

PostgreSQL Administration

Course Duration ~65 Hours · 9 Weeks · Live Online · Lab-Heavy · You Corrupt, Fail Over and Restore Real Clusters

Curriculum reviewed: July 2026 — the current major release changed operational practice; this syllabus is checked against the official PostgreSQL documentation

[Architecture](#)
[Tuning](#)
[Vacuum & Bloat](#)
[Pooling](#)
[PITR](#)
[Replication & HA](#)
[Vector Search](#)
[Migration](#)

In short: Own the database everything depends on — tune it, replicate it, secure it, and restore it under pressure, on the engine that now runs everything from transactions to vector search. This is **the data layer of the platform track**: Linux is the host, containers are the runtime, and this is where the state lives — the one component you cannot simply redeploy when it breaks. Free content teaches SQL and basic configuration; it doesn't teach what actually takes databases down, so this course is built around the hard part and around **recovery as the spine**: autovacuum, bloat and transaction ID wraparound; connection cost and pooling; locking and migrations under load; backup verification and point-in-time recovery as a graded drill; and reading execution plans as the primary diagnostic skill. It is a 2026 syllabus, not a 2016 one: PostgreSQL is now the default database for AI applications, so vector similarity search (pgvector) gets real teaching time — index types, build cost, recall-versus-latency, hybrid search — and the current major release's operational changes are taught with nuance (the new asynchronous I/O helps reads and maintenance, not write-heavy workloads; planner statistics now survive a major-version upgrade; skip scan, timestamp-ordered UUIDs, virtual generated columns, temporal/range constraints, token-based auth, and data checksums on by default for new databases). And because licence cost is driving sustained migration off Oracle and SQL Server, it covers assessment, schema/code conversion, the semantic differences that break naïve migrations, and cutover. On AI: it writes SQL fluently and ships exactly the mistakes that kill databases — the value is judgment about locking, plans, data volume and recovery, and there's a block on using AI tooling without shipping its errors. 9 weeks, ~65 hours, lab-heavy — you corrupt, fail over and restore real clusters.

Duration	9 weeks · ~65 hours · live + hands-on	Format	Live online · lab-heavy · you will corrupt, fail over and restore real clusters
Prerequisite	Basic SQL and comfort on the Linux command line. No prior database administration experience required.	Curriculum	10 modules, from the architecture to a defended production cluster
Certification	Aligned to the recognised vendor certification track (EDB); the field hires primarily on demonstrated operational ability	Ideal for	Backend developers who own a database but were never taught to operate one, DBAs, SRE/platform engineers, and commercial-database migrators

Anyone Can Install It. The Job Is Keeping It Alive.

Anyone can install PostgreSQL. The job is keeping it fast as the data grows, keeping it up when a node dies, and getting the data back when someone drops the wrong table — with the business waiting.

Database mistakes are the hardest in software to reverse, so recovery is the spine, not a final chapter. The operational failure modes that cause real incidents are foregrounded: **autovacuum, bloat and transaction ID wraparound** (the most characteristic PostgreSQL failure); **connection handling and pooling** (the architecture makes connections expensive, so pooling isn't optional at scale); **locking and migrations under load** (how a routine schema change stops an application dead); **backup verification and point-in-time recovery** (an untested backup is not a backup — restore drills are graded); and **reading query plans** as the primary diagnostic skill rather than guessing at indexes.

AI Writes SQL Fluently — And Ships the Mistakes That Kill Databases

AI writes SQL and schema definitions fluently, and produces exactly the things that destroy a production database later: missing or redundant indexes, migrations that take an exclusive lock on a large table, queries that look fine on ten rows and collapse on ten million, connection handling that exhausts the server, and schema choices that can't be undone once there's data in them. The value this course builds is **judgment about locking, plans, data volume and recovery** — what an assistant can't supply, because it can't see your data distribution, your lock waits

or your recovery objectives. A short block covers reading what AI proposes against an execution plan, a lock analysis and the real size of your tables before it reaches production.

A 2026 Syllabus, Not a 2016 One

- **Default database for AI applications** — vector similarity search (pgvector) gets real teaching time: index types, build cost and memory, recall-versus-latency, hybrid full-text + vector search, and the operational cost of adding this to a transactional database. Cross-linked to our GenAI, AI Agent and Prompt Engineering courses
- **The current major release changed operational practice**, taught with nuance: asynchronous I/O improves reads and maintenance (sequential scans, bitmap heap scans, vacuum) — *not* a step change for write-heavy workloads; planner statistics now survive a major-version upgrade (lower upgrade risk; extended statistics still need refreshing); skip scan widens multicolumn index usefulness; timestamp-ordered UUIDs fix the index-locality cost of random UUID keys; plus virtual generated columns, temporal/range constraints, token-based authentication, and data checksums enabled by default for new databases (checksum settings must match on upgrade)
- **Migration off commercial databases** — licence cost is driving sustained migration from Oracle, SQL Server and others: assessment, schema and code conversion, the semantic differences that break naïve migrations, and data-movement and cutover strategy. Funded, deadline-driven work

Where This Course Sits in the Track

- **Snowflake Data Warehouse** owns analytics at warehouse scale — this course owns operational/transactional databases; complementary, not competing
- **Docker & Kubernetes** covers stateful workloads on clusters conceptually — here we go deep on the database side and give an honest view of when running a database on Kubernetes is and isn't a good idea
- **Application courses (Django, FastAPI)** cover ORMs and data access from the application side — this is the server side: administration, performance, availability and recovery
- **Linux System Administration** is the host underneath — storage, systemd and OS tuning a database depends on

Course Highlights

- Every module leads with what you configure, tune, diagnose or recover — not what you "learn about"
- Recovery as the spine — graded restore drills with measured recovery objectives
- The failure modes nobody teaches — wraparound, bloat, lock storms, connection exhaustion
- Execution plans taught as a method, not a guessing game
- Schema migrations that don't lock a production table under live load
- Rehearsed failover and a real major-version upgrade via logical replication
- Vector search workloads — index tuning and hybrid retrieval alongside relational data
- A migration-off-commercial-database assessment and cutover plan

Hands-on Projects & Portfolio

- **Build & Baseline a Cluster** — from scratch, with a justified configuration for a given workload
- **Query Optimisation Workshop** — cut a slow application's worst queries substantially, with execution-plan evidence before and after
- **Bloat & Autovacuum Lab** — create the problem deliberately, detect it, remediate it without downtime
- **Zero-Downtime Migration** — apply schema changes to a large table under live load
- **Connection Pooling Build-Out** — sizing justified by measurement
- **Disaster Recovery Drill** — point-in-time restore to a moment just before a destructive change, with a measured recovery time
- **HA Cluster & Failover** — automated failover, then a rehearsed failover under load
- **Upgrade via Logical Replication** — a major-version upgrade with minimal downtime
- **Vector Search Implementation** — index tuning and hybrid retrieval alongside relational data
- **Commercial-DB Migration Assessment** — convert a schema and produce a cutover plan
- **Capstone** — a production-grade cluster, tuned, replicated, secured, monitored and defended

Tools & Stack

PostgreSQL · psql · pgAdmin · pg_stat_statements · pgBouncer · pgBackRest · Patroni · pgvector · PostGIS · Citus · pg_dump/pg_restore · pg_upgrade · EXPLAIN/ANALYZE · pgbench · Prometheus · Grafana · postgres_exporter · Ansible · Docker · Kubernetes operators · AWS RDS & Aurora · Cloud SQL · Azure Database for PostgreSQL · Linux

Certification Alignment

- Aligned to the recognised vendor-backed PostgreSQL certification track from **EDB (EnterpriseDB)**, which offers tiered associate, professional and master-level credentials for production database administration
- An honest note: this field hires primarily on **demonstrated operational ability** — a cluster you have tuned, replicated, broken and recovered speaks louder than any certificate, which is why this course produces exactly that evidence
- Presented as preparation and alignment only — no official partnership or accreditation implied, and no exam outcome guaranteed

Career Outcomes

PostgreSQL has become the default open-source relational database for new applications, and its extension ecosystem has extended that into geospatial, time-series and now vector and AI workloads — one skill set covers a widening range of systems. Licence costs are driving sustained migration from proprietary databases, creating steady demand for people who can plan and execute those moves. And database expertise is unusually durable: the fundamentals of storage, concurrency, locking and recovery outlast any framework cycle. Roles: Database Administrator · PostgreSQL DBA · Database Reliability Engineer · Data Platform Engineer · DevOps/SRE with database ownership · Backend Engineer owning the data layer · Cloud Database Engineer · Migration Consultant.

Who This Course Is For

- Final-year students and fresh IT graduates targeting database or platform roles
- **Backend developers who own a production database but were never taught to operate one** — the largest and most common gap this course closes
- DevOps, SRE and platform engineers inheriting database responsibility
- **Oracle, SQL Server and other commercial database professionals migrating to open source** — funded, deadline-driven work
- Data engineers who need operational depth beyond writing queries
- **AI and application engineers running vector search on PostgreSQL** who need to operate it properly
- Anyone who has restored from a backup for the first time during an actual incident

Module 1: Architecture That Explains Everything Else

6 Hours

YOU CONFIGURE

- A cluster from the ground up — and, more importantly, the mental model that makes every later decision about memory, locking and recovery obvious

TOPICS

- The process model and why connections are expensive; shared buffers and the OS page cache; the write-ahead log; checkpoints
- Multiversion concurrency control and what it implies for storage; transaction isolation levels in practice; the system catalogs
- Installation and cluster layout

Assessment: Quiz + Lab

Module 2: Configuration & Day-One Operations

6 Hours

YOU CONFIGURE

- A sane baseline for a new server — memory, logging, roles and extensions set deliberately, not copied from a blog

TOPICS

- The configuration files; reload-versus-restart parameters; memory settings that matter and the ones people copy blindly
- Logging configuration for diagnosis rather than noise; extensions; roles and databases

- A sane baseline for a new server

Assessment: Quiz + Lab

Module 3: Schema, Data Modelling & Migrations

7 Hours

YOU CONFIGURE

- Schema migrations that don't lock a production table — the change patterns that keep an application online while the structure underneath it changes

TOPICS

- Data types including JSON and arrays used well; constraints; primary key strategy and why random UUIDs hurt index locality; generated and temporal/range constraint features
- Indexing strategies across the available index types; partial and expression indexes; partitioning
- Schema migrations that don't lock a production table

Assessment: Quiz + Lab

Module 4: Query Performance

7 Hours

YOU DIAGNOSE

- A slow query taken apart with a systematic method — reading the plan, fixing the cause, and proving the improvement with evidence rather than folklore

TOPICS

- Reading execution plans properly; planner statistics and how they go wrong; index selection and skip scan; join strategies
- Common inefficient patterns; query rewriting; extended statistics
- Tracking slow queries with the standard statistics extension; a systematic method for taking a slow query apart

Assessment: Quiz + Lab

Module 5: Vacuum, Bloat & Maintenance

7 Hours

YOU DIAGNOSE

- Transaction ID wraparound before it becomes an emergency — and remediate table and index bloat without taking the application down

TOPICS

- How multiversion concurrency creates dead tuples; autovacuum tuning for real workloads; table and index bloat detection and remediation
- Transaction ID wraparound and how to avoid the emergency; reindexing without downtime; analyse and statistics maintenance
- Long-running transactions as the root of most maintenance problems

Assessment: Quiz + Lab

Module 6: Connections, Pooling & Concurrency

6 Hours

YOU DIAGNOSE

- A blocked-query pile-up and an idle-in-transaction leak — then size a pool against real resources so neither happens again

TOPICS

- Connection cost; pooling with an external pooler; pooling modes and their trade-offs; sizing pools against available resources
- Lock types and lock waits; deadlocks; blocked-query diagnosis
- Idle-in-transaction as a silent killer

Assessment: Quiz + Lab

Module 7: Backup, Recovery & Business Continuity

7 Hours

YOU RECOVER

- A database to a moment just before a destructive change — a graded point-in-time restore with a measured recovery objective, because an untested backup is not a backup

TOPICS

- Logical versus physical backup; continuous archiving; point-in-time recovery; backup tooling for large databases
- Retention and verification policy; tested restore drills with measured recovery objectives; corruption detection with checksums
- Recovering from the incidents that actually happen — a dropped table, a bad deployment, a lost node

Assessment: Quiz + Lab (Graded Restore Drill)

Module 8: Replication & High Availability

7 Hours

YOU RECOVER

- Service continuity through a node failure — a rehearsed failover under load, treated as a drill rather than a theory

TOPICS

- Streaming replication and replica configuration; synchronous versus asynchronous trade-offs; replication slots and their risks; replication lag and monitoring
- Logical replication for selective and cross-version replication; automated failover and cluster management; connection routing and read scaling
- Split-brain avoidance; rehearsed failover as a drill rather than a theory

Assessment: Quiz + Lab

Module 9: Security, Upgrades & Compliance

6 Hours

YOU CONFIGURE

- A hardened, access-controlled server — then plan and rehearse a major-version upgrade with a minimal-downtime approach and post-upgrade validation

TOPICS

- Authentication methods including certificate and modern token-based options; host-based access configuration; roles and privileges; row-level security
- Encryption in transit and at rest; auditing; secrets handling; patch strategy
- Major-version upgrade planning — in-place upgrade, logical replication cutover and minimal-downtime approaches; post-upgrade validation

Assessment: Quiz + Lab

Module 10: Scale, Cloud, Vector Workloads & Capstone

6 Hours

YOU DELIVER

- One cluster taken from installation to tuned, replicated, monitored, secured and recoverable — and defended — after adding a vector-search workload and weighing managed cloud against self-hosted

TOPICS

- Monitoring and alerting with the standard observability stack; capacity planning; sharding and distributed extensions; read replicas
- Vector search workloads: index types, build cost, recall-versus-latency tuning and hybrid search alongside relational data; managed cloud services versus self-hosted — what managed hides and what it doesn't
- Running databases on Kubernetes and when not to; migration from commercial databases; capstone: take one cluster to tuned, replicated, monitored, secured, recoverable — and defend it

Assessment: Capstone Project + Defence