Data Engineering 101: Building Reliable Data Pipelines

Author: Kamakshaiah Musunuru | **Published:** 11 September 2026 | **Category:** Blog | **Read time:** 2 min read

Abstract

Reliable data pipelines turn raw sources into trustworthy analytics and machine-learning inputs. This article covers ingestion, transformation, orchestration, quality and the modern data stack.

Keywords: data engineering, data pipeline, ETL, ELT, data warehouse, lakehouse, orchestration, data quality

Tags: data engineering, pipelines, ETL, data warehouse, orchestration

Data engineering is the discipline of building dependable systems that collect, transform and serve data. Well-engineered pipelines are the foundation of analytics, reporting and machine learning.

The Data Pipeline Lifecycle

1. Ingestion Move data from source systems into a storage layer. Choose between: - Batch ingestion for scheduled, predictable transfers - Streaming ingestion for real-time event processing - Change data capture (CDC) for synchronising databases Handle schema drift, partial failures and backfills reliably.

2. Storage Select storage that fits the workload: - Data lakehouses for flexible, large-scale analytics - Data warehouses for structured, query-optimised analytics - Message streaming platforms for real-time pipelines Keep raw and processed layers separated and versioned.

3. Transformation Shape raw data into clean, usable form: - Clean, deduplicate and validate records - Join, aggregate and enrich to build fact and dimension tables - Use SQL-first transformation frameworks (dbt-style) for maintainability - Model the data explicitly to document meaning and lineage

4. Orchestration Automate and sequence pipeline steps: - Use schedulers and workflow engines (Airflow, Dagster, Prefect) - Define dependencies, retries, timeouts and alerting - Make pipelines idempotent so they can be re-run safely - Support backfills and incremental loading

5. Data Quality Guard the trustworthiness of your data: - Define and test quality rules on freshness, volume, uniqueness and completeness - Alert when checks fail and block downstream consumption - Track data lineage so issues can be traced to their source - Establish ownership and a data catalog for discoverability

6. Serving and Governance Expose data to consumers: - Serve analytics through BI tools and APIs - Provide feature stores for machine learning - Enforce access control, encryption and retention policies - Comply with regulatory and privacy requirements

Modern Stack Patterns

The modern data stack combines a lakehouse for storage, SQL transformation tools, workflow orchestration, and a catalog for governance. Streaming platforms extend this to real-time use cases, while feature stores bridge analytics and machine learning.

Key Principles

- Make pipelines idempotent, observable and resumable - Test data as rigorously as code - Document lineage, ownership and semantics - Automate alerting on failures and anomalies - Measure pipeline reliability and data freshness as core metrics

Reliable data engineering is what separates data-driven organisations from those that merely collect data. Invest in foundations before scaling.

Codingfigs builds production-grade data platforms — ingestion, transformation, orchestration and quality — for analytics and ML teams.

References

Designing Data-Intensive Applications by Martin Kleppmann; The Data Engineering Cookbook; dbt Documentation; Apache Airflow Documentation.