# 🧠 Data Extraction with Regex & `grep`

## 📘 Scenario Overview

In a data-driven environment, logs often contain a mixture of valuable and irrelevant information. As part of this challenge, I acted as a data analyst tasked with extracting **structured insights**—numerical records and email addresses—from an unstructured log file using **regular expressions**.

This lab demonstrates **real-world log parsing and data extraction**, a critical skill for system administrators, DevOps engineers, SOC analysts, and data professionals.

---

## 🎯 Objectives

- Extract lines beginning with numerical data
- Identify and extract valid email addresses
- Use regular expressions safely and accurately
- Preserve original data integrity
- Redirect extracted results into separate output files

---

## 📂 Source Data

**Input file:**
`/home/labex/project/data`

-

The file contains:

- Numerical identifiers
- Email addresses with varying domains
- Mixed, unstructured text

---

## 🔢 Step 1: Extract Lines Beginning with a Number

### Purpose

Capture log entries that begin with numeric values (e.g., customer IDs or transaction numbers).

### Command Used

```
grep '^[0-9]' /home/labex/project/data > /home/labex/project/num
```

### Explanation

- `^` → anchors the match to the **start of the line**
- `[0-9]` → matches any digit
- Output redirected to `num`

### Result

- `/home/labex/project/num` contains **only lines starting with numbers**
- Original data file remains unchanged

---

## 📧 Step 2: Extract Valid Email Addresses

### Purpose

Identify and isolate correctly formatted email addresses for CRM and analytics use.

### Command Used

```
grep -E '[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}'
/home/labex/project/data > /home/labex/project/mail
```

### Explanation

- `-E` → enables extended regular expressions
- Local part: `a-zA-Z0-9._%+-`
- `@` → literal at symbol
- Domain supports:
    - `.com`
    - `.org`
    - `.co.uk`

- Output redirected to `mail`

## Result

- `/home/labex/project/mail` contains **only valid email addresses**
- Handles multiple domain formats accurately

```
labex:project/ $ ls
data
labex:project/ $ cat data
1234567890
john.doe@example.com
miscellaneous data entry
9876543210
jane_smith123@company.co.uk
random text line
5555555555
support@our-company.com
3141592653
tech.support@company.com
irrelevant information here
1010101010
alice.wonderland@gmail.com
2718281828
bob_builder@construction.net
more random text
1123581321
charlie.brown@peanuts.com
9999999999
david_copperfield@magic.io
unrelated data line
```


---

## 🧠 Skills Demonstrated

- Regular expression fundamentals
- Anchors (`^`) and character classes
- Pattern matching with `grep -E`
- Safe data extraction without file modification
- Output redirection for structured results
- Log and dataset parsing techniques