Computer Science 204 Assignment

USE library2;

/\*Creating Tables\*/

CREATE TABLE Author (

Author\_ID INT PRIMARY KEY,

Author\_First\_Name VARCHAR(40),

Author\_Last\_Name VARCHAR(80),

Author\_Nationality VARCHAR(100)

);

CREATE TABLE Books (

Book\_ID INT PRIMARY KEY,

Book\_Title VARCHAR (200),

Author\_ID INT,

Genre VARCHAR (30)

FOREIGN KEY (AUTHOR\_ID) REFERENCES Author (Author\_ID)

);

CREATE TABLE Borrower (

Borrower\_ID INT PRIMARY KEY,

Client\_ID INT,

Book\_ID INT,

Borrow\_Date date,

FOREIGN KEY(Client\_ID) REFERENCES Client (Client\_ID)

);

/\*Making some adjustments/troubleshooting to the Borrower table\*/

* ALTER TABLE Borrower

DROP FOREIGN KEY borrower\_ibfk\_1;

* ALTER TABLE Borrower

ADD CONSTRAINT fk\_Client\_ID

FOREIGN KEY (Client\_ID) REFERENCES Client(Client\_ID);

CREATE TABLE Client (

Client\_ID INT PRIMARY KEY,

Client\_First\_Name VARCHAR (40),

Client\_Last\_Name VARCHAR (80),

Client\_DOB date,

Client\_Occupation VARCHAR(100)

);

/\* Inserting data into all the tables \*/

INSERT INTO Author 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', 'Wislon', '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');

/\*Fixing Fabian Wilson’s last name\*/

UPDATE Author

Set Author\_Last\_Name = 'Wilson'

WHERE Author\_ID = '9';

INSERT INTO Books 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');

/\*For Client table the year by itself wasn’t working(and I ripped my hair out trying to find a solution) so I settled at adding month and day\*/

INSERT INTO Client VALUES (1,'Kaiden', 'Hill', 2006, 'Student'),

(2,'Alina', 'Morton', "2010-09-26", 'Student'),

(3, 'Fania', 'Brooks', "1983-12-19", 'Food Scientist'),

(4, 'Courtney', 'Jensen', "2006-05-09", 'Student'),

(5, 'Brittany', 'Hill', "1983-06-12", 'Firefighter'),

(6, 'Max', 'Rogers', "2005-11-26", 'Student'),

(7, 'Margaret', 'McCarthy', "1981-04-18", 'School Psychologist'),

(8, 'Julie', 'McCarthy', "1973-03-12", 'Professor'),

(9, 'Ken', 'McCarthy', "1974-08-03", 'Securities Clerk'),

(10, 'Britany', 'Quinn', "1984-07-08", 'Violinist'),

(11, 'Conner', 'Gardner', "1998-02-27", 'Licensed Massage Therapist'),

(12, 'Mya', 'Austin', "1960-01-21", 'Parquet Floor Layer'),

(13, 'Thierry', 'Rogers', "2004-10-17", 'Student'),

(14, 'Eloise', 'Rogers', "1984-06-19", 'Computer Security Manager'),

(15, 'Gerard', 'Jackson', "1979-01-07", 'Oil Exploration Engineer'),

(16, 'Randy', 'Day', "1986-02-02", 'Aircraft Electrician'),

(17, 'Jodie', 'Page', "1990-03-09", 'Manufacturing Director'),

(18, 'Coral', 'Rice', "1996-04-03", 'Window Washer'),

(19, 'Ayman', 'Austin', "2002-05-06", 'Student'),

(20, 'Jaxson', 'Austin', "1999-06-10", 'Repair Worker'),

(21, 'Joel', 'Austin', "1973-07-05", 'Police Officer'),

(22, 'Alina', 'Austin', "2010-08-10", 'Student'),

(23, 'Elin', 'Austin', "1962-09-01", 'Payroll Clerk'),

(24, 'Ophelia', 'Wolf', "2004-10-02", 'Student'),

(25, 'Eliot', 'McGuire', "1967-11-25", 'Dentist'),

(26, 'Peter', 'McKinney', "1968-12-18", 'Professor'),

(27, 'Annabella', 'Henry', "1974-01-02", 'Nurse'),

(28, 'Anastasia', 'Baker', "2001-02-01", 'Student'),

(29, 'Tyler', 'Baker', "1984-03-01", 'Police Officer'),

(30, 'Lilian', 'Ross', "1983-04-01", 'Insurance Agent'),

(31, 'Thierry', 'Arnold', "1975-05-01", 'Bus Driver'),

(32, 'Angelina', 'Rowe', "1979-06-01", 'Firefighter'),

(33, 'Marcia', 'Rowe', "1974-07-01", 'Health Educator'),

(34, 'Martin', 'Rowe', "1976-08-01", 'Ship Engineer'),

(35, 'Adeline', 'Rowe', "2005-09-02", 'Student'),

(36, 'Colette', 'Rowe', "1963-10-01", 'Professor'),

(37, 'Diane', 'Clark', "1975-11-01", 'Payroll Clerk'),

(38, 'Caroline', 'Clark', "1960-12-01", 'Dentist'),

(39, 'Dalton', 'Clayton', "1982-01-03", 'Police Officer'),

(40, 'Steve', 'Clayton', "1990-02-03", 'Bus Driver'),

(41, 'Melanie', 'Clayton', "1987-03-02", 'Computer Engineer'),

(42, 'Alana', 'Wilson', "2007-04-02", 'Student'),

(43, 'Carson', 'Byrne', "1995-05-02", 'Food Scientist'),

(44, 'Conrad', 'Byrne', "2007-06-02", 'Student'),

(45, 'Ryan', 'Porter', "2008-07-02", 'Student'),

(46, 'Elin', 'Porter', "1978-08-02", 'Computer Programmer'),

(47, 'Tyler', 'Harvey', "2007-09-03", 'Student'),

(48, 'Arya', 'Harvey', "2008-10-03", 'Student'),

(49, 'Serena', 'Harvey', "1978-11-02", 'School Teacher'),

(50, 'Lilly', 'Franklin', "1976-12-02", 'Doctor'),

(51, 'Mai', 'Franklin', "1994-01-04", 'Dentist'),

(52, 'John', 'Franklin', "1999-02-04", 'Firefighter'),

(53, 'Judy', 'Franklin', "1995-03-03", 'Firefighter'),

(54, 'Katy', 'Lloyd', "1992-04-04", 'School Teacher'),

(55, 'Tamara', 'Allen', "1963-05-03", 'Ship Engineer'),

(56, 'Maxim', 'Lyons', "1985-06-03", 'Police Officer'),

(57, 'Allan', 'Lyons', "1983-07-03", 'Computer Engineer'),

(58, 'Marc', 'Harris', "1980-08-03", 'School Teacher'),

(59, 'Elin', 'Young', "2009-09-04", 'Student'),

(60, 'Diana', 'Young', "2008-10-04", 'Student'),

(61, 'Diane', 'Young', "2006-11-03", 'Student'),

(62, 'Alana', 'Bird', "2003-12-03", 'Student'),

(63, 'Anna', 'Becker', "1979-01-05", 'Security Agent'),

(64, 'Katie', 'Grant', "1977-02-05", 'Manager'),

(65, 'Joan', 'Grant', "2010-03-04", 'Student'),

(66, 'Bryan', 'Bell', "2001-04-05", 'Student'),

(67, 'Belle', 'Miller', "1970-05-04", 'Professor'),

(68, 'Peggy', 'Stevens', "1990-06-04", 'Bus Driver'),

(69, 'Steve', 'Williamson', "1975-07-04", 'HR Clerk'),

(70, 'Tyler', 'Williamson', "1999-08-04", 'Doctor'),

(71, 'Izabelle', 'Williamson', "1990-09-04", 'Systems Analyst'),

(72, 'Annabel', 'Williamson', "1960-10-05", 'Cashier'),

(73, 'Mohamed', 'Waters', "1966-11-04", 'Insurance Agent'),

(74, 'Marion', 'Newman', "1970-12-04", 'Computer Programmer'),

(75, 'Ada', 'Williams', "1986-01-06", 'Computer Programmer'),

(76, 'Sean', 'Scott', "1983-02-06", 'Bus Driver'),

(77, 'Farrah', 'Scott', "1974-03-05", 'Ship Engineer'),

(78, 'Christine', 'Lambert', "1973-04-06", 'School Teacher'),

(79, 'Alysha', 'Lambert', "2007-05-12", 'Student'),

(80, 'Maia', 'Grant', "1984-06-22", 'School Teacher');

/\*Specifically mentioning the columns where data is to be entered in the INSERT statement because the table is very long and to ensure precision\*/

INSERT INTO Borrower (Borrower\_ID, Client\_ID, Book\_ID, Borrow\_Date) 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');

Query1: Query 1 was successful and if we scroll down it lists all 80 entries in the clients table.

Command used: USE library 2.0

SELECT \* FROM CLIENT;

A screenshot of a computer

Description automatically generated

Query2: Is successful as well it lists the first name, last name, age and occupations of the Clients.  
Command used:  
SELECT Client\_First\_Name, Client\_Last\_Name, TIMESTAMPDIFF(YEAR, Client\_DOB, CURDATE()) AS Age, Client\_Occupation

FROM Client;

A screenshot of a computer

Description automatically generated

Query3: Query works and displays Client first and last names where borrow date is March 2018

Command used:

USE library2;

SELECT c.Client\_First\_Name, c.Client\_Last\_Name

FROM Borrower b

JOIN Client c ON b.Client\_ID = c.Client\_ID

WHERE b.Borrow\_Date BETWEEN '2018-03-01' AND '2018-03-31';

A screenshot of a computer

Description automatically generated

Query4: Names of the top 5 authors are displayed successfully.

Command used:

SELECT

A.Author\_First\_Name,

A.Author\_Last\_Name

FROM

Authors A

JOIN

Books B ON A.Author\_ID = B.Author\_ID

JOIN

Borrowers BR ON B.Book\_ID = BR.Book\_ID

WHERE

YEAR(BR.Borrow\_Date) = 2017

GROUP BY

A.Author\_ID

ORDER BY

COUNT(\*) DESC

LIMIT 5;

A screenshot of a computer

Description automatically generated

Query 5:

I used multiple join statements to pull the info from tables together. Then I filtered by the date, then I grouped and ordered by author nationality. Finally I changed the SELECT statement to only look for the author nationality.

Command used:

SELECT a.Author\_Nationality

FROM borrower b

JOIN books bk ON b.Book\_ID = bk.Book\_ID

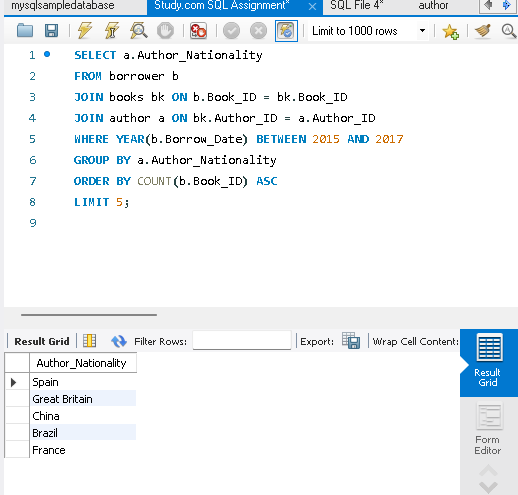
JOIN author a ON bk.Author\_ID = a.Author\_ID

WHERE YEAR(b.Borrow\_Date) BETWEEN 2015 AND 2017

GROUP BY a.Author\_Nationality

ORDER BY COUNT(b.Book\_ID) ASC

LIMIT 5;



Query 6:

Query 6 is very very similar to Query 5. In this query we need to modify the SQL commands in order to retrieve the book most borrowed/top borrowed book. Instead of the 5 author nationalities least borrowed.

Command used:

SELECT bk.Book\_Title, bk.Book\_ID, COUNT(b.Book\_ID) AS Borrow\_Count

FROM borrower b

JOIN books bk ON b.Book\_ID = bk.Book\_ID

WHERE YEAR(b.Borrow\_Date) BETWEEN 2015 AND 2017

GROUP BY b.Book\_ID

ORDER BY Borrow\_Count DESC

LIMIT 1;

A screenshot of a computer

Description automatically generated

Query 7:

Query 7 is asking for the top borrowed genre. This can also be achieved using multiple JOINs. Then we can filter for the DOBs of the clients. Then use the COUNT function to count them and sort them. Finally I changed the SELECT statement to select the distinct Genre from Book table because we didn’t need redundant results showing up.

Command used:

USE library2;

SELECT DISTINCT bk.Genre

FROM borrower b

JOIN client c ON b.Client\_ID = c.Client\_ID

JOIN books bk ON b.Book\_ID = bk.Book\_ID

WHERE YEAR(c.Client\_DoB) BETWEEN 1970 AND 1980

GROUP BY bk.Genre

ORDER BY COUNT(\*) DESC;

A screenshot of a computer

Description automatically generated

Question 8:

I was a little bit confused by this query as on surface it seems very similar to previous queries and can be crafted with multiple JOINs. But I soon realized I had two tables one named Client and another named Clients. Hence a lot of headscratching for this query. But yes it is a similar query with a different filtering requirement.

Command used:

USE library2;

SELECT c.Client\_Occupation, COUNT(b.Book\_ID) AS Borrow\_Count

FROM borrower b

JOIN client c ON b.Client\_ID = c.Client\_ID

WHERE YEAR(b.Borrow\_Date) = 2016

GROUP BY c.Client\_Occupation

ORDER BY Borrow\_Count DESC

LIMIT 5;

A screenshot of a computer

Description automatically generated

Query 9:

For this query first we can join the Borrower and Client tables. Then we can filter in select statement for the client occupations and use the count function to count how many books were borrowed by each occupation. Next best way to get our desired result is to make our previous query a nested/sub query. And put the count function together with the average function to get the average books borrowed by client occupation.

Command used:

USE library2;

SELECT Client\_Occupation, AVG(Borrow\_Count) AS Avg\_Books\_Borrowed

FROM (

SELECT c.Client\_Occupation, COUNT(b.Book\_ID) AS Borrow\_Count

FROM borrower b

JOIN client c ON b.Client\_ID = c.Client\_ID

GROUP BY c.Client\_Occupation, b.Client\_ID

) AS subquery

GROUP BY Client\_Occupation;

A screenshot of a computer

Description automatically generated

Query 10:

For query 10 we can start by joining the Books and the Borrower tables. Grouping by Book\_ID. Then we can use the HAVING clause instead of the WHERE clause because we are going to be using aggregate functions. We can count the number of clients who borrowed books from the Borrower table using DISTINCT because we do not want to count repeating figures. Then we count the 20% of clients. Finally putting it all together we can filter for the book title, count the client ID and display it as borrower count. In the very end we can store it as a view by creating the view as popularbooks.

I was trying to simplify the query using 16. Because the 20 percent of 80 clients is 80. But kept getting error 1064.

Command used:

CREATE VIEW PopularBooks AS

SELECT

B.Book\_Title,

COUNT(DISTINCT BR.Client\_ID) AS BorrowerCount

FROM

Books B

JOIN

Borrower BR ON B.Book\_ID = BR.Book\_ID

GROUP BY

B.Book\_ID

HAVING

COUNT(DISTINCT BR.Client\_ID) >= (SELECT COUNT(\*) \* 0.20 FROM Client);

A close-up of a web page

Description automatically generated

Query 11:

Compared to the past few queries this is a much simpler query and doesn’t require so much heavy lifting. We are only required to filter for the month where most books are borrowed in 2017. The top months are July, August and October.

Command used:

SELECT

MONTH(Borrow\_Date) AS 'Top Months',

COUNT(\*) AS BorrowCount

FROM

Borrower

WHERE

YEAR(Borrow\_Date) = 2017

GROUP BY

MONTH(Borrow\_Date)

ORDER BY

BorrowCount DESC;

A screenshot of a computer program

Description automatically generated

Query 12:

We can join Borrower and Client tables. Then we can compute the age by using the Client DoBs. Then divide number of borrows by the actual clients borrowing to get the average number of borrows. We can also do this step by step but that would make our query much longer and more complex.

Command used:

SELECT

TIMESTAMPDIFF(YEAR, C.Client\_DOB, CURDATE()) AS Age,

COUNT(B.Borrower\_ID) / COUNT(DISTINCT C.Client\_ID) AS AverageBorrows

FROM

Client C

JOIN

Borrower B ON C.Client\_ID = B.Client\_ID

GROUP BY

Age

ORDER BY

Age;

A screenshot of a computer

Description automatically generated

Query 13:

We can either find both the youngest and oldest borrowers. By either two separate queries or we can use a Union to join the two queries together. Which reduces the steps we must take. And the code is pretty simple anyways.

Hence the command we end up with:

USE library2;

SELECT

Client\_First\_Name,

Client\_Last\_Name,

Client\_DOB,

'Oldest' AS Category

FROM

Client

WHERE

Client\_DOB = (SELECT MIN(Client\_DOB) FROM Client)

UNION

SELECT

Client\_First\_Name,

Client\_Last\_Name,

Client\_DOB,

'Youngest' AS Category

FROM

Client

WHERE

Client\_DOB = (SELECT MAX(Client\_DOB) FROM Client);

A screenshot of a computer

Description automatically generated

Query 14:

Honestly, I have spent soooo much time trying to figure out how to do indexes. I cannot for the life of me figure it out. So, I give up on this question ☹

Some helpful resources used:

<https://www.w3schools.com/mysql/mysql_sql.asp>

<https://www.youtube.com/watch?v=HXV3zeQKqGY&t=6605s>

<https://www.youtube.com/watch?v=7S_tz1z_5bA>