

Library Management System SQL Project

- 1) Creating the database for the Library Management System named as Library.

```
▶ Run □ Cancel ⚙ Disconnect ☰ Change | Database: |
1   CREATE DATABASE Library;
```

Messages

12:06:57	Started executing query at Line 1 Commands completed successfully. Total execution time: 00:00:00.428
----------	---

- 2) Using the Library Database for performing tasks related to the Library Management System.

```
▶ Run □ Cancel ⚙ Disconnect ☰ Change | Database: Library ▾
2
3   USE Library;
```

Messages

12:09:28	Started executing query at Line 3 Commands completed successfully. Total execution time: 00:00:00.003
----------	---

3) Creating the Authors Table.

The screenshot shows a SQL query editor interface. At the top, there are buttons for 'Run', 'Cancel', 'Disconnect', 'Change', and a dropdown menu for 'Database' set to 'Library'. The main area contains a numbered SQL script:

```
5  -- Creating the Authors Table
6  CREATE TABLE Authors (
7      AuthorID INT PRIMARY KEY IDENTITY(1,1),
8      Name NVARCHAR(255) NOT NULL,
9      BirthDate DATE,
10     Nationality NVARCHAR(100)
11 );
12
13
```

The code is syntax-highlighted, with 'CREATE TABLE' in blue, column definitions in black, and data types in green. Lines 6 through 10 are highlighted with a light blue background.

Messages

12:12:39 Started executing query at Line 6
Commands completed successfully.
Total execution time: 00:00:00.037

4) Inserting the values into the Authors Table.

In the **Authors** table, the **AuthorID** column is defined as an **IDENTITY** column. Therefore, to explicitly insert values into this column, we must temporarily enable identity insert using **SET IDENTITY_INSERT ON**. After inserting the values, we should turn it off again using **SET IDENTITY_INSERT OFF**, as demonstrated in the screenshot below.

▶ Run □ Cancel ⚙ Disconnect ⚙ Change | Database: Library ▾ | ⚙ Est

```
13 SET IDENTITY_INSERT Authors ON;
14
15 -- Inserting the values into the Authors Table
16 INSERT INTO Authors (AuthorID, Name, BirthDate, Nationality) VALUES
17 (1, 'F. Scott Fitzgerald', '1896-09-24', 'American'),
18 (2, 'Jane Austen', '1775-12-16', 'British'),
19 (3, 'George Orwell', '1903-06-25', 'British'),
20 (4, 'J.K. Rowling', '1965-07-31', 'British'),
21 (5, 'J.R.R. Tolkien', '1892-01-03', 'British'),
22 (6, 'Ernest Hemingway', '1899-07-21', 'American'),
23 (7, 'Agatha Christie', '1890-09-15', 'British'),
24 (8, 'Leo Tolstoy', '1828-09-09', 'Russian'),
25 (9, 'Mark Twain', '1835-11-30', 'American'),
26 (10, 'Charles Dickens', '1812-02-07', 'British'),
27 (11, 'Stephen King', '1947-09-21', 'American'),
28 (12, 'Isaac Asimov', '1920-01-02', 'American'),
29 (13, 'Harper Lee', '1926-04-28', 'American'),
30 (14, 'Emily Brontë', '1818-07-30', 'British'),
31 (15, 'Mary Shelley', '1797-08-30', 'British'),
32 (16, 'Oscar Wilde', '1854-10-16', 'Irish'),
33 (17, 'Gabriel Garcia Marquez', '1927-03-06', 'Colombian'),
34 (18, 'Arthur Conan Doyle', '1859-05-22', 'British'),
35 (19, 'Franz Kafka', '1883-07-03', 'Austrian'),
36 (20, 'Virginia Woolf', '1882-01-25', 'British');
37
38 SET IDENTITY_INSERT Authors OFF;
39
```

Messages

12:17:03 Started executing query at Line 13
(20 rows affected)
Total execution time: 00:00:00.016

5) Showing all the records of the Authors Table.

Run Cancel ⚙ Disconnect ⚙ Change Database: Library Est

```
39
40 -- Showing all the records of the Authors Table
41 SELECT * from Authors;
42
```

Results Messages

	AuthorID	Name	BirthDate	Nationality
1	1	F. Scott Fitzgerald	1896-09-24	American
2	2	Jane Austen	1775-12-16	British
3	3	George Orwell	1903-06-25	British
4	4	J.K. Rowling	1965-07-31	British
5	5	J.R.R. Tolkien	1892-01-03	British
6	6	Ernest Hemingway	1899-07-21	American
7	7	Agatha Christie	1890-09-15	British
8	8	Leo Tolstoy	1828-09-09	Russian
9	9	Mark Twain	1835-11-30	American
10	10	Charles Dickens	1812-02-07	British
11	11	Stephen King	1947-09-21	American
12	12	Isaac Asimov	1920-01-02	American
13	13	Harper Lee	1926-04-28	American
14	14	Emily Brontë	1818-07-30	British
15	15	Mary Shelley	1797-08-30	British
16	16	Oscar Wilde	1854-10-16	Irish
17	17	Gabriel Garcia Mar...	1927-03-06	Colombian
18	18	Arthur Conan Doyle	1859-05-22	British
19	19	Franz Kafka	1883-07-03	Austrian
20	20	Virginia Woolf	1882-01-25	British

6) Creating the Books Table.

The screenshot shows a SQL query editor interface. At the top, there are buttons for Run, Cancel, Disconnect, Change, and a dropdown menu for the Database set to 'Library'. Below the interface, the SQL code for creating the Books table is displayed:

```
43 -- Creating the Books Table
44 CREATE TABLE Books (
45     BookID INT PRIMARY KEY IDENTITY(1,1),
46     Title NVARCHAR(255) NOT NULL,
47     AuthorID INT FOREIGN KEY REFERENCES Authors(AuthorID),
48     ISBN NVARCHAR(20) UNIQUE NOT NULL,
49     Availability BIT NOT NULL DEFAULT 1
50 );
```

Messages

12:25:09 Started executing query at Line 44
Commands completed successfully.
Total execution time: 00:00:00.042

7) Inserting the values into the Books Table.

In the **Books** table, the **BookID** column is defined as an **IDENTITY** column. Therefore, to explicitly insert values into this column, we must temporarily enable identity insert using **SET IDENTITY_INSERT ON**. After inserting the values, we should turn it off again using **SET IDENTITY_INSERT OFF**, as demonstrated in the screenshot below.

▶ Run □ Cancel ⚙ Disconnect ⚙ Change | Database: Library ▾ | ⏴ Estimated

```
52 SET IDENTITY_INSERT Books ON;
53
54 -- Inserting the values into the Books Table
55 INSERT INTO Books (BookID, Title, AuthorID, ISBN, Availability) VALUES
56 (1, 'The Great Gatsby', 1, '9780743273565', 1),
57 (2, 'Pride and Prejudice', 2, '9780141439518', 1),
58 (3, '1984', 3, '9780451524935', 0),
59 (4, 'Harry Potter and the Sorcerer''s Stone', 4, '9780439554930', 1),
60 (5, 'The Hobbit', 5, '9780547928227', 1),
61 (6, 'The Old Man and the Sea', 6, '9780684801223', 0),
62 (7, 'Murder on the Orient Express', 7, '9780062693662', 1),
63 (8, 'War and Peace', 8, '9780199232765', 1),
64 (9, 'Adventures of Huckleberry Finn', 9, '9780486280615', 0),
65 (10, 'Oliver Twist', 10, '9780141439747', 1),
66 (11, 'The Shining', 11, '9780307743657', 1),
67 (12, 'Foundation', 12, '9780553293357', 1),
68 (13, 'To Kill a Mockingbird', 13, '9780061120084', 0),
69 (14, 'Wuthering Heights', 14, '9780141439556', 1),
70 (15, 'Frankenstein', 15, '9780486282114', 1),
71 (16, 'The Picture of Dorian Gray', 16, '9780141439570', 1),
72 (17, 'One Hundred Years of Solitude', 17, '9780060883287', 1),
73 (18, 'The Adventures of Sherlock Holmes', 18, '9781508475316', 0),
74 (19, 'The Metamorphosis', 19, '9780553213690', 1),
75 (20, 'Mrs Dalloway', 20, '9780156628709', 1);
76
77 SET IDENTITY_INSERT Books OFF;
```

Messages

12:28:25 [Started executing query at Line 52](#)
(20 rows affected)
Total execution time: 00:00:00.084

8) Showing all the records of the Books Table.

Run Cancel ⚙ Change Database: Library Estimated Plan Enable Actual Plan ✓ P

78
79 -- Showing all the records of the Books Table
80 SELECT * from Books;

Results Messages

	BookID	Title	AuthorID	ISBN	Availability
1	1	The Great Gatsby	1	9780743273565	1
2	2	Pride and Prejudice	2	9780141439518	1
3	3	1984	3	9780451524935	0
4	4	Harry Potter and the Sorcerer's Stone	4	9780439554930	1
5	5	The Hobbit	5	9780547928227	1
6	6	The Old Man and the Sea	6	9780684801223	0
7	7	Murder on the Orient Express	7	9780062693662	1
8	8	War and Peace	8	9780199232765	1
9	9	Adventures of Huckleberry Finn	9	9780486280615	0
10	10	Oliver Twist	10	9780141439747	1
11	11	The Shining	11	9780307743657	1
12	12	Foundation	12	9780553293357	1
13	13	To Kill a Mockingbird	13	9780061120084	0
14	14	Wuthering Heights	14	9780141439556	1
15	15	Frankenstein	15	9780486282114	1
16	16	The Picture of Dorian Gray	16	9780141439570	1
17	17	One Hundred Years of Solitude	17	9780060883287	1
18	18	The Adventures of Sherlock Holmes	18	9781508475316	0
19	19	The Metamorphosis	19	9780553213690	1
20	20	Mrs Dalloway	20	9780156628709	1

9) Creating the Patrons Table.

The screenshot shows a SQL query editor interface. At the top, there are buttons for Run, Cancel, Disconnect, Change, and a dropdown for Database set to Library. Below the toolbar, the text 'o1' is displayed. The main area contains a SQL script:

```
82  -- Creating the Patrons Table
83  CREATE TABLE Patrons (
84      PatronID INT PRIMARY KEY IDENTITY(1,1),
85      Name NVARCHAR(255),
86      ContactInfo NVARCHAR(255)
87 );
```

The code is syntax-highlighted, with 'CREATE TABLE' in blue, column definitions in green, and the primary key definition in orange.

Messages

12:32:02

Started executing query at Line 83

Commands completed successfully.

Total execution time: 00:00:00.034

10) Inserting the values into the Patrons Table.

In the **Patrons** table, the **PatronID** column is defined as an **IDENTITY** column. Therefore, to explicitly insert values into this column, we must temporarily enable identity insert using **SET IDENTITY_INSERT ON**. After inserting the values, we should turn it off again using **SET IDENTITY_INSERT OFF**, as demonstrated in the screenshot below.

▶ Run □ Cancel ⚙ Disconnect ⚙ Change | Database: Library

```
88
89 SET IDENTITY_INSERT Patrons ON;
90
91 -- Inserting the values into the Patrons Table
92 INSERT INTO Patrons (PatronID, Name, ContactInfo) VALUES
93 (1, 'Alice Johnson', 'alice.j@gmail.com'),
94 (2, 'Bob Smith', 'bob.smith@yahoo.com'),
95 (3, 'Carol Davis', 'carol.davis@hotmail.com'),
96 (4, 'David Brown', 'david.brown@gmail.com'),
97 (5, 'Eve Thompson', 'eve.t@gmail.com'),
98 (6, 'Frank Harris', 'frank.h@hotmail.com'),
99 (7, 'Grace Lee', 'grace.lee@yahoo.com'),
100 (8, 'Hank Moore', 'hank.moore@gmail.com'),
101 (9, 'Ivy Clark', 'ivy.clark@gmail.com'),
102 (10, 'Jack Lewis', 'jack.l@hotmail.com'),
103 (11, 'Kathy Young', 'kathy.young@yahoo.com'),
104 (12, 'Leo Allen', 'leo.a@gmail.com'),
105 (13, 'Mia Hall', 'mia.h@gmail.com'),
106 (14, 'Noah Scott', 'noah.s@hotmail.com'),
107 (15, 'Olivia Adams', 'olivia.adams@yahoo.com'),
108 (16, 'Paul Baker', 'paul.b@gmail.com'),
109 (17, 'Quinn Campbell', 'quinn.c@hotmail.com'),
110 (18, 'Ruby Mitchell', 'ruby.m@gmail.com'),
111 (19, 'Sam Carter', 'sam.c@gmail.com'),
112 (20, 'Tina Ross', 'tina.ross@yahoo.com');
113
114 SET IDENTITY_INSERT Patrons OFF;
```

Messages

12:35:39 Started executing query at Line 89
(20 rows affected)
Total execution time: 00:00:00.037

11) Showing all the records of the Patrons Table.

▶ Run □ Cancel ⚙ Disconnect ☰ Change Database: Library

115

116 -- Showing all the records of the Patrons Table

117 `SELECT * from Patrons;`

Results Messages

	PatronID	Name	ContactInfo
1	1	Alice Johnson	alice.j@gmail.com
2	2	Bob Smith	bob.smith@yahoo.com
3	3	Carol Davis	carol.davis@hotmail.com
4	4	David Brown	david.brown@gmail.com
5	5	Eve Thompson	eve.t@gmail.com
6	6	Frank Harris	frank.h@hotmail.com
7	7	Grace Lee	grace.lee@yahoo.com
8	8	Hank Moore	hank.moore@gmail.com
9	9	Ivy Clark	ivy.clark@gmail.com
10	10	Jack Lewis	jack.l@hotmail.com
11	11	Kathy Young	kathy.young@yahoo.com
12	12	Leo Allen	leo.a@gmail.com
13	13	Mia Hall	mia.h@gmail.com
14	14	Noah Scott	noah.s@hotmail.com
15	15	Olivia Adams	olivia.adams@yahoo.com
16	16	Paul Baker	paul.b@gmail.com
17	17	Quinn Campbell	quinn.c@hotmail.com
18	18	Ruby Mitchell	ruby.m@gmail.com
19	19	Sam Carter	sam.c@gmail.com
20	20	Tina Ross	tina.ross@yahoo.com

12) Creating the BorrowHistory Table.

The screenshot shows a SQL query editor interface. At the top, there are buttons for Run, Cancel, Disconnect, Change, and a dropdown menu for Database set to Library. The main area contains the following SQL code:

```
118  
119 -- Creating the BorrowHistory Table  
120 CREATE TABLE BorrowHistory (  
121     BorrowID INT PRIMARY KEY IDENTITY(1,1),  
122     BookID INT FOREIGN KEY REFERENCES Books(BookID),  
123     PatronID INT FOREIGN KEY REFERENCES Patrons(PatronID),  
124     BorrowDate DATETIME DEFAULT GETDATE(),  
125     ReturnDate DATETIME NULL  
126 );
```

Messages

12:38:58 Started executing query at Line 120
Commands completed successfully.
Total execution time: 00:00:00.043

13) Inserting the values into the BorrowHistory Table.

In the **BorrowHistory** table, the **BorrowID** column is defined as an **IDENTITY** column. Therefore, to explicitly insert values into this column, we must temporarily enable identity insert using `SET IDENTITY_INSERT ON`. After inserting the values, we should turn it off again using `SET IDENTITY_INSERT OFF`, as demonstrated in the screenshot below.

▶ Run □ Cancel ⚙ Disconnect ⚙ Change Database: Library ✓ Estimated Plan ⚙ Enable Ac

```
127
128 SET IDENTITY_INSERT BorrowHistory ON;
129
130 -- Inserting the values into the BorrowHistory Table
131 INSERT INTO BorrowHistory (BorrowID, BookID, PatronID, BorrowDate, ReturnDate) VALUES
132 (1, 1, 1, '2024-01-15', '2024-01-25'),
133 (2, 2, 2, '2024-01-20', '2024-01-28'),
134 (3, 3, 3, '2024-01-25', NULL),
135 (4, 4, 4, '2024-01-28', '2024-02-05'),
136 (5, 5, 5, '2024-02-01', '2024-02-10'),
137 (6, 6, 6, '2024-02-10', NULL),
138 (7, 7, 7, '2024-02-15', '2024-02-25'),
139 (8, 8, 8, '2024-02-18', '2024-02-28'),
140 (9, 9, 9, '2024-03-01', NULL),
141 (10, 10, 10, '2024-03-05', '2024-03-15'),
142 (11, 11, 11, '2024-03-08', '2024-03-20'),
143 (12, 12, 12, '2024-03-10', '2024-03-20'),
144 (13, 13, 13, '2024-03-15', NULL),
145 (14, 14, 14, '2024-03-20', '2024-03-30'),
146 (15, 15, 15, '2024-03-22', '2024-04-01'),
147 (16, 16, 16, '2024-03-25', '2024-04-05'),
148 (17, 17, 17, '2024-04-01', NULL),
149 (18, 18, 18, '2024-04-05', '2024-04-15'),
150 (19, 19, 19, '2024-04-10', '2024-04-20'),
151 (20, 20, 20, '2024-04-12', NULL);
152
153 SET IDENTITY_INSERT BorrowHistory OFF;
```

Messages

12:41:31 Started executing query at Line 128
(20 rows affected)
Total execution time: 00:00:00.076

14) Showing all the records of the BorrowHistory Table.

Run Cancel ⚙ Disconnect ⚙ Change Database: Library ↴ Estimated Plan ↴ Enable Actual Plan

154

155 -- Showing all the records of the BorrowHistory Table

156 `SELECT * from BorrowHistory;`

Results Messages

	BorrowID	BookID	PatronID	BorrowDate	ReturnDate
1	1	1	1	2024-01-15 00:00:00.000	2024-01-25 00:00:00.000
2	2	2	2	2024-01-20 00:00:00.000	2024-01-28 00:00:00.000
3	3	3	3	2024-01-25 00:00:00.000	NULL
4	4	4	4	2024-01-28 00:00:00.000	2024-02-05 00:00:00.000
5	5	5	5	2024-02-01 00:00:00.000	2024-02-10 00:00:00.000
6	6	6	6	2024-02-10 00:00:00.000	NULL
7	7	7	7	2024-02-15 00:00:00.000	202 Results grid :00:00.000
8	8	8	8	2024-02-18 00:00:00.000	2024-02-28 00:00:00.000
9	9	9	9	2024-03-01 00:00:00.000	NULL
10	10	10	10	2024-03-05 00:00:00.000	2024-03-15 00:00:00.000
11	11	11	11	2024-03-08 00:00:00.000	2024-03-20 00:00:00.000
12	12	12	12	2024-03-10 00:00:00.000	2024-03-20 00:00:00.000
13	13	13	13	2024-03-15 00:00:00.000	NULL
14	14	14	14	2024-03-20 00:00:00.000	2024-03-30 00:00:00.000
15	15	15	15	2024-03-22 00:00:00.000	2024-04-01 00:00:00.000
16	16	16	16	2024-03-25 00:00:00.000	2024-04-05 00:00:00.000
17	17	17	17	2024-04-01 00:00:00.000	NULL
18	18	18	18	2024-04-05 00:00:00.000	2024-04-15 00:00:00.000
19	19	19	19	2024-04-10 00:00:00.000	2024-04-20 00:00:00.000
20	20	20	20	2024-04-12 00:00:00.000	NULL

- 15) Adding the new book record into the data.

```
▶ Run □ Cancel ⌘ Disconnect ⌘ Change Database: Library ▾ Estimated Pla
158 -- Adding the new book record into the data
159 INSERT INTO Books (Title, AuthorID, ISBN, Availability)
160 VALUES ('The Silmarillion', 5, '9780618391110', 1);
161
162
```

Messages

12:54:10 Started executing query at Line 159
(1 row affected)
Total execution time: 00:00:00.037

- 16) Updating the Books details in the data.

```
▶ Run □ Cancel ⌘ Disconnect ⌘ Change Database: Library ▾ Estimated Pla
162 -- Updating the book details in the data
163 UPDATE Books
164 SET Title = 'The Chronicles of Middle-earth',
165     AuthorID = 5,
166     ISBN = '9780987654321'
167 WHERE BookID = 10;
168
```

Messages

12:57:35 Started executing query at Line 163
(1 row affected)
Total execution time: 00:00:00.042

- 17) Showing the records where we added the book as well as updated the book details.

The screenshot shows a SQL query window with the following details:

- Toolbar:** Run, Cancel, Disconnect, Change, Database dropdown set to Library, Estimated Plan, Enable Actual Plan, and a Print icon.
- Text Area:**

```
168
169  SELECT * FROM Books;
170
```
- Results Tab:** The tab is selected and shows the output of the query.
- Table Output:** A grid of 21 rows representing book records. The columns are BookID, Title, AuthorID, ISBN, and Availability.

	BookID	Title	AuthorID	ISBN	Availability
1	1	The Great Gatsby	1	9780743273565	1
2	2	Pride and Prejudice	2	9780141439518	1
3	3	1984	3	9780451524935	0
4	4	Harry Potter and the Sorcerer's Stone	4	9780439554930	1
5	5	The Hobbit	5	9780547928227	1
6	6	The Old Man and the Sea	6	9780684801223	0
7	7	Murder on the Orient Express	7	9780062693662	1
8	8	War and Peace	8	9780199232765	1
9	9	Adventures of Huckleberry Finn	9	9780486280615	0
10	10	The Chronicles of Middle-earth	5	9780987654321	1
11	11	The Shining	11	9780307743657	1
12	12	Foundation	12	9780553293357	1
13	13	To Kill a Mockingbird	13	9780061120084	0
14	14	Wuthering Heights	14	9780141439556	1
15	15	Frankenstein	15	9780486282114	1
16	16	The Picture of Dorian Gray	16	9780141439570	1
17	17	One Hundred Years of Solitude	17	9780060883287	1
18	18	The Adventures of Sherlock Holmes	18	9781508475316	0
19	19	The Metamorphosis	19	9780553213690	1
20	20	Mrs Dalloway	20	9780156628709	1
21	22	The Silmarillion	5	9780618391110	1

- 18) Showing below the examples of when the book is borrowed and returned.

Marking the Book as Borrowed or Returned:-

Borrowed: Set Availability = 0 (not available)

```
▶ Run □ Cancel ⚙ Disconnect ☰ Change | Database: Library
168
169   SELECT * FROM Books;
170
171   -- Marking the book as borrowed
172   UPDATE Books
173     SET Availability = 0
174   WHERE BookID = 3;
175
```

Messages

13:05:57 Started executing query at Line 172
(1 row affected)
Total execution time: 00:00:00.034

Below Screenshot showing that BookID = 3 is borrowed as Availability = 0.

```
▶ Run □ Cancel ⚙ Disconnect ☰ Change | Database: Library | Estimated Plan | Enable Actual Plan | Par
167   WHERE BookID = 1,
168
169   SELECT * FROM Books;
170
171   -- Marking the book as borrowed
172   UPDATE Books
173     SET Availability = 0
174   WHERE BookID = 3;
175
```

Results **Messages**

	BookID	Title	AuthorID	ISBN	Availability
1	1	The Great Gatsby	1	9780743273565	1
2	2	Pride and Prejudice	2	9780141439518	1
3	3	1984	3	9780451524935	0
4	4	Harry Potter and the Sorcerer's Stone	4	9780439554930	1
5	5	The Hobbit	5	9780547928227	1

Returned: Set Availability = 1 (available)

▶ Run □ Cancel ⚙ Disconnect ☰ Change | Database: Library

```
176    -- Marking the book as returned
177    UPDATE Books
178    SET Availability = 1
179    WHERE BookID = 3;
180
181    SELECT * FROM Books;
```

Messages

13:09:05 Started executing query at Line 177
(1 row affected)
Total execution time: 00:00:00.016

Below Screenshot showing that BookID = 3 is borrowed as Availability = 1.

▶ Run □ Cancel ⚙ Disconnect ☰ Change | Database: Library | Estimated Plan | Enable Actual Plan ✓

```
176    -- Marking the book as returned
177    UPDATE Books
178    SET Availability = 1
179    WHERE BookID = 3;
180
181    SELECT * FROM Books;
```

Results grid

	BookID	Title	AuthorID	ISBN	Availability
1	1	The Great Gatsby	1	9780743273565	1
2	2	Pride and Prejudice	2	9780141439518	1
3	3	1984	3	9780451524935	1
4	4	Harry Potter and the Sorcerer's Stone	4	9780439554930	1
5	5	The Hobbit	5	9780547928227	1

19) Searching the Books by Title, Author Name, or ISBN.

The screenshot shows a SQL query being run against a 'Library' database. The query is designed to search for books based on title, author name, or ISBN. It uses a combination of LIKE and OR clauses to find matches. The results table shows one book entry: 'Harry Potter and the Sorcerer's Stone' by J.K. Rowling, with ISBN 9780439554930.

```

183  -- Searching the Books by Title, Author, or ISBN
184  SELECT b.BookID, b.Title, a.Name AS Author, b.ISBN, b.Availability
185  FROM Books b
186  JOIN Authors a ON b.AuthorID = a.AuthorID
187  WHERE b.Title LIKE '%Harry%'
188  OR a.Name LIKE '%Rowling%'
189  OR b.ISBN = '9780439554930';
190

```

	BookID	Title	Author	ISBN	Availability
1	4	Harry Potter and the Sorcerer's Stone	J.K. Rowling	9780439554930	1

20) Tracking the Borrowing and Returning History of the Books.

The screenshot shows a SQL query being run against a 'Library' database. The query tracks borrowing history, joining three tables: BorrowHistory, Books, and Patrons. It orders the results by borrow date in descending order. The results table lists 20 borrow entries, each with a BorrowID, Title, Patron name, BorrowDate, and ReturnDate.

```

190
191  -- Tracking the Borrowing and Returning History of the Books
192  SELECT bh.BorrowID, b.Title, p.Name AS Patron, bh.BorrowDate, bh.ReturnDate
193  FROM BorrowHistory bh
194  JOIN Books b ON bh.BookID = b.BookID
195  JOIN Patrons p ON bh.PatronID = p.PatronID
196  ORDER BY bh.BorrowDate DESC;
197

```

	BorrowID	Title	Patron	BorrowDate	ReturnDate
1	20	Mrs Dalloway	Tina Ross	2024-04-12 00:00:00.000	NULL
2	19	The Metamorphosis	Sam Carter	2024-04-10 00:00:00.000	2024-04-20 00:00:00.000
3	18	The Adventures of Sherlock Holmes	Ruby Mitchell	2024-04-05 00:00:00.000	2024-04-15 00:00:00.000
4	17	One Hundred Years of Solitude	Quinn Campbell	2024-04-01 00:00:00.000	NULL
5	16	The Picture of Dorian Gray	Paul Baker	2024-03-25 00:00:00.000	2024-04-05 00:00:00.000
6	15	Frankenstein	Olivia Adams	2024-03-22 00:00:00.000	2024-04-01 00:00:00.000
7	14	Wuthering Heights	Noah Scott	2024-03-20 00:00:00.000	2024-03-30 00:00:00.000
8	13	To Kill a Mockingbird	Mia Hall	2024-03-15 00:00:00.000	NULL
9	12	Foundation	Leo Allen	2024-03-10 00:00:00.000	2024-03-20 00:00:00.000
10	11	The Shining	Kathy Young	2024-03-08 00:00:00.000	2024-03-20 00:00:00.000
11	10	The Chronicles of Middle-earth	Jack Lewis	2024-03-05 00:00:00.000	2024-03-15 00:00:00.000
12	9	Adventures of Huckleberry Finn	Ivy Clark	2024-03-01 00:00:00.000	NULL
13	8	War and Peace	Hank Moore	2024-02-18 00:00:00.000	2024-02-28 00:00:00.000
14	7	Murder on the Orient Express	Grace Lee	2024-02-15 00:00:00.000	2024-02-25 00:00:00.000
15	6	The Old Man and the Sea	Frank Harris	2024-02-10 00:00:00.000	NULL
16	5	The Hobbit	Eve Thompson	2024-02-01 00:00:00.000	2024-02-10 00:00:00.000
17	4	Harry Potter and the Sorcerer's Stone	David Brown	2024-01-28 00:00:00.000	2024-02-05 00:00:00.000
18	3	1984	Carol Davis	2024-01-25 00:00:00.000	NULL
19	2	Pride and Prejudice	Bob Smith	2024-01-20 00:00:00.000	2024-01-28 00:00:00.000
20	1	The Great Gatsby	Alice Johnson	2024-01-15 00:00:00.000	2024-01-25 00:00:00.000

- 21) Using Window Functions and CTEs: The following example displays the number of books borrowed by each patron, ordered by the most recent borrow date.

```

Run Cancel & Disconnect & Change Database: Library v Estimated Plan Enable Actual Plan Parse Enable SQLCMD To Notebook
198
199 -- Using Window Function and CTE:- Below Example shows the Count number of books borrowed by each patron, showing the most recent borro
200 WITH BorrowCounts AS ( SELECT
201     p.PatronID, p.Name,
202     COUNT(bh.BorrowID) OVER (PARTITION BY p.PatronID) AS TotalBorrows,
203     ROW_NUMBER() OVER (PARTITION BY p.PatronID ORDER BY bh.BorrowDate DESC) AS RecentBorrowRank,
204     bh.BorrowDate, b.Title
205     FROM BorrowHistory bh
206     JOIN Patrons p ON bh.PatronID = p.PatronID
207     JOIN Books b ON bh.BookID = b.BookID
208 )
209     SELECT PatronID, Name, Title, BorrowDate, RecentBorrowRank, TotalBorrows
210     FROM BorrowCounts WHERE RecentBorrowRank = 1;

```

Results Messages

Pride and Prejudice

	PatronID	Name	Title	BorrowDate	RecentBorrowRank	TotalBorrows
1	1	Alice Johnson	The Great Gatsby	2024-01-15 00:00:00.000	1	1
2	2	Bob Smith	Pride and Prejudice	2024-01-20 00:00:00.000	1	1
3	3	Carol Davis	1984	2024-01-25 00:00:00.000	1	1
4	4	David Brown	Harry Potter and the Sorcerer's Stone	2024-01-28 00:00:00.000	1	1
5	5	Eve Thompson	The Hobbit	2024-02-01 00:00:00.000	1	1
6	6	Frank Harris	The Old Man and the Sea	2024-02-10 00:00:00.000	1	1
7	7	Grace Lee	Murder on the Orient Express	2024-02-15 00:00:00.000	1	1
8	8	Hank Moore	War and Peace	2024-02-18 00:00:00.000	1	1
9	9	Ivy Clark	Adventures of Huckleberry Finn	2024-03-01 00:00:00.000	1	1
10	10	Jack Lewis	The Chronicles of Middle-earth	2024-03-05 00:00:00.000	1	1
11	11	Kathy Young	The Shining	2024-03-08 00:00:00.000	1	1
12	12	Leo Allen	Foundation	2024-03-10 00:00:00.000	1	1
13	13	Mia Hall	To Kill a Mockingbird	2024-03-15 00:00:00.000	1	1
14	14	Noah Scott	Wuthering Heights	2024-03-20 00:00:00.000	1	1
15	15	Olivia Adams	Frankenstein	2024-03-22 00:00:00.000	1	1
16	16	Paul Baker	The Picture of Dorian Gray	2024-03-25 00:00:00.000	1	1
17	17	Quinn Campbell	One Hundred Years of Solitude	2024-04-01 00:00:00.000	1	1
18	18	Ruby Mitchell	The Adventures of Sherlock Holmes	2024-04-05 00:00:00.000	1	1
19	19	Sam Carter	The Metamorphosis	2024-04-10 00:00:00.000	1	1
20	20	Tina Ross	Mrs Dalloway	2024-04-12 00:00:00.000	1	1

- 22) Identifying the TOP borrowers using Subquery.

Run Cancel Disconnect Change Database: Library

```
212
213 -- Identifying the Top Borrowers Using Subquery
214 SELECT TOP 5 p.Name, COUNT(bh.BorrowID) AS BorrowCount
215 FROM BorrowHistory bh
216 JOIN Patrons p ON bh.PatronID = p.PatronID
217 GROUP BY p.Name
218 ORDER BY BorrowCount DESC;
```

Results **Messages**

	Name	BorrowCount
1	Frank Harris	1
2	Eve Thompson	1
3	David Brown	1
4	Carol Davis	1
5	Bob Smith	1

- 23) Creating a backup of the Books table using the SELECT INTO clause.

Run Cancel Disconnect Change Database: Library

```
219
220 -- Creating a backup of the Books table using the SELECT INTO clause
221 SELECT * INTO BooksBackup FROM Books;
```

Messages

13:31:40 Started executing query at Line 221
(21 rows affected)
Total execution time: 00:00:00.063

24) Showing the backup of all the records for Books Table.

```
222  
223 -- Showing the backup of all the records for Books Table  
224 | SELECT * FROM BooksBackup;
```

Results Messages

	BookID	Title	AuthorID	ISBN	Availability
1	1	The Great Gatsby	1	9780743273565	1
2	2	Pride and Prejudice	2	9780141439518	1
3	3	1984	3	9780451524955	1
4	4	Harry Potter and the Sorcerer's Stone	4	9780439554930	1
5	5	The Hobbit	5	9780547928227	1
6	6	The Old Man and the Sea	6	9780684801223	0
7	7	Murder on the Orient Express	7	9780062693662	1
8	8	War and Peace	8	9780199232765	1
9	9	Adventures of Huckleberry Finn	9	9780486280615	0
10	10	The Chronicles of Middle-earth	5	9780987654321	1
11	11	The Shining	11	9780307743657	1
12	12	Foundation	12	9780553293357	1
13	13	To Kill a Mockingbird	13	9780061120084	0
14	14	Wuthering Heights	14	9780141439556	1
15	15	Frankenstein	15	9780486282114	1
16	16	The Picture of Dorian Gray	16	9780141439570	1
17	17	One Hundred Years of Solitude	17	9780060883287	1
18	18	The Adventures of Sherlock Holmes	18	9781508475316	0
19	19	The Metamorphosis	19	9780553213690	1
20	20	Mrs Dalloway	20	9780156628709	1
21	22	The Silmarillion	5	9780618391110	1

- 25) Implementing the MERGE command to streamline borrow and return operations while maintaining data integrity.

A) Borrow Operation:-

```
Run Cancel Disconnect Change Database: Library Estimated Plan Enable Actual Plan Parse Enable S
220
227 -- Implementing the MERGE command to streamline borrow and return operations while maintaining data integrity
228 -- a) Borrow Operation :-
229 DECLARE @BookID INT = 3,
230      @PatronID INT = 1,
231      @BorrowDate DATE = GETDATE();
232
233 MERGE BorrowHistory AS target
234 USING (
235     SELECT @BookID AS BookID, @PatronID AS PatronID
236 ) AS source
237 ON target.BookID = source.BookID
238 AND target.ReturnDate IS NULL -- Book is currently borrowed
239
240 WHEN NOT MATCHED BY TARGET
241     AND EXISTS (SELECT 1 FROM Books WHERE BookID = @BookID AND Availability = 1)
242 THEN
243     INSERT (BookID, PatronID, BorrowDate, ReturnDate)
244     VALUES (@BookID, @PatronID, @BorrowDate, NULL);
245
246 -- Updating the availability after successful insert
247 IF @@ROWCOUNT > 0
248 BEGIN
249     UPDATE Books
250     SET Availability = 0
251     WHERE BookID = @BookID;
252
253     PRINT 'Book borrowed successfully.';
254 END
255 ELSE
256 BEGIN
257     PRINT 'Book is currently borrowed and not available.';
258 END
259
```

Messages

13:47:22 Started executing query at Line 229
(0 rows affected)
Book is currently borrowed and not available.
Total execution time: 00:00:00.091

B) Return Operation:-

```
Run Cancel Disconnect Change Database: Library Estimated Pla
261
262
263 -- B) Return Operation :-
264 DECLARE @BookID INT = 3,
265      @ReturnDate DATE = GETDATE();
266
267 -- Update BorrowHistory with MERGE
268 MERGE BorrowHistory AS target
269 USING (SELECT @BookID AS BookID) AS source
270 ON target.BookID = source.BookID AND target.ReturnDate IS NULL
271
272 WHEN MATCHED THEN
273     UPDATE SET ReturnDate = @ReturnDate;
274
275 -- If the MERGE matched and updated a record, set book availability to 1
276 IF @@ROWCOUNT > 0
277 BEGIN
278     UPDATE Books
279     SET Availability = 1
280     WHERE BookID = @BookID;
281
282     PRINT 'Book return processed successfully.';
283 END
284 ELSE
285 BEGIN
286     PRINT 'No active borrow record found for this book.';
287 END
288
```

Messages

13:54:17 Started executing query at Line 264
(1 row affected)
(1 row affected)
Book return processed successfully.
Total execution time: 00:00:00.056