

Analytics Engineer Roadmap 2026

SQL, dbt, metrics, BI, data quality and AI-assisted analytics roadmap

Goal: Become the bridge between raw data and trusted business decisions: model data as code, define metrics, test quality, document lineage and enable BI/AI consumption.

Prepared: 31 May 2026. **Use-case:** self-study, portfolio building, interview prep and job-readiness.

How to use this roadmap

- Follow the phases in order, but do not wait to build projects. Every topic should end with a public artifact: code, dashboard, demo, README, diagram or case study.
- Use YouTube for intuition and demos; use official documentation for correctness; use practice platforms for repetition; use projects for proof.
- Track progress with a weekly scorecard: hours learned, problems solved, project commits, notes written, demos recorded and applications/interviews completed.
- AI tools are allowed, but every project must show that you understand, test and can debug the output.

2026 market calibration

- Analytics engineering remains central because AI systems, dashboards and business users all need trusted, documented, well-modeled data.
- The 2026 analytics engineer must be excellent at SQL and dbt, but also comfortable with semantic layers, metrics governance, data contracts, BI, CI/CD and AI-assisted analytics workflows.
- Hiring signals increasingly emphasize data trust, metric consistency, documentation, ownership and the ability to collaborate with analysts, data engineers and business teams.

Role title mappings

- Analytics Engineer, dbt Developer, BI Engineer, Data Modeler, Analytics Developer, Data Product Analyst, Semantic Layer Engineer

2026 hiring-ready stack

- SQL mastery, dbt Core/Cloud, Snowflake/BigQuery/Databricks, Git and CI/CD, data quality tests, Power BI/Tableau/Looker/Superset, MetricFlow/semantic layers, Python for automation, orchestration basics, documentation and stakeholder communication

End-to-end roadmap overview

Phase	Focus	What to learn	Job-ready proof
0-4 weeks	Analytical SQL foundation	Joins, CTEs, windows, date logic, aggregations, query plans, dimensional modeling basics.	Solve 80+ SQL questions and publish annotated solutions.
1-2 months	Modern analytics stack	Warehouse basics, dbt models, sources, tests, docs, snapshots, seeds, environments, Git workflow.	Build a dbt project from raw orders/events to marts and docs.
2-4 months	Metrics and BI	Kimball dimensional modeling, semantic layer, KPI definitions, DAX/LookML basics, dashboard UX.	Deliver an executive dashboard with a metric glossary and testing.
4-6 months	Quality, governance and CI	Data tests, freshness, contracts, exposures, lineage, PR review, Slim CI, documentation standards.	Add CI, alerts, docs and lineage to the dbt project.
6-12 months	Advanced analytics engineering	Incremental models, performance tuning, Python automation, experimentation data, event modeling, AI consumption.	Ship a production-style analytics platform portfolio with 3 business domains.

Weekly operating system

- 10-12 hours learning/building: 60% project implementation, 20% docs, 10% YouTube/course intuition, 10% notes and revision.
- Minimum weekly output: 5 commits, 1 written note, 1 demo screenshot/video, 5-20 practice problems depending on role.
- Every Sunday: review blockers, update README, write what you learned, plan next week, and compare against target job descriptions.
- Every month: ship one small project or one major milestone of a bigger capstone.

Complete topic and subtopic checklist

1. SQL for analytics

- Select, joins, anti-joins, CTEs, subqueries, windows, date/time functions, arrays/JSON, pivoting, cohort logic, funnel logic, performance tuning, explain plans.
- Proof to create: Create a SQL notebook with 50 business questions and tested answers.

2. Data modeling

- Grain, dimensions, facts, slowly changing dimensions, star schema, wide tables, snapshots, bridge tables, event models, metric tables, naming standards.
- Proof to create: Model sales, marketing and product events into marts.

3. dbt as software engineering

- Sources, refs, staging/intermediate/marts, macros, tests, docs, exposures, snapshots, seeds, packages, tags, selectors, environments, CI/CD.
- Proof to create: Publish dbt docs and enforce PR checks.

4. Warehouses and lakehouses

- Snowflake/BigQuery/Databricks SQL, partitions/clustering, cost control, query optimization, roles, access, governance.
- Proof to create: Tune slow models and document cost/performance tradeoffs.

5. BI and semantic layer

- Power BI/Tableau/Looker/Superset, dashboard layout, semantic metrics, DAX/LookML basics, drill-downs, row-level security, stakeholder storytelling.
- Proof to create: Build dashboards that answer business decisions, not just display charts.

6. Data quality and observability

- Freshness, uniqueness, not-null, accepted values, custom tests, anomaly detection, lineage, incident process, SLAs, data contracts.
- Proof to create: Create tests, alerts and a data incident runbook.

7. Analytics product thinking

- KPI trees, north-star metrics, product funnels, retention, cohorts, experimentation, stakeholder interviews, metric disputes.
- Proof to create: Write a metric spec with owner, definition, grain, caveats and examples.

8. AI-assisted analytics

- LLM-assisted SQL review, metadata for AI, natural-language BI risks, RAG over metrics docs, human verification, hallucination-safe data answers.
- Proof to create: Create a small analytics assistant that only answers from governed metrics docs.

Portfolio projects and capstones

Do not treat these as toy tutorials. Each project should have a README, architecture diagram, setup steps, screenshots, demo link when possible, tests/checks, limitations and future improvements.

Project 1: E-commerce analytics warehouse

- Outcome: Raw orders/events to staging, facts, dimensions and revenue/customer marts.
- Suggested stack: dbt, DuckDB/Postgres/Snowflake, GitHub Actions
- Stretch goal: Add incremental models, docs, tests and BI dashboard.

Project 2: Metrics layer and KPI dictionary

- Outcome: Define revenue, retention, conversion and active user metrics with ownership.
- Suggested stack: dbt Semantic Layer/MetricFlow or LookML
- Stretch goal: Build a metric dispute playbook and validation queries.

Project 3: Data quality incident system

- Outcome: Freshness, uniqueness and anomaly checks with alerts and incident notes.
- Suggested stack: dbt tests, Great Expectations, GitHub Actions
- Stretch goal: Write a postmortem for a simulated broken source.

Project 4: Product funnel and cohort analysis

- Outcome: Model events to answer activation, retention and churn questions.
- Suggested stack: SQL, Python, BI
- Stretch goal: Compare metric definitions and business interpretations.

Project 5: AI-ready analytics docs

- Outcome: Documentation set that an LLM can safely retrieve from.
- Suggested stack: dbt docs, RAG, vector search
- Stretch goal: Add citations and confidence rules.

Interview preparation checklist

- Expect SQL rounds with windows, joins, date logic, revenue definitions and edge cases.
- Explain dimensional modeling decisions and tradeoffs around grain.
- Review a dbt model and add tests, docs and refactoring.
- Discuss a data quality incident and how you would detect, communicate and prevent it.
- Translate vague business questions into measurable KPIs.

Portfolio checklist before applying

- One dbt repo with docs site screenshots, lineage, tests and CI.
- One dashboard with business questions, assumptions and decision recommendations.
- One metric dictionary with definitions, SQL, owners and caveats.
- One data quality incident write-up.

Resume keywords to include when true

- SQL mastery, dbt Core/Cloud, Snowflake/BigQuery/Databricks, Git and CI/CD, data quality tests, Power BI/Tableau/Looker/Superset, MetricFlow/semantic layers, Python for automation, orchestration basics, documentation and stakeholder communication, Analytics Engineer, dbt Developer, BI Engineer

Best official docs and learning references

Prioritize these over random snippets. Read docs when implementation breaks, when preparing interviews, and when upgrading tools.

- **SQLBolt** - Interactive SQL basics. ([open](#))
- **Mode SQL Tutorial** - Analyst-oriented SQL tutorials. ([open](#))
- **dbt Docs** - Analytics engineering, models, tests, docs and CI. ([open](#))
- **Snowflake Docs** - Cloud data warehouse. ([open](#))
- **BigQuery Docs** - Google Cloud warehouse. ([open](#))
- **Databricks Docs** - Lakehouse, Spark, ML and governance. ([open](#))
- **Apache Spark Docs** - Distributed data processing. ([open](#))
- **Apache Airflow Docs** - Workflow orchestration. ([open](#))
- **Dagster Docs** - Data assets and orchestration. ([open](#))
- **Prefect Docs** - Pythonic orchestration. ([open](#))
- **Apache Kafka Docs** - Event streaming. ([open](#))
- **Apache Flink Docs** - Stream processing. ([open](#))
- **Apache Iceberg Docs** - Open table format. ([open](#))
- **Delta Lake Docs** - Lakehouse table format. ([open](#))
- **DuckDB Docs** - Local analytics database. ([open](#))
- **Great Expectations Docs** - Data quality tests. ([open](#))
- **OpenLineage Docs** - Lineage metadata. ([open](#))
- **Microsoft Power BI Docs** - Power BI modeling, DAX and dashboards. ([open](#))
- **Tableau Help** - Tableau visual analytics. ([open](#))
- **Looker Docs** - Semantic modeling and BI. ([open](#))
- **Apache Superset Docs** - Open-source BI. ([open](#))
- **Git documentation** - Version control fundamentals and workflows. ([open](#))
- **GitHub Docs** - GitHub repos, Issues, PRs, Actions and security features. ([open](#))
- **Python Docs** - Language reference and standard library. ([open](#))
- **Real Python** - Practical Python tutorials. ([open](#))
- **NumPy Docs** - Arrays and numerical computing. ([open](#))
- **pandas Docs** - Dataframes, cleaning and analysis. ([open](#))
- **dbt Developer Hub** - dbt guides, best practices and reports. ([open](#))
- **dbt Semantic Layer** - Semantic metrics for analytics engineering. ([open](#))
- **SQLFluff Docs** - SQL linting and style. ([open](#))
- **Elementary Docs** - dbt data observability. ([open](#))
- **Datafold Docs** - Data diff and quality workflows. ([open](#))

Best YouTube sources and how to use them

Use these for intuition, project walkthroughs and high-level context. Always verify implementation details against official documentation because tools change quickly.

- **freeCodeCamp.org** - Long-form free courses across web, data, cloud and AI. ([open](#))
- **ByteByteGo** - System design, architecture and backend concepts. ([open](#))
- **DataTalksClub** - Data engineering and MLOps zoomcamps. ([open](#))
- **Seattle Data Guy** - Data engineering career and architecture. ([open](#))
- **dbt Labs** - Analytics engineering and dbt. ([open](#))
- **Databricks** - Spark, lakehouse, ML and data platform. ([open](#))
- **Snowflake Inc.** - Warehouse, data cloud and AI platform. ([open](#))
- **Alex The Analyst** - Data analyst portfolio, SQL, Excel, Power BI. ([open](#))
- **Luke Barousse** - Data analyst career and Python/SQL projects. ([open](#))
- **Guy in a Cube** - Power BI and analytics reporting. ([open](#))
- **codebasics** - Data analytics, SQL and Python projects. ([open](#))
- **DeepLearning.AI** - ML, GenAI and agentic AI courses. ([open](#))
- **StatQuest** - Statistics and ML explained clearly. ([open](#))

Recommended YouTube search strategy

- **YouTube search: Analytics Engineer roadmap 2026 project based** - Use filters for upload date and length; prefer recent, project-based content. ([open](#))
- **YouTube search: Analytics Engineer interview preparation 2026** - Use filters for upload date and length; prefer recent, project-based content. ([open](#))
- **YouTube search: Analytics Engineer portfolio project end to end** - Use filters for upload date and length; prefer recent, project-based content. ([open](#))
- **YouTube search: Analytics Engineer system design real world project** - Use filters for upload date and length; prefer recent, project-based content. ([open](#))

Best coding and practice platforms

Practice platforms build repetition. They do not replace projects. For each platform, maintain a solved-problems log with mistakes and revised patterns.

- **DataLemur SQL Questions** - Data interview SQL. ([open](#))
- **StrataScratch** - SQL, Python and analytics interview questions. ([open](#))
- **SQLBolt** - Beginner SQL drills. ([open](#))
- **Mode SQL Tutorial** - Analytical SQL walkthroughs. ([open](#))
- **HackerRank SQL** - SQL exercises. ([open](#))
- **Kaggle Learn** - Micro-courses for Python, pandas, ML and SQL. ([open](#))
- **Data Engineering Zoomcamp** - Hands-on pipelines, warehouses and orchestration. ([open](#))
- **Kaggle Datasets** - Data sources for portfolio pipelines. ([open](#))
- **Google Cloud Skills Boost** - Cloud labs. ([open](#))
- **LeetCode** - DSA patterns for interviews. ([open](#))
- **NeetCode Roadmap** - Structured DSA practice. ([open](#))
- **HackerRank** - Algorithms, SQL and language tracks. ([open](#))

30-60-90 day execution plan

Period	Primary objective	Deliverables
Days 1-30	Foundation and first small project	Finish basics, solve starter practice, publish one small project and notes.
Days 31-60	Core role skills and second project	Complete core docs, build a stronger project, add tests/checks and write a case study.
Days 61-90	Production proof and interview prep	Deploy capstone, create architecture diagram, record demo, start mock interviews and applications.

Common mistakes to avoid

- Only watching videos without building public proof.
- Building tutorial clones without explaining decisions, tradeoffs and failure cases.
- Ignoring documentation, testing, security, accessibility, observability or data quality.
- Using AI-generated code without understanding it or adding tests.
- Applying to jobs before your portfolio proves the core responsibilities of the role.

Market research sources consulted

- **World Economic Forum Future of Jobs 2025** - Macro skill demand for 2025-2030; big data, AI/ML, software, cybersecurity and tech literacy signals. ([open](#))
- **Stack Overflow Developer Survey 2025** - Developer technology, AI-tool use, documentation, languages and web framework signals. ([open](#))
- **GitHub Octoverse 2025** - Open-source and language momentum; TypeScript, Python, agents and typed-language shift. ([open](#))
- **DORA State of AI-assisted Software Development 2025** - AI in software delivery; verification, platform maturity and organizational system effects. ([open](#))
- **CNCF Annual Cloud Native Survey** - Kubernetes, cloud-native maturity and AI infrastructure adoption. ([open](#))
- **dbt State of Analytics Engineering 2025** - Analytics engineering trends: AI use, data trust, quality and investment. ([open](#))
- **LangChain State of Agent Engineering** - Agent adoption, production challenges, evaluation and observability trends. ([open](#))
- **McKinsey State of AI 2025** - Enterprise AI adoption, scaling practices, governance and value capture. ([open](#))
- **dbt State of Analytics Engineering 2025** - Role and market report. ([open](#))
- **dbt Docs** - Primary skill documentation. ([open](#))