

DATABASES

INTERMEDIATE

3 DAYS

PostgreSQL Development and Advanced Features

For developers who need PostgreSQL to do more than store rows. Three days covering PL/pgSQL functions and procedures, triggers, JSONB, arrays, ranges and custom types, full text search, advanced querying with LATERAL and window functions, transaction isolation, and the extension ecosystem including PostGIS and pgvector. Delivered on PostgreSQL 18 with labs written against a realistic application schema.

Why PostgreSQL Development Training?

Uses what you already pay for

Teams stop bolting on separate search, queue and document stores for capabilities PostgreSQL already provides.

Covers the AI workload path

Includes pgvector and similarity search for teams building retrieval features on their existing database.

Server-side logic done safely

Functions, procedures and triggers taught with the maintainability and debugging practices that keep them from becoming a liability.

Written for application developers

Every module connects back to application code, migrations and testing rather than staying inside psql.

What you'll be able to do

- ✓ Select and apply PostgreSQL data types beyond the standard set
- ✓ Write, debug and maintain PL/pgSQL functions and procedures
- ✓ Implement triggers and event-driven logic without creating hidden side effects
- ✓ Store, query and index JSONB documents effectively
- ✓ Use arrays, ranges, composite types and enumerations appropriately
- ✓ Build full text search with ranking, highlighting and language configurations
- ✓ Apply LATERAL joins, window functions and recursive CTEs to complex queries
- ✓ Choose transaction isolation levels and handle serialisation failures in application code
- ✓ Extend PostgreSQL using PostGIS, pgvector and other common extensions
- ✓ Integrate schema changes into migrations, tests and deployment pipelines

Who this is for

- Backend and full stack application developers
- Data engineers building on PostgreSQL
- Database developers moving from Oracle, SQL Server or MySQL
- Technical leads defining data access standards
- Engineers building search or retrieval features on PostgreSQL

Curriculum

01 Data Types Beyond the Basics

- Numeric, text and identifier types and the cost of choosing badly
- Timestamps, time zones and interval arithmetic
- Enumerations, domains and constrained value sets
- UUID generation options including uuidv7 and its index behaviour
- Generated and identity columns, including virtual generated columns
- **Hands-on:** refactor a schema to use appropriate types and measure the storage change

02 Functions and Procedures in PL/pgSQL

- Function structure, parameter modes and return types
- Procedures and transaction control inside the database
- Volatility categories and their effect on planning and indexing
- Security definer versus invoker and safe search_path handling
- Returning sets, tables and refcursors
- **Hands-on:** build and unit test a set of business logic functions

03 Control Flow, Cursors and Dynamic SQL

- Conditional logic, loops and exception blocks
- Explicit cursors and when row-by-row processing is justified
- EXECUTE and dynamic statement construction
- format, quote_ident and quote_literal for injection-safe dynamic SQL
- RAISE, logging and debugging techniques inside functions
- **Hands-on:** write a safe dynamic query builder and prove it resists injection

04 Triggers and Event-Driven Logic

- Row and statement level triggers, BEFORE, AFTER and INSTEAD OF
- Trigger functions, OLD and NEW records and conditional WHEN clauses
- Audit trails, denormalised counters and derived column maintenance
- LISTEN and NOTIFY for application event notification
- When trigger logic belongs in the application instead
- **Hands-on:** implement an audit trail and an event notification path

05 JSON and JSONB

- json versus jsonb storage, operators and trade-offs
- Path expressions, containment and existence operators
- SQL/JSON functions and the JSON_TABLE construct
- GIN indexing strategies and jsonb_path_ops
- Deciding what belongs in a column and what belongs in a document
- **Hands-on:** model a variable-attribute entity in JSONB and index it for real queries

06 Arrays, Ranges and Composite Types

- Array construction, unnesting and array operators
- Range and multirange types for periods, prices and reservations
- Exclusion constraints to prevent overlapping bookings
- Composite types and returning structured results
- Creating custom types and operators
- **Hands-on:** build a booking model that makes double booking impossible

07 Full Text Search

- tsvector, tsquery and text search configurations
- Dictionaries, stemming and stop words across languages
- Ranking, weighting and result highlighting
- GIN and GiST indexes for search workloads
- Trigram similarity and fuzzy matching with pg_trgm
- **Hands-on:** build a ranked search feature with typo tolerance

08 Advanced Query Techniques

- LATERAL joins for per-row subqueries
- Window functions for ranking, running totals and period comparison
- Recursive CTEs for hierarchies and graph traversal
- DISTINCT ON and other PostgreSQL specific shortcuts
- GROUPING SETS, ROLLUP and CUBE for subtotals
- **Hands-on:** replace application-side loops with single set-based queries

09 Transactions and Concurrency for Developers

- Isolation levels and the anomalies each one permits
- Serialisation failures and writing retry logic that works
- SELECT FOR UPDATE, SKIP LOCKED and building a job queue
- Advisory locks for application level coordination
- Upserts with ON CONFLICT and their concurrency behaviour
- **Hands-on:** build a concurrent-safe job queue and test it under load

10 Extensions and Specialised Workloads

- Installing, versioning and trusting extensions
- pgvector for embeddings, similarity search and index choices
- PostGIS fundamentals for spatial data and queries
- Foreign data wrappers for querying external systems
- Useful additions: pg_trgm, hstore, pgcrypto, pg_partman
- **Hands-on:** build a semantic search endpoint using pgvector

11 Integration, Migrations and Testing

- Connection pooling and prepared statements from application code
- Migration tooling and expand-contract schema change patterns
- Testing database logic and seeding reproducible fixtures
- Reading EXPLAIN output well enough to catch regressions in review
- Coding standards and review checklist for database changes
- **Hands-on:** capstone feature delivered with migrations, tests and a plan review