**Creating Tables**

**-- Create Category\_Details Table**

CREATE TABLE Category\_Details (

Category\_ID INT PRIMARY KEY,

Category\_Name VARCHAR(50)

);

-- **Create Book\_Details Table**

CREATE TABLE Book\_Details (

ISBN INT PRIMARY KEY,

Title VARCHAR(100),

Publication\_Year INT,

Language VARCHAR(50),

Category\_ID INT,

No\_Of\_Copies\_Actual INT,

No\_Of\_Copies\_Current INT,

FOREIGN KEY (Category\_ID) REFERENCES Category\_Details(Category\_ID)

);

**-- Create Author\_Details Table**

CREATE TABLE Author\_Details (

Author\_ID INT PRIMARY KEY,

Name VARCHAR(100),

Nationality VARCHAR(50)

);

**-- Create Publisher\_Details Table**

CREATE TABLE Publisher\_Details (

Publisher\_ID INT PRIMARY KEY,

Name VARCHAR(100),

Location VARCHAR(100)

);

**-- Create Staff\_Details Table**

CREATE TABLE Staff\_Details (

Staff\_ID INT PRIMARY KEY,

Name VARCHAR(100),

Is\_Admin BOOLEAN,

Role VARCHAR(100)

);

**-- Create Student\_Details Table**

CREATE TABLE Student\_Details (

Student\_ID INT PRIMARY KEY,

Name VARCHAR(100),

Gender VARCHAR(10),

Date\_Of\_Birth DATE,

Department VARCHAR(100),

Contact\_Number VARCHAR(20)

);

**-- Create Shelf\_Details Table**

CREATE TABLE Shelf\_Details (

Shelf\_ID INT PRIMARY KEY,

Floor VARCHAR(50)

);

**-- Create Borrower\_Details Table**

CREATE TABLE Borrower\_Details (

Borrower\_ID INT PRIMARY KEY,

Book\_ID INT,

Borrowed\_Date DATE,

Due\_Date DATE,

Returned\_Date DATE,

Issued\_by INT,

FOREIGN KEY (Book\_ID) REFERENCES Book\_Details(ISBN),

FOREIGN KEY (Issued\_by) REFERENCES Staff\_Details(Staff\_ID)

);

**-- Create Reservation\_Details Table**

CREATE TABLE Reservation\_Details (

Reservation\_ID INT PRIMARY KEY,

Borrower\_ID INT,

Book\_ID INT,

Reservation\_Date DATE,

Status VARCHAR(20),

FOREIGN KEY (Borrower\_ID) REFERENCES Student\_Details(Student\_ID),

FOREIGN KEY (Book\_ID) REFERENCES Book\_Details(ISBN)

);

**-- Create Transaction\_Details Table**

CREATE TABLE Transaction\_Details (

Transaction\_ID INT PRIMARY KEY,

Borrower\_ID INT,

Book\_ID INT,

Transaction\_Type VARCHAR(20),

Transaction\_Date DATE,

FOREIGN KEY (Borrower\_ID) REFERENCES Student\_Details(Student\_ID),

FOREIGN KEY (Book\_ID) REFERENCES Book\_Details(ISBN)

);

**Inserting Dummy Records:-**

**-- Insert into Category\_Details**

INSERT INTO Category\_Details (Category\_ID, Category\_Name) VALUES

(1, 'Science Fiction'),

(2, 'Fantasy'),

(3, 'Mystery'),

(4, 'Biography');

**-- Insert into Book\_Details**

INSERT INTO Book\_Details (ISBN, Title, Publication\_Year, Language, Category\_ID, No\_Of\_Copies\_Actual, No\_Of\_Copies\_Current) VALUES

(1001, 'Dune', 1965, 'English', 1, 10, 5),

(1002, 'Harry Potter and the Philosopher''s Stone', 1997, 'English', 2, 15, 10),

(1003, 'The Da Vinci Code', 2003, 'English', 3, 8, 8),

(1004, 'The Hobbit', 1937, 'English', 2, 12, 12),

(1005, '1984', 1949, 'English', 1, 20, 15),

(1006, 'To Kill a Mockingbird', 1960, 'English', 4, 10, 5),

(1007, 'Pride and Prejudice', 1813, 'English', 4, 10, 10),

(1008, 'The Catcher in the Rye', 1951, 'English', 4, 8, 8),

(1009, 'Lord of the Flies', 1954, 'English', 2, 10, 10),

(1010, 'The Great Gatsby', 1925, 'English', 4, 15, 15);

**-- Insert into Author\_Details**

INSERT INTO Author\_Details (Author\_ID, Name, Nationality) VALUES

(1, 'Frank Herbert', 'American'),

(2, 'J.K. Rowling', 'British'),

(3, 'Dan Brown', 'American'),

(4, 'J.R.R. Tolkien', 'British'),

(5, 'George Orwell', 'British'),

(6, 'Harper Lee', 'American'),

(7, 'Jane Austen', 'British'),

(8, 'J.D. Salinger', 'American'),

(9, 'William Golding', 'British'),

(10, 'F. Scott Fitzgerald', 'American');

**-- Insert into Publisher\_Details**

INSERT INTO Publisher\_Details (Publisher\_ID, Name, Location) VALUES

(1, 'Ace Books', 'USA'),

(2, 'Bloomsbury Publishing', 'UK'),

(3, 'Doubleday', 'USA'),

(4, 'Allen & Unwin', 'UK'),

(5, 'Secker & Warburg', 'UK'),

(6, 'J. B. Lippincott & Co.', 'USA'),

(7, 'Thomas Egerton', 'UK'),

(8, 'Little, Brown and Company', 'USA'),

(9, 'Faber and Faber', 'UK'),

(10, 'Charles Scribner''s Sons', 'USA');

**-- Insert into Staff\_Details**

INSERT INTO Staff\_Details (Staff\_ID, Name, Is\_Admin, Role) VALUES

(1, 'Alice', true, 'Librarian'),

(2, 'Bob', false, 'Assistant Librarian'),

(3, 'Charlie', true, 'Admin Officer'),

(4, 'Diana', false, 'Library Assistant');

**-- Insert into Student\_Details**

INSERT INTO Student\_Details (Student\_ID, Name, Gender, Date\_Of\_Birth, Department, Contact\_Number) VALUES

(1, 'John Doe', 'Male', '2000-01-15', 'Computer Science', '123-456-7890'),

(2, 'Jane Smith', 'Female', '1999-05-20', 'Literature', '987-654-3210'),

(3, 'Michael Johnson', 'Male', '2001-03-10', 'History', '555-123-4567'),

(4, 'Emily Brown', 'Female', '2000-12-05', 'Mathematics', '111-222-3333'),

(5, 'William Wilson', 'Male', '2001-08-25', 'Physics', '999-888-7777');

-- Insert into Shelf\_Details

INSERT INTO Shelf\_Details (Shelf\_ID, Floor) VALUES

(1, 'Ground Floor'),

(2, 'First Floor'),

(3, 'Second Floor'),

(4, 'Third Floor'),

(5, 'Basement');

-- Insert into Borrower\_Details

INSERT INTO Borrower\_Details (Borrower\_ID, Book\_ID, Borrowed\_Date, Due\_Date, Returned\_Date, Issued\_by) VALUES

(1, 1001, '2023-05-10', '2023-05-24', '2023-05-24', 1),

(2, 1002, '2023-05-12', '2023-05-26', NULL, 2),

(3, 1003, '2023-05-15', '2023-05-29', NULL, 3),

(4, 1004, '2023-05-18', '2023-06-01', NULL, 1),

(5, 1005, '2023-05-20', '2023-06-03', NULL, 2);

**-- Insert into Reservation\_Details**

INSERT INTO Reservation\_Details (Reservation\_ID, Borrower\_ID, Book\_ID, Reservation\_Date, Status) VALUES

(1, 1, 1003, '2023-05-08', 'pending'),

(2, 2, 1004, '2023-05-10', 'fulfilled'),

(3, 3, 1005, '2023-05-12', 'pending'),

(4, 4, 1006, '2023-05-15', 'fulfilled'),

(5, 5, 1007, '2023-05-18', 'pending');

**-- Insert into Transaction\_Details**

INSERT INTO Transaction\_Details (Transaction\_ID, Borrower\_ID, Book\_ID, Transaction\_Type, Transaction\_Date) VALUES

(1, 1, 1001, 'borrow', '2023-05-10'),

(2, 2, 1002, 'borrow', '2023-05-12'),

(3, 3, 1003, 'borrow', '2023-05-15'),

(4, 4, 1004, 'borrow', '2023-05-18'),

(5, 5, 1005, 'borrow', '2023-05-20');

**SQL Queries with Operators, Insert, Update, Delete**

**1.Retrieve the titles of all books published between 2000 and 2010**.

SELECT Title FROM Book\_Details WHERE Publication\_Year BETWEEN 2000 AND 2010;

2. Find the total number of books available in each category.

SELECT Category\_Name, COUNT(\*) AS Total\_Books FROM Book\_Details

JOIN Category\_Details ON Book\_Details.Category\_ID = Category\_Details.Category\_ID

GROUP BY Category\_Name;

3. List the names of all authors whose nationality is not American.

SELECT DISTINCT Name FROM Author\_Details WHERE Nationality != 'American';

4. Find the titles of books written by authors whose names start with 'J'.

SELECT Title

FROM Book\_Details

WHERE ISBN IN (

SELECT ISBN

FROM Author\_Details

WHERE Name LIKE 'J%'

);

5. Retrieve the names of all students who borrowed a book.

SELECT DISTINCT Name FROM Student\_Details

JOIN Borrower\_Details ON Student\_Details.Student\_ID = Borrower\_Details.Borrower\_ID;

6.Insert a new category into the Category\_Details table.

INSERT INTO Category\_Details (Category\_ID, Category\_Name) VALUES (5, 'Thriller');

7.Update the publication year of a specific book.

UPDATE Book\_Details SET Publication\_Year = 2015 WHERE ISBN = 1003;

8.Delete a record from the Staff\_Details table.

DELETE FROM Staff\_Details WHERE Staff\_ID = 4;

9.Count the number of reserved books for each student.

SELECT Borrower\_ID, COUNT(\*) AS Reserved\_Books

FROM Reservation\_Details

GROUP BY Borrower\_ID;

10.Calculate the average publication year of all books.

SELECT AVG(Publication\_Year) AS Avg\_Publication\_Year FROM Book\_Details;

**SQL Queries with Joins**

1.Retrieve the titles of all books along with their corresponding categories.

SELECT b.Title, c.Category\_Name

FROM Book\_Details b

JOIN Category\_Details c ON b.Category\_ID = c.Category\_ID;

2. List the names of all authors along with the titles of the books they have written.

SELECT s.Name AS Student\_Name, b.Title AS Book\_Title

FROM Student\_Details s

JOIN Borrower\_Details br ON s.Student\_ID = br.Borrower\_ID

JOIN Book\_Details b ON br.Book\_ID = b.ISBN;

3.Find the names of all students who borrowed a book along with the titles of the borrowed books.

SELECT s.Name AS Student\_Name, b.Title AS Book\_Title

FROM Student\_Details s

JOIN Borrower\_Details br ON s.Student\_ID = br.Borrower\_ID

JOIN Book\_Details b ON br.Book\_ID = b.ISBN;

4. **List all transactions along with the book titles and borrower names.**

SELECT t.Transaction\_ID, t.Transaction\_Type, b.Title, s.Name

FROM Transaction\_Details t

JOIN Book\_Details b ON t.Book\_ID = b.ISBN

JOIN Borrower\_Details br ON t.Borrower\_ID = br.Borrower\_ID

JOIN Student\_Details s ON br.Borrower\_ID = s.Student\_ID;

5. List the names of all students along with the number of books they have borrowed.

SELECT s.Name AS Student\_Name, COUNT(br.Book\_ID) AS Borrowed\_Books

FROM Student\_Details s

LEFT JOIN Borrower\_Details br ON s.Student\_ID = br.Borrower\_ID

GROUP BY s.Student\_ID;

6. Retrieve the titles of all books published by American authors.

SELECT b.Title

FROM Book\_Details b

JOIN Author\_Details a ON b.ISBN = a.Author\_ID

WHERE a.Nationality = 'American';

7. Find the names of all authors who have written books in the 'Fantasy' category.

SELECT a.Name AS Author\_Name

FROM Author\_Details a

JOIN Book\_Details b ON a.Author\_ID = b.ISBN

JOIN Category\_Details c ON b.Category\_ID = c.Category\_ID

WHERE c.Category\_Name = 'Fantasy';

8. List the names of all publishers who have published books in the 'Science Fiction' category

SELECT Name

FROM Student\_Details

WHERE Student\_ID IN (

SELECT Borrower\_ID

FROM Borrower\_Details

WHERE Book\_ID IN (

SELECT ISBN

FROM Book\_Details

WHERE ISBN IN (

SELECT ISBN

FROM Book\_Details

JOIN Author\_Details ON Book\_Details.ISBN = Author\_Details.Author\_ID

WHERE Nationality = 'American'

)

)

);

9. Retrieve the titles of all books borrowed by students in the 'Computer Science' department.

SELECT b.Title

FROM Book\_Details b

JOIN Borrower\_Details br ON b.ISBN = br.Book\_ID

JOIN Student\_Details s ON br.Borrower\_ID = s.Student\_ID

WHERE s.Department = 'Computer Science';

10. Find the names of all students who have borrowed books published after 2010.

SELECT s.Name AS Student\_Name

FROM Student\_Details s

JOIN Borrower\_Details br ON s.Student\_ID = br.Borrower\_ID

JOIN Book\_Details b ON br.Book\_ID = b.ISBN

WHERE b.Publication\_Year > 2010;

**SQL Queries with Nested Queries**

1.Retrieve the names of all students who have borrowed books published before 2000.

SELECT Name

FROM Student\_Details

WHERE Student\_ID IN (SELECT Borrower\_ID FROM Borrower\_Details WHERE Book\_ID IN (SELECT ISBN FROM Book\_Details WHERE Publication\_Year < 2000));

2.Find the titles of all books borrowed by students in the 'Physics' department.

SELECT Title

FROM Book\_Details

WHERE ISBN IN (SELECT Book\_ID FROM Borrower\_Details WHERE Borrower\_ID IN (SELECT Student\_ID FROM Student\_Details WHERE Department = 'Physics'));

3.List the names of all authors who have written books published in the 'Fantasy' category.

SELECT Name

FROM Author\_Details

WHERE Author\_ID IN (SELECT Author\_ID FROM Book\_Details WHERE Category\_ID IN (SELECT Category\_ID FROM Category\_Details WHERE Category\_Name = 'Fantasy'));

4.Retrieve the names of all students who have borrowed more than 3 books.

SELECT Name

FROM Student\_Details

WHERE Student\_ID IN (SELECT Borrower\_ID FROM Borrower\_Details GROUP BY Borrower\_ID HAVING COUNT(Book\_ID) > 3);

5.Find the titles of all books with a publication year not between 2005 and 2015.

SELECT Title

FROM Book\_Details

WHERE ISBN NOT IN (SELECT ISBN FROM Book\_Details WHERE Publication\_Year BETWEEN 2005 AND 2015);

6.Retrieve the names of all authors who have written books borrowed by students in the 'Chemistry' department.

SELECT Name

FROM Author\_Details

WHERE Author\_ID IN (SELECT Author\_ID FROM Book\_Details WHERE ISBN IN (SELECT Book\_ID FROM Borrower\_Details WHERE Borrower\_ID IN (SELECT Student\_ID FROM Student\_Details WHERE Department = 'Chemistry')));

7.List the titles of all books borrowed by students who have borrowed books published by 'Penguin Publishers'.

8. List Authors of Books Reserved but Not Yet Fulfilled

SELECT Name

FROM Author\_Details

WHERE Author\_ID IN (

SELECT Author\_ID

FROM Book\_Details

WHERE ISBN IN (

SELECT Book\_ID

FROM Reservation\_Details

WHERE Status = 'pending'

)

);

9. Find Books Borrowed by Students in the Literature Department

SELECT Title

FROM Book\_Details

WHERE Publication\_Year = (SELECT MAX(Publication\_Year) FROM Book\_Details);

10.List the names of all authors who have not written books in the 'Mystery' category.

SELECT Name

FROM Author\_Details

WHERE Author\_ID NOT IN (SELECT Author\_ID FROM Book\_Details WHERE Category\_ID IN (SELECT Category\_ID FROM Category\_Details WHERE Category\_Name = 'Mystery'));

**SQL Queries with Views**

Create a view to display the names and categories of all books.

CREATE VIEW Book\_Names\_Categories AS

SELECT b.Title, c.Category\_Name

FROM Book\_Details b

JOIN Category\_Details c ON b.Category\_ID = c.Category\_ID;

2. Retrieve all records from the created view.

SELECT \* FROM Book\_Names\_Categories;

3. Create a view to list the names of all students along with their departments.

CREATE VIEW Student\_Departments AS

SELECT Name, Department

FROM Student\_Details;

4. Retrieve all records from the created view.

SELECT \* FROM Student\_Departments;

5.Create a view to display the names and nationalities of all authors.

CREATE VIEW Author\_Nationalities AS

SELECT Name, Nationality

FROM Author\_Details;

6. Retrieve all records from the created view.

SELECT \* FROM Author\_Nationalities;

7. Create a view to list the titles and publication years of all books published after 2010.

CREATE VIEW Recent\_Books AS

SELECT Title, Publication\_Year

FROM Book\_Details

WHERE Publication\_Year > 2010;

8. Retrieve all records from the created view.

SELECT \* FROM Recent\_Books;

9.Create a view to display the names of all publishers along with their locations

CREATE VIEW Publisher\_Locations AS

SELECT Name, Location

FROM Publisher\_Details;

10Retrieve all records from the created view.

SELECT \* FROM Publisher\_Locations;

**SQL Queries with Triggers**

Let's create a trigger that updates the current date when a book is borrowed:

CREATE TRIGGER UpdateBorrowDate

AFTER INSERT ON Borrower\_Details

FOR EACH ROW

BEGIN

UPDATE Borrower\_Details

SET Borrowed\_Date = CURDATE()

WHERE B\_ID = NEW.B\_ID;

END;

**Stored Procedure**

**1.Procedure to Insert a Book:**

CREATE PROCEDURE InsertBook(

IN p\_Title VARCHAR(255),

IN p\_PublicationYear INT,

IN p\_Language VARCHAR(255),

IN p\_CategoryID INT,

IN p\_NoOfCopiesActual INT,

IN p\_NoOfCopiesCurrent INT

)

BEGIN

INSERT INTO Book\_Details (Title, Publication\_Year, Language, Category\_ID, No\_Of\_Copies\_Actual, No\_Of\_Copies\_Current)

VALUES (p\_Title, p\_PublicationYear, p\_Language, p\_CategoryID, p\_NoOfCopiesActual, p\_NoOfCopiesCurrent);

END;

**2.Procedure to Update Book Title:**

CREATE PROCEDURE UpdateBookTitle(

IN p\_ISBN INT,

IN p\_NewTitle VARCHAR(255)

)

BEGIN

UPDATE Book\_Details

SET Title = p\_NewTitle

WHERE ISBN = p\_ISBN;

END;

**3.Procedure to Delete Book by ISBN:**

CREATE PROCEDURE DeleteBookByISBN(

IN p\_ISBN INT

)

BEGIN

DELETE FROM Book\_Details

WHERE ISBN = p\_ISBN;

END;

4.**Procedure to Get Total Number of Books in a Category:**

CREATE PROCEDURE GetTotalBooksInCategory(

IN p\_CategoryName VARCHAR(255),

OUT p\_TotalBooks INT

)

BEGIN

SELECT COUNT(\*) INTO p\_TotalBooks

FROM Book\_Details bd

JOIN Category\_Details cd ON bd.Category\_ID = cd.Category\_ID

WHERE cd.Category\_Name = p\_CategoryName;

END**;**

**5.Procedure to Get Book Details by ISBN:**

CREATE PROCEDURE GetBookDetailsByISBN(

IN p\_ISBN INT

)

BEGIN

SELECT \*

FROM Book\_Details

WHERE ISBN = p\_ISBN;

END;