For our League of Legends counter pick application, we originally planned to use a database to store all of the information required for our selection algorithm. This included every champion’s basic information, possible team roles, and relative match strength versus every other champion. Champion information and roles were retrieved from leagueoflegends.net (League of Legends’ official website) and stored in the Champions and Roles tables respectively. Relative match strengths were to be determined using win percentages of online ranked matches. These percentages were retrieved from elophant.com and stored in the Matchups table. The relational models and schemas for this version of the database can be found elsewhere in this report (marked as “before refining”).

In the original version of the database, the data representing champion’s relative strengths were treated as a relation between one champion and another; so, each relation between two champions was its own tuple in the Matchups table. This meant the Matchups table would contain n2 tuples where n is the number of champions (currently ~120). As a result, some queries were taking too long for our application to compute the optimal selection in less than one second (one of our functional requirements). Thus, the database needed to be refined.

In order to minimize the amount of time for queries to be processed, the number of tuples in the Matchups table needed to be reduced significantly (query complexity is based partially on the number of tuples to be searched). Because it’s possible for any champion to oppose any other champion, the Matchups relation has total participation for all champions. Thus, the data represented in the Matchups relation could also be represented using a composite attribute on the Champion entity. This composite attribute would contain a separate atomic attribute for every champion and would indicate the relative strength of the entity against that particular champion. This change was implemented in the second version of the database, resulting in the refined relational models and schemas seen elsewhere in this report.

The refined version of the database was equivalent to the original in terms of functionality; both contained the exact same data. However, refining improved the database in both needed storage and retrieval speed. While the Champions and Roles tables were unaltered, meaning they required the same amount of storage space in both versions, the Matchups table, which was by far the largest, was reduced in size by roughly a factor of 10. This is largely because the champion’s names took the most space to store and the original version repeated each champion’s name for every relation in which it took part. Smaller storage space is helpful, but the more important benefit of the database refinement was its speed. The new way the matchups table was stored reduced its number of tuples by two orders of magnitude, giving it fewer than were required for the Roles table. This reduced the slowest query time from about .6 seconds to a little over .01 seconds.