



Report

Assignment 3

2DV513- Database Theory



Författare: Amelie Löwe
Date: 2020-01-12
Course: Database Theory
Course-code: 2DV513



Table of Contents

Table of Contents	2
1.0 Introduction	3
2.0 Project Idea	3
2.1 Logical model	4
2.1.1 E-R Model	4
3.1 Design in SQL and and Implementation	5
4.1 SQL queries	7
6.1.2 Youtube Link	9



1.0 Introduction

Following report describes the features and results of the Book Club project for the third Database assignment. For the assignment we were allowed to use whichever preferred relational database management systems and programming languages to solve the tasks along with designing and creating an interface to present the queries.

2.0 Project Idea

Book clubs are getting quite popular these days, however, usually the participants have to meet up physically to discuss and review the books they've just read. Perhaps this is not always possible for everyone because of various different reasons such as there might not be a suitable group in the area, high travel expenses, scheduling issues or different disabilities.

Furthermore it can be quite hard to find inspiration and books that fit everyone in the group. To solve this issue I chose to create a book club system where everyone can keep track and share the books they've just read, rating- and discussing them along with finding the perfect next book. This software can be used by larger groups or individual readers who would like to belong to a like minded community.

The target audience for this project are people who love to read and want to share this hobby with other friends or family. Furthermore book publishers could use this to gain information on which books are read and reviewed often and how the book clubs enjoyed them. This could also be of use for book clubs that want to track all of the books read and how these books performed.

Following features should be included:

- The user should be able to add a new review of a book
- The user should be able to add a new Book club
- The user should be able to see which active book clubs they could join
- The user should be able to see the top rated books added in the book club system along with the books information such as author,title, genre and comments, to be able to find a new book to read.

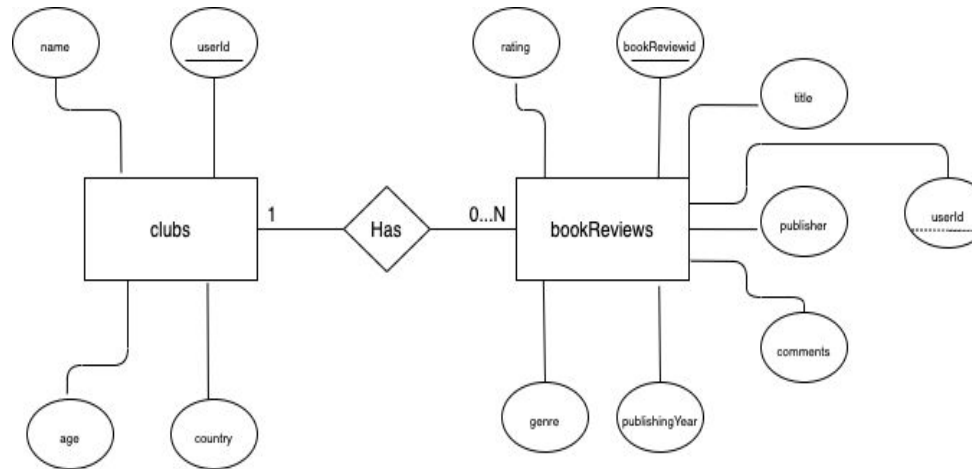
2.1 Logical model

The cardinality of the tables in relation to each other is considered “zero to many” in this entity-relationship model since every Book-club can have zero bookreview entries but also many of them. Furthermore every entity in this database must have at least one uniquely identifying attribute, which I designed it to be `userId` in the `clubs` table and `bookReviewsid` in the `bookReview` table. These are not considered weak entities and therefore have a primary key.



The attributes with a primary key are underlined in the E-R model to better visualize the importance, the same applies to my attribute bookReview.userId which is the foreign key in the system. This one is represented by dotted line.

2.1.1 E-R Model



3.1 Design in SQL and and Implementation

The connection to the database was created by providing a host, port, user and password. Differently from the previous assignment the host was an ip address instead of just localhost. For this section we used a w3schools tutorial as a source and inspiration material- "Node.js MySQL - Get Started, Link: https://www.w3schools.com/nodejs/nodejs_mysql.asp"

```
var con = mysql.createConnection({
  host: '127.0.0.1',
  port: 3306,
  user: 'root',
  password: 'Ulrik123456',
  database: 'sys'
})
con.connect(function (err) {
  if (err) {
    console.log('IT DIDNT WORK')
    throw err
  } else {
    console.log('Connected!')
  }
})
```

The design for this assignment was decided upon following thoughts: The database needed at least two tables, one for managing all of the book clubs information and one for storing the data of the book reviews. Then I group all of the similar entities together which my relations between the tables. All of the columns in the tables do have the constraints "NOT NULL" since I only wanted



real values inserted into the database. Furthermore the UserId and bookReviewId are set to be my primary keys in the table “clubs and bookReviews”. Since I needed a way of tracking each added Book-club and review. Additionally these ids were also set “ auto increment” which means that they increment the id value for each newly added club and review.

The tables were created by using the following schema:

```
CREATE TABLE clubs(  
  userId INT NOT NULL AUTO_INCREMENT,  
  name VARCHAR(20) NOT NULL,  
  age INT(11) NOT NULL,  
  country TEXT,  
  PRIMARY KEY(userId)
```

Column	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	G	Default / Expression
userId	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
name	VARCHAR(20)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
age	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
country	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
<click to edit>										

```
CREATE TABLE bookReviews(  
  bookReviewId INT NOT NULL AUTO_INCREMENT,  
  author TEXT NOT NULL,  
  title TEXT NOT NULL,  
  publisher VARCHAR(20) NOT NULL,  
  publishingYear INT(11) NOT NULL,  
  genre VARCHAR(50) NOT NULL,  
  rating INT(11) NOT NULL,  
  PRIMARY KEY(bookReviewId));
```

Column	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	G	Default / Expression
author	TEXT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
title	TEXT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
publisher	VARCHAR(20)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
publishingYear	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
genre	VARCHAR(50)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
rating	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
comments	TEXT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
bookReviewId	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
<click to edit>										

Furthermore to add an foreign key I used the following query:

```
ALTER TABLE bookReviews  
ADD FOREIGN KEY (userId) REFERENCES clubs (userId);
```



The foreign key is then later used in my queries since it links the clubs to the written book reviews, so I can fetch data from multiple tables. I used the FOREIGN KEY constraint to prevent faulty actions such as inserting erroneous data, that would destroy this link between the foreign key column in the tables.

To insert new data into my tables I used this query:

```
const sqlTable1 = `INSERT INTO bookReviews(author, title,publisher,
publishingYear,genre, rating, userId) VALUES
('${author}','${bookTitle}','${publisher}','${year}','${category}', '${ratings}',
(SELECT userId FROM clubs WHERE userId= ' '));`
```

4.1 SQL queries

The information in the tables on the website are presented by using these queries:

1. What are the top rated books
2. How many Book-clubs are in the management system.
3. How many active Book-clubs are for children (under the age of 18)
4. How many active Book-clubs are for Adults (over the age of 18)
5. Get all of the swedish clubs in the management system

The top rated books query was used to fulfill the inspiration requirement stated in the project idea section. For this I used the following statement:

```
SELECT rating, genre, author,title, comments  
FROM sys.bookReviews  
WHERE rating >=3 ORDER BY rating DESC
```

TOP RATED BOOKS

Rating	Genre	Author	Title	Comment
5	Romance	J.K Rowling	Harry potter	I loved this book etc etc
5	Romance	Maja blom	det är en gåspappa	very nice
4	Sci-fi	Jovana Lundqvist	Red dress	enjoyable
3	Memoir	Amelie Löwe	Detta är en bok	It was alright but not really my cup of tea

How many Book-clubs are in the management system.

According to the slides a database management uses aggregation functions to group values of multiple rows together to form a single more significant value. For my query I used the standard function COUNT() to count the number of rows in my clubs table . This returned the amount of Book-clubs added in the book-club management system.

```
SELECT COUNT(*) FROM sys.clubs
```



BOOK-CLUBS:

11

“How many active Book-clubs are for children (under the age of 18)“ and “How many active Book-clubs are for Adults (over the age of 18)”

I've used the Join clause to combine columns from multiple tables in order to fulfill the following requirements : “At least 2 queries should query data from more than one table, i.e., you should use at least two multirelation queries 2. You should make use of SQL JOIN”. These queries select the active book-clubs and showcases them on the web page into two tables. By the statement ” active” I mean book-clubs that already have one or more review added into them. Without using the “**DISTINCT**” statement the bookclubs would be added several times, as many times as there is a book review added. The SELECT DISTINCT statement returns only distinct (different) values and disregards the duplicate values..

```
SELECT DISTINCT clubs.name,clubs.country FROM sys.clubs INNER JOIN  
sys.bookReviews ON bookReviews.userId=clubs.userId WHERE clubs.age <= 18  
ORDER BY clubs.country'
```

```
SELECT DISTINCT clubs.name, clubs.country FROM sys.clubs INNER JOIN  
sys.bookReviews ON bookReviews.userId=clubs.userId WHERE clubs.age >= 18  
ORDER BY clubs.country'
```

ACTIVE BOOKCLUBS FOR ADULTS

Boknördarna	Island
ChickenPocs	Sweden

ACTIVE BOOKCLUBS FOR CHILDREN

WeLoveBooks	America
Books123	Sweden
Socks	Sweden

Lastly, I created a View called swedishBookclubs which I then accessed to display the last statement: Get all of the swedish clubs in the management system. The view was created by adding this into my database:

```
CREATE VIEW sys.swedishBookclubs AS  
SELECT name  
FROM sys.clubs  
WHERE country = "Sweden";
```

Which I then could access by using a simple:

```
SELECT * FROM sys.swedishBookclubs' - statement in my code. This returned all  
of the bookclubs which where located in Sweden.
```



SWEDISH CLUBS TO JOIN

Amelie
BookLovers
ChickenPocs
Books123
Socks

6.1 Discussion

The setup of this database management system was quite difficult and I struggled a lot. My computer crashed a couple of times which interrupted the client server connection. This resulted in workbench errors and it couldn't startup the server again or connect to the client. To solve this issue I tried starting over by creating new tables in both a terminal version and by trying phpMyadmin which also didn't want to connect to the client like I hoped it would. The only solution I could think of was starting completely over by reinstalling Mysql as well as the workbench. This however lead to a lot of "MySQL 8.0 - Client does not support authentication protocol requested by server; consider upgrading MySQL client" errors which took a while to resolve. As a result of this I do recognize that my assignment 3 contribution is very basic and only has the bare minimum requirements.

6.1.2 Youtube Link

Link for the youtube video: <https://youtu.be/eRdUxSUBZd8>