Project 1

Sumit Singh (2016PH10575), Prashant Singh Chandel(2016PH10566),
Prathamesh Pradeep(2016PH105600)

Due date: February 24, 2019, 11:55pm IST

1 Project Description

Our project involves building a website which functions similar to the website goodreads.com for the purpose of which we used a subset of the data of the site available on kaggle.
In the website the user can search any book available on the database using either the authorname,title name or year of publication.The user can also create his id which enables him to add a book to the database.Any user with or without an account can add a review for the books available on the database.

2 Data Sources and Statistics

We used the goodbooks database available on kaggle at https://www.kaggle.com/zygmunt/goodbooks-10k. We cleaned the data by removing the columns which were not useful to us and took only the columns bookid, smallimageurl, yearofpublication, bookscount, isbn, average rating, authors, title,language code for the books table. For the remaining tables we used all the values and replaced missing values with default values using python script. We also used python to create table for reviews with default reviews for each book. The reviews table was made as a weak entity set with the book-id as the foreign key.

Entity	**Attributes**
Books	bookid,smallimageurl,yearofpublication,bookscount,isbn,average rating,authors,title,language code
booktags	bookid,tagid,count
reviews	bookid,review,username
tags	tagid,tagname
users	userid,password

Table Name	Number of tuples	Time to load	Initial Size(Mb)	Size After Cleanup(Mb)
books	10000	208 msec	3.2	1.8
book_tags	999912	2.845 msec	16.2	16.2
tags	34252	235 msecs	.7	.7
reviews	10000	960 msecs	Self created	10

Among these users and reviews table was generated by us

3 Functionality and Working

3.1 User's View of the system

- A section containing the top 5 most rated books as per our database.
- A link to the top 20 books of last decade as per our database.
- A link to the top favourite books of the users in the database
- A drop-down menu of links to top books by year
- A search bar giving user the option of searching for the books by year, by author and by title of book
- Every view of book-list gives the user to see the details of the book on a button click.
- Every view of book-list gives the user to see the reviews for the books.
- Every view of book-list gives the user to add ones own review for a book.
- A sign-up section for new users to register into our database to get the login privileges.
- A log-in section.
- A section for logged-in users to add a new book of their choice to the database.
- A section for logged-in users to see ten books which they have selected as to-read.
- A section to browse the new books that the users have added to the database.

3.2 The Queries That Are Fired For The Above Functions

5 Top-Rated Books (and their details)	`SELECT * FROM (SELECT * FROM books ORDER BY average_rating DESC LIMIT 5)AS foo;`
20 Top-Rated Books (and their details) of Last Decade	`SELECT * FROM books WHERE original_publication_year >= 2009 ORDER BY average_rating DESC LIMIT 20;`
50 All-time Favourite Books (and their details)	`SELECT books.book_id, books.isbn13, books.authors, books.original_publication_year, books.title, books.language_code, books.average_rating FROM (SELECT * FROM viewbookidtagname WHERE tag_name ILIKE 'all-time-favourites') AS foo INNER JOIN books ON books.book_id = foo.book_id ORDER BY average_rating DESC LIMIT 50;`
Top Books (and their details) By year	`SELECT * FROM books WHERE original_publication_year = $1;`
Search For Books (and their details) By Year, Book-Name or Author-Name	`SELECT * from books where title ilike '%' || $1 || '%' and authors ilike '%' || $2 || '%' and original_publication_year=$3;`
Add Reviews For A Book	`INSERT INTO reviews VALUES($1,$2,$3);`
Sign-Up	`INSERT INTO users(username, userid) VALUES($1,$2);`
Log-In	`SELECT * FROM users WHERE username= $1 AND userid=$2;`
Add Book	`INSERT INTO userbook(userid, book_name) VALUES($1, $2);`
10 To-Read Books Of A Logged-In User	`SELECT DISTINCT title FROM toread,books WHERE user_id=$1 limit 10;`
Browse Books Added By A Logged-In User	`SELECT * FROM userbook WHERE userid=$1;`

3.3 The Run-Times Of Queries

Query	Run-Time(ms)
`SELECT * FROM (SELECT * FROM books ORDER BY average_rating DESC LIMIT 5)AS foo;`	5.073
`SELECT * FROM books WHERE original_publication_year >= 2009 ORDER BY average_rating DESC LIMIT 20;`	5.454
`SELECT books.book_id, books.isbn13, books.authors, books.original_publication_year, books.title, books.language_code, books.average_rating FROM (SELECT * FROM viewbookidtagname WHERE tag_name ILIKE 'all-time-favourites') AS foo INNER JOIN books ON books.book_id = foo.book_id ORDER BY average_rating DESC LIMIT 50;`	1056.352
`SELECT * FROM books WHERE original_publication_year = $1;`	6.466 (avg)
`SELECT * from books where title ilike '%' || $1 || '%' and authors ilike '%' || $2 || '%' and original_publication_year=$3;`	Depends on input parameters
`INSERT INTO reviews VALUES($1,$2,$3);`	0.605
`INSERT INTO users(username, userid) VALUES($1,$2);`	0.599
`SELECT * FROM users WHERE username= $1 AND userid=$2;`	11.894
`INSERT INTO userbook(userid, book_name) VALUES($1, $2);`	0.743
`SELECT DISTINCT title FROM toread,books WHERE user_id=$1 limit 10;`	143.097
`SELECT * FROM userbook WHERE userid=$1;`	0.633

Book Review Site

Entity-Relationship Diagram