DATA ENGINEER ROADMAP: SQL → Spark → Scala

🟦 PHASE 1 — MASTER SQL (2–4 weeks)

SQL is the **foundation** of all data engineering.

✅ 1. SQL Basics

- SELECT, WHERE, ORDER BY
- LIMIT, DISTINCT
- Basic filters
- Arithmetic & logical operators

Practice:
Write queries on real tables: employees, orders, customers.

✅ 2. Joins

- INNER
- LEFT / RIGHT
- FULL
- CROSS
- Self join

Goal: Explain and visualize join outputs in interviews.

✅ 3. Grouping & Aggregations

- GROUP BY
- HAVING
- COUNT, SUM, AVG, MIN, MAX

✅ 4. Advanced SQL

- Window Functions
- CTEs
- Subqueries
- Date/Time functions
- CASE WHEN
- COALESCE, NULL handling

✔ Deliverables:

- Write 40+ complex SQL queries
- Solve LeetCode Data SQL problems
- Implement 5 SQL-based ETL pipelines

🟧 PHASE 2 — SPARK FOUNDATIONS (3–5 weeks)

🔹 1. Spark Basics

Learn:

- Spark architecture
- Driver & Executors
- Transformations vs Actions
- Lazy evaluation
- DAG

🔹 2. Spark with DataFrames (Main skill required for real jobs)

Must learn:

- Reading/Writing files
- Select, filter, cast
- groupBy, agg
- joins
- union, limit
- withColumn, drop, rename

These are used in *every ETL job*.

🔹 3. Spark SQL

- Creating temporary views
- Writing SQL on Spark
- UDFs
- Built-in functions

🔹 4. Intermediate Spark

- Repartition vs Coalesce
- Cache vs Persist
- Shuffle, skew
- Broadcast join
- Window functions in Spark

🔹 5. Advanced Spark

- Spark on AWS EMR
- Partitioning, Bucketing
- Optimizing cluster usage
- Spark UI & performance tuning
- Catalyst optimizer
- Stages & tasks breakdown

🔹 6. Project to build

ETL Pipeline:

- Read JSON from S3
- Clean data
- Apply business logic
- Write to Parquet
- Register Delta table

🟩 PHASE 3 — SCALA FOR DATA ENGINEERS (3–6 weeks)

Learn **only the Scala required for Spark** (not advanced FP unless needed).

🔹 1. Essential Scala Syntax

- Variables, types
- Functions
- Loops, conditionals
- Collections (List, Seq, Map)
- Tuples
- Options (Some / None)

🔹 2. Scala OOP Basics

- Classes
- Objects
- Case classes
- Companion objects

🔹 3. Functional Programming (Only DE essentials)

- map
- flatMap
- filter
- reduce
- fold
- Anonymous functions

Know enough to write clean Spark code.

🔹 4. Error Handling

- Try / Success / Failure
- Either
- Pattern matching

🔹 5. Scala + Spark Integration

Learn writing:

- UDFs
- case class-based schemas
- Datasets

Example:

```
case class Person(id: Int, name: String)
val ds = df.as[Person]
```

🟦 PHASE 4 — END-TO-END PROJECTS (4–6 weeks)

Do 3–5 fully working projects.

---

1. Batch ETL Project

- Read CSV
- Validate schema
- Clean data
- Window functions
- Agg logic
- Write to Delta

---

2. S3 → EMR ETL Pipeline

- Triggered by S3 upload
- Spark job on EMR
- Output to Redshift

---

3. Data Quality Framework

Write Spark-based DQ checks:

- null checks
- uniqueness checks
- referential checks
- regex validation

---

4. Real-time pipeline (Optional but good)

Kafka → Spark Structured Streaming → Delta → Dashboard

🟥 PHASE 5 — TOOLING & ECOSYSTEM (Parallel Learning)

Learn tools alongside Spark/Scala.

Must-learn tools:

- Git
- IntelliJ
- sbt
- Maven (optional)
- Linux + Shell
- AWS (S3, EMR, Lambda, Glue, Athena, Redshift)
- Databricks

⭐ Recommended Weekly Plan (Fast Track)

Week 1–2 → SQL

Week 3–4 → Scala Basics

Week 5–8 → Spark Core + Spark SQL + ETL

Week 9–12 → Projects + AWS + performance tuning