Intern Training Plan (4 Weeks) – Based on Data Engineering
Prerequisites

- Understanding of programming Python & SQL
- Fundamental database concepts
- AWS Free Tier account created

NYC yellow Trips Taxi Data

- Yellow Trips Data Dictionary
- Taxi Zone Lookup Table (CSV)
- Yellow Taxi Trip Records (PARQUET)

Week 1 –Foundations of AWS & Data Engineering Basics

Goal: Learn AWS basics, Python , SQL for Data engineering , Data Modeling fundamentals and core intern skills.

- **Day 1: AWS Fundamentals & Core Skills**
    - Introduction to AWS Core Services, IAM basics, Security/Compliance.
    - Services overview: S3, Lambda, CloudWatch, CloudTrail, Glue, AWS PostgreSQL RDS
    - Setup IAM (Identity and Access Management) & billing alerts
    - **Hands-on**:
        - Create S3 bucket (Upload Different File types, Bucket Versioning)
        - GitHub basics (repo setup, PRs)
        - Create IAM users, groups, and roles

Day 2: Python for Data Engineering & Amazon S3 Deep Dive

- Review Python fundamentals , Pandas for data manipulation , API's
- Learn Boto3 (AWS SDK for Python) , JSON and CSV file handling
- Core concepts of s3 buckets, objects, storage classes, versioning, prefixes lifecycle policies
- Learn about S3 storage classes (Standard, IA, Glacier)
- Metadata as governance backbone:
    - Technical metadata (schema, size, format)
    - Business metadata (owner, purpose, classification)
    - Operational metadata (lineage, quality, usage)
- Data classification: Public, Internal, Confidential, Restricted
- Access governance: Who should see what, and why?
- Hands-on Lab:
    - Create S3 buckets with proper naming conventions
    - Configure bucket policies and access controls

- Practice: Upload data, organize with prefixes, set lifecycle rules
- Tag S3 objects with governance metadata (Owner, Classification, Domain)
- Create a metadata manifest file (YAML/JSON) documenting bucket purpose and ownership

Day 3: Data Modeling with Governance Lens

- Review SQL basics, Relational vs No Relational
- Database fundamentals : ACID properties, normalization basics and Data Types with Schema Evolution
- Data Dictionary, Entity-Relationship concepts : Entities, attributes, relationships, cardinality
- Design patterns : Star schema basics, slowly changing dimensions (SCD Type 1 & Type 2)
- Master data vs transactional vs reference data (with business ownership examples)
- Critical data elements (CDEs): What makes data "critical" to govern?
- Hands-on Lab:
  - Practice SQL operation JOINs, aggregations, window functions, Common Table Expressions (CTE)
  - Identify master data domains in NYC taxi dataset with business justification
  - Create a data dictionary with governance attributes:
    - Business owner, Steward, Sensitivity level, Retention policy, Quality rules
  - Use platforms like Hacker Rank, Leet Code (database section)

- **Day 4: Master Data Management Fundamentals**
  - What is MDM and why it matters
  - Master data vs transactional vs reference data
  - Master data domains: Customer, Product, Location, Vendor
  - Golden records and data consolidation
  - Data quality dimensions: accuracy, completeness, consistency, timeliness, validity, uniqueness
  - MDM implementation styles: Registry, Consolidation, Coexistence, Centralized
  - MDM components: repository, integration layer, quality engine, APIs
  - Data stewardship and governance basics
  - Match and merge strategies
  - Data lineage and audit trails
- **Hands-on:**
  - Identify master data in NYC taxi dataset (zones, vendors)
  - Create simple matching algorithm for duplicate detection
  - Design golden record strategy for taxi zones
  - Build master data tables in RDS

- **Day 5: CI/CD & Infrastructure as Code**
  - Git workflows: branching, PRs, code reviews

- CI/CD concepts and benefits
- GitHub Actions for data pipelines
- Automated testing in CI/CD
- Terraform basics: providers, resources, state
- CloudFormation overview
- Environment management (dev/staging/prod)
- **Hands-on:**
  - Create Terraform templates for S3, Lambda, RDS
  - Deploy infrastructure through CI/CD

- Document "Week 1 Learnings" in wiki, data processing application with CI/CD, create RDS database with master data schema, and complete MDM governance framework document.

Week 2: Data Lakes, ETL & Advanced MDM

Goal : Master AWS S3 and storage strategies, understand data lake architecture and Learn basic ETL concepts

- **Day 6: Data Lake Concepts & Modern Storage**
  - Data lake vs warehouse vs data mart
  - Architecture patterns: raw/processed/curated zones
  - AWS Lake Formation basics
  - Metadata management and cataloging
  - Master data layer in data lakes
  - Parquet, ORC, Avro comparison
  - Delta Lake: ACID transactions, time travel, schema evolution
  - Apache Iceberg & Hudi overview
  - Lakehouse architecture
  - **Hands-on Lab**:
    - Design data lake architecture with Draw.io
    - Implement data lake zones in S3
    - Convert data to Delta Lake format
    - Use Delta Lake time travel features

- **Day 7: Introduction to Apache Spark and AWS Glue**
  - Spark is, its lazy evaluation architecture, and its main components
  - AWS Glue Data Catalog - Central metadata repository for all your data and integrations
  - **Glue Job types**: Spark (standard), Python Shell, Streaming ETL with **DPUs** (Data Processing Units) ,Built-in transformations and data quality rules
  - **Glue Crawlers** Configure data sources (S3, RDS, etc.)**,** Schedule crawlers to run automatically**,** Handle schema evolution and partitioning
  - **Hands-on:**
    - Write PySpark scripts for NYC taxi transformations
    - Implement master data lookups/enrichment
    - Optimize Spark jobs

- **Day 8: AWS Glue & Data Quality**

- Glue Data Catalog setup , Glue Crawlers: schema discovery, scheduling
- Glue ETL jobs: visual and script-based
- AWS Glue Data Quality rules
    - Great Expectations framework
    - Data profiling and validation
    - Quality monitoring and alerting
- MDM in ETL:
    - Reference data validation
    - Master data enrichment
    - SCD implementation in Glue
- **Hands-on:**
    - Create Glue crawlers and catalog
    - Build ETL job: CSV to Parquet
    - Implement data quality checks
    - Validate transaction data against master tables

- **Day 9-10: Advanced MDM - Matching & APIs**
    - Data profiling techniques
    - Fuzzy matching algorithms: Levenshtein, Jaro-Winkler
    - Probabilistic matching
    - ML-based matching
    - Survivorship rules
    - RESTful API design for master data
    - CRUD operations, bulk operations
    - Authentication and authorization
    - API documentation with Swagger
    - Change Data Capture (CDC) for master data
    - **Hands-on:**
        - Implement fuzzy matching with Python (fuzzywuzzy, recordlinkage)
        - Build deduplication pipeline for vendors
        - Create data quality scorecard
        - Build API with API Gateway + Lambda for taxi zones

End of Week Deliverable: Document "Week 2 Learnings", build data lake with Delta Lake format, create streaming pipeline with Kafka, implement Glue ETL with data quality checks, and deploy MDM API with matching/deduplication engine.

Week 3: Orchestration, Transformation & Data Warehousing

Goal : Master AWS Glue ETL jobs, Learn Apache Spark fundamentals , Understand data transformation patterns

- **Day 11 : Data Pipeline Orchestration & Monitoring**
    - AWS Step Functions, Event Bridge, workflow orchestration, CloudWatch, logging, cost optimization

- Review AWS Well-Architected Framework for data
- cost optimization techniques and logging strategies
- Build a complete workflow pipeline
- **Hands-On Lab**
    - Create Step Functions state machines to Orchestrate multiple AWS services (Glue, lambda etc.)
    - Implement error handling and retries & Set up CloudWatch alarms and dashboards

- **Day 12-13: Advanced SQL Transformations & Data Quality**
    - Complex SQL patterns for data transformations
    - CTEs (Common Table Expressions) for modularity
    - SQL functions and stored procedures
    - Materialized views for performance
    - SQL testing strategies
    - Version controlling SQL scripts
    - Data validation patterns in SQL
    - Implementing data quality checks
    - Creating reusable SQL quality functions
    - Scheduling SQL transformations with Glue
    - Documenting SQL transformations
    - Hands-on:
        - Create modular SQL transformation scripts
        - Build data quality validation queries
        - Implement SQL-based tests
        - Create documentation for transformations
        - Implement SQL workflows in Glue
        - Version control SQL scripts in Git

- **Day 14-15: SCD Deep Dive & Master Data Versioning**
    - SCD Types 0-6 detailed
    - Temporal tables and bitemporal modeling
    - Surrogate keys vs natural keys
    - Implementation patterns
    - Version control for master data
    - Audit trails and change tracking
    - Point-in-time queries
    - Rollback strategies
    - Hands-on:
        - Implement SCD Type 2 in PostgreSQL RDS
        - Write stored procedures for SCD operations
        - Create PySpark script for SCD with Delta Lake
        - Implement in SQL with version control
        - Query historical data
        -

End of Week Deliverable

End of Week Deliverable: Document "Week 3 Learnings", create end-to-end orchestrated pipeline with Step functions, implement SQL transformations with quality tests and version control

---

Week 4 – Data Warehousing & Analytics

Goal: Learn Data warehouse design techniques, analytics, visualization and master Amazon Redshift

- **Day 16: Amazon Redshift & Dimensional Modeling**
  - Redshift clusters, nodes, slices
  - Distribution styles: KEY, ALL, EVEN, AUTO
  - Sort keys and compression
  - COPY commands for bulk loading
  - Redshift Spectrum for S3
  - Star schema and snowflake schema design
  - Fact and dimension tables
  - Conformed dimensions
  - Aggregate tables
  - MDM in Analytics:
    - Dimension tables as master data
    - Conforming dimensions across marts
  - Hands-on:
    - Provision Redshift cluster
    - Design dimensional model for NYC taxi
    - Load data from S3 using COPY
    - Optimize with distribution/sort keys
    - Run analytical queries
- **Day 17 : Athena & Serverless Analytics**
  - Serverless querying with Presto/Trino
  - Creating external tables
  - Partitioning strategies
  - Query optimization and cost management
  - Integration with Glue Catalog
  - Query performance tuning
  - Columnar format optimization
  - AWS Quick Sight for visualization
  - Dashboard design best practices
  - Hands-on:
    - Deploy dimensional model in Redshift with SCD Type 2, and create analytics dashboard in Quick Sight.
- **Day 18 : Monitoring, Observability & Performance**
  - CloudWatch: logs, metrics, alarms, dashboards
  - CloudTrail for audit logging
  - AWS X-Ray for distributed tracing
  - Data lineage tracking
  - Data observability best practices

- Incident response for data issues
- Query optimization and EXPLAIN plans
- Spark optimization: partitioning, broadcast joins, AQE
- Storage optimization: file sizing, compaction, Z-ordering
- Troubleshooting: OOM errors, data skew, slow queries
- Hands-on:
    - Set up CloudWatch dashboards
    - Create alarms for pipeline failures
    - Implement data lineage visualization

- **Day 19 : Security, Compliance & MDM Governance**
    - AWS KMS and Secrets Manager
    - IAM policies and least privilege
    - PII detection and masking (AWS Macie)
    - Complete governance framework
    - MDM tools: AWS solutions
    - Multi-domain MDM patterns
    - Hands-on:
        - Implement encryption for S3 and RDS
        - Configure VPC with private subnets
        - Set up IAM with least privilege
        - Implement PII masking
        - Design multi-domain MDM architecture

- **Day 20 (Final Demo Day)**
    - Live demo of working system (20 min):
        - Master data operations
        - Data quality monitoring
        - Analytics dashboard
        - **Working Demonstrations:**
            - Batch ETL with AWS Step functions orchestration
            - SQL transformations with data quality tests and version control
            - Master Data Management:
                - RESTful API for master data CRUD
                - Deduplication and matching engine
                - SCD Type 2 implementation
                - Data quality dashboard
            - Analytics dashboard in QuickSight
            - CI/CD deployment in action
            - Monitoring and alerting
    - **Complete End-to-End Data Platform Architecture**

      Draw.io diagram showing:
        - Ingestion: batch (S3)
        - Storage: S3 data lake with zones (raw/processed/curated/master)
        - Processing: Spark/Glue/Lambda with Delta Lake
        - Transformation: SQL scripts with version control

- Serving: Redshift dimensional model + Athena
- Orchestration: AWS Step functions
- Master Data Management layer with APIs
- Monitoring: CloudWatch dashboards
- CI/CD: Cloud formation pipeline