SQL JOIN - Full Day Intensive Practice

Complete Beginner's Guide

IMPORTANT: This workbook is designed for students with NO prior knowledge of JOINs. We will build understanding step-by-step, starting from the basics.

Today's Learning Path

This full-day practice is divided into progressive sessions:

- Session 1: Understanding the Concept (Theory + Simple Examples)
- Session 2: INNER JOIN - Matching Data
- Session 3: LEFT JOIN - Keeping All Left Records
- Session 4: RIGHT JOIN - Keeping All Right Records
- Session 5: FULL OUTER JOIN - Keeping Everything
- Session 6: Combining Multiple Joins
- Session 7: Real-World Challenge Queries

Session 1: Understanding the Concept

Why Do We Need JOINs?

In a relational database, data is stored in separate tables to avoid repetition. For example:

- **Member table** stores member information (MemberID, Name, Email)
- **Loan table** stores loan information (LoanID, MemberID, BookID, LoanDate)
- **Book table** stores book information (BookID, Title, Genre)

The Problem: If you want to see 'Who borrowed which book?', you need data from BOTH the Member and Loan tables. SELECT from one table alone won't give you the complete picture.

The Solution: JOINs allow you to combine data from multiple tables based on a related column (usually Foreign Key).

Basic JOIN Syntax Structure

```
SELECT columns
FROM table1
[JOIN TYPE] table2
ON table1.column = table2.column;
```

Understanding Table Aliases

Instead of writing full table names repeatedly, we use short aliases:

```
SELECT M.FullName, L.LoanDate
FROM Member M              -- M is alias for Member
INNER JOIN Loan L          -- L is alias for Loan
ON M.MemberID = L.MemberID;
```

Warm-Up Tasks (Understanding Relationships)

Before writing JOINs, answer these questions to understand table relationships:

Task 1.1: Look at the Loan table. Which column connects it to the Member table?

Answer: LoanID

Task 1.2: Which column connects the Loan table to the Book table?

Answer: BookID

Task 1.3: Which column connects the Staff table to the Library table?

Answer: LivararyID

Task 1.4: Which column connects the Review table to the Book table?

Answer: BookID

Session 2: INNER JOIN - Matching Data Only

Concept Explanation

INNER JOIN returns only the rows where there is a match in BOTH tables.

Think of it like this: If a member has no loans, they won't appear. If a book has never been borrowed, it won't appear. Only members WHO HAVE loans will be shown.

Basic Syntax

```
SELECT columns
FROM table1
INNER JOIN table2
ON table1.common_column = table2.common_column;
```

Guided Example - Step by Step

Example: Show all loans with member names

Step 1: Identify which tables you need
- We need Member (for names) and Loan (for loan information)

Step 2: Find the connecting column
- Both tables have MemberID

Step 3: Write the query

```
SELECT M.FullName, L.LoanDate, L.DueDate
```

```
FROM Member M
INNER JOIN Loan L
ON M.MemberID = L.MemberID;
```

Practice Tasks - INNER JOIN

Part A: Two-Table INNER JOINs

Task 2.1: Display all loans with the book titles.
Columns needed: LoanID, Book Title, LoanDate, DueDate
Tables: Loan and Book
Hint: Connect using BookID

Task 2.2: Show all staff members with their library names.
Columns needed: Staff FullName, Position, Library Name, Library Location

Task 2.3: Display all books with their library information.
Columns needed: Book Title, Genre, Price, Library Name, Library Location

Task 2.4: Show all reviews with member names.
Columns needed: Member FullName, Rating, Comments, ReviewDate

Task 2.5: Display all reviews with book titles.
Columns needed: Book Title, Rating, Comments, ReviewDate

Task 2.6: Show all payments with loan information.
Columns needed: LoanID, PaymentDate, Amount, PaymentMethod, Loan Status

Part B: Three-Table INNER JOINs

Task 2.7: Show member names, book titles, and loan dates together.
Columns needed: Member FullName, Book Title, LoanDate, DueDate, Status
Hint: You need Member, Loan, and Book tables. Join Member to Loan, then Loan to Book

Task 2.8: Display reviews with both member names and book titles.
Columns needed: Member FullName, Book Title, Rating, Comments

Task 2.9: Show books with their library name and location.
Columns needed: Book Title, Genre, Library Name, Library Location, Library ContactNumber

Task 2.10: Display complete loan information: member name, book title, loan dates, and status.
Columns needed: Member FullName, Email, Book Title, Genre, LoanDate, DueDate, ReturnDate, Status

Session 3: LEFT JOIN - Keep All Left Records

Concept Explanation

LEFT JOIN returns ALL rows from the left table, and matching rows from the right table. If no match exists, NULL values appear for the right table.

Example: If you want to see ALL members (even those who never borrowed a book), use LEFT JOIN. Members without loans will show NULL for loan information.

Visual Understanding

INNER JOIN: Only members WHO HAVE borrowed books
LEFT JOIN: ALL members (borrowed or not)

Basic Syntax

```
SELECT columns
FROM table1          -- This is the LEFT table
LEFT JOIN table2     -- This is the RIGHT table
ON table1.column = table2.column;
```

Guided Example

Example: Show ALL members and their loans (if any)

```
SELECT M.FullName, L.LoanDate, L.Status
FROM Member M        -- ALL members will appear
LEFT JOIN Loan L     -- Loan info if available, NULL if not
ON M.MemberID = L.MemberID;
```

Practice Tasks - LEFT JOIN

Task 3.1: Show ALL books and their reviews (if any). Books without reviews should still appear.
Columns needed: Book Title, Genre, Rating, Comments
Expected: All 18 books should appear, even if some have no reviews

Task 3.2: Display ALL members with their loan information (if they have any loans).
Columns needed: Member FullName, Email, LoanDate, DueDate, Status

Task 3.3: Show ALL loans with payment information (if any). Loans without payments should appear.
Columns needed: LoanID, LoanDate, Status, PaymentDate, Amount

Task 3.4: Display ALL libraries with their staff (if any).
Columns needed: Library Name, Location, Staff FullName, Position

Task 3.5: Find members who have NEVER borrowed a book (using LEFT JOIN with WHERE clause).
Hint: Use LEFT JOIN, then add WHERE L.LoanID IS NULL

Task 3.6: Find books that have NEVER been reviewed.

Columns needed: Book Title, Genre, Price
Hint: LEFT JOIN with WHERE R.ReviewID IS NULL

Task 3.7: Show ALL books with member names who reviewed them (if reviewed).
Columns needed: Book Title, Member FullName, Rating, Comments
Challenge: This needs a three-table join: Book LEFT JOIN Review LEFT JOIN Member

Session 4: RIGHT JOIN - Keep All Right Records

Concept Explanation

RIGHT JOIN is the opposite of LEFT JOIN. It returns ALL rows from the right table and matching rows from the left table.

Note: RIGHT JOIN is less commonly used because you can achieve the same result by swapping tables and using LEFT JOIN. Most developers prefer LEFT JOIN for consistency.

Basic Syntax

```
SELECT columns
FROM table1          -- This is the LEFT table
RIGHT JOIN table2    -- ALL rows from this table will appear
ON table1.column = table2.column;
```

Comparison Example

These two queries produce the SAME result:

```
-- Using LEFT JOIN
SELECT M.FullName, L.LoanDate
FROM Member M
LEFT JOIN Loan L ON M.MemberID = L.MemberID;

-- Using RIGHT JOIN (same result, tables swapped)
SELECT M.FullName, L.LoanDate
FROM Loan L
RIGHT JOIN Member M ON M.MemberID = L.MemberID;
```

Practice Tasks - RIGHT JOIN

Task 4.1: Rewrite Task 3.1 using RIGHT JOIN instead of LEFT JOIN.
Original (LEFT JOIN): Book LEFT JOIN Review
Rewrite (RIGHT JOIN): Review RIGHT JOIN Book

Task 4.2: Show ALL members with their loans using RIGHT JOIN.
Hint: FROM Loan L RIGHT JOIN Member M

Task 4.3: Display ALL books using RIGHT JOIN with Library table.
Columns needed: Library Name, Book Title, Genre

Task 4.4: Compare the results: Write the same query using both LEFT JOIN and RIGHT JOIN.
Goal: Show all staff with library names

Session 5: FULL OUTER JOIN - Keep Everything

Concept Explanation

FULL OUTER JOIN returns ALL rows from both tables. If there's no match, NULL values appear on the side without a match.

Think of it as combining LEFT JOIN and RIGHT JOIN together. You get everything from both sides.

Basic Syntax

```
SELECT columns
FROM table1
FULL OUTER JOIN table2
ON table1.column = table2.column;
```

Important Note for SQL Server

Note: MySQL does not support FULL OUTER JOIN directly. If you're using MySQL, you can simulate it using UNION of LEFT JOIN and RIGHT JOIN. SQL Server and PostgreSQL support it natively.

Guided Example

Example: Show ALL members and ALL loans (matched or unmatched)

```
SELECT M.FullName, L.LoanDate, L.Status
FROM Member M
FULL OUTER JOIN Loan L
ON M.MemberID = L.MemberID;
```

Practice Tasks - FULL OUTER JOIN

Task 5.1: Show ALL books and ALL reviews (whether matched or not).
Columns needed: Book Title, Rating, Comments

Task 5.2: Display ALL loans and ALL payments.
Columns needed: LoanID, LoanDate, PaymentDate, Amount

Task 5.3: Show complete data: ALL members and ALL books (through loans).
Challenge: This requires multiple FULL OUTER JOINs

Session 6: Combining Multiple JOIN Types

Concept

You can mix different JOIN types in one query. For example, INNER JOIN one table and LEFT JOIN another.

Example Structure

```
SELECT columns
FROM table1
INNER JOIN table2 ON table1.id = table2.id
LEFT JOIN table3 ON table2.id = table3.id;
```

Practice Tasks - Mixed JOINs

Task 6.1: Show all ACTIVE loans (INNER JOIN) with member info and payment info (LEFT JOIN for payments).
Columns: Member FullName, LoanDate, DueDate, Status, PaymentDate, Amount
Logic: INNER JOIN Member (only show actual loans), LEFT JOIN Payment (show payment if exists)

Task 6.2: Display all books (INNER JOIN Library) and their reviews (LEFT JOIN Review).
Columns: Library Name, Book Title, Genre, Rating, Comments

Task 6.3: Show member borrowing history with book and library information.
Columns: Member FullName, Book Title, Library Name, LoanDate, ReturnDate
Tables needed: Member, Loan, Book, Library (4 tables!)

Session 7: Real-World Challenge Queries

These challenges combine everything you've learned. Read each requirement carefully and determine which JOIN type(s) to use.

Challenge 7.1: Library Performance Report
Create a report showing: Library Name, Total Books, Total Staff, Total Loans
Hint: You'll need to JOIN multiple tables and use COUNT

Challenge 7.2: Member Activity Summary
Show ALL members with: FullName, Email, Total Loans, Total Reviews Written
Hint: LEFT JOIN to include members with zero loans/reviews

Challenge 7.3: Book Popularity Analysis
Display: Book Title, Library Name, Times Borrowed, Average Rating, Total Reviews
Hint: Need Book, Library, Loan (count), Review (avg and count)

Challenge 7.4: Overdue Books Report
Show all overdue loans with: Member Name, Email, Book Title, Library Name, Days Overdue, Fine Paid (if any)

Hint: INNER JOIN for overdue loans, LEFT JOIN for payments, calculate days with DATEDIFF

Challenge 7.5: Complete Member Loan History
Create a detailed view: Member Name, Book Title, Genre, Library Location, Loan Date, Return Date, Days Borrowed, Rating Given (if reviewed)
Challenge: This requires joining 5+ tables with mixed JOIN types!

Challenge 7.6: Books Never Borrowed
Find books that exist but have NEVER been borrowed. Show: Book Title, Genre, Price, Library Name
Hint: Book LEFT JOIN Loan WHERE LoanID IS NULL

Challenge 7.7: Members With No Activity
Find members who have NEVER borrowed a book AND NEVER written a review.
Hint: Need two LEFT JOINs with WHERE both are NULL

Challenge 7.8: Staff Workload Analysis
Show: Staff Name, Position, Library Name, Number of Books in Library, Number of Active Loans
Complex: Requires grouping and counting across multiple joins

Submission Guidelines

What to Submit

1. **SQL Script File (.sql):** Organized by session with all queries
2. **Comments:** Explain what each query does and which JOIN type you used
3. **Screenshots:** At least 3 screenshots per session showing query results
4. **Reflection Document:** 1-2 pages explaining:
   - Differences between JOIN types you learned
   - Which JOIN type was most challenging and why
   - Real-world scenarios where you would use each JOIN
   - Questions you still have

Script Format Example

```
-- ================================================
-- SESSION 2: INNER JOIN PRACTICE
-- ================================================

-- Task 2.1: Display all loans with book titles
-- Using INNER JOIN because we only want loans that have books
SELECT L.LoanID, B.Title, L.LoanDate, L.DueDate
FROM Loan L
INNER JOIN Book B ON L.BookID = B.BookID;
```

Evaluation Criteria

Criteria	Points
Session 1 (Concept Understanding)	5 points
Session 2 (INNER JOIN)	20 points
Session 3 (LEFT JOIN)	20 points
Session 4 (RIGHT JOIN)	10 points
Session 5 (FULL OUTER JOIN)	10 points
Session 6 (Multiple JOINs)	15 points
Session 7 (Challenges)	10 points
Code Quality & Comments	5 points
Reflection Document	5 points
TOTAL	**100 points**

Tips for Success

- **Start Simple:** Master 2-table joins before attempting 3+ tables
- **Test Incrementally:** Run each query and verify results before adding complexity
- **Understand the Difference:** Know when to use INNER vs LEFT vs RIGHT
- **Use Aliases:** Always use table aliases (M for Member, L for Loan, etc.) for readability
- **Check for NULLs:** When using LEFT/RIGHT/FULL joins, check for NULL values in results
- **Ask Questions:** Don't hesitate to ask the instructor if confused
- **Practice Makes Perfect:** The more queries you write, the more natural JOINs become