REVATURE

LTIMindtree - Azure Databricks

	MON	TUE	WED	THU	FRI	ENVIRONMENT
Week 1 **Advanced Python, Data Engineering & Big Data Fundamentals**	Project 1 Python Advanced - Collections & Error Handling Collections • Counters • Named Tuples • Ordered Dicts • Default Dict • ChainMap Advanced Exception Handling • Exception hierarchy and custom exceptions • Context managers (with statement) • Exception chaining and traceback Logging & Debugging • Logging module: handlers, formatters, levels	Project 1 Python Advanced - Data Visualization & Testing NumPy & Pandas Advanced • NumPy: array operations, broadcasting, vectorization • Pandas: advanced groupby, pivot tables, merging • Performance optimization techniques • Memory management with large datasets Matplotlib & Data Visualization • Creating professional visualizations • Subplots, multiple axes, and layouts • Customization and styling • Interactive visualizations basics Unit Testing with Pytest • Test-driven development (TDD) principles • Pytest fixtures and parameterization • Mocking and patching • Code coverage analysis	Project 1 Advanced SQL & Database Optimization Advanced Queries - Window Functions & CTEs • Window functions: ROW_NUMBER, RANK, DENSE_RANK, LEAD, LAG • Running totals and moving averages • Common Table Expressions (CTEs) and recursive CTEs • Query structuring best practices Joins & Subqueries Optimization • Join types and performance implications • Correlated vs non-correlated subqueries • EXISTS vs IN performance • Join elimination techniques Database Performance Tuning • Indexes: types, strategies, and trade-offs • Execution plans and query analysis • Statistics and query optimizer • Partitioning and sharding concepts	Project 1 Big Data & Cloud Fundamentals Big Data Architecture • Big Data characteristics: 4Vs (Volume, Velocity, Variety, Veracity) • Big Data components and ecosystem • Distributed computing concepts • CAP theorem and eventual consistency • Data Lifecycle Stages File Formats & Data Types • Structured vs Semi-structured vs Unstructured data • File formats: CSV, JSON, XML, Avro, Parquet, ORC • Columnar vs Row-based storage • Compression algorithms and strategies • Schema evolution considerations Cloud Computing Architecture • Cloud computing models: IaaS, PaaS, SaaS • Azure ecosystem overview: Compute, Storage, Networking • Azure Data Services: ADLS Gen2,	Project 1 Data Warehousing & Modeling Data Warehousing Concepts • OLTP vs OLAP systems • Data Warehouse vs Data Lake vs Lakehouse • Data Warehouse architecture patterns • Operational Data Store (ODS) • Data Mart concepts Dimensional Modeling • Conceptual vs Logical vs Physical modeling • Star Schema design • Snowflake Schema design • Fact tables: additive, semi-additive, non-additive measures • Dimension tables: conformed dimensions Slowly Changing Dimensions & ETL/ELT • SCD Type 0, 1, 2, 3, 4, 6 • Implementation strategies for each type • ETL vs ELT: when to use which • Data quality and	

	MON	TUE	WED	THU	FRI	ENVIRONMENT
			• Views, Materialized Views • Stored Procedures and User Defined Functions ACID Properties & Transaction Management • Transaction isolation levels • Deadlock detection and prevention • Transaction commit, rollback patterns • CRUD operations optimization	Synapse, Databricks • Pricing models and cost optimization	cleansing strategies • BigQuery overview: datasets, tables, queries	
Week 2 **PySpark Deep Dive**	Project 2 Spark Ecosystem & Architecture Big Data Processing & Spark Evolution • Hadoop MapReduce limitations • Spark vs Hadoop: performance comparison • Spark ecosystem: Core, SQL, Streaming, MLlib, GraphX • Use cases for each Spark component Spark Architecture Deep Dive • Driver, Executor, Cluster Manager roles • Spark application lifecycle • Job, Stage, Task hierarchy • Shuffle operations	Project 2 RDD Advanced Operations RDD Advanced Operations • RDD characteristics: Immutable, Distributed, Fault-tolerant • Creating RDDs: parallelize, textFile, external sources • RDD lineage and dependencies • Narrow vs Wide transformations RDD Transformations & Actions • map, flatMap, filter, distinct • reduce, fold, aggregate • collect, count, first, take, foreach • saveAsTextFile and output operations Key-Value Pair	Project 2 DataFrames & Spark SQL Evolution to DataFrames • RDDs vs DataFrames vs Datasets • Benefits of DataFrames: Catalyst optimizer, Tungsten execution • Spark Session API • Schema definition: StructType, StructField • Schema inference vs explicit schema Creating & Managing DataFrames • Creating from: RDDs, JSON, CSV, Parquet, databases • Reading with options: header, delimiter, schema • Column data types and casting • DataFrame	Project 2 Advanced DataFrame Operations Data Aggregation • GroupBy operations: groupBy, rollup, cube • Built-in aggregation functions: sum, avg, count, min, max • agg() with multiple aggregations • pivot and unpivot operations Window Functions • Window specification: partitionBy, orderBy • Ranking functions: row_number, rank, dense_rank • Analytical functions: lead, lag, first, last • Aggregate functions with windows	Project 2 Data I/O & Performance Tuning Data Loading Patterns • Reading CSV, JSON with complex schemas • Reading Parquet, ORC, Avro • Writing with partitioning: partitionBy • Writing with bucketing: bucketBy • Compression codecs: snappy, gzip, lzo External Data Sources • JDBC connections and database reads • Read/Write optimization for databases • Azure Blob Storage and ADLS Gen2 integration • S3 integration patterns	

	MON	TUE	WED	THU	FRI	ENVIRONMENT
	and impact • Memory management: storage vs execution Spark Environment Setup • Local vs Cluster mode configuration • Spark Context vs Spark Session • Configuration parameters and tuning • Spark submit options • Deployment modes: client vs cluster Execution Model & DAG • Lazy evaluation concepts • Transformation vs Action operations • DAG (Directed Acyclic Graph) visualization • Lineage and fault tolerance	Operations • Creating pair RDDs • groupByKey vs reduceByKey vs combineByKey • aggregateByKey and foldByKey • mapValues and flatMapValues • Join operations: join, leftOuterJoin, rightOuterJoin • cogroup operations RDD Persistence & Shared Variables • Storage levels: MEMORY_ONLY, MEMORY_AND_DISK, etc. • Caching strategies and when to use • Broadcast variables: use cases and implementation • Accumulators: built-in and custom • Partitioning: HashPartitioner, RangePartitioner, custom	schema operations DataFrame Operations • select, selectExpr, drop, withColumn, withColumnRenamed • filter, where, distinct • orderBy, sort • limit, sample • Column expressions and functions Spark SQL Integration • Creating temporary views: createOrReplaceTempView • Creating global views • SQL queries on DataFrames • Advanced SQL: CTEs, subqueries, window functions • Catalog API	• Frame specification: rows vs range Joins in Detail • Inner, outer, left, right, left_semi, left_anti • Cross joins and broadcast joins • Join optimization strategies • Broadcast hints • Skewed join handling Set Operations & UDFs • union, unionAll, unionByName • intersect, except (subtract) • User Defined Functions (UDFs): creation and registration • UDF performance considerations • Pandas UDFs (Vectorized UDFs)	• Schema evolution handling Spark Cluster Management • Cluster managers: Standalone, YARN, Kubernetes, Mesos • Driver and executor configuration • spark-submit parameters • Dynamic allocation • Resource allocation strategies Performance Tuning Deep Dive • Memory management and tuning • Serialization: Java vs Kryo • Partition tuning: repartition vs coalesce • Caching strategies • Avoiding common pitfalls	
Week 3 **Databricks & Delta Lake Mastery**	Project 2 Databricks Platform & Setup Lakehouse Architecture • Evolution: Data Warehouse → Data Lake → Lakehouse • Lakehouse architecture principles • Benefits: ACID, schema enforcement, time travel on data lake	Project 2 Delta Lake Fundamentals Delta Lake Overview • ACID transactions in data lakes • Schema evolution and enforcement • Time travel and versioning capabilities • Transaction log architecture Delta Lake vs.	Project 2 Advanced Delta Lake and Medallion Architecture Delta Lake Advanced Features • Liquid Clustering and optimization • Deletion Vectors and performance • OPTIMIZE command and Z-ordering • VACUUM	Project 2 Databricks SQL and Streaming Databricks SQL and Warehousing • SQL warehouses vs. compute clusters • Warehouse configuration and scaling • Query optimization and performance tuning Data Reading and	Project 2 Integration and Advanced Storage Azure Data Lake Storage Integration • ADLS Gen2 connectivity • Mounting strategies and authentication • OAuth and Service Principal setup File Types and	

	MON	TUE	WED	THU	FRI	ENVIRONMENT
	• Databricks Lakehouse platform overview Databricks Architecture • Control Plane vs Data Plane • Azure Databricks architecture components • Account structure and workspace hierarchy • Networking and security architecture Azure Databricks Setup • Account creation and subscription setup • Workspace creation and configuration • Pricing models and SKUs • Cost optimization strategies • Resource tagging and management Workspace Components & DBFS • UI navigation and customization • Cluster management and configuration • Notebook environment and collaboration Databricks FileStore (DBFS) • DBFS structure and file management • Mount points and external storage • File upload and organization	Traditional Formats • Performance comparisons and benefits • Migration strategies from Parquet • File format optimization Basic Delta Operations • Creating and managing Delta tables • Reading and writing Delta format • Basic table operations and metadata Advanced Delta Features • MERGE operations and UPSERT patterns • Slowly Changing Dimensions (SCD) implementation • Soft delete with incremental data	operations and retention policies Delta Table Performance Optimization • Partitioning strategies for Delta • Clustering key selection • File size optimization Medallion Architecture Implementation • Bronze, Silver, Gold layers concept • Data quality progression patterns • Architecture design principles Medallion Implementation • Bronze layer: raw data ingestion • Silver layer: data cleaning and standardization • Gold layer: business-ready datasets	Writing • External table creation and management • Data source connections and formats • Batch processing optimization Auto Loader and Streaming • Auto Loader concepts and implementation • File detection modes and schema evolution • Incremental processing patterns Data Streams • Structured streaming fundamentals • Stream processing and windowing • Real-time analytics patterns	External Integration • Multi-format support and optimization • Compression algorithms and strategies • External system integration patterns Advanced Storage Patterns • File organization and naming conventions • Access pattern optimization • Cross-region data management Data Pipeline Architecture • End-to-end pipeline design • Error handling and recovery strategies • Monitoring and alerting basics	

	MON	TUE	WED	THU	FRI	ENVIRONMENT
Week 4 **Unity Catalog and Advanced Governance**	Project 2 Unity Catalog Setup and Configuration Unity Catalog Introduction • Three-level namespace architecture • Centralized governance and security • Unity Catalog vs. Legacy Hive Metastore Unity Catalog Setup • Account-level metastore creation • Workspace assignment and configuration • Regional considerations and planning Catalog and External Location Management • Creating and managing catalogs • External location configuration • Storage credential management Managed vs. External Tables • Table types and use cases • Location and lifecycle management • Performance implications and best practices	Project 2 Advanced Delta Operations and Security Delta Tables Advanced Operations • MERGE statement syntax and patterns • Conditional updates and complex logic • Change Data Capture (CDC) implementation Incremental Data Processing • Incremental loading strategies • Delta time travel for auditing • Data versioning and rollback Security and Secret Management • Secret scopes and management • Azure Key Vault integration • Databricks-backed vs. Azure Key Vault-backed scopes User Management and Access Control • User, service principal, and group management • Role-based access control (RBAC) • Object-level permissions and inheritance	Project 2 Volumes and Advanced Features Volumes in Unity Catalog • Managed vs. External volumes • Volume management and use cases • File-based operations in volumes Advanced Storage Management • Volume-based file organization • Access pattern optimization • Integration with external storage systems Functions in Unity Catalog • Scalar User Defined Functions (UDFs) • Table-valued functions • Function versioning and management Advanced SQL Functions • Complex analytical functions • Custom aggregation functions • Performance optimization techniques	Project 2 Data Sharing and Federation Delta Sharing • Cross-organization data sharing concepts • Provider and recipient configuration • Databricks Express Edition setup Data Sharing Implementation • Share creation and management • Recipient access and permissions • Monitoring and audit trails Lakehouse Federation • Query federation concepts • Foreign catalog configuration • Cross-platform query execution External Database Integration • JDBC connection setup and optimization • Query pushdown strategies • Hybrid architecture patterns	Project 2 Performance Optimization and Monitoring Advanced Performance Tuning • Partitioning and bucketing strategies • Clustering and optimization techniques • Memory management and resource allocation Query Optimization • Execution plan analysis • Index and statistics usage • Cost-based optimization Monitoring and Observability • Performance monitoring tools • Query profiling and analysis • Resource utilization tracking Troubleshooting and Debugging • Common performance issues • Debug techniques and tools • Error handling and recovery	
Week 5 **Workflows, Orchestration, and Production**	Project 2 Workflow Orchestration and Job	Project 2 Delta Live Tables and Streaming	Project 2 Apache Airflow Integration	Project 2 Databricks-Airflow Integration and DBT	Project 2 Production Deployment	

	MON	TUE	WED	THU	FRI	ENVIRONMENT
	Management **Databricks Workflows** • Job creation and configuration • Multi-task workflows and dependencies • Conditional execution and branching **Notebook Orchestration** • Notebook parameters and communication • Dynamic parameter passing **Advanced Job Features** • For-Each loops and conditional logic • Task failures and recovery strategies • Job scheduling and triggers **Job Monitoring and Management** • Job history and performance tracking • Alert configuration and notifications • Resource optimization for jobs	**Delta Live Tables (DLT) Introduction** • Declarative ETL concepts • DLT vs. traditional ETL approaches • Pipeline architecture and design **DLT Pipeline Development** • Table and view definitions • Data quality constraints • Error handling and retry logic • Incremental processing patterns **Advanced DLT Features** • Change Data Capture (CDC) with DLT • Pipeline monitoring and debugging • Performance optimization techniques **Streaming with DLT** • Real-time data processing • Stream-batch integration • Windowing and aggregation	**Airflow Introduction** • Workflow orchestration concepts • DAGs, operators, and tasks • Airflow architecture overview **Airflow Setup and Configuration** • Installation and database setup • Web UI overview and navigation • Connection and variable management **DAG Development** • DAG design principles and patterns • Task dependencies and relationships • Dynamic and parameterized DAGs **Operators and Hooks** • Common operators usage • Custom operator development • Airflow connections and hooks	**Databricks with Airflow** • Databricks operators and integration • Job submission and monitoring • Parameter passing and error handling **Hybrid Orchestration** • Combining Databricks workflows with Airflow • Architecture patterns and best practices • Performance optimization strategies **DBT Introduction** • Analytics engineering concepts • DBT vs. traditional ETL approaches • Modern data stack integration **DBT with Databricks** • Databricks adapter setup • Model development and testing • Deployment strategies	**Production Architecture** • Environment management (dev/test/prod) • CI/CD pipeline setup • Infrastructure as Code concepts **Monitoring and Observability** • Logging and metrics collection • Alert configuration and response • Performance monitoring strategies	
Week 6 Project	Project 2	Project 2	Project 2	Project 2	Project Presentation	

PROJECT	TECHNOLOGIES
Project 1	Python, SQL, ETL
Project 2	Spark, ETL, Databricks

REVATURE