



PostgreSQL Database Administration: Performance Tuning, Replication and Recovery



PostgreSQL Database Administration: Performance Tuning, Replication and Recovery

Introduction:

Many PostgreSQL databases reach production with default settings, untested backups and indexes added only after users complain, so outages, bloat and slow queries become routine. This Core Concept course trains engineers in PostgreSQL database administration: how the server uses processes, memory and the write-ahead log, and how to design schemas, tune queries with EXPLAIN, manage MVCC and vacuum, lock down access, recover to a point in time and run replicas with failover. Participants leave with a PostgreSQL Production Readiness Runbook for a database they have designed, tuned and secured in the lab.

Course Objectives:

- Configure and size a PostgreSQL cluster from its process, memory, storage and write-ahead log architecture
- Design schemas, data types, constraints and partitions that hold up under production write and read loads
- Diagnose slow statements with EXPLAIN ANALYZE, planner statistics and index access methods and remove the bottleneck
- Control concurrency, MVCC bloat and transaction ID wraparound through isolation choices and autovacuum tuning
- Secure and recover a database with roles, row-level security, encrypted connections, logical backups and point-in-time recovery
- Operate streaming and logical replicas with automated failover and monitor workload health against defined KPIs

Target Audience:

- Database administration staff responsible for the availability, performance and recovery of PostgreSQL servers
 - Backend development staff who write schemas, queries and stored logic against PostgreSQL
 - Data engineering staff who load, transform and serve operational data held in PostgreSQL
 - DevOps and platform staff who provision, automate and monitor database infrastructure on premises or in the cloud
 - Site reliability staff who handle database incidents, failovers and capacity alerts
 - Migration project staff moving workloads from commercial database engines to PostgreSQL
-

Course Outline:

Day 1: PostgreSQL Server Architecture: Processes, Memory and Storage

- Postmaster, Backend and Background Worker Process Model
- Shared Buffers, work_mem and maintenance_work_mem Memory Areas
- Data Directory Layout: Tablespaces, Relation Files, Pages and TOAST
- Write-Ahead Log Records, Checkpoints and Crash Recovery Sequence
- Release Support Policy, Minor Updates and pg_upgrade Major Version Paths

Day 2: Installation, Configuration and Schema Design

- Package and Container Installation with initdb Cluster Initialisation
- postgresql.conf Parameter Contexts, ALTER SYSTEM and pg_hba.conf Authentication Rules
- Data Type Selection: numeric, timestamptz, uuid, Arrays, Ranges and Enums
- Constraints, Foreign Keys, Generated Columns and Declarative Partitioning
- Schemas, search_path and Naming Conventions for Multi-Application Databases

Day 3: Advanced SQL for Database Developers

- Recursive Common Table Expressions for Hierarchies and Graph Traversal
- Window Function Frames for Gaps, Islands and Moving Aggregates
- JSONB Operators, SQL/JSON Path Queries and Document-Relational Hybrid Models
- PL/pgSQL Functions, Procedures and Trigger Design
- INSERT ON CONFLICT Upserts and MERGE Statements

Day 4: Indexing Strategies and Query Tuning with EXPLAIN

- B-tree, Hash, GIN, GiST, SP-GiST and BRIN Index Access Methods
- Partial, Expression, Covering and Multicolumn Index Design
- EXPLAIN ANALYZE BUFFERS Plan Reading: Scan Nodes, Join Algorithms and Row Estimates
- Planner Statistics, ANALYZE, default_statistics_target and Extended Statistics
- pgbench Load Generation and auto_explain Slow Plan Capture

Day 5: Transactions, MVCC and Vacuum: Guided Case on a Busy OLTP Database

- Transaction Isolation Levels: Read Committed, Repeatable Read and Serializable
 - MVCC Tuple Visibility, xmin and xmax and Transaction ID Wraparound
 - VACUUM, Autovacuum Thresholds and Table Bloat Diagnosis with pgstattuple
 - Lock Modes, Deadlock Detection and pg_locks Contention Analysis
 - Guided Case: Diagnosing a Slow Order-Processing Database from Plans, Locks and Bloat
-

Day 6: Replication and High Availability

- Streaming Replication Build with `pg_basebackup` and Replication Slots
- Synchronous and Asynchronous Commit with `synchronous_standby_names`
- Logical Replication Publications and Subscriptions for Selective Sync and Low-Downtime Upgrades
- Automated Failover with Patroni and a Distributed Configuration Store
- Connection Pooling and Routing with PgBouncer and HAProxy

Day 7: Access Control, Row-Level Security and Recovery Assurance

- Role Hierarchy, Predefined Roles, GRANT and Default Privileges
- Row-Level Security Policies: Permissive, Restrictive and Default Deny
- SCRAM-SHA-256 Authentication, TLS Connections and `pgaudit` Session Logging
- Logical Backups with `pg_dump`, `pg_dumpall` and Parallel `pg_restore`
- Continuous WAL Archiving and Point-in-Time Recovery with `pgBackRest`

Day 8: Cloud-Managed PostgreSQL, Extensions and Change Delivery

- Managed Service Models: Amazon RDS and Aurora, Azure Database for PostgreSQL and Google Cloud SQL
- Shared Responsibility Boundaries and Parameter Groups in Managed Services
- Extension Overview: PostGIS, `pg_trgm`, `pg_partman` and TimescaleDB
- Schema Migration Pipelines with Flyway or Liquibase and Peer Review
- Maintenance Windows, Runbooks and Change Notices for Application Teams

Day 9: Monitoring, Database KPIs and Capacity Planning

- `pg_stat_statements` Workload Ranking by Total and Mean Execution Time
- Cumulative Statistics Views: `pg_stat_activity`, `pg_stat_io` and `pg_stat_replication`
- `postgres_exporter`, Prometheus and Grafana Dashboards for Database KPIs
- Log Analysis with `log_min_duration_statement` and `pgBadger` Reports
- Capacity Forecasts for Storage Growth, Connections and IOPS

Day 10: Capstone: Design, Tune and Secure a Production Database

- Capstone Brief: Multi-Tenant Transaction Database Requirements and Service Levels
 - Schema, Partitioning and Index Build for the Capstone Workload
 - Load Test and Tuning Iteration with `pgbench` and EXPLAIN Evidence
 - Security, Backup, Replica and Restore Test Configuration
 - PostgreSQL Production Readiness Runbook Presentation and Peer Review
-

Skills You Will Gain:

- PostgreSQL Server Configuration
- Relational Schema Engineering
- Execution Plan Analysis
- Index Design
- Autovacuum and Bloat Management
- Replication and Failover Operations
- Database Access Hardening
- Backup and Point-in-Time Recovery

Why Attend This Course:

- Build and tune a PostgreSQL database hands on in labs, from initdb through replication to a tested restore
- Find the real cause of slow queries, lock waits and bloat instead of adding hardware or random indexes
- Leave week one able to run a single server well, and week two able to run it replicated, secured, monitored and in the cloud
- Take back a PostgreSQL Production Readiness Runbook covering configuration, indexing, security, backup, failover and monitoring

Conclusion:

PostgreSQL rewards engineers who understand what happens beneath a query: the write-ahead log, MVCC row versions, the planner and vacuum. Week one builds that foundation, from architecture and configuration through schema design, advanced SQL, indexing and concurrency. Week two adds what a single-server course leaves out: replication and failover, access control and point-in-time recovery, cloud-managed services, extensions, change delivery and workload monitoring. The final day brings it together in a capstone database that participants design, tune, secure and document in their own runbook.
