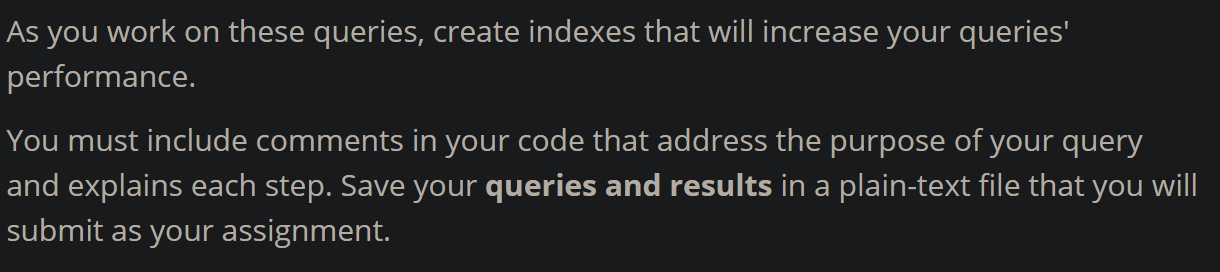
Creating & Manipulating a Database

Computer science 204

Husnan Ejaz

Abstract

The Computer Science 204 course requires building and manipulating a database in order to demonstrate knowledge of database programming. The project prompt entails creating a database for a public library, populating each table of the database with predetermined data, and then calling various queries to extract specific data requested by the assignment. Tables created in the database must have relationships highlighted by primary and foreign keys, and indexes are suggested in order to generate faster queries. This assignment requires to be submitted in the form of a text document, consisting of all the necessary queries AND the results of these queries. Because the sample data tables provided in the assignment cannot be duplicated into MySQL to populate the database, each entry must be done manually. Furthermore, the results of these queries were taken from MySQL queries, converted to a Comma Separated Value (CSV) file and pasted into this document – leading to several pages of text. In order to better illustrate the flow of the submission, highlights to easily signal out the query to make it easy for the grader to grade according to the rubric are provided. Lastly, as with any code-based project – especially one with sample data provided – plagiarism checkers may raise flags due to similar data being used. In order to combat this, comments within the code are written to further explain the reasoning behind the structure and implication of code.



Create tables for the database

/\* Part A of the assignment requires you to create the tables, define the relationships, and generate indexes \*/

# Create database if it doesn’t already exist

CREATE DATABASE IF NOT EXISTS librarySDC;

/\* Create Borrower Table, with BorrowID as Primary Key, BookID as Foreign Key from Book, ClientID as Foreign Key from Client

CREATE TABLE IF NOT EXISTS Borrower

(

BorrowID INT PRIMARY KEY,

ClientID INT,

BookID INT,

BorrowDate DATE,

FOREIGN KEY (ClientId)

REFERENCES Client(ClientId),

FOREIGN KEY (BookId)

REFERENCES Book(BookId),

/\* Create indexes for faster queries \*/

INDEX (BorrowID, ClientID, BookID, BorrowDate)

);

# Create Client Table, with ClientId as Primary Key

CREATE TABLE IF NOT EXISTS Client

(

ClientId INT PRIMARY KEY,

ClientFirstName VARCHAR(12),

ClientLastName VARCHAR(12),

ClientDoB YEAR,

Occupation VARCHAR(30),

/\* Create indexes for faster queries \*/

INDEX (ClientId, ClientFirstName, ClientLastName, ClientDOB)

);

# Create Author Table, with AuthorID as Primary Key

CREATE TABLE IF NOT EXISTS Author

(

AuthorId INT PRIMARY KEY,

AuthorFirstName VARCHAR(12),

AuthorLastName VARCHAR(12),

AuthorNationality VARCHAR(15),

/\* Create indexes for faster queries \*/

INDEX (AuthorId, AuthorFirstName, AuthorLastName, AuthorNationality)

);

/\* Create Book Table, with BookID as Primary Key, Book Author (AuthorID) as Foreign Key from Author \*/

CREATE TABLE IF NOT EXISTS Book

(

BookId INT PRIMARY KEY,

BookTitle VARCHAR(40),

BookAuthor INT,

Genre VARCHAR(12),

/\* Foreign key BookAuthor is created by defining the foreign key from reference AuthorID \*/

FOREIGN KEY (BookAuthor)

REFERENCES Author(AuthorId),

/\* Create indexes for faster queries \*/

INDEX (BookID, BookTitle, BookAuthor, Genre)

);

14 Queries to Retrieve Information

# Query 1 Prompt: Display all contents of the Clients table

SELECT \*

FROM Client;

# Query 1 Results:

ClientId,ClientFirstName,ClientLastName,ClientDoB,Occupation

1,Kaiden,Hill,2006,Student

2,Alina,Morton,2010,Student

3,Fania,Brooks,1983,"Food Scientist"

4,Courtney,Jensen,2006,Student

5,Brittany,Hill,1983,Firefighter

6,Max,Rogers,2005,Student

7,Margaret,McCarthy,1981,"School Psychologist"

8,Julie,McCarthy,1973,Professor

9,Ken,McCarthy,1974,"Securities Clerk"

10,Britany,O'Quinn,1984,Violinist

11,Conner,Gardner,1998,"Licensed Massage Therapist"

12,Mya,Austin,1960,"Parquet Floor Layer"

13,Thierry,Rogers,2004,Student

14,Eloise,Rogers,1984,"Computer Security Manager"

15,Gerard,Jackson,1979,"Oil Exploration Engineer"

16,Randy,Day,1986,"Aircraft Electrician"

17,Jodie,Page,1990,"Manufacturing Director"

18,Coral,Rice,1996,"Window Washer"

19,Ayman,Austin,2002,Student

20,Jaxson,Austin,1999,"Repair Worker"

21,Joel,Austin,1973,"Police Officer"

22,Alina,Austin,2010,Student

23,Elin,Austin,1962,"Payroll Clerk"

24,Ophelia,Wolf,2004,Student

25,Eliot,McGuire,1967,Dentist

26,Peter,McKinney,1968,Professor

27,Annabella,Henry,1974,Nurse

28,Anastasia,Baker,2001,Student

29,Tyler,Baker,1984,"Police Officer"

30,Lilian,Ross,1983,"Insurance Agent"

31,Thierry,Arnold,1975,"Bus Driver"

32,Angelina,Rowe,1979,Firefighter

33,Marcia,Rowe,1974,"Health Educator"

34,Martin,Rowe,1976,"Ship Engineer"

35,Adeline,Rowe,2005,Student

36,Colette,Rowe,1963,Professor

37,Diane,Clark,1975,"Payroll Clerk"

38,Caroline,Clark,1960,Dentist

39,Dalton,Clayton,1982,"Police Officer"

40,Steve,Clayton,1990,"Bus Driver"

41,Melanie,Clayton,1987,"Computer Engineer"

42,Alana,Wilson,2007,Student

43,Carson,Byrne,1995,"Food Scientist"

44,Conrad,Byrne,2007,Student

45,Ryan,Porter,2008,Student

46,Elin,Porter,1978,"Computer Programmer"

47,Tyler,Harvey,2007,Student

48,Arya,Harvey,2008,Student

49,Serena,Harvey,1978,"School Teacher"

50,Lilly,Franklin,1976,Doctor

51,Mai,Franklin,1994,Dentist

52,John,Franklin,1999,Firefighter

53,Judy,Franklin,1995,Firefighter

54,Katy,Lloyd,1992,"School Teacher"

55,Tamara,Allen,1963,"Ship Engineer"

56,Maxim,Lyons,1985,"Police Officer"

57,Allan,Lyons,1983,"Computer Engineer"

58,Marc,Harris,1980,"School Teacher"

59,Elin,Young,2009,Student

60,Diana,Young,2008,Student

61,Diane,Young,2006,Student

62,Alana,Bird,2003,Student

63,Anna,Becker,1979,"Security Agent"

64,Katie,Grant,1977,Manager

65,Joan,Grant,2010,Student

66,Bryan,Bell,2001,Student

67,Belle,Miller,1970,Professor

68,Peggy,Stevens,1990,"Bus Driver"

69,Steve,Williamson,1975,"HR Clerk"

70,Tyler,Williamson,1999,Doctor

71,Izabelle,Williamson,1990,"Systems Analyst"

72,Annabel,Williamson,1960,Cashier

73,Mohamed,Waters,1966,"Insurance Agent"

74,Marion,Newman,1970,"Computer Programmer"

75,Ada,Williams,1986,"Computer Programmer"

76,Sean,Scott,1983,"Bus Driver"

77,Farrah,Scott,1974,"Ship Engineer"

78,Christine,Lambert,1973,"School Teacher"

79,Alysha,Lambert,2007,Student

80,Maia,Grant,1984,"School Teacher"

# Query 2 Prompt: First names, last names, ages and occupations of all clients

/\* Because ages were not provided, we have to calculate age by subtracting the year of birth (ClientDoB) from the current year (2023). \*/

SELECT ClientFirstName, ClientLastName, (2023 - ClientDoB) AS Age, Occupation

FROM Client;

#Query 2 Results:

ClientFirstName,ClientLastName,Age,Occupation

Kaiden,Hill,17,Student

Alina,Morton,13,Student

Fania,Brooks,40,"Food Scientist"

Courtney,Jensen,17,Student

Brittany,Hill,40,Firefighter

Max,Rogers,18,Student

Margaret,McCarthy,42,"School Psychologist"

Julie,McCarthy,50,Professor

Ken,McCarthy,49,"Securities Clerk"

Britany,O'Quinn,39,Violinist

Conner,Gardner,25,"Licensed Massage Therapist"

Mya,Austin,63,"Parquet Floor Layer"

Thierry,Rogers,19,Student

Eloise,Rogers,39,"Computer Security Manager"

Gerard,Jackson,44,"Oil Exploration Engineer"

Randy,Day,37,"Aircraft Electrician"

Jodie,Page,33,"Manufacturing Director"

Coral,Rice,27,"Window Washer"

Ayman,Austin,21,Student

Jaxson,Austin,24,"Repair Worker"

Joel,Austin,50,"Police Officer"

Alina,Austin,13,Student

Elin,Austin,61,"Payroll Clerk"

Ophelia,Wolf,19,Student

Eliot,McGuire,56,Dentist

Peter,McKinney,55,Professor

Annabella,Henry,49,Nurse

Anastasia,Baker,22,Student

Tyler,Baker,39,"Police Officer"

Lilian,Ross,40,"Insurance Agent"

Thierry,Arnold,48,"Bus Driver"

Angelina,Rowe,44,Firefighter

Marcia,Rowe,49,"Health Educator"

Martin,Rowe,47,"Ship Engineer"

Adeline,Rowe,18,Student

Colette,Rowe,60,Professor

Diane,Clark,48,"Payroll Clerk"

Caroline,Clark,63,Dentist

Dalton,Clayton,41,"Police Officer"

Steve,Clayton,33,"Bus Driver"

Melanie,Clayton,36,"Computer Engineer"

Alana,Wilson,16,Student

Carson,Byrne,28,"Food Scientist"

Conrad,Byrne,16,Student

Ryan,Porter,15,Student

Elin,Porter,45,"Computer Programmer"

Tyler,Harvey,16,Student

Arya,Harvey,15,Student

Serena,Harvey,45,"School Teacher"

Lilly,Franklin,47,Doctor

Mai,Franklin,29,Dentist

John,Franklin,24,Firefighter

Judy,Franklin,28,Firefighter

Katy,Lloyd,31,"School Teacher"

Tamara,Allen,60,"Ship Engineer"

Maxim,Lyons,38,"Police Officer"

Allan,Lyons,40,"Computer Engineer"

Marc,Harris,43,"School Teacher"

Elin,Young,14,Student

Diana,Young,15,Student

Diane,Young,17,Student

Alana,Bird,20,Student

Anna,Becker,44,"Security Agent"

Katie,Grant,46,Manager

Joan,Grant,13,Student

Bryan,Bell,22,Student

Belle,Miller,53,Professor

Peggy,Stevens,33,"Bus Driver"

Steve,Williamson,48,"HR Clerk"

Tyler,Williamson,24,Doctor

Izabelle,Williamson,33,"Systems Analyst"

Annabel,Williamson,63,Cashier

Mohamed,Waters,57,"Insurance Agent"

Marion,Newman,53,"Computer Programmer"

Ada,Williams,37,"Computer Programmer"

Sean,Scott,40,"Bus Driver"

Farrah,Scott,49,"Ship Engineer"

Christine,Lambert,50,"School Teacher"

Alysha,Lambert,16,Student

Maia,Grant,39,"School Teacher"

# Query 3 Prompt: First and last names of clients that borrowed books in March 2018

SELECT Borrower.BorrowDate, Client.ClientFirstName, Client.ClientLastName

FROM Borrower

INNER JOIN Client

ON Borroewr.ClientId = Client.ClientId

# Only show results of books from March 2018

WHERE Borrower.BorrowDate > “2018-03-01” AND Borrower.BorrowDate < “2018-04-01”;

# Query 3 Results:

BorrowDate,ClientFirstName,ClientLastName

2018-03-18,Maia,Grant

2018-03-18,Marcia,Rowe

2018-03-07,Alysha,Lambert

2018-03-11,Tyler,Baker

2018-03-14,Katy,Lloyd

2018-03-10,Angelina,Rowe

2018-03-02,Gerard,Jackson

2018-03-15,Carson,Byrne

# Query 4 Prompt: First and last names of the top 5 authors clients borrowed in 2017

SELECT COUNT(Borrower.BorrowId) AS CheckedOut, Authoer.AuthorLastName, Author.AuthorFirstName

FROM Borrower

INNER JOIN Book

ON Borrower.BookId = Book.BookId

INNER JOIN Authoer

ON Book.BookAuthor = Author.AuthorId

# Limit the Year to 2017

WHERE YEAR(Borrower.BorrowDate) = ‘2017’

GROUP BY Author.AuthorLastName, AuthorFirstName

ORDER BY CheckedOut DESC

# Only show top 5 results as suggested by prompt

LIMIT 5;

# Query 4 Results:

CheckedOut,AuthorLastName,AuthorFirstName

7,Smith,Sofia

7,Martin,Elena

7,Moore,Logan

6,Brown,Maria

5,Roy,Zoe

# Query 5 Prompt: Nationalities of the least 5 authors that clients borrowed during 2015-2017

SELECT COUNT(Borrower.BorrowId) AS CheckedOut, Author.AuthorNationality

FROM Borrower

INNER JOIN Book

ON Borrower.BookId = Book.BookId

INNER JOIN Author

ON Book.BookAuthor = Author.AuthorId

# Only show the years 2015-2017 as required by prompt

WHERE YEAR(Borrower.BorrowDate) = '2017' OR '2016' OR '2015'

# Group by Nationality

GROUP BY Author.AuthorNationality

ORDER BY CheckedOut

# Show only the bottom 5 as required by prompt

LIMIT 5;

# Query 5: Results

CheckedOut,AuthorNationality

3,Spain

8,China

12,"Great Britain"

21,Brazil

27,France

# Query 6 Prompt: The book that was most borrowed during the years 2015-2017

SELECT COUNT(Borrower.BorrowId) AS CheckedOut, Book.BookTitle

FROM Borrower

INNER JOIN Book

ON Borrower.BookId = Book.BookId

INNER JOIN Author

ON Book.BookAuthor = Author.AuthorId

# Show year 2015-2017 as required by prompt

WHERE YEAR(Borrower.BorrowDate) = '2017' OR '2016' OR '2015'

# Group by BookTitle

GROUP BY Book.BookTitle

ORDER BY CheckedOut DESC

# Show only the first result in descending order

LIMIT 1;

# Query 6 Results:

CheckedOut,BookTitle

18,"Electrical transformers"

# Query 7 Prompt: Top borrowed genres for client born in 1970-1980

SELECT COUNT(Borrower.BorrowId) AS CheckedOut, Book.Genre

FROM Borrower

INNER JOIN Book

ON Borrower.BookId = Book.BookId

INNER JOIN Client

ON Borrower.ClientId = Client.ClientId

# Only show results from 1970-1980 as required by prompt

WHERE Client.ClientDoB BETWEEN '1969' AND '1981'

# Group by Genre

GROUP BY Book.Genre

ORDER BY CheckedOut DESC

Limit 5;

# Query 7: Results

CheckedOut,Genre

26,Science

16,Fiction

15,"Well being"

5,Humor

4,Society

# Query 8 Prompt: Top 5 occupations that borrowed the most in 2016

SELECT COUNT(Borrower.BorrowId) AS CheckedOut, Client.Occupation

FROM Borrower

INNER JOIN Book

ON Borrower.BookId = Book.BookId

INNER JOIN Client

ON Borrower.ClientId = Client.ClientId

# Show year of 2016 as required by prompt

WHERE YEAR(Borrower.BorrowDate) = '2016'

GROUP BY Client.Occupation

ORDER BY CheckedOut DESC

# Show only the top 5 as required by prompt

LIMIT 5;

# Query 8: Results

CheckedOut,Occupation

32,Student

8,"Bus Driver"

6,Dentist

6,"Computer Programmer"

5,"School Teacher"

# Query 9 Prompt: Average number of borrowed books by job title

# Query 10 Prompt: Create a VIEW and display the titles that were borrowed by at least 20% of clients

CREATE VIEW DesiredBooks AS

SELECT Book.BookTitle, COUNT(Client.ClientId) AS CheckOuts

FROM Borrower

JOIN Book

ON Book.BookId = Borrower.BookId

JOIN Client

ON Borrower.ClientId = Client.ClientId

GROUP BY Book.BookTitle

# Creating a subquery to calculate 20% seems to be the most efficient

HAVING CheckOuts >=

(SELECT (COUNT(Client.ClientId) \* 0.2) AS 20pctTitles

FROM Client);

# Query 10 Results:

BookTitle,CheckOuts

"Electrical transformers",18

# Query 11 Prompt: The top month of borrows in 2017

SELECT COUNT(Borrower.BorrowId) AS NumberBorrows, MONTH(Borrower.BorrowDate) AS MonthOf2017

FROM Borrower

# Show the year 2017 as required by the prompt

WHERE YEAR(Borrower.BorrowDate) = '2017'

GROUP BY MonthOf2017

# Query 12 Prompt: Average number of borrows by age

# Query 13: The oldest and the youngest clients of the library

SELECT (2023 - Client.ClientDoB) AS Age, Client.ClientFirstName AS FirstName, Client.ClientLastName AS LastName

FROM Client

# Just like specifying age, 2023- ClientDoB . Min and Max to display prompt

HAVING Age =

(SELECT MAX(2023 - Client.ClientDoB)

FROM Client) OR Age =

(SELECT MIN(2023 - Client.ClientDoB)

FROM Client)

ORDER BY Age;

# Query 13 Results:

Age,FirstName,LastName

13,Alina,Morton

13,Alina,Austin

13,Joan,Grant

63,Mya,Austin

63,Caroline,Clark

63,Annabel,Williamson

# Query 14: First and last names of authors that wrote books in more than one genre

SELECT Author.AuthorFirstName AS FirstName, Author.AuthorLastName AS LastName, COUNT(subtable.Genre) AS GenresWritten

FROM

(SELECT Author.AuthorId, Book.Genre

FROM Author

INNER JOIN Book

ON Author.AuthorId = Book.BookAuthor

GROUP BY AuthorId) subtable

INNER JOIN Author

ON subtable.authorId = Author.AuthorId

GROUP BY LastName, FirstName

HAVING GenresWritten > 1;

/\* After running this query, no results are shown. After checking the data in the sample tables, I confirmed no author has written books in more than one genre. The result of the query is blank. \*/