UnCodemy

Data Engineer Roadmap PDF

A data engineer builds the pipelines and storage that deliver reliable data to analysts and models. The roadmap is: strong SQL, Python, data modelling, ETL and ELT pipelines, a data warehouse, orchestration, cloud basics, Linux and Git, then streaming and data quality. This page lists the stages with a project and a check for each, and makes no job promises.

Key takeaways

- SQL and Python are the two core languages for data engineering.
- Learn data modelling: tables, keys, normalisation and star schemas.
- Pipelines move and transform data; orchestration schedules and monitors them.
- A warehouse or lakehouse on the cloud is the usual destination for analytics data.
- Data quality checks and monitoring are part of the job, not extras.

What is data engineer roadmap?

A data engineer designs, builds and maintains the systems that collect, move, store and prepare data so that analysts, data scientists and applications can use it reliably.

What does a data engineer do?

A data engineer makes data available and trustworthy. Tasks include pulling data from applications and files, cleaning and transforming it, loading it into a warehouse, scheduling jobs, monitoring quality, controlling cost and helping analysts and scientists use the data. Roles vary: some focus on batch pipelines, others on streaming, platforms or analytics engineering.

What are the stages?

Stage 1: computing basics. Command line, Linux basics, Git, networking ideas, how files and processes work. *Check:* you can move around a server, run scripts and commit your work.

Stage 2: SQL. Joins, aggregation, subqueries, CTEs, window functions, indexes and reading query plans. *Project:* design and query a small sales database. *Check:* you can solve window function problems without help.

Stage 3: Python for data. Data structures, files, JSON and CSV, functions, error handling, packages, testing and a data library such as pandas. *Project:* a script that reads messy CSV files, cleans them and loads them into a database.

Stage 4: data modelling. Entities and keys, normalisation, fact and dimension tables, star schemas, slowly changing dimensions. *Project:* model a retail business as a star schema and

explain each table.

Stage 5: ETL and ELT pipelines. Extract from sources, transform, load, incremental loads, idempotency (safe to rerun) and error handling. *Project:* a daily pipeline that loads new rows from an API or file into a warehouse table.

Stage 6: warehouses and storage. Row and column storage, data lakes, lakehouses, file formats such as CSV, JSON and Parquet, and partitioning. *Project:* load the same data to a warehouse and compare query speed with partitioning.

Stage 7: orchestration. Scheduling, dependencies, retries, alerts and backfills with a workflow tool such as Airflow. *Project:* orchestrate three dependent tasks with retries and an alert.

Stage 8: cloud basics. Storage, compute, identity and permissions, networking basics and cost control on one cloud provider. *Project:* run your pipeline in the cloud with a budget alert.

Stage 9: big data and streaming. Distributed processing ideas with a tool such as Spark, and message streams with a tool such as Kafka. *Project:* process a larger file with Spark and write a small streaming demo.

Stage 10: quality, security and operations. Data tests, lineage, monitoring, access control, privacy and documentation. *Project:* add checks for nulls, duplicates and freshness to your pipeline.

How do ETL and ELT differ?

ETL transforms data before loading it into the target. ELT loads raw data first and transforms it inside the warehouse using SQL, which is common with modern cloud warehouses. Choose by data size, tools and team skills.

What does a good pipeline look like?

It is idempotent, so rerunning does not duplicate data. It logs what it processed, handles bad records safely, validates input and output, has alerts and is documented. Test with small data first and keep secrets out of code.

Which tools should I learn?

Learn the ideas first, then one tool in each class: a relational database, Python, a workflow scheduler, a cloud warehouse, a distributed processing engine and a version control and CI setup. Job postings in your region show which tools are popular.

What portfolio should I build?

One end-to-end project: ingest data from an API or files, clean and model it, load it to a warehouse, schedule it, add quality checks, build a small dashboard on top and document the design with a diagram. Use public or made-up data.

How long does it take?

It depends on your starting point and study time. Data engineering has many topics, so expect several months of steady work to reach entry level, and more to become strong. No roadmap can promise a job.

What mistakes should I avoid?

Skipping SQL depth, building pipelines that cannot be rerun, ignoring data quality, learning many tools without projects and forgetting cost and security.

What does a first month look like?

Week one: Linux basics, Git and installing a database. Week two: SQL queries on a sample schema. Week three: Python scripts that read and write files. Week four: a first mini pipeline that loads a CSV into a database and checks row counts. Keep each week small and finish a working output, however simple, before moving on.

How do I practise data quality?

Add checks to your project: no duplicate keys, no unexpected nulls, values within a valid range, new data arriving on time and totals matching the source. Log failures and stop or quarantine bad data rather than loading it silently. Showing that you test your pipelines is a strong signal of engineering maturity.

Frequently asked questions

What does a data engineer do?

A data engineer builds and maintains the pipelines and storage that collect, transform and deliver reliable data for analysis and applications.

What should I learn first for data engineering?

Start with SQL and Python, then data modelling, ETL pipelines, a warehouse, orchestration and cloud basics.

What is the difference between ETL and ELT?

ETL transforms data before loading it, while ELT loads raw data first and transforms it inside the warehouse.

What does idempotent mean for a pipeline?

Running the pipeline again produces the same result without duplicating data, which makes retries and backfills safe.

Does this roadmap guarantee a job?

No. It is a learning path. Hiring depends on the market, your projects and interviews.

Sources and further reading

- PostgreSQL documentation
- Python documentation
- Apache Airflow documentation
- Apache Spark documentation

About this PDF

These PDFs are study material for learning and revision. They are not a certificate, a degree or a promise of a job, and technology changes, so check official documentation for the latest details. Spot an error? Write to info@uncodemy.com and the trainer team will review it.

More free resources and courses: uncodemy.com | info@uncodemy.com | +91 98183 66550