INDEX

[I&D (Big Data) LoT Course Structure 2](#_Toc43311885)

[Oracle SQL Overview 3](#_Toc43311886)

[Unix Fundamentals and Shell Programming 4](#_Toc43311887)

[Introduction to Software Engineering 6](#_Toc43311888)

[Java Programming 6](#_Toc43311889)

[Data Warehouse Concepts 9](#_Toc43311890)

[Data Modeling for Business Intelligence 10](#_Toc43311891)

[ETL Basics 11](#_Toc43311892)

[Software Testing for BI 12](#_Toc43311893)

[Big Data Platform 12](#_Toc43311894)

[Spark in memory 15](#_Toc43311895)

[Introduction to Python and PySpark 16](#_Toc43311896)

[Talend 19](#_Toc43311897)

[Big Data Testing 20](#_Toc43311898)

I&D (Big Data) LoT Course Structure

I&D (BIG DATA) LoT provides exposure to a band of data warehousing technologies. It focuses on application development for data warehouses. The following table lists the course structure for I&D LoT

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl No** | **Course** | **Duration** | **Immersive Approach** |
| **(In Days)** |
| 1 | Discover (Induction) | 1 |  |
| 2 | Soft Skills Day 1 | 1 | Soft Skills Foundation – Part 1 |
| 3 | Java Programming | 7 |  |
| 4 | Oracle SQL Overview | 2 |  |
| 5 | Module 1 Test | 1 | Module Test (MCQ + Coding) |
| 6 | Soft Skills Day 2 | 1 | Soft Skills Foundation – Part 2 |
| 7 | Data Warehouse Concepts | 1 | Sprint 1- Oracle Database as backend & Building the dimension model and Implementing Big Data |
| 8 | Data Modeling for Business Intelligence | 1.5 |
| 9 | ETL Basics | 0.25 |
| 10 | Software Testing for BI | 0.25 |
| 11 | Unix Fundamentals and Shell Programming | 3.5 |
| 12 | Introduction to Software Engineering | 0.5 |
| 13 | Big Data Platform | 7 |
| 14 | Sprint 1 Evaluation | 2 | Sprint 1- Evaluation |
| 15 | Soft Skills Day 3 | 1 | Soft Skills Foundation – Part 3 |
| 17 | Talend | 5 | Sprint 2-  Talend,Spark in memory & Python Programming and Introduction to PySpark |
| 16 | Soft Skills Day 4 | 1 | Soft Skills Foundation – Part 4 |
| 18 | Big Data Testing | 2 |  |
| 19 | Spark in memory | 3 |  |
| 20 | Python Programming and Introduction to PySpark | 4 |  |
| 21 | Sprint 2 Evaluation | 2 | Sprint 2- Evaluation |
| 22 | L1 Test | 1 | Code and Concept based MCQ |
| Total Training Duration | | 48 |  |

**BI (I&D) Curriculum**

**Oracle SQL Overview**

|  |  |
| --- | --- |
| **Program Duration:** 2 days. |  |
|  |  |
| **Contents:** |  |
| Introduction to Database |  |
|  | Introduction to DBMS |
|  | Characteristics of DBMS |
|  | DBMS Models |
|  | Relational DBMS |
|  | Data Integrity |
| Structured Query Language |  |
|  | Interacting SQL using SQL \*Plus |
|  | Using SQL \*Plus |
|  | What is SQL? |
|  | Rules for SQL statements |
|  | Standard SQL Statement Groups |
|  | Basic Data Types |
|  | Rules for naming a Table |
|  | Specifying Integrity Constraints |
|  | DDL Statements: Create, Alter, Drop |
|  | Regular vs Temporary tables |
| Data Manipulation Language |  |
|  | Inserting Rows Into a Table |
|  | Deleting Rows from a Table |
|  | Updating Rows in a Table |
| Database Objects |  |
|  | Index |
|  | Synonym |
|  | Sequence |
|  | Views |
| Data Query Language  (Select Statement) |  |
|  | Select Statement |
|  | Distinct Clause |
|  | Comparison, arithmetic & Logical Operators SQL Operators |
|  | The ORDER BY Clause |
|  | Tips and Tricks |
| Aggregate Functions, Group By and Having Clause |  |
|  | Aggregate Functions |
|  | The GROUP BY Clause |
|  | HAVING Clause |
|  | ROLLUP Operation |
|  | CUBE Operation |
|  | Tips and Tricks |
| SQL (Single Row) Functions |  |
|  | Character Functions |
|  | Number Functions |
|  | Data Conversion Function |
|  | Formats for Date functions |
|  | Date Functions |
|  | Miscellaneous Functions |
|  | Tips and Tricks |
| Transactions |  |
|  | Transaction |
|  | Commit Command |
|  | Rollback and Savepoints |
| Joins and Subqueries |  |
|  | Inner/Equi Join |
|  | Outer Join |
|  | Self-Join |
|  | Subquery |
|  | SUBQUERIES Using Comparison Operators Co-related Subquery |
|  | Exists / Not Exists Operator |
|  | Connect By and Start with clauses |
|  | Tips and Tricks |
| PL/SQL Basics |  |
|  | Introduction to PL/SQL |
|  | PL/SQL Block Structure |
|  | Handling Variables in PL/SQL |
|  | SQL in PL/SQL |
|  | Programmatic Constructs |

**Unix Fundamentals and Shell Programming**

**Program Duration:** 3.5 days.

**Contents:**

Introduction to UNIX Operating System and Basic UNIX commands

• Introduction to UNIX Operating System

• Operating System architecture & components

• Basic UNIX Commands

UNIX File System

• UNIX File System

• File Types

• File Permissions

• I-node entries

• File Related Commands

Unix Environment

• Shells (bash, Korn, C shell)

• System Variables

• Set options

• The Process

• Running commands in background

• Cron scheduler

Filters

• Simple Filters (grep, head, tail & tr)

• Advanced Filters (sort & find)

• Tools (sed & awk)

Vi Editor

• Vi Editor

• Input Mode Commands

• Vi Editor – Save & Quit

• Cursor Movement Commands

Shell Programming

• Shell Variables

• Environmental Variables

• Shell script Commands

• Arithmetic Operations

• Command Substitution

• Command Line Arguments

• Conditional Execution

• if Statement Format

• Test - String Comparison

• The Case Statement

• While Statement

• Break & Continue Statement

• Until Statement

• Shell functions

• Using arrays

**Introduction to Software Engineering**

**Program Duration:** 0.5

**Contents:**

To understand the following:

* What is Software Engineering (SE)
* Common life cycle models
* Phases in SE
* Familiarizing Requirements Phase
* Familiarizing Design Phase
* Familiarizing Construction Phase
* Familiarizing Testing and acceptance Phase
* Review and Configuration Management Process

**Java Programming**

**Program Duration:** 7 days

Introduction to Java

* Introduction to Java
* Features of Java
* Evolution in Java
* Developing software in Java

Eclipse 4.4 (Oxygen/Neon) as an IDE

* Installation and Setting up Eclipse
* Introduction to Eclipse IDE
* Creating and Managing Java Projects
* Use of Java docs
* Miscellaneous  Options

Language Fundamentals

* Keywords
* Primitive Data Types
* Operators and Assignments
* Variables and Literals
* Flow Control: Java’s Control Statements
* Best Practices

Classes and Objects

* Classes and Objects
* Packages
* Access Specifiers
* Constructors - Default and Parameterized
* this reference
* using static keyword
* Best Practices

Exploring Java Basics

* The Object Class
* Wrapper Classes
* Type casting
* Using Scanner Class
* String Handling
* Date and Time API
* Best Practices

Inheritance and Polymorphism

* Inheritance
* Using super keyword
* InstanceOf Operator
* Method & Constructor overloading
* Method overriding
* @override annotation
* Using final keyword
* Best Practices

Abstract Classes and Interfaces

* Abstract class
* Interfaces
* default methods
* static methods on Interface
* Runtime Polymorphism
* Best Practices

Regular Expressions

* Regular Expressions
* Validating data
* Best Practices

Exception Handling

* Introduction
* Exception Types
* Exception Hierarchy
* Try-catch-finally
* Try-with-resources
* Multi catch blocks
* Throwing exceptions using throw
* Declaring exceptions using throws
* User defined Exceptions
* Best Practices

Array

* One dimensional array
* Multidimensional array
* Using varargs
* Using Arrays class
* Best Practices

Collection

* Collections Framework
* Collection Interfaces
* Implementing Classes
* Iterating Collections (using foreach & iterator)
* Comparable and Comparator
* Best Practices

File IO

* Overview of I/O Streams
* Types of Streams
* The Byte-stream  I/O hierarchy
* Character Stream Hierarchy
* Buffered Stream
* The File class
* The Path class
* Object Stream
* Best Practices

Property Files

* What are Property Files?
* Types of Property files
* User defined Properties

Java Database Connectivity

* Java Database Connectivity - Introduction
* Database Connectivity Architecture
* JDBC APIs
* Database Access Steps
* Calling database procedures
* Using Transaction
* Connection Pooling
* DAO Design Pattern
* Best Practices

Lambda Expressions

* Introduction
* Writing Lambda Expressions
* Functional Interfaces
* Types of Functional Interfaces
* Method reference

**Data Warehouse Concepts**

**Program Duration:** 1 day.

**Contents:**

Business Intelligence

* Business Intelligence
* Need for Business Intelligence
* Terms used in BI
* Components of BI

General concept of Data Warehouse

* Data Warehouse
* History of Data Warehousing
* Need for Data Warehouse
* Data Warehouse Architecture
* Data Mining Works with DWH
* Features of Data warehouse
* Data Mart
* Application Areas

Dimensional modeling

* Dimension modeling
* Fact and Dimension tables
* Database schema
* Schema Design for Modeling
* Star
* Snow Flake
* Fact Constellation schema

ETL and Metadata

* ETL process
* Metadata used in ETL
* Metadata in Data Warehousing
* Simple Data warehouse model

Online Analytical Processing (OLAP)

* Online Analytical Processing (OLAP)
* Nature of OLAP analysis
* Types of OLAP
* OLAP Tools
* OLTP and OLAP
* OLAP Functional requirements
* OLAP Fast and Selective
* Operational versus Informational System

Data Mining

* Data mining
* The Knowledge Discovery process
* Need of Data Mining
* Use of Data mining
* Data mining and Business Intelligence
* Types of data used in Data mining
* Data mining applications
* Data mining products
* Data mining market

Best Practices for Building Data Warehouse

* Recipe for a Successful data warehouse
* Data warehouse pitfalls
* Popular BI DW tools and suits
* Trends in BIDW

**Data Modeling for Business Intelligence**

**Program Duration:** 1.5 days

**Contents:**

Introduction to Data Modeling

* Importance of data modeling
* Features of a good data model
* Who should be involved in data modeling
* Database design stages and deliverables
* Classification of information

Understanding Business Requirements

* Need of Requirement Analysis
* Characteristics of a Good Requirement
* The Data Life cycle
* Methods of Collecting requirement
* Business Requirement Specification (BRS)

Conceptual Model

* Define conceptual model
* Objectives of conceptual model
* Components of Conceptual Model
* Types of Modeling
* Entity-Relationship (ER) model
* Types of Attributes
* Join Problems
* Steps of dimension modeling
* Star Schema
* Snowflake Schema
* Bill Inmon Vs Ralph Kimball Approach

Logical Model

* Define logical model
* List features of a logical model
* Transformations required to be done while converting a conceptual model into a
* Logical model
* Activities in table specification
* Activities in column specification
* Activities in Primary key specification

**ETL Basics**

**Program Duration:** 0.25 day

**Contents:**

Basic Concepts

* Data warehouse
* Data warehousing strategies
* Data warehouse architecture
* ETL Meaning
* Need for ETL
* ETL Process
* Operational Considerations

ETL Process

* Data extraction
* Data transformation
* Data Loading

Operational Considerations

* Exceptional Handling
* Alerts and Notification
* Process restart-ability
* Job Scheduling and Monitoring

ETL Tools

* Leading ETL tool vendors
* ETL tool strengths / weaknesses
* Choosing the correct ETL tool

**Software Testing for BI**

**Program Duration:** 0.25 day

**Contents:**

Introduction to Software testing for BI

* Business requirements
* BI Project versus BI Program
* How is BI testing different from traditional code based testing?
* BI SDLC

Testing concepts

* What is testing? Testing – Why? Testing – How?
* Principles of Testing
* Test Case and Test Suite
* Testing scope
* Test Strategy
* Verification and Validation

Types of Testing

* Static Testing, Dynamic Testing, Automated testing
* V Model for BI Testing

Testing for BI

* Testing document purpose (Test documentation)
* General BI Testing Principles
* BI Testing Mission
* Production Verification Testing
* Possible Areas of Automation

**Big Data Platform**

**Program Duration:** 7 days

**Contents:**

Big Data Overview

* What’s Big Data?
* Big Data: 3V’s
* Explosion of Data
* What’s driving Big Data
* Applications for Big Data Analytics
* Big Data Use Cases
* Benefits of Big Data

Hadoop(HDFS)

* History of Hadoop
* Distributed File System
* What is Hadoop
* Characteristics of Hadoop
* RDBMS Vs Hadoop
* Hadoop Generations
* Components of Hadoop
* HDFS Blocks and Replication
* How Files Are Stored
* HDFS Commands
* Hadoop Daemons

Hadoop 2.0 & YARN

* Difference between Hadoop 1.0 and 2.0
* New Components in Hadoop 2.x
* YARN/MRv2
* Configuration Files in Hadoop 2.x
* Major Hadoop Distributors/Vendors
* Cluster Management & Monitoring
* Hadoop Downloads

Map Reduce

* What is distributed computing
* Introduction to Map Reduce
* Map Reduce components
* How MapReduce works
* Word Count execution
* Suitable & unsuitable use cases for MapReduce

Sqoop

* Architecture
* Basic Syntax
* Import data from a table in a relational database into HDFS
* import the results of a query from a relational database into HDFS
* Import a table from a relational database into a new or existing Hive table
* Insert or update data from HDFS into a table in a relational database

Flume

* Given a Flume configuration file, start a Flume agent
* Given a configured sink and source, configure a Flume memory channel with a specified capacity

Hive Programming

* Define a Hive-managed table
* Define a Hive external table
* Define a partitioned Hive table
* Define a bucketed Hive table
* Define a Hive table from a select query
* Define a Hive table that uses the ORCFile format
* Create a new ORCFile table from the data in an existing non-ORCFile Hive table
* Specify the delimiter of a Hive table
* Load data into a Hive table from a local directory
* Load data into a Hive table from an HDFS directory
* Load data into a Hive table as the result of a query
* Load a compressed data file into a Hive table
* Update a row in a Hive table
* Delete a row from a Hive table
* Insert a new row into a Hive table
* Join two Hive tables
* Run a Hive query using Tez
* Run a Hive query using vectorization
* Output the execution plan for a Hive query
* Use a subquery within a Hive query

Catalog

* Hcatalog

Kafka

* Need for ingestion of data streams
* Introduction to Kafka
* Kafka Cluster Architecture
* Producers, Consumers & brokers
* Messages, Topics & Partitions
* Interaction with zookeeper
* Java APIS
* Kafka & Flume integration

Scala

* An overview of functional programming
* Why Scala?
* REPL
* Working with functions
* objects and inheritance
* Working with lists and collections

**Spark in memory**

**Program Duration:** 3 days

**Contents:**

SPARK Basics

* What is Spark?
* History of Spark
* Spark Architecture
* Spark Shell

Working with RDDs in Spark

* RDD Basics
* Creating RDDs in Spark.
* RDD Operations.
* Passing Functions to Spark.
* Transformations and Actions in Spark
* Spark RDD Persistence

Working with Key/Value Pairs

* Pair RDDs
* Transformations on Pair RDDs
* Actions Available on Pair RDDs.
* Data Partitioning (Advanced)
* Example: PageRank
* Loading and Saving the Data.

Spark Advanced

* Accumulators
* Broadcast Variables
* Piping to External Programs
* Numeric RDD Operations
* Spark Runtime Architecture
* Deploying Applications

SPARK with SQL

* Spark SQL Overview
* Spark SQL Architecture
* Catalyst
* Plan Optimization & Execution
* ROW API

Spark streaming

* What is Spark streaming?
* Spark streaming : How it works ?
* Spark DStreams
* A Twitter example
* Fault-tolerance
* Stateful Stream Processing

pyspark introduction

**Introduction to Python and PySpark**

**Program Duration:** 4 Days

**Contents:**

1 1: Introduction to python and history

* 1. What is Python? And Why Python?
  2. Applications of Python
  3. Salient Features of Python

1. : Installation
   1. Installation on windows
   2. Setting up environment variables
   3. Installation on linux
   4. Installation on Mac
   5. A brief on Python IDEs
2. : Variables
   1. Dos and donts in variable declaration
   2. Commenting
   3. Indentation
   4. Printing
   5. Taking input
   6. Datatypes
   7. Using help(), dir(), type() functions
3. : String functions
   1. string functions
   2. slicing
   3. indexing, negative indexing
   4. math functions
4. : Operators
   1. Logical Operator
   2. Bitwise Operator
   3. Assignment Operator
   4. Arithmetic Operator
   5. Conditional Operator
   6. Membership Operator
   7. Identity Operator
5. : Decision Making
   1. Using IF
   2. Using IF ELSE
   3. Using IF ELIF ELSE
   4. Using ternary operator
6. : Program Flow / Recursion
   1. for loop
   2. range()
   3. break, continue, pass
   4. while loop
7. : Data Structures
   1. Lists
   2. List methods
   3. Tuple
   4. Packing and unpacking
   5. Zip() function
   6. Dictionary
   7. Ditionary methods
   8. Sets
   9. Set methods
   10. List comprehention, dictionary comprehension, set comprehension
8. : Functions and advanced functions
   1. creating a function
   2. types of argument passing
   3. using Lambda, map(), filter(), reduce()
   4. generators
   5. iterator
   6. decorator
9. : Some of Core Modules [The Batteries !!]
   1. importing a module
   2. datetime, os, sys, random, math, string modules
   3. creating & importing own module
   4. use of \_\_name\_\_ == ‘\_\_main\_\_’ function
10. : Regular Expressions, ‘re’ module
    1. creating a pattern
    2. use of wild cards
    3. use of meta characters
    4. RE methods
11. : Logging
    1. introduction to logging
    2. levels of logging
    3. saving to a log file
    4. log string formatters
    5. file handlers
12. : Basic file management
    1. opening a file
    2. reading a file
    3. writing into a file
    4. using ‘with’ keyword
    5. file functions
    6. reading CSV file and writing into CSV file
13. : Database module
    1. installing database module
    2. introduction to SQLite
    3. creating a connection
    4. creating cursor
    5. executing queries
    6. using commit(), lastrowid.. etc functions
14. : Python testing
    1. installing pytest
    2. writing a test cases
    3. using parameterize
    4. using markers
15. : Object Oriented Programming
    1. creating class
    2. creating object
    3. using self
    4. using \_\_init\_\_() method
    5. using setattr(), hasattr(), delattr() methods
    6. using @classmethod and @staticmethod
    7. inheritance
    8. polymorphism
    9. operator overloading
    10. access modifiers
16. : Multithreading and Multi Processing
    1. importing threading module
    2. usint start(), run() join() methods
    3. controlling threads
    4. All you need to know about ‘GIL’
17. : Error and exception Handling in Python
    1. Core Exception handling concept
    2. Custom Exceptions
    3. Importance of Messaging.
18. Introduction to Spark and Pyspark API
    1. Overview of Saprk 2.0 and Pysaprk
    2. Spark Data Structures
    3. Deep dive in to RDDs
    4. Deep dive in to SparkSQL .

**Talend**

|  |  |
| --- | --- |
| **Program Duration:** 5 days  Talend Basics | Pre-requisites |
|  | Introduction |
|  | Architecture |
| Talend Data Integration | Installation and Configuration |
|  | Repository |
|  | Projects |
|  | Metadata Connection |
|  | Context Parameters |
|  | Jobs / Job lets / Components |
|  | Important Components |
|  | Deployment |
| Talend Big Data | DI Vs BD |
|  | BD Components |
|  | Configuring and running the jobs on remote server from Studio |
|  | Configuration and usage of Azure components |
|  | Configuration and usage of AWS components |
| Talend Cloud | Introduction, configuration and job deployment |
| Talend Administration center | Users |
|  | Scheduling |
|  | Job Server |
| Lab Exercises | Activities/Q&A, Sample jobs |

**Big Data Testing**

**Program Duration:** 2 day

**Contents:**

Big Data Testing Overview

       Characteristics of Big Data (3Vs)

       Why testing is needed?

       Challenges in Big Data testing

       Traditional testing vs Big data testing

       Performance Testing in Big Data

       Testing strategy

       Functional and Non-functional testing

       MRUnit, PigUnit, Spark, Beetest and HBase Testing

       Cluster sanity test

       Testing Cluster Security