Ryan Russon

***Information Retrieval Queries***

1. Display all contents of the Clients table

/\*Display all the date from the client table\*/

SELECT \*

FROM client;

1. First names, last names, ages and occupations of all clients

/\*The standard columns will be referenced and an additional column called age will be found by taking the year from the current date and subtracting the year of birth from the clients\*/

SELECT ClientFirstName, ClientLastName, YEAR(CURDATE()) - ClientDOB AS age, Occupation

FROM client;

1. First and last names of clients that borrowed books in March 2018

/\*This is going to find the client names and reference them to the borrower table where clientid’s are found between the March 2018 dates\*/

SELECT ClientFirstName, ClientLastName

FROM client

WHERE client.ClientID IN (SELECT borrower.ClientID

FROM borrower

WHERE BorrowDate BETWEEN '2018-03-01' AND '2018-03-31');

1. First and last names of the top 5 authors clients borrowed in 2017

/\*For this query it will need to link data from a couple tables together. Mainly the author for their names, the book to link the borrower table to the author table, and the borrower table to find the count of the most borrowed books. The authorid’s will get grouped together, so that it can get an appropriate count. And finally count from order from highest to lowest and only show the top five. \*/

SELECT AuthorFirstName, authorlastname

FROM author

INNER JOIN book ON author.authorid = book.bookauthor

INNER JOIN borrower ON book.bookid = borrower.bookid

WHERE YEAR(borrower.borrowdate) = 2017

GROUP BY author.authorid

ORDER BY COUNT(\*) DESC

LIMIT 5;

1. Nationalities of the least 5 authors that clients borrowed during the years 2015-2017

/\*Again join the author table for their nationalities, the book table to link author and borrower and the borrower table to find the least borrowed between 2015 and 2017. Group the authors, count them in least to greatest order and then show top five.\*/

SELECT authornationality

FROM author

INNER JOIN book ON author.authorid = book.bookauthor

INNER JOIN borrower ON book.bookid = borrower.BookID

WHERE borrower.BorrowDate BETWEEN '2015-01-01' AND '2017-012-31'

GROUP BY author.authorid

ORDER BY COUNT(\*) ASC

LIMIT 5;

1. The book that was most borrowed during the years 2015-2017

/\*Join the book and borrow class then filter on 2015-2017. Then group by book and order by descending count and pull only the top row.\*/

SELECT booktitle

FROM book

INNER JOIN borrower ON book.bookid = borrower.BorrowerID

WHERE borrower.BorrowDate BETWEEN '2015-01-01' AND '2017-12-31'

GROUP BY borrower.bookid

ORDER BY COUNT(borrower.bookid) DESC

LIMIT 1;

1. Top borrowed genres for client born in years 1970-1980

/\*Join the borrower, client and book genres. Borrower to find a count, client to filter by the years, and book to find the genres. Then group by the bookID, order by the descending and return the genres.\*/

SELECT genre

FROM book

INNER JOIN borrower ON book.bookid = borrower.BookID

INNER JOIN client ON borrower.clientID = client.ClientID

WHERE client.ClientDOB BETWEEN 1970 AND 1980

GROUP BY book.genre

ORDER BY COUNT(\*) DESC

1. Top 5 occupations that borrowed the most in 2016

/\*Start with the client table and join the borrower table to find a count and to filter from the year 2016. Then group by the occupation and order by the occupation.\*/

SELECT occupation

FROM client

INNER JOIN borrower ON client.clientID = borrower.clientID

WHERE borrower.borrowdate BETWEEN '2016-01-01' AND '2016-12-31'

GROUP BY client.occupation

ORDER BY (client.occupation) DESC

LIMIT 5;

1. Average number of borrowed books by job title

/\*Use an inner select in order to create a count for each client, then the outer query will group by occupation and find an average amount borrowed by each occupation\*/

SELECT client.occupation, AVG(num\_borrowed) AS Average

FROM (SELECT clientid, COUNT(clientid) AS num\_borrowed

FROM borrower

GROUP BY clientid) borrower

INNER JOIN client ON borrower.clientid = client.clientid

GROUP BY Occupation

1. Create a VIEW and display the titles that were borrowed by at least 20% of clients

/\*Couldn’t function through the “at least 20%” \*/

CREATE VIEW pop\_titles AS

SELECT booktitle, COUNT(borrower.clientid) AS num\_clients

FROM book

INNER JOIN borrower ON book.bookid = borrower.BookID

GROUP BY book.booktitle

ORDER BY COUNT(borrower.clientid) DESC;

1. The top month of borrows in 2017

/\*In order to get this query group by the month of the borrowdate and then order by the counts for each month.\*/

SELECT borrowdate, COUNT(BorrowerID) as borrows\_per\_month

FROM borrower

GROUP BY month(borrowdate)

ORDER BY borrows\_per\_month DESC

LIMIT 1;

1. Average number of borrows by age

/\*Couldn’t quite get the function to solve \*/

SELECT YEAR(CURDATE()) - ClientDOB AS Age, AVG(borrows\_per\_client) AS AverageBorrows

From client,

(SELECT borrower.clientid, COUNT(borrower.clientid) AS borrows\_per\_client

FROM borrower

INNER JOIN client ON borrower.clientid = client.clientid

GROUP BY borrower.clientid) AS Sub;

1. The oldest and the youngest clients of the library

/\*In order to do this query use a union to combine the two separate select queries\*/

(SELECT clientid, YEAR(curdate()) - clientdob AS Age

FROM client

GROUP BY clientid

ORDER BY Age DESC

LIMIT 1)

UNION

(SELECT clientid, YEAR(curdate()) - clientdob AS Age

FROM client

GROUP BY clientid

ORDER BY Age ASC

LIMIT 1);

1. First and last names of authors that wrote books in more than one genre

/\*In this query use a normal join in order to allow duplicates and then use the HAVING clause to make sure each author has more than 1 genre.\*/

SELECT author.AuthorFirstName, author.AuthorLastName

FROM author

JOIN (

SELECT bookauthor

FROM book

GROUP BY bookauthor

HAVING COUNT(genre) > 1

)

AS duplicate\_table ON author.AuthorID = duplicate\_table.bookauthor;