

Chapter - Database

In our project we used a graph database instead of a relational database.

This was due to the fact, that relational databases are much faster when it comes down to look for objects in a list that fulfill some constraints to other objects. Normally in a relational database you would use multiple joins to get your desired result. In the graph way it's much easier because you traverse the graph (previously mentioned as list) from top down using either breadth- or depth-first algorithm and look for a specific relation between two nodes (previously objects). In respect of that we save a lot of time traversing instead of writing complicated queries which run mostly slow.

Therefore we use Neo4J (Additional info can be found under <http://neo4j.com/developer/graph-db-vs-rdbms/>). Neo4J is a graph database that is capable of managing millions of nodes and relationships and returning or changing them within logarithmic or even constant time. That means that it won't make a big difference whether you use 10 nodes, or 1 million. You can find additional information on benchmarking in the related chapter of this documentation.

Our database is mainly one big graph. We use different Labels for Nodes and Relationships to increase the search speed.

For nodes we use the labels:

- ROOT
- TRADITION
- USER
- WORD
- STEMMA
- WITNESS

For relationships we use the labels:

- RELATIONSHIP
- STEMMA
- NORMAL

The database structure is as follows:



