SEMATEC
IT Training Center

دوره حضوری / آنلاین جامع مهندسی داده Data Engineer

در دوره جامع مهندسی داده به درک عمیقی از اجزای پلتفرم داده خواهید رسید و انواع مختلف پایگاه داده ای رابطه‌ای مانند PostgreSQL و MySQL و پایگاه داده غیر رابطه ای مانند MongoDB ،زبان برنامه نویسی پایتون، Scala، انبار داده, Data Lake , Lakehouse , مدل سازی داده ها، کلان داده ها و تکنیک های جستجو در آن با Elasticsearch ، Apache Spark ،Hadoop، انتقال و پردازش کلان داده ها، ذخیره سازی بهینه، نظارت بر داده ها، کنترل نسخ توزیع شده، استقرار و اجرای برنامه ها، ابزار مدیریت لاگ و مانیتورینگ با Grafana، Data Mesh و تکنیک های آماده سازی داده برای دانشمند علم داده جهت پیاده سازی الگوریتم های یادگیری ماشین بر بستر ویندوز و لینوکس ارائه خواهد شد.
این دوره به شما درک درستی از چرخه مهندسی داده را می دهد که شامل طراحی و معماری پلتفرم های داده ، طراحی انبارهای داده، آماده سازی و گرد آوری، جست جو و تجزیه و تحلیل داده ها داده می باشد.

آموزش دوره جامع مهندسی داده:

دوره جامع مهندسی داده Data Engineer

مدت دوره جامع مهندسی داده:

140ساعت

پیش نیاز دوره جامع مهندسی داده:

بدون پیش نیاز

مخاطب آموزش مهندسی داده:

علاقمندان به ورود به حوزه مهندسی داده ، پردازش کلان داده ها، تحلیلگران داده، دانشمندان علم داده

در انتهای دوره جامع مهندسی داده دانشجویان قادر خواهند بود:

یادگیری ایجاد، طراحی و مدیریت پایگاه داده های رابطه ای و اعمال مفاهیم مدیریت پایگاه داده MySQL, PostgreSQL

توسعه دانش کاری NoSQL و Big Data با استفاده از MongoDB، Hadoop، Apache Spark، Spark SQL، Spark ML و Spark Streaming

مدل‌سازی داده‌ها (نمودارهای موجودیت–رابطه، مفاهیم انبار داده، مفاهیم دریاچه داده، مدل‌سازی ابعادی)

یکپارچه سازی داده ها با ETL, Apache Kafka ,Airflow ,Data pipelines

فناوری های کلان داده (اکوسیستم آپاچی هادوپ، محاسبات توزیع شده)

پیاده سازی روش‌های پاکسازی و اعتبارسنجی داده‌ها و اطمینان از بهره برداری اطلاعات به‌طور مناسب برای کاربران

ایجاد داشبوردهای تعاملی

سرفصل دوره جامع مهندسی داده:

Learn a Programming Language: (Python, Scala)

- **Python**
  - **Install Python and write your first program**
  - **Install IDE (PyCharm, Visual Studio Code)**
  - **Types, Variables, Loops**
  - **Operators, Functions, Conditional**
  - **Built-in Functions**
  - **Build-in data structures (List, Tuple, Dictionary, Set)**
  - **Object Oriented Concepts**
  - **Connection to Database**
  - **Generators, Decorators**
  - **Testing with unit test**
  - **Using libraries**

- Threading & Multiprocessing
- Queue, Stack, Linked list and Tree
- Extracting and Loading data with Python
- Transforming data with Python
- Data Quality checks with Python
- Web Scraping

➢ **Scala**

- Introducing Scala
- Programming Paradigms
- Elements of Programming
- Evaluation Strategies and Termination
- Conditionals and Value Definitions
- Tail recursion
- Working on Assignments
- Cheat Sheet
- Learning Resources
- Higher Order Function
- Currying
- Finding Fixed Points
- Functions and Data
- Evaluations and Operators
- Class hierarchies
- Polymorphism
- Objects Everywhere
- Decomposition
- Pattern Matching
- Lists, Tuples
- Enums
- Subtyping and Generics
- Variance
- Reduction of Lists

Database Fundamentals (NoSQL Databases, Relational Databases)

➢ **NoSQL Databases (MongoDB)**

- Introduction to NoSQL Databases
- Creating and Deploying an Atlas Cluster
- The MongoDB Document Model
- Managing Databases, Collections, and Documents in Atlas Data Explorer
- Data Modeling
- Types of Data Relationships
- Modeling Data Relationships
- Scaling a Data Model
- Using Atlas Tools for Schema
- Using MongoDB Connection Strings
- Connecting to a MongoDB Atlas Cluster
- Troubleshooting MongoDB Atlas Connection Errors
- Using MongoDB Python Client Libraries
- CRUD Operations
- Sorting and Limiting Query Results in MongoDB
- Returning Specific Data from a Query in MongoDB
- Working with MongoDB in Python
- MongoDB Aggregation

- Using MongoDB Aggregation Stages with Python
- Using MongoDB Indexes in Collections
- Working with Compound Indexes in MongoDB
- Using Relevance-Based Search and Search Indexes
- Grouping Search Results by Using Facets
- ACID Transactions in MongoDB
- Setting up your Compass Development Environment
- Writing Data Back to MongoDB
- MongoDB VS Code extension
-

➢ **Relational Databases (PostgreSQL, MySQL)**

- Database Architecture
- Information and Data Models
- ERDs and Types of Relationships
- Mapping Entities to Tables
- Data Types, Modifying Data, Database Constraints
- Relational Model Concepts
- Distributed Architecture and Clustered Databases
- Database Usage Patterns
- Database Objects and Hierarchy (Including Schemas)
- Managing & Join Table
- Loading Data
- Normalization
- Filtering Data
- Querying Data
- Using Subqueries
- Writing Trigger Stored Procedures
- Indexing
- Using Commands
- Grouping and Aggregating Data
- Working with Transactions
- Alternative Storage Engines
- Full-text Search Functions & Operators
- pgAdmin Overview
- Configuration MYSQL
- MySQL AdminAPI
- MySQL InnoDB Cluster
- PostgreSQL Recipes

Data Modeling (Data Warehouse Concepts, Data Lake Concepts, Dimensional Modeling, Entity-Relationship Diagrams, Elasticsearch, Kibana)

➢ **Data Warehouse Concepts, Data Lake Concepts, Dimensional Modeling, Entity-Relationship Diagrams**

- Introduction Data Warehouse
- Data Warehousing Concepts
- Logical Design
- Physical Design
- Hardware and I/O Considerations
- Partitioning Strategy
- Parallel Execution
- Indexes
- Integrity Constraints
- Basic Materialized Views

- Advanced Materialized Views
- Dimensions
- Relational OLAP
- Multidimensional OLAP
- System Manager, Process Manager
- Security
- Tuning
- Extraction, Transformation, and Loading
- Change Data Capture
- Analysis and Reporting
- Hadoop Data Lakes Architecture
- Data Lake Key Concepts
- Challenges and Criticism of Data Lakes
- Data Lake Technology Vendors

➢ **Elastic Search, Kibana**

- Introduction to Elastic Search
- Elastic Search as a distributed framework
- Features
- Terminology
- Elastic Search configuration
- Setting up Elastic Search &Kibana
- Mapping& Analysis
- Elastic Search indexing - behind the scenes
- Execute Elastic Search queries
- Elastic Search scalability
- Elastic Search type mappings
- Inspecting the Cluster
- Searching for Data
- Joining Queries
- Shading
- Replication
- Indexing
- Controlling Query Results
- Aggregations
- Improving Search Results

ETL and Data Integration (ETL Concepts, Data Pipeline Design, ETL Tools) Apache Spark

➢ **Linux LPIC1**

- Introduction
- Determine and configure hardware settings
- Manage shared libraries
- Use Ubuntu package management
- Use RPM and YUM package management
- Work on the command line
- Process Text streams using filters
- Perform basic file management
- Use streams, pipes, and redirects
- Create, monitor, and kill processes
- Search text files using regular expressions
- Work with partitions and filesystems
- Manage file permissions and ownership
- Customize and use the shell environment
- Perform security administration tasks

- Localization and internationalization
- Modify process execution priorities
- Customize and use the shell environment
- System logging
- Accessibility
- System logging
- Mail Transfer Agent (MTA) basics

Apache Spark

- Introduction to Big Data, Apache Spark
- Configuring Apache Spark
- Spark Programming Model
- Spark Data Sources and Sinks
- Setting Apache Spark Configuration
- Spark Data frame and Datasets Transformation
- Aggregations in Apache Spark
- Internal Types, Data Frames, Datasets, RDDs, and the Spark SQL API
- Working with RDDs - Resilient Distributed Datasets
- Cleaning and Transforming Data with Data frames
- Working with Spark SQL, UDFs, and Common Data frame Operations
- Working With Dates and Times, Strings, Arrays
- Evolution of Apache Spark
- Type Conversion of Data frame Columns
- Storage Layout
- Data Skew
- Spark Configurations for Partitions
- Partitioning Recap
- Modifying Data
- Caching Recap

PySpark

- Introduction to PySpark
- Spark Main Components
- PySpark and Python Integration
- Data Processing with PySpark
- PySpark Libraries and Ecosystem
- Resource Management
- Data Serialization
- Data Partitioning
- Broadcast Variables
- Key features of PySpark
- Conditional Statements
- Transformations (Key Aggregations, Sorting, Ranking, Set, Sampling, Partition, Repartition, Coalesce)
- Spark Cluster Execution Architecture
- Spark Shared Variables
- Structured Streaming
- PySpark RDD
- Spark Streaming (Legacy)
- Performance and Optimization
- Pandas API on Spark

- Spark NLP
- Spark GrpahX
- Viewing Data

Big Data Technologies (Apache Hadoop Ecosystem, Distributed Computing) Databricks

➢ Apache Hadoop Ecosystem

- Introduction Apache Hadoop
- Architecture of Apache Hadoop
- Components of the Hadoop Ecosystem
- Core Hadoop (HDFS, YARN, MapReduce)
- Data Access (Apache Pig, Apache Hive)
- Data Storage (HBase, Cassandra)
- Interaction - execution & development (Hcatalog, Crunch, Hama, Solr & Lucene)
- Data Intelligence (Apache Drill, Apache Mahout, Apache Spark)
- Serialization (Apache Avro , Apache Thrift)
- Integration (Apache Chukwa, Apache Sqoop, Apache Flume)
- Management & Support (Apache Zookeeper, Apache Oozie, Apache Ambari)

Data Streaming (Distributed Event Stores, Stream Processing) Apache Kafka

➢ Apache Kafka

- Understand data evolution, the significance of big data, analytics applications, and messaging systems
- key features, components, architecture, and industry use cases
- Set up Kafka environments and install Zookeeper and Kafka
- Kafka producer and consumer basics, Configurations, and operations
- analyze Kafka internals and operations
- techniques for performance tuning in Kafka
- configuration of reliable producer
- Kafka cluster architecture and administration
- Monitor metrics for Kafka brokers
- Manage quotas and monitor consumer lag effectively
- insight into Kafka Streams including tasks
- Perform operations involving K-Streams and K-Tables
- Apache Storm's architecture and topology analysis
- Integrating Kafka Spouts into Apache Storm topologies
- Exploring Apache Spark's components, functions, and practical applications
- configuring Flume connectors for Kafka to HDFS

Data Warehousing (Delta Lake, Data Lake Concepts) Delta Lake, AWS Athena

➢ Data Lake

- Introduction
- Data Lake vs Data Warehouse
- Components and Architectures
- Data Lake Storage
- Data Ingestion
- Crawl and Catalog Data
- Formatting the Data in the Lake
- Partitioning, Compressing, and Compacting the Data in the Lake
- Query Data with Amazon Athena
- Columnar data formats with Amazon Athena
- AWS Lake Formation

- AWS Lake Formation Basic Permission Model
- Data Transformation
- Data Processing with AWS Glue
- Glue Databrew
- Tech Talk - Glue / Athena Federated Queries
- sing Blueprints and Workflows
- Fine-grained Access Control
- Visualizing Data with QuickSight
- Data Movement Scenarios
- Data Sharing Models

Workflow Orchestration (Apache Airflow)

➢ Apache Airflow

- Introduction Apache Airflow
- Core Concepts, Components
- Different Architectures
- Installing Apache Airflow
- Important Vies of the Airflow UI
- Coding your First Data Pipeline with Airflow
- Using New Way of Scheduling DAGs
- Databases and Executors
- Implementing Advanced Concepts in Airflow
- Creating Airflow Plugins with Elastic Search and PostgreSQL
- Using the Bash Operator

Version Control and Collaboration (GitHub, GitLab)

➢ GitHub, GitLab

- Introduction to Git & GitHub
- Installation & authenticate
- Configure & customize
- Working with Branches & Merge in Git
- Terminal
- Merge Conflicts
- Basic Git Workflow with GitHub
- Working With Fork and Clone
- Collaboration in GitHub
- Managing commits
- Work with your remote repo
- SSH Authentication
- Update and Errata
- GitHub Repository
- GitHub Tags and Releases
- Comparing Differences
- Continuous Integration
- Pushing your First Project
- GitHub Issues
- Organization
- Dynamic Data
- Useful Git Features
- Deeply Watching the History in Git
- Cleaning & Organizing History
- Most Used Git Commands for Every Developer
- using git with visual studio

SEMATEC
IT Training Center

Containerization and Orchestration (Docker)

➢ Docker

- Introduction to Docker
- Docker file Architecture & Overview
- Docker Virtual Machines
- Docker vs. Traditional Virtualization
- Docker terminology (Images, Containers, registries and tags)
- Install Docker Desktop & DockerHub
- Pulling images &Running Containers in CLI
- CLI Cheat Sheet
- Docker Compose Explained
- Using the DockerHub Image Registry
- Understanding Image Layers
- Deploymwnt of Containers in Production
- Security
- Managing Docker Images& Containers with portainer
- Docker Commands for Container & Image Management
- Command Structure in Docker
- Docker Container Lifecycle
- Data Persistence in Docker
- Networking in Docker
- Monitoring & Logging in Docker Containers
- Real-World Docker Management Scenarios
- Securing & Optimizing Docker Containers
- Deploying a Secure Docker Application
- Beginning a Career in Docker

Monitoring and Logging (Grafana)

➢ Grafana

- Introduction
- Evolution of Software Architecture and Observability
- Methods of Metric Collection
- Prometheus Full Course (Installation and Use)
- Installing and Configuring Graphite and StatsD
- Installing and Configuring Grafana
- Connecting to a primary data source
- Using Grafana
- Monitoring Windows Servers with InfluxDB, Telegraf and Grafana
- Increasing the visibility of data with logarithmic scaling
- Configuring and connecting data sources to Grafana
- Integration of Grafana with SQL Server, MySQL, Elasticsearch
- Monitoring Google Cloud Platform with out-of-the-box dashboards
- Installing Grafana Loki with Docker
- Visualizing Loki Queries on Dashboards
- Visualization Techniques
- Working with variables and template queries
- Security and User Management
- Configuring and managing alerts in Grafana
- Grafana Plugins and Extensions
- Conclusion and Next Steps
- Monitor Docker Containers with Prometheus and Grafana
- Monitor Elasticsearch with Prometheus and Grafana

- Performance optimization and troubleshooting techniques
- Grafana advance concepts and closure

Advanced Topics (Machine learning Pipelines)

➢ Machine learning Pipelines

- Introduction to Machine Learning and Linear Regression
- Introduction on Machine Learning Pipeline
- Data Preparation
- Formatting Data
- Data Transformation
- Building the Models
- Analyzing the Models