AI Friday Chennai Regional Finals – Problem Statements

4. **ENTERPRISE DATA LINEAGE**: TRACING & VISUALIZING OF DATA FOR COMPLIANCE & OPERATIONAL EXCELLENCE

Domain: Data Engineering

PROBLEM STATEMENT:

In large organizations, data flows across multiple systems and technologies (e.g., COBOL, SAS, PySpark, Scala, Java, Python, Oracle, Teradata, SQL Server, MongoDB). The absence of automated, end-to-end data lineage leads to major challenges:

- Root cause analysis for data issues is slow and manual.
- Regulatory submissions (e.g., BCBS 239, SOX, CECL) lack traceable, auditable lineage documentation.
- Change impact analysis is error-prone, risking compliance and operational failures.
- Documentation is fragmented, inconsistent, and often out-of-date.

IMPACT:

- Increased operational risk and delayed incident resolution.
- Audit exceptions and regulatory non-compliance due to incomplete or unverifiable lineage.
- High manual effort and cost for preparing documentation.
- Poor visibility in data transformations and dependencies across systems.

EXPECTED SOLUTION:

- Automated extraction of data lineage from at least two technologies (e.g., PySpark + Scala/Java or Python + COBOL/SAS).
- Storage of lineage in OpenLineage or custom JSON format in a NoSQL database (e.g., MongoDB; stretch: Neo4j).
- Self-service UI (React or Angular) to search data elements and visualize lineage as a color-coded, interactive graph.
- Ability to export lineage flows and transformation summaries in Excel or Word for regulatory submission.