ACCIJOB SQL CURRICULUM

PostgreSQL + pgAdmin 4 + Visual Studio Code Edition

Instructor:	Sayyed Siraj Ali
Duration:	6 Weeks | 24 Sessions (4 sessions/week)
Session Length:	2 hours per session
Total Hours:	48 hours
Platform:	PostgreSQL 16 · pgAdmin 4 · Visual Studio Code
Database:	RetailMart (10+ Tables · 10k-50k Rows)

Curriculum Flow Philosophy: Concept Before Code, Practice Before Project

This curriculum follows a progressive, dependency-based learning structure ensuring a complete transition from foundational SQL concepts to analytical and performance-based SQL proficiency. Each session incrementally builds upon prior knowledge to ensure conceptual retention and industry application readiness.

TABLE 1: Day-Wise Overview

Day	Major Topic
1	Introduction to SQL & Installation
2	Basic Queries & DDL Commands
3	PostgreSQL Data Types
4	Constraints & Keys
5	Data Filtering & Sorting
6	Aggregate Functions & Grouping
7	Conditional Logic & Derived Columns
8	Joins - Part 1 (Foundations)
9	Joins - Part 2 (Advanced)
10	Self Join, Cross Join & Set Operations
11	Subqueries - Part 1
12	Subqueries - Part 2 & CTEs
13	Window Functions - Part 1 (Ranking)
14	Window Functions - Part 2 (Analytics)
15	Transactions & Error Control
16	ACID Properties & Exception Handling
17	Normalization & Data Modeling
18	Views & Materialized Views

19	Functions (PL/pgSQL)
20	Stored Procedures
21	Query Optimization, Indexing & Security
22	RetailMart Analytics Project - Part 1
23	RetailMart Analytics Project - Part 2
24	Dashboard Creation & Deployment

TABLE 2: Detailed Day-Wise Curriculum

Day	Topic	Subtopics
1	**Introduction to SQL & Installation**	• Introduction to SQL (What, Why, When) • Components of SQL (DDL, DML, DCL, TCL, DQL) • Database vs Schema • Database objects (Table, View, Index) • DBMS vs RDBMS • PostgreSQL Architecture • Install PostgreSQL 16, pgAdmin 4, VS Code • Configure VS Code extensions
2	**Basic Queries & DDL Commands**	• Writing basic SQL queries • DDL: CREATE DATABASE, CREATE SCHEMA, CREATE TABLE • DROP, ALTER commands • SELECT version() and system queries • Create RetailMart database • Onboard all 8 schemas • Create all tables • Import CSV data
3	**PostgreSQL Data Types**	• Numeric types (INT, DECIMAL, SERIAL) • Text types (CHAR, VARCHAR, TEXT) • Date/Time types (DATE, TIMESTAMP, INTERVAL) • Boolean type • Advanced types (JSON, JSONB, UUID, ARRAY) • Other types (BYTEA, MONEY, INET) • Selecting correct data types • Naming conventions • Altering columns
4	**Constraints & Keys**	• Constraints: NOT NULL, UNIQUE, CHECK, DEFAULT • PRIMARY KEY (single column) • COMPOSITE PRIMARY KEY • FOREIGN KEY • Natural vs Surrogate keys • Alternate keys • Cascading actions (CASCADE, RESTRICT, SET NULL) • Adding/Dropping constraints

5	**Data Filtering & Sorting**	• SELECT and WHERE clauses • Logical operators (AND, OR, NOT) • Comparison operators (=, !=, >, <, >=, <=) • Pattern matching (LIKE, ILIKE, %, _) • BETWEEN, IN, NOT IN • IS NULL, IS NOT NULL • DISTINCT, DISTINCT ON • ORDER BY (ASC, DESC) • LIMIT and OFFSET for pagination
6	**Aggregate Functions & Grouping**	• Aggregate functions: COUNT, SUM, AVG, MIN, MAX • STRING_AGG, ARRAY_AGG • GROUP BY (single and multiple columns) • HAVING clause for filtering aggregated data • HAVING vs WHERE • Sorting grouped results • Common patterns (sales by month, top customers) • Performance considerations
7	**Conditional Logic & Derived Columns**	• CASE WHEN (simple and searched) • Nested CASE statements • COALESCE for NULL handling • NULLIF to avoid division by zero • Derived/computed columns • Column aliases • Complex data categorization • Customer segmentation examples
8	**Joins - Part 1 (Foundations)**	• JOIN fundamentals and concepts • INNER JOIN syntax and use cases • LEFT JOIN (LEFT OUTER JOIN) • RIGHT JOIN (RIGHT OUTER JOIN) • Multi-table joins • Join conditions and syntax variations • Table aliases • Real-world examples
9	**Joins - Part 2 (Advanced)**	• FULL OUTER JOIN • Join order and performance implications • Handling NULLs in joins • Optimizing join performance • Using indexes for joins • EXPLAIN plan for joins • Complex multi-table scenarios • Join anti-patterns to avoid

10	**Self Join, Cross Join & Set Operations**	• Self Join concepts and use cases • Hierarchical data (employee-manager) • CROSS JOIN (Cartesian product) • Set operations: UNION (removes duplicates) • UNION ALL (keeps duplicates) • INTERSECT (common records) • EXCEPT (difference between sets) • Combining multiple operations
11	**Subqueries - Part 1**	• Introduction to subqueries • Scalar subqueries (single value) • Multi-row subqueries • Subqueries in WHERE clause • Subqueries in SELECT clause • Subqueries in FROM clause (derived tables) • IN, NOT IN operators • EXISTS, NOT EXISTS • Performance considerations
12	**Subqueries - Part 2 & CTEs**	• Correlated subqueries • ANY, SOME, ALL operators • Common Table Expressions (CTEs) • WITH clause syntax • Multiple CTEs in single query • Recursive CTEs • Base case and recursive case • Hierarchical queries (org charts, category trees) • Depth-first vs breadth-first
13	**Window Functions - Part 1 (Ranking)**	• Introduction to window functions • Difference from aggregate functions • OVER clause • PARTITION BY (dividing data) • ORDER BY in window functions • ROW_NUMBER() - sequential numbering • RANK() - ranking with gaps • DENSE_RANK() - ranking without gaps • Top N per category problems • Frame clause basics
14	**Window Functions - Part 2 (Analytics)**	• LEAD() - access next row • LAG() - access previous row • FIRST_VALUE(), LAST_VALUE(), NTH_VALUE() • SUM() OVER - running totals • AVG() OVER - moving averages • Window frames (ROWS BETWEEN, RANGE BETWEEN) • Time-series analysis • Period-over-period comparisons • Trend analysis • Percentiles (NTILE, PERCENT_RANK)

15	**Transactions & Error Control**	• Transaction fundamentals • BEGIN/START TRANSACTION • COMMIT • ROLLBACK • SAVEPOINT (creating savepoints) • Partial rollbacks (ROLLBACK TO SAVEPOINT) • RELEASE SAVEPOINT • Auto-commit mode • Practical order processing scenarios • Best practices
16	**ACID Properties & Exception Handling**	• Atomicity - all or nothing • Consistency - valid state transitions • Isolation - concurrent transactions • Durability - permanent changes • Isolation levels (READ COMMITTED, REPEATABLE READ, SERIALIZABLE) • Concurrency issues (dirty reads, phantom reads, deadlocks) • Exception handling in PL/pgSQL • BEGIN...EXCEPTION blocks • RAISE statements • DO blocks
17	**Normalization & Data Modeling**	• Database design principles • Data redundancy and anomalies • 1NF - atomic values, no repeating groups • 2NF - meet 1NF, no partial dependencies • 3NF - meet 2NF, no transitive dependencies • BCNF - stricter version of 3NF • Dependency diagrams • Denormalization strategies • When to denormalize • ER diagrams
18	**Views & Materialized Views**	• CREATE VIEW syntax • Benefits (abstraction, security, simplification) • Updatable views vs read-only • CREATE OR REPLACE VIEW • Dropping views • Row-level and column-level security • CREATE MATERIALIZED VIEW • Physical storage of query results • REFRESH MATERIALIZED VIEW • CONCURRENTLY option • Indexed materialized views

19	**Functions (PL/pgSQL)**	• Introduction to PL/pgSQL • CREATE FUNCTION syntax • Function structure • Parameters: IN, OUT, INOUT • DEFAULT values, VARIADIC • RETURN types (scalar, TABLE, SETOF, VOID) • DECLARE section • Variable declaration • BEGIN...END block • EXCEPTION blocks and error handling • RAISE statements • Function overloading
20	**Stored Procedures**	• CREATE PROCEDURE syntax • Difference from functions • IF-ELSE statements (simple, nested) • IF-ELSIF-ELSE • CASE statements in procedures • LOOP - basic loop with EXIT • WHILE - conditional loop • FOR - iterating over ranges • FOR IN - iterating over query results • CONTINUE and EXIT • Combining functions and procedures • Transaction control in procedures
21	**Query Optimization, Indexing & Security**	• EXPLAIN command • EXPLAIN ANALYZE • Reading execution plans • Identifying bottlenecks • B-Tree, Hash, GiST, GIN indexes • CREATE INDEX syntax • Unique, partial, multi-column indexes • When NOT to use indexes • REINDEX, VACUUM • SQL Injection prevention • Parameterized queries • CREATE USER/ROLE • GRANT and REVOKE • Information Schema queries

22	**RetailMart Analytics Project - Part 1**	• Project structure setup • 01_setup: create_analytics_schema.sql • Create metadata tables (kpi_metadata, log_table, refresh_history) • 02_kpi_queries/sales_analytics.sql: - Total sales by period - Sales trends and growth - Revenue by category/region - Top selling products - AOV, seasonal analysis • 02_kpi_queries/customer_analytics.sql: - Customer acquisition trends - CLV calculation - Retention and churn - RFM analysis - Customer segmentation
23	**RetailMart Analytics Project - Part 2**	• 02_kpi_queries/store_analytics.sql: - Store performance comparison - Store profitability - Employee productivity - Inventory turnover • 02_kpi_queries/product_analytics.sql: - Product category performance - Product profitability - Fast/slow moving products - Return analysis • JSON functions (json_agg, json_build_object, jsonb_set) • Creating JSON API functions • 04_documentation/export_all_json.sh • Data quality checks
24	**Dashboard Creation & Deployment**	• 03_dashboard/data: Export query results to JSON • 03_dashboard/index.html: - Semantic HTML5 structure - KPI cards, chart containers • 03_dashboard/css/styles.css: - Responsive grid layout - Modern card design - Mobile-responsive • 03_dashboard/js/dashboard.js: - Fetch JSON data - Chart.js integration - Interactive filters • Visualization (Line, Bar, Pie, Doughnut charts) • Deploy to GitHub Pages • project_overview.md documentation • Final presentation

RetailMart Database Architecture

Schema	Description	Key Tables
core	Dimension tables	dim_date, dim_region, dim_category, dim_brand
customers	Customer data	customers, addresses, reviews, loyalty_points
sales	Sales transactions	orders, order_items, payments, shipments, returns
products	Product catalog	products, suppliers, inventory, promotions
stores	Store operations	stores, departments, employees, expenses
hr	Human resources	attendance, salary_history
finance	Financial data	expenses, revenue_summary
marketing	Marketing campaigns	campaigns, ads_spend, email_clicks

Course Highlights

✓ **Industry-Ready:** Real-world business scenarios with proper schema design

✓ **Practical Focus:** Every concept practiced with RetailMart data

✓ **Analytical Emphasis:** Strong focus on window functions, CTEs, and analytics

✓ **Modern Tools:** PostgreSQL 16, pgAdmin 4, VS Code with SQLTools

✓ **Complete Workflow:** From design to optimization to visualization

✓ **Indian Context:** Examples from Flipkart, Amazon, Zomato, Swiggy

✓ **Career Focused:** Real-world project with dashboard deployment