

**For this assignment I have used **red italic font** for comments.**

**I created large **red bold indentation** within code to break up the queries.**

**The results of the queries will be above this indentation in the code.**

**I have labeled every question in the following format:**

**Objective C Question (ENTER QUESTION NUMBER HERE)**

**This is done in **bold green font** so you can search or find it in the code. I also added comments in the bottom in regards to indexing in the same font as the questions.**

**I have screenshots of queries and errors in code as well.**

-- *I had originally planned to auto-increment the ID values*  
-- *When I started inputting data however, I noticed the IDs were not starting 1*  
-- *I had inputted all entries for the Clients and borrowers*  
-- *I created references and built relationships so dropping the table would cause major errors*  
-- *Instead of doing that I have decided to create a new table without auto-increment*  
-- *I will attach screenshots as well*

CREATE TABLE Authors ( -- This will create the Authors table without auto-increment  
    AuthorID INT PRIMARY KEY,  
    AuthorFirstName VARCHAR(50),  
    AuthorLastName VARCHAR(50),  
    AuthorNationality VARCHAR(50));

-- *I manually entered the AuthorID which will be the primary key for this table*  
INSERT INTO Authors (AuthorID, AuthorFirstName, AuthorLastName, AuthorNationality)  
VALUES

```
(1, 'Sofia', 'Smith', 'Canada'),
(2, 'Maria', 'Brown', 'Brazil'),
(3, 'Elena', 'Martin', 'Mexico'),
(4, 'Zoe', 'Roy', 'France'),
(5, 'Sebastian', 'Lavoie', 'Canada'),
(6, 'Dylan', 'Garcia', 'Spain'),
(7, 'Ian', 'Cruz', 'Mexico'),
(8, 'Lucas', 'Smith', 'USA'),
(9, 'Fabian', 'Wilson', 'USA'),
(10, 'Liam', 'Taylor', 'Canada'),
(11, 'William', 'Thomas', 'Great Britain'),
(12, 'Logan', 'Moore', 'Canada'),
(13, 'Oliver', 'Martin', 'France'),
(14, 'Alysha', 'Thompson', 'Canada'),
(15, 'Isabelle', 'Lee', 'Canada'),
(16, 'Emily', 'Clark', 'USA'),
(17, 'John', 'Young', 'China'),
(18, 'David', 'Wright', 'Canada'),
(19, 'Thomas', 'Scott', 'Canada'),
(20, 'Helena', 'Adams', 'Canada'),
(21, 'Sofia', 'Carter', 'USA'),
(22, 'Liam', 'Parker', 'Canada'),
(23, 'Emily', 'Murphy', 'USA');
```

CREATE TABLE Books ( -- Since I ran into the BookID field before, I will not auto-increment  
BookID INT PRIMARY KEY,

BookTitle VARCHAR(255),

-- *I don't know how long the titles are so i just put 255 to not have to count it*

BookAuthor INT,

Genre VARCHAR(100),

-- *I didn't max this out since looking at the database none of the genres will pass this value*  
FOREIGN KEY (BookAuthor) REFERENCES Authors(AuthorID));

-- *The Foreign key is put to make sure that every Book Author has an Author ID*

-- *I am updating the Books database with the following INSERT INTO command*

-- *The BookID will NOT be auto-incremented as the primary key as I had issues mentioned above*

```
INSERT INTO Books (BookID,BookTitle, BookAuthor, Genre) VALUES
```

```
(1, 'Build your database system', 1, 'Science'),
(2, 'The red wall', 2, 'Fiction'),
(3, 'The perfect match', 3, 'Fiction'),
(4, 'Digital Logic', 4, 'Science'),
(5, 'How to be a great lawyer', 5, 'Law'),
(6, 'Manage successful negotiations', 6, 'Society'),
(7, 'Pollution today', 7, 'Science'),
(8, 'A gray park', 2, 'Fiction'),
(9, 'How to be rich in one year', 8, 'Humor'),
(10, 'Their bright fate', 9, 'Fiction'),
(11, 'Black lines', 10, 'Fiction'),
(12, 'History of theater', 11, 'Literature'),
(13, 'Electrical transformers', 12, 'Science'),
(14, 'Build your big data system', 1, 'Science'),
```

(15, 'Right and left', 13, 'Children'),  
(16, 'Programming using Python', 1, 'Science'),  
(17, 'Computer networks', 14, 'Science'),  
(18, 'Performance evaluation', 15, 'Science'),  
(19, 'Daily exercise', 16, 'Well being'),  
(20, 'The silver uniform', 17, 'Fiction'),  
(21, 'Industrial revolution', 18, 'History'),  
(22, 'Green nature', 19, 'Well being'),  
(23, 'Perfect football', 20, 'Well being'),  
(24, 'The chocolate love', 21, 'Humor'),  
(25, 'Director and leader', 22, 'Society'),  
(26, 'Play football every week', 20, 'Well being'),  
(27, 'Maya the bee', 13, 'Children'),  
(28, 'Perfect rugby', 20, 'Well being'),  
(29, 'The end', 23, 'Fiction'),  
(30, 'Computer security', 1, 'Science'),  
(31, 'Participate', 22, 'Society'),  
(32, 'Positive figures', 3, 'Society'); -- I had put the ID as 24 and changed it to 3

-- Double checking the table again like I did with the Authors table

-- The following SELECT command will be executed and a screenshot will be taken

-- SELECT \* FROM Books gave me the following ERROR CODE

-- The error code I got Error Code: 1452. Cannot add or update a child row:

-- a foreign key constraint fails ('cs 204 project library`.`books`, CONSTRAINT `books\_ibfk\_1` FOREIGN KEY (BookAuthor) REFERENCES `authors` (AuthorID))

-- SELECT AuthorID, AuthorFirstName, AuthorLastName FROM Authors; showed that the Authors were correct

-- I then looked at the books table and noticed that the last entry 'Positive figures' was incorrect

-- It had an ID of 24 when it should be 3

-- I will use the update command below to fix this error

-- UPDATE Books

-- SET BookAuthor = 3

-- WHERE BookTitle = 'Positive Figures';

-- This actually did not correct the issue since the table would not update due to the AuthorID error

-- It was showing a null table so I simply just corrected the entry in the table above.

SELECT \* FROM Books; -- I am just running this to see if the BookID is correct :)

-- I AM CREATING THIS SPACE TAB MAKING IT  
EASIER FOR ME TO COPY THE CODE

-- I CAN EASILY SEE WHERE I HAVE COPIED THE CODE  
TO EDIT AND THEN INSERT INTO TXT

-- THIS IS JUST HERE FOR ME TO EASILY COPY AND  
PASTE THE CODE

-- *Creating the Client Table*

-- *For continuity, I will not use auto-increment for any of the tables*

```
CREATE TABLE Clients (
    ClientID INT PRIMARY KEY,
    ClientFirstName VARCHAR(100),
    ClientLastName VARCHAR(100),
    ClientDoB YEAR,
    -- Only the year is listed in this section so i did not put DATE
    Occupation VARCHAR(100));
```

```
INSERT INTO Clients (ClientID, ClientFirstName, ClientLastName, ClientDoB, Occupation)
VALUES
(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');

-- SELECT \* FROM CLIENTS; Command to confirm proper entries

-- This is also where Prompt C. 1. Display all Clients is and I took a screenshot of it

**-- I AM CREATING THIS SPACE TAB MAKING IT  
EASIER FOR ME TO COPY THE CODE**

**-- I CAN EASILY SEE WHERE I HAVE COPIED THE CODE  
TO EDIT AND THEN INSERT INTO TXT**

**-- THIS IS JUST HERE FOR ME TO EASILY COPY AND  
PASTE THE CODE**

```
CREATE TABLE Borrowers (
    BorrowID INT PRIMARY KEY,
    ClientID INT,
    BookID INT,
    BorrowDate DATE,
    -- Date is used vs year since the full date is listed in the table
    FOREIGN KEY (ClientID) REFERENCES Clients(ClientID),
    FOREIGN KEY (BookID) REFERENCES Books(BookID));
    -- The foreign keys are used to build relationships between the two tables involved in each statement
    -- The ClientID are connected since they are both present in the Borrowers and Clients table
    -- The BookID are connected since they are both present in the Borrowers and Books table
```

-- I will now enter the 300 values for the Borrowers table

```
INSERT INTO Borrowers (BorrowID, ClientID, BookID, BorrowDate) VALUES
(1, 35, 17, '2016-07-20'),
(2, 1, 3, '2017-04-19'),
(3, 42, 8, '2016-10-03'),
(4, 62, 16, '2016-04-05'),
(5, 53, 13, '2017-01-17'),
(6, 33, 15, '2015-11-26'),
(7, 40, 14, '2015-01-21'),
(8, 64, 2, '2017-09-10'),
(9, 56, 30, '2017-08-02'),
(10, 23, 2, '2018-06-28'),
(11, 46, 19, '2015-11-18'),
(12, 61, 20, '2015-11-24'),
(13, 58, 7, '2017-06-17'),
(14, 46, 16, '2017-02-12'),
(15, 80, 21, '2018-03-18'),
(16, 51, 23, '2015-09-01'),
```

(17, 49, 18, '2015-07-28'),  
(18, 43, 18, '2015-11-04'),  
(19, 30, 2, '2018-08-10'),  
(20, 48, 24, '2015-05-13'),  
(21, 71, 5, '2016-09-05'),  
(22, 35, 3, '2016-07-03'),  
(23, 57, 1, '2015-03-17'),  
(24, 23, 25, '2017-08-16'),  
(25, 20, 12, '2018-07-24'),  
(26, 25, 7, '2015-01-31'),  
(27, 72, 29, '2016-04-10'),  
(28, 74, 20, '2017-07-31'),  
(29, 53, 14, '2016-02-20'),  
(30, 32, 10, '2017-07-24'),  
(31, 12, 15, '2018-04-25'),  
(32, 77, 13, '2017-06-09'),  
(33, 30, 4, '2017-10-24'),  
(34, 37, 24, '2016-01-14'),  
(35, 27, 26, '2017-06-05'),  
(36, 1, 16, '2018-05-06'),  
(37, 21, 9, '2016-03-19'),  
(38, 69, 28, '2017-03-29'),  
(39, 17, 19, '2017-03-14'),  
(40, 8, 9, '2016-04-22'),  
(41, 63, 18, '2015-01-25'),  
(42, 65, 20, '2016-10-10'),  
(43, 51, 19, '2015-07-28'),  
(44, 23, 12, '2017-01-25'),  
(45, 17, 4, '2017-04-18'),  
(46, 68, 5, '2016-09-06'),  
(47, 46, 13, '2017-09-30'),  
(48, 15, 13, '2017-07-05'),  
(49, 11, 19, '2017-12-14'),  
(50, 78, 15, '2017-01-26'),  
(51, 47, 9, '2015-03-03'),  
(52, 68, 7, '2016-05-26'),  
(53, 37, 26, '2017-02-06'),  
(54, 48, 27, '2015-12-30'),  
(55, 9, 21, '2017-10-21'),  
(56, 29, 8, '2018-04-01'),  
(57, 64, 18, '2017-08-29'),  
(58, 61, 26, '2018-02-21'),  
(59, 39, 28, '2016-07-26'),  
(60, 73, 18, '2018-08-22'),  
(61, 11, 13, '2018-01-17'),  
(62, 45, 6, '2016-07-20'),  
(63, 33, 13, '2018-03-18'),  
(64, 10, 17, '2016-06-06'),  
(65, 28, 18, '2017-02-17'),  
(66, 51, 3, '2016-12-09'),  
(67, 29, 2, '2015-09-18'),  
(68, 28, 30, '2017-09-14'),  
(69, 74, 20, '2015-12-12'),  
(70, 15, 22, '2015-01-14'),

(71, 57, 8, '2017-08-20'),  
(72, 2, 5, '2015-01-18'),  
(73, 74, 12, '2018-04-14'),  
(74, 51, 10, '2016-02-25'),  
(75, 25, 17, '2015-02-24'),  
(76, 45, 21, '2017-02-10'),  
(77, 27, 25, '2016-08-03'),  
(78, 32, 28, '2016-06-15'),  
(79, 71, 21, '2017-05-21'),  
(80, 75, 26, '2016-05-03'),  
(81, 56, 32, '2015-12-23'),  
(82, 26, 32, '2015-05-16'),  
(83, 66, 32, '2015-05-30'),  
(84, 57, 18, '2017-09-15'),  
(85, 40, 15, '2016-09-02'),  
(86, 65, 4, '2017-08-17'),  
(87, 54, 7, '2015-12-19'),  
(88, 29, 4, '2017-07-22'),  
(89, 44, 9, '2017-12-31'),  
(90, 56, 31, '2015-06-13'),  
(91, 17, 4, '2015-04-01'),  
(92, 35, 16, '2018-07-19'),  
(93, 22, 18, '2017-06-22'),  
(94, 39, 24, '2015-05-29'),  
(95, 63, 14, '2018-01-20'),  
(96, 53, 21, '2016-07-31'),  
(97, 40, 9, '2016-07-10'),  
(98, 52, 4, '2017-04-05'),  
(99, 27, 20, '2016-09-04'),  
(100, 72, 29, '2015-12-06'),  
(101, 49, 16, '2017-12-19'),  
(102, 6, 12, '2016-12-04'),  
(103, 74, 31, '2016-07-27'),  
(104, 48, 32, '2016-06-29'),  
(105, 69, 2, '2016-12-27'),  
(106, 60, 32, '2017-10-29'),  
(107, 45, 22, '2017-06-12'),  
(108, 42, 15, '2017-05-14'),  
(109, 79, 8, '2016-10-13'),  
(110, 70, 18, '2016-12-04'),  
(111, 34, 8, '2016-03-06'),  
(112, 43, 8, '2015-12-19'),  
(113, 42, 32, '2016-04-20'),  
(114, 67, 5, '2017-03-06'),  
(115, 80, 25, '2015-06-23'),  
(116, 54, 11, '2017-05-03'),  
(117, 34, 28, '2017-08-30'),  
(118, 65, 20, '2017-08-26'),  
(119, 61, 19, '2018-01-05'),  
(120, 38, 12, '2018-01-17'),  
(121, 51, 4, '2016-05-13'),  
(122, 7, 16, '2016-03-17'),  
(123, 46, 16, '2016-11-25'),  
(124, 75, 30, '2018-08-12'),

(125, 72, 32, '2015-03-12'),  
(126, 44, 17, '2015-06-15'),  
(127, 68, 15, '2016-02-21'),  
(128, 21, 1, '2016-06-19'),  
(129, 14, 25, '2016-10-10'),  
(130, 68, 21, '2016-05-27'),  
(131, 35, 20, '2015-03-19'),  
(132, 16, 27, '2016-08-08'),  
(133, 79, 31, '2018-03-07'),  
(134, 14, 17, '2018-04-28'),  
(135, 29, 28, '2018-03-11'),  
(136, 41, 4, '2018-08-08'),  
(137, 42, 3, '2016-02-23'),  
(138, 45, 3, '2017-07-10'),  
(139, 36, 16, '2018-07-19'),  
(140, 36, 30, '2015-08-07'),  
(141, 54, 32, '2018-03-14'),  
(142, 61, 15, '2017-03-28'),  
(143, 1, 13, '2018-05-17'),  
(144, 43, 1, '2015-05-14'),  
(145, 37, 14, '2015-07-30'),  
(146, 62, 17, '2015-09-19'),  
(147, 50, 22, '2016-12-02'),  
(148, 45, 1, '2016-07-24'),  
(149, 32, 17, '2018-03-10'),  
(150, 13, 28, '2016-02-14'),  
(151, 15, 9, '2018-08-11'),  
(152, 10, 19, '2018-08-29'),  
(153, 66, 3, '2016-11-27'),  
(154, 68, 29, '2017-07-12'),  
(155, 21, 14, '2018-06-27'),  
(156, 35, 9, '2016-01-22'),  
(157, 17, 24, '2016-08-25'),  
(158, 40, 21, '2015-07-09'),  
(159, 1, 24, '2016-03-28'),  
(160, 70, 27, '2015-07-10'),  
(161, 80, 26, '2016-04-24'),  
(162, 29, 5, '2015-10-18'),  
(163, 76, 12, '2018-04-25'),  
(164, 22, 4, '2016-12-24'),  
(165, 2, 2, '2017-10-26'),  
(166, 35, 13, '2016-02-28'),  
(167, 40, 8, '2017-10-02'),  
(168, 68, 9, '2016-01-03'),  
(169, 32, 5, '2016-11-13'),  
(170, 34, 17, '2016-09-15'),  
(171, 34, 16, '2018-04-13'),  
(172, 80, 30, '2016-10-13'),  
(173, 20, 32, '2015-11-17'),  
(174, 36, 10, '2017-09-01'),  
(175, 78, 12, '2018-06-27'),  
(176, 57, 8, '2016-03-22'),  
(177, 75, 11, '2017-06-27'),  
(178, 71, 10, '2015-08-01'),

(179, 48, 22, '2015-09-29'),  
(180, 19, 16, '2016-02-21'),  
(181, 79, 30, '2018-08-20'),  
(182, 70, 13, '2016-09-16'),  
(183, 30, 6, '2017-02-10'),  
(184, 45, 12, '2017-10-12'),  
(185, 30, 27, '2016-11-23'),  
(186, 26, 3, '2016-08-13'),  
(187, 66, 6, '2017-01-14'),  
(188, 47, 15, '2016-02-10'),  
(189, 53, 30, '2018-08-08'),  
(190, 80, 16, '2016-03-31'),  
(191, 70, 13, '2018-02-03'),  
(192, 14, 25, '2016-03-27'),  
(193, 46, 22, '2016-01-13'),  
(194, 30, 32, '2015-08-06'),  
(195, 60, 14, '2016-11-27'),  
(196, 14, 13, '2018-05-23'),  
(197, 71, 15, '2016-06-22'),  
(198, 38, 21, '2015-12-27'),  
(199, 69, 30, '2017-04-29'),  
(200, 49, 31, '2018-06-03'),  
(201, 28, 28, '2015-05-29'),  
(202, 49, 3, '2016-08-30'),  
(203, 75, 1, '2015-10-29'),  
(204, 78, 3, '2017-05-12'),  
(205, 43, 18, '2015-03-25'),  
(206, 27, 21, '2016-02-22'),  
(207, 64, 22, '2015-04-03'),  
(208, 21, 11, '2017-12-09'),  
(209, 66, 29, '2016-12-20'),  
(210, 45, 13, '2017-04-15'),  
(211, 48, 30, '2015-01-31'),  
(212, 20, 25, '2017-12-20'),  
(213, 41, 20, '2018-01-29'),  
(214, 51, 12, '2015-07-05'),  
(215, 5, 1, '2015-04-12'),  
(216, 40, 3, '2018-02-24'),  
(217, 79, 4, '2018-06-27'),  
(218, 15, 10, '2016-11-01'),  
(219, 42, 22, '2016-12-28'),  
(220, 17, 9, '2018-01-29'),  
(221, 38, 13, '2016-05-09'),  
(222, 79, 2, '2017-12-06'),  
(223, 74, 3, '2015-12-07'),  
(224, 46, 8, '2016-06-05'),  
(225, 78, 22, '2018-08-11'),  
(226, 45, 2, '2015-04-20'),  
(227, 72, 31, '2015-11-11'),  
(228, 18, 17, '2015-03-21'),  
(229, 29, 3, '2017-08-13'),  
(230, 66, 11, '2018-06-05'),  
(231, 36, 16, '2016-04-28'),  
(232, 26, 2, '2016-10-23'),

(233, 32, 1, '2017-10-31'),  
(234, 62, 14, '2017-07-25'),  
(235, 12, 4, '2015-07-08'),  
(236, 38, 32, '2015-02-24'),  
(237, 29, 16, '2016-07-28'),  
(238, 36, 25, '2017-05-07'),  
(239, 76, 7, '2015-06-13'),  
(240, 28, 16, '2016-08-15'),  
(241, 60, 13, '2016-08-26'),  
(242, 8, 3, '2017-07-28'),  
(243, 25, 1, '2016-07-30'),  
(244, 62, 29, '2018-08-24'),  
(245, 51, 8, '2016-09-01'),  
(246, 27, 23, '2015-02-08'),  
(247, 69, 12, '2018-06-25'),  
(248, 51, 12, '2015-07-04'),  
(249, 7, 4, '2015-05-01'),  
(250, 31, 15, '2017-10-29'),  
(251, 14, 23, '2015-01-15'),  
(252, 14, 1, '2018-05-21'),  
(253, 39, 25, '2015-12-26'),  
(254, 79, 24, '2016-05-31'),  
(255, 40, 15, '2016-03-18'),  
(256, 51, 13, '2018-04-13'),  
(257, 61, 1, '2015-02-11'),  
(258, 15, 24, '2018-03-02'),  
(259, 10, 22, '2018-01-21'),  
(260, 67, 10, '2017-07-08'),  
(261, 79, 11, '2016-12-11'),  
(262, 19, 32, '2016-05-04'),  
(263, 35, 11, '2017-08-01'),  
(264, 27, 13, '2017-12-15'),  
(265, 30, 22, '2015-12-22'),  
(266, 8, 7, '2015-06-26'),  
(267, 70, 9, '2016-03-20'),  
(268, 56, 18, '2016-01-29'),  
(269, 13, 19, '2015-03-06'),  
(270, 61, 2, '2016-06-18'),  
(271, 47, 13, '2017-09-18'),  
(272, 30, 22, '2016-02-19'),  
(273, 18, 22, '2016-12-31'),  
(274, 34, 29, '2017-10-27'),  
(275, 32, 21, '2015-06-03'),  
(276, 9, 28, '2016-03-30'),  
(277, 62, 24, '2015-03-23'),  
(278, 44, 22, '2017-04-29'),  
(279, 27, 5, '2015-03-25'),  
(280, 61, 28, '2017-07-14'),  
(281, 5, 13, '2016-12-04'),  
(282, 43, 19, '2018-03-15'),  
(283, 34, 19, '2016-06-05'),  
(284, 35, 5, '2018-02-19'),  
(285, 13, 12, '2016-09-23'),  
(286, 74, 18, '2016-12-26'),

(287, 70, 31, '2017-08-15'),  
(288, 42, 17, '2016-06-15'),  
(289, 51, 24, '2018-07-30'),  
(290, 45, 30, '2015-01-15'),  
(291, 70, 17, '2017-10-07'),  
(292, 77, 7, '2017-01-06'),  
(293, 74, 25, '2015-09-25'),  
(294, 47, 14, '2018-02-01'),  
(295, 10, 2, '2017-04-18'),  
(296, 16, 21, '2016-10-03'),  
(297, 48, 5, '2016-09-17'),  
(298, 72, 3, '2017-02-10'),  
(299, 26, 23, '2016-03-01'),  
(300, 49, 23, '2016-10-25');

**-- I AM CREATING THIS SPACE TAB MAKING IT  
EASIER FOR ME TO COPY THE CODE**  
**-- I CAN EASILY SEE WHERE I HAVE COPIED THE CODE  
TO EDIT AND THEN INSERT INTO TXT**  
**-- THIS IS JUST HERE FOR ME TO EASILY COPY AND  
PASTE THE CODE**

-- Now that I have populated the database as instructed in Objective B  
-- I will now continue with Objective C and the queries

## **-- Objective C Question 1:**

-- Display all contents of the Clients table  
SELECT \* FROM Clients;

'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'
```

-- I AM CREATING THIS SPACE TAB MAKING IT  
EASIER FOR ME TO COPY THE CODE

-- I CAN EASILY SEE WHERE I HAVE COPIED THE CODE  
TO EDIT AND THEN INSERT INTO TXT

-- THIS IS JUST HERE FOR ME TO EASILY COPY AND  
PASTE THE CODE

## -- Objective C Question 2:

```
-- Display First names, last names, ages and occupations of all clients  
-- This is tough to do since I can not use the TIMESTAMPDIFF command since only birth year  
is given  
-- My only workaround this is to use an approximate age  
-- Instead of Age I will name the column ApproximateAge  
    -- To do this I will simply subtract the current date and ClientDoB  
    -- I hope this is the correct way to do it since no birth date was given
```

```
SELECT  
    ClientFirstName,  
    ClientLastName,  
    Occupation,  
    YEAR(CURDATE()) - ClientDoB AS ApproximateAge  
FROM  
    Clients;  
-- The primary key ClientID is used to order the results
```

```
'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'

**-- I AM CREATING THIS SPACE TAB MAKING IT  
EASIER FOR ME TO COPY THE CODE**

**-- I CAN EASILY SEE WHERE I HAVE COPIED THE CODE  
TO EDIT AND THEN INSERT INTO TXT**

**-- THIS IS JUST HERE FOR ME TO EASILY COPY AND  
PASTE THE CODE**

### **-- Objective C Question 3:**

-- First and Last names of clients that borrowed books in March 2018  
-- For me to find this I will need to join the Clients and Borrowers table  
-- I will join them using the ClientID field  
-- I will also create an Alias for both tables  
    -- Clients will use the alias 'c'  
    -- Borrowers will use the alias 'b'  
-- Then I will filter the results to only include:

```

-- Borrow year = 2018 and Borrow month = 3

SELECT -- Using 'c' and 'b' for Alias
    c.ClientFirstName,
    c.ClientLastName,
    b.BorrowDate -- I included this to make sure the date retrieved was correct
FROM
    Clients c
JOIN
    Borrowers b
    ON c.ClientID = b.ClientID
        -- Joining on ClientID
WHERE
-- This will filter for books borrowed on March 2018
    YEAR(b.BorrowDate) = 2018
    AND
    MONTH(b.BorrowDate) = 3;

'Gerard','Jackson','2018-03-02'
'Tyler','Baker','2018-03-11'
'Angelina','Rowe','2018-03-10'
'Marcia','Rowe','2018-03-18'
'Carson','Byrne','2018-03-15'
'Katy','Lloyd','2018-03-14'
'Alysha','Lambert','2018-03-07'
'Maia','Grant','2018-03-18'

```

**-- I AM CREATING THIS SPACE TAB MAKING IT  
EASIER FOR ME TO COPY THE CODE**  
**-- I CAN EASILY SEE WHERE I HAVE COPIED THE CODE  
TO EDIT AND THEN INSERT INTO TXT**  
**-- THIS IS JUST HERE FOR ME TO EASILY COPY AND  
PASTE THE CODE**

## **-- Objective C Question 4:**

- First and Last names of the top 5 authors clients borrowed in 2017
- For this question I will be joining the Books table with the Borrowers table
- I will also join the Authors table with the Books table
- Books will have an alias of 'bk'
- Borrowers will have an alias of 'b'

```

-- Authors will have an alias of 'a'

SELECT
    a.AuthorFirstName,
    a.AuthorLastName,
    -- Alias 'a' is used for Authors table
    COUNT(*) as BorrowCount
    -- This will count the time books borrowed by each Author

FROM
    Borrowers b
    -- Alias 'b' is used for the Borrowers table

JOIN
    Books bk
    -- Alias 'bk' used for the Books table
    ON
    b.BookID = bk.BookID
    -- Links the Books and Borrowers table

JOIN
    Authors a
    ON
    bk.BookAuthor = a.AuthorID
    -- Links the Authors and Books table

WHERE
    YEAR(b.BorrowDate) = 2017
    -- Filtering for only this year
GROUP BY
    a.AuthorID
    -- Groups the AuthorID from the Author table and count the borrowing for each Author
ORDER BY
    BorrowCount DESC
    -- Orders the count by descending order
LIMIT
    5; -- This will give me the top 5 authors

'Sofia','Smith','7'
'Elena','Martin','7'
'Logan','Moore','7'
'Maria','Brown','6'
'Zoe','Roy','5'

```

-- I AM CREATING THIS SPACE TAB MAKING IT  
EASIER FOR ME TO COPY THE CODE  
-- I CAN EASILY SEE WHERE I HAVE COPIED THE CODE  
TO EDIT AND THEN INSERT INTO TXT  
-- THIS IS JUST HERE FOR ME TO EASILY COPY AND  
PASTE THE CODE

## -- Objective C Question 5:

-- Nationalities of the least 5 authors that clients borrowed during the years 2015-2017  
-- I will use the AuthorID and BookAuthor fields as the link between Authors and Books  
-- I will use the BookID field to link the Books and Borrowers  
-- I will filter the borrow dates for only 2015-2017  
-- The Alias for Authors, Books and Borrowers ('a', 'bk' and 'b') will be used again  
-- I will use the COUNT and Limit to get the bottom 5 results

```
SELECT
    a.AuthorNationality,
    COUNT(*) as BorrowCount
    -- I am selecting the nationality for the Author
    -- Counting the total books for each Author
FROM
    Borrowers b
JOIN
    Books bk ON b.BookID= bk.BookID
    -- Joining the Borrowers and Books table with the BookID field
JOIN
    Authors a ON bk.BookAuthor = a.AuthorID
    -- Joining the Authors and Book table
WHERE
    YEAR(b.BorrowDate) BETWEEN 2015 AND 2017
    -- This will give me results for the indicated years in the question
GROUP BY
    a.AuthorID
    -- This is to count the borrowings for each Author
ORDER BY
    BorrowCount ASC
    -- This will sort the list from least borrowed first
LIMIT 5;
-- This will give me the 5 results only
```

'Spain','3'  
'USA','5'  
'Canada','5'

'USA','6'  
'Great Britain','6'

-- I AM CREATING THIS SPACE TAB MAKING IT  
EASIER FOR ME TO COPY THE CODE  
-- I CAN EASILY SEE WHERE I HAVE COPIED THE CODE  
TO EDIT AND THEN INSERT INTO TXT  
-- THIS IS JUST HERE FOR ME TO EASILY COPY AND  
PASTE THE CODE

## -- Objective C Question 6:

-- The books that was most borrowed during the years 2015-2017  
-- I will use the BookID field from Borrowers and Books (Alias 'b' and 'bk')  
-- I will group the BookID from the Borrowers table to keep count  
-- I will filter the dates from 2015-2017  
-- I will order by DESC to show the top books

```
SELECT
    bk.BookTitle,
    COUNT(*) as BorrowCount
    -- Using the title of the book from the Books table with the correct Alias
FROM
    Borrowers b
    -- Using the Alias for the Borrowers table
JOIN
    Books bk ON b.BookID = bk.BookID
    -- Joining the tables with BookID field
WHERE
    YEAR(b.BorrowDate) BETWEEN 2015 AND 2017
    -- Filtering the correct years from the question
GROUP BY
    bk.BookID
    -- Counting the borrowings per book
ORDER BY
    BorrowCount DESC
```

-- Sorting the count from highest to lowest  
LIMIT 5;  
-- Limiting to just the top 5 for top books

'The perfect match','13'  
'Electrical transformers','12'  
'Green nature','11'  
'Positive figures','11'  
'Performance evaluation','11'

**-- I AM CREATING THIS SPACE TAB MAKING IT EASIER FOR ME TO COPY THE CODE**  
**-- I CAN EASILY SEE WHERE I HAVE COPIED THE CODE TO EDIT AND THEN INSERT INTO TXT**  
**-- THIS IS JUST HERE FOR ME TO EASILY COPY AND PASTE THE CODE**

## **-- Objective C Question 7:**

-- Top borrowed genres for client born in years 1970-1980  
-- This question was a little more complex and I realized that I can use a subquery  
-- The subquery will filter the Clients DoB making it easier for me

```
SELECT
    bk.Genre,
    COUNT(*) as BorrowCount
    -- This will select the genre and count the total number each genre was borrowed
FROM (
    SELECT ClientID
    FROM Clients
    WHERE ClientDoB
    BETWEEN 1970 AND 1980)
    as filteredClients
    -- This is the subquery to only list for clients born between 1970-1980
JOIN
    Borrowers b ON filteredClients.ClientID = b.ClientID
    -- Joins the Borrowers table on ClientID
JOIN
    Books bk ON b.BookID = bk.BookID
```

```

-- Joins the Books table with the correct Alias and on BookID
GROUP BY
    bk.Genre
    -- Grouping the books by Genre
ORDER BY
    BorrowCount DESC;
    -- Allows to view the order from most borrowed

'Science','24'
'Fiction','16'
'Well being','15'
'Humor','5'
'Society','4'
'History','3'
'Law','3'
'Children','3'
'Literature','3'

```

**-- I AM CREATING THIS SPACE TAB MAKING IT  
EASIER FOR ME TO COPY THE CODE**  
**-- I CAN EASILY SEE WHERE I HAVE COPIED THE CODE  
TO EDIT AND THEN INSERT INTO TXT**  
**-- THIS IS JUST HERE FOR ME TO EASILY COPY AND  
PASTE THE CODE**

## **-- Objective C Question 8:**

- Top 5 occupations that borrowed the most in 2016
- Here I will sue a temporary table that will take ClientID from the Borrowers table for 2016
- I can now perform a join with just clients borrowed in the specific year
- I can join with the Clients table to get the occupations and sort the list accordingly

```

CREATE TEMPORARY TABLE Temp2016Borrowers AS
SELECT ClientID
FROM Borrowers
WHERE YEAR(BorrowDate) = 2016;
-- Creating the temporary table for 2016 ONLY
-- I will give this table an Alias as well

```

```

SELECT
    c.Occupation,
    COUNT(*) as BorrowCount
    -- Getting the occupations from the Client table and counting borrowings
FROM
    Temp2016Borrowers tb
    -- Alias for temp table
JOIN
    Clients c ON tb.ClientID = c.ClientID
    -- Joining on ClientID
GROUP BY
    c.Occupation
    -- Tally the occupation
ORDER BY
    BorrowCount DESC
    -- Give me the highest to lowest order
LIMIT 5;
-- This will give me the top 5 as the question requests
-- DROP TEMPORARY TABLE IF EXISTS Temp2016Borrowers; will be used when I am done

'Student','32'
'Bus Driver','8'
'Dentist','6'
'Computer Programmer','6'
'Police Officer','5'

```

**-- I AM CREATING THIS SPACE TAB MAKING IT  
EASIER FOR ME TO COPY THE CODE**  
**-- I CAN EASILY SEE WHERE I HAVE COPIED THE CODE  
TO EDIT AND THEN INSERT INTO TXT**  
**-- THIS IS JUST HERE FOR ME TO EASILY COPY AND  
PASTE THE CODE**

## **-- Objective C Question 9:**

-- Average number of borrowed books by job title  
-- Since the Borrowers table is quite large, I will group total borrowings per Client ID

-- This will be a temporary table  
-- The Alias 'tb' will work since I have removed the temp table in question 8 before this query

```
CREATE TEMPORARY TABLE TotalBorrowings AS
SELECT
    ClientID,
    COUNT(*) as borrowCount
-- Creating the temporary table counting the total borrowings per client
FROM
    Borrowers
GROUP BY
    ClientID;
-- This will group all the borrows per Client ID making it more efficient
```

```
SELECT
    -- Calculating the average number of books per Occupation
    c.Occupation,
    AVG(tb.borrowCount) as AverageBorrowCount
    -- Getting the average borrow count per Occupation
FROM
    TotalBorrowings tb
    -- Using the temporary table that will give the total borrowings
JOIN
    Clients c ON tb.ClientID = c.ClientID
    -- Joining the Clients table using the ClientID to get the Occupation
GROUP BY
    c.Occupation
    -- Grouping by Occupation and getting the average for each
ORDER BY
    AverageBorrowCount DESC;
-- Although this was not in the question, I wanted to see who borrowed the most

-- DROP TEMPORARY TABLE IF EXISTS TotalBorrowings;
-- Like Question 8, I will issue the drop command after the query
```

```
'Nurse','7.0000'
'Computer Security Manager','6.0000'
'Dentist','5.6667'
'Computer Programmer','5.6667'
'Cashier','5.0000'
'Oil Exploration Engineer','5.0000'
'Manufacturing Director','5.0000'
'Food Scientist','5.0000'
'Police Officer','4.5000'
'Student','4.4211'
'Violinist','4.0000'
'Insurance Agent','4.0000'
'Bus Driver','4.0000'
'Ship Engineer','4.0000'
'Doctor','4.0000'
'HR Clerk','4.0000'
```

```
'Systems Analyst','4.0000'  
'School Teacher','3.6000'  
'Professor','3.5000'  
'Firefighter','3.2500'  
'Repair Worker','3.0000'  
'Payroll Clerk','3.0000'  
'Computer Engineer','3.0000'  
'Manager','3.0000'  
'Licensed Massage Therapist','2.0000'  
'Parquet Floor Layer','2.0000'  
'Aircraft Electrician','2.0000'  
'Window Washer','2.0000'  
'Securities Clerk','2.0000'  
'School Psychologist','2.0000'  
'Health Educator','2.0000'  
'Security Agent','2.0000'
```

-- I AM CREATING THIS SPACE TAB MAKING IT  
EASIER FOR ME TO COPY THE CODE

-- I CAN EASILY SEE WHERE I HAVE COPIED THE CODE  
TO EDIT AND THEN INSERT INTO TXT

-- THIS IS JUST HERE FOR ME TO EASILY COPY AND  
PASTE THE CODE

## -- Objective C Question 10:

-- Create a VIEW and display the titles that were borrowed by at least 20% of clients  
-- For this question I will go back to the subquery approach as the code is simpler  
-- The subquery will be used to calculate the % of clients who borrowed each book  
-- I will name the view BooksBorrowedBy20Percent

```
CREATE VIEW BooksBorrowedBy20Percent AS  
SELECT  
    bk.BookID,  
    bk.BookTitle,  
    -- Selecting the BookID and BookTitle from the Books table  
    (COUNT(DISTINCT b.ClientID) / (SELECT COUNT(*) FROM Clients) * 100) as  
    BorrowPercentage  
    -- Equation used and Distinct to make sure the Client is unique per book borrowed  
FROM  
    Borrowers b
```

```
-- 'b' Alias used beginning with the Borrowers table
JOIN
    Books bk ON b.BookID = bk.BookID
    -- Joining the Books table with the BookID to get the Books
GROUP BY
    bk.BookID,
    bk.BookTitle
    -- Grouping the BookID and Title to tally the borrowing
HAVING
    BorrowPercentage >= 20;
    -- This will only show Books borrowed by at least 20%
```

```
SELECT *
FROM BooksBorrowedBy20Percent;
-- Retrieve data from the VIEW
```

'13','Electrical transformers','21.2500'

-- I AM CREATING THIS SPACE TAB MAKING IT  
EASIER FOR ME TO COPY THE CODE  
-- I CAN EASILY SEE WHERE I HAVE COPIED THE CODE  
TO EDIT AND THEN INSERT INTO TXT  
-- THIS IS JUST HERE FOR ME TO EASILY COPY AND  
PASTE THE CODE

## -- Objective C Question 11:

```
-- The top month of borrows in 2017
-- I will use a simple query where I will select the month from the Borrows table
-- I will then filter the year to 2017 group by the month
-- I will order in DESC and Limit to 1

SELECT
    MONTH(b.BorrowDate) AS BorrowMonth,
    COUNT(*) AS TotalBorrows
    -- This will get the month from the Borrows Table and count the total borrows per month
    FROM
```

```

    Borrowers b
-- Using the same Alias as previous queries
WHERE
    YEAR(b.BorrowDate) = 2017
-- This will filter the results for the specific year
GROUP BY
    BorrowMonth
-- Refining the query to group it by month
ORDER BY
    TotalBorrows DESC
-- This will give the top borrows per month first
LIMIT 1;
-- This will display the month in 2017 with the most borrows

```

'8','10'  
The top month is 8 and Total borrows is 10

-- I AM CREATING THIS SPACE TAB MAKING IT  
EASIER FOR ME TO COPY THE CODE  
-- I CAN EASILY SEE WHERE I HAVE COPIED THE CODE  
TO EDIT AND THEN INSERT INTO TXT  
-- THIS IS JUST HERE FOR ME TO EASILY COPY AND  
PASTE THE CODE

## -- Objective C Question 12:

- Average number of borrows by age
- Like mentioned earlier, the DoB only contains the year
  - I will have to create an inner subquery to calculate the approximate age
  - I will also get the borrow count for each client here as well
- The outer query will give me the average borrows for the approximate age
- Since the month and day was not given this will not be the most accurate
  - This is why I have named it ApproximateAge

```

SELECT
    ApproximateAge,
    AVG(BorrowCount) as AverageBorrows
-- Outer query to get the average borrows per approximate age

```

```

FROM
    -- Subquery to get the approximate age and count borrows per client
    (SELECT
        c.ClientID,
        YEAR(CURRENT_DATE) - c.ClientDoB as ApproximateAge,
        COUNT(b.BookID) as BorrowCount
    FROM
        Clients c
    -- Still using the Alias from previous queries
    JOIN
        Borrowers b ON c.ClientID = b.ClientID
    -- Joining the Borrowers table with the Clients table using Client ID
    GROUP BY
        c.ClientID, ApproximateAge) as AgeBorrowData
    -- Grouping by Client ID and Approximate Age
GROUP BY
    ApproximateAge
ORDER BY
    ApproximateAge;
-- Ordering by Approximate Age

```

```

'14','2.3333'
'16','6.0000'
'17','5.0000'
'18','5.5000'
'19','4.5000'
'20','3.0000'
'21','5.0000'
'22','2.0000'
'23','4.5000'
'25','3.6667'
'26','2.0000'
'28','2.0000'
'29','4.5000'
'30','10.0000'
'32','3.0000'
'34','5.5000'
'37','2.0000'
'38','3.0000'
'39','4.0000'
'40','5.5000'
'41','3.7500'
'42','3.0000'
'43','2.0000'
'44','1.0000'
'45','4.3333'
'46','5.5000'
'47','3.0000'
'48','3.5000'
'49','2.6667'
'50','3.2500'
'51','3.6667'
'54','4.5000'

```

```
'56','4.0000'  
'57','3.0000'  
'58','1.0000'  
'61','5.0000'  
'62','3.0000'  
'64','3.6667'
```

-- I AM CREATING THIS SPACE TAB MAKING IT  
EASIER FOR ME TO COPY THE CODE

-- I CAN EASILY SEE WHERE I HAVE COPIED THE CODE  
TO EDIT AND THEN INSERT INTO TXT

-- THIS IS JUST HERE FOR ME TO EASILY COPY AND  
PASTE THE CODE

### -- Objective C Question 13:

-- The oldest and youngest clients of the library

-- For this one I almost forgot to use the UNION operator and was going to simply do two queries

-- This query is a little more tricky since the ClientsDoB only indicates the year

-- Knowing that multiple Clients will have the same year I will use the WHERE and Max/Min statements

-- The MAX will give me the clients with the youngest birth year as that is the highest value

-- The MIN will give me the clients with the oldest birth year as that is the lowest values

```
SELECT  
    ClientID,  
    ClientFirstName,  
    ClientLastName,  
    ClientDoB  
FROM  
    Clients  
WHERE  
    ClientDoB = (SELECT MAX(ClientDoB)  
    FROM  
    Clients)  
UNION -- Used to combine both queries
```

```

SELECT
    ClientID,
    ClientFirstName,
    ClientLastName,
    ClientDoB
FROM
    Clients
WHERE
    ClientDoB = (SELECT MIN(ClientDoB)
    FROM
        Clients);

```

# ClientID, ClientFirstName, ClientLastName, ClientDoB  
'2', 'Alina', 'Morton', 2010  
'22', 'Alina', 'Austin', 2010  
'65', 'Joan', 'Grant', 2010  
'12', 'Mya', 'Austin', 1960  
'38', 'Caroline', 'Clark', 1960  
'72', 'Annabel', 'Williamson', 1960

-- I AM CREATING THIS SPACE TAB MAKING IT  
**EASIER FOR ME TO COPY THE CODE**  
**-- I CAN EASILY SEE WHERE I HAVE COPIED THE CODE**  
**TO EDIT AND THEN INSERT INTO TXT**  
**-- THIS IS JUST HERE FOR ME TO EASILY COPY AND**  
**PASTE THE CODE**

## -- Objective C Question 14:

-- First and Last names of authors that wrote books in more than one genre  
-- For this query I will join the Authors and Books table using the AuthorID and BookAuthor column  
-- I will continue using the Aliases 'a' for Authors and 'bk' for Books  
-- I will use Distinct in the Genre column to get different values per Author  
-- This will show that the Author has written books in multiple Genres  
-- I will then show the values of being greater than 1

```

SELECT
    a.AuthorID,
    a.AuthorFirstName,
    a.AuthorLastName,

```

```

COUNT(DISTINCT bk.Genre) AS GenreCount
-- This will count distinct genres for the Author
FROM
    Authors a
    -- Starting with the Authors table with the Alias mentioned above
JOIN
    Books bk
    -- Alias for the Books table
    ON a.AuthorID = bk.BookAuthor
    -- Joining the Books and Authors table with the AuthorID and BookAuthor
GROUP BY
    a.AuthorID,
    a.AuthorFirstName,
    a.AuthorLastName
    -- Grouping it by the Author details to get the distinct count
HAVING
    COUNT(DISTINCT bk.Genre) > 1;
-- This will filter out Authors and only give me Authors who wrote in more than 1 Genre

```

#	AuthorID	AuthorFirstName	AuthorLastName	GenreCount
3	Elena	Martin	2	

## -- INDEXES

-- Looking over at the assignment I realized that I missed on creating the index as it was reminded at the end of the query questions

-- I have thought about some of the queries and will create Indexes that would have benefited me in the comments below

-- Joining the Books with Authors would make the joins easier

- CREATE INDEX idx\_bookauthor
  - ON Books (BookAuthor);

-- Categorizing books on Genre

- CREATE INDEX idx\_genre
  - ON Books (Genre);

-- The ClientID on the Borrowers and Clients table would be useful since some of the queries involved it

- CREATE INDEX idx\_borrower\_client
  - ON Borrowers (ClientID);

-- Linking the BookID and the Borrowers table to join the Books table

- CREATE INDEX idx\_borrower\_book ON Borrowers (BookID);

-- Although this isn't the most ideal I found it a good learning experience to be able to use indexes to make queries more efficient

# **SCREENSHOTS OF QUERIES:**

Administration   Schemas   Query 2   CS 204 Project Library - Schema

**SCHEMAS**

Filter objects

**CS 204 Project Library**

- Tables
- Views
- Stored Procedures
- Functions

```

21 ('Liam', 'Taylor', 'Canada'),
22 ('William', 'Thomas', 'Great Britain'),
23 ('Logan', 'Moore', 'Canada'),
24 ('Oliver', 'Martin', 'France'),
25 ('Alysha', 'Thompson', 'Canada'),
26 ('Isabelle', 'Lee', 'Canada'),
27 ('Emily', 'Clark', 'USA'),
28 ('John', 'Young', 'China'),
29 ('David', 'Wright', 'Canada'),
30 ('Thomas', 'Scott', 'Canada'),
31 ('Helena', 'Adams', 'Canada'),
32 ('Sofia', 'Carter', 'USA'),
33 ('Liam', 'Parker', 'Canada'),
34 ('Emily', 'Murphy', 'USA');

35
36 -- I am double checking my creation of the Authors table by viewing the table
37
38 • SELECT * FROM Authors;

```

100% 23:38

**Result Grid** Filter Rows: Search Edit: Export/Import:

AuthorID	AuthorFirstName	AuthorLastName	AuthorNational...
1	Sofia	Smith	Canada
2	Maria	Brown	Brazil
3	Elena	Martin	Mexico
4	Zoe	Roy	France
5	Sebastian	Lavoie	Canada
6	Dylan	Garcia	Spain
7	Ian	Cruz	Mexico
8	Lucas	Smith	USA
9	Fabian	Wilson	USA
10	Liam	Taylor	Canada
11	William	Thomas	Great Britain
12	Logan	Moore	Canada
13	Oliver	Martin	France
14	Alysha	Thompson	Canada
15	Isabelle	Lee	Canada
16	Emily	Clark	USA
17	John	Young	China
18	David	Wright	Canada
19	Thomas	Scott	Canada
20	Helena	Adams	Canada
21	Sofia	Carter	USA
22	Liam	Parker	Canada
23	Emily	Murphy	USA
NONE	NONE	NONE	NONE



Limit to 1000 rows



Filter objects

## CS 204 Project Library

Tables

Views

Stored Procedures

Functions

```

97  -- I will use the update command below to fix this error
98      -- UPDATE Books
99      -- SET BookAuthor = 3
100     -- WHERE BookTitle = 'Positive Figures';
101     -- This actually did not correct the issue since the table would not update due to the
102     -- It was showing a null table so I simply just corrected the entry in the table above
103
104     -- I will now view the books table to make sure the entries are correct
105 •   SELECT * FROM Books

```

100% 20:105

Result Grid



Filter Rows:

Search

Edit:



Export/Import:



BookID	BookTitle	BookAuthor	Genre
65	Build your database system	1	Science
66	The red wall	2	Fiction
67	The perfect match	3	Fiction
68	Digital Logic	4	Science
69	How to be a great lawyer	5	Law
70	Manage successful negotiations	6	Society
71	Pollution today	7	Science
72	A gray park	2	Fiction
73	How to be rich in one year	8	Humor
74	Their bright fate	9	Fiction
75	Black lines	10	Fiction
76	History of theater	11	Literat...
77	Electrical transformers	12	Science
78	Build your big data system	1	Science
79	Right and left	13	Children
80	Programming using Python	1	Science
81	Computer networks	14	Science
82	Performance evaluation	15	Science
83	Daily exercise	16	Well b...
84	The silver uniform	17	Fiction
85	Industrial revolution	18	History
86	Green nature	19	Well b...
87	Perfect football	20	Well b...
88	The chocolate love	21	Humor
89	Director and leader	22	Society
90	Play football every week	20	Well b...
91	Maya the bee	13	Children
92	Perfect rugby	20	Well b...
93	The end	23	Fiction
94	Computer security	1	Science
95	Participate	22	Society
96	Positive figures	3	Society
HULL	NULL	NULL	HULL

Administration      Schemas      Query 2      CS 204 Project Library - Schema      CS 204 w/o auto-increment - Schema      SQL File 3\*

**SCHEMAS**

Filter objects

**CS 204 Project Library**

- Tables
- Views
- Stored Procedures
- Functions

> CS 204 w/o auto-incre...

92  
93  
94     -- SELECT AuthorID, AuthorFirstName, AuthorLastName FROM Authors; showed that th  
95     -- I then looked at the books table and noticed that the last entry 'Positive fi  
96     -- It had an ID of 24 when it should be 3  
97     -- I will use the update command below to fix this error  
98         -- UPDATE Books  
99         -- SET BookAuthor = 3  
100        -- WHERE BookTitle = 'Positive Figures';  
101        -- This actually did not correct the issue since the table would not update due  
102        -- It was showing a null table so I simply just corrected the entry in the table  
103  
104        -- I will now view the books table to make sure the entries are correct with the  
105 •     **SELECT \* FROM Books;**

100%      21:105      1 error found

**Result Grid**      Filter Rows:  Search      Edit:      Export/Import:

BookID	BookTitle	BookAuthor	Genre
65	Build your database system	1	Science
66	The red wall	2	Fiction
67	The perfect match	3	Fiction
68	Digital Logic	4	Science
69	How to be a great lawyer	5	Law
70	Manage successful negotiations	6	Society
71	Pollution today	7	Science
72	A gray park	2	Fiction
73	How to be rich in one year	8	Humor
74	Their bright fate	9	Fiction
75	Black lines	10	Fiction
76	History of theater	11	Literat...
77	Electrical transformers	12	Science
78	Build your big data system	1	Science
79	Right and left	13	Children
80	Programming using Python	1	Science
81	Computer networks	14	Science
82	Performance evaluation	15	Science
83	Daily exercise	16	Well b...

Books 9      Apply

Local Instance - Warning - not supported

Administration Schemas

SCHEMAS

Filter objects

CS 204 Project Library

CS 204 w/o auto-incr...

Query 2 CS 204 w/o auto-increment - Schema SQL File 3\*

Limit to 1000 rows

89 -- The error code I got Error Code: 1452. Cannot add or update a child row:  
 90 -- a foreign key constraint fails (`cs 204 project library`.`books`, CONSTRAINT `books`  
 91  
 92  
 93 -- SELECT AuthorID, AuthorFirstName, AuthorLastName FROM Authors; showed that the Authors  
 94 -- I then looked at the books table and noticed that the last entry 'Positive figures' was  
 95 -- It had an ID of 24 when it should be 3  
 96 -- I will use the update command below to fix this error  
 97 -- UPDATE Books  
 98 -- SET BookAuthor = 3  
 99 -- WHERE BookTitle = 'Positive Figures';  
 100 -- This actually did not correct the issue since the table would not update due to the Au  
 101 -- It was showing a null table so I simply just corrected the entry in the table above.  
 102  
 103 • **SELECT \* FROM Books;** -- I am just running this to see if the BookID is correct :)

100% 21:103

**Result Grid** Filter Rows: Search Edit: Export/Import:

BookID	BookTitle	BookAuthor	Genre
1	Build your database system	1	Science
2	The red wall	2	Fiction
3	The perfect match	3	Fiction
4	Digital Logic	4	Science
5	How to be a great lawyer	5	Law
6	Manage successful negotiations	6	Society
7	Pollution today	7	Science
8	A gray park	2	Fiction
9	How to be rich in one year	8	Humor
10	Their bright fate	9	Fiction
11	Black lines	10	Fiction
12	History of theater	11	Literat...
13	Electrical transformers	12	Science
14	Build your big data system	1	Science
15	Right and left	13	Children
16	Programming using Python	1	Science
17	Computer networks	14	Science
18	Performance evaluation	15	Science
19	Daily exercise	16	Well b...
20	The silver uniform	17	Fiction
21	Industrial revolution	18	History

Books 1 Apply Revert

Action Output

Time	Action	Response
16:03:59	INSERT INTO Books (BookID,BookTitle,...)	32 row(s) affected Records: 32 Duplicates: 0 Warnings: 0
16:05:08	SELECT * FROM Books LIMIT 0, 1000	32 row(s) returned

197 (72, 'Annabel', 'Williamson', 1960, 'Cashier'),  
 198 (73, 'Mohamed', 'Waters', 1966, 'Insurance Agent'),  
 199 (74, 'Marion', 'Newman', 1970, 'Computer Programmer'),  
 200 (75, 'Ada', 'Williams', 1986, 'Computer Programmer'),  
 201 (76, 'Sean', 'Scott', 1983, 'Bus Driver'),  
 202 (77, 'Farrah', 'Scott', 1974, 'Ship Engineer'),  
 203 (78, 'Christine', 'Lambert', 1973, 'School Teacher'),  
 204 (79, 'Alysha', 'Lambert', 2007, 'Student'),  
 205 (80, 'Maia', 'Grant', 1984, 'School Teacher'));  
 206  
 207 • SELECT \* FROM CLIENTS;

100% 41:185

Result Grid Filter Rows: Search Edit: Export/Import:

ClientID	ClientFirstName	ClientLastName	ClientDoB	Occupation
8	Julie	McCarthy	1973	Professor
9	Ken	McCarthy	1974	Securities Clerk
10	Britany	O'Quinn	1984	Violinist
11	Conner	Gardner	1998	Licensed Massage...
12	Mya	Austin	1960	Parquet Floor Layer
13	Thierry	Rogers	2004	Student
14	Eloise	Rogers	1984	Computer Security...
15	Gerard	Jackson	1979	Oil Exploration En...
16	Randy	Day	1986	Aircraft Electrician
17	Jodie	Page	1990	Manufacturing Dire...
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

CLIENTS 2

Apply

Action Output



	Time	Action	Response
1	16:30:31	SELECT * FROM CLIENTS LIMIT 0, 1000	80 row(s) returned



## MySQL Workbench

g - not supported



Query 2

CS 204 w/o auto-increment - Schema

SQL File 3\*

```
524 (292, 77, 7, '2017-01-06'),  
525 (293, 74, 25, '2015-09-25'),  
526 (294, 47, 14, '2018-02-01'),  
527 (295, 10, 2, '2017-04-18'),  
528 (296, 16, 21, '2016-10-03'),  
529 (297, 48, 5, '2016-09-17'),  
530 (298, 72, 3, '2017-02-10'),  
531 (299, 26, 23, '2016-03-01'),  
532 (300, 49, 23, '2016-10-25');
```

533

534

535

536

537

538

```
-- I AM CREATING THIS SPACE TAB MAKING IT EASIER FOR ME TO COPY THE CODE  
-- I CAN EASILY SEE WHERE I HAVE COPIED THE CODE TO EDIT AND THEN INSERT INTO TXT  
-- THIS IS JUST HERE FOR ME TO EASILY COPY AND PASTE THE CODE
```

542

543

```
-- Now that I have populated the database as instructed in Objective B  
-- I will now continue with Objective C and the queries
```

546

```
-- Objective C Question 1:
```

```
-- Display all contents of the Clients table
```

```
549 • SELECT * FROM Clients;
```

550

100% 1:550

Result Grid



Filter Rows:

Search

Edit:



Export/Import:



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...
12	Mya	Austin	1960	Parquet Floor Layer
13	Thierry	Rogers	2004	Student
14	Eloise	Rogers	1984	Computer Security...
15	Gerard	Jackson	1970	Oil Exploration En...



Clients 3

Apply

Revert

Schemas      Create a new view in the active schema in the connected server      SQL File 3\*

556      -- I AM CREATING THIS SPACE TAB MAKING IT EASIER FOR ME TO COPY THE CODE  
 557      -- I CAN EASILY SEE WHERE I HAVE COPIED THE CODE TO EDIT AND THEN INSERT INTO TXT  
 558      -- THIS IS JUST HERE FOR ME TO EASILY COPY AND PASTE THE CODE  
 559  
 560  
 561  
 562      -- Objective C Question 2:  
 563      -- Display First names, last names, ages and occupations of all clients  
 564      -- This is tough to do since I can not use the TIMESTAMPDIFF command since only birth year is given  
 565      -- My only workaround this is to use an approximate age  
 566      -- Instead of Age I will name the column ApproximateAge  
 567          -- To do this I will simply subtract the current date and ClientDoB  
 568          -- I hope this is the correct way to do it since no birth date was given  
 569 •      **SELECT**  
 570          ClientFirstName,  
 571          ClientLastName,  
 572          Occupation,  
 573          **YEAR(CURDATE()) - ClientDoB AS ApproximateAge**  
 574      **FROM**  
 575          Clients;  
 576          -- The primary key ClientID is used to order the results|

100% 61:576

**Result Grid** Filter Rows: Search Export:

ClientFirstName	ClientLastName	Occupation	ApproximateAge
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...	26
Mya	Austin	Parquet Floor Layer	64
Thierry	Rogers	Student	20
Eloise	Rogers	Computer Security...	40
Gerard	Jackson	Oil Exploration En...	45

Result 5

! Read Only

Action Output

	Time	Action	Response
✓ 1	11:03:02	SELECT ClientFirstName, ClientLastName, Occupation, Y... 80 row(s) returned	





Limit to 1000 rows



```
579  
580  
581  
582  
583      -- I AM CREATING THIS SPACE TAB MAKING IT EASIER FOR ME TO COPY THE CODE  
584      -- I CAN EASILY SEE WHERE I HAVE COPIED THE CODE TO EDIT AND THEN INSERT INTO TXT  
585      -- THIS IS JUST HERE FOR ME TO EASILY COPY AND PASTE THE CODE  
586  
587  
588  
589      -- Objective C Question 3:  
590      -- First and Last names of clients that borrowed books in March 2018  
591      -- For me to find this I will need to join the Clients and Borrowers table  
592      -- I will join them using the ClientID field  
593      -- I will also create an Alias for both tables  
594          -- Clients will use the alias 'c'  
595          -- Borrowers will use the alias 'b'  
596      -- Then I will filter the results to only include:  
597          -- Borrow year = 2018 and Borrow month = 3  
598  
599 • SELECT -- Using 'c' and 'b' for Alias  
600     c.ClientFirstName
```

100% 29:613

Result Grid



Filter Rows:



Search

Export:



ClientFirstName ClientLastName BorrowDate

Gerard	Jackson	2018-03-02
Tyler	Baker	2018-03-11
Angelina	Rowe	2018-03-10
Marcia	Rowe	2018-03-18
Carson	Byrne	2018-03-15
Katy	Lloyd	2018-03-14
Alysha	Lambert	2018-03-07
Maia	Grant	2018-03-18



Result Grid



Form Editor



Field Types



Query Stats

Result 7

Read Only

Query 2

CS 204 w/o auto-increment - Schema

SQL File 3\*



Limit to 1000 rows



```
641   FROM
642     Borrowers b
643       -- Alias 'b' is used for the Borrowers table
644
645   JOIN
646     Books bk
647       -- Alias 'bk' used for the Books table
648   ON
649     b.BookID = bk.BookID
650       -- Links the Books and Borrowers table
651
652   JOIN
653     Authors a
654   ON
655     bk.BookAuthor = a.AuthorID
656       -- Links the Authors and Books table
657
658   WHERE
659     YEAR(b.BorrowDate) = 2017
660       -- Filtering for only this year
661   GROUP BY
662     a.AuthorID
663       -- Groups the AuthorID from the Author table and count the borrowing for each Author
664   ORDER BY
665     BorrowCount DESC
666       -- Orders the count by descending order
667   LIMIT
668     5; -- This will give me the top 5 authors
```

100% ▾ 5:652

Result Grid



Filter Rows:



Search

Export:



Fetch rows:



Result Grid

AuthorFirstName	AuthorLastName	BorrowCount
Sofia	Smith	7
Elena	Martin	7
Logan	Moore	7
Maria	Brown	6
Zoe	Roy	5

Result 8

Read Only



Form

Editor



Action Output ▾

Time Action

Response

✓ 1 12:14:17 SELECT a.AuthorFirstName, a.AuthorLastName, -- Alias 'a' is... 5 row(s) returned

Local Instance - Warning - not supported



Schemas

Query 2 CS 204 w/o auto-increment - Schema SQL File 3\*



Limit to 1000 rows



```

690 -- I will use the COUNT and Limit to get the bottom 5 results
691
692 • SELECT
693     a.AuthorNationality,
694     COUNT(*) as BorrowCount
695     -- I am selecting the nationality for the Author
696     -- Counting the total books for each Author
697     FROM
698     Borrowers b
699     JOIN
700     Books bk ON b.BookID= bk.BookID
701     -- Joining the Borrowers and Books table with the BookID field
702     JOIN
703     Authors a ON bk.BookAuthor = a.AuthorID
704     -- Joining the Authors and Book table
705     WHERE
706     YEAR(b.BorrowDate) BETWEEN 2015 AND 2017
707     -- This will give me results for the indicated years in the question
708     GROUP BY
709     a.AuthorID
710     -- This is to count the borrowings for each Author
711     ORDER BY
712         BorrowCount ASC
713         -- This will sort the list from least borrowed first
714     LIMIT 5;
715     -- This will give me the 5 results only

```

100% ◊ 17:712

Result Grid



Filter Rows:

Search

Export:



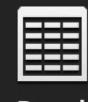
Fetch rows:



AuthorNational... BorrowCount

Spain	3
USA	5
Canada	5
USA	6
Great Britain	6

Result 9



Read Only

Action Output



	Time	Action	Response
✓ 1	12:59:25	SELECT a.AuthorNationality, COUNT(*) as BorrowCount -- I...	5 row(s) returned





```

776    -- Top borrowed genres for client born in years 1970-1980
777    -- This question was a little more complex and I realized that I can use a subquery
778    -- The subquery will filter the Clients DoB making it easier for me
779
780 • SELECT
781     bk.Genre,
782     COUNT(*) as BorrowCount
783     -- This will select the genre and count the total number each genre was borrowed
784     FROM (
785         SELECT ClientID
786         FROM Clients
787         WHERE ClientDoB
788         BETWEEN 1970 AND 1980)
789         as filteredClients
790         -- This is the subquery to only list for clients born between 1970-1980
791     JOIN
792         Borrowers b ON filteredClients.ClientID = b.ClientID
793         -- Joins the Borrowers table on ClientID
794     JOIN |
795         Books bk ON b.BookID = bk.BookID
796         -- Joins the Books table with the correct Alies and on BookID
797     GROUP BY
798     bk.Genre
799     -- Grouping the books by Genre
800     ORDER BY
801     BorrowCount DESC;
802     -- Allows to view the order from most borrowed

```

100% ▾ 6:794

Result Grid



Filter Rows:

Search

Export:

Genre	BorrowCount
Science	24
Fiction	16
Well being	15
Humor	5
Society	4
History	3
Law	3
Children	3
Literature	3

Result 11

! Read Only

Action Output ▾

	Time	Action	Response
3	13:53:17	SELECT bk.Genre, COUNT(*) as BorrowCount -- This will sele...	9 row(s) returned

```
823 -- I can now perform a join with just clients borrowed in the specific year
824 -- I can join with the Clients table to get the occupations and sort the list accordingly
825
826 • CREATE TEMPORARY TABLE Temp2016Borrowers AS
827     SELECT ClientID
828     FROM Borrowers
829     WHERE YEAR(BorrowDate) = 2016;
830     -- Creating the temporary table for 2016 ONLY
831     -- I will give this table an Alias as well
832
833 • SELECT
834     c.Occupation,
835     COUNT(*) as BorrowCount
836     -- Getting the occupations from the Client table and counting borrowings
837     FROM
838         Temp2016Borrowers tb
839         -- Alias for temp table
840     JOIN
841         Clients c ON tb.ClientID = c.ClientID
842         -- Joining on ClientID
843     GROUP BY
844         c.Occupation
845         -- Tally the occupation
846     ORDER BY
847         BorrowCount DESC
848         -- Give me the highest to lowest order
849     LIMIT 5;
850     -- This will give me the top 5 as the question requests
851     -- DROP TEMPORARY TABLE IF EXISTS Temp2016Borrowers; will be used when I am done
852
```

100% □ 1:866

Result Grid Filter Rows: Search Export: Fetch rows:

Occupation	BorrowCount
Student	32
Bus Driver	8
Dentist	6
Computer Programmer	6
Police Officer	5

Result 12 Read Only

Action Output

Query 2 CS 204 w/o auto-increment - Schema SQL File 3\*

888 -- Getting the average borrow count per Occupation  
889 FROM TotalBorrowings tb  
890 -- Using the temporary table that will give the total borrowings  
891  
892 JOIN Clients c ON tb.ClientID = c.ClientID  
893 -- Joining the Clients table using the ClientID to get the Occupation  
894  
895 GROUP BY c.Occupation  
896 -- Grouping by Occupation and getting the average for each  
897  
898 ORDER BY AverageBorrowCount DESC;  
899  
900 -- Although this was not in the question, I wanted to see who borrowed the most

100% 67:903

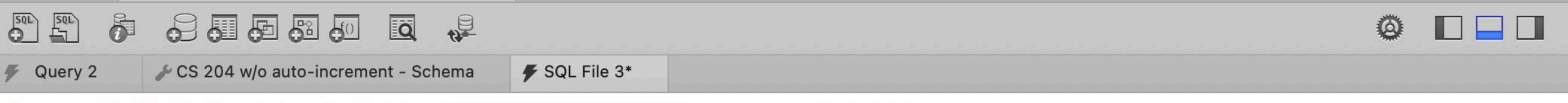
**Result Grid** Filter Rows: Search Export:

Occupation	AverageBorrowCou...
Nurse	7.0000
Computer Security Manager	6.0000
Dentist	5.6667
Computer Programmer	5.6667
Cashier	5.0000
Oil Exploration Engineer	5.0000
Manufacturing Director	5.0000
Food Scientist	5.0000
Police Officer	4.5000
Student	4.4211
Violinist	4.0000
Insurance Agent	4.0000
Bus Driver	4.0000
Ship Engineer	4.0000
Doctor	4.0000
HR Clerk	4.0000
Systems Analyst	4.0000
School Teacher	3.6000
Professor	3.5000
Firefighter	3.2500
Repair Worker	3.0000
Payroll Clerk	3.0000
Computer Engineer	3.0000
Manager	3.0000
Licensed Massage Therapist	2.0000
Parquet Floor Layer	2.0000
Aircraft Electrician	2.0000

Result 14 Read Only

Action Output

	Time	Action	Response
1	15:01:39	CREATE TEMPORARY TABLE TotalBorrowings AS SELECT ClientID,...	75 row(s) affected Records: 75 Duplicates: 0 Warnings: 0
2	15:01:56	SELECT -- Calculating the average number of books per Occupati...	32 row(s) returned
3	15:05:45	SELECT -- Calculating the average number of books per Occupati...	32 row(s) returned



```
920 -- For this question I will go back to the subquery approach as the code is simpler
921 -- The subquery will be used to calculate the % of clients who borrowed each book
922 -- I will name the view BooksBorrowedBy20Percent
923
924 • CREATE VIEW BooksBorrowedBy20Percent AS
925     SELECT
926         bk.BookID,
927         bk.BookTitle,
928         -- Selecting the BookID and BookTitle from the Books table
929         (COUNT(DISTINCT b.ClientID) / (SELECT COUNT(*) FROM Clients) * 100) as BorrowPercentage
930         -- Equation used and Distinct to make sure the Client is unique per book borrowed
931     FROM
932         Borrowers b
933         -- 'b' Alias used beginning with the Borrowers table
934     JOIN
935         Books bk ON b.BookID = bk.BookID
936         -- Joining the Books table with the BookID to get the Books
937     GROUP BY
938         bk.BookID,
939         bk.BookTitle
940         -- Grouping the BookID and Title to tally the borrowing
941     HAVING
942         BorrowPercentage >= 20;
943         -- This will only show Books borrowed by at least 20%
944
945 • SELECT *
946     FROM BooksBorrowedBy20Percent;
947     -- Retrieve data from the VIEW
948
949
```

100% 31:947

Result Grid Filter Rows: Search Export:

BookID	BookTitle	BorrowPercentage
13	Electrical transformers	21.2500

BooksBorrowedBy20Percent 16

Result Grid

Read Only

Action Output

Time	Action	Response
1 15:41:21	SELECT * FROM BooksBorrowedBy20Percent LIMIT 0, 1000	1 row(s) returned

Local Instance - Warning - not supported

Query 2 CS 204 w/o auto-increment - Schema SQL File 3\*

Limit to 1000 rows

```

959
960
961
962
963
964 -- Objective C Question 11:
965 -- The top month of borrows in 2017
966 -- I will use a simple query where I will select the month from the Borrows table
967 -- I will then filter the year to 2017 group by the month
968 -- I will order in DESC and Limit to 1
969
970 • SELECT
971     MONTH(b.BorrowDate) AS BorrowMonth,
972     COUNT(*) AS TotalBorrows
973     -- This will get the month from the Borrows Table and count the total borrows per month
974     FROM
975         Borrowers b
976         -- Using the same Alias as previous queries
977     WHERE
978         YEAR(b.BorrowDate) = 2017
979         -- This will filter the results for the specific year
980     GROUP BY
981         BorrowMonth
982         -- Refining the query to group it by month
983     ORDER BY
984         TotalBorrows DESC
985         -- This will give the top borrows per month first
986     LIMIT 1;
987         -- This will display the month in 2017 with the most borrows

```

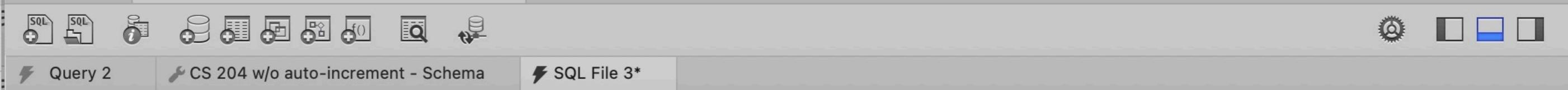
Result Grid Filter Rows: Search Export: Fetch rows:

BorrowMonth	TotalBorrows
8	10

Result 17 Read Only

Action Output

	Time	Action	Response
1	16:02:54	SELECT MONTH(b.BorrowDate) AS BorrowMonth, COUNT(*) AS...	1 row(s) returned



Query 2 CS 204 w/o auto-increment - Schema SQL File 3\*

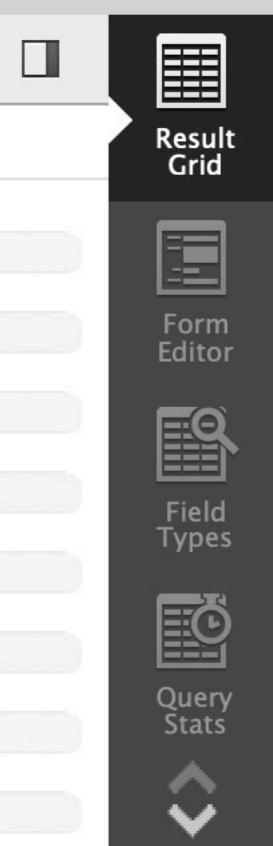
```
1017    FROM
1018        -- Subquery to get the approximate age and count borrows per client
1019        (SELECT
1020            c.ClientID,
1021            YEAR(CURRENT_DATE) - c.ClientDoB as ApproximateAge,
1022            COUNT(b.BookID) as BorrowCount
1023        FROM
1024            Clients c
1025            -- Still using the Alias from previous queries
1026        JOIN
1027            Borrowers b ON c.ClientID = b.ClientID
1028            -- Joining the Borrowers table with the Clients table using Client ID
1029        GROUP BY
1030            c.ClientID, ApproximateAge) as AgeBorrowData
1031            -- Grouping by Client ID and Approximate Age
1032        GROUP BY
1033            ApproximateAge
1034        ORDER BY
1035            ApproximateAge;
1036            -- Ordering by Approximate Age
1037
1038
```

100% 9:1032

Result Grid Filter Rows: Search Export:

ApproximateAge	AverageBorrows
14	2.3333
16	6.0000
17	5.0000
18	5.5000
19	4.5000
20	3.0000
21	5.0000
22	2.0000
23	4.5000
25	3.6667
26	2.0000
28	2.0000
29	4.5000
30	10.0000
32	3.0000
34	5.5000
37	2.0000

Result 18



Read Only

Action Output

Time	Action	Response
------	--------	----------

1 16:32:08 SELECT ApproximateAge, AVG(BorrowCount) as AverageBorro... 38 row(s) returned



Query 2 CS 204 w/o auto-increment - Schema SQL File 3\*

```
1055 -- The oldest and youngest clients of the library
1056 -- For this one I almost forgot to use the UNION operator and was going to simply do two queries
1057 -- This query is a little more tricky since the ClientsDoB only indicates the year
1058 -- Knowing that multiple Clients will have the same year I will use the WHERE and Max/Min statements
1059 -- The MAX will give me the clients with the youngest birth year as that is the highest value
1060 -- The MIN will give me the clients with the oldest birth year as that is the lowest values
1061
1062 • SELECT
1063     ClientID,
1064     ClientFirstName,
1065     ClientLastName,
1066     ClientDoB
1067     FROM
1068     Clients
1069     WHERE
1070     ClientDoB = (SELECT MAX(ClientDoB)
1071         FROM
1072         Clients)
1073     UNION -- Used to combine both queries
1074     SELECT
1075         ClientID,
1076         ClientFirstName,
1077         ClientLastName,
1078         ClientDoB
1079         FROM
1080         Clients
1081     WHERE
1082     ClientDoB = (SELECT MIN(ClientDoB)
1083         FROM
1084         Clients);
1085
```

100% 6:1079

Result Grid Filter Rows: Search Export:

ClientID	ClientFirstName	ClientLastName	ClientDoB
----------	-----------------	----------------	-----------

2	Alina	Morton	2010
22	Alina	Austin	2010
65	Joan	Grant	2010
12	Mya	Austin	1960
38	Caroline	Clark	1960
72	Annabel	Williamson	1960



Result Grid



Form Editor

Query 2 | CS 204 w/o auto-increment - Schema | SQL File 3\*

1097  
1098  
1099 -- Objective C Question 14:  
1100 -- First and Last names of authors that wrote books in more than one genre  
1101 -- For this query I will join the Authors and Books table using the AuthorID and BookAuthor column  
1102 -- I will continue using the Alias 'a' for Authors and 'bk' for Books  
1103 -- I will use Distinct in the Genre column to get different values per Author  
1104 -- This will show that the Author has written books in multiple Genres  
1105 -- I will then show the values of being greater than 1

1106 • **SELECT**

1107     a.AuthorID,  
1108     a.AuthorFirstName,  
1109     a.AuthorLastName,  
1110     COUNT(DISTINCT bk.Genre) AS GenreCount  
1111       -- This will count distinct genres for the Author

1112 **FROM**  
1113     Authors a  
1114       -- Starting with the Authors table with the Alias mentioned above

1115 **JOIN**  
1116     Books bk  
1117       -- Alias for the Books table

1118     **ON** a.AuthorID = bk.BookAuthor  
1119       -- Joining the Books and Authors table with the AuthorID and BookAuthor

1120 **GROUP BY**

1121     a.AuthorID,  
1122     a.AuthorFirstName,  
1123     a.AuthorLastName  
1124       -- Grouping it by the Author details to get the distinct count

1125 **HAVING**

1126     COUNT(DISTINCT bk.Genre) > 1;  
1127       -- This will filter out Authors and only give me Authors who wrote in more than 1 Genre

1128  
1129

100% 13:1121

**Result Grid** Filter Rows: Search Export:

AuthorID	AuthorFirstName	AuthorLastName	GenreCount
3	Elena	Martin	2

Result 23 Read Only

Query Completed