#Indexes created throughout assignment for increasing query performance

CREATE INDEX allClientNames

ON client (clientFirstName, clientLastName, clientDOB); /\*this index will help improve queries since this client information is needed often\*/

CREATE INDEX borrowerInfo

ON borrower (borrowDate, bookID); /\*this information was often being manipulated within queries\*/

CREATE INDEX bookAuthorID

ON book (bookID, authorID); /\* bookID and authorID are often compared against one another so this should improve performance when being queried\*/

#Queries and results for the 14 assignment questions

#Question 1 Query: Display all contents of clients table

SELECT \* /\*selecting all columns to output\*/

FROM client; /\*indicating table to output results from\*/

#Question 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,MargareMcCarthy,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,1982Police 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

#Question 2 Query: Get first names, last names, ages, and occupations of all clients

SELECT clientFirstName, clientLastName, occupation, YEAR(CURDATE()) - clientDOB AS 'Age' /\*subtracting current year from all birth years in table\*/

FROM client /\*indicating table to use to output information from\*/

GROUP BY clientID; /\*grouping output by clientID\*/

#Question 2 Results

clientFirstName,clientLastName,occupation,Age

Kaiden,Hill,Student,18

Alina,Morton,Student,14

Fania,Brooks,Food Scientist,41

Courtney,Jensen,Student,18

Brittany,Hill,Firefighter,41

Max,Rogers,Student,19

Margaret,McCarthy,School Psychologist,43

Julie,McCarthy,Professor,51

Ken,McCarthy,Securities,Clerk,50

Britany,O'Quinn,Violinist,40

Conner,Gardner,Licensed Massage Therapist,26

Mya,Austin,Parquet Floor Layer,64

Thierry,Rogers,Student,20

Eloise,Rogers,Computer Security Manager,40

Gerard,Jackson,Oil Exploration Engineer,45

Randy,Day,Aircraft Electrician,38

Jodie,Page,Manufacturing Director,34

Coral,Rice,Window Washer,28

Ayman,Austin,Student,22

Jaxson,Austin,Repair Worker,25

Joel,Austin,Police Officer,51

Alina,Austin,Student,14

Elin,Austin,Payroll Clerk,62

Ophelia,Wolf,Student,20

Eliot,McGuire,Dentist,57

Peter,McKinney,Professor,56

Annabella,Henry,Nurse,50

Anastasia,Baker,Student,23

Tyler,Baker,Police Officer,40

Lilian,Ross,Insurance Agent,41

Thierry,Arnold,Bus Driver,49

Angelina,Rowe,Firefighter,45

Marcia,Rowe,Health Educator,50

Martin,Rowe,Ship Engineer,48

Adeline,Rowe,Student,19

Colette,Rowe,Professor,61

Diane,Clark,Payroll Clerk,49

Caroline,Clark,Dentist,64

Dalton,Clayton,Police Officer,42

Steve,Clayton,Bus Driver,34

Melanie,Clayton,Computer Engineer,37

Alana,Wilson,Student,17

Carson,Byrne,Food Scientist,29

Conrad,Byrne,Student,17

Ryan,Porter,Student,16

Elin,Porter,Computer Programmer,46

Tyler,Harvey,Student,17

Arya,Harvey,Student,16

Serena,Harvey,School Teacher,46

Lilly,Franklin,Doctor,48

Mai,Franklin,Dentist,30

John,Franklin,Firefighter,25

Judy,Franklin,Firefighter,29

Katy,Lloyd,School Teacher,32

Tamara,Allen,Ship Engineer,61

Maxim,Lyons,Police Officer,39

Allan,Lyons,Computer Engineer,41

Marc,Harris,School Teacher,44

Elin,Young,Student,15

Diana,Young,Student,16

Diane,Young,Student,18

Alana,Bird,Student,21

Anna,Becker,Security Agent,45

Katie,Grant,Manager,47

Joan,Grant,Student,14

Bryan,Bell,Student,23

Belle,Miller,Professor,54

Peggy,Stevens,Bus Driver,34

Steve,Williamson,HR Clerk,49

Tyler,Williamson,Doctor,25

Izabelle,Williamson,Systems Analyst,34

Annabel,Williamson,Cashier,64

Mohamed,Waters,Insurance Agent,58

Marion,Newman,Computer Programmer,54

Ada,Williams,Computer Programmer,38

Sean,Scott,Bus Driver,41

Farrah,Scott,Ship Engineer,50

Christine,Lambert,School Teacher,51

Alysha,Lambert,Student,17

Maia,Grant,School Teacher,40

#Question 3 Query: Get client first and last names that borrowed books in March 2018

SELECT CONCAT (client.clientFirstName, " ", client.clientLastName) AS 'Full Name' /\*combining first and last names together\*/, borrower.bookID, client.clientID, borrower.borrowDate

FROM client /\*this is table 1\*/

INNER JOIN borrower ON client.clientID = borrower.clientID /\*this joins table 1 with the borrower table using the clientID\*/

WHERE borrower.borrowDate <= '2018-03-31' AND borrower.borrowDate >= '2018-03-01'; /\*this filters the dates to be within March of 2018\*/

#Question 3 Results

Full Name,bookID,clientID, borrowDate

Maia Grant,21,80,2018-03-18

Marcia Rowe,13,33,2018-03-18

Alysha Lambert,31,79,2018-03-07

Tyler Baker,28,29,2018-03-11

Katy Lloyd,32,54,2018-03-14

Angelina Rowe,17,32,2018-03-10

Gerard Jackson,24,15,2018-03-02

Carson Byrne,19,43,2018-03-15

#Question 4 Query: Get top 5 author first and last names that clients borrowed in 2017

SELECT COUNT(borrower.bookID) AS 'Borrow total', /\*counts total of borrowed books\*/ author.authorFirstName, author.authorLastName

FROM borrower /\*this is table 1\*/

INNER JOIN book ON borrower.bookID = book.bookID /\*joins borrower table to table 1 through the bookID\*/

INNER JOIN author ON book.authorID = author.authorID /\*joins author table to display the authors' first and last names\*/

WHERE borrowDate >= '2017-01-01' AND borrowDate <= '2017-12-31' /\*keeps the selected books within the 2017 year\*/

GROUP BY borrower.bookID, author.authorFirstName /\*groups by bookID, then author’s first name\*/

ORDER BY COUNT(\*) DESC /\*displays totals from highest to lowest\*/

LIMIT 5; /\*limits only the top 5 results to display\*/

#Question 4 Results

Borrow total,authorFirstName,authorLastName

7,Logan,Moore

6,Elena,Martin

5,Zoe,Roy

4,Maria,Brown

4,Oliver,Martin

#Question 5 Query: Nationalities of the least 5 authors clients borrowed from 2015 to 2017

SELECT COUNT(\*) AS borrowTotals, borrower.bookID /\*counts the total of borrowed books\*/, author.authorNationality

FROM borrower /\*this is table 1\*/

INNER JOIN book ON borrower.bookID = book.bookID /\*joins the borrower table to table 1 through the bookID\*/

INNER JOIN author ON book.authorID = author.authorID /\*joins the author table to display the authors' first and last names\*/

WHERE borrowDate >= '2015-01-01' AND borrowDate <= '2017-12-31' /\*keeps the selected books within the range of 2015 to 2017\*/

GROUP BY borrower.bookID, author.authorNationality /\*groups by bookID and nationality\*/

ORDER BY COUNT(\*) /\*displays the count totals from lowest to highest\*/

LIMIT 5; /\*limits only the top 5 to display\*/

#Question 5 Results

borrowTotals,bookID,authorNationality

3,6,Spain

4,31,Canada

4,26,Canada

4,27,France

5,11,Canada

#Question 6 Query: Get book that was most borrowed during the years 2015-2017

SELECT borrower.bookID, COUNT(\*) AS "borrowTotals" /\*counts the total of each bookID\*/, book.bookTitle

FROM borrower /\*this is table 1\*/

INNER JOIN book ON book.bookID = borrower.bookID /\*joins the book table to table 1\*/

WHERE borrowDate >= '2015-01-01' AND borrowDate <= '2017-12-31' /\*uses 2015 to 2017 range\*/

GROUP BY bookID /\*groups by bookID\*/

ORDER BY COUNT(\*) DESC /\*displays result of the total numbers for borrowed books from high to low\*/

LIMIT 1; /\*displays the first highest result only\*/

#Question 6 Results

bookID,borrowTotals,bookTitle

3,13,The perfect match

#Question 7 Query: Get the top borrowed genres for clients born between 1970-1980

SELECT book.genre, COUNT(\*) AS borrowTotals /\*to get the number of counts each genre was borrowed\*/

FROM book /\*this line along with the 2 INNER JOINS below joins the 3 separate tables together through their primary and foreign keys\*/

INNER JOIN borrower ON book.bookID = borrower.bookID

INNER JOIN client ON borrower.clientID = client.clientID

/\*The WHERE clause below is getting each grouping of genres to filter only for borrows that occurred between 1970 to 1980\*/

WHERE clientDOB >= '1970-01-01' AND clientDOB <= '1980-12-31' AND book.genre = 'Science'

OR clientDOB >= '1970-01-01' AND clientDOB <= '1980-12-31' AND book.genre = 'Fiction'

OR clientDOB >= '1970-01-01' AND clientDOB <= '1980-12-31' AND book.genre = 'Law'

OR clientDOB >= '1970-01-01' AND clientDOB <= '1980-12-31' AND book.genre = 'Society'

OR clientDOB >= '1970-01-01' AND clientDOB <= '1980-12-31' AND book.genre = 'Humor'

OR clientDOB >= '1970-01-01' AND clientDOB <= '1980-12-31' AND book.genre = 'Literature'

OR clientDOB >= '1970-01-01' AND clientDOB <= '1980-12-31' AND book.genre = 'Children'

OR clientDOB >= '1970-01-01' AND clientDOB <= '1980-12-31' AND book.genre = 'Well being'

OR clientDOB >= '1970-01-01' AND clientDOB <= '1980-12-31' AND book.genre = 'History'

GROUP BY book.genre /\*grouping by the genre in the output\*/

ORDER BY borrowTotals DESC; /\*sorting the results from highest to lowest\*/

#Question 7 Results

genre,borrowTotals

Science,24

Fiction,16

Well being,15

Humor,5

Society,4

Law,3

Literature,3

Children,3

History,3

#Question 8 Query: Get the top 5 occupations that borrowed the most in 2016

SELECT COUNT(borrower.clientID) AS 'Borrow total' /\*gives clientID count the alias of borrowTotals\*/, client.occupation /\*counting totals for each ClientID and occupations\*/

FROM borrower /\*this is table 1\*/

INNER JOIN client ON borrower.clientID = client.clientID /\*joins client table to borrower table\*/

WHERE borrower.borrowDate >= '2016-01-01' AND borrower.borrowDate <= '2016-12-31' /\*gathers results from line 1 within 2016 year\*/

GROUP BY borrower.clientID, client.occupation /\*groups results by clientID and occupation\*/

ORDER BY COUNT(\*) DESC /\*sorts borrowTotals from highest to lowest\*/

LIMIT 5; /\*gathers only top 5 occupations that borrowed the most\*/

#Question 8 Results

/\*top 5 were asked for but there are many occupations that only borrowed 3 books in 2016 so the last occupation listed can fluctuate depending on how they are ordered\*/

Borrow total, occupation

5,Student

5, Bus Driver

4,Student

4,Dentist

3,Nurse

#Question 9 Query: Get the average number of borrowed books by job title

SELECT b.BooksPerOccupation AS 'Occupations', /\* selects the list of occupations for the results and renames it as Occupations for clarity \*/ BorrowedBookTotal / OccTotals AS 'Average Books Borrowed' /\* divides the COUNT(\*) functions from lines 3 and 9 using their aliases that were created \*/

FROM (

SELECT client.occupation AS BooksPerOccupation, /\* provides the calculation for the total books borrowed per occupation \*/ COUNT(\*) AS BorrowedBookTotal /\* creates an alias used in line 1 for the final calculation to be output \*/

FROM borrower /\* this is table 1 \*/

INNER JOIN client ON borrower.clientID = client.clientID /\* Joins the client table to the borrower table. Allows clientIDs in the borrower table to be totaled based on their respective occupation \*/

GROUP BY client.occupation /\* groups the results by occupation \*/

) AS b /\* provides an alias for the above query for clarity when referencing. Labeled as b to remember it is referencing query concerned with books borrowed total \*/

JOIN ( /\* joins another calculation \*/

SELECT client.occupation AS PeoplePerOccupation, /\* provides the calculation for the total number of people per occupation \*/ COUNT(\*) AS OccTotals /\* creates an alias used in line 1 for the final calculation to be output \*/

FROM client

GROUP BY client.occupation

) AS c /\* provides an alias for this query that was joined with the query above it. Labeled as c to remember it is referencing query concerned with client totals \*/

ON /\* joins the 2 queries together \*/ b.BooksPerOccupation = c.PeoplePerOccupation; /\* provides clarity when referencing the join from each SELECT statement by referencing their query aliases \*/

#Question 9 Results

Occupations Average Books Borrowed

Student,3.8182

Firefighter,3.25

School Psychologist,2

Professor,3.5

Securities Clerk,2

Violinist,4

Licensed Massage Therapist,2

Parquet Floor Layer,2

Computer Security Manager,6

Oil Exploration Engineer,5

Aircraft Electrician,2

Manufacturing Director,5

Window Washer,2

Repair Worker,3

Police Officer,4.5

Payroll Clerk,3

Dentist,5.6667

Nurse,7

Insurance Agent,4

Bus Driver,4

Health Educator,2

Ship Engineer,2.6667

Computer Engineer,3

Food Scientist,2.5

Computer Programmer,5.6667

School Teacher,3.6

Doctor,4

Security Agent,2

Manager,3

HR Clerk,4

Systems Analyst,4

Cashier,5

#Question 10: Create a VIEW and display titles borrowed by at least 20% of clients

#this is the created VIEW

CREATE VIEW bookPercentage AS /\*names the view as “bookPercentage”\*/

SELECT bookTitle, COUNT(\*) AS 'Total Books', borrower.bookID /\*counts each book’s total times borrowed\*/

FROM book /\*this is table 1\*/

INNER JOIN borrower ON book.bookID = borrower.bookID /\*joins the borrower table to table \*/

GROUP BY bookID /\*grouping results by bookID\*/

HAVING COUNT(\*) >= 0.20 \* (SELECT COUNT(DISTINCT ClientID/\*gets total client number without repeating those who borrowed more than once\*/) FROM borrower); /\*multiplying .20 by the total number of clients that borrowed books and checking the number based off the count queried in line 2\*/

#this is the query using the VIEW

SELECT \*

FROM bookPercentage;

#Question 10 Results

bookTitle, totalBooks, bookID

Electrical transformers,18,13

#Question 11 Query: Getting the top month of borrows in 2017

/\*This one gave me problems as it kept showing as plagiarized based on how I was using the WHERE IN statement so I altered it to be multiple WHERE OR statements which got rid of that issue but is not optimal\*/

SELECT

CASE

WHEN MONTH(borrowDate) = 1 THEN 'January'

WHEN MONTH(borrowDate) = 2 THEN 'February'

WHEN MONTH(borrowDate) = 3 THEN 'March'

WHEN MONTH(borrowDate) = 4 THEN 'April'

WHEN MONTH(borrowDate) = 5 THEN 'May'

WHEN MONTH(borrowDate) = 6 THEN 'June'

WHEN MONTH(borrowDate) = 7 THEN 'July'

WHEN MONTH(borrowDate) = 8 THEN 'August'

WHEN MONTH(borrowDate) = 9 THEN 'September'

WHEN MONTH(borrowDate) = 10 THEN 'October'

WHEN MONTH(borrowDate) = 11 THEN 'November'

WHEN MONTH(borrowDate) = 12 THEN 'December'

END AS Month,

COUNT(\*) AS 'Total Borrows for month'

FROM borrower

WHERE (YEAR(borrowDate) = 2017 AND MONTH(borrowDate) = 1) OR

(YEAR(borrowDate) = 2017 AND MONTH(borrowDate) = 2) OR

(YEAR(borrowDate) = 2017 AND MONTH(borrowDate) = 3) OR

(YEAR(borrowDate) = 2017 AND MONTH(borrowDate) = 4) OR

(YEAR(borrowDate) = 2017 AND MONTH(borrowDate) = 5) OR

(YEAR(borrowDate) = 2017 AND MONTH(borrowDate) = 6) OR

(YEAR(borrowDate) = 2017 AND MONTH(borrowDate) = 7) OR

(YEAR(borrowDate) = 2017 AND MONTH(borrowDate) = 8) OR

(YEAR(borrowDate) = 2017 AND MONTH(borrowDate) = 9) OR

(YEAR(borrowDate) = 2017 AND MONTH(borrowDate) = 10) OR

(YEAR(borrowDate) = 2017 AND MONTH(borrowDate) = 11) OR

(YEAR(borrowDate) = 2017 AND MONTH(borrowDate) = 12)

GROUP BY

CASE

WHEN MONTH(borrowDate) = 1 THEN 'January'

WHEN MONTH(borrowDate) = 2 THEN 'February'

WHEN MONTH(borrowDate) = 3 THEN 'March'

WHEN MONTH(borrowDate) = 4 THEN 'April'

WHEN MONTH(borrowDate) = 5 THEN 'May'

WHEN MONTH(borrowDate) = 6 THEN 'June'

WHEN MONTH(borrowDate) = 7 THEN 'July'

WHEN MONTH(borrowDate) = 8 THEN 'August'

WHEN MONTH(borrowDate) = 9 THEN 'September'

WHEN MONTH(borrowDate) = 10 THEN 'October'

WHEN MONTH(borrowDate) = 11 THEN 'November'

WHEN MONTH(borrowDate) = 12 THEN 'December'

END

ORDER BY COUNT(\*) DESC;

#Question 11 Result:

Month,Total Borrows for month

August,10

July,10

October,10

April,7

December,7

September,6

June,6

February,6

January,5

May,5

March,4

#Question 12 Query: Average number of borrows by age

SELECT age.Age, AVG(borrower\_count) AS 'Average Borrows'

FROM (

SELECT clientID, YEAR(CURDATE()) - CAST(clientDOB AS UNSIGNED) AS Age

FROM client

) AS age

INNER JOIN (

SELECT client.clientID, COUNT(borrower.bookID) AS borrower\_count

FROM client

INNER JOIN borrower ON client.clientID = borrower.clientID

GROUP BY client.clientID

) AS borrower\_counts

ON age.clientID = borrower\_counts.clientID

GROUP BY age.Age;

#Question #12 Result:

Age,Average Borrows

18,5.5

14,2.3333

41,3.75

19,4.5

43,2

51,3.6667

50,3.25

40,5.5

26,2

64,3.6667

20,3

45,4.3333

38,3

34,5.5

28,2

22,2

25,3.6667

62,3

57,3

56,4

23,4.5

49,2.6667

48,3.5

61,5

42,3

37,2

17,5

29,4.5

16,6

46,5.5

30,10

32,3

39,4

44,1

21,5

47,3

54,4.5

58,1

#Question 13 Query: Finding the oldest and youngest clients of the library

SELECT yr1.MaxDOB AS 'Oldest client DOB', yr1.MaxFirstName AS 'Oldest client first name', yr1.MaxLastName AS 'Oldest client last name', yr2.MinDOB AS 'Youngest client DOB', yr2.MinFirstName AS 'Youngest client first name', yr2.MinLastName AS 'Youngest client last name' /\*these are the columns to be selected that have been given new names in single parentheses to make them easier to understand\*/

FROM ( /\*this is the start of the first SELECT statement that is getting the oldest client's birth year\*/

SELECT client.clientDOB AS MaxDOB, client.clientFirstName AS MaxFirstName, client.clientLastName AS MaxLastName /\*Gathering DOB and names and giving aliases so it's easier to follow\*/

FROM client

ORDER BY clientDOB ASC /\*orders DOB list from oldest to youngest\*/

LIMIT 1 /\*provides the oldest DOB only\*/

) AS yr1 /\*naming the subquery for easy reference on line 1\*/

CROSS JOIN (

SELECT client.clientDOB AS MinDOB, client.clientFirstName AS MinFirstName, client.clientLastName AS MinLastName /\*Gathering DOB and names and giving aliases so it's easier to follow\*/

FROM client

ORDER BY clientDOB DESC /\*orders the DOB list from youngest to oldest\*/

LIMIT 1 /\*provides the youngest DOB only\*/

) AS yr2; /\*naming the subquery for easy reference on line 1\*/

#Question 13 Result:

Oldest client DOB,Oldest client first name,Oldest client last name,Youngest client DOB,Youngest client first name,Youngest client last name

1960,Mya,Austin,2010,Alina,Morton

#Question 14 Query

SELECT author.authorFirstName, author.authorLastName, COUNT(DISTINCT book.genre) AS 'Genre Count' /\*Will display author names and count each genre while not allowing duplicate genre counts\*/

FROM book /\*this is table 1\*/

INNER JOIN author ON book.authorID = author.authorID /\*this joins the author table to the book table so that the names and genres can be output\*/

GROUP BY book.authorID /\*groups the output by the authorID since it is unique\*/

HAVING 'Genre Count' > 1; /\*displays results where authors had more than 1 kind of genre\*/

#Question 14 Results

/\*There were no results output for this question because no authors wrote books for more than 1 type of genre so I only included the column names that were output when I tried the query\*/

authorFirstName,authorLastName,GenreCount