Computer Science 204: Database Programming

A.

```
CREATE SCHEMA `library` ;

CREATE TABLE Client (
    ClientID INT PRIMARY KEY, -- Define ClientID column as an integer primary key
    ClientFirstName VARCHAR(50), -- Define ClientFirstName column as a variable-length string with a maximum length of 50 characters
    ClientLastName VARCHAR(50), -- Define ClientLastName column as a variable-length string with a maximum length of 50 characters
    ClientDOB DATE, -- Define ClientDOB column as a date type
    Occupation VARCHAR(50) -- Define Occupation column as a variable-length string with a maximum length of 50 characters
);


CREATE TABLE Author (
    AuthorID INT PRIMARY KEY, -- Define AuthorID column as an integer primary key
    AuthorFirstName VARCHAR(50), -- Define AuthorFirstName column as a variable-length string with a maximum length of 50 characters
    AuthorLastName VARCHAR(50), -- Define AuthorLastName column as a variable-length string with a maximum length of 50 characters
    AuthorNationality VARCHAR(50) -- Define AuthorNationality column as a variable-length string with a maximum length of 50 characters
);


CREATE TABLE Book (
    BookID INT PRIMARY KEY, -- Define BookID column as an integer primary key
    BookTitle VARCHAR(100), -- Define BookTitle column as a variable-length string with a maximum length of 100 characters
    AuthorID INT, -- Define AuthorID column as an integer
    Genre VARCHAR(50), -- Define Genre column as a variable-length string with a maximum length of 50 characters
    FOREIGN KEY (AuthorID) REFERENCES Author(AuthorID) -- Define a foreign key constraint on AuthorID referencing the Author table's AuthorID column
);
```

```
CREATE TABLE Borrower (
    BorrowID INT PRIMARY KEY, -- Define BorrowID column as an integer primary key
    ClientID INT, -- Define ClientID column as an integer
    BookID INT, -- Define BookID column as an integer
    BorrowDate DATE, -- Define BorrowDate column as a date type
    FOREIGN KEY (ClientID) REFERENCES Client(ClientID), -- Define a foreign key constraint on ClientID referencing the Client table's ClientID column
    FOREIGN KEY (BookID) REFERENCES Book(BookID) -- Define a foreign key constraint on BookID referencing the Book table's BookID column
);
```

B.

```
INSERT INTO Author (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');
```

```
INSERT INTO Book (BookID, BookTitle, AuthorID, 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, 'Fiction');

INSERT INTO Client (ClientID, ClientFirstName, ClientLastName, ClientDOB, Occupation)
VALUES
    (1, 'Kaiden', 'Hill', '2006-01-01', 'Student'),
    (2, 'Alina', 'Morton', '2010-01-01', 'Student'),
    (3, 'Fania', 'Brooks', '1983-01-01', 'Food Scientist'),
    (4, 'Courtney', 'Jensen', '2006-01-01', 'Student'),
    (5, 'Brittany', 'Hill', '1983-01-01', 'Firefighter'),
    (6, 'Max', 'Rogers', '2005-01-01', 'Student'),
    (7, 'Margaret', 'McCarthy', '1981-01-01', 'School Psychologist'),
    (8, 'Julie', 'McCarthy', '1973-01-01', 'Professor'),
    (9, 'Ken', 'McCarthy', '1974-01-01', 'Securities Clerk'),
    (10, 'Britany', 'O''Quinn', '1984-01-01', 'Violinist'),
    (11, 'Conner', 'Gardner', '1998-01-01', 'Licensed Massage Therapist'),
    (12, 'Mya', 'Austin', '1960-01-01', 'Parquet Floor Layer'),
    (13, 'Thierry', 'Rogers', '2004-01-01', 'Student'),
    (14, 'Eloise', 'Rogers', '1984-01-01', 'Computer Security Manager'),
    (15, 'Gerard', 'Jackson', '1979-01-01', 'Oil Exploration Engineer'),
    (16, 'Randy', 'Day', '1986-01-01', 'Aircraft Electrician'),
    (17, 'Jodie', 'Page', '1990-01-01', 'Manufacturing Director'),
    (18, 'Coral', 'Rice', '1996-01-01', 'Window Washer'),
    (19, 'Ayman', 'Austin', '2002-01-01', 'Student'),
    (20, 'Jaxson', 'Austin', '1999-01-01', 'Repair Worker'),
    (21, 'Joel', 'Austin', '1973-01-01', 'Police Officer'),
    (22, 'Alina', 'Austin', '2010-01-01', 'Student'),
    (23, 'Elin', 'Austin', '1962-01-01', 'Payroll Clerk'),
    (24, 'Ophelia', 'Wolf', '2004-01-01', 'Student'),
    (25, 'Eliot', 'McGuire', '1967-01-01', 'Dentist'),
    (26, 'Peter', 'McKinney', '1968-01-01', 'Professor'),
    (27, 'Annabella', 'Henry', '1974-01-01', 'Nurse'),
    (28, 'Anastasia', 'Baker', '2001-01-01', 'Student'),
    (29, 'Tyler', 'Baker', '1984-01-01', 'Police Officer'),
    (30, 'Lilian', 'Ross', '1983-01-01', 'Insurance Agent'),
    (31, 'Thierry', 'Arnold', '1975-01-01', 'Bus Driver'),
    (32, 'Angelina', 'Rowe', '1979-01-01', 'Firefighter'),
    (33, 'Marcia', 'Rowe', '1974-01-01', 'Health Educator'),
    (34, 'Martin', 'Rowe', '1976-01-01', 'Ship Engineer'),
    (35, 'Adeline', 'Rowe', '2005-01-01', 'Student'),
    (36, 'Colette', 'Rowe', '1963-01-01', 'Professor'),
    (37, 'Diane', 'Clark', '1975-01-01', 'Payroll Clerk'),
    (38, 'Caroline', 'Clark', '1960-01-01', 'Dentist'),
    (39, 'Dalton', 'Clayton', '1982-01-01', 'Police Officer'),
    (40, 'Steve', 'Clayton', '1990-01-01', 'Bus Driver'),
    (41, 'Melanie', 'Clayton', '1987-01-01', 'Computer Engineer'),
    (42, 'Alana', 'Wilson', '2007-01-01', 'Student'),
    (43, 'Carson', 'Byrne', '1995-01-01', 'Food Scientist'),
    (44, 'Conrad', 'Byrne', '2007-01-01', 'Student'),
    (45, 'Ryan', 'Porter', '2008-01-01', 'Student'),
    (46, 'Elin', 'Porter', '1978-01-01', 'Computer Programmer'),
    (47, 'Tyler', 'Harvey', '2007-01-01', 'Student'),
    (48, 'Arya', 'Harvey', '2008-01-01', 'Student'),
    (49, 'Serena', 'Harvey', '1978-01-01', 'School Teacher'),
    (50, 'Lilly', 'Franklin', '1976-01-01', 'Doctor'),
    (51, 'Mai', 'Franklin', '1994-01-01', 'Dentist'),
        (52, 'John', 'Franklin', '1999-01-01', 'Firefighter'),
        (53, 'Judy', 'Franklin', '1995-01-01', 'Firefighter'),
        (54, 'Katy', 'Lloyd', '1992-01-01', 'School Teacher'),
        (55, 'Tamara', 'Allen', '1963-01-01', 'Ship Engineer'),
        (56, 'Maxim', 'Lyons', '1985-01-01', 'Police Officer'),
        (57, 'Allan', 'Lyons', '1983-01-01', 'Computer Engineer'),
        (58, 'Marc', 'Harris', '1980-01-01', 'School Teacher'),
        (59, 'Elin', 'Young', '2009-01-01', 'Student'),
        (60, 'Diana', 'Young', '2008-01-01', 'Student'),
        (61, 'Diane', 'Young', '2006-01-01', 'Student'),
        (62, 'Alana', 'Bird', '2003-01-01', 'Student'),
        (63, 'Anna', 'Becker', '1979-01-01', 'Security Agent'),
        (64, 'Katie', 'Grant', '1977-01-01', 'Manager'),
        (65, 'Joan', 'Grant', '2010-01-01', 'Student'),
        (66, 'Bryan', 'Bell', '2001-01-01', 'Student'),
        (67, 'Belle', 'Miller', '1970-01-01', 'Professor'),
        (68, 'Peggy', 'Stevens', '1990-01-01', 'Bus Driver'),
        (69, 'Steve', 'Williamson', '1975-01-01', 'HR Clerk'),
        (70, 'Tyler', 'Williamson', '1999-01-01', 'Doctor'),
        (71, 'Izabelle', 'Williamson', '1990-01-01', 'Systems Analyst'),
        (72, 'Annabel', 'Williamson', '1960-01-01', 'Cashier'),
        (73, 'Mohamed', 'Waters', '1966-01-01', 'Insurance Agent'),
        (74, 'Marion', 'Newman', '1970-01-01', 'Computer Programmer'),
        (75, 'Ada', 'Williams', '1986-01-01', 'Computer Programmer'),
        (76, 'Sean', 'Scott', '1983-01-01', 'Bus Driver'),
        (77, 'Farrah', 'Scott', '1974-01-01', 'Ship Engineer'),
        (78, 'Christine', 'Lambert', '1973-01-01', 'School Teacher'),
        (79, 'Alysha', 'Lambert', '2007-01-01', 'Student'),
        (80, 'Maia', 'Grant', '1984-01-01', 'School Teacher');


    INSERT INTO Borrower (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');
```

C.

1. Display all contents of the Clients table

```
SELECT * FROM Client;
```

Result:

```
ClientID    ClientFirstName         ClientLastName          ClientDOB              Occupation
1           Kaiden      Hill        2006-01-01              Student
2           Alina       Morton      2010-01-01              Student
3           Fania       Brooks      1983-01-01              Food Scientist
4           Courtney                Jensen      2006-01-01              Student
5           Brittany    Hill        1983-01-01              Firefighter
6           Max         Rogers      2005-01-01              Student
7           Margaret                McCarthy                1981-01-01              School Psychologist
8           Julie       McCarthy                1973-01-01              Professor
9           Ken         McCarthy                1974-01-01              Securities Clerk
10          Britany     O'Quinn                 1984-01-01              Violinist
11          Conner      Gardner                 1998-01-01              Licensed Massage Therapist
12          Mya         Austin      1960-01-01              Parquet Floor Layer
13          Thierry     Rogers      2004-01-01              Student
14          Eloise      Rogers      1984-01-01              Computer Security Manager
15          Gerard      Jackson                 1979-01-01              Oil Exploration Engineer
16          Randy       Day         1986-01-01              Aircraft Electrician
17          Jodie       Page        1990-01-01              Manufacturing Director
18          Coral       Rice        1996-01-01              Window Washer
19          Ayman       Austin      2002-01-01              Student
20          Jaxson      Austin      1999-01-01              Repair Worker
21          Joel        Austin      1973-01-01              Police Officer
22          Alina       Austin      2010-01-01              Student
23          Elin        Austin      1962-01-01              Payroll Clerk
24          Ophelia     Wolf        2004-01-01              Student
25          Eliot       McGuire                 1967-01-01              Dentist
26          Peter       McKinney                1968-01-01              Professor
27          Annabella               Henry       1974-01-01              Nurse
28          Anastasia               Baker       2001-01-01              Student
29          Tyler       Baker       1984-01-01              Police Officer
30          Lilian      Ross        1983-01-01              Insurance Agent
31          Thierry     Arnold      1975-01-01              Bus Driver
32          Angelina                Rowe        1979-01-01              Firefighter
33          Marcia      Rowe        1974-01-01              Health Educator
34          Martin      Rowe        1976-01-01              Ship Engineer
35          Adeline     Rowe        2005-01-01              Student
36          Colette     Rowe        1963-01-01              Professor
37          Diane       Clark       1975-01-01              Payroll Clerk
38          Caroline                Clark       1960-01-01              Dentist
39          Dalton      Clayton     1982-01-01              Police Officer
40          Steve       Clayton     1990-01-01              Bus Driver
```

41	*Melanie*	*Clayton*	*1987-01-01*	*Computer Engineer*
42	*Alana*	*Wilson*	*2007-01-01*	*Student*
43	*Carson*	*Byrne*	*1995-01-01*	*Food Scientist*
44	*Conrad*	*Byrne*	*2007-01-01*	*Student*
45	*Ryan*	*Porter*	*2008-01-01*	*Student*
46	*Elin*	*Porter*	*1978-01-01*	*Computer Programmer*
47	*Tyler*	*Harvey*	*2007-01-01*	*Student*
48	*Arya*	*Harvey*	*2008-01-01*	*Student*
49	*Serena*	*Harvey*	*1978-01-01*	*School Teacher*
50	*Lilly*	*Franklin*	*1976-01-01*	*Doctor*
51	*Mai*	*Franklin*	*1994-01-01*	*Dentist*
52	*John*	*Franklin*	*1999-01-01*	*Firefighter*
53	*Judy*	*Franklin*	*1995-01-01*	*Firefighter*
54	*Katy*	*Lloyd*	*1992-01-01*	*School Teacher*
55	*Tamara*	*Allen*	*1963-01-01*	*Ship Engineer*
56	*Maxim*	*Lyons*	*1985-01-01*	*Police Officer*
57	*Allan*	*Lyons*	*1983-01-01*	*Computer Engineer*
58	*Marc*	*Harris*	*1980-01-01*	*School Teacher*
59	*Elin*	*Young*	*2009-01-01*	*Student*
60	*Diana*	*Young*	*2008-01-01*	*Student*
61	*Diane*	*Young*	*2006-01-01*	*Student*
62	*Alana*	*Bird*	*2003-01-01*	*Student*
63	*Anna*	*Becker*	*1979-01-01*	*Security Agent*
64	*Katie*	*Grant*	*1977-01-01*	*Manager*
65	*Joan*	*Grant*	*2010-01-01*	*Student*
66	*Bryan*	*Bell*	*2001-01-01*	*Student*
67	*Belle*	*Miller*	*1970-01-01*	*Professor*
68	*Peggy*	*Stevens*	*1990-01-01*	*Bus Driver*
69	*Steve*	*Williamson*	*1975-01-01*	*HR Clerk*
70	*Tyler*	*Williamson*	*1999-01-01*	*Doctor*
71	*Izabelle*	*Williamson*	*1990-01-01*	*Systems Analyst*
72	*Annabel*	*Williamson*	*1960-01-01*	*Cashier*
73	*Mohamed*	*Waters*	*1966-01-01*	*Insurance Agent*
74	*Marion*	*Newman*	*1970-01-01*	*Computer Programmer*
75	*Ada*	*Williams*	*1986-01-01*	*Computer Programmer*
76	*Sean*	*Scott*	*1983-01-01*	*Bus Driver*
77	*Farrah*	*Scott*	*1974-01-01*	*Ship Engineer*
78	*Christine*	*Lambert*	*1973-01-01*	*School Teacher*
79	*Alysha*	*Lambert*	*2007-01-01*	*Student*
80	*Maia*	*Grant*	*1984-01-01*	*School Teacher*

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

```
SELECT
  ClientFirstName, -- Select the first name of the client
  ClientLastName,  -- Select the last name of the client
  TIMESTAMPDIFF(YEAR, ClientDOB, CURDATE()) AS Age, -- Calculate the age of the client based on their date of birth
  Occupation       -- Select the occupation of the client
FROM
  Client;          -- Specify the table name "Client"
```

Result:

```
ClientFirstName,ClientLastName,Age,Occupation
Kaiden,Hill,17,Student
Alina,Morton,13,Student
Fania,Brooks,40,"Food Scientist"
Courtney,Jensen,17,Student
Brittany,Hill,40,Firefighter
Max,Rogers,18,Student
Margaret,McCarthy,42,"School Psychologist"
Julie,McCarthy,50,Professor
Ken,McCarthy,49,"Securities Clerk"
Britany,O'Quinn,39,Violinist
Conner,Gardner,25,"Licensed Massage Therapist"
Mya,Austin,63,"Parquet Floor Layer"
Thierry,Rogers,19,Student
Eloise,Rogers,39,"Computer Security Manager"
Gerard,Jackson,44,"Oil Exploration Engineer"
Randy,Day,37,"Aircraft Electrician"
Jodie,Page,33,"Manufacturing Director"
Coral,Rice,27,"Window Washer"
Ayman,Austin,21,Student
Jaxson,Austin,24,"Repair Worker"
Joel,Austin,50,"Police Officer"
Alina,Austin,13,Student
Elin,Austin,61,"Payroll Clerk"
Ophelia,Wolf,19,Student
Eliot,McGuire,56,Dentist
Peter,McKinney,55,Professor
Annabella,Henry,49,Nurse
Anastasia,Baker,22,Student
Tyler,Baker,39,"Police Officer"
Lilian,Ross,40,"Insurance Agent"
Thierry,Arnold,48,"Bus Driver"
Angelina,Rowe,44,Firefighter
Marcia,Rowe,49,"Health Educator"
```

```
Martin,Rowe,47,"Ship Engineer"
Adeline,Rowe,18,Student
Colette,Rowe,60,Professor
Diane,Clark,48,"Payroll Clerk"
Caroline,Clark,63,Dentist
Dalton,Clayton,41,"Police Officer"
Steve,Clayton,33,"Bus Driver"
Melanie,Clayton,36,"Computer Engineer"
Alana,Wilson,16,Student
Carson,Byrne,28,"Food Scientist"
Conrad,Byrne,16,Student
Ryan,Porter,15,Student
Elin,Porter,45,"Computer Programmer"
Tyler,Harvey,16,Student
Arya,Harvey,15,Student
Serena,Harvey,45,"School Teacher"
Lilly,Franklin,47,Doctor
Mai,Franklin,29,Dentist
John,Franklin,24,Firefighter
Judy,Franklin,28,Firefighter
Katy,Lloyd,31,"School Teacher"
Tamara,Allen,60,"Ship Engineer"
Maxim,Lyons,38,"Police Officer"
Allan,Lyons,40,"Computer Engineer"
Marc,Harris,43,"School Teacher"
Elin,Young,14,Student
Diana,Young,15,Student
Diane,Young,17,Student
Alana,Bird,20,Student
Anna,Becker,44,"Security Agent"
Katie,Grant,46,Manager
Joan,Grant,13,Student
Bryan,Bell,22,Student
Belle,Miller,53,Professor
Peggy,Stevens,33,"Bus Driver"
Steve,Williamson,48,"HR Clerk"
Tyler,Williamson,24,Doctor
Izabelle,Williamson,33,"Systems Analyst"
Annabel,Williamson,63,Cashier
Mohamed,Waters,57,"Insurance Agent"
Marion,Newman,53,"Computer Programmer"
Ada,Williams,37,"Computer Programmer"
Sean,Scott,40,"Bus Driver"
Farrah,Scott,49,"Ship Engineer"
Christine,Lambert,50,"School Teacher"
Alysha,Lambert,16,Student
Maia,Grant,39,"School Teacher"
```

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

```
SELECT
  Client.ClientFirstName, -- Select the first name of the client
  Client.ClientLastName   -- Select the last name of the client
FROM
  Client                 -- Specify the table name "Client"
JOIN
  Borrower ON Client.ClientID = Borrower.ClientID -- Join the "Client" and "Borrower" tables
based on the client ID
JOIN
  Book ON Borrower.BookID = Book.BookID -- Join the "Borrower" and "Book" tables based on
the book ID
WHERE
  MONTH(Borrower.BorrowDate) = 3       -- Filter the results to include only borrowings in March
  AND YEAR(Borrower.BorrowDate) = 2018; -- Filter the results to include only borrowings in the
year 2018
```

Result:

```
ClientFirstName,ClientLastName
Katy,Lloyd
Marcia,Rowe
Angelina,Rowe
Carson,Byrne
Maia,Grant
Tyler,Baker
Gerard,Jackson
Alysha,Lambert
```

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

```
SELECT
  Author.AuthorFirstName, -- Select the first name of the author
  Author.AuthorLastName   -- Select the last name of the author
FROM
  Author                  -- Specify the table name "Author"
JOIN
  Book ON Author.AuthorID = Book.AuthorID -- Join the "Author" and "Book" tables based on the author ID
JOIN
  Borrower ON Book.BookID = Borrower.BookID -- Join the "Book" and "Borrower" tables based on the book ID
JOIN
  Client ON Borrower.ClientID = Client.ClientID -- Join the "Borrower" and "Client" tables based on the client ID
WHERE
  YEAR(Borrower.BorrowDate) = 2017 -- Filter the results to include only borrowings in the year 2017
GROUP BY
  Author.AuthorID -- Group the results by the author ID
ORDER BY
  COUNT(Borrower.BookID) DESC -- Sort the results in descending order based on the count of book borrowings
LIMIT 5; -- Limit the result set to the top 5 rows
```

Result:

```
AuthorFirstName,AuthorLastName
Sofia,Smith
Elena,Martin
Logan,Moore
Maria,Brown
Zoe,Roy
```

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

```
SELECT
  Author.AuthorNationality -- Select the nationality of the author
FROM
  Author                -- Specify the table name "Author"
JOIN
  Book ON Author.AuthorID = Book.AuthorID -- Join the "Author" and "Book" tables based on the author ID
JOIN
  Borrower ON Book.BookID = Borrower.BookID -- Join the "Book" and "Borrower" tables based on the book ID
WHERE
  YEAR(Borrower.BorrowDate) BETWEEN 2015 AND 2017 -- Filter the results to include borrowings between the years 2015 and 2017
GROUP BY
  Author.AuthorNationality -- Group the results by the author's nationality
ORDER BY
  COUNT(Borrower.BookID) ASC -- Sort the results in ascending order based on the count of book borrowings
LIMIT 5; -- Limit the result set to the bottom 5 rows
```

Result:

```
AuthorNationality
Spain
"Great Britain"
China
Brazil
France
```

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

```
SELECT
  Book.BookTitle -- Select the title of the book
FROM
  Book -- Specify the table name "Book"
JOIN
  Borrower ON Book.BookID = Borrower.BookID -- Join the "Book" and "Borrower" tables based on the book ID
WHERE
  YEAR(Borrower.BorrowDate) BETWEEN 2015 AND 2017 -- Filter the results to include borrowings between the years 2015 and 2017
GROUP BY
  Book.BookID -- Group the results by the book ID
ORDER BY
  COUNT(Borrower.BookID) DESC -- Sort the results in descending order based on the count of book borrowings
LIMIT 1; -- Limit the result set to 1 row, i.e., the most borrowed book
```

Result:

The perfect match

7. Top borrowed genres for clients born in years 1970-1980

```
SELECT
  Book.Genre, -- Select the genre of the book
  COUNT(*) AS BorrowCount -- Count the number of borrowings for each genre
FROM
  Book -- Specify the table name "Book"
JOIN
  Borrower ON Book.BookID = Borrower.BookID -- Join the "Book" and "Borrower" tables based on the book ID
JOIN
  Client ON Borrower.ClientID = Client.ClientID -- Join the "Borrower" and "Client" tables based on the client ID
WHERE
  YEAR(Client.ClientDOB) BETWEEN 1970 AND 1980 -- Filter the results to include clients born in the years 1970-1980
GROUP BY
  Book.Genre -- Group the results by the genre of the book
ORDER BY
  BorrowCount DESC; -- Sort the results in descending order based on the count of borrowings
```

Result:

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

8. Top 5 occupations that borrowed the most in 2016

```
SELECT
  Client.Occupation, -- Select the occupation of the client
  COUNT(*) AS BorrowCount -- Count the number of borrowings for each occupation
FROM
  Client -- Specify the table name "Client"
JOIN
  Borrower ON Client.ClientID = Borrower.ClientID -- Join the "Client" and "Borrower" tables based on the client ID
JOIN
  Book ON Borrower.BookID = Book.BookID -- Join the "Borrower" and "Book" tables based on the book ID
WHERE
  YEAR(Borrower.BorrowDate) = 2016 -- Filter the borrowings to those that occurred in 2016
GROUP BY
  Client.Occupation -- Group the results by the occupation of the client
ORDER BY
  BorrowCount DESC -- Sort the results in descending order based on the borrow count
LIMIT 5; -- Limit the output to the top 5 occupations with the highest borrow count
```

Result:

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

9. Average number of borrowed books by job title

```
SELECT
  Client.Occupation, -- Select the occupation of the client
  AVG(BookCount) AS AverageBooksBorrowed -- Calculate the average number of books borrowed and assign it an alias
FROM
  (
    -- Subquery to calculate the number of books borrowed per client and per book
    SELECT
      ClientID, -- Select the client ID
      COUNT(*) AS BookCount -- Count the number of books borrowed by each client and each book
    FROM
      Borrower -- Specify the table name "Borrower"
    GROUP BY
      ClientID, BookID -- Group the borrowings by client ID and book ID
  ) AS BorrowCounts -- Assign the subquery an alias
JOIN
  Client ON BorrowCounts.ClientID = Client.ClientID -- Join the subquery and the "Client" table based on the client ID
GROUP BY
  Client.Occupation; -- Group the results by the occupation of the client
```

Result:

```
Occupation,AverageBooksBorrowed
Student,1.0120
Firefighter,1.0000
"Health Educator",1.0000
"Bus Driver",1.0667
Manager,1.0000
"Police Officer",1.0000
"Payroll Clerk",1.0000
"Computer Programmer",1.1333
"School Teacher",1.0000
Dentist,1.0625
"Food Scientist",1.2500
"Insurance Agent",1.1429
"Systems Analyst",1.0000
"Computer Engineer",1.2000
"Repair Worker",1.0000
Cashier,1.2500
"Parquet Floor Layer",1.0000
"Ship Engineer",1.0000
Nurse,1.0000
```

```
"HR Clerk",1.0000
"Manufacturing Director",1.2500
Professor,1.0769
"Security Agent",1.0000
"Oil Exploration Engineer",1.0000
"Licensed Massage Therapist",1.0000
"Securities Clerk",1.0000
Violinist,1.0000
Doctor,1.1429
"School Psychologist",1.0000
"Computer Security Manager",1.2000
"Aircraft Electrician",1.0000
"Window Washer",1.0000
```

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

```
CREATE VIEW PopularTitles AS
SELECT
  Book.BookTitle, -- Select the book title
  COUNT(DISTINCT Borrower.ClientID) AS NumClients -- Count the number of distinct clients who borrowed each book
FROM
  Book -- Specify the table name "Book"
JOIN
  Borrower ON Book.BookID = Borrower.BookID -- Join the "Book" and "Borrower" tables based on the book ID
GROUP BY
  Book.BookID -- Group the borrowings by book ID
HAVING
  COUNT(DISTINCT Borrower.ClientID) >= 0.2 * (SELECT COUNT(DISTINCT ClientID) FROM Borrower); -- Filter out titles that were borrowed by less than 20% of clients


SELECT * FROM PopularTitles;
```

Result:

```
BookTitle,NumClients
"Electrical transformers",17
```

11. The top month of borrows in 2017

```
SELECT
  BorrowMonth,  BorrowCount
FROM
  (
    SELECT
      EXTRACT(MONTH FROM BorrowDate) AS BorrowMonth, -- Extract the month from the
BorrowDate column and alias it as BorrowMonth
      COUNT(*) AS BorrowCount -- Count the number of borrowings in each month
    FROM
      Borrower -- Specify the table name "Borrower"
    WHERE
      YEAR(BorrowDate) = 2017 -- Filter the borrowings for the year 2017
    GROUP BY
      BorrowMonth -- Group the borrowings by month
  ) AS SubQuery -- Use a subquery to calculate the borrow count for each month in the year
2017
JOIN
  (
    SELECT
      MAX(BorrowCount) AS MaxBorrowCount -- Find the maximum borrow count in the year
2017
    FROM
      (
        SELECT
          EXTRACT(MONTH FROM BorrowDate) AS BorrowMonth, -- Extract the month from the
BorrowDate column and alias it as BorrowMonth
          COUNT(*) AS BorrowCount -- Count the number of borrowings in each month
        FROM
          Borrower -- Specify the table name "Borrower"
        WHERE
          YEAR(BorrowDate) = 2017 -- Filter the borrowings for the year 2017
        GROUP BY
          BorrowMonth -- Group the borrowings by month
      ) AS SubQuery2 -- Use another subquery to calculate the borrow count for each month in
the year 2017
  ) AS MaxQuery -- Use a subquery to find the maximum borrow count in the year 2017
ON
  SubQuery.BorrowCount = MaxQuery.MaxBorrowCount -- Join the subqueries on the borrow
count to filter the results
```

Result:

```
BorrowMonth,BorrowCount
8,10
7,10
10,10
```

12. Average number of borrows by age

```
SELECT
    FLOOR(DATEDIFF(CURRENT_DATE(), Client.ClientDOB) / 365.25) AS Age, -- Calculate the age of clients by subtracting their date of birth from the current date and dividing it by 365.25 (to account for leap years) and assign it an alias "Age"
    AVG(BorrowCounts.BorrowCount) AS AverageBorrows -- Calculate the average number of borrows for each age group and assign it an alias "AverageBorrows"
FROM
    Client -- Specify the table name "Client"
JOIN (
    SELECT
        Borrower.ClientID,
        COUNT(*) AS BorrowCount
    FROM
        Borrower -- Specify the table name "Borrower"
    GROUP BY
        Borrower.ClientID -- Group the borrowings by client ID
) AS BorrowCounts ON Client.ClientID = BorrowCounts.ClientID -- Join the BorrowCounts subquery with the Client table on client ID
GROUP BY
    Age -- Group the results by age
ORDER BY
    Age ASC; -- Sort the results in ascending order based on age
```

Result:

```
Age,AverageBorrows
13,2.3333
15,6.0000
16,5.0000
17,5.5000
18,4.5000
19,3.0000
20,5.0000
21,2.0000
22,4.5000
```

```
24,3.6667
25,2.0000
27,2.0000
28,4.5000
29,10.0000
31,3.0000
33,5.5000
36,2.0000
37,3.0000
38,4.0000
39,5.5000
40,3.7500
41,3.0000
42,2.0000
43,1.0000
44,4.3333
45,5.5000
46,3.0000
47,3.5000
48,2.6667
49,3.2500
50,3.6667
53,4.5000
55,4.0000
56,3.0000
57,1.0000
60,5.0000
61,3.0000
63,3.6667
```

13. The oldest and the youngest clients of the library

```
SELECT
    Client.ClientFirstName,
    Client.ClientLastName,
    FLOOR(DATEDIFF(CURRENT_DATE(), Client.ClientDOB) / 365.25) AS Age -- Calculate the age of clients by subtracting their date of birth from the current date and dividing it by 365.25 (to account for leap years) and assign it an alias "Age"
FROM
    Client -- Specify the table name "Client"
WHERE
    Client.ClientDOB = (SELECT MIN(ClientDOB) FROM Client) -- Retrieve the clients with the minimum date of birth
    OR Client.ClientDOB = (SELECT MAX(ClientDOB) FROM Client) -- Retrieve the clients with the maximum date of birth
ORDER BY
    Client.ClientDOB; -- Sort the results in ascending order based on the date of birth
```

Result:

```
ClientFirstName,ClientLastName,Age
Mya,Austin,63
Caroline,Clark,63
Annabel,Williamson,63
Alina,Morton,13
Alina,Austin,13
Joan,Grant,13
```

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

```
SELECT
   Author.AuthorFirstName,
   Author.AuthorLastName
FROM
   Author -- Specify the table name "Author"
JOIN
   Book ON Author.AuthorID = Book.AuthorID -- Join the Author and Book tables on the AuthorID column
GROUP BY
   Author.AuthorID, Author.AuthorFirstName, Author.AuthorLastName -- Group the results by AuthorID, AuthorFirstName, and AuthorLastName to eliminate duplicate authors
HAVING
   COUNT(DISTINCT Book.Genre) > 1; -- Filter the groups by authors who have written books in more than one genre
```

Result:

AuthorFirstName,AuthorLastName

Indexes:

By creating the following indexes, the database engine can quickly locate the required data during query execution:

```
-- Index for Book table
CREATE INDEX idx_book_authorid ON Book (AuthorID);

-- Index for Client table
CREATE INDEX idx_client_occupation ON Client (Occupation);

-- Index for Borrower table
CREATE INDEX idx_borrower_clientid ON Borrower (ClientID);
CREATE INDEX idx_borrower_bookid ON Borrower (BookID);
CREATE INDEX idx_borrower_borrowdate ON Borrower (BorrowDate);
```