Advanced Database Management

Project Report

Team: 29Digit
Project repository: 29Digit/DigitalWorm
Final release: digitalworm.kz, Admin Panel: digitalworm.kz/admins
Credentials: a@a.com, **password**: aa

Tech. Stack and architecture of the project:

Client Server **Architecture | Backend**: Laravel, MySQL | **Frontend:** Vue JS

1. **Idea**

   We came up with an idea of reading books on the browser and developing an online bookstore with an admin panel. Summed up details, prepared use-case diagram. We designed the app primarily for smartphones, that's why it looks silly on desktop.

2. **Dataset**

   Then, we worked on the dataset. Preparation of the dataset was one of the interesting parts, aside from generating random .csv files, we'd like to generate the books in .epub format and its cover.

   To generate random .epub books we researched and found that .epub files are simply a bunch of .xhtml files, which can be parsed, for example with a *beautiful soup* python module, and replaced with other text. After completing the script, that replaces the text of the ready .epub file, we generated 5000 fake books.

   We discussed the idea of generating cover and ended up with the philosophical question "What is beauty?". To not waste time with this kind of question we used a ready module for python and generated **5000** high-quality .png files, each **~2mb size**. Which of course was the bad idea, it took a lot of space and time, *it was the first time when we faced problems with a big amount of data.* Something stopped us from simply using the public API for images...

3. **E/R Diagram and database modelling**

   After finishing E/R diagram and preparing models in Laravel, we faced problems uploading the dataset (with a minimum 5000 rows in each table, some tables had over 15000 rows) took too much time, some members could not upload due to limited resources, so they decreased the amount of data.

4. **SQL queries**

   Based on the questions that we wrote previously, we started writing sql queries. Selecting books of specific authors, searching books by title or description etc. We found that we had problems in the dataset, some item's ids were the same, some columns were missing.

5. **UI implementation**

   After implementing the frontend part, we faced problems during deployment of the project. Aside from tables with 5000 rows, we found that we don't have enough resources to store our high quality covers and .epub books on our server and our backend architecture was not designed for big amounts of data. Then we figured out that we should build another server for sending such kind of data, but we did not have enough

time and resources, so that's why we uploaded only 100 books (.epub) and covers. How did we deal with the books with *book_id* greater than 100 ?

- **book_id % 100 +1** -> on frontend

We agreed that it does not violate requirements of the project, because we should focus on manipulating database tables.

Then, connected the frontend and backend using API and tested our web application, everything worked as we expected, but it took too much to handle the requests.

6. **Calculating true cardinalities and plan evaluation, performance evaluation**

Finally, using knowledge from lectures and lab works we analyzed our SELECT queries, drew relation algebra trees and evaluated their true cardinalities and performance.

We evaluated the effectiveness of each plan in terms of the size of the connection power. To do this, we followed the following point: Request preparation Selecting an attachment order Calculate The True Connection capacities Connection Power Rating Boost and Launch Join Order Plans.

Also we measured the execution speed of the Phase 7 functions that run the SQL query. To ensure the average time, we ran each query 5 times, and the results were returned in milliseconds.

All of this was implemented using MySQL EXPLAIN ANALYZE. Which will show us where MySQL is spending time on our query and why. It will plan the query, instrument it, and execute it by counting rows and measuring the time spent at various points in the execution plan.