Local AI Development, PostgreSQL Migration & Intelligent Document Search

Duration: 15 Days | 120 Hours

Week 1: PostgreSQL Migration, Local AI Infrastructure & RAG

Day 1: Oracle to PostgreSQL Migration – Strategy & Setup

Module 1: Migration Planning & Architecture

- Oracle vs PostgreSQL
- Licensing cost analysis
- Data sovereignty strategy
- Migration approaches
- Risk assessment
- Rollback planning
- Team roles

Lab Exercises:

1. Audit a sample Oracle schema; identify objects that need conversion and classify by complexity
2. Build a phased migration plan for 3 enterprise systems; document risks and rollback strategy for each

Module 2: PostgreSQL Installation & Configuration

- PostgreSQL architecture
- Ubuntu/RHEL installation
- postgresql.conf tuning
- Tablespaces and roles
- PgBouncer pooling
- SSL encryption
- Backup configuration

Lab Exercises:

3. Install and configure PostgreSQL on a Lab VM; apply production-grade security settings
4. Configure PgBouncer for connection pooling; test with 50 concurrent connections

Day 2: Oracle to PostgreSQL – Schema & Data Migration

Module 3: Schema Conversion & SQL Migration

- ora2pg conversion
- Data type mapping
- Stored procedures
- Oracle to standard SQL
- Indexes and constraints
- View migration
- Validation checklist

Lab Exercises:

5. Run ora2pg on a sample Oracle schema; review the conversion report and fix 5 incompatible objects
6. Migrate stored procedures from Oracle PL/SQL to PostgreSQL PL/pgSQL; test each for identical output

Module 4: Data Migration & Validation

- Full load with ora2pg
- Incremental replication
- Data type mismatches
- Row count validation
- Integrity checks
- Connection string updates
- Performance comparison

Lab Exercises:

7. Migrate a 1-million-row financial table; validate row counts and checksums
8. Compare execution plans against migrated PostgreSQL database; fix slow queries

Day 3: PostgreSQL for AI & Vector Search

Module 5: pgvector for AI-Ready PostgreSQL

- pgvector extension
- Vector data types
- Similarity metrics
- Vector indexes
- Embedding storage
- Hybrid search
- pgvector tuning

Lab Exercises:

9. Install pgvector; store 500 document embeddings and run a semantic similarity search query
10. Build a hybrid search query combining full-text and vector similarity on a policy document table

Module 6: PostgreSQL Full-Text Search & Data Warehousing

- tsvector and tsquery
- GIN indexes
- Multilingual search
- Table partitioning
- Materialized views
- EXPLAIN ANALYZE
- Query optimisation

Lab Exercises:

11. Build a full-text search index on 10,000 HR policy documents; measure and optimise query speed
12. Partition a transactions table by year; compare query performance before and after

Day 4: Local AI Infrastructure – Deploying Open-Source Models

Module 7: Local AI Infrastructure & GPU Requirements

- Centralised AI services
- GPU requirements
- NVIDIA CUDA setup
- Ollama deployment
- vLLM deployment
- Model quantisation
- Air-gapped deployment

Lab Exercises:

13. Deploy Llama 3 using Ollama on a GPU server; verify zero outbound network traffic during inference
14. Deploy the same model using vLLM; compare throughput and latency against Ollama

Module 8: Managing Open-Source AI Models Locally

- Open-source model selection
- GGUF and GPTQ formats
- Model versioning
- Fine-tuning vs RAG
- LoRA fine-tuning
- Training data preparation
- Model evaluation

Lab Exercises:

15. Prepare a training dataset from 200 internal policy documents; format for fine-tuning
16. Run a LoRA fine-tune on Llama 3; evaluate on 20 domain-specific questions

Day 5: RAG – Intelligent Document Search on Internal Documents

Module 9: RAG Architecture & Document Pipeline

- RAG pipeline overview
- Document loaders
- Text chunking strategies
- Local embedding models
- ChromaDB and pgvector
- Semantic retrieval
- Re-ranking

Lab Exercises:

17. Build a RAG pipeline ingesting 50 HR policy PDFs; store embeddings in pgvector
18. Test retrieval with 15 employee questions; identify retrieval failures and fix chunking strategy

Module 10: Secure RAG with PostgreSQL Backend

- All data in PostgreSQL
- Role-based access control
- Audit logging
- Document classification
- Department data isolation
- Query result filtering
- RAG evaluation metrics

Lab Exercises:

19. Implement role-based access: HR staff see HR docs only, Finance staff see Finance docs only
20. Enable audit logging; run 10 queries and verify every query is logged with user and timestamp

Week 2: AI Agents, Document Intelligence, FastAPI & AI Testing

Day 6: Intelligent Agents & Automated Workflows

Module 11: Building Intelligent Agents

- Intelligent agents overview
- Agent architecture
- Tool integration
- Agent memory
- Automated workflows
- LangChain agents
- Agent logging

Lab Exercises:

21. Build an intelligent agent that queries PostgreSQL and generates a summary report automatically
22. Add tool integration: file reader, database connector and API caller as agent tools

Module 12: Multi-Agent Systems & LangGraph

- Multi-agent systems
- LangGraph stateful graphs
- MCP integration
- Orchestrator agents
- Agent communication
- Parallel execution
- Error handling

Lab Exercises:

23. Build a multi-agent system: one agent fetches data, another validates, a third generates the report
24. Implement a LangGraph stateful workflow with MCP tool integration; test state persistence across steps

Day 7: Secure Multilingual Chatbot Development

Module 13: Chatbot Development – Internal Documents

- Secure chatbot architecture
- Multilingual chatbots
- Internal document access
- Reliable responses
- Context management
- Conversation memory
- Open WebUI deployment

Lab Exercises:

25. Deploy a secure multilingual chatbot connected to local Llama 3 and pgvector document store
26. Test the chatbot in two languages; verify it retrieves only authorised internal documents

Module 14: AnythingLLM & Chatbot Security

- AnythingLLM setup
- Multi-user sessions
- Authentication controls
- Access boundaries
- Response filtering
- Internal DNS setup
- Docker deployment

Lab Exercises:

27. Deploy AnythingLLM with role-based access; connect to local model and internal document store
28. Configure authentication; verify one user cannot access another department's document responses

Day 8: Document Intelligence – OCR & Scanned Document Processing

Module 15: Document Intelligence & OCR

- Document intelligence overview
- OCR processing
- Scanned document processing
- Document classification
- Document summarisation

- Multilingual document search
- Table and form extraction

Lab Exercises:

29. Process 30 scanned PDF documents using OCR; extract text and store in PostgreSQL
30. Build a document classification pipeline that tags each document by type, language and department

Module 16: Advanced Document Processing Pipeline

- PDF and Word loaders
- Image document handling
- Arabic text processing
- Multilingual embeddings
- Chunk and summarise
- Metadata extraction
- Pipeline automation

Lab Exercises:

31. Build an end-to-end document processing pipeline: ingest scanned docs, OCR, classify, embed and store
32. Run multilingual document search across Arabic and English documents; compare retrieval accuracy

Day 9: Python, FastAPI & REST API Development

Module 17: Python & FastAPI for AI Applications

- Python for AI apps
- FastAPI framework
- REST APIs
- Pydantic data models
- Async endpoints
- JWT authentication
- API documentation

Lab Exercises:

33. Build a FastAPI REST API exposing a RAG endpoint; secure with JWT and document with Swagger
34. Add async endpoints for document upload and query; test with 10 concurrent requests

Module 18: Integrating AI into Existing Applications

- Laravel/PHP integration
- Existing application integration
- API gateway patterns
- Webhook callbacks
- Data transformation
- Error handling
- Integration testing

Lab Exercises:

35. Connect a Laravel/PHP application to the FastAPI AI backend; call the RAG endpoint from PHP
36. Build a webhook integration; trigger AI summarisation when a new document is uploaded from PHP

Day 10: AI Testing & Evaluation

Module 19: AI Testing & Model Evaluation

- AI testing strategy
- Comparing AI models
- Measuring response accuracy

- Identifying hallucinations
- Reliability testing
- Benchmarking models
- Evaluation datasets

Lab Exercises:

37. Compare three open-source models on 20 domain questions; score accuracy and hallucination rate
38. Build an automated evaluation pipeline that scores RAG responses against ground-truth answers

Module 20: Testing AI Application Reliability

- End-to-end testing
- Load testing AI
- Regression testing
- Response consistency
- Failure scenario testing
- Test coverage reports
- CI test integration

Lab Exercises:

39. Run load tests on the AI application; measure response time and accuracy under 50 concurrent users
40. Write an automated regression test suite; verify model responses remain consistent after updates

Week 3: AI Security, Governance, Deployment & Capstone

Day 11: AI Security – Securing AI Agents & Institutional Data

Module 21: Securing AI Agents

- Securing AI agents
- Institutional data control
- Preventing prompt injection
- Proper audit trails
- Agent sandboxing
- Zero trust for AI
- OWASP LLM Top 10

Lab Exercises:

41. Attempt prompt injection on the AI agent; implement and verify sanitisation fixes
42. Audit the full AI stack against OWASP LLM Top 10; produce a risk register with mitigations

Module 22: Access Control & Data Security

- PostgreSQL row-level security
- Column-level encryption
- Secrets management
- Network segmentation
- API security
- Penetration testing
- Security audit checklist

Lab Exercises:

43. Enable row-level security; verify staff access only their department's records
44. Run a penetration test on AI endpoints; fix the top 3 findings and re-verify

Day 12: AI Governance – Policies, Risks & Model Management

Module 23: AI Governance & Institutional Policies

- AI risks overview
- Institutional policies
- Responsible AI use
- Locally deployed models
- Acceptable use policy
- Vendor independence
- Governance committee

Lab Exercises:

45. Draft an AI governance policy for the institution; include prohibited uses, oversight and escalation
46. Design a model lifecycle process: approval, versioning, monitoring, retirement criteria

Module 24: Managing Locally Deployed Models

- Model versioning
- Change management
- Human in the loop
- Model retirement
- Staff AI literacy
- Open-source strategy
- Review cadence

Lab Exercises:

47. Set up a model version registry; deploy a new model version and roll back without downtime
48. Human-in-the-loop review gate for high-risk AI responses; test the override process

Day 13: Monitoring, Observability & Production Deployment

Module 25: AI System Monitoring & Observability

- Prometheus metrics
- Grafana dashboards
- Response quality monitoring
- Model drift detection
- Alert configuration
- Log aggregation
- Incident runbooks

Lab Exercises:

49. Set up Prometheus to monitor Ollama/vLLM inference latency, GPU memory and PostgreSQL query time
50. Build a Grafana dashboard showing AI system health, chatbot usage and error rates with alerts

Module 26: Production Deployment & High Availability

- Docker Compose stack
- Nginx reverse proxy
- PostgreSQL replication
- Model server scaling
- Health checks
- Disaster recovery
- Deployment runbook

Lab Exercises:

51. Deploy the full AI stack with Docker Compose; verify all services connect and health checks pass
52. Simulate a database failover; verify the application reconnects automatically within 30 seconds

Day 14: Capstone Preparation - AI Environment Design

Module 27: Capstone Architecture & Planning

- Requirements analysis
- Architecture design
- Technology selection
- Security planning
- Integration mapping
- Testing strategy
- Deployment plan

Lab Exercises:

53. Design the full capstone architecture: locally hosted AI model, chatbot, agent, database API – draw and document
54. Define acceptance criteria for each component; assign responsibilities and set up the development environment

Module 28: Component Build – Local Model & Chatbot

- Locally hosted AI model
- Internal chatbot
- Document store connection
- Multilingual support
- Authentication setup
- Audit trail setup
- Integration testing

Lab Exercises:

55. Deploy the locally hosted AI model and connect the internal chatbot to internal documents
56. Test the chatbot with 15 real queries across two languages; verify accuracy and access control

Day 15: Capstone – Working AI Environment & Presentation

Module 29: Capstone Build – AI Agent & API Integration

- AI agent connected to authorised database
- API integration with existing applications
- FastAPI AI endpoint
- Laravel/PHP connection
- End-to-end testing
- Security validation
- Performance benchmarking

Lab Exercises:

57. Complete the working AI environment: locally hosted model + internal chatbot + AI agent connected to authorised database + API integrated with existing applications
58. Run full acceptance testing across all components; fix issues and verify all acceptance criteria are met