# Task 6: Subqueries and Nested Queries

Objective: Use subqueries in SELECT, WHERE, and FROM

Tools: DB Browser for SQLite / MySQL Workbench

Deliverables: SQL queries with nested logic

Outcome: Advanced query logic skills

Schema used: Library Management System

## Assumed Tables

-- Book Table  
CREATE TABLE Book (  
 BookID INT PRIMARY KEY,  
 Title VARCHAR(100),  
 Author VARCHAR(100),  
 CategoryID INT,  
 PublisherID INT,  
 Price DECIMAL(6,2)  
);  
  
-- Borrower Table  
CREATE TABLE Borrower (  
 BorrowerID INT PRIMARY KEY,  
 Name VARCHAR(100),  
 MembershipDate DATE  
);  
  
-- BorrowedBooks Table  
CREATE TABLE BorrowedBooks (  
 BorrowID INT PRIMARY KEY,  
 BookID INT,  
 BorrowerID INT,  
 BorrowDate DATE,  
 ReturnDate DATE  
);

### 1. Subquery in WHERE Clause (IN): Find the names of borrowers who have borrowed books priced above 500.

SELECT Name  
FROM Borrower  
WHERE BorrowerID IN (  
 SELECT BorrowerID  
 FROM BorrowedBooks  
 WHERE BookID IN (  
 SELECT BookID  
 FROM Book  
 WHERE Price > 500  
 )  
);

### 2. Scalar Subquery in SELECT Clause: Display each borrower's name and the total number of books they have borrowed.

SELECT Name,  
 (SELECT COUNT(\*)  
 FROM BorrowedBooks  
 WHERE BorrowedBooks.BorrowerID = Borrower.BorrowerID) AS TotalBorrowed  
FROM Borrower;

### 3. Subquery in FROM Clause: Get the average price of books borrowed by each borrower.

SELECT b.Name, AVG\_Books.PriceAvg  
FROM Borrower b  
JOIN (  
 SELECT bb.BorrowerID, AVG(bk.Price) AS PriceAvg  
 FROM BorrowedBooks bb  
 JOIN Book bk ON bb.BookID = bk.BookID  
 GROUP BY bb.BorrowerID  
) AS AVG\_Books  
ON b.BorrowerID = AVG\_Books.BorrowerID;

### 4. Correlated Subquery: Find books that have never been borrowed.

SELECT Title  
FROM Book b  
WHERE NOT EXISTS (  
 SELECT 1  
 FROM BorrowedBooks bb  
 WHERE bb.BookID = b.BookID  
);

### 5. Subquery using = (Scalar): Find the borrower who borrowed the most expensive book.

SELECT Name  
FROM Borrower  
WHERE BorrowerID = (  
 SELECT BorrowerID  
 FROM BorrowedBooks  
 WHERE BookID = (  
 SELECT BookID  
 FROM Book  
 ORDER BY Price DESC  
 LIMIT 1  
 )  
);

### 6. Subquery using EXISTS: List all books that have been borrowed at least once.

SELECT Title  
FROM Book b  
WHERE EXISTS (  
 SELECT 1  
 FROM BorrowedBooks bb  
 WHERE bb.BookID = b.BookID  
);