*Template from www.lineupx.com/job-descriptions/data-engineer. Replace every highlighted [field] and keep the version you need: the full-time job for freshers or the internship.*

---

# Data engineer (fresher)

Company: [Company name]
Location: [Location] | Work mode: [Work mode]
Employment type: Full-time | CTC: [CTC for the full-time role]
Joining: [Joining date] | Apply by: [Apply by]

## About [Company name]

[About your company]

## About the role

[Company name] is hiring a fresher data engineer for the [Team] team to build and maintain small data pipelines. You will write SQL and Python jobs that bring data from files, applications and databases into our warehouse, check that each load is complete and fix it when it fails. Analysts and managers rely on these tables every morning, so a senior engineer will review each change and teach you how we test before anything reaches them.

## What you will do

- Write SQL and Python jobs that extract data from files, databases and APIs and load it into the warehouse or database, with a senior reviewing each pipeline
- Read and clean incoming CSV, JSON and Parquet files by fixing data types, dates and encodings, and set aside rows that fail checks together with the reason
- Make loads safe to repeat using the team's agreed keys and update rules, check for duplicate source keys before an upsert or merge, and test that repeating the same input after a partial failure leaves the expected rows and values
- Schedule jobs in Airflow or cron, add retries and failure alerts, and keep jobs that depend on each other in the right order
- Add data-quality checks after each load: row counts against the source, null and duplicate checks on key columns, and a check that yesterday's data has arrived
- Investigate failed or late loads from the logs, find the cause, such as a renamed column, a missing file or a timeout, fix it or escalate, and write a short incident note [If applicable: you will join the out-of-hours rota after shadowing a senior]
- Write SQL transformations that turn raw tables into clean ones for analysts, and compare their totals with the source before release
- Run a backfill for past dates when a senior asks, and confirm that the loaded row counts match the source and that no day is loaded twice
- Document each table and pipeline: what the columns mean, where the data comes from, when it refreshes and who owns it
- Handle requests from analysts, such as a missing column or a total that does not match, and track each one in the team's ticket tool

## What we are looking for

- SQL beyond the basics: joins, GROUP BY, subqueries and common table expressions, on PostgreSQL, MySQL or a similar database
- Python for scripts: reading and writing files, dictionaries, error handling, and pandas for cleaning CSV and JSON
- Relational database basics: primary and foreign keys, indexes and why tables are normalised

- A clear idea of extract, transform and load, and of the difference between a full load and an incremental load
- Git and the Linux command line: running a script, reading a log file and finding the line that failed
- Care with numbers: you compare row counts and totals before and after a load, and report a mismatch instead of ignoring it
- One small pipeline you built, such as a script that downloads a public file, cleans it and loads it into a database. Share the repository with a README, or walk through the steps if the work is private

### Good to have

- Apache Airflow: a small workflow (DAG) you wrote or ran, even on your laptop
- Hands-on use of a cloud warehouse such as BigQuery, Snowflake or Redshift
- Parquet and the pyarrow library, and how a column-oriented format differs from CSV for large files
- dbt basics: a model, a test and a documentation page
- Docker basics for running a database or a job on your laptop

### Who can apply

Final-year students and graduates of B.E./B.Tech in computer science, IT or electronics fit well, as do BCA, MCA, B.Sc and M.Sc candidates in computer science, statistics or mathematics. Candidates from other streams are equally welcome. Judge every applicant on the requirements above, not the degree.

### How to apply

Apply through this job post by [Apply by]. Shortlisted candidates will hear from us about the next steps.

# Data engineer intern

Company: [Company name]
Location: [Location] | Work mode: [Work mode]
Employment type: Internship | Duration: [Internship duration] | Stipend: [Stipend for the internship]
Joining: [Joining date] | Apply by: [Apply by]

## About [Company name]

[About your company]

## About the role

[Company name] offers a data engineering internship for students who want to see how data reaches the reports teams use each day. You will work with a mentor on one small pipeline: loading files into a database, checking the data and reading logs when a run fails. You will finish by presenting the pipeline and a one-page guide to running and repairing it.

## What you will do

- Load a folder of sample or masked CSV files into a PostgreSQL table with a Python script, and record how many rows loaded and how many were rejected
- Make that script safe to run twice on the same input, then compare keys, row counts and stored values after both runs to confirm that the result is unchanged
- Write SQL checks that find duplicate keys, blank required columns and missing dates in a table
- Read the log of a failed run in the test environment and write down what failed and how it was fixed
- Convert one large CSV file to Parquet and note the change in file size and read time
- Document one table with its columns, source and refresh time, and present the pipeline to the team

## What we are looking for

- Pursuing a degree in computer science, IT, statistics or a related subject, or recently graduated
- SQL from coursework: joins, grouping and filters
- Python scripts that read a file and write results, with basic error handling
- A small project that moves data from a file or an API into a database, shared on GitHub or described if it is private
- Willing to read logs and error messages patiently and to ask when a failure stays unclear

## Who can apply

Final-year students and graduates of B.E./B.Tech in computer science, IT or electronics fit well, as do BCA, MCA, B.Sc and M.Sc candidates in computer science, statistics or mathematics. Candidates from other streams are equally welcome. Judge every applicant on the requirements above, not the degree.

## How to apply

Apply through this job post by [Apply by]. Shortlisted candidates will hear from us about the next steps.

## Screening questions to add to your job post

1. **A load failed halfway through. How do you re-run it safely without creating duplicates?** Look for a repeatable load strategy, such as staging followed by an upsert on agreed keys, or replacing the affected date's rows in one transaction. A good answer checks duplicate keys, row counts and stored values after the rerun. Ask how the candidate would confirm that a partial failure has not changed the intended result.
2. **How do you find duplicate rows in a table?** Look for grouping by the key columns and keeping the groups with a count above one, or numbering the rows within each key. A good extra is asking what counts as a duplicate here: the same key or the whole row.
3. **You join orders to customers and the row count drops. What is the likely cause, and what would a left join from orders to customers change?** Look for an inner join dropping orders that have no matching customer, and a left join from orders retaining unmatched orders with NULL values in the customer columns. If the candidate mixes up the two joins, ask them to work through a three-row example.
4. **What is a primary key, and why does a pipeline care about it?** Look for a column or set of columns that identifies each row uniquely, and its use in joins, duplicate checks and upserts. If the answer stops at the definition, ask where a load would use the key.
5. **How would you check that yesterday's load is complete?** Look for a row count against the source, the latest date or timestamp loaded, null checks on key columns and an alert when a check fails. If the candidate would look at a few rows by eye, ask what else they could check.
6. **A daily file arrives with a new column and dates in a different format. What should the pipeline do?** Look for detecting the change, avoiding loading wrong values and telling the file's owner. Accept stopping the load, setting the file aside, or loading only the known columns while rejecting the unexpected dates. A good extra is standardising the date format in one place and checking a few known rows afterwards.