AI ENGINEER

SQL • Python • ML • NLP • AWS • GenAI • Agentic AI

SQL Foundations

DAY 1 SQL & DBMS Intro

- SQL intro, DBMS vs RDBMS
- DDL, DML, DQL, TCL, DCL commands
- Relational concepts: Understand tables, rows, columns, keys and relationships; compare transactional databases with analytical use cases.
- Command practice: Create a small schema and distinguish schema changes, row changes, queries, transaction control and access permissions.
- Lab: Explore an employee dataset, inspect table definitions and write basic SELECT statements with clear column aliases.

DAY 2 Constraints & Operations

- Data types, constraints (PK, FK, UNIQUE)
- Create/Drop databases, import data
- Schema design: Choose suitable numeric, character and date types; apply NOT NULL, CHECK and DEFAULT alongside primary, foreign and unique keys.
- Data operations: Create and remove training databases safely; load CSV files and diagnose delimiter, encoding, type-conversion and constraint errors.
- Lab: Build customer and order tables, insert valid records and test how constraints reject invalid or duplicate data.

DAY 3 Filtering & Aggregations

- WHERE clause, logical operators
- ORDER BY, LIMIT, aggregate functions
- Filtering: Combine AND, OR and NOT with comparison operators, IN, BETWEEN, LIKE and explicit NULL checks.
- Summaries: Use COUNT, SUM, AVG, MIN and MAX with GROUP BY and HAVING; distinguish filtering rows from filtering groups.
- Lab: Rank products by sales, calculate category totals and return deterministic top-N results using sorting and limits.

DAY 4 Joins & Set Ops

- Inner, left, right, full joins
- UNION, INTERSECT, EXCEPT
- Join reasoning: Identify one-to-one and one-to-many relationships; trace unmatched rows, duplicate matches and NULL values in join results.
- Set semantics: Compare UNION with UNION ALL and use INTERSECT and EXCEPT where supported by the selected SQL engine.
- Lab: Combine customers, orders and payments, then reconcile totals to detect accidental row multiplication.

DAY 5 Functions & Subqueries

- Subqueries in clauses
- Text, date/time, window functions
- Query patterns: Write scalar, multi-row and correlated subqueries; compare IN and EXISTS for common filtering tasks.
- Function practice: Clean strings, calculate date intervals and use ROW_NUMBER, RANK, LAG and running totals with explicit window definitions.
- Lab: Find each customer's latest order and compare monthly revenue with the previous month.

DAY 6 Advanced Objects

- CTEs, Views, Indexing strategies
- Stored procedures basics
- Readable SQL: Break complex queries into CTEs and create views for reusable logic; understand scope and recursion at an introductory level.

Greens Technologies

- Performance basics: Inspect query plans, select useful indexes and discuss read benefits versus storage and write overhead.
- Lab: Create a parameterized stored procedure in the chosen database dialect and compare a query before and after indexing.

DAY 7 Doubt Clearance

- Sample interview questions
- Query optimization tips
- Review clinic: Revisit joins, grouping, subqueries, windows and NULL handling through short debugging and interview-style exercises.
- Optimization workflow: Check correctness first, inspect plans and row counts, then reduce unnecessary columns, scans and repeated computations.
- Assessment: Solve a mini reporting case and explain query choices, assumptions and performance trade-offs.

Python Programming & Data Manipulation

DAY 8 Basics & Syntax

- Setup, data types, variables, operators
- Simple programs, I/O
- Environment: Create a virtual environment, install packages and use an editor or notebook; understand scripts, imports and basic debugging.
- Language essentials: Practice numbers, strings, booleans, casting, comparison, branching and loops with simple validation rules.
- Lab: Build a command-line calculator and a small input-processing program with readable names and helpful error messages.

DAY 9 Data Structures

- Strings, lists, tuples, sets, dicts
- Hands-on exercises
- Collection behavior: Compare mutability, indexing, slicing, membership and common methods across Python's built-in data structures.
- Practical patterns: Use comprehensions, unpacking, iteration and dictionary-based lookups; handle missing keys and empty collections.
- Lab: Clean a list of customer records, remove duplicates and calculate grouped counts using lists, sets and dictionaries.

DAY 10 Functions & Files

- User-defined, anonymous functions
- File I/O, Regex basics
- Email extraction project
- Reusable code: Define function arguments, return values and scope; use lambda expressions appropriately and introduce exception handling.
- File processing: Read and write text, CSV and JSON with context managers; use regular expressions for matching and extraction.
- Project: Extract email-like strings from sample files, normalize and deduplicate results, flag invalid entries and save a structured output.

DAY 11 Numpy & Pandas

- Arrays, indexing, broadcasting
- Pandas Series & DataFrames
- NumPy foundations: Understand shapes, dimensions, dtypes, vectorized operations, boolean masks and broadcasting compatibility.
- Pandas workflow: Load datasets, select rows and columns, inspect types and calculate grouped summaries with clear index handling.
- Lab: Transform a sales dataset using arrays and DataFrames, then validate totals and export a clean analysis table.

DAY 12 Advanced Analytics

- Missing data handling, duplicates

Greens Technologies

- Matplotlib & Seaborn visualization
- Merging, joining, pivot tables
- Data quality: Profile missing values and duplicates; select an imputation or removal strategy and document how it affects results.
- Combining data: Use merge keys, join validation, groupby and pivot tables while checking row counts and aggregation consistency.
- Visualization lab: Create labeled distribution, relationship and comparison plots with Matplotlib and Seaborn; summarize findings without overstating causation.

Cloud Data Engineering (AWS)

DAY 13 Storage & IAM

- IAM security, S3 storage
- Snowflake modern warehouse architecture
- AWS security: Work with IAM users, roles and policies; apply least privilege and avoid placing long-lived credentials in notebooks or code.
- S3 foundations: Organize buckets and prefixes, manage object access, encryption, versioning and lifecycle policies using a sample data lake layout.
- Warehouse concepts: Explain Snowflake's separate storage and compute layers, virtual warehouses and databases; compare its role with object storage.

DAY 14 Compute & Monitoring

- AWS Lambda serverless functions
- CloudWatch Logs
- EventBridge automation
- Serverless compute: Configure Lambda inputs, execution roles, timeouts and dependencies; design handlers for duplicate events and safe retries.
- Observability: Read structured CloudWatch logs, define useful metrics and alarms, and trace failures across a small event-driven workflow.
- Lab: Trigger a sample Lambda through EventBridge; review permissions, failure handling and expected AWS charges before running resources.

DAY 15 ETL & Warehousing

- AWS Glue (Crawlers, jobs, pipelines)
- Amazon Redshift Warehousing
- Data discovery: Use Glue Crawlers and the Data Catalog to describe source schemas and detect schema changes in sample S3 data.
- ETL design: Build Glue jobs to clean, transform and partition data; understand job bookmarks, scheduling and basic data-quality checks.
- Warehouse lab: Load transformed data into Redshift, inspect table design and run analytical queries; compare warehouse loading with lake storage.

DAY 16 Orchestration & ML Ops

- AWS Step Functions Orchestration
- AWS SageMaker for ML Pipelines
- Workflow building labs
- Workflow orchestration: Model Step Functions states, branching, retries and error paths; pass task outputs explicitly between processing stages.
- ML operations: Introduce SageMaker processing, training and evaluation steps, pipeline parameters, model artifacts and model-registry approval.
- Lab: Connect ingestion, validation, training and evaluation in a small workflow; record outputs, monitor failures and clean up billable resources.

Machine Learning Engineering

DAY 17 Regression Basics

- Supervised learning, Linear Regression
- Metrics & Evaluation

Greens Technologies

- Learning workflow: Define features and targets, create train/validation/test splits and prevent target leakage before fitting a baseline.
- Regression models: Fit simple and multiple linear regression; interpret coefficients, residuals, scaling and introductory regularization.
- Evaluation lab: Compare MAE, RMSE and R-squared against a simple baseline; explain why a strong training score may not generalize.

DAY 18 Decision Trees

- Trees, ensembles, Random Forests
- Tree mechanics: Understand split criteria, tree depth, minimum leaf size and overfitting in classification and regression trees.
- Ensembles: Compare a single tree with bagging and Random Forests; discuss feature sampling, validation and model-size trade-offs.
- Lab: Tune tree complexity, review confusion-matrix or regression metrics and interpret feature importance with its limitations.

DAY 19 Clustering

- K-Means, DBSCAN Techniques
- Unsupervised workflow: Scale suitable features, choose distance measures and distinguish clustering from supervised classification.
- Algorithm behavior: Explore K-Means initialization and cluster counts; compare DBSCAN's density parameters, noise labels and irregular cluster shapes.
- Lab: Build customer segments, inspect silhouette scores and visualize clusters; check stability and usefulness before assigning business labels.

DAY 20 Feature Engineering

- PCA & Dimensionality Reduction
- Feature preparation: Encode categories, scale numeric inputs, transform skewed variables and create domain-informed features without leaking future information.
- PCA concepts: Explain variance, principal components and explained-variance ratios; fit transformations only on training data.
- Lab: Compare a model with original features against a PCA-based model and discuss interpretability, computation and predictive performance.

DAY 21 ML Pipelines

- Hyperparameter Tuning, Pipelines
- Pipeline design: Chain preprocessing and estimators with scikit-learn Pipeline and ColumnTransformer so cross-validation includes every fitted transformation.
- Tuning strategy: Use grid or randomized search with an appropriate scoring metric and split strategy; keep the final test set untouched.
- Lab: Save a fitted pipeline, reload it for consistent prediction and record parameters, metrics and dataset versions for reproducibility.

Natural Language Processing

DAY 22 NLP Fundamentals

- Text preprocessing, POS tagging
- Text preparation: Compare tokenization, case normalization, punctuation handling, stop-word removal and lemmatization for different NLP tasks.
- Linguistic structure: Introduce POS tags and sentence boundaries; inspect how domain vocabulary and ambiguous words affect preprocessing choices.
- Lab: Prepare a sample support-ticket corpus and compare preprocessing choices without removing useful negation or sensitive contextual signals.

DAY 23 Feature Extraction

- Bag of Words, TF-IDF, Embeddings
- Sparse features: Build count and TF-IDF matrices with n-grams, vocabulary limits and consistent fit/transform behavior across data splits.

Greens Technologies

- Dense representations: Introduce word and sentence embeddings, cosine similarity and semantic search; contrast dense similarity with exact keyword matching.
- Lab: Train a baseline text classifier and build a small similarity search, then inspect errors and vocabulary coverage.

DAY 24 Advanced Transformers

- SpaCy NER, Transformer models (BERT) overview
- Named entities: Use spaCy pipelines to detect people, organizations and locations; inspect spans, labels and domain-specific extraction errors.
- Transformer foundations: Explain tokenization, attention, contextual representations and BERT's encoder architecture at a conceptual level.
- Lab: Run a pretrained transformer for classification or embeddings, compare it with a simpler baseline and discuss latency and hardware needs.

Generative AI & Agentic AI

DAY 25 LLM Core

- BERT & GPT concepts
- Evaluation & Fine-tuning basics
- Model families: Contrast BERT-style encoder objectives with GPT-style autoregressive generation; introduce tokens, context windows and generation settings.
- Prompting and evaluation: Define task-specific test sets and rubrics for correctness, groundedness, safety, latency and cost rather than relying on fluent output.
- Adaptation basics: Compare prompting, RAG and fine-tuning; discuss training-data quality, held-out evaluation, privacy and when adaptation is unnecessary.

DAY 26 GenAI Apps

- LangChain, Conversational memory
- Application building: Use LangChain model interfaces, prompt templates, runnable chains and structured outputs; validate outputs before downstream use.
- Conversation state: Manage chat history through explicit session state and checkpoints; compare short-term context, summaries and persistent memory.
- Lab: Build a small assistant with bounded history, error handling and usage logging; avoid retaining sensitive conversation data without a clear need.

DAY 27 RAG Architectures

- Vector DBs (FAISS, ChromaDB)
- RAG systems with LLaMA3
- Retrieval pipeline: Load documents, split text, generate embeddings and index vectors; compare FAISS local indexing with ChromaDB collection management.
- Grounded generation: Connect retrieval to a LLaMA3-based generator, return source citations and add abstention when evidence is missing or contradictory.
- Lab: Evaluate retrieval recall and answer groundedness separately; test chunk sizes, top-k, metadata filters and optional reranking while tracking latency.

DAY 28 Agentic AI

- Independent AI Agents
- Agent design: Explain goal-directed loops, planning, tool selection and state; distinguish deterministic workflows from autonomous model-driven choices.
- Control boundaries: Use typed tool schemas, least-privilege access, step and cost limits, human approval for consequential actions and clear stop conditions.
- Lab: Build a bounded agent over mock tools, trace each action and test tool failures, prompt injection, retries and unsupported requests.

DAY 29 Deployment

- Fine-tuning techniques
- MCP Server deployment
- Fine-tuning workflow: Introduce supervised fine-tuning, LoRA and QLoRA, dataset formatting, held-out evaluation and versioned adapter artifacts.

Greens Technologies

- MCP foundations: Explain Model Context Protocol client/server roles, tools and resources; expose a narrow server interface with authentication and access checks.
- Deployment lab: Containerize a sample MCP server and inference API, configure secrets securely and add health checks, logs, rate limits and rollback steps.

Capstone Project

DAY 30 Project Phase

- Capstone Project: Integrating Data Eng, ML, and GenAI
- Capstone scope: Ingest a sample dataset with the data-engineering workflow, train an ML baseline and add a citation-backed GenAI interface.
- Implementation: Deliver versioned data transformations, a reproducible model pipeline, retrieval and application code, and an API or simple demonstration interface.
- Review and handoff: Present architecture, evaluation results, failure cases, access controls, cost estimates and deployment steps with a README and live demonstration.

Greens Technologies