

---

## Report Results

---

Vi har i dette projekt har haft mulighed for at arbejde med 2 forskellige Database Management System(DBMS), vores valg har været at arbejde med MySQL og MongoDB.

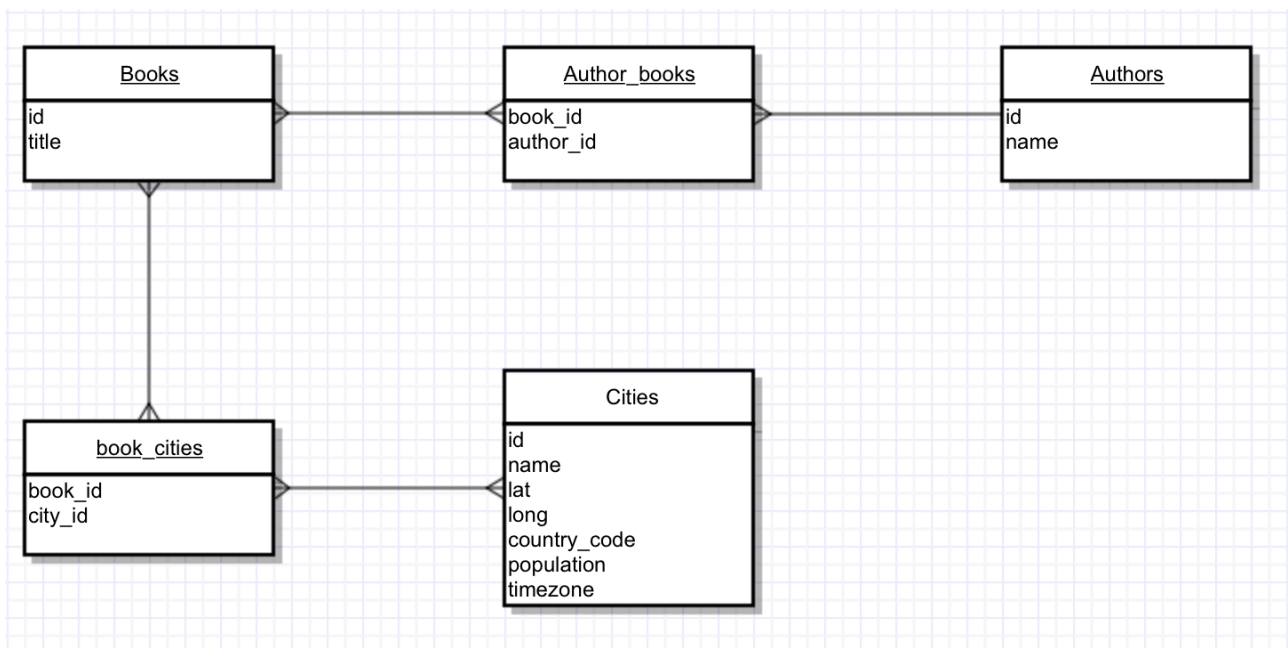
Grundet til det valg har været at, vi har følt at det ville være god ide at arbejde med en database engine vi havde erfaring indenfor, så valget MySQL kom meget naturligt.

Det andet valg for en database engine var MongoDB, her er grunden simpelthen at vi vil have mere erfaring inden for MongoDB.

Måden vi har moduleret vores data på er, vi har startet med at importere de forskellige data fra Gutenberg tekst filer, som vi har downloadet via en ekstern Linux 16.04.2 x64 version server, som vi har genereret på Digital Ocean. Her har vi brugt bash filen, som er stillet til rådighed for os.

Moduleringen er forekommet således, at vi først har scannet Gutenberg tekstfilerne og derefter importeret dataene til csv filer, som vi senere har brugt til at importere data til databaserne.

Modulering af vores data i vores csv filer har været afhængig af hvordan vi har designet MySQL databasen, vi har i denne fase brugt et E/R-diagram til at gå ud fra, som ser således ud:



Figur 1 Diagram, med 5 entiteter hvoraf 2 af dem er pivot tabeller

Da vi har 3 entiteter og 2 pivoter, som er tilsvarende til 5 csv filer, hvor alt data bliver transporteret fra disse csv filer til tabellerne.

I MongoDB har vi dog valgt ikke at bruge csv filerne ved import af data. Måden vi har grebet dette and på, er ved at hente dataene fra MySQL via vores Java applikation, og moduleret dem så de nemt kunne indsættes til MongoDB via MongoDriver, dette har vi følt vil være den bedste og hurtigste løsning.

Vi er dog klar over at vi kunne brugt mongoImport til at importere alt data fra csv filerne dog ville dette tag længere tid, derfor har vi valgt at bruge MongoDriver til at modulere vores objekter derefter importere det til MongoDB.

Vores executions mål kan findes på vores readme.md filen på Github.

Vi anbefaler helt klart MongoDB, da det har været nemmere arbejde med de her Geo locations når, dette er fordi MongoDB forskellige ting build in som gør det nemmere at skrive queries og dermed bruger man mindre tid.

Hvorimod i MySQL er de her Geo locations være lidt vanskelige, da man skal hen bruge de forskellige matematiske funktioner.

**Github: <https://github.com/bigstepdenmark/gutenbergdb>**

**Github release: <https://github.com/bigstepdenmark/gutenbergdb/releases>**