamic or static. The data provided simply shows what this database could look like after open source contributions from aerialists all over.

# User Cases

List specific non-trivial questions that users could ask of your database. Summarize the question in English, provide a corresponding SQL query, and display your tabular output. Where appropriate, you should supplement your tabular output with charts, graphs, or other types of visualizations in order to better convey key insights.

# Conclusions

Summarize the results of your project. Be concrete about your accomplishments as well as the limitations of your work.

# Author Contributions

If you worked in a team, describe how each member of your group contributed to the success of your project. There are many ways to make meaningful contributions to a project. I don’t expect each person to contribute to each aspect of the project. One person may have been more focused on database design while another performed the bulk of the data cleaning operations.

# References

1. Adibuzzaman, M., et al., *Big data in healthcare - the promises, challenges and opportunities from a research perspective: A case study with a model database.* AMIA Annu Symp Proc, 2017. **2017**: p. 384-392.

2. Murthy S., A.R., Goodwin R., Keskinocak P. Rachlin J., Wu F., Yeh J., Fuhrer R., Kumaran S., Aggarwal A., Sturzenbecker M., Jayaraman R., Daigle R., *Cooperative Multiobjective Decision Support for the Paper Industry.* Interfaces, 1999. **29**(5): p. 5-30.

3. Rachlin, J., et al., *Biological context networks: a mosaic view of the interactome.* Mol Syst Biol, 2006. **2**: p. 66.