

dbt Mastery: Analytics Engineering Excellence

Course Description

This hands-on course equips data professionals to master dbt as full-fledged developers, covering foundational concepts through advanced production techniques. Participants learn to build, test, document, and deploy modular SQL transformations in the warehouse, leveraging Jinja, incremental models, snapshots, state management, and emerging features like dbt Mesh and Python models. Ideal for analytics engineers transitioning legacy code to scalable pipelines.

Course Duration : 32 hours

Prerequisites

- Proficiency in SQL (ability to write and understand complex queries across data warehouses like Snowflake, BigQuery, or Databricks)
- Familiarity with a data warehouse and basic Git/version control
- Optional: Introductory exposure to dbt fundamentals

Table of Contents

Module 1: dbt Fundamentals

- Connecting dbt to a data warehouse
- Configuring Git integration for dbt projects
- Understanding models, sources, tests, documentation, and deployment concepts
- Building your first dbt model
- Running and interpreting basic dbt tests
- Navigating the dbt project structure and workflows

Module 2: Refactoring SQL for Modularity

- Analyzing legacy SQL code and stored procedures for migration

- Breaking monolithic SQL into modular dbt models
- Porting stored procedures and SQL scripts into dbt projects
- Applying best practices for reusable, maintainable transformations
- Organizing models into layers (staging, intermediate, marts) for clarity

Module 3: Jinja, Macros, and Packages

- Introduction to Jinja templating within dbt
- Creating and using macros to avoid repetitive SQL
- Parameterizing logic and configurations with Jinja
- Installing and managing dbt packages
- Reusing models and macros from community and internal packages

Module 4: Materialization Fundamentals

- Understanding built-in materializations: table, view, ephemeral
- Choosing the appropriate materialization for different use cases
- Configuring model materializations in dbt
- Performance and cost considerations across materializations
- Refactoring existing models to use optimal materializations

Module 5: Incremental Models

- Concept of incremental processing and when to use it
- Configuring incremental models in dbt
- Defining unique keys and incremental strategies
- Managing schema changes and backfills with incremental models
- Reducing build time and compute cost through incremental runs

Module 6: Snapshots

- Slowly changing dimensions and historical tracking concepts
- Defining and configuring dbt snapshots
- Capturing and storing historical records over time
- Querying snapshot data for change history and audit
- Managing snapshot performance and storage

Module 7: Analyses and Seeds

- Creating analyses for ad hoc queries within dbt
- Structuring and organizing analysis files in projects
- Using seeds to load version-controlled CSV data
- Updating and maintaining seed data as part of the codebase
- Integrating seeds into downstream models and tests

Module 8: Exposures

- Purpose and value of Exposures in dbt projects
- Configuring Exposures for dashboards, reports, and data products
- Visualizing downstream dependencies from models to consumers
- Keeping Exposures definitions and dependencies up to date
- Using Exposures for stakeholder communication and impact analysis

Module 9: Understanding State

- Concept of state management in dbt
- Capturing and using project state between runs
- Running state-aware commands for efficient builds
- Leveraging state to target modified or new models

- Operational benefits of state-based workflows

Module 10: dbt Retry

- Use cases for retrying dbt pipeline executions
- Understanding the dbt retry command
- Rebuilding failed parts of a pipeline efficiently
- Strategies for handling intermittent warehouse or network issues
- Integrating retry into operational runbooks

Module 11: dbt Mesh

- Core concepts of dbt Mesh and data products
- Scaling development across multiple teams with Mesh
- Managing ownership and contracts between domains
- Improving reliability and development speed using Mesh patterns
- Governance considerations in a Mesh-enabled architecture

Module 12: Advanced Testing

- Testing philosophy and types of data tests
- Creating custom generic tests in dbt
- Leveraging tests shipped in dbt packages
- Applying advanced test configurations and selectors
- Building a robust, automated testing layer for critical models

Module 13: Advanced Deployment

- Designing deployment workflows for dbt projects

- Implementing continuous integration for dbt changes
- Orchestrating jobs that might conflict or overlap
- Customizing behavior per environment (dev, test, prod)
- Monitoring, alerting, and operational best practices for deployments

Module 14: dbt Clone

- Concept and benefits of dbt clone
- Creating logical copies of database objects for development
- Using clones for testing without full data duplication
- Managing cloned environments and lifecycle
- Integrating cloning into development workflows and CI

Module 15: Grants

- Overview of permissions and access control in dbt
- Configuring grants for models, seeds, and snapshots
- Aligning dbt grants with warehouse-level security policies
- Automating grants as part of deployments
- Ensuring secure, least-privilege access to data products

Module 16: Python Models

- When to use Python models instead of pure SQL
- Creating and configuring Python models in dbt
- Using Python data science and statistics libraries in models
- Combining SQL and Python models within a single project
- Example advanced analytics and ML-oriented transformations with Python models