Goal: Learn SQL from scratch (installation → advanced)
Focus: Accounting and financial use cases
Outcome: Ability to query, analyze, and visualize accounting/finance data using open-source tools.

🧩 STRUCTURE OF THE COURSE / MATERIAL

Level	Topics	Accounting Context / Examples
1. Installation & Setup	• Install PostgreSQL (recommended) or SQLite (lightweight). • Install pgAdmin or DBeaver (SQL IDE). • Basic database concepts (tables, rows, columns, primary key, foreign key).	Setup "AccountingDB" with tables: `Customers`, `Invoices`, `Payments`, `Accounts`, `JournalEntries`, `Expenses`.
2. Basic SQL	• SELECT, FROM, WHERE, ORDER BY • DISTINCT, LIMIT • Arithmetic & logical operators • Filtering dates, text, and numbers	- Get all unpaid invoices. - List top 10 customers by total invoice amount. - Find expenses above ₹10,000.
3. Intermediate SQL	• Joins (INNER, LEFT, RIGHT, FULL) • GROUP BY, HAVING • Aggregate functions (SUM, AVG, COUNT, MIN, MAX) • String & date functions	- Join invoices with customers. - Monthly revenue report. - Year-over-year expense trend. - Calculate profit margin by department.
4. Advanced SQL	• Subqueries • Common Table Expressions (CTEs) • Window functions (ROW_NUMBER, RANK, SUM OVER, etc.) • Case statements	- Find top 3 customers by revenue. - Calculate running totals of revenue. - Compare this year vs last year income. - Detect duplicate transactions.

5. Data Cleaning & Preparation	• Handling NULLs, COALESCE • CAST/CONVERT • Creating views for reporting • Using functions for rounding, text cleanup	- Fix missing customer addresses. - Clean transaction data for year-end reporting.
6. Accounting-Focused Use Cases	• Building trial balance queries • Preparing balance sheets using SQL • Income statement (P&L) queries • Reconciling accounts • VAT/GST reporting	- Trial balance grouped by account type. - Income statement by month. - GST collected vs paid.
7. Data Visualization (Open Source)	• Install **Metabase**, **Apache Superset**, or **Grafana**. • Connect to PostgreSQL. • Create dashboards: Revenue trend, expense ratio, outstanding payments, profitability.	- Build a "CFO Dashboard" for visualization.
8. Automation & Integration Tools	• Export SQL reports to Excel/CSV • Schedule queries • Use **Python** or **Jupyter Notebook** for automation.	- Automate daily balance summary. - Email overdue payment reports.

🧾 SAMPLE ACCOUNTING DATABASE STRUCTURE

Tables:

1. **Accounts**
   (AccountID, AccountName, AccountType)

2. **Customers**
   (CustomerID, CustomerName, City, Email)

3. **Invoices**
   (InvoiceID, CustomerID, InvoiceDate, TotalAmount, Status)

4. **Payments**
   (PaymentID, InvoiceID, PaymentDate, Amount)

5. **Expenses**
   (ExpenseID, ExpenseType, Amount, ExpenseDate, Department)

6. **JournalEntries**
   (EntryID, AccountID, Debit, Credit, EntryDate)

🧮 EXAMPLE SQL QUERIES (ACCOUNTING CONTEXT)

1️⃣ Unpaid Invoices

```sql
SELECT InvoiceID, CustomerID, TotalAmount
FROM Invoices
WHERE Status = 'Unpaid';
```

2️⃣ Monthly Revenue

```sql
SELECT DATE_TRUNC('month', InvoiceDate) AS Month,
       SUM(TotalAmount) AS TotalRevenue
FROM Invoices
WHERE Status = 'Paid'
GROUP BY 1
ORDER BY 1;
```

3️⃣ Trial Balance

```sql
SELECT
    a.AccountName,
    SUM(j.Debit) AS TotalDebit,
    SUM(j.Credit) AS TotalCredit,
    (SUM(j.Debit) - SUM(j.Credit)) AS Balance
FROM JournalEntries j
JOIN Accounts a ON j.AccountID = a.AccountID
GROUP BY a.AccountName
ORDER BY a.AccountName;
```

4️⃣ Profit and Loss Statement

```sql
SELECT
    CASE
        WHEN a.AccountType = 'Revenue' THEN 'Revenue'
        WHEN a.AccountType = 'Expense' THEN 'Expense'
    END AS Category,
    SUM(j.Debit - j.Credit) AS Amount
FROM JournalEntries j
JOIN Accounts a ON j.AccountID = a.AccountID
GROUP BY Category;
```

🧰 OPEN-SOURCE TOOLS FOR ACCOUNTANTS

Category	Tool	Description
Database	PostgreSQL / SQLite / MySQL	Store accounting data & practice SQL
SQL IDE	DBeaver / pgAdmin / Beekeeper Studio	GUI-based query tools
Data Visualization	**Metabase**, **Apache Superset**, **Grafana**, **Redash**	Build interactive dashboards (open-source)
Automation	**Python (pandas + sqlalchemy)**, **Jupyter Notebook**	Automate accounting reports
Spreadsheet Integration	**LibreOffice Calc**, **Google Sheets SQL connectors**	Combine SQL with spreadsheets
ETL (optional)	**Airbyte**, **Meltano**, **Apache Nifi**	Automate data movement

---

🧑 PRACTICAL MINI PROJECTS

1. **Invoice Dashboard** – Track paid/unpaid invoices, customer aging.

2. **Expense Analyzer** – Find expense trends by department/month.

3. **Trial Balance & P&L** – Auto-generated financial reports using SQL.

4. **Cash Flow Projection** – Predict cash inflows/outflows using SQL + Excel.

5. **Tax Compliance** – Identify transactions eligible for GST/VAT.