

George Mason University

Library Management System

CS-504--002

Final Project

Microsoft SQL Server

Satyam Singh
G#01389368

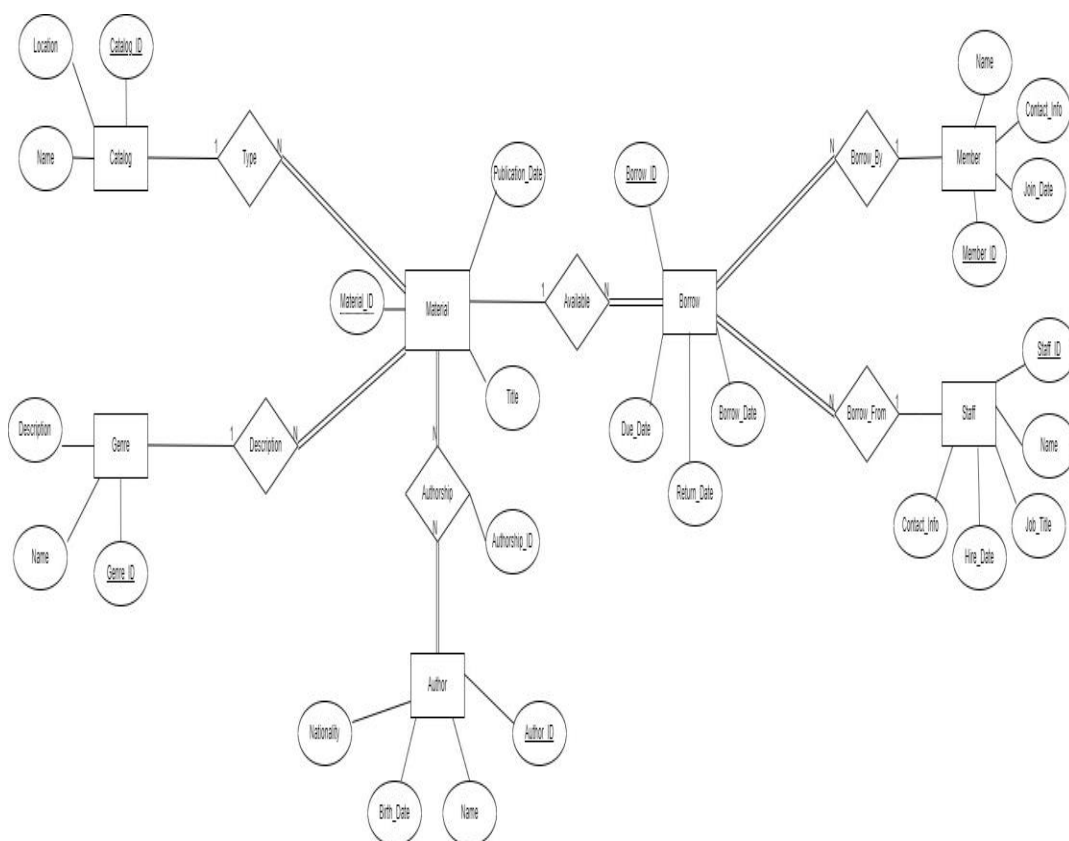
Scope

This project aims to develop a library management system for a public library which can help in tracking of the library resources. As a library consists of different types of materials, managing everything via traditional methods would be lengthy and can be faulty as well. Using a Relational database at this scenario is the perfect fit as it can add many other facilities. For this database system to be in a proper form so that data integrity is maintained we must go according to the guidelines.

In this Library Database we have 7 entities and many attributes within,

1. Material
2. Borrow
3. Author
4. Catalog
5. Genre
6. Member
7. Staff

To better understand the relationships and entities we use the ER (Entity Relationship) Diagram



ER DIAGRAM

ER model is a conceptual model of the database and through this we can start implementing the database. It is the first step in the implementation of database.

In the ER Diagram we can see all the entities with their attributes and connecting relationships. It is important for us to know how the relationship affects the database. All the attributes are simple attributes and primary key is marked with an underline.

The cardinality ration for different relations is different. Most of the relations are 1:N , example,

Catalog : Material is one to many, this is because one catalog can have many materials within but one material can only have one catalog. Similarly for Genre and Material, Material and Borrow, Borrow and Member, Borrow and Staff. One material can have many borrows, but one borrow can only have one member is another example of one to many relationships.

Now we see, many: many relationships. M:N,

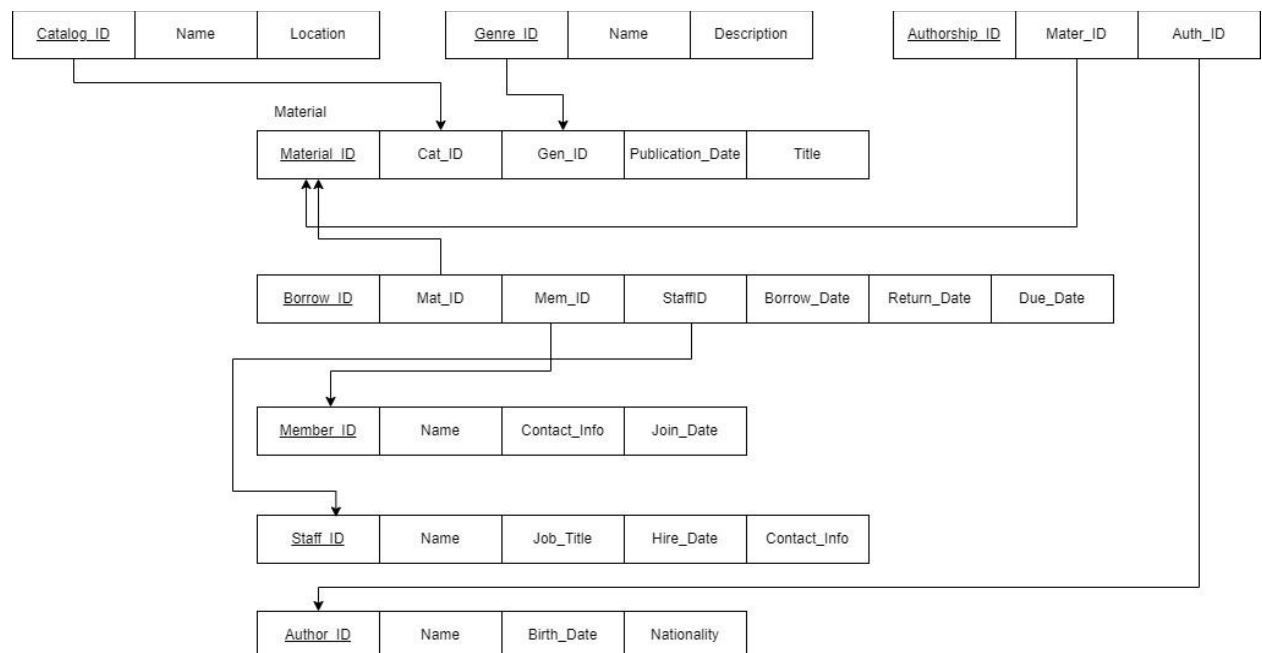
Material:Author is a many to many relationship, as one author can have many books (materials) and one material can have many authors. Hence it is the only many to many relationship in the database.

Partial participation is when all the elements need not take part in the relationship. So, in the case of Catalog and Material. Catalog is partially participating and Material has full participation as it is not necessary for all catalog types to take part in the relationship, some types may not have any materials within, but materials has total participation as all materials must have a catalog type. This same logic can be applied for Genre and Material. For Author and Material, All authors have a material so it is fully participating and all materials have an author hence it has total participation as well.

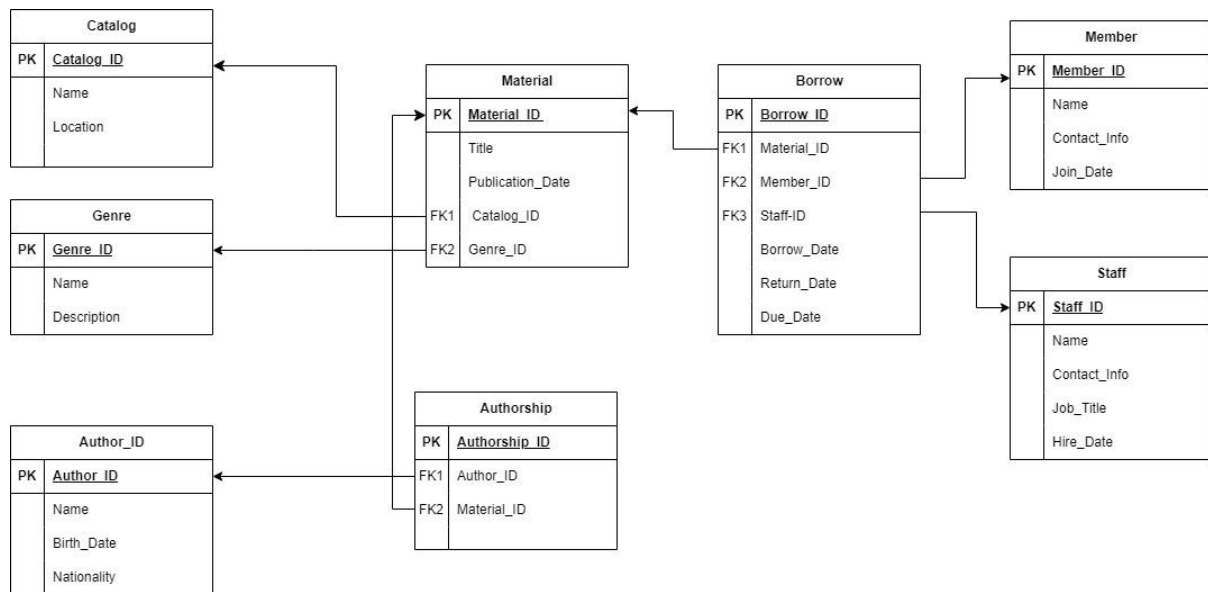
After understanding the ER diagram, we move on to the relational schema design. This is made after the ER diagram as we need to understand the relationships between the entities to make this.

To convert the ER model into a relational schema we must follow 7 steps. After following all the steps, we get the following as our relational schema.

Relational schema



Relational Schema



For this project We will use Microsoft SQL server as the database management system. After understanding the structure of the database we create tables and add attributes to it via DDL(Data Definition Language) Commands. Creation of all tables must be done with due diligence by adding attributes with their specific datatype and constraints. As we have made the ER diagram and relational schema before we now know all the primary and foreign keys and how all entities are linked with each other.

Also, the database in this schema diagram is already in 3NF form hence we do not need to further normalize it. This ensures that data integrity is maintained, and redundancy is reduced.

Data Definition Language (DDL) Commands

We use create statements to CREATE tables and add foreign key constraints via ALTER table commands. The following are snapshots of DDL statements.

```
--DDL COMMANDS
--Table Creation
CREATE TABLE Material (...);
CREATE TABLE Catalog (
    Catalog_ID INT NOT NULL,
    Name VARCHAR(255) NOT NULL,
    Location VARCHAR(255) NOT NULL,
    PRIMARY KEY (Catalog_ID)
);
CREATE TABLE Genre (
    Genre_ID INT PRIMARY KEY,
    Name VARCHAR(50) NOT NULL,
    Description VARCHAR(255) NOT NULL
);
CREATE TABLE Borrow (
    Borrow_ID INT PRIMARY KEY,
    Material_ID INT,
    Member_ID INT,
    Staff_ID INT,
    Borrow_Date DATE,
    Due_Date DATE,
    Return_Date DATE
);
CREATE TABLE Author (
    Author_ID INT PRIMARY KEY,
    Name VARCHAR(255),
    Birth_Date DATE,
    Nationality VARCHAR(255)
);
CREATE TABLE Authorship (
    Authorship_ID INT PRIMARY KEY,
    Author_ID INT,
    Material_ID INT
);
CREATE TABLE Member (
    Member_ID INT NOT NULL PRIMARY KEY,
    Name VARCHAR(50) NOT NULL,
    Contact_Info VARCHAR(50) NOT NULL,
    Join_Date DATE NOT NULL
);

);
CREATE TABLE Staff (
    Staff_ID INT PRIMARY KEY,
    Name VARCHAR(255) NOT NULL,
    Contact_Info VARCHAR(255),
    Job_Title VARCHAR(255) NOT NULL,
    Hire_Date DATE
);
--ADDING CONSTRAINTS
ALTER TABLE Material
ADD CONSTRAINT fk_catalog
FOREIGN KEY (Catalog_ID)
REFERENCES Catalog(Catalog_ID);

ALTER TABLE Material
ADD CONSTRAINT fk_genre
FOREIGN KEY (Genre_ID)
REFERENCES Genre(Genre_ID);

ALTER TABLE Borrow
ADD CONSTRAINT fk_Borrow_Material
FOREIGN KEY (Material_ID)
REFERENCES Material(Material_ID);

ALTER TABLE Borrow
ADD CONSTRAINT fk_Borrow_Member
FOREIGN KEY (Member_ID)
REFERENCES Member(Member_ID);

ALTER TABLE Borrow
ADD CONSTRAINT FK_Borrow_Staff
FOREIGN KEY (Staff_ID)
REFERENCES Staff(Staff_ID);

ALTER TABLE Authorship
ADD CONSTRAINT FK_Authorship_Author_ID
FOREIGN KEY (Author_ID)
REFERENCES Author(Author_ID)
ON UPDATE CASCADE
ON DELETE CASCADE;

ALTER TABLE Authorship
ADD CONSTRAINT FK_Authorship_Material_ID
FOREIGN KEY (Material_ID)
REFERENCES Material(Material_ID)
ON UPDATE CASCADE
ON DELETE CASCADE;
```

Data Manipulation Language (DML)

Here we are going to insert the table values via INSERT query.

```
--DML
--INSERTING VALUES
INSERT INTO Catalog (Catalog_ID, Name, Location)
VALUES
(1, 'Books', 'A2.1'),
(2, 'Magazines', 'B2.1'),
(3, 'E-books', 'C2.1'),
(4, 'Audiobooks', 'D4.1'),
(5, 'Journals', 'E5.1'),
(6, 'Newspapers', 'F6.1'),
(7, 'Maps', 'G7.1'),
(8, 'News', 'H8.1'),
(9, 'Sweet Music', 'I9.1'),
(10, 'Educational', 'J10.1');

INSERT INTO Genre (Genre_ID, Name, Description)
VALUES
(1, 'General Fiction', 'Library works with a focus on character and plot development, exploring various t'),
(2, 'Mystery & Thriller', 'Suspenseful stories centered around crime, investigation, or espionage with an'),
(3, 'Science Fiction & Fantasy', 'Imaginative works that explore alternate realities, futuristic concepts, a'),
(4, 'Horror & Suspense', 'Stories designed to evoke fear, unease, or dread, often featuring supernatural e'),
(5, 'Dystopian & Apocalyptic', 'Depictions of societies in decline or collapse, often exploring themes of'),
(6, 'Classics', 'Enduring works of literature that have stood the test of time, often featuring rich lang'),
(7, 'Historical Fiction', 'Fictional stories set in the past, often based on real historical events or fig'),
(8, 'Epic Poetry & Mythology', 'Ancient or traditional stories and poems, often featuring heroes, gods, a'),

);
INSERT INTO Member (Material_ID, Title, Publication_Date, Catalog_ID, Genre_ID)
VALUES
(1, 'The Catcher in the Rye', '1951-07-16', 1, 1),
(2, 'To Kill a Mockingbird', '1960-07-11', 2, 1),
(3, 'The Da Vinci Code', '2003-04-01', 3, 2),
(4, 'The Hobbit', '1937-09-11', 4, 3),
(5, 'The Wizard of Oz', '1939-08-01', 5, 4),
(6, 'Pride and Prejudice', '1813-01-28', 1, 1),
(7, 'The Great Gatsby', '1925-04-10', 2, 1),
(8, 'Harry Potter', '1997-06-19', 3, 5),
(9, 'Crime and Punishment', '1866-01-01', 4, 1),
(10, 'The Hitchhiker's Guide to the Galaxy', '1979-10-12', 5, 3),
(11, '1984', '1949-06-08', 1, 5),
(12, 'Animal Farm', '1945-08-17', 2, 5),
(13, 'The Hunting of the Snail', '1950-10-17', 3, 4),
(14, 'Brave New World', '1932-08-01', 4, 5),
(15, 'The Chronicles of Narnia: The Lion, the Witch and the Wardrobe', '1950-06-16', 5, 3),
(16, 'The Adventures of Huckleberry Finn', '1884-12-10', 6, 1),
(17, 'The Catcher in the Rye', '1951-07-16', 7, 1),
(18, 'The Picture of Dorian Gray', '1890-07-01', 8, 1),
(19, 'The Call of Cthulhu', '1928-02-01', 9, 4),
(20, 'Harry Potter and the Philosopher's Stone', '1997-06-20', 10, 3),
(21, 'Frankenstein', '1818-01-01', 6, 4),
(22, 'A Tale of Two Cities', '1859-04-30', 7, 1),
(23, 'The Idiot', '1869-02-01', 8, 6),
(24, 'The Odyssey', '1719-06-01', 9, 8);

INSERT INTO Author (Author_ID, Name, Birth_Date, Nationality)
VALUES
(1, 'Jane Austen', '1775-12-16', 'British'),
(2, 'Ernest Hemingway', '1899-07-21', 'American'),
(3, 'George Orwell', '1903-01-25', 'British'),
(4, 'Scott Fitzgerald', '1896-09-24', 'American'),
(5, 'J.K. Rowling', '1965-07-31', 'British'),
(6, 'Mark Twain', '1835-11-30', 'American'),
(7, 'Leo Tolstoy', '1828-09-09', 'Russian'),
(8, 'Virginia Woolf', '1882-01-23', 'British'),
(9, 'Gabriel Marquez', '1927-03-06', 'Colombian'),
(10, 'Charles Dickens', '1812-02-07', 'British'),
(11, 'Harper Lee', '1926-04-28', 'American'),
(12, 'Oscar Wilde', '1854-10-15', 'Irish'),
(13, 'William Shakespeare', '1564-04-26', 'British'),
(14, 'Franz Kafka', '1883-07-03', 'Czech'),
(15, 'James Joyce', '1882-02-02', 'Irish'),
(16, 'J.R.R. Tolkien', '1892-01-03', 'British'),
(17, 'Emily Bronte', '1818-07-30', 'British'),
(18, 'Toni Morrison', '1931-02-18', 'American'),
(19, 'Fyodor Dostoevsky', '1818-11-11', 'Russian'),
(20, 'Lucas Pike', '1847-10-16', 'British');

INSERT INTO Authorship (Authorship_ID, Author_ID, Material_ID)
VALUES
(1, 1, 1),
(2, 2, 2),
(3, 3, 3),
(4, 4, 4),
(5, 5, 5),
(6, 6, 6),
(7, 7, 7),
(8, 8, 8),
(9, 9, 9),
(10, 10, 10),
(11, 11, 11),
(12, 12, 12),
(13, 13, 13),
(14, 14, 14),
(15, 15, 15),
(16, 16, 16),
(17, 17, 17),
(18, 18, 18),
(19, 19, 19),
(20, 20, 20),
(21, 21, 21),
(22, 22, 22),
(23, 23, 23);

INSERT INTO Staff (Staff_ID, Name, Contact_Info, Job_Title, Hire_Date)
VALUES
(1, 'Jay Green', 'jay.green@gmail.com', 'Librarian', '2017-08-01'),
(2, 'Brian Taylor', 'brian.taylor@gmail.com', 'Library Assistant', '2020-12-15'),
(3, 'Christine King', 'christine.king@gmail.com', 'Library Assistant', '2020-04-08'),
(4, 'David Winger', 'david.winger@gmail.com', 'Library Technician', '2020-05-02');

INSERT INTO Borrow (Borrow_ID, Material_ID, Member_ID, Staff_ID, Borrow_Date, Due_Date, Return_Date)
VALUES
(1, 1, 1, 1, '2020-09-12', '2020-09-01', '2020-09-30'),
(2, 2, 1, '2020-10-25', '2020-11-05', '2020-10-20'),
(3, 3, 1, '2020-11-20', '2020-12-05', '2020-11-08'),
(4, 4, 1, '2020-01-01', '2020-01-01', '2020-01-01'),
(5, 5, 1, '2020-01-20', '2020-01-11', '2020-01-05'),
(6, 6, 2, '2020-01-07', '2020-01-20', '2020-01-12'),
(7, 7, 2, '2020-01-20', '2020-01-01', '2020-01-20'),
(8, 8, 1, '2020-11-08', '2020-11-20', '2020-11-20'),
(9, 9, 1, '2020-02-01', '2020-02-05', '2020-02-01'),
(10, 10, 1, '2020-01-01', '2020-01-01', '2020-01-01'),
(11, 1, 1, '2020-05-10', '2020-05-04', '2020-05-10'),
(12, 1, 1, '2020-07-11', '2020-07-11', '2020-07-11'),
(13, 2, 1, '2020-07-11', '2020-07-11', '2020-07-11'),
(14, 2, 2, '2020-09-11', '2020-09-11', '2020-09-11'),
(15, 5, 2, '2020-01-01', '2020-01-01', '2020-01-01'),
(16, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(17, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(18, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(19, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(20, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(21, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(22, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(23, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(24, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(25, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(26, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(27, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(28, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(29, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(30, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(31, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(32, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(33, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(34, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(35, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(36, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(37, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(38, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(39, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(40, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(41, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(42, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(43, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(44, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(45, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(46, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(47, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(48, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(49, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(50, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(51, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(52, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(53, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(54, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(55, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(56, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(57, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(58, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(59, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(60, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(61, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(62, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(63, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(64, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(65, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(66, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(67, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(68, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(69, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(70, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(71, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(72, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(73, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(74, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(75, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(76, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(77, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(78, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(79, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(80, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(81, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(82, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(83, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(84, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(85, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(86, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(87, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(88, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(89, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(90, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(91, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(92, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(93, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(94, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(95, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(96, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(97, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(98, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(99, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(100, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(101, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(102, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(103, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(104, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(105, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(106, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(107, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(108, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(109, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(110, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(111, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(112, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(113, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(114, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(115, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(116, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(117, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(118, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(119, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(120, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(121, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(122, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(123, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(124, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(125, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(126, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(127, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(128, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(129, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(130, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(131, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(132, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(133, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(134, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(135, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(136, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(137, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(138, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(139, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(140, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(141, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(142, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(143, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(144, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(145, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(146, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(147, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(148, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(149, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(150, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(151, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(152, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(153, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(154, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(155, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(156, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(157, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(158, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(159, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(160, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(161, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(162, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(163, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(164, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(165, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(166, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(167, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(168, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(169, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(170, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(171, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(172, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(173, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(174, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(175, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(176, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(177, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(178, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(179, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(180, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(181, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(182, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(183, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(184, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(185, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(186, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(187, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(188, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(189, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(190, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(191, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(192, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(193, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(194, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(195, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(196, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(197, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(198, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(199, 6, 3, '2020-01-01', '2020-01-01', '2020-01-01'),
(200, 6, 3, '2020-01-0
```

Tables

After running the DDL and DML commands we have the following tables.

Material_ID	Title	Publication_Date	Catalog_ID	Genre_ID
1	The Catcher in the Rye	1951-07-16	1	1
2	To Kill a Mockingbird	1960-07-11	2	1
3	The Da Vinci Code	2003-04-01	3	2
4	The Hobbit	1937-09-21	4	3
5	The Shining	1977-01-28	5	4
6	Pride and Prejudice	1813-01-28	1	1
7	The Great Gatsby	1925-04-10	2	1
8	Moby Dick	1851-10-18	3	1
9	Crime and Punishment	1866-01-01	4	1
10	The Hitchhiker's Guide to the Galaxy	1979-10-12	5	3
11	1984	1949-06-08	1	5
12	Animal Farm	1945-08-17	2	5
13	The Hound of the Baskin House	1959-10-17	3	4
14	Brave New World	1932-08-01	4	5
15	The Chronicles of Narnia: The Lion, the Witch & the Wardrobe	1950-10-16	5	3
16	The Adventures of Huckleberry Finn	1884-12-10	6	1
17	The Catch-22	1961-10-11	7	1
18	The Picture of Dorian Gray	1890-07-01	8	1
19	The Divine Comedy	1320-01-01	9	6
20	Harry Potter and the Philosopher's Stone	1997-06-25	10	3
21	Frankenstein	1818-01-01	6	4
22	A Tale of Two Cities	1859-04-30	7	1
23	The Iliad	1750-01-01	8	6
24	The Odyssey	1725-01-01	9	6
25	The Brothers Karamazov	1880-01-01	10	1
26	The Divine Comedy	1320-01-01	6	6
27	The Grapes of Wrath	1939-04-14	7	1
28	The Old Man and the Sea	1952-09-01	8	1
29	The Court of Monte Cristo	1844-01-01	9	1
30	A Midsummer Night's Dream	1596-01-01	10	7
31	The Tricky Book	1888-01-01	10	7

Catalog_ID	Name	Location
1	Books	A1.1
2	Magazines	B2.1
3	E-Books	C3.1
4	Audiobooks	D4.1
5	Journals	E5.1
6	Newspaper	F6.1
7	Maps	G7.1
8	Novels	H8.1
9	Sheet Music	I9.1
10	Educational	J10.1

Author_ID	Name	Birth_Date	Nationality
1	Jane Austen	1775-12-16	British
2	Ernest Hemingway	1899-07-21	American
3	George Orwell	1903-06-25	British
4	Scott Fitzgerald	1896-09-24	American
5	J.K. Rowling	1965-07-31	British
6	Mark Twain	1835-11-30	American
7	Leo Tolstoy	1828-09-09	Russian
8	Virginia Woolf	1882-01-25	British
9	Gabriel Márquez	1927-03-06	Colombian
10	Charles Dickens	1812-02-07	British
11	Harper Lee	1926-04-28	American
12	Oscar Wilde	1854-10-16	Irish
13	William Shakespeare	1564-04-26	British
14	Franz Kafka	1883-07-03	Czech
15	James Joyce	1882-02-02	Irish
16	J.R.R. Tolkien	1892-01-03	British
17	Emily Brontë	1818-07-30	British
18	Toni Morrison	1931-02-18	American
19	Fyodor Dostoevsky	1821-11-11	Russian
20	Lucas Plki	1847-10-16	British

Material

Catalog

Author

Borrow_ID	Material_ID	Member_ID	Staff_ID	Borrow_Date	Due_Date	Return_Date
1	1	1	1	2018-08-12	2018-10-03	2018-08-30
2	2	2	1	2018-10-15	2018-11-05	2018-10-29
3	3	3	1	2018-12-20	2019-01-10	2019-01-08
4	4	4	1	2019-01-11	2019-04-01	2019-03-27
5	5	5	1	2019-04-20	2019-05-11	2019-05-05
6	6	6	1	2019-07-05	2019-07-26	2019-07-21
7	7	7	1	2019-08-10	2019-10-01	2019-09-29
8	8	8	1	2019-11-03	2019-11-29	2019-11-20
9	9	9	1	2020-01-15	2020-02-05	2020-02-03
10	10	10	1	2020-03-12	2020-04-02	2020-03-28
11	11	1	2	2020-05-14	2020-06-04	2020-05-28
12	12	2	2	2020-07-21	2020-08-11	2020-08-02
13	13	3	2	2020-09-25	2020-10-16	2020-10-10
14	14	4	2	2020-11-03	2020-11-29	2020-11-24
15	15	5	2	2021-01-03	2021-01-24	2021-01-19
16	16	6	2	2021-02-18	2021-03-11	2021-03-12
17	17	7	2	2021-04-27	2021-05-18	2021-05-20
18	18	8	2	2021-06-13	2021-07-04	2021-06-28
19	19	9	2	2021-08-15	2021-09-06	2021-09-03
20	20	10	2	2021-10-21	2021-11-11	2021-11-05
21	21	1	3	2021-11-29	2021-12-20	2021-12-08
22	22	2	3	2022-01-10	2022-01-31	2022-01-25
23	23	3	3	2022-02-07	2022-02-28	2022-02-23
24	24	4	3	2022-03-11	2022-04-01	2022-03-28
25	25	5	3	2022-04-28	2022-05-19	2022-05-18
26	26	6	3	2022-06-22	2022-07-13	2022-07-08
27	27	7	3	2022-08-04	2022-08-26	2022-08-23
28	28	8	3	2022-09-11	2022-10-04	2022-09-28
29	29	9	3	2022-10-16	2022-11-06	2022-11-05
30	30	10	3	2022-11-21	2022-12-12	2022-12-05
31	31	1	4	2022-12-28	2023-01-18	2023-01-17
32	32	2	4	2023-01-23	2023-02-12	2023-02-11
33	33	3	4	2023-02-02	2023-02-23	2023-02-17
34	34	4	4	2023-03-01	2023-03-22	2023-03-19
35	35	5	4	2023-03-10	2023-03-31	2023-03-28
36	36	6	4	2023-03-15	2023-04-05	2023-04-05
37	37	7	4	2023-03-25	2023-04-15	2023-04-15
38	38	8	4	2023-03-30	2023-04-20	2023-04-20
39	39	9	4	2023-03-26	2023-04-16	2023-04-16
40	40	10	4	2023-03-28	2023-04-18	2023-04-18

Authorship_ID	Author_ID	Material_ID
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9
10	10	10
11	11	11
12	12	12
13	13	13
14	14	14
15	15	15
16	16	16
17	17	17
18	18	18
19	19	19
20	20	20
21	21	21
22	22	22
23	23	23
24	24	24
25	25	25
26	26	26
27	27	27
28	28	28
29	29	29
30	30	30
31	31	31
32	32	32
33	33	33

Member_ID	Name	Contact_Info	Join_Date
1	Alice Johnson	alice.johnson@email.com	2018-01-10
2	Bob Smith	bob.smith@email.com	2018-03-15
3	Carol Brown	carol.brown@email.com	2018-06-20
4	David Williams	david.williams@email.com	2018-09-18
5	Emily Miller	emily.miller@email.com	2019-02-12
6	Frank Davis	frank.davis@email.com	2019-05-25
7	Grace Wilson	grace.wilson@email.com	2019-08-15
8	Harry Garcia	harry.garcia@email.com	2019-11-27
9	Isla Thomas	isla.thomas@email.com	2020-03-04
10	Jack Martinez	jack.martinez@email.com	2020-07-01
11	Kate Anderson	kate.anderson@email.com	2020-09-30
12	Luke Jackson	luke.jackson@email.com	2021-01-18
13	Mia White	mia.white@email.com	2021-04-27
14	Noah Harris	noah.harris@email.com	2021-07-13
15	Olivia Clark	olivia.clark@email.com	2021-10-05
16	Peter Lewis	peter.lewis@email.com	2021-12-01
17	Quinn Hall	quinn.hall@email.com	2022-02-28
18	Rachel Young	rachel.young@email.com	2022-06-17
19	Sam Walker	sam.walker@email.com	2022-09-25
20	Tiffany Allen	tiffany.allen@email.com	2022-12-10

Borrow

Authorship

Member

Staff_ID	Name	Contact_Info	Job_Title	Hire_Date
1	Amy Green	amy.green@email.com	Librarian	2017-06-01
2	Brian Taylor	brian.taylor@email.com	Library Assistant	2018-11-15
3	Christine King	chris.king@email.com	Library Assistant	2019-05-20
4	Daniel Wright	dan.wright@email.com	Library Technician	2020-02-01

Genre_ID	Name	Description
1	General Fiction	Literary works with a focus on character and plot.
2	Mystery & Thriller	Suspenseful stories centered around crime, investigation, and mystery.
3	Science Fiction & Fantasy	Imaginative works that explore alternate realities, futuristic technology, and mythical worlds.
4	Horror & Suspense	Stories designed to evoke fear, unease, or dread.
5	Dystopian & Apocalyptic	Depictions of societies in decline or collapse, often exploring themes of power, resistance, and survival.
6	Classics	Enduring works of literature that have stood the test of time.
7	Historical Fiction	Fictional stories set in the past, often based on historical events or figures.
8	Epic Poetry & Mythology	Ancient or traditional stories and poems, often featuring heroic figures and divine interventions.

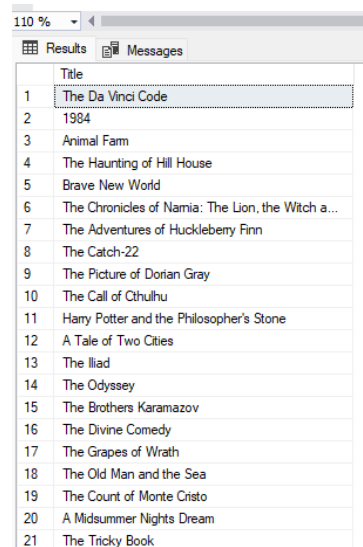
Staff

Genre

Query

Q1. Which materials are currently available in the library?

```
SELECT Title
FROM Material
WHERE Material_ID NOT IN
(SELECT Material_ID FROM Borrow WHERE Return_Date IS NULL);
```

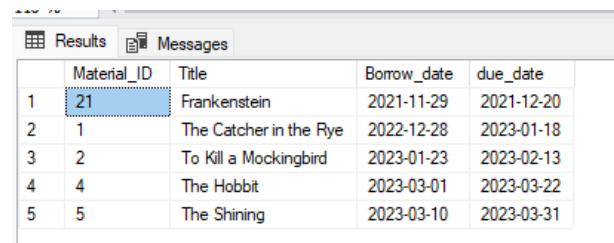


The screenshot shows a database query result window with a zoom level of 110%. The 'Results' tab is active, displaying a list of 21 book titles. The first title, 'The Da Vinci Code', is highlighted. The titles are numbered 1 through 21.

	Title
1	The Da Vinci Code
2	1984
3	Animal Farm
4	The Haunting of Hill House
5	Brave New World
6	The Chronicles of Narnia: The Lion, the Witch a...
7	The Adventures of Huckleberry Finn
8	The Catch-22
9	The Picture of Dorian Gray
10	The Call of Cthulhu
11	Harry Potter and the Philosopher's Stone
12	A Tale of Two Cities
13	The Iliad
14	The Odyssey
15	The Brothers Karamazov
16	The Divine Comedy
17	The Grapes of Wrath
18	The Old Man and the Sea
19	The Count of Monte Cristo
20	A Midsummer Nights Dream
21	The Tricky Book

Q2. Which materials are currently overdue? Suppose today is 04/01/2023, and show the borrow date and due date of each material.

```
SELECT Material.Material_ID, Material.Title, Borrow.Borrow_date, Borrow.due_date
FROM Material
JOIN Borrow ON Material.Material_id = Borrow.Material_ID
WHERE Borrow.Return_Date IS NULL AND Borrow.Due_Date < '2023-04-01';
```



The screenshot shows a database query result window with a zoom level of 110%. The 'Results' tab is active, displaying a table with 5 rows of overdue materials. The first row is highlighted. The columns are Material_ID, Title, Borrow_date, and due_date.

	Material_ID	Title	Borrow_date	due_date
1	21	Frankenstein	2021-11-29	2021-12-20
2	1	The Catcher in the Rye	2022-12-28	2023-01-18
3	2	To Kill a Mockingbird	2023-01-23	2023-02-13
4	4	The Hobbit	2023-03-01	2023-03-22
5	5	The Shining	2023-03-10	2023-03-31

Q3. What are the top 10 most borrowed materials in the library? Show the title of each material and order them based on their available counts.

```
Select Material.Title, count(Borrow.Material_ID) FROM Borrow join Material on
borrow.Material_Id = Material.Material_ID
Group By Material.Title
Order By count(Borrow.Material_ID) DESC
OFFSET 0 ROWS FETCH FIRST 10 ROWS ONLY;
```

	Title	(No column name)
1	The Catcher in the Rye	3
2	Pride and Prejudice	3
3	The Da Vinci Code	3
4	The Hobbit	3
5	The Shining	3
6	To Kill a Mockingbird	3
7	Moby Dick	2
8	Crime and Punishment	2
9	The Great Gatsby	2
10	The Hitchhiker's Guide to the Galaxy	2

Q4. How many books has the author Lucas Piki written?

```
SELECT COUNT(authorship.Author_ID) AS num_books
FROM Authorship
JOIN Author ON Authorship.Author_ID = Author.Author_ID
WHERE Author.Name = 'Lucas Piki'
```

	num_books
1	1

Q5. How many books were written by two or more authors?

```
SELECT COUNT(*) AS Number_of_Books
FROM (
  SELECT COUNT(Author_ID) AS Num_Aut, Material_ID
  FROM Authorship
  GROUP BY Material_ID
  HAVING COUNT(Author_ID) > 1
) AS books_with_multiple_authors;
```

	Number_of_Books
1	3

Q6. What are the most popular genres in the library?

```
Select g.Name, count(*) From  
Material m join Borrow b ON m.Material_ID = b.Material_ID  
JOIN Genre g ON m.Genre_ID = g.Genre_ID  
GROUP BY g.Name  
Order by Count(*) DESC
```

Results Messages		
	Name	(No column name)
1	General Fiction	22
2	Science Fiction & Fantasy	6
3	Horror & Suspense	5
4	Mystery & Thriller	3
5	Classics	3
6	Historical Fiction	1

Q7. How many materials have been borrowed from 09/2020-10/2020?

```
SELECT COUNT(*) as Num_of_Borrows  
FROM borrow  
WHERE borrow_date >= '2020-09-01' AND borrow_date < '2020-11-01';
```

Results Messages	
	Num_of_Borrows
1	1

Q8. How do you update the “Harry Potter and the Philosopher's Stone” when it is returned on 04/01/2023?

In the Material Table we first see the Material_ID value for ‘Harry Potter and the Philosopher’s Stone’. Then we check the return date for the ID value in the borrow table. We can see that it is currently, 2021-11-05. As per the query demands we need to change it to 04-01-2023.

```
Select * FROM Material
SELECT * FROM Borrow

UPDATE Borrow
SET Return_Date = '2023-04-01'
WHERE Material_ID = (SELECT Material_ID FROM Material WHERE Title = 'Harry Potter and the Philosopher's Stone');

SELECT * FROM Borrow
```

We again check the Return Date value of Material_ID 20 in Borrow Table to make sure changes have taken place.

Borrow_ID	Material_ID	Member_ID	Staff_ID	Borrow_Date	Due_Date	Return_Date
1	1	1	1	2018-09-12	2018-10-03	2018-09-30
2	2	2	1	2018-10-15	2018-11-05	2018-10-29
3	3	3	1	2018-12-20	2019-01-10	2019-01-08
4	4	4	1	2019-03-11	2019-04-01	2019-03-27
5	5	5	1	2019-04-20	2019-05-11	2019-05-05
6	6	6	1	2019-07-05	2019-07-26	2019-07-21
7	7	7	1	2019-09-10	2019-10-01	2019-09-25
8	8	8	1	2019-11-08	2019-11-29	2019-11-20
9	9	9	1	2020-01-15	2020-02-05	2020-02-03
10	10	10	1	2020-03-12	2020-04-02	2020-03-28
11	11	11	2	2020-05-14	2020-06-04	2020-05-28
12	12	12	2	2020-07-21	2020-08-11	2020-08-02
13	13	13	2	2020-09-25	2020-10-16	2020-10-15
14	14	4	1	2020-11-08	2020-11-29	2020-11-24
15	15	5	2	2021-01-03	2021-01-24	2021-01-19
16	16	3	2	2021-02-18	2021-03-11	2021-03-12
17	17	4	2	2021-04-27	2021-05-18	2021-05-20
18	18	5	2	2021-06-13	2021-07-04	2021-06-28
19	19	6	2	2021-08-15	2021-09-05	2021-09-03
20	20	7	2	2021-10-21	2021-11-11	2023-04-01
21	21	1	3	2021-11-29	2021-12-20	NULL
22	22	2	3	2022-01-10	2022-01-31	2022-01-25
23	23	3	3	2022-02-07	2022-02-28	2022-02-23
24	24	4	3	2022-03-11	2022-04-01	2022-03-28
25	25	5	3	2022-04-28	2022-05-19	2022-05-18
26	26	6	3	2022-06-22	2022-07-13	2022-07-08
27	27	7	3	2022-08-04	2022-08-25	2022-08-23
28	28	8	3	2022-09-13	2022-10-04	2022-09-28
29	29	9	3	2022-10-16	2022-11-06	2022-11-05
30	30	8	3	2022-11-21	2022-12-12	2022-12-05
31	31	9	4	2022-12-28	2023-01-18	NULL
32	32	1	4	2023-01-23	2023-02-13	NULL
33	33	10	4	2023-02-02	2023-02-23	2023-02-17
34	34	11	4	2023-03-01	2023-03-22	NULL
35	35	12	4	2023-03-10	2023-03-31	NULL
36	36	13	4	2023-03-15	2023-04-05	NULL
37	37	17	4	2023-03-25	2023-04-15	NULL
38	38	8	4	2023-03-30	2023-04-20	NULL
39	39	9	4	2023-03-26	2023-04-16	NULL
40	40	20	4	2023-03-28	2023-04-18	NULL

Q.9 . How do you delete the member Emily Miller and all her related records from the database?

```
DELETE FROM Borrow WHERE Member_ID = '5'
DELETE FROM Member WHERE Name = 'Emily Miller';
```

Member_ID	Name	Contact_Info	Join_Date
1	Alice Johnson	alice.johnson@email.com	2018-01-10
2	Bob Smith	bob.smith@email.com	2018-03-15
3	Carol Brown	carol.brown@email.com	2018-06-20
4	David Williams	david.williams@email.com	2018-09-18
5	Frank Davis	frank.davis@email.com	2019-05-25
6	Grace Wilson	grace.wilson@email.com	2019-08-15
7	Harry Garcia	harry.garcia@email.com	2019-11-27
8	Isla Thomas	isla.thomas@email.com	2020-03-04
9	Jack Martinez	jack.martinez@email.com	2020-07-01
10	Kate Anderson	kate.anderson@email.c...	2020-09-30
11	Luke Jackson	luke.jackson@email.com	2021-01-18
12	Mia White	mia.white@email.com	2021-04-27
13	Noah Harris	noah.harris@email.com	2021-07-13

We then check both the tables for Emily Miler and Member_ID 5, but cannot find any.

Q10. How do you add the following material to the database?

Title: New book

Date: 2020-08-01

Catalog: E-Books

Genre: Mystery & Thriller

Author: Lucas Pipi

```
SELECT Catalog_ID FROM Catalog
WHERE Catalog.Name = 'E-Books';
```

```
SELECT Genre_ID FROM Genre
WHERE Name = 'Mystery & Thriller';
```

```
INSERT INTO Material ("Material_ID", "Title", "Publication_Date", "Catalog_ID",
"Genre_ID")
VALUES (32, 'New book', '2020-08-01', 3, 2);
```

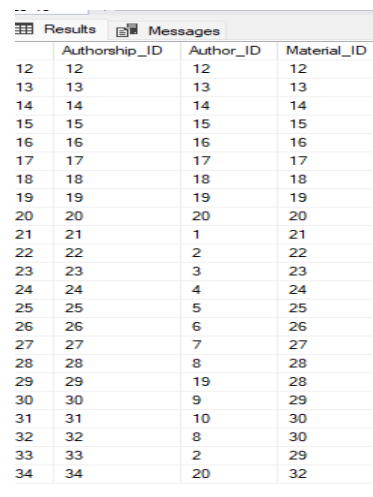
```
INSERT INTO Authorship("Authorship_ID", "Author_ID", "Material_ID")
Values(34, 20, 32)
```

We first find out the Catalog_ID and Genre_ID based on the Catalog and Genre. Then We add the new material into the table with the recent known ID's. After this we add this in the authorship table.



Material_ID	Title	Publication_Date	Catalog_ID	Genre_ID
10	The Hitchhiker's Guide to the Galaxy	1979-10-12	5	3
11	1984	1949-06-08	1	5
12	Animal Farm	1945-08-17	2	5
13	The Haunting of Hill House	1959-10-17	3	4
14	Brave New World	1932-08-01	4	5
15	The Chronicles of Narnia: The Lion, the Witch a...	1950-10-16	5	3
16	The Adventures of Huckleberry Finn	1884-12-10	6	1
17	The Catch-22	1961-10-11	7	1
18	The Picture of Dorian Gray	1890-07-01	8	1
19	The Call of Cthulhu	1928-02-01	9	4
20	Harry Potter and the Philosopher's Stone	1997-06-26	10	3
21	Frankenstein	1818-01-01	6	4
22	A Tale of Two Cities	1859-04-30	7	1
23	The Iliad	1750-01-01	8	6
24	The Odyssey	1725-01-01	9	6
25	The Brothers Karamazov	1880-01-01	10	1
26	The Divine Comedy	1320-01-01	6	6
27	The Grapes of Wrath	1939-04-14	7	1
28	The Old Man and the Sea	1952-09-01	8	1
29	The Count of Monte Cristo	1844-01-01	9	1
30	A Midsummer Night's Dream	1596-01-01	10	7
31	The Tricky Book	1888-01-01	10	7
32	New book	2020-08-01	3	2

Material



Authorship_ID	Author_ID	Material_ID
12	12	12
13	13	13
14	14	14
15	15	15
16	16	16
17	17	17
18	18	18
19	19	19
20	20	20
21	21	1
22	22	2
23	23	3
24	24	4
25	25	5
26	26	6
27	27	7
28	28	8
29	29	19
30	30	9
31	31	10
32	32	8
33	33	2
34	34	20

Authorship

We can see the addition of the new records in each table.

Advanced Design (Section 4.2)

1. Alert staff about overdue materials on a daily-basis?

To add the feature of alerting staff about overdue materials daily, you can create a stored procedure that checks the Borrow table for any records where the Return_Date is NULL (i.e., the materials have not been returned) and the Due_Date is less than the current date. The stored procedure can then send an alert to the staff members responsible for handling overdue materials.

Here's an example of a stored procedure that can be used for this purpose:

```
CREATE PROCEDURE AlertOverdueMaterials()
BEGIN DECLARE today DATE;
SET today = DATE(NOW());
SELECT m.Title, b.Borrow_Date, b.Due_Date, s.Name, s.Contact_Info FROM Borrow b
JOIN Material m ON m.Material_ID = b.Material_ID
JOIN Staff s ON s.Staff_ID = b.Staff_ID
WHERE b.Return_Date IS NULL AND b.Due_Date < today;
END
```

This stored procedure selects the Title of the overdue material, the date it was borrowed, the due date, and the name and contact information of the staff member responsible for the material. We can also add the member name if join the member table, so we can customize the SELECT statement to include additional information as needed. To run this stored procedure daily, we can set up a scheduled task using a tool like cron or Windows Task Scheduler.

2. Automatically deactivate the membership based on the member's overdue occurrence (>= three times). And reactivate the membership once the member pays the overdue fee

To automatically deactivate the membership based on the member's overdue occurrence (>= three times), We should create a trigger that checks the number of overdue occurrences for a member whenever a new borrowing record is added or an existing one is updated. If the number of overdue occurrences is greater than or equal to three, then the trigger can update the Member table to set the IsActive flag to false. Likewise, to reactivate the membership once the member pays the overdue fee, you can create another trigger that updates the IsActive flag to true whenever a payment record is added or updated for a member.

First trigger:

```
CREATE TRIGGER DeactivateMembership
AFTER INSERT OR UPDATE ON Borrow
FOR EACH ROW
BEGIN
    DECLARE overdue_count INT;
    SELECT COUNT(*) INTO overdue_count FROM Borrow
    WHERE Member_ID = NEW.Member_ID AND Due_Date < CURRENT_DATE AND Return_Date IS NULL;
    IF overdue_count >= 3 THEN
        UPDATE Member SET IsActive = false WHERE Member_ID = NEW.Member_ID;
    END IF;
END;
```

Second trigger:

```
CREATE TRIGGER ReactivateMembership
AFTER INSERT OR UPDATE ON Payment
FOR EACH ROW
BEGIN
    UPDATE Member SET IsActive = true WHERE Member_ID = NEW.Member_ID;
END;
```

these triggers assume the existence of a Payment table to track payments made by members for overdue fees. Thus we would need to create this table and add appropriate constraints and triggers to keep track of payments and update the IsActive flag accordingly.

Conclusion

Thus, we have implemented a Library management system with features for managing Materials, Borrows, Members, Staffs and Reporting Analytics using MS SQL. This can help Libraries help track all their items and would increase the efficiency of the library.

References

1. <https://www.w3schools.com/sql/>
2. <https://www.sqlservertutorial.net/>
3. https://www.tutorialspoint.com/ms_sql_server/index.htm
4. <https://www.sqlshack.com/sql-server-stored-procedures-for-beginners/>
5. <https://www.simplilearn.com/tutorials/sql-tutorial/stored-procedure-in-sql>
6. <https://learn.microsoft.com/en-us/sql/t-sql/statements/create-trigger-transact-sql?view=sql-server-ver16>
7. <https://www.c-sharpcorner.com/UploadFile/63f5c2/triggers-in-sql-server/>