| **Stakeholder** | **What They Do** | **What They Need** | **How You Help** |
| --- | --- | --- | --- |
| **Business Execs** | Make strategy decisions | High-level trends, KPIs | Provide summarized dashboards |
| **Analysts** | Explore & interpret data | Clean, structured data | Give them access via SQL or BI tools |
| **Data Scientists** | Build ML models | Well-structured datasets | Feed them enriched, joined data |
| **Software Developers** | Build apps | API-accessible data | Provide them with fast, queryable endpoints |
| **Customers** | Use products | Privacy, performance | Secure their data, keep systems fast |
| **YOU** – Data Engineer | The pipeline master | Raw data → usable formats | Ensure entire system doesn’t break 🔥 |

**🔗Your Position Among Stakeholders (Like a Boss):**

| **Role** | **Nickname** | **Main Focus** | **Tools/Skills** |
| --- | --- | --- | --- |
| **Data Analyst** | Storyteller | Explains data trends | Excel, SQL, Tableau |
| **Data Engineer** | Plumber | Builds data pipelines | Python, Spark, ETL, Cloud |
| **Data Scientist** | Detective | Predicts outcomes | Python, ML, Stats |
| **Data Architect** | Architect | Designs data systems | SQL, Cloud, Modeling |

**🍔🛵 Scenario: You Work at “SwiggyZom Eats” (Fake but iconic)**

**🔍 The Business Problem:**

Management wants to know:

1. **Which cities are facing the most delivery delays?**
2. **Which customers might uninstall the app soon? (churn)**

But... they don’t have that answer ready. They have only this:

**🔢 Raw Data Lying Around in 4 Different Systems:**

| **Data Source** | **What’s Inside** |
| --- | --- |
| 🧾 Orders Table | Order ID, Customer ID, Delivery Time, Order Time |
| 🤝 Customers Table | Customer ID, Name, City, Signup Date |
| 🧍 Delivery Agents Table | Agent ID, Assigned Orders, Region |
| ⭐ Ratings Table | Order ID, Rating (1–5), Feedback text |

**💥 YOU, the Data Engineer, JUMP IN:**

Let’s break your role down step-by-step.

**STEP 1: Data Ingestion**

You collect all this data **from different databases, APIs, and logs** into a central platform (like BigQuery, Snowflake, or a data lake).

🛠️ You write code (maybe with Python + Airflow or Kafka) to pull:

* Orders from MySQL
* Customer info from a CRM
* Feedback logs from an API

**STEP 2: Data Cleaning**

Now comes the dirty laundry 🧺

You:

* Remove entries with null delivery times
* Fix wrong city names (e.g., “Banglore” → “Bangalore”)
* Drop duplicate rows
* Standardize timestamps into same format

Because *garbage in = garbage out*.  
You make raw data **clean and consistent**.

**STEP 3: Data Transformation**

Now you **join and reshape** the data into useful formats.

You:

* Join Orders + Customers → get customer-wise delivery time
* Join Orders + Ratings → link bad ratings to late deliveries
* Add a column: Delivery Delay = Delivery Time - Expected Time

Now you can answer:

“Which customer got late delivery?”  
“Which city has most delays?”

✨ This is **the chef moment**—you’ve cooked raw ingredients into a usable dish 🍝

**STEP 4: Serving (Dashboard + ML Model)**

Now the magic happens.

**✅ You send transformed data to:**

* 📊 **A Dashboard**:
  + Created by an analyst
  + Shows: *"Bangalore has 25% delivery delays at peak hours"*
* 🧠 **A Machine Learning Model**:
  + Built by data scientists
  + Predicts: *“Customers who got 2+ late deliveries + 1-star ratings are likely to uninstall the app”*