PySpark Certification T...

DataFrames and Spark SQL > Presentation

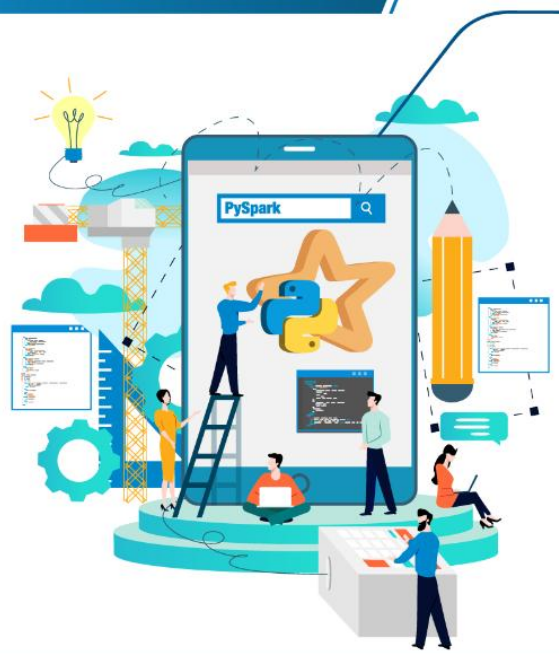
Course Content

- Presentation
- Class 6 Recording
- Case Study I - Perfor...
- Case Study II - Perfor...
- Case Study Solution
- In-Class Demo - Inpu...
- Dataset
- Dataset
- Spark Sql Dataframe...

Personal Library

Presentation 3 / 93

COURSE OUTLINE



MODULE 06

Introduction to Big Data Hadoop and Spark

Introduction to Python for Apache Spark

Functions, OOPs, Modules in Python

Deep Dive into Apache Spark Framework

Playing with Spark RDDs

Data Frames and Spark SQL

Machine Learning using Spark MLlib

Deep Dive into Spark MLlib

Understanding Apache Kafka and Apache Flume

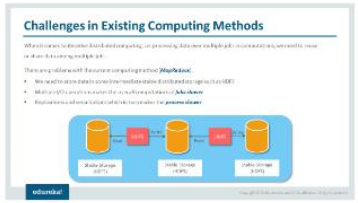
Apache Spark Streaming – Processing Multiple Batches

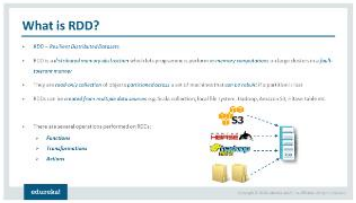
Apache Spark Streaming - Data Sources

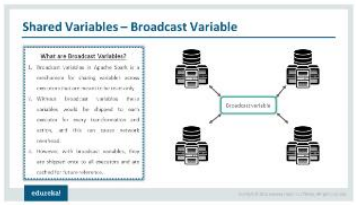
Implementing an End-to-End Project

Presentation 4 / 93

Recap





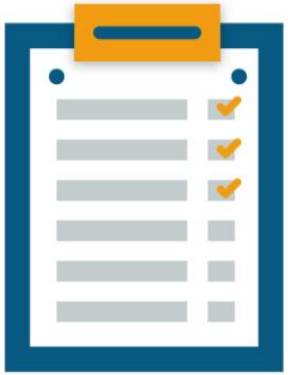


edureka!

Presentation 5 / 93

Topics

- Why Spark SQL ?
- Spark SQL Success Story
- What is Spark SQL?
- Spark SQL Features
- Comparison between Hive, Impala & Spark SQL
- Spark SQL Architecture
- Spark SQL Libraries
- Data Source API
- DataFrame API
- SQL interpreter and Optimizer
- SQL Service
- What is SQL context
- Creating Dataframe
- User Defined Function
- Inferring the Schema using Reflection
- Hive integration with Spark SQL
- Analysis Using Spark SQL



edureka!

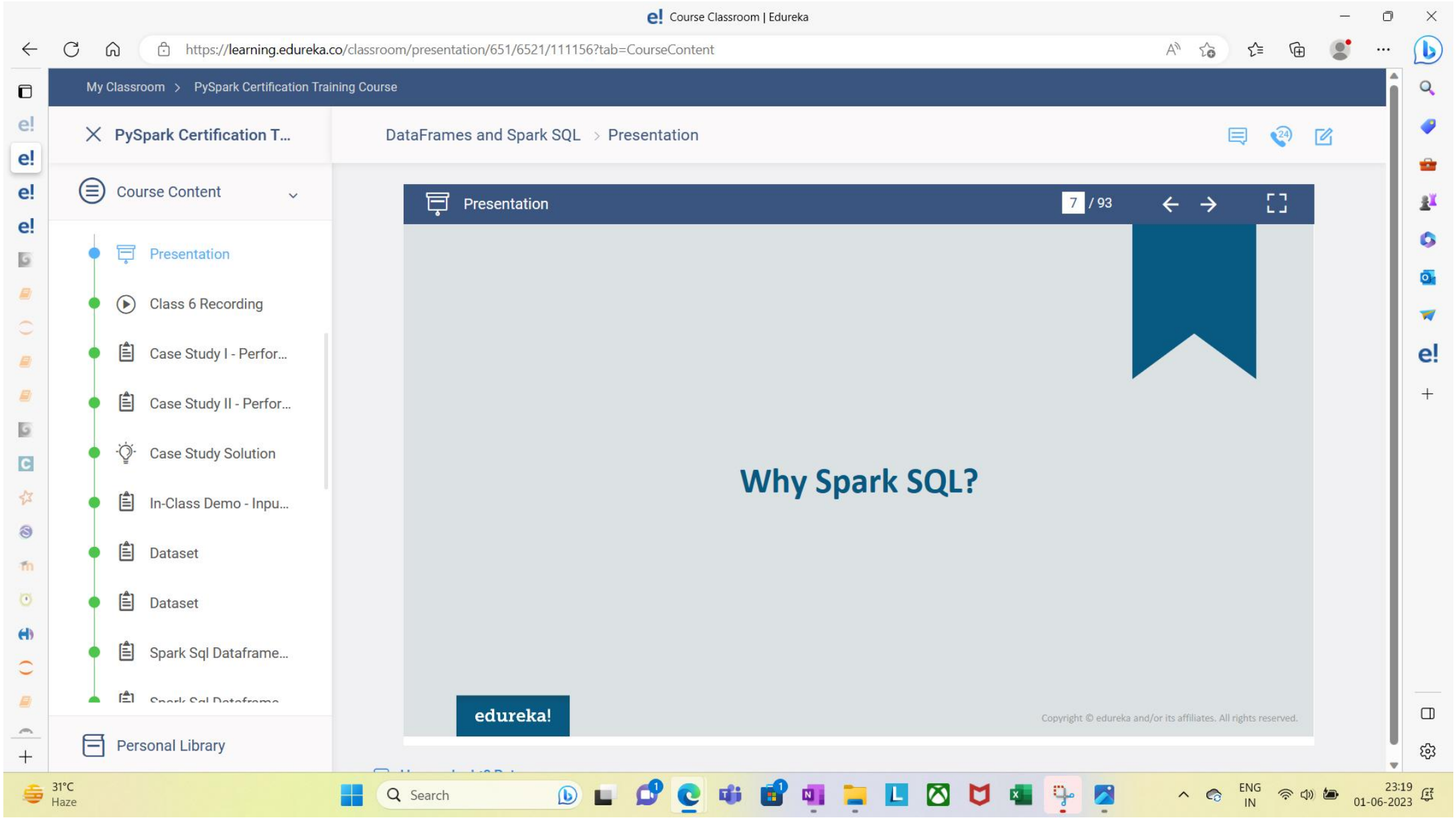
Presentation 6 / 93

Objectives

After completing this module, you should be able to:

- Describe What is Spark SQL
- Understand Spark SQL Architecture
- Describe SQL Context in Spark SQL
- Work with Data Frames
- Interoperate with RDDs
- Understand JSON and Parquet File Formats
- Integrate Spark with Hive

edureka!


e! Course Classroom | Edureka
https://learning.edureka.co/classroom/presentation/651/6521/111156?tab=CourseContent
My Classroom > PySpark Certification Training Course
PySpark Certification T...
DataFrames and Spark SQL > Presentation
Course Content
Presentation
Class 6 Recording
Case Study I - Perfor...
Case Study II - Perfor...
Case Study Solution
In-Class Demo - Inpu...
Dataset
Dataset
Spark Sql Dataframe...
Personal Library
Presentation
7 / 93
Why Spark SQL?
edureka!
Copyright © edureka and/or its affiliates. All rights reserved.
31°C
Haze
Search
ENG
IN
23:19
01-06-2023

Presentation 8 / 93

Why do we need Spark SQL?

Spark SQL

Spark SQL was built to overcome the limitations of Apache Hive running on top of Spark

Limitations of Apache Hive

- Hive uses MapReduce which lags in performance with medium and small sized datasets (<200 GB)

- No resume capability

- Hive cannot drop encrypted databases

edureka!

Spark SQL Advantages over Hive

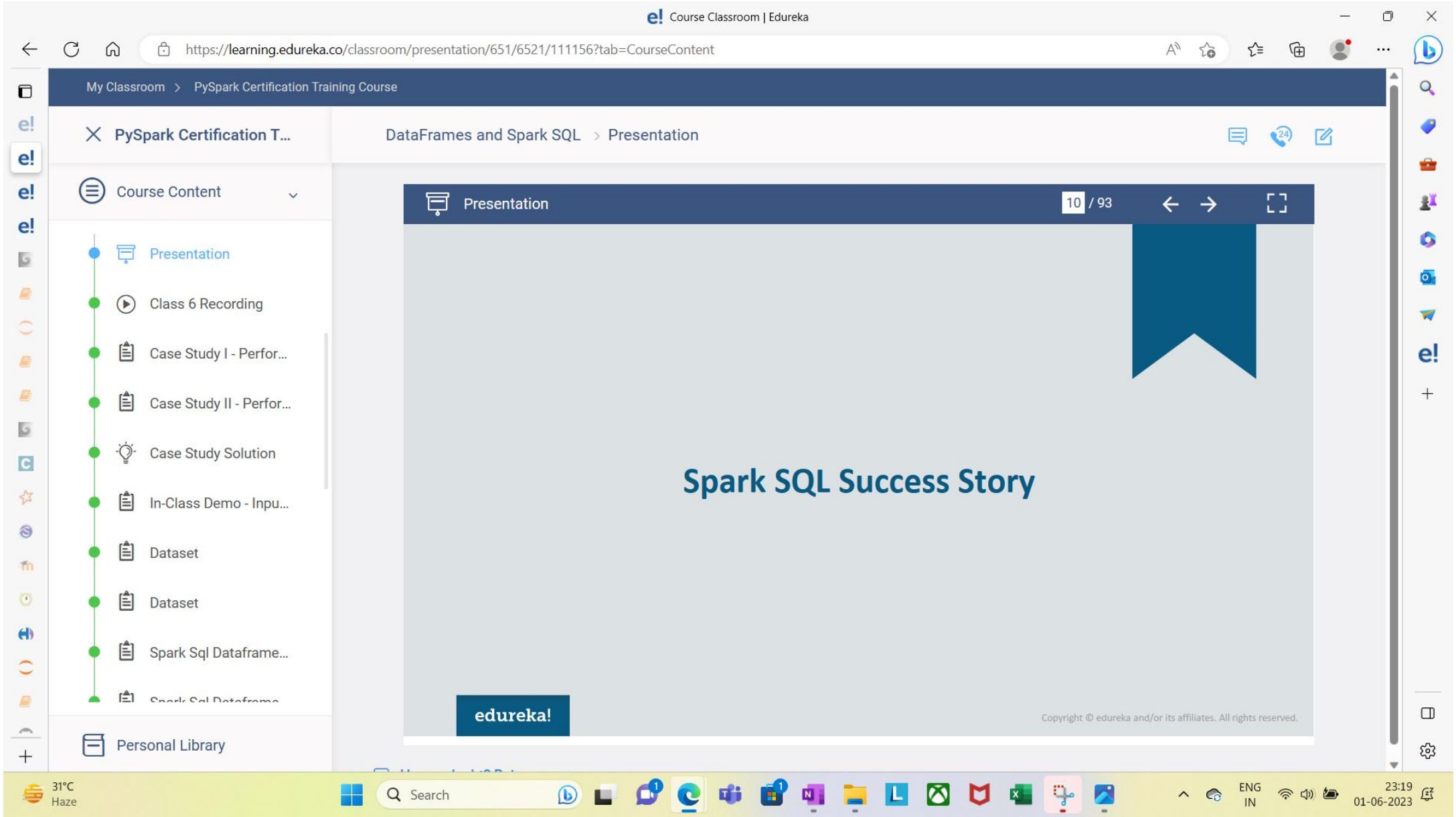
Advantages

1. Faster execution — HIVE → 600 secs; Spark SQL → 50 secs
2. No migration hurdles — HIVE → Spark SQL
3. Real time querying — hadoop Batch

How?

Spark SQL uses the metastore services of Hive to query the data stored and managed by Hive

edureka!


Course Classroom | Edureka
https://learning.edureka.co/classroom/presentation/651/6521/111156?tab=CourseContent
My Classroom > PySpark Certification Training Course
PySpark Certification T...
DataFrames and Spark SQL > Presentation
Course Content
Presentation
Class 6 Recording
Case Study I - Perfor...
Case Study II - Perfor...
Case Study Solution
In-Class Demo - Inpu...
Dataset
Dataset
Spark Sql Dataframe...
Personal Library
Presentation
10 / 93
Spark SQL Success Story
edureka!
Copyright © edureka and/or its affiliates. All rights reserved.
31°C
Haze
Search
ENG
IN
23:19
01-06-2023

Presentation 11 / 93

Spark SQL Success Story

Twitter Sentiment Analysis With Spark SQL

Trending Topics can be used to create campaigns and attract larger audience

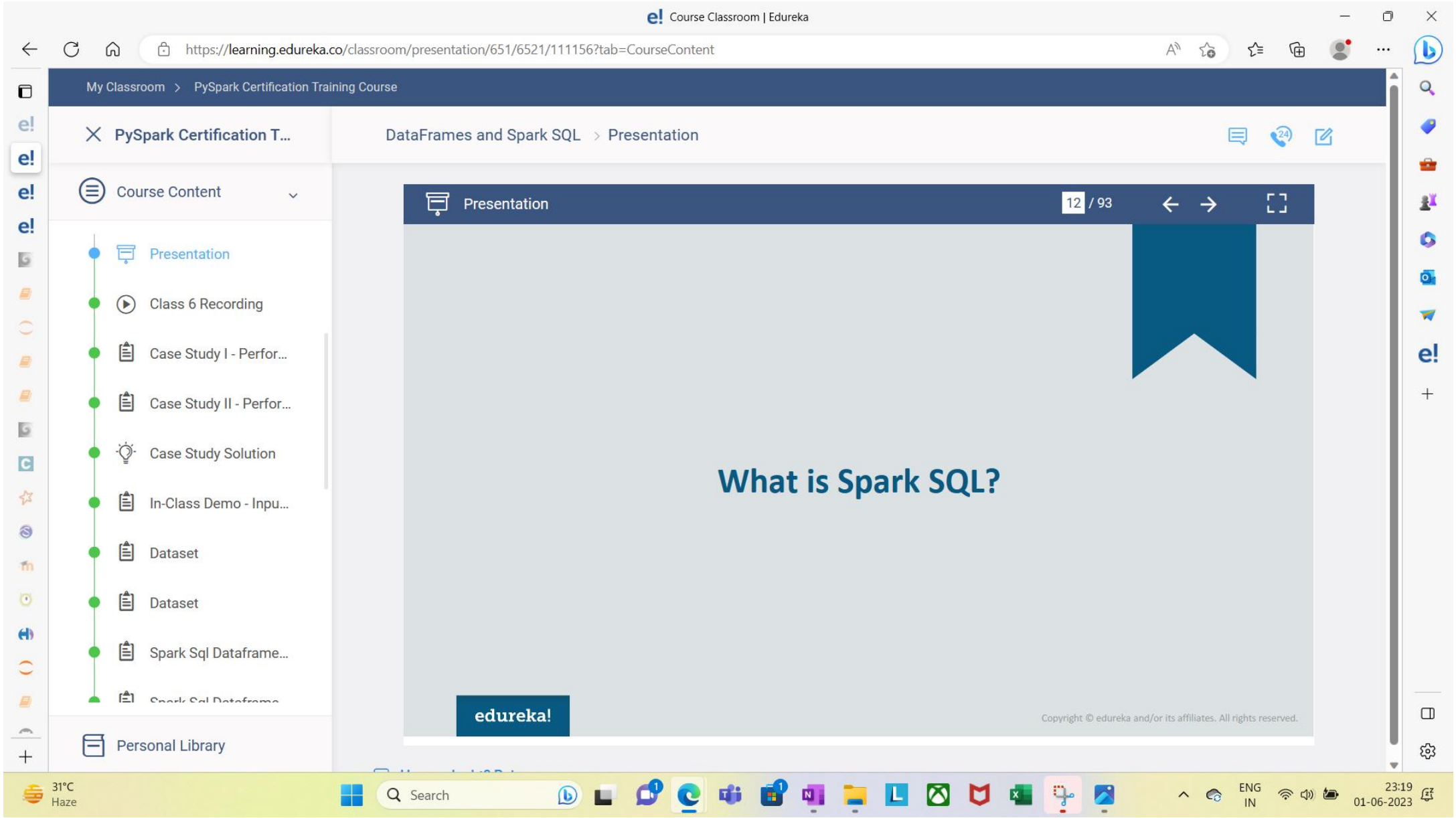
Sentiment helps in crisis management, service adjusting and target marketing

NYSE: Real Time Analysis of Stock Market Data

Banking: Credit Card Fraud Detection

Genomic Sequencing

edureka!


Course Classroom | Edureka
https://learning.edureka.co/classroom/presentation/651/6521/111156?tab=CourseContent
My Classroom > PySpark Certification Training Course
PySpark Certification T...
DataFrames and Spark SQL > Presentation
Course Content
Presentation
Class 6 Recording
Case Study I - Perfor...
Case Study II - Perfor...
Case Study Solution
In-Class Demo - Inpu...
Dataset
Dataset
Spark Sql Dataframe...
Personal Library
Presentation
12 / 93
What is Spark SQL?
edureka!
Copyright © edureka and/or its affiliates. All rights reserved.
31°C
Haze
Search
ENG
IN
23:19
01-06-2023

Presentation 13 / 93

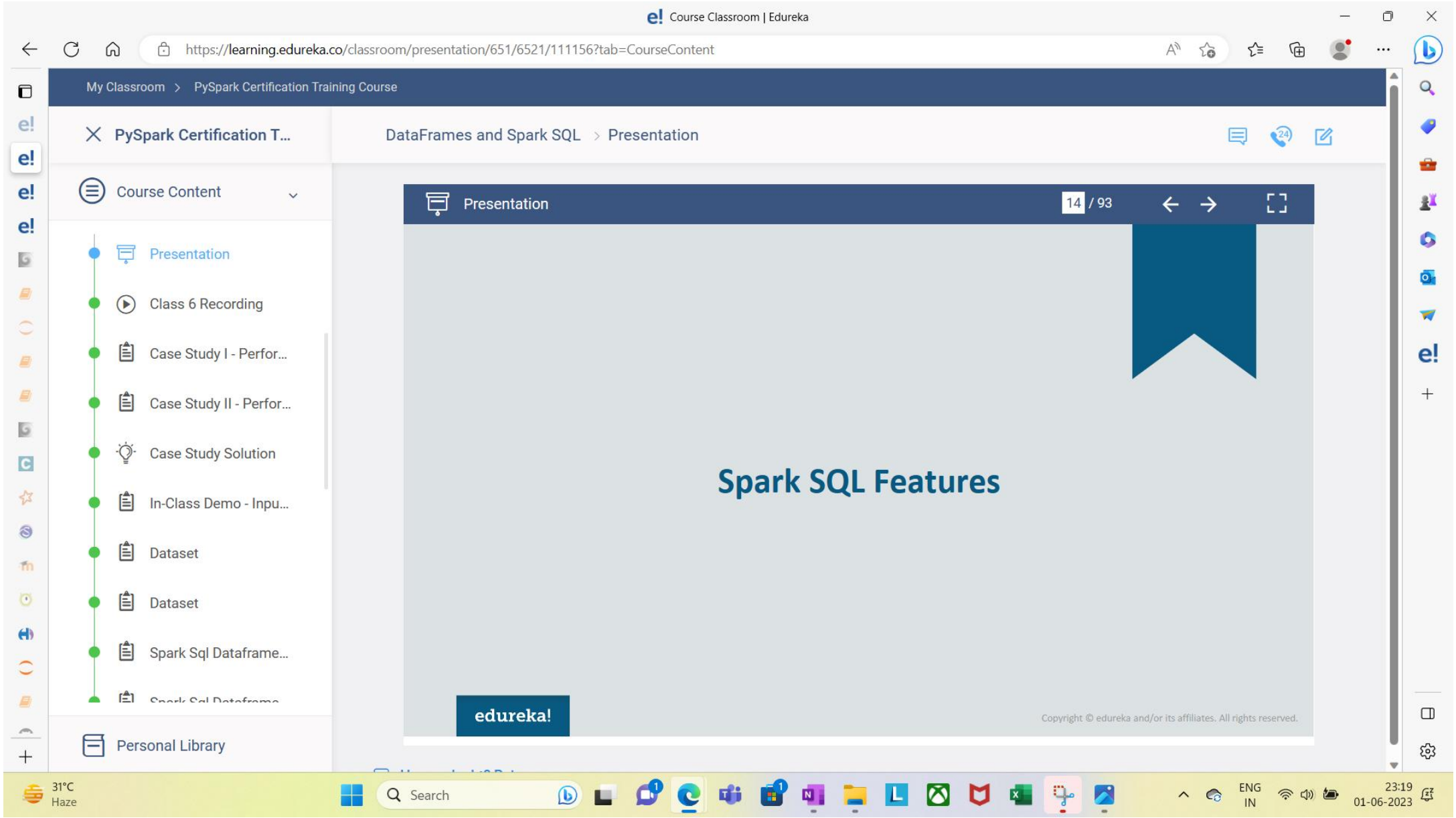
What is Spark SQL?

Spark SQL is a Spark module for structured data processing

Spark SQL Capabilities

1. It provides a DataFrame abstraction in Python, Java, and Scala to simplify working with structured datasets. DataFrames are similar to tables in a relational database
2. It can read and write data in a variety of structured formats (e.g., JSON, Hive Tables, and Parquet)
3. It lets you query the data using SQL, both inside a Spark program and from external tools

edureka!


Course Classroom | Edureka
https://learning.edureka.co/classroom/presentation/651/6521/111156?tab=CourseContent
My Classroom > PySpark Certification Training Course
PySpark Certification T...
DataFrames and Spark SQL > Presentation
Course Content
Presentation
Class 6 Recording
Case Study I - Perfor...
Case Study II - Perfor...
Case Study Solution
In-Class Demo - Inpu...
Dataset
Dataset
Spark Sql Dataframe...
Personal Library
Presentation
14 / 93
Spark SQL Features
edureka!
Copyright © edureka and/or its affiliates. All rights reserved.
31°C
Haze
Search
ENG
IN
23:19
01-06-2023

Presentation 15 / 93

Spark SQL Features

1 Spark SQL integrates relational processing with Spark's functional programming

2 Spark SQL is used for the structured/semi structured data analysis in Spark

edureka!

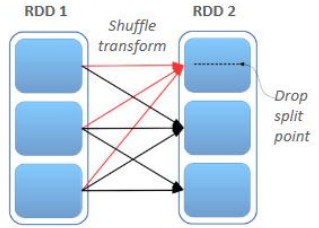
Presentation 16 / 93

Spark SQL Features

3 Support for various data formats

Parquet JSON HIVE

4 SQL queries can be converted into RDDs for transformations



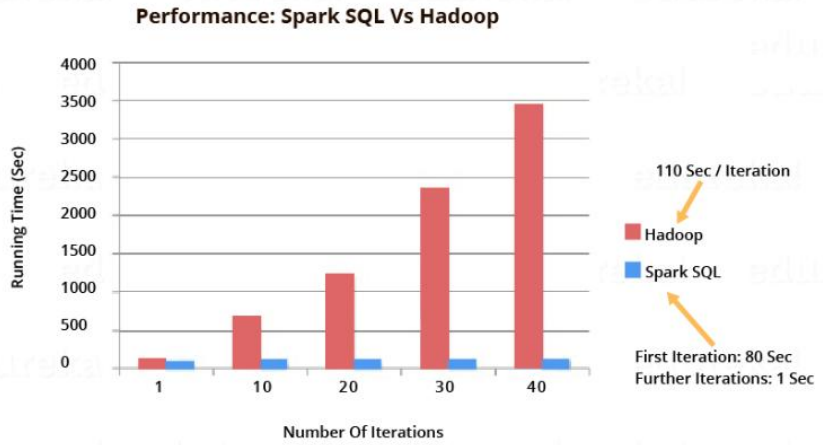
Invoking RDD 2 computes all partitions of RDD 1

edureka!

Spark SQL Features

5

Performance And Scalability



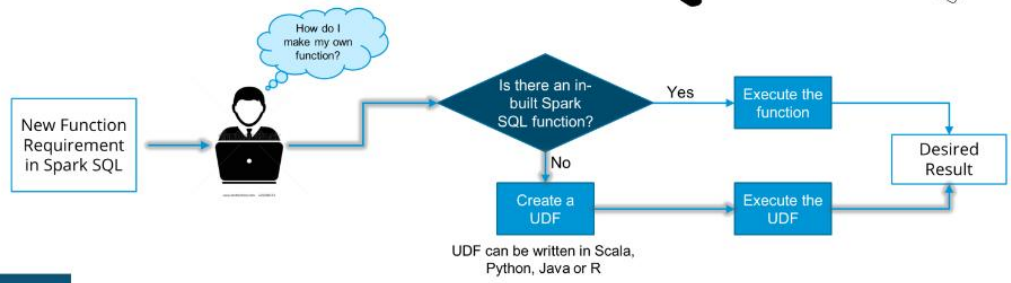
edureka!

Presentation 18 / 93

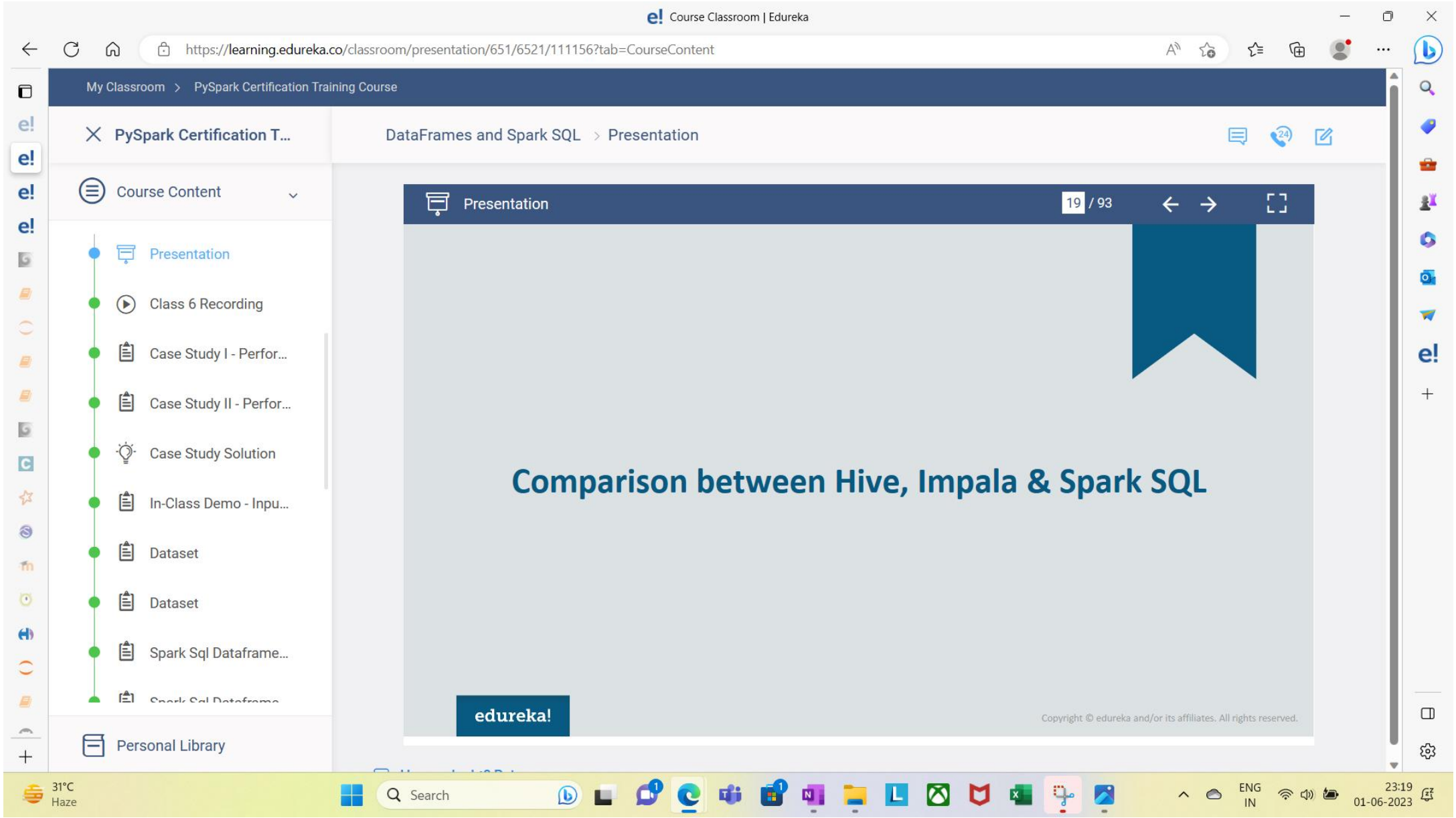
Spark SQL Features

6 Standard JDBC/ODBC Connectivity

7 User Defined Functions lets users define new Column-based functions to extend the Spark vocabulary



edureka!

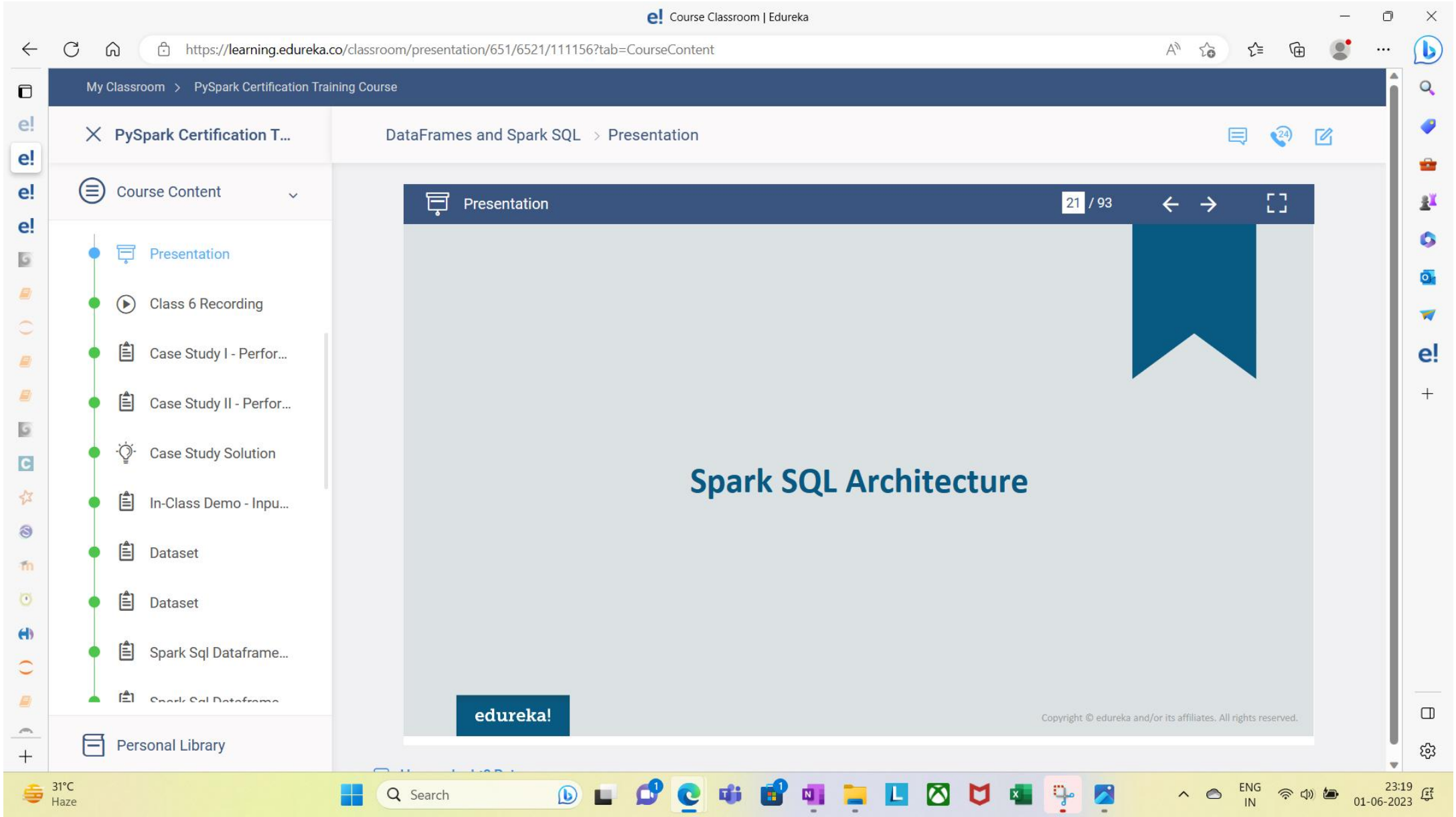

e! Course Classroom | Edureka
https://learning.edureka.co/classroom/presentation/651/6521/111156?tab=CourseContent
My Classroom > PySpark Certification Training Course
PySpark Certification T...
DataFrames and Spark SQL > Presentation
Course Content
Presentation
Class 6 Recording
Case Study I - Perfor...
Case Study II - Perfor...
Case Study Solution
In-Class Demo - Inpu...
Dataset
Dataset
Spark Sql Dataframe...
Personal Library
Presentation
19 / 93
Comparison between Hive, Impala & Spark SQL
edureka!
Copyright © edureka and/or its affiliates. All rights reserved.
31°C
Haze
Search
ENG
IN
23:19
01-06-2023

Presentation 20 / 93

Choosing the right SQL Engine

	Apache Hive	Apache Impala	Apache Spark SQL
Users	ETL Developers	Business Analysts	Data Engineers & Data Scientists
Strengths	▪ Built for very long running ETL, data processing or batch processing ▪ Supports custom file formats ▪ Handles massive ETL sorts with joins	▪ Scales to high-concurrency ▪ Supports high-performance interactive SQL ▪ Compatible with BI tools & skills ▪ Hadoop integration and usability	▪ Easily embed SQL into Java, Scala or Python applications ▪ Simple language for common operations ▪ Seamlessly mix SQL & Spark code within a single applications ▪ Automatic performance optimizations

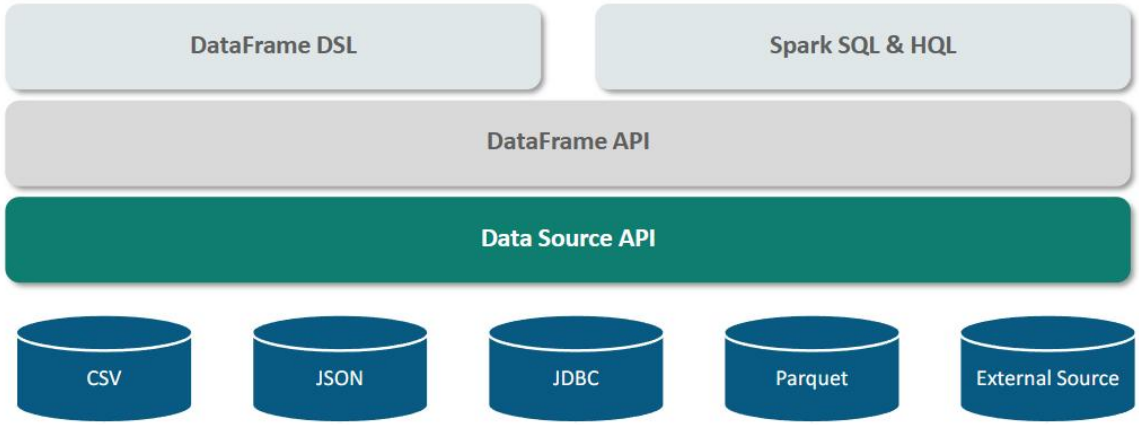
edureka!


Course Classroom | Edureka
https://learning.edureka.co/classroom/presentation/651/6521/111156?tab=CourseContent
My Classroom > PySpark Certification Training Course
PySpark Certification T...
DataFrames and Spark SQL > Presentation
Course Content
Presentation
Class 6 Recording
Case Study I - Perfor...
Case Study II - Perfor...
Case Study Solution
In-Class Demo - Inpu...
Dataset
Dataset
Spark Sql Dataframe...
Personal Library
Presentation
21 / 93
Spark SQL Architecture
edureka!
Copyright © edureka and/or its affiliates. All rights reserved.
31°C
Haze
Search
ENG
IN
23:19
01-06-2023

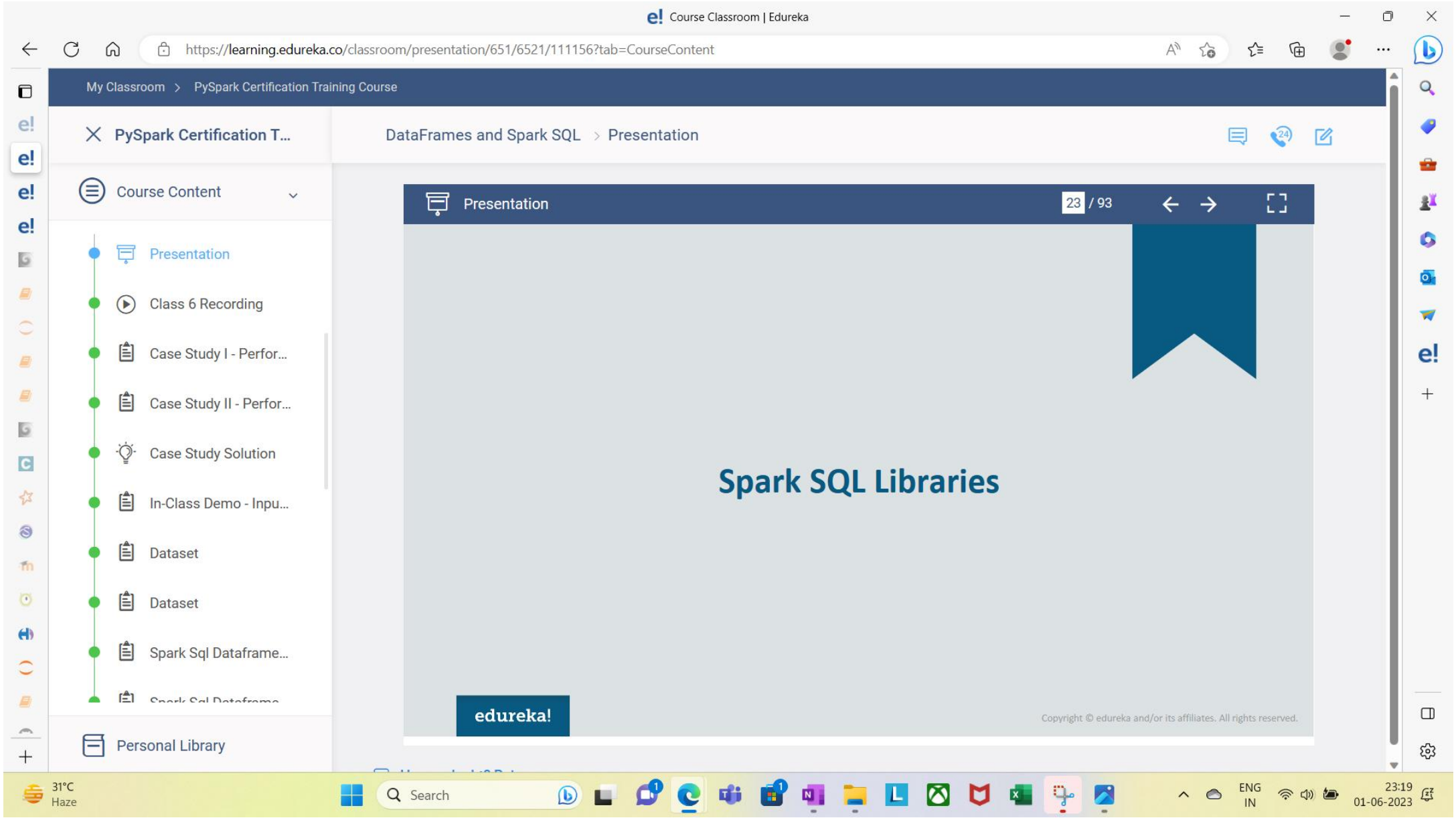
Presentation 22 / 93

Spark SQL Architecture

Architecture Of Spark SQL



edureka!

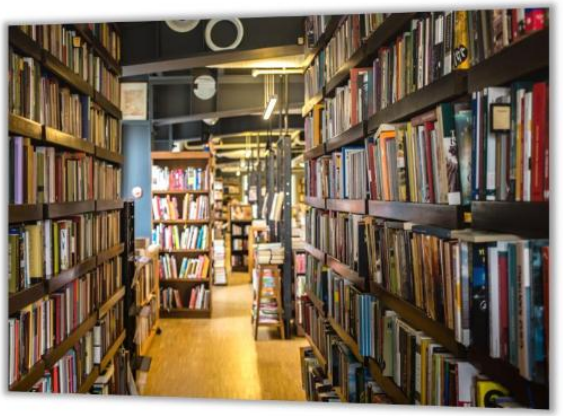

Course Classroom | Edureka
https://learning.edureka.co/classroom/presentation/651/6521/111156?tab=CourseContent
My Classroom > PySpark Certification Training Course
PySpark Certification T...
DataFrames and Spark SQL > Presentation
Course Content
Presentation
Class 6 Recording
Case Study I - Perfor...
Case Study II - Perfor...
Case Study Solution
In-Class Demo - Inpu...
Dataset
Dataset
Spark Sql Dataframe...
Personal Library
Presentation
23 / 93
Spark SQL Libraries
edureka!
Copyright © edureka and/or its affiliates. All rights reserved.
31°C
Haze
Search
ENG
IN
23:19
01-06-2023

Presentation 24 / 93

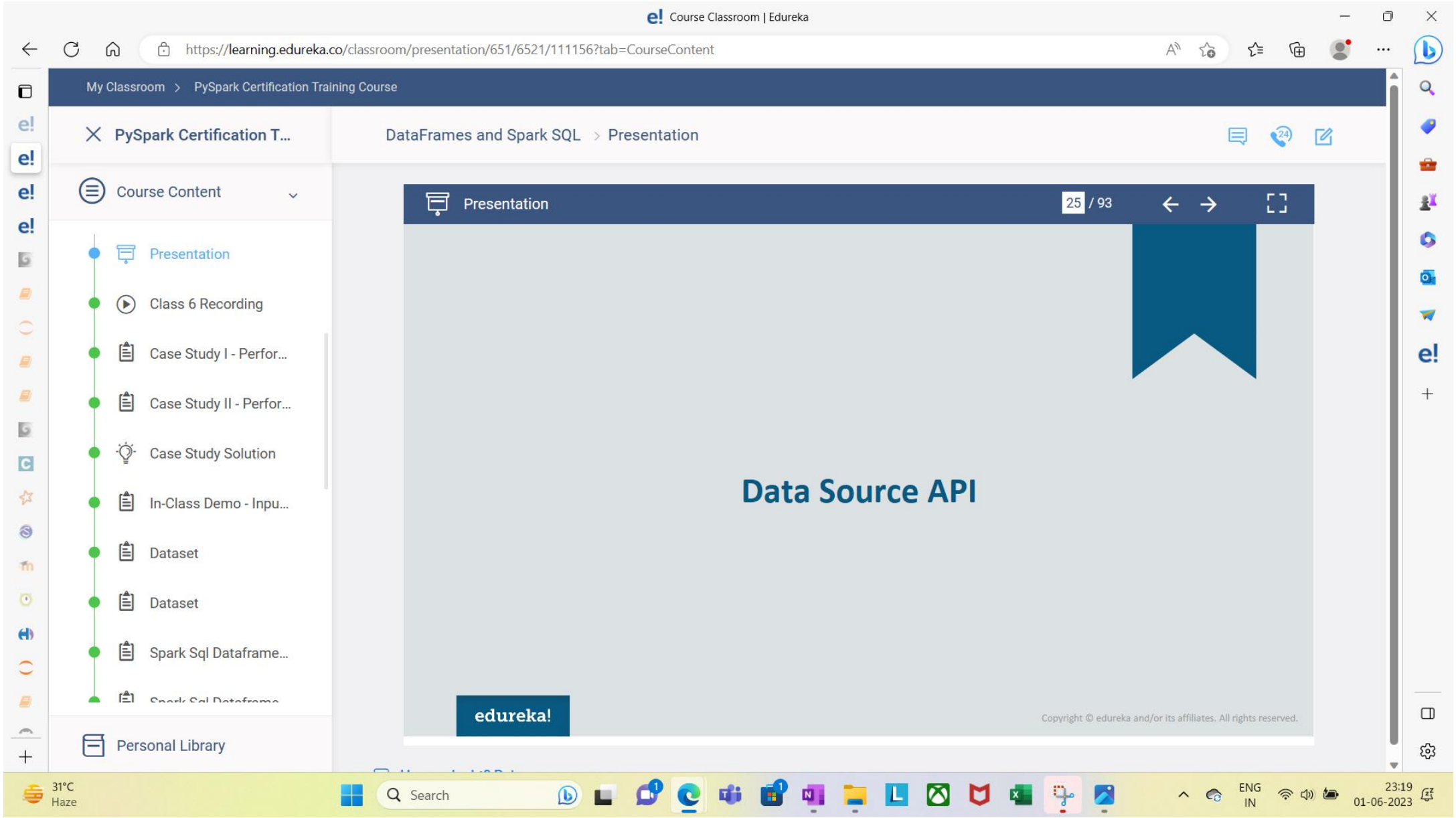
Spark SQL Libraries

Spark SQL has the following libraries:

1. Data Source API
2. DataFrame API
3. Interpreter & Optimizer
4. SQL Service



edureka!


Course Classroom | Edureka
https://learning.edureka.co/classroom/presentation/651/6521/111156?tab=CourseContent
My Classroom > PySpark Certification Training Course
PySpark Certification T...
DataFrames and Spark SQL > Presentation
Course Content
Presentation
Class 6 Recording
Case Study I - Perfor...
Case Study II - Perfor...
Case Study Solution
In-Class Demo - Inpu...
Dataset
Dataset
Spark Sql Dataframe...
Personal Library
Presentation
25 / 93
Data Source API
edureka!
Copyright © edureka and/or its affiliates. All rights reserved.
31°C
Haze
Search
ENG
IN
23:19
01-06-2023

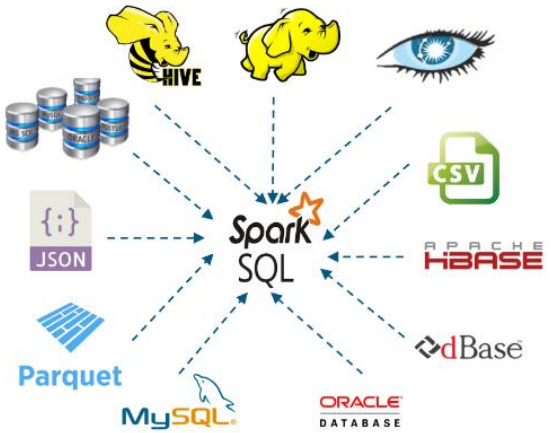
Presentation 26 / 93

Data Source API

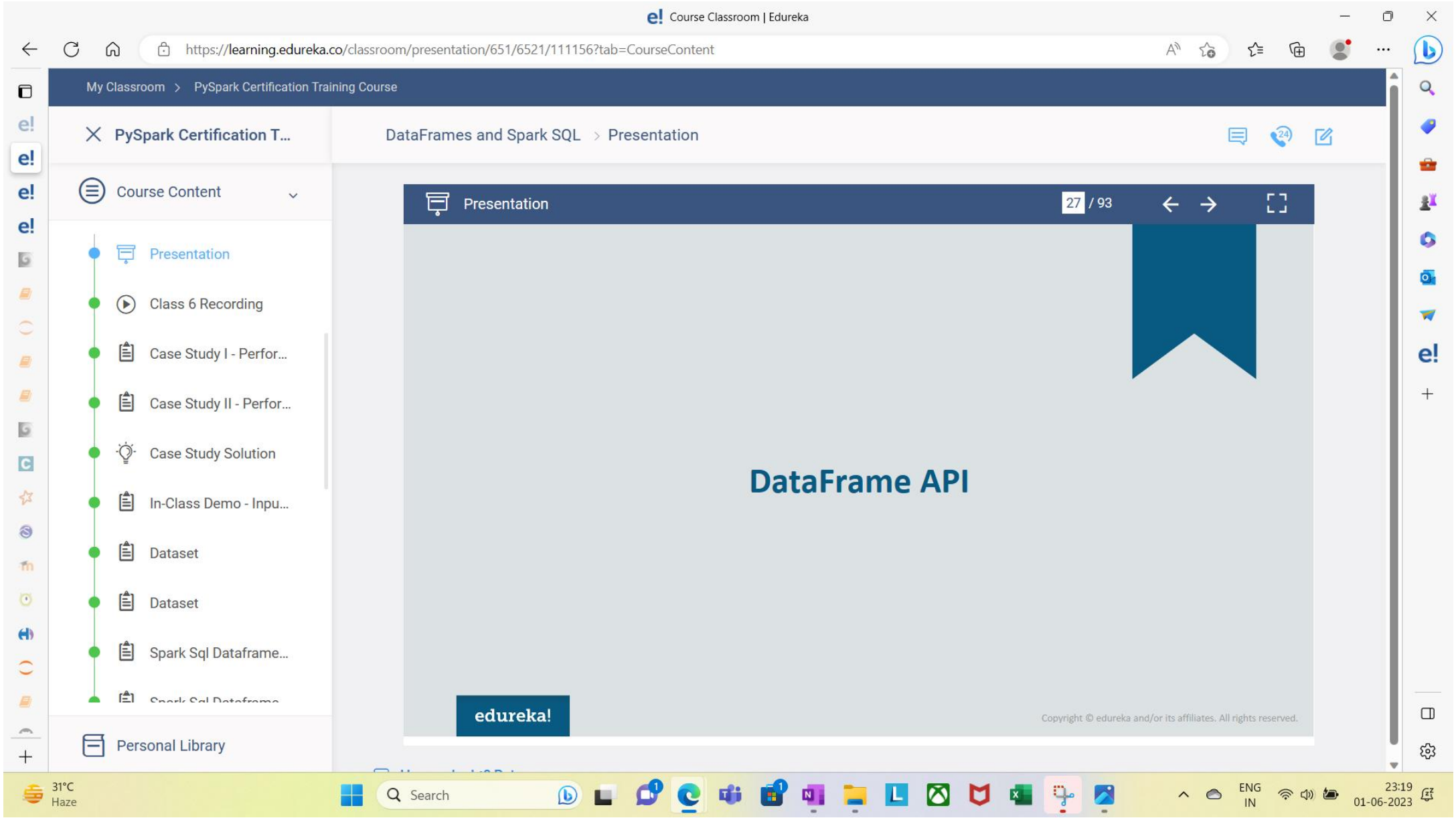
Data Source API is used to *read* and *store structured* and *semi-structured data* into *Spark SQL*

Features:

- Can handle structured/ semi-structured data
- Can load files in multiple formats
- 3rd party integration can be done



edureka!


Course Classroom | Edureka
https://learning.edureka.co/classroom/presentation/651/6521/111156?tab=CourseContent
My Classroom > PySpark Certification Training Course
PySpark Certification T...
DataFrames and Spark SQL > Presentation
Course Content
Presentation
Class 6 Recording
Case Study I - Perfor...
Case Study II - Perfor...
Case Study Solution
In-Class Demo - Inpu...
Dataset
Dataset
Spark Sql Dataframe...
Personal Library
Presentation
27 / 93
DataFrame API
edureka!
Copyright © edureka and/or its affiliates. All rights reserved.
31°C
Haze
Search
ENG
IN
23:19
01-06-2023

Presentation 28 / 93

DataFrame API

DataFrame API converts the data that is read through Data Source API into tabular columns to help perform SQL operations

Data Source API

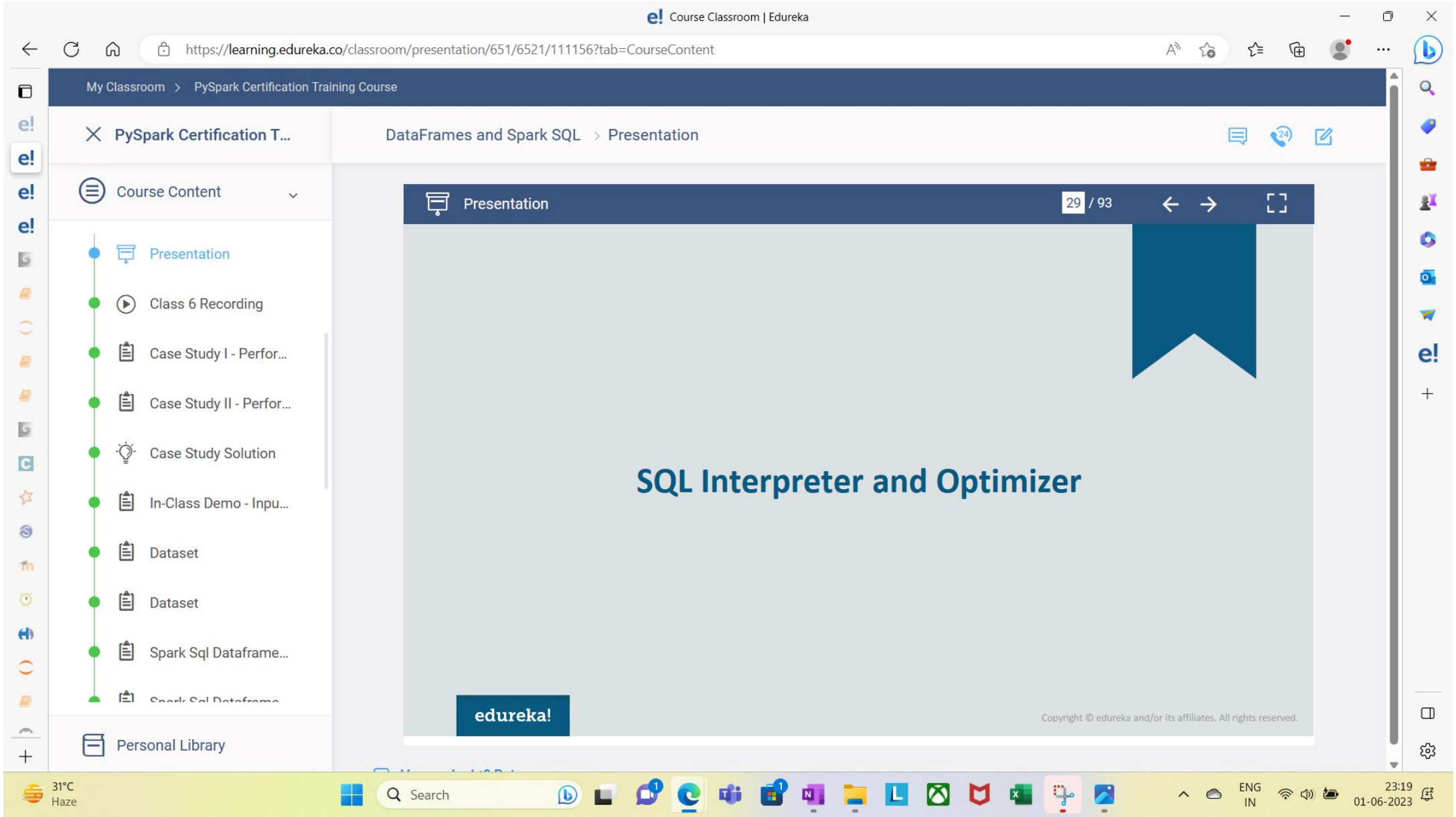
Features:

- Distributed collection of data organized into named columns
- Equivalent to a relational table in SQL
- Lazily evaluated

DataFrame API

Named Columns

edureka!


Course Classroom | Edureka
https://learning.edureka.co/classroom/presentation/651/6521/111156?tab=CourseContent
My Classroom > PySpark Certification Training Course
PySpark Certification T...
DataFrames and Spark SQL > Presentation
Course Content
Presentation
Class 6 Recording
Case Study I - Perfor...
Case Study II - Perfor...
Case Study Solution
In-Class Demo - Inpu...
Dataset
Dataset
Spark Sql Dataframe...
Personal Library
Presentation
29 / 93
SQL Interpreter and Optimizer
edureka!
Copyright © edureka and/or its affiliates. All rights reserved.
31°C
Haze
Search
ENG
IN
23:19
01-06-2023

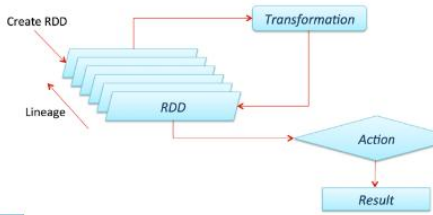
SQL Interpreter and Optimizer

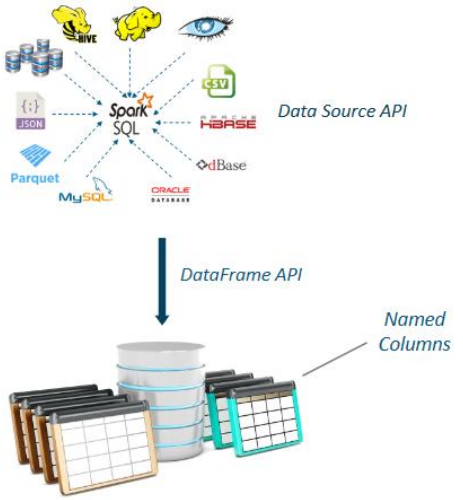
SQL Interpreter & Optimizer handles the *functional programming* part of *Spark SQL*. It transforms the *DataFrames RDDs* to get the *required results* in the *required formats*

Features:

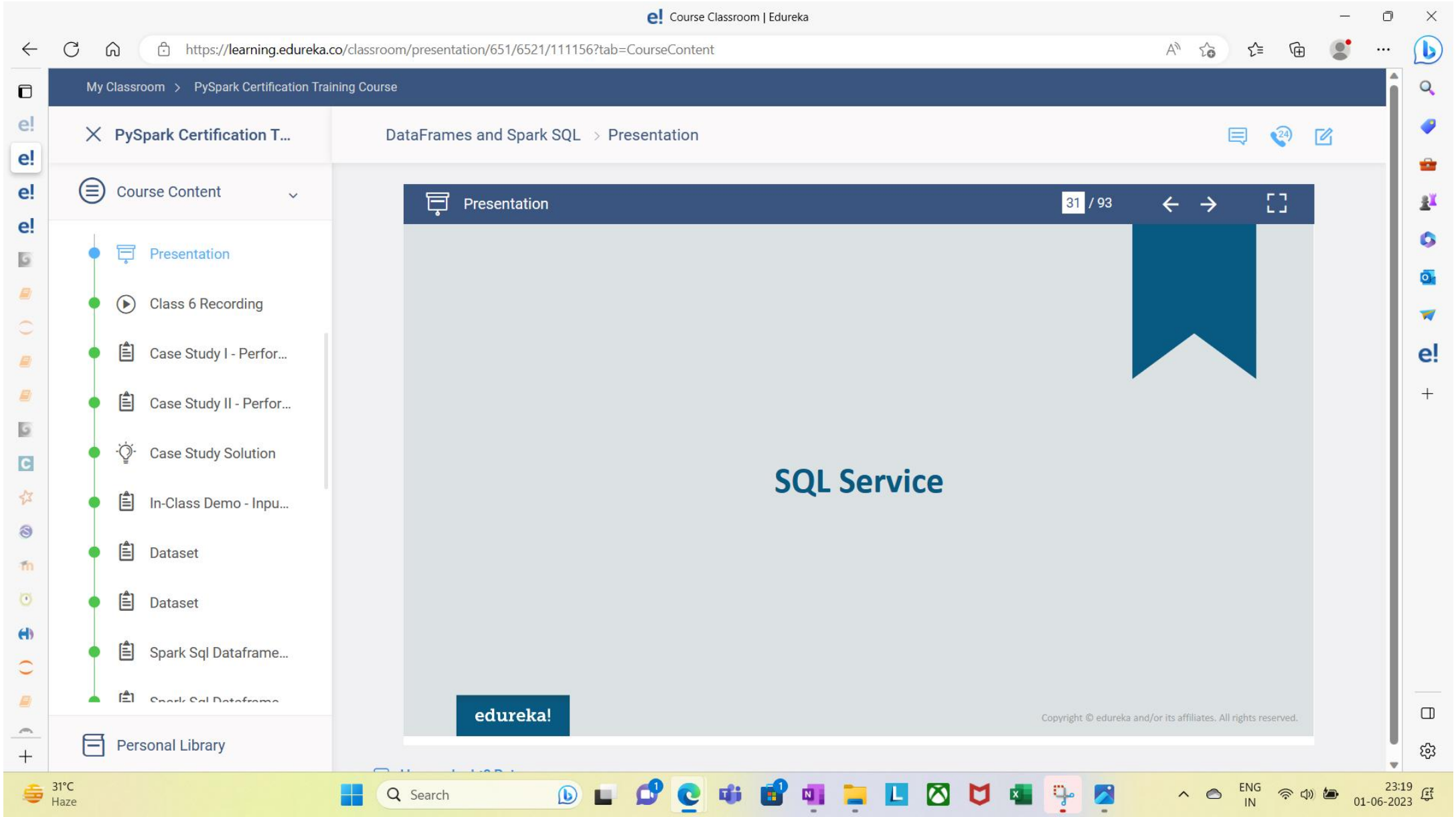
- Functional programming
- Transforming trees
- Faster than RDDs
- Processes all size data

e.g. Catalyst: A modular library for distinct optimization





edureka!


Course Classroom | Edureka
https://learning.edureka.co/classroom/presentation/651/6521/111156?tab=CourseContent
My Classroom > PySpark Certification Training Course
PySpark Certification T...
DataFrames and Spark SQL > Presentation
Course Content
Presentation
Class 6 Recording
Case Study I - Perfor...
Case Study II - Perfor...
Case Study Solution
In-Class Demo - Inpu...
Dataset
Dataset
Spark Sql Dataframe...
Personal Library
Presentation
31 / 93
SQL Service
edureka!
Copyright © edureka and/or its affiliates. All rights reserved.
31°C
Haze
Search
ENG
IN
23:19
01-06-2023

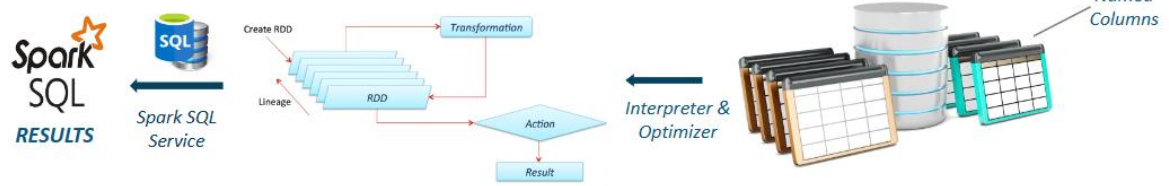
Presentation 32 / 93

SQL Service

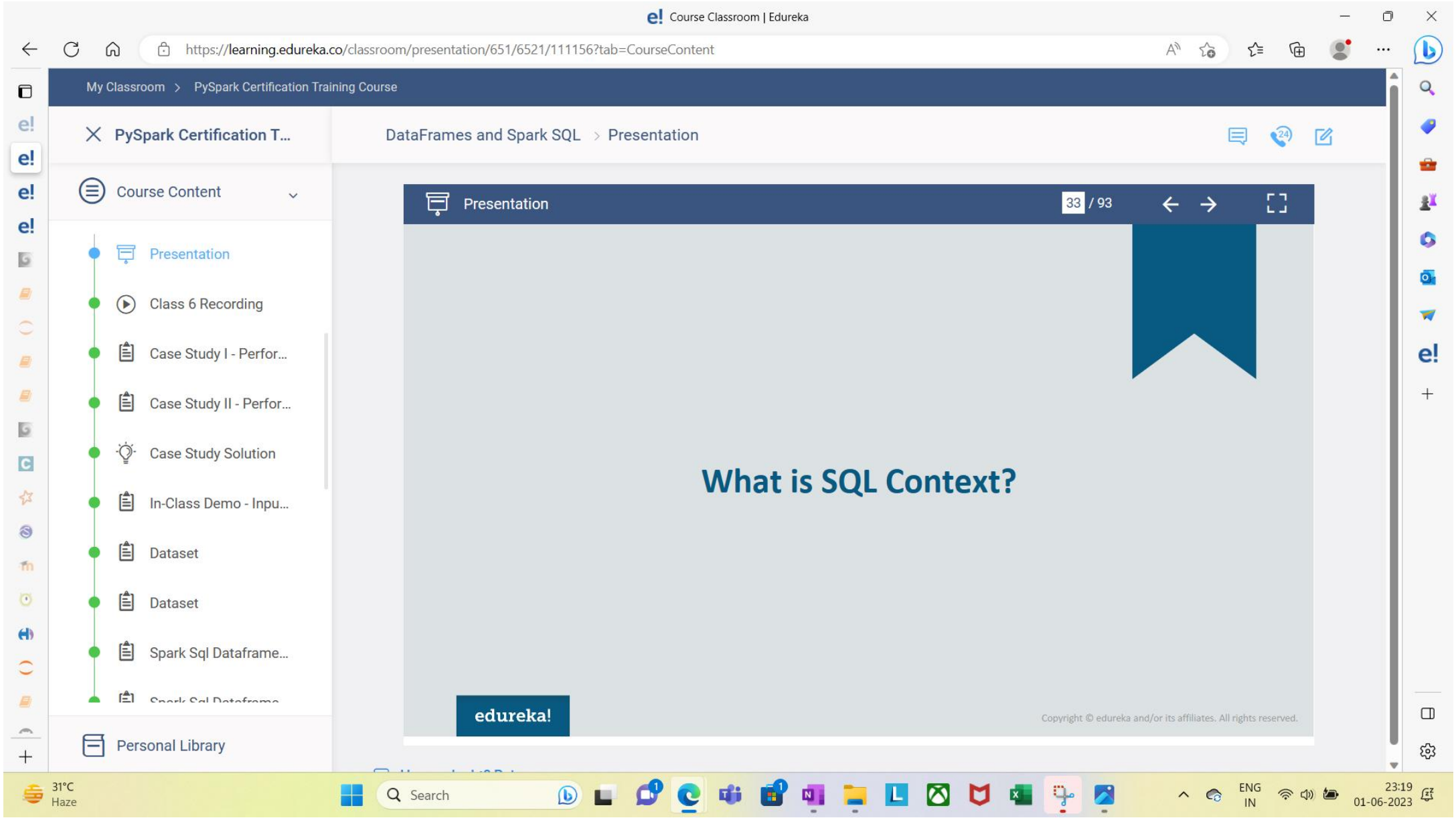
- ***SQL Service*** is the entry point for working along ***structured data*** in Spark
- ***SQL*** is used to fetch the result from the interpreted & optimized data

We have thus used all the four libraries in sequence. This completes a Spark SQL process

DataFrame API



edureka!


Course Classroom | Edureka
https://learning.edureka.co/classroom/presentation/651/6521/111156?tab=CourseContent
My Classroom > PySpark Certification Training Course
PySpark Certification T...
DataFrames and Spark SQL > Presentation
Course Content
Presentation
Class 6 Recording
Case Study I - Perfor...
Case Study II - Perfor...
Case Study Solution
In-Class Demo - Inpu...
Dataset
Dataset
Spark Sql Dataframe...
Personal Library
Presentation
33 / 93
What is SQL Context?
edureka!
Copyright © edureka and/or its affiliates. All rights reserved.
31°C
Haze
Search
ENG
IN
23:19
01-06-2023

Presentation 34 / 93

SQL Context

- ***SQLContext*** is a class and is used for initializing the functionalities of Spark SQL
- SparkContext class object (sc) is required for initializing SQLContext class object

Spark SQL

The following command is used for initializing the SparkContext through Pyspark-shell:

```
$ pyspark2
```

- By default, the SparkContext object is initialized with the name **sc** when the Pyspark-shell starts

Use the following command to create SQLContext:

```
sqlContext=SQLContext(sc)
```

edureka!

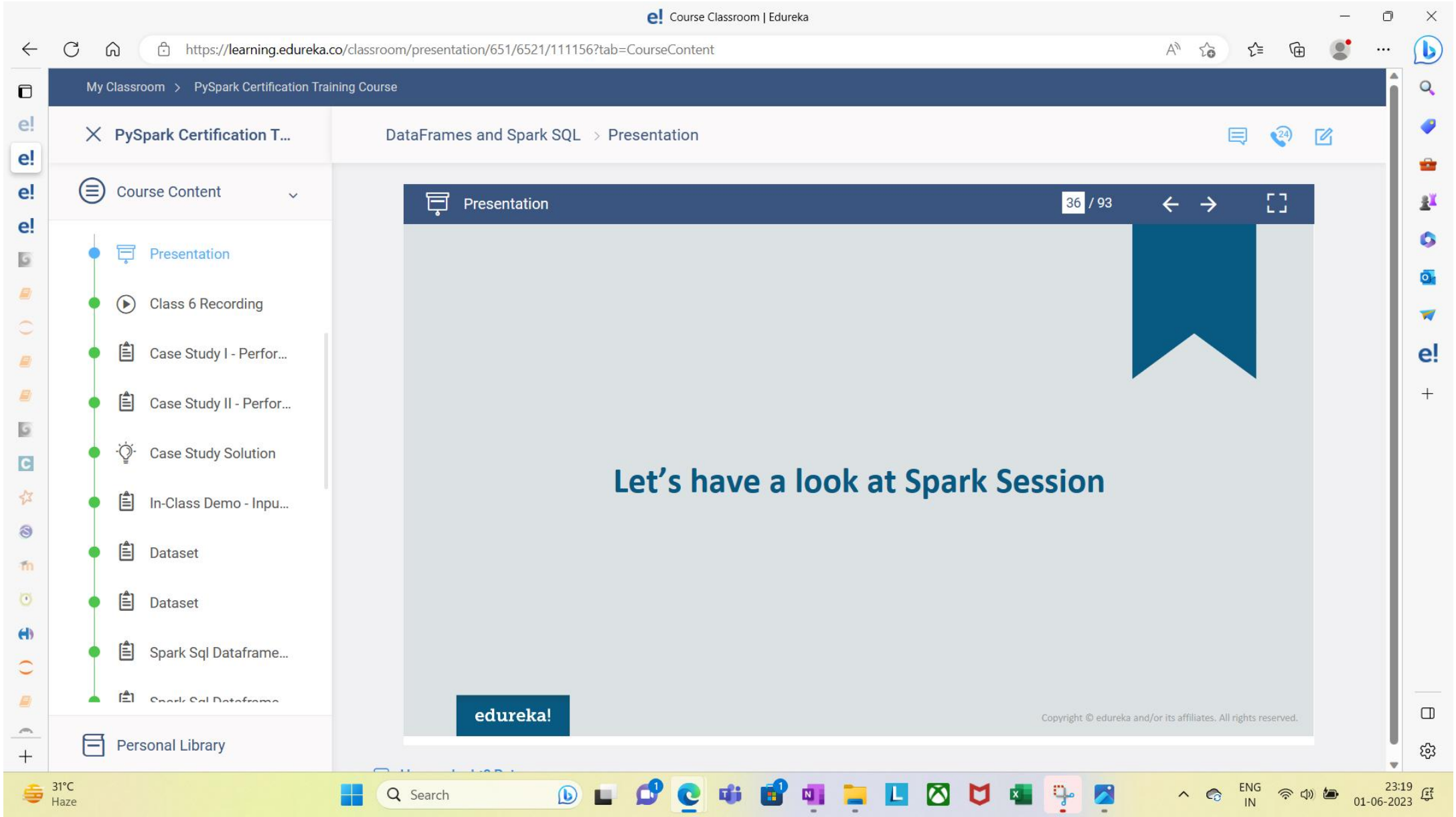
Presentation 35 / 93

SQL Context

```
[edureka_294428@ip-20-0-41-190 ~]$ pyspark2
Python 2.7.5 (default, Aug  4 2017, 00:39:18)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-16)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "ERROR".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.1.0.cloudera2
      /_/

Using Python version 2.7.5 (default, Aug  4 2017 00:39:18)
SparkSession available as 'spark'.
>>> sqlContext=SQLContext(sc)
>>>
```

edureka!


Course Classroom | Edureka
https://learning.edureka.co/classroom/presentation/651/6521/111156?tab=CourseContent
My Classroom > PySpark Certification Training Course
PySpark Certification T...
DataFrames and Spark SQL > Presentation
Course Content
Presentation
Class 6 Recording
Case Study I - Perfor...
Case Study II - Perfor...
Case Study Solution
In-Class Demo - Inpu...
Dataset
Dataset
Spark Sql Dataframe...
Personal Library
Presentation
36 / 93
Let's have a look at Spark Session
edureka!
Copyright © edureka and/or its affiliates. All rights reserved.
31°C
Haze
Search
ENG
IN
23:19
01-06-2023

Presentation 37 / 93

Spark Session - New Entry Point for Spark

- In earlier versions of spark, ***Spark Context*** was entry point for Spark
- All functionality previously available through Spark Context, SQLContext or HiveContext in early versions of Spark are now available via ***Spark Session***

- In Spark 2.x, we have a new entry point for DataSet and Dataframe API's called as ***Spark Session***

So in rest of this module, we will implement different functionalities of Spark SQL using Spark Session

edureka!

Presentation 38 / 93

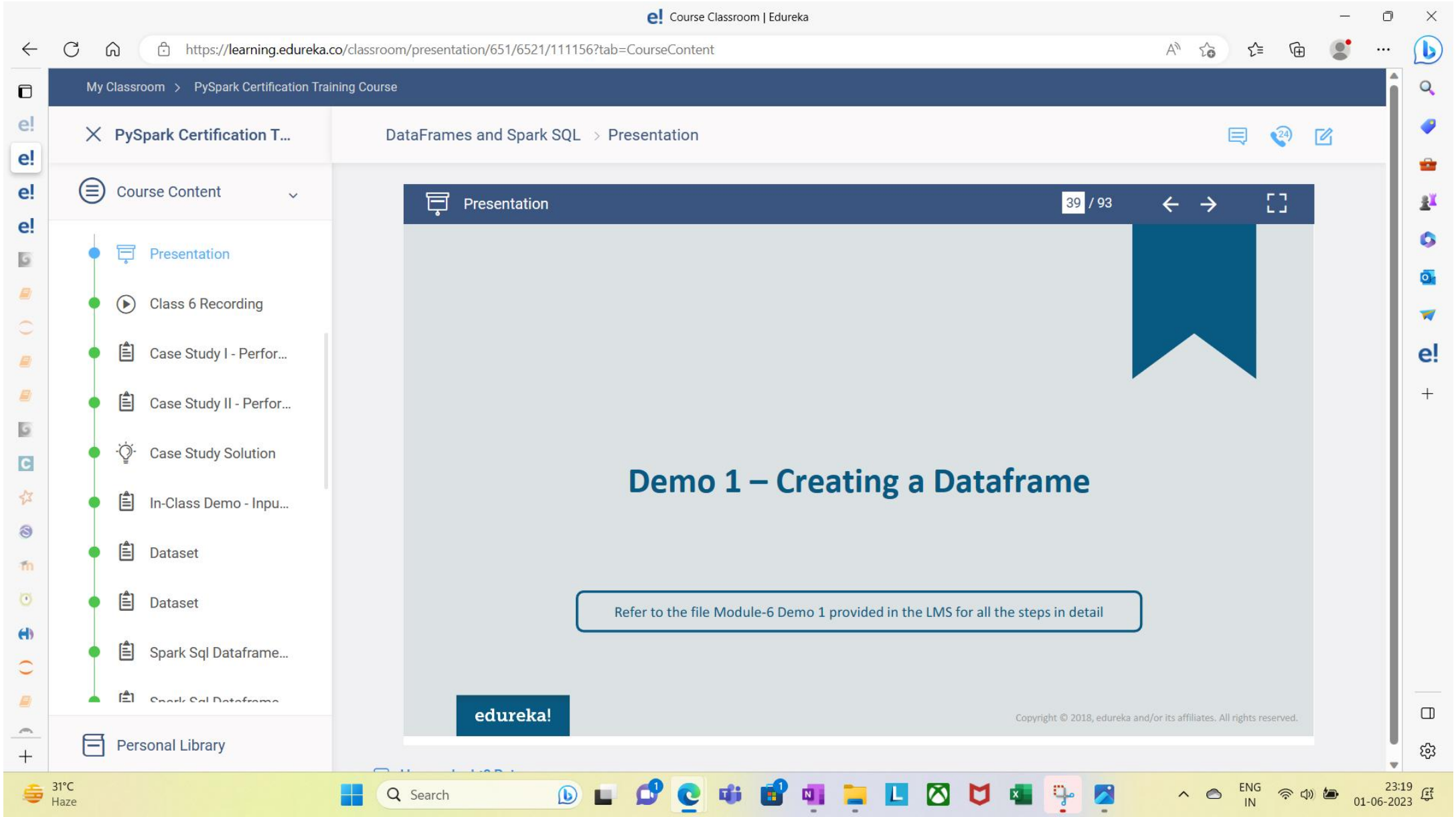
Starting up Spark Session

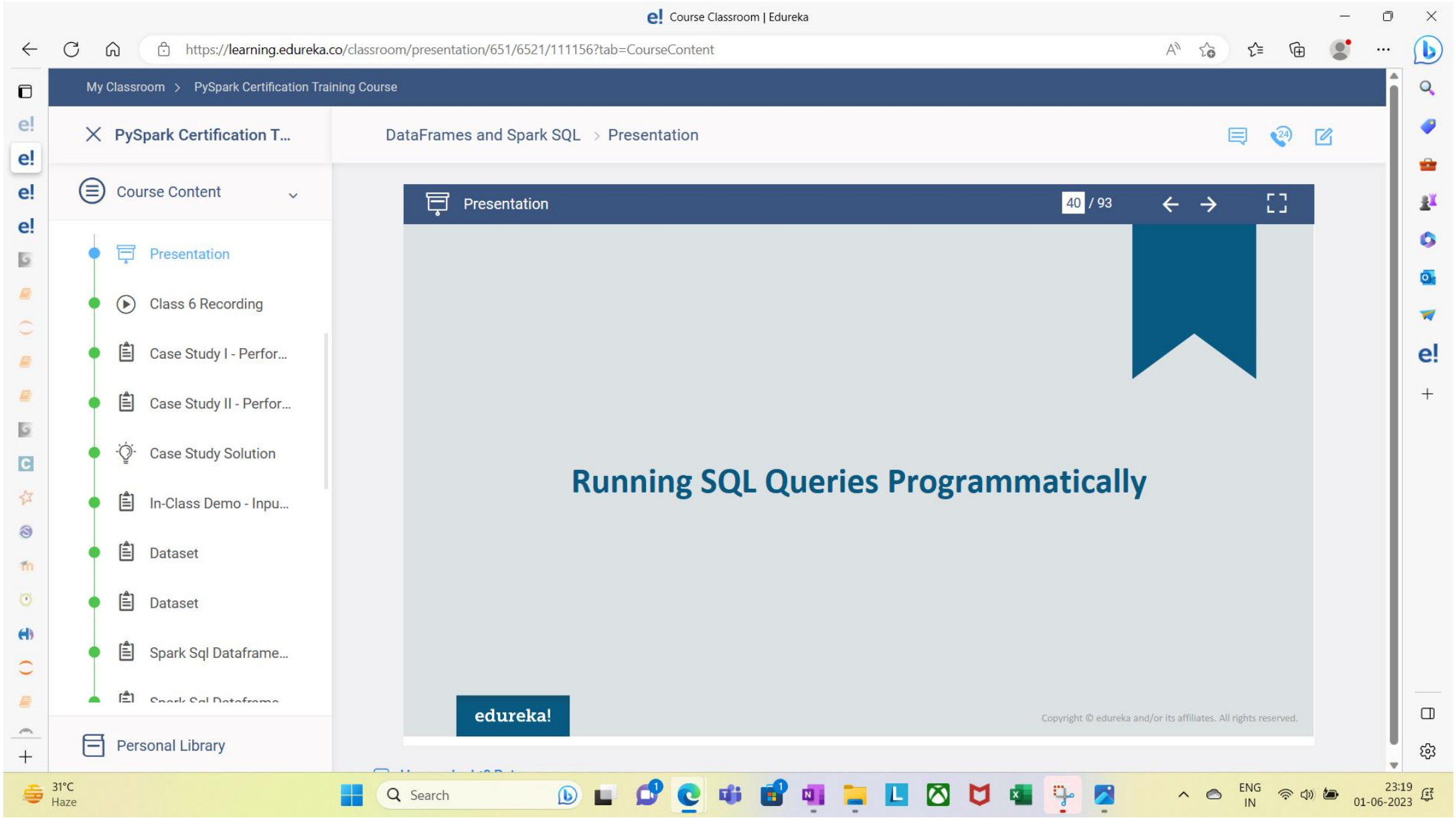
```
[edureka_294428@ip-20-0-41-190 ~]$ pyspark2
Python 2.7.5 (default, Aug  4 2017, 00:39:18)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-16)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "ERROR".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.1.0.cloudera2
      /_/

Using Python version 2.7.5 (default, Aug  4 2017 00:39:18)
SparkSession available as 'spark'.
>>>
```

You can either use "**spark**" or "**sqlContext**" to start working on Spark SQL

edureka!


Course Classroom | Edureka
https://learning.edureka.co/classroom/presentation/651/6521/111156?tab=CourseContent
My Classroom > PySpark Certification Training Course
PySpark Certification T...
DataFrames and Spark SQL > Presentation
Course Content
Presentation
Class 6 Recording
Case Study I - Perfor...
Case Study II - Perfor...
Case Study Solution
In-Class Demo - Inpu...
Dataset
Dataset
Spark Sql Dataframe...
Personal Library
Presentation
39 / 93
Demo 1 – Creating a Dataframe
Refer to the file Module-6 Demo 1 provided in the LMS for all the steps in detail
edureka!
Copyright © 2018, edureka and/or its affiliates. All rights reserved.
31°C
Haze
Search
ENG
IN
23:19
01-06-2023


Course Classroom | Edureka
https://learning.edureka.co/classroom/presentation/651/6521/111156?tab=CourseContent
My Classroom > PySpark Certification Training Course
PySpark Certification T...
DataFrames and Spark SQL > Presentation
Course Content
Presentation
Class 6 Recording
Case Study I - Perfor...
Case Study II - Perfor...
Case Study Solution
In-Class Demo - Inpu...
Dataset
Dataset
Spark Sql Dataframe...
Personal Library
Presentation
40 / 93
Running SQL Queries Programmatically
edureka!
Copyright © edureka and/or its affiliates. All rights reserved.
31°C
Haze
Search
ENG
IN
23:19
01-06-2023

Presentation 41 / 93

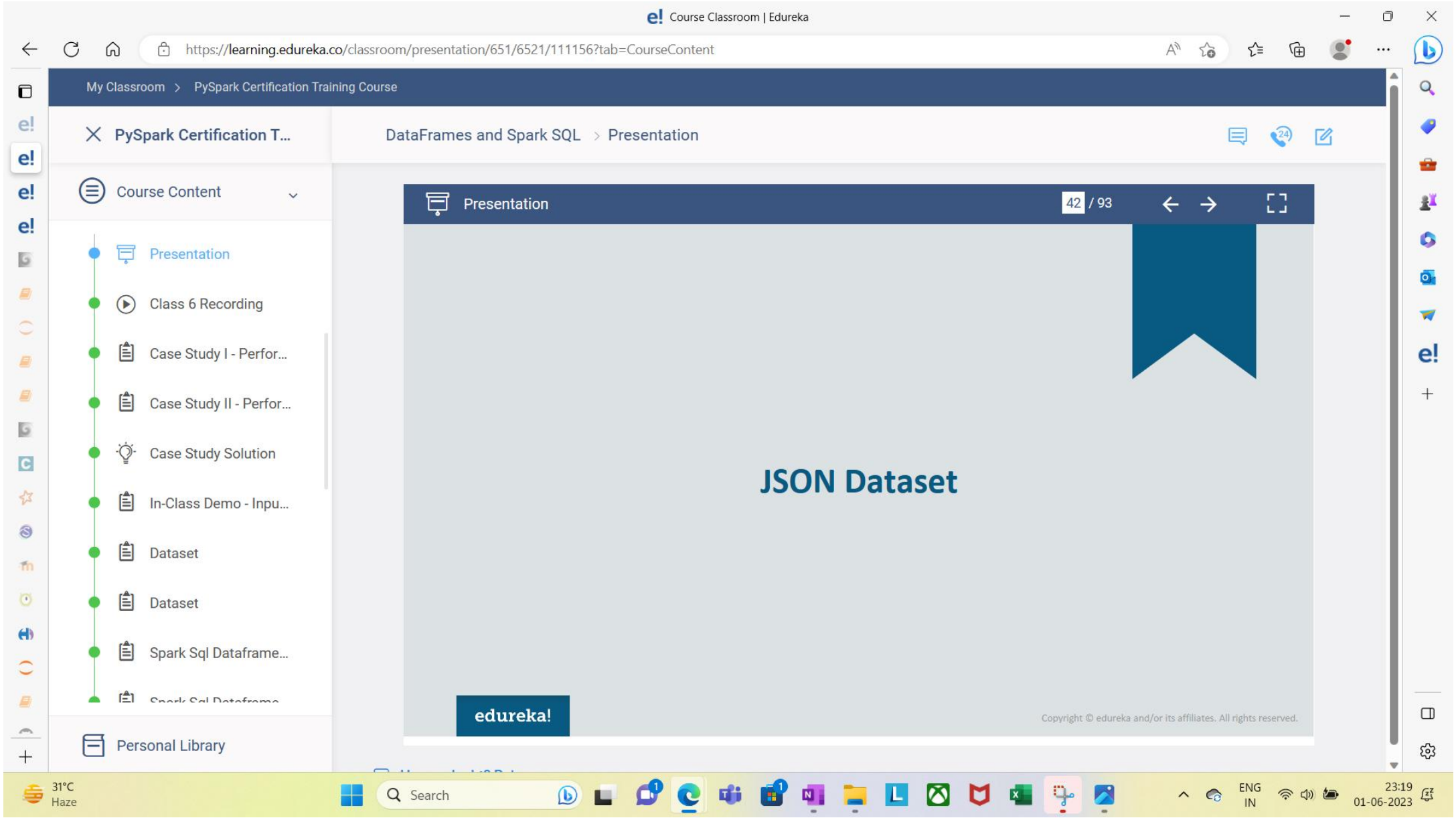
Running SQL Queries Programmatically

The ***sql*** function on a ***SparkSession*** enables applications to run SQL queries programmatically and returns the result as a ***DataFrame***

```
>>> employee_df.createOrReplaceTempView("EMP")
>>> sqlDF = spark.sql("SELECT * FROM EMP")
>>> sqlDF.show()
+---+---------+------+
| ID|     Name|Salary|
+---+---------+------+
|  1|     John| 20000|
|  2|    Rohit| 15000|
|  3|    Parth| 14600|
|  4|  Rishabh| 20500|
|  5|    Daisy| 34000|
|  6|    Annie| 23000|
|  7| Sushmita| 50000|
|  8| Kaivalya| 20000|
|  9|    Varun| 70000|
| 10|Shambhavi| 21500|
| 11|  Johnson| 25500|
| 12|     Riya| 17000|
| 13|    Krish| 17000|
| 14| Akanksha| 20000|
| 15|   Rutuja| 21000|
+---+---------+------+
```

// Register the DataFrame as a SQL temporary view

edureka!


Course Classroom | Edureka
https://learning.edureka.co/classroom/presentation/651/6521/111156?tab=CourseContent
My Classroom > PySpark Certification Training Course
PySpark Certification T...
DataFrames and Spark SQL > Presentation
Course Content
Presentation
Class 6 Recording
Case Study I - Perfor...
Case Study II - Perfor...
Case Study Solution
In-Class Demo - Inpu...
Dataset
Dataset
Spark Sql Dataframe...
Personal Library
Presentation
42 / 93
JSON Dataset
edureka!
Copyright © edureka and/or its affiliates. All rights reserved.
31°C
Haze
Search
ENG
IN
23:19
01-06-2023

Presentation 43 / 93

JSON Data - Loading File

Spark SQL can automatically infer the schema of a JSON dataset and load it as a Dataset[Row]. This conversion can be done using **SparkSession.read.json()** on a JSON file

```
>>> employee_df=spark.read.json("hdfs://nameservice1/user/edureka_294428/Employees.json")
```

edureka!

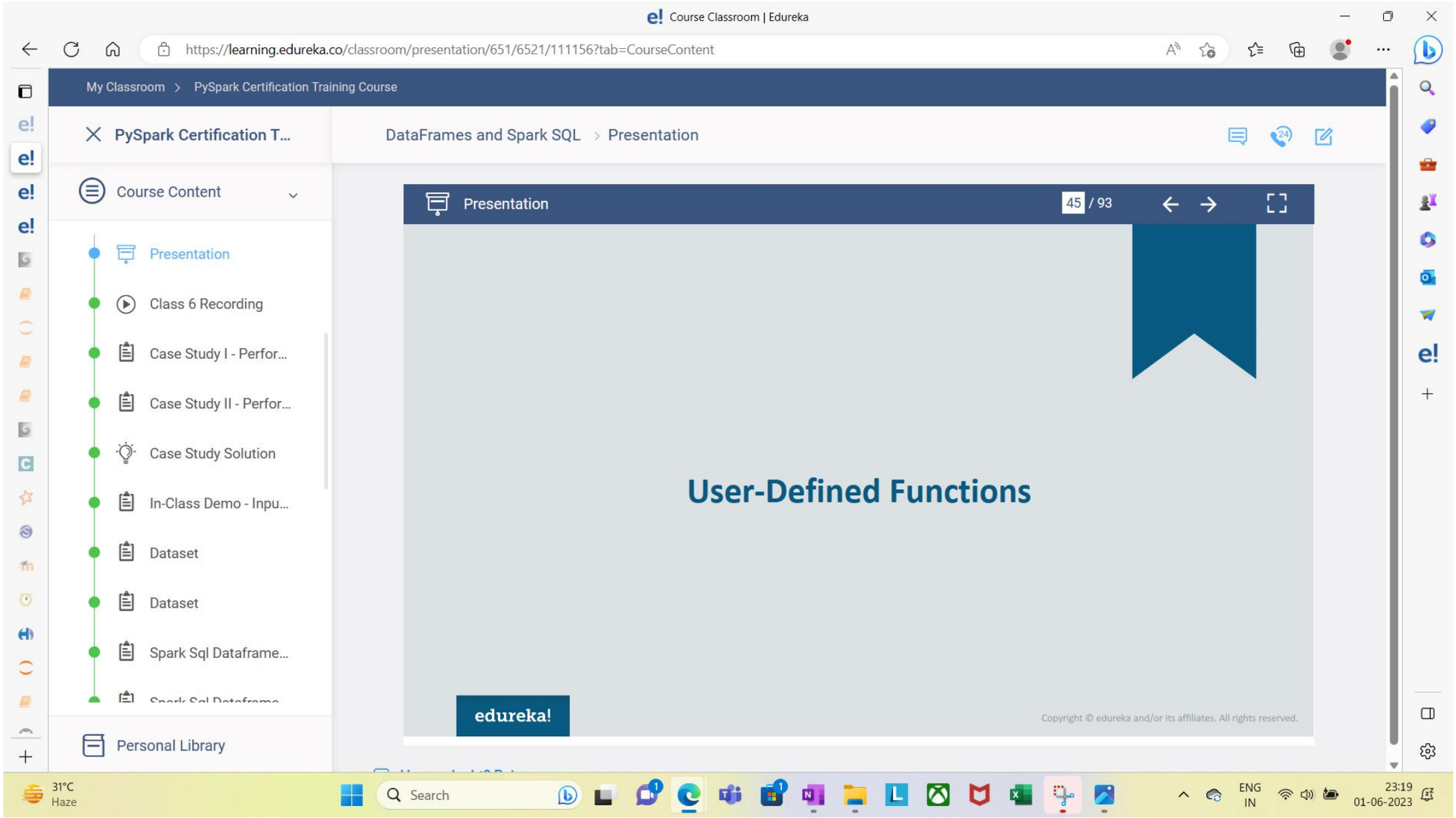
Presentation 44 / 93

JSON Data - Parquet File

Parquet is a *columnar format* and Spark SQL *provides support* for both *reading* and *writing Parquet files* that automatically *preserves the schema* of the original data

```
>>> peopleDF=spark.read.json("hdfs://nameservice1/user/edureka_294428/people.json")
>>> peopleDF.write.parquet("hdfs://nameservice1/user/edureka_294428/people8.parquet")
>>> parquetFile=spark.read.parquet("people8.parquet")
>>> parquetFile.createOrReplaceTempView('parquetFile')
>>> teenagers=spark.sql("SELECT name FROM parquetFile WHERE age>=13 AND age<=19")
>>> teenagers.show()
+------+
|  name|
+------+
|justin|
+------+
```

edureka!

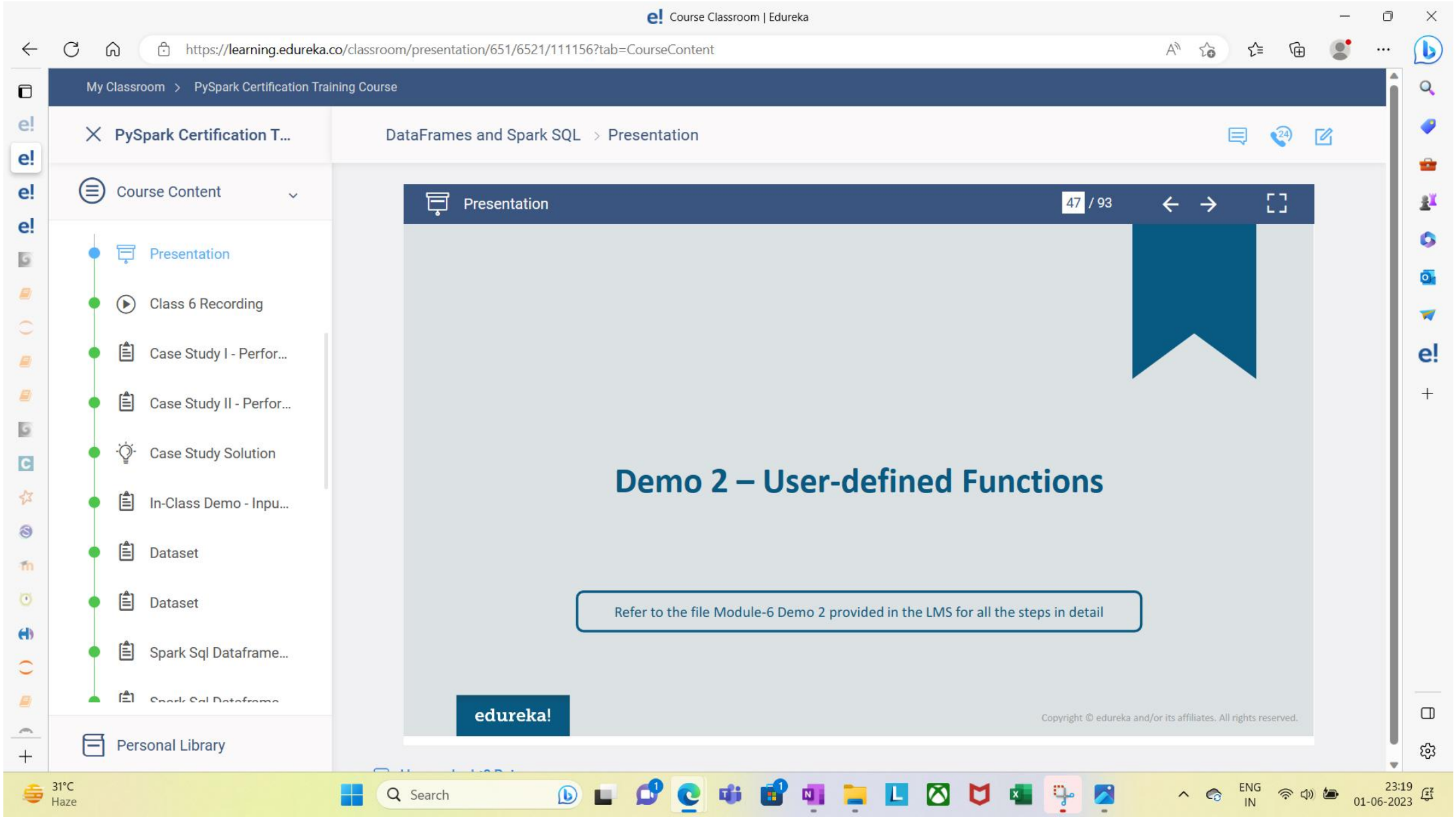

e! Course Classroom | Edureka
https://learning.edureka.co/classroom/presentation/651/6521/111156?tab=CourseContent
My Classroom > PySpark Certification Training Course
PySpark Certification T...
DataFrames and Spark SQL > Presentation
Course Content
Presentation
Class 6 Recording
Case Study I - Perfor...
Case Study II - Perfor...
Case Study Solution
In-Class Demo - Inpu...
Dataset
Dataset
Spark Sql Dataframe...
Personal Library
Presentation
45 / 93
User-Defined Functions
edureka!
Copyright © edureka and/or its affiliates. All rights reserved.
31°C
Haze
Search
ENG
IN
23:19
01-06-2023

Presentation 46 / 93

User-Defined Functions

- Spark allows to define custom SQL functions, called **User-Defined Functions (UDFs)**
- A very popular way to expose advanced functionality to SQL users in an organization, so that these users can call into it without writing code
- UDF can work with various data types and can return different type than one we are calling the function

edureka!

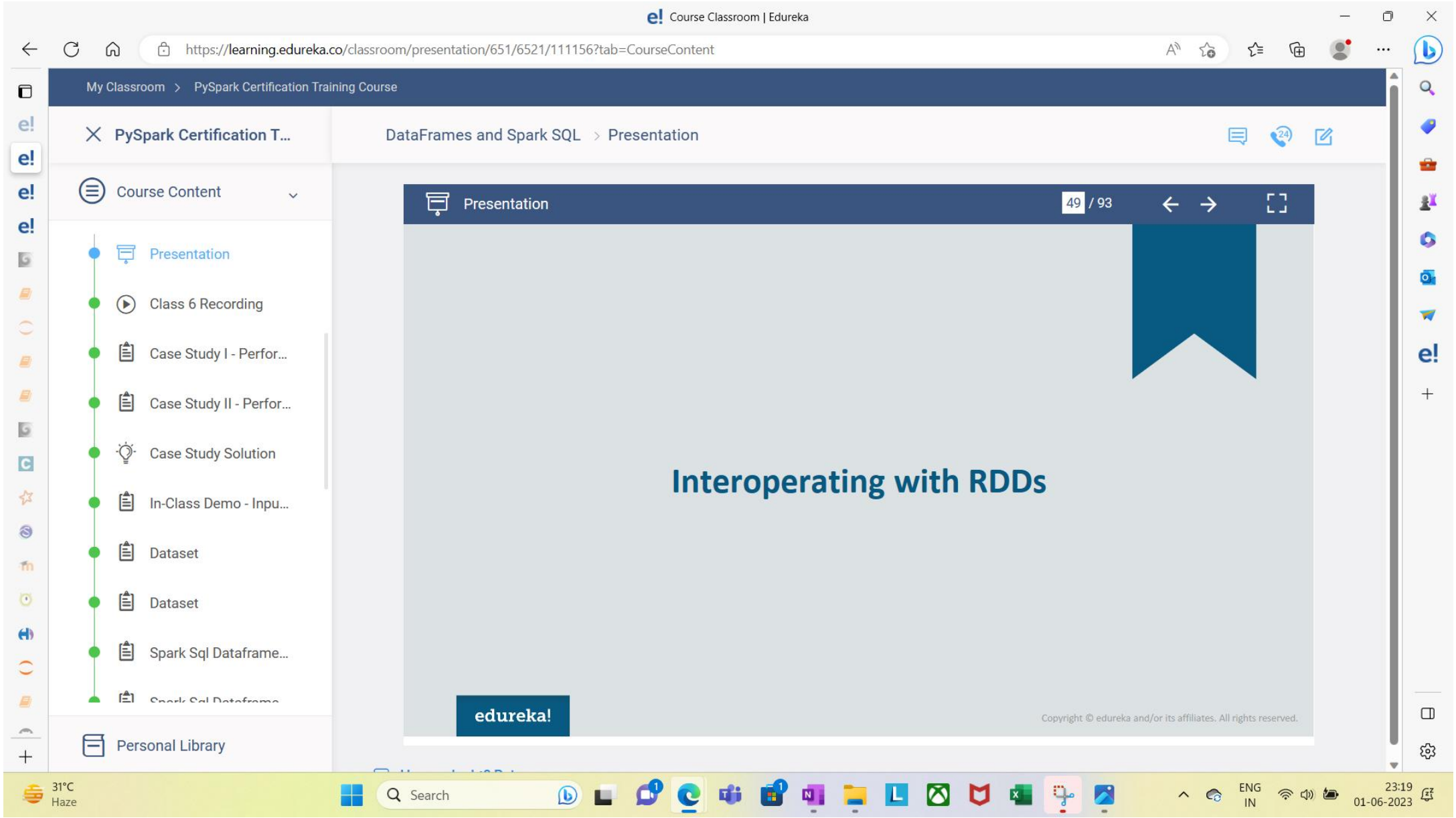

e! Course Classroom | Edureka
https://learning.edureka.co/classroom/presentation/651/6521/111156?tab=CourseContent
My Classroom > PySpark Certification Training Course
PySpark Certification T...
DataFrames and Spark SQL > Presentation
Course Content
Presentation
Class 6 Recording
Case Study I - Perfor...
Case Study II - Perfor...
Case Study Solution
In-Class Demo - Inpu...
Dataset
Dataset
Spark Sql Dataframe...
Personal Library
Presentation
47 / 93
Demo 2 – User-defined Functions
Refer to the file Module-6 Demo 2 provided in the LMS for all the steps in detail
edureka!
Copyright © edureka and/or its affiliates. All rights reserved.
31°C
Haze
Search
ENG
IN
23:19
01-06-2023

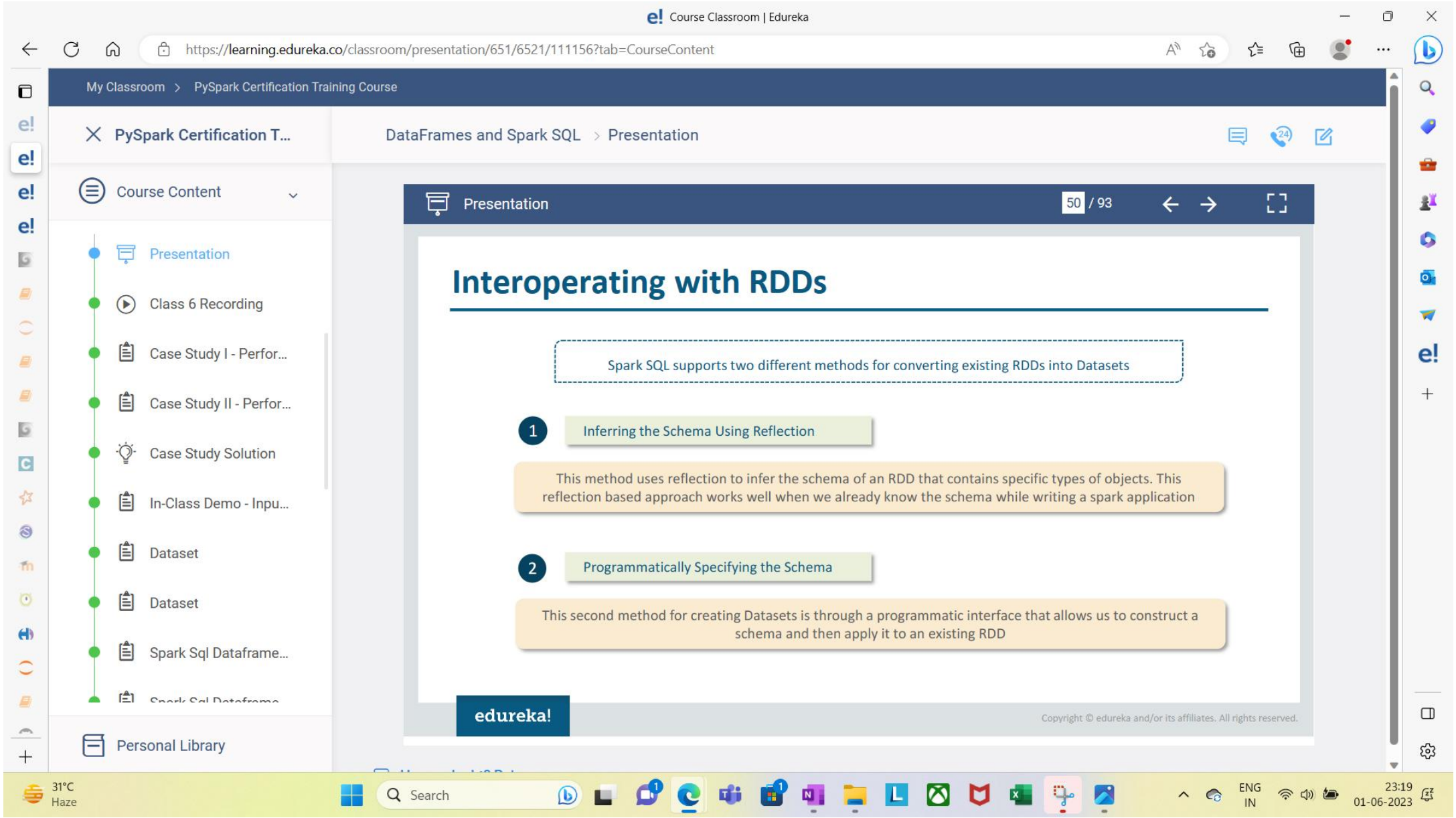
Presentation 48 / 93

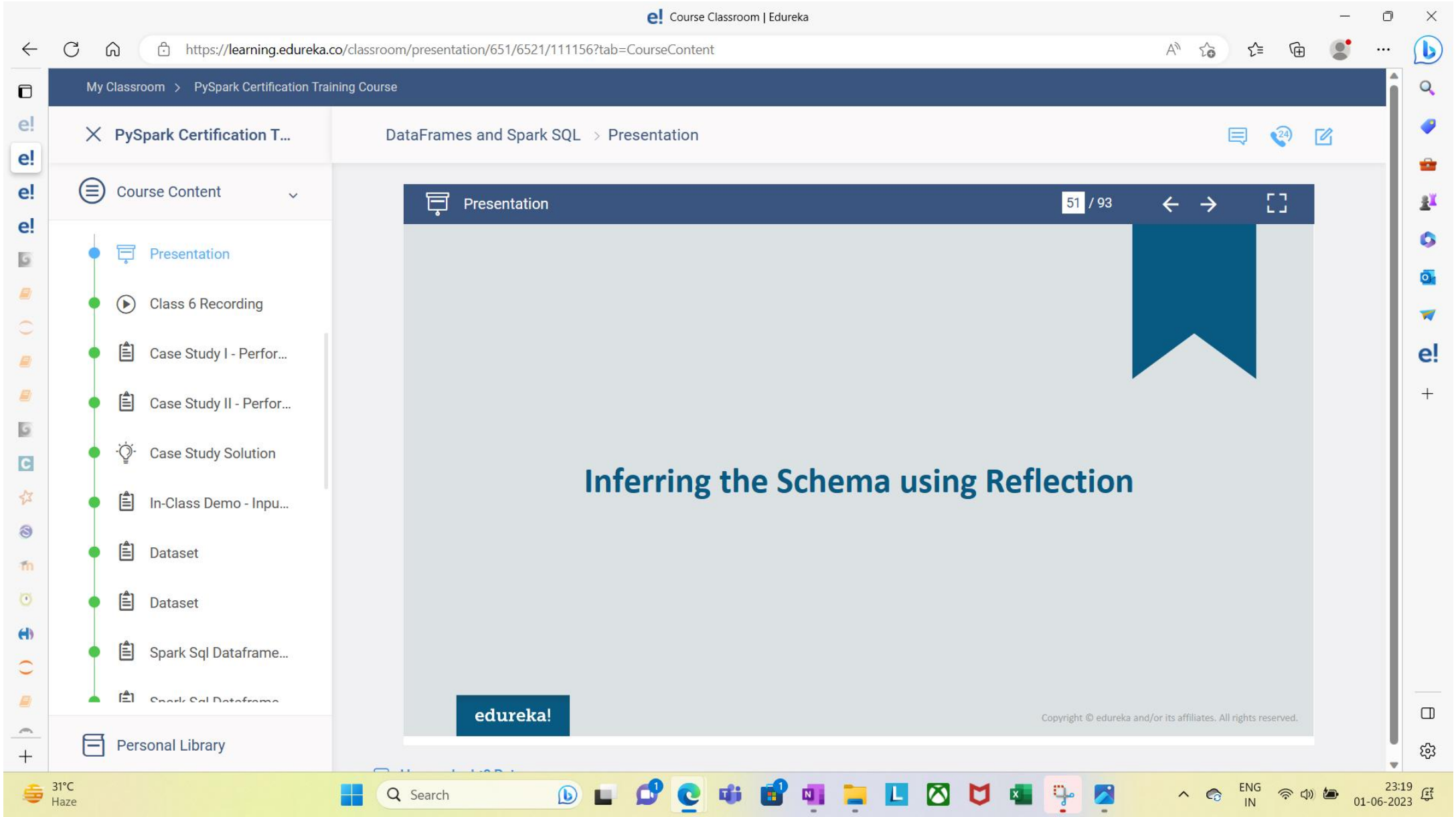
UDFs are Blackbox

UDFs are great when built-in SQL functions aren't sufficient, but should be used sparingly because they're not performant

edureka!


Course Classroom | Edureka
https://learning.edureka.co/classroom/presentation/651/6521/111156?tab=CourseContent
My Classroom > PySpark Certification Training Course
PySpark Certification T...
DataFrames and Spark SQL > Presentation
Course Content
Presentation
Class 6 Recording
Case Study I - Perfor...
Case Study II - Perfor...
Case Study Solution
In-Class Demo - Inpu...
Dataset
Dataset
Spark Sql Dataframe...
Personal Library
Presentation
49 / 93
Interoperating with RDDs
edureka!
Copyright © edureka and/or its affiliates. All rights reserved.
31°C
Haze
Search
ENG
IN
23:19
01-06-2023


Course Classroom | Edureka
https://learning.edureka.co/classroom/presentation/651/6521/111156?tab=CourseContent
My Classroom > PySpark Certification Training Course
PySpark Certification T...
DataFrames and Spark SQL > Presentation
Course Content
Presentation
Class 6 Recording
Case Study I - Perfor...
Case Study II - Perfor...
Case Study Solution
In-Class Demo - Inpu...
Dataset
Dataset
Spark Sql Dataframe...
Personal Library
Presentation
50 / 93
Interoperating with RDDs
Spark SQL supports two different methods for converting existing RDDs into Datasets
1
Inferring the Schema Using Reflection
This method uses reflection to infer the schema of an RDD that contains specific types of objects. This reflection based approach works well when we already know the schema while writing a spark application
2
Programmatically Specifying the Schema
This second method for creating Datasets is through a programmatic interface that allows us to construct a schema and then apply it to an existing RDD
edureka!
Copyright © edureka and/or its affiliates. All rights reserved.
31°C
Haze
Search
ENG
IN
23:19
01-06-2023


Course Classroom | Edureka
https://learning.edureka.co/classroom/presentation/651/6521/111156?tab=CourseContent
My Classroom > PySpark Certification Training Course
PySpark Certification T...
DataFrames and Spark SQL > Presentation
Course Content
Presentation
Class 6 Recording
Case Study I - Perfor...
Case Study II - Perfor...
Case Study Solution
In-Class Demo - Inpu...
Dataset
Dataset
Spark Sql Dataframe...
Personal Library
Presentation
51 / 93
Inferring the Schema using Reflection
edureka!
Copyright © edureka and/or its affiliates. All rights reserved.
31°C
Haze
Search
ENG
IN
23:19
01-06-2023

Presentation 52 / 93

Inferring Schema

- Data exists in multiple formats including TXT, CSV, DOC, PDF etc.
- At times you might be aware of the schema (order of data) of the data.
- When the schema is already known we use the concept of Reflection to directly infer the schema

edureka!

Presentation 53 / 93

Demo: Inferring Schema

```
John, 20
Rohit, 15
Parth, 14
Rishabh, 25
Daisy, 34
Annie, 23
Sushmita, 50
Kaivalya, 26
Varun, 16
Shambhavi, 21
Johnson, 22
Riya, 17
Krish, 19
Akanksha, 28
Rutuja, 33
```

Consider we have a text file of the following format. Please note that here we are considering that schema of the file is already known and is stored in a TXT format (Non-Columnar)

edureka!

Presentation 54 / 93

Demo: Inferring Schema

```
[edureka_321047@ip-20-0-41-202 ~]$ pyspark2
Python 2.7.5 (default, Aug  4 2017, 00:39:18)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-16)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "ERROR".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.1.0.cloudera2
      /_/

Using Python version 2.7.5 (default, Aug  4 2017 00:39:18)
SparkSession available as 'spark'.
>>> from pyspark.sql import Row
>>> sc = spark.sparkContext
>>> lines = sc.textFile("/user/edureka_321047/sampleText")
>>> parts = lines.map(lambda l: l.split(","))
>>> people = parts.map(lambda p: Row(name=p[0], age=int(p[1])))
>>> schemaPeople = spark.createDataFrame(people)
>>> schemaPeople.createOrReplaceTempView("people")
>>> teenagers = spark.sql("SELECT name FROM people WHERE age >= 13 AND age <= 19")
>>> teenNames = teenagers.rdd.map(lambda p: "Name: " + p.name).collect()
```

Importing the PySQL Row module. This module is needed to create Row objects using named arguments

edureka!

Demo: Inferring Schema

```
[edureka_321047@ip-20-0-41-202 ~]$ pyspark2
Python 2.7.5 (default, Aug  4 2017, 00:39:18)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-16)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "ERROR".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.1.0.cloudera2
      /_/

Using Python version 2.7.5 (default, Aug  4 2017 00:39:18)
SparkSession available as 'spark'.
>>> from pyspark.sql import Row
>>> sc = spark.sparkContext
>>> lines = sc.textFile("/user/edureka_321047/sampleText")
>>> parts = lines.map(lambda l: l.split(","))
>>> people = parts.map(lambda p: Row(name=p[0], age=int(p[1])))
>>> schemaPeople = spark.createDataFrame(people)
>>> schemaPeople.createOrReplaceTempView("people")
>>> teenagers = spark.sql("SELECT name FROM people WHERE age >= 13 AND age <= 19")
>>> teenNames = teenagers.rdd.map(lambda p: "Name: " + p.name).collect()
```

SparkContext is a constructor. The given command creates a SparkContext that loads settings from system properties (for instance, when launching with /spark2-submit)

edureka!

Presentation 56 / 93

Demo: Inferring Schema

```
[edureka_321047@ip-20-0-41-202 ~]$ pyspark2
Python 2.7.5 (default, Aug  4 2017, 00:39:18)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-16)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "ERROR".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.1.0.cloudera2
      /_/

Using Python version 2.7.5 (default, Aug  4 2017 00:39:18)
SparkSession available as 'spark'.
>>> from pyspark.sql import Row
>>> sc = spark.sparkContext
>>> lines = sc.textFile("/user/edureka_321047/sampleText")
>>> parts = lines.map(lambda l: l.split(","))
>>> people = parts.map(lambda p: Row(name=p[0], age=int(p[1])))
>>> schemaPeople = spark.createDataFrame(people)
>>> schemaPeople.createOrReplaceTempView("people")
>>> teenagers = spark.sql("SELECT name FROM people WHERE age >= 13 AND age <= 19")
>>> teenNames = teenagers.rdd.map(lambda p: "Name: " + p.name).collect()
```

We will now read our sample text file and store it in our lines RDD

edureka!

Presentation 57 / 93

Demo: Inferring Schema

```
[edureka_321047@ip-20-0-41-202 ~]$ pyspark2
Python 2.7.5 (default, Aug  4 2017, 00:39:18)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-16)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "ERROR".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.1.0.cloudera2
      /_/

Using Python version 2.7.5 (default, Aug  4 2017 00:39:18)
SparkSession available as 'spark'.
>>> from pyspark.sql import Row
>>> sc = spark.sparkContext
>>> lines = sc.textFile("/user/edureka_321047/sampleText")
>>> parts = lines.map(lambda l: l.split(","))
>>> people = parts.map(lambda p: Row(name=p[0], age=int(p[1])))
>>> schemaPeople = spark.createDataFrame(people)
>>> schemaPeople.createOrReplaceTempView("people")
>>> teenagers = spark.sql("SELECT name FROM people WHERE age >= 13 AND age <= 19")
>>> teenNames = teenagers.rdd.map(lambda p: "Name: " + p.name).collect()
```

We will now split each line based on “,” (delimiter). Each of these lines will be considered as a separate row inside the table

edureka!

Presentation 58 / 93

Demo: Inferring Schema

```
[edureka_321047@ip-20-0-41-202 ~]$ pyspark2
Python 2.7.5 (default, Aug  4 2017, 00:39:18)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-16)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "ERROR".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.1.0.cloudera2
      /_/

Using Python version 2.7.5 (default, Aug  4 2017 00:39:18)
SparkSession available as 'spark'.
>>> from pyspark.sql import Row
>>> sc = spark.sparkContext
>>> lines = sc.textFile("/user/edureka_321047/sampleText")
>>> parts = lines.map(lambda l: l.split(","))
>>> people = parts.map(lambda p: Row(name=p[0], age=int(p[1])))
>>> schemaPeople = spark.createDataFrame(people)
>>> schemaPeople.createOrReplaceTempView("people")
>>> teenagers = spark.sql("SELECT name FROM people WHERE age >= 13 AND age <= 19")
>>> teenNames = teenagers.rdd.map(lambda p: "Name: " + p.name).collect()
```

Each of the lines will now be mapped into a table with specific column names. Here we will consider the first column to be Name and second to be Age

edureka!

Presentation 59 / 93

Demo: Inferring Schema

```
[edureka_321047@ip-20-0-41-202 ~]$ pyspark2
Python 2.7.5 (default, Aug  4 2017, 00:39:18)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-16)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "ERROR".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.1.0.cloudera2
      /_/

Using Python version 2.7.5 (default, Aug  4 2017 00:39:18)
SparkSession available as 'spark'.
>>> from pyspark.sql import Row
>>> sc = spark.sparkContext
>>> lines = sc.textFile("/user/edureka_321047/sampleText")
>>> parts = lines.map(lambda l: l.split(","))
>>> people = parts.map(lambda p: Row(name=p[0], age=int(p[1])))
>>> schemaPeople = spark.createDataFrame(people)
>>> schemaPeople.createOrReplaceTempView("people")
>>> teenagers = spark.sql("SELECT name FROM people WHERE age >= 13 AND age <= 19")
>>> teenNames = teenagers.rdd.map(lambda p: "Name: " + p.name).collect()
```

- Creates a DataFrame from an RDD of tuple/list, list or pandas.DataFrame
- When schema is a list of column names, the type of each column will be inferred from data
- When schema is None, it will try to infer the schema (column names and types) from data, which should be an RDD of Row, or namedtuple, or dict

edureka!

Presentation 60 / 93

Demo: Inferring Schema

```
[edureka_321047@ip-20-0-41-202 ~]$ pyspark2
Python 2.7.5 (default, Aug  4 2017, 00:39:18)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-16)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "ERROR".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.1.0.cloudera2
      /_/

Using Python version 2.7.5 (default, Aug  4 2017 00:39:18)
SparkSession available as 'spark'.
>>> from pyspark.sql import Row
>>> sc = spark.sparkContext
>>> lines = sc.textFile("/user/edureka_321047/sampleText")
>>> parts = lines.map(lambda l: l.split(","))
>>> people = parts.map(lambda p: Row(name=p[0], age=int(p[1])))
>>> schemaPeople = spark.createDataFrame(people)
>>> schemaPeople.createOrReplaceTempView("people")
>>> teenagers = spark.sql("SELECT name FROM people WHERE age >= 13 AND age <= 19")
>>> teenNames = teenagers.rdd.map(lambda p: "Name: " + p.name).collect()
```

createOrReplaceTempView creates (or replaces if that view name already exists) a lazily evaluated "view" that you can then use like a hive table in Spark SQL. It does not persist to memory unless you cache the dataset that underpins the view

edureka!

Presentation 61 / 93

Demo: Inferring Schema

```
[edureka_321047@ip-20-0-41-202 ~]$ pyspark2
Python 2.7.5 (default, Aug  4 2017, 00:39:18)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-16)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "ERROR".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.1.0.cloudera2
      /_/

Using Python version 2.7.5 (default, Aug  4 2017 00:39:18)
SparkSession available as 'spark'.
>>> from pyspark.sql import Row
>>> sc = spark.sparkContext
>>> lines = sc.textFile("/user/edureka_321047/sampleText")
>>> parts = lines.map(lambda l: l.split(","))
>>> people = parts.map(lambda p: Row(name=p[0], age=int(p[1])))
>>> schemaPeople = spark.createDataFrame(people)
>>> schemaPeople.createOrReplaceTempView("people")
>>> teenagers = spark.sql("SELECT name FROM people WHERE age >= 13 AND age <= 19")
>>> teenNames = teenagers.rdd.map(lambda p: "Name: " + p.name).collect()
```

We will run a SQL Query on the created table and store it inside an RDD

edureka!

Presentation 62 / 93

Demo: Inferring Schema

```
[edureka_321047@ip-20-0-41-202 ~]$ pyspark2
Python 2.7.5 (default, Aug  4 2017, 00:39:18)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-16)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "ERROR".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.1.0.cloudera2
      /_/

Using Python version 2.7.5 (default, Aug  4 2017 00:39:18)
SparkSession available as 'spark'.
>>> from pyspark.sql import Row
>>> sc = spark.sparkContext
>>> lines = sc.textFile("/user/edureka_321047/sampleText")
>>> parts = lines.map(lambda l: l.split(","))
>>> people = parts.map(lambda p: Row(name=p[0], age=int(p[1])))
>>> schemaPeople = spark.createDataFrame(people)
>>> schemaPeople.createOrReplaceTempView("people")
>>> teenagers = spark.sql("SELECT name FROM people WHERE age >= 13 AND age <= 19")
>>> teenNames = teenagers.rdd.map(lambda p: "Name: " + p.name).collect()
```

You can then map the RDD according to the type of value you want to fetch from it. Here we are fetching the Name of teenagers from the RDD and collect it inside another RDD

edureka!

Presentation 63 / 93

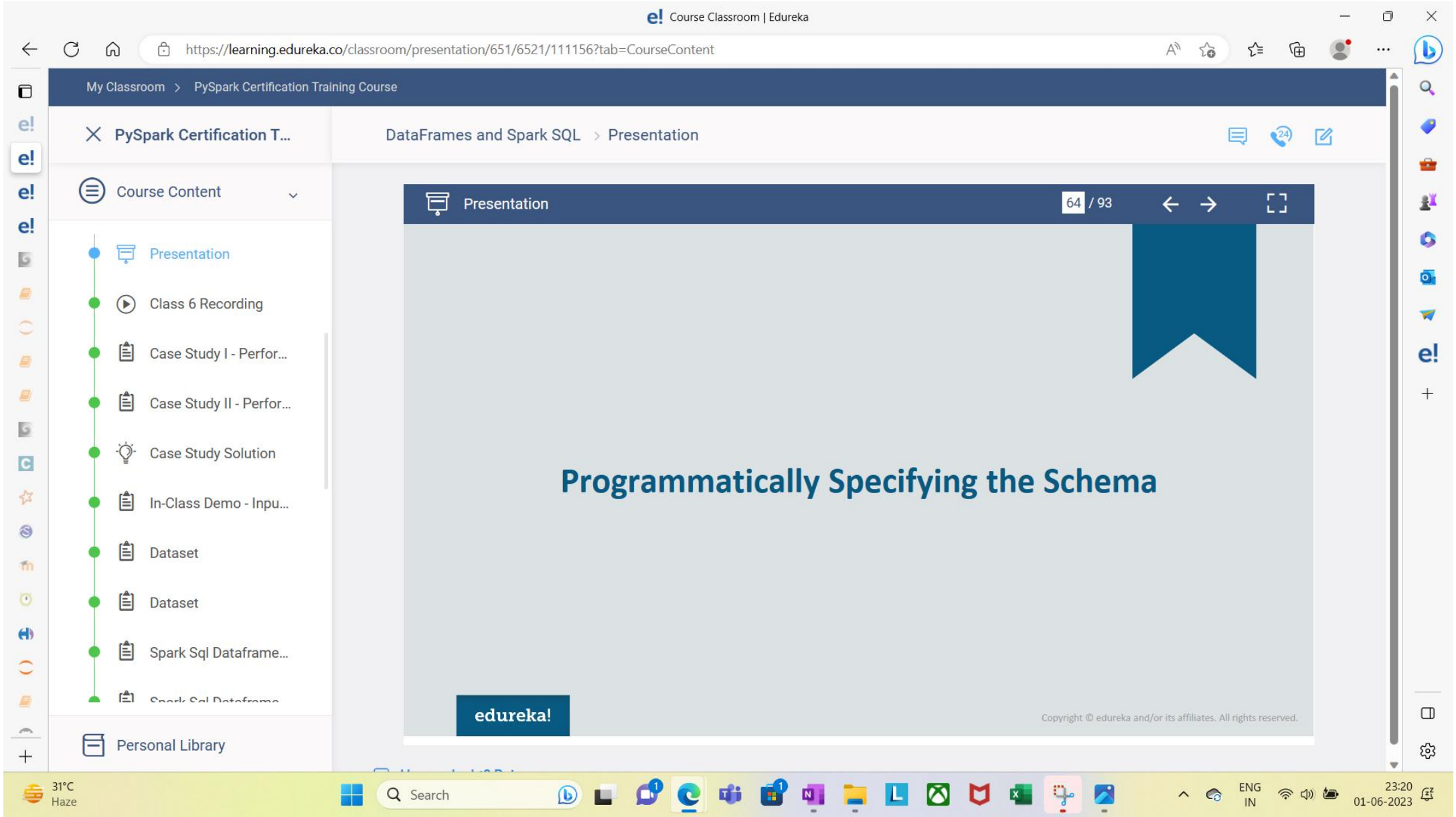
Demo: Inferring Schema

```
ip-20-0-41-190 login: edureka_321047
Password:
Last login: Thu Jul  5 09:11:39 on pts/72
[edureka_321047@ip-20-0-41-190 ~]$ pyspark2
Python 2.7.5 (default, Aug  4 2017, 00:39:18)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-16)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
Setting default log level to "ERROR".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.1.0.cloudera2
      /_/

Using Python version 2.7.5 (default, Aug  4 2017 00:39:18)
SparkSession available as 'spark'.
>>> from pyspark.sql import Row
>>> sc = spark.sparkContext
>>> lines = sc.textFile("/user/edureka_321047/sampleText")
>>> parts = lines.map(lambda l: l.split(","))
>>> people = parts.map(lambda p: Row(name=p[0], age=int(p[1])))
>>> schemaPeople = spark.createDataFrame(people)
>>> schemaPeople.createOrReplaceTempView("people")
>>> teenagers = spark.sql("SELECT name FROM people WHERE age >= 13 AND age <= 19")
>>> teenNames = teenagers.rdd.map(lambda p: "Name: " + p.name).collect()
>>> for name in teenNames:
...     print(name)
...
Name: Rohit
Name: Parth
Name: Varun
Name: Riya
Name: Krish
>>>
```

Now that we have created an RDD, we will print the list of values inside the RDD. You can see the resulting output of the print command wherein the Names of teenagers is printed

edureka!

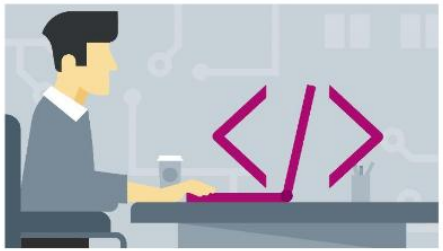

Course Classroom | Edureka
https://learning.edureka.co/classroom/presentation/651/6521/111156?tab=CourseContent
My Classroom > PySpark Certification Training Course
PySpark Certification T...
DataFrames and Spark SQL > Presentation
Course Content
Presentation
Class 6 Recording
Case Study I - Perfor...
Case Study II - Perfor...
Case Study Solution
In-Class Demo - Inpu...
Dataset
Dataset
Spark Sql Dataframe...
Personal Library
Presentation
64 / 93
Programmatically Specifying the Schema
edureka!
Copyright © edureka and/or its affiliates. All rights reserved.
31°C
Haze
Search
ENG
IN
23:20
01-06-2023

Presentation 65 / 93

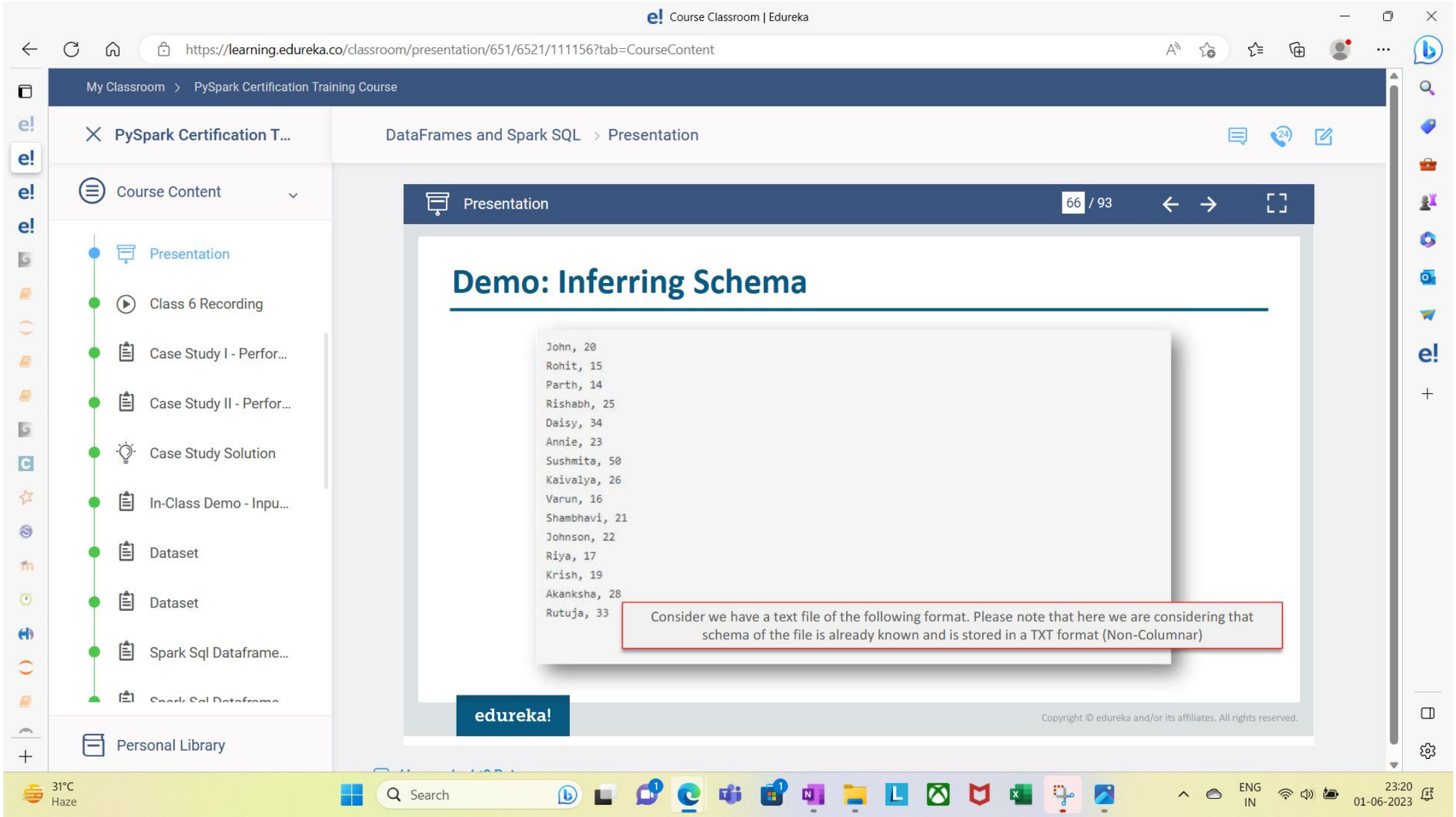
Specifying Schema Programmatically

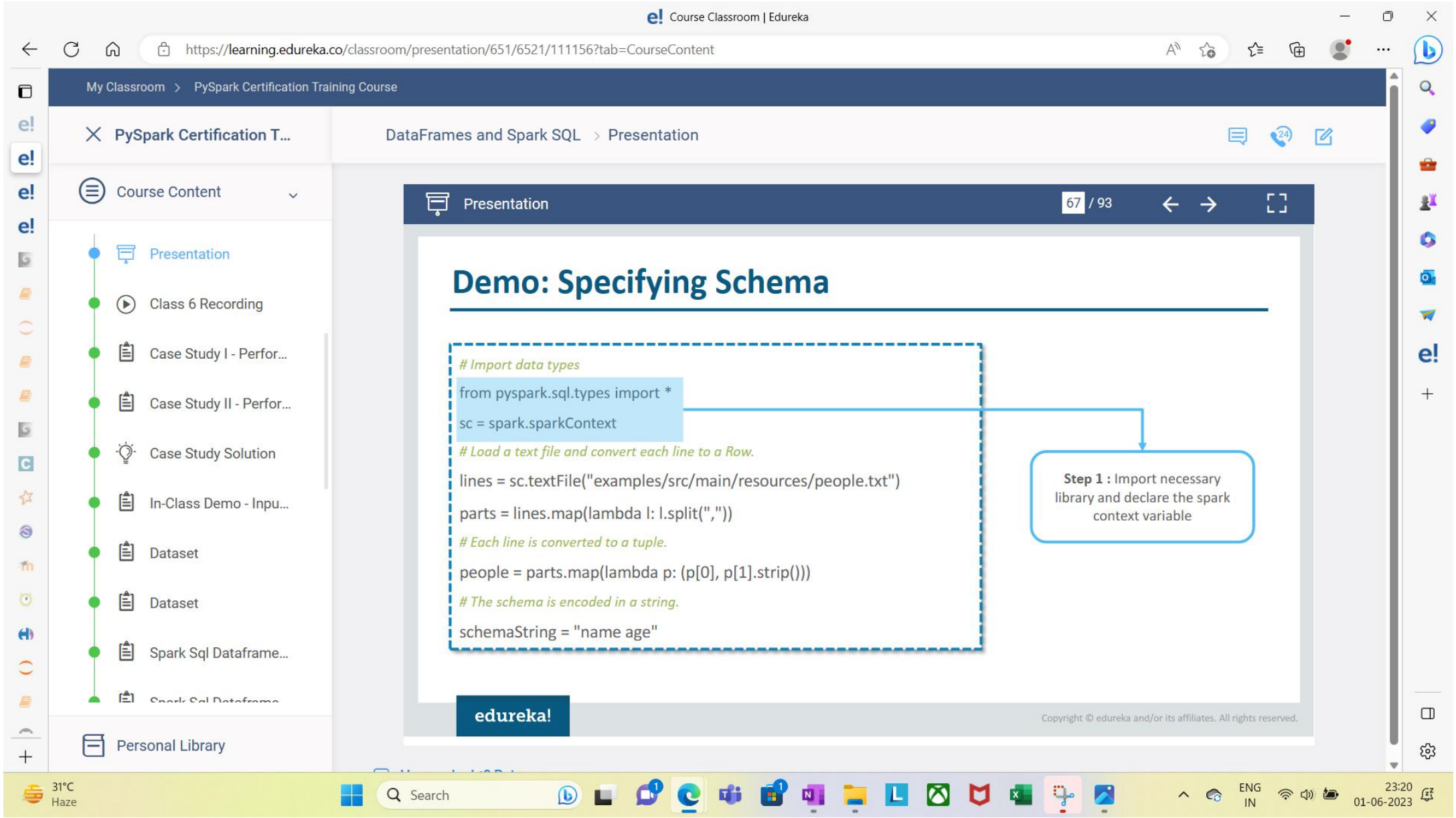
When case classes cannot be defined ahead of time (for example, the structure of records is encoded in a string, or a text dataset will be parsed and fields will be projected differently for different users), a DataFrame can be created programmatically with three steps

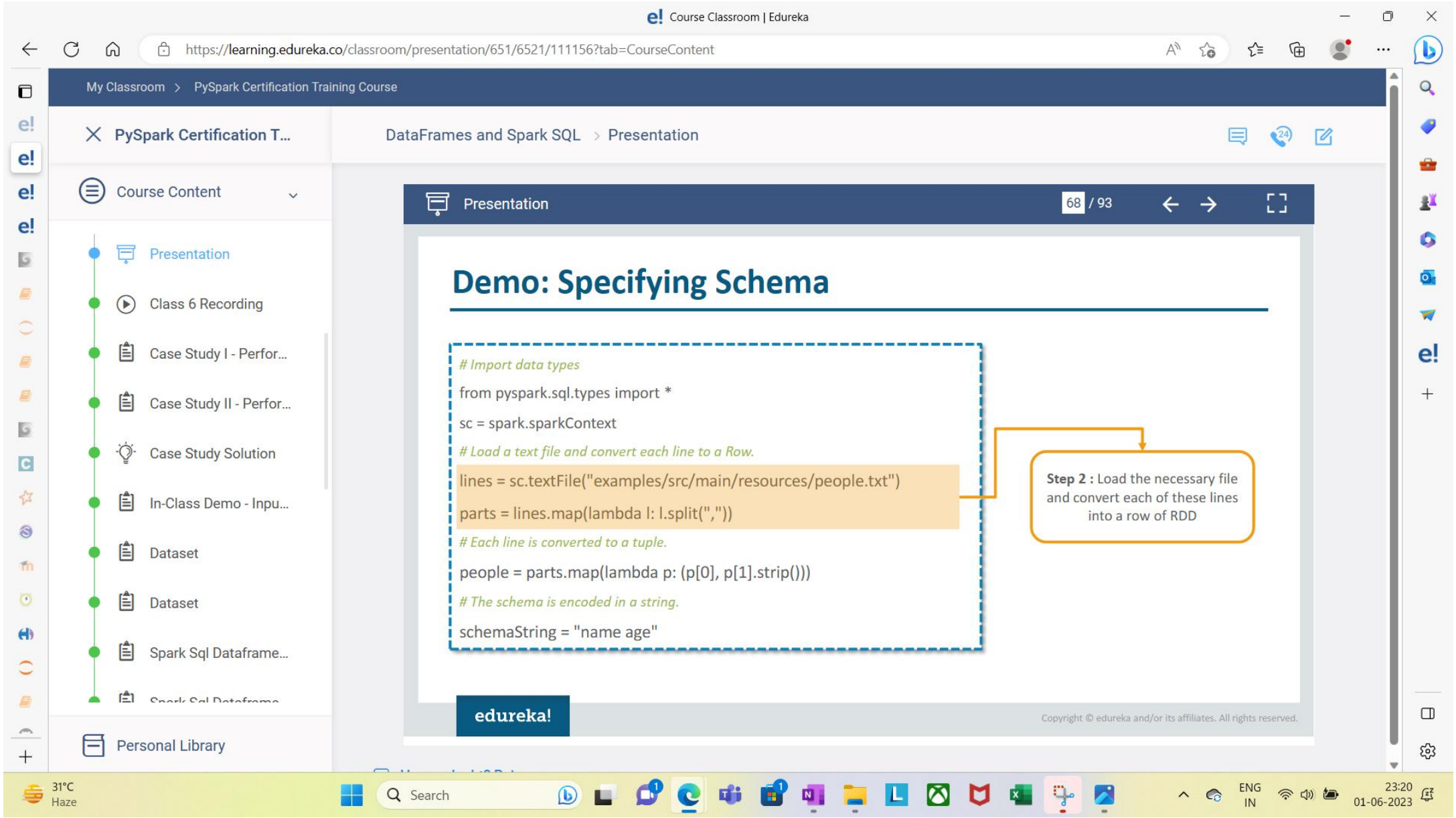
1. Create an RDD of Rows from the original RDD
2. Create the schema represented by a StructType matching the structure of Rows in the RDD created in Step 1
3. Apply the schema to the RDD of Rows via createDataFrame method provided by SparkSession

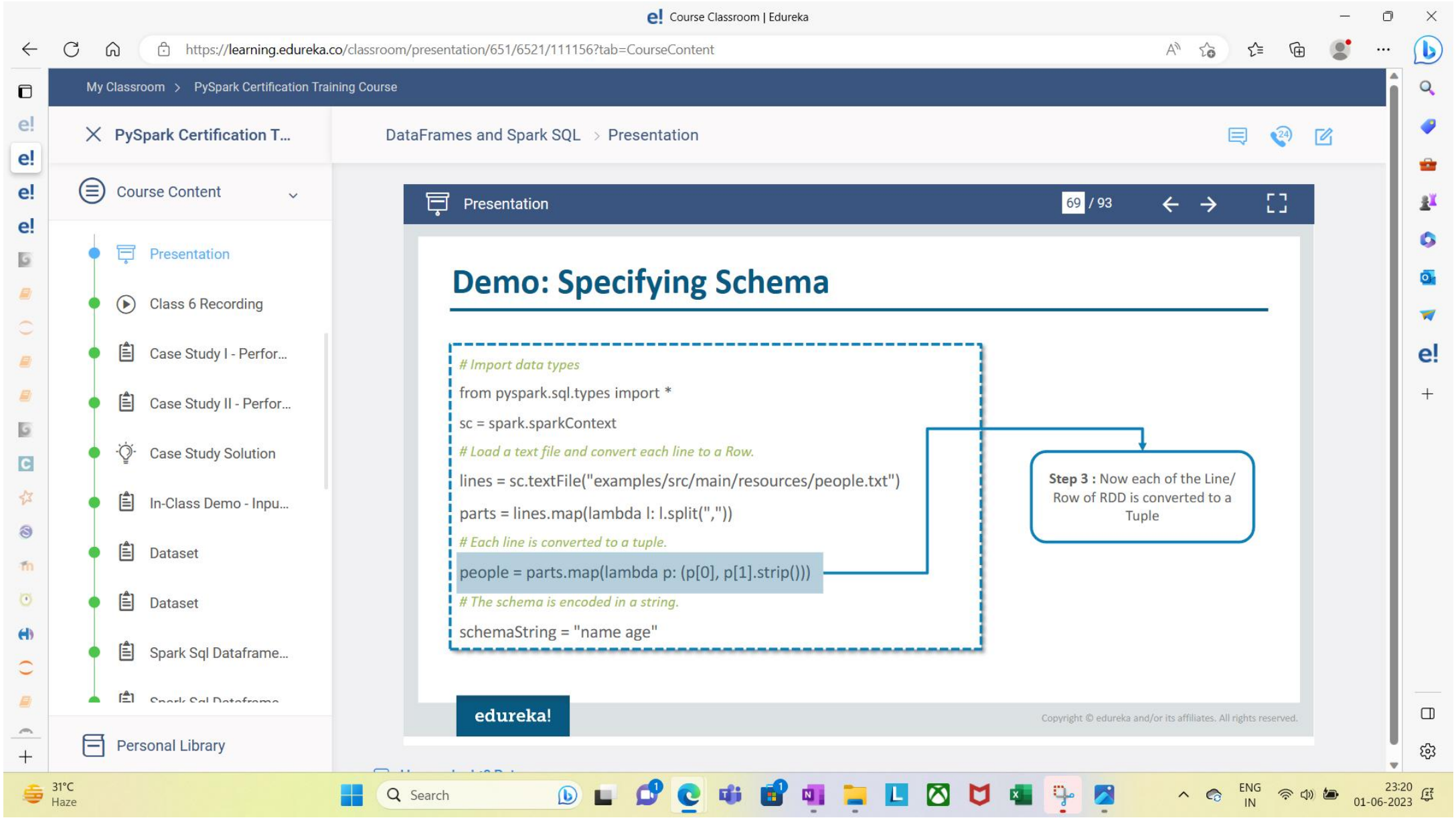


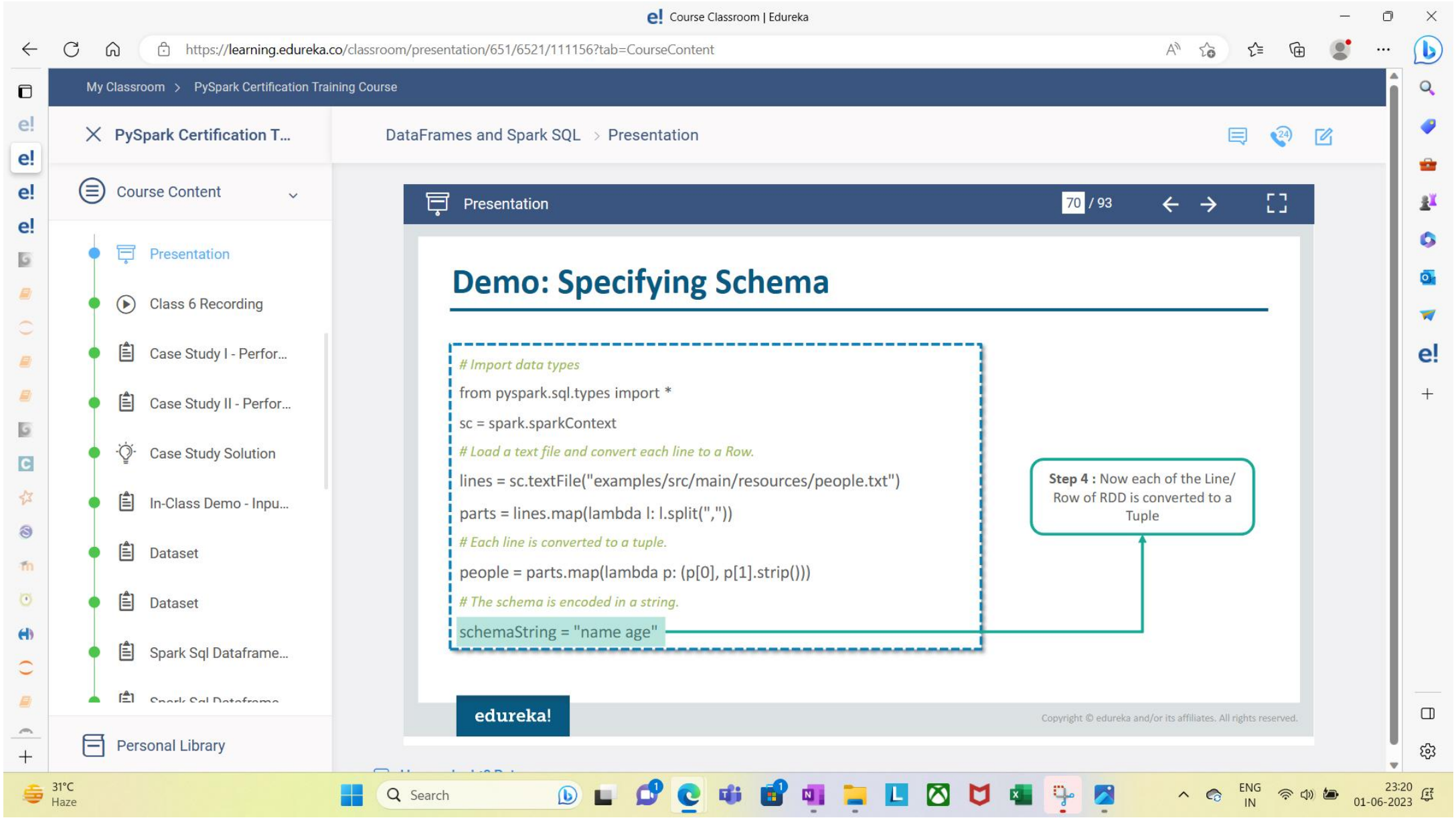
edureka!

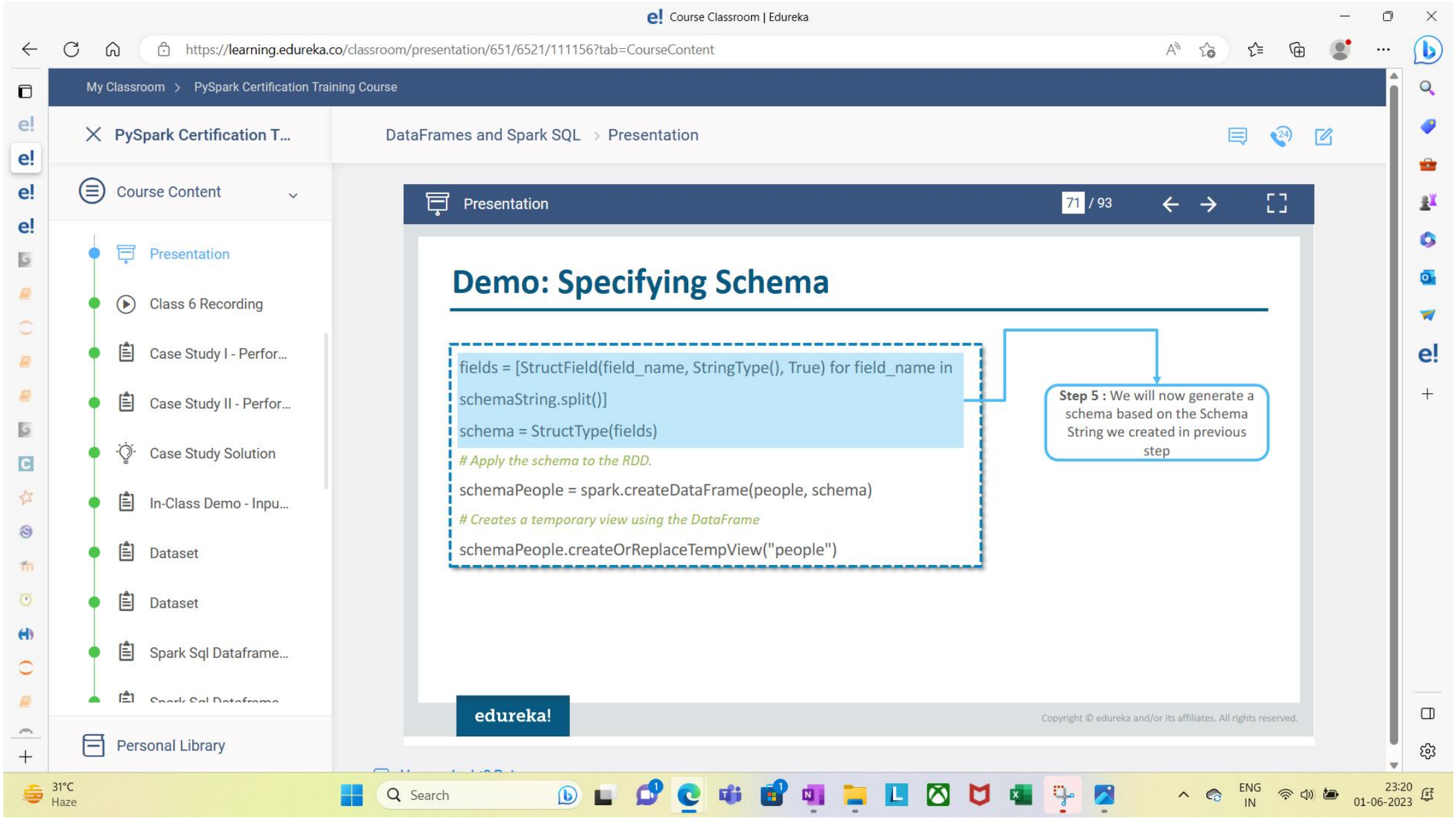

Course Classroom | Edureka
https://learning.edureka.co/classroom/presentation/651/6521/111156?tab=CourseContent
My Classroom > PySpark Certification Training Course
PySpark Certification T...
DataFrames and Spark SQL > Presentation
Course Content
Presentation
Class 6 Recording
Case Study I - Perfor...
Case Study II - Perfor...
Case Study Solution
In-Class Demo - Inpu...
Dataset
Dataset
Spark Sql Dataframe...
Personal Library
Presentation
66 / 93
Demo: Inferring Schema
John, 20
Rohit, 15
Parth, 14
Rishabh, 25
Daisy, 34
Annie, 23
Sushmita, 50
Kaivalya, 26
Varun, 16
Shambhavi, 21
Johnson, 22
Riya, 17
Krish, 19
Akanksha, 28
Rutuja, 33
Consider we have a text file of the following format. Please note that here we are considering that schema of the file is already known and is stored in a TXT format (Non-Columnar)
edureka!
Copyright © edureka and/or its affiliates. All rights reserved.
31°C Haze
Search
ENG IN
23:20 01-06-2023

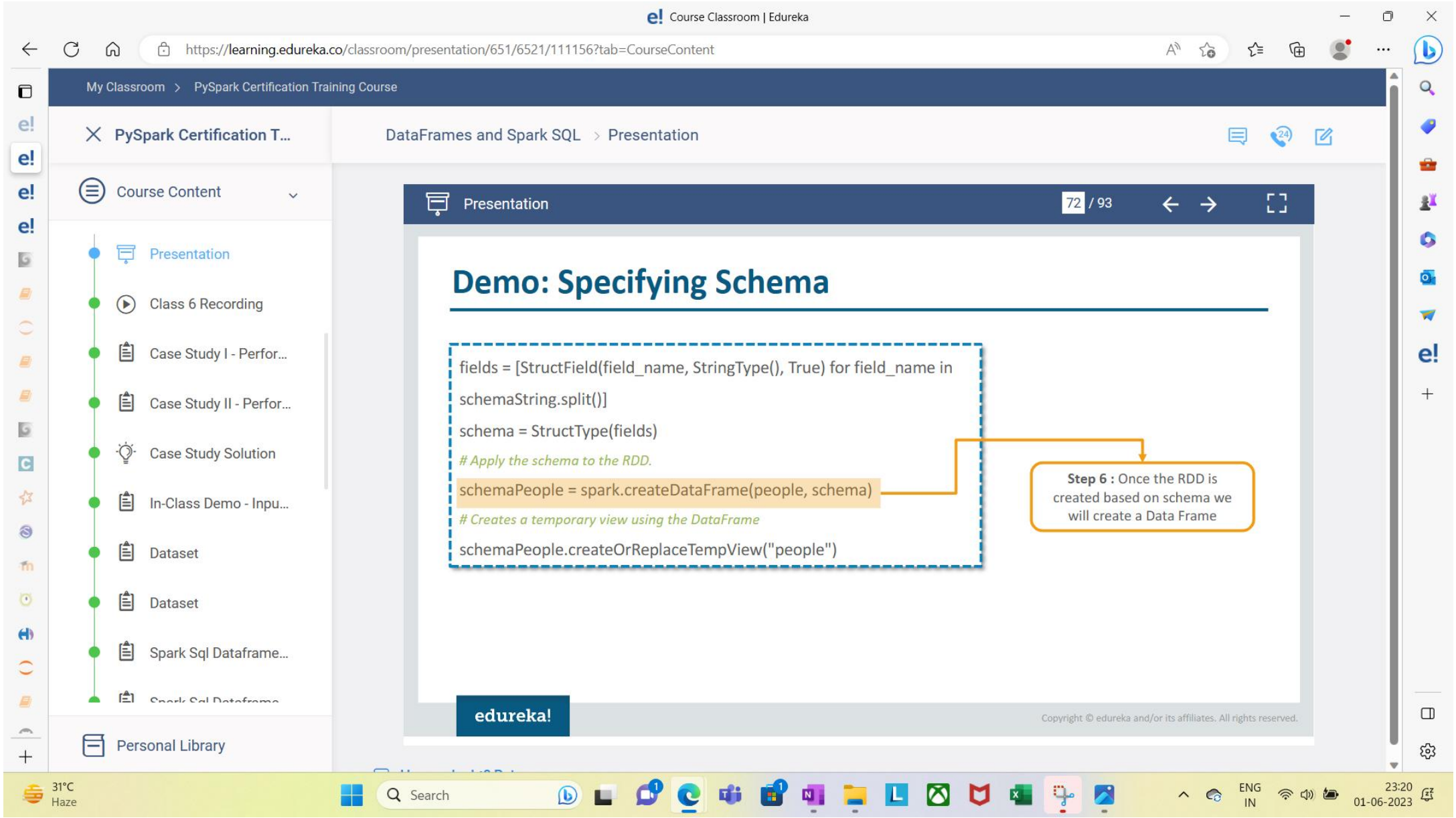

Course Classroom | Edureka
https://learning.edureka.co/classroom/presentation/651/6521/111156?tab=CourseContent
My Classroom > PySpark Certification Training Course
PySpark Certification T...
DataFrames and Spark SQL > Presentation
Course Content
Presentation
Class 6 Recording
Case Study I - Perfor...
Case Study II - Perfor...
Case Study Solution
In-Class Demo - Inpu...
Dataset
Dataset
Spark Sql Dataframe...
Personal Library
Presentation
67 / 93
Demo: Specifying Schema
# Import data types
from pyspark.sql.types import *
sc = spark.sparkContext
# Load a text file and convert each line to a Row.
lines = sc.textFile("examples/src/main/resources/people.txt")
parts = lines.map(lambda l: l.split(","))
# Each line is converted to a tuple.
people = parts.map(lambda p: (p[0], p[1].strip()))
# The schema is encoded in a string.
schemaString = "name age"
Step 1 : Import necessary library and declare the spark context variable
edureka!
Copyright © edureka and/or its affiliates. All rights reserved.
31°C Haze
Search
ENG IN
23:20 01-06-2023

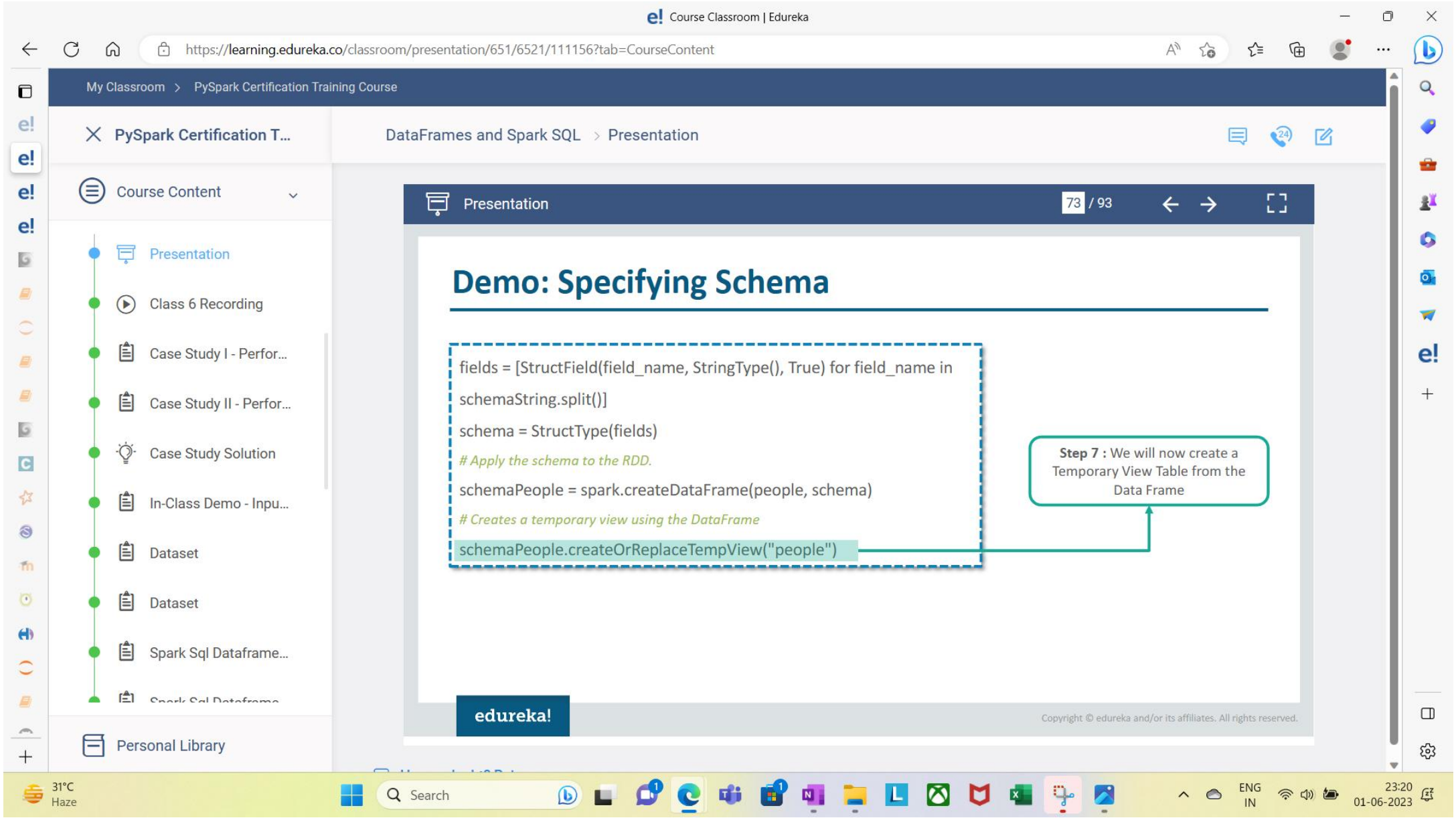

Course Classroom | Edureka
https://learning.edureka.co/classroom/presentation/651/6521/111156?tab=CourseContent
My Classroom > PySpark Certification Training Course
PySpark Certification T...
DataFrames and Spark SQL > Presentation
Course Content
Presentation
Class 6 Recording
Case Study I - Perfor...
Case Study II - Perfor...
Case Study Solution
In-Class Demo - Inpu...
Dataset
Dataset
Spark Sql Dataframe...
Personal Library
Presentation
68 / 93
Demo: Specifying Schema
# Import data types
from pyspark.sql.types import *
sc = spark.sparkContext
# Load a text file and convert each line to a Row.
lines = sc.textFile("examples/src/main/resources/people.txt")
parts = lines.map(lambda l: l.split(","))
# Each line is converted to a tuple.
people = parts.map(lambda p: (p[0], p[1].strip()))
# The schema is encoded in a string.
schemaString = "name age"
Step 2 : Load the necessary file and convert each of these lines into a row of RDD
edureka!
Copyright © edureka and/or its affiliates. All rights reserved.
31°C Haze
Search
ENG IN
23:20 01-06-2023


Course Classroom | Edureka
https://learning.edureka.co/classroom/presentation/651/6521/111156?tab=CourseContent
My Classroom > PySpark Certification Training Course
PySpark Certification T...
DataFrames and Spark SQL > Presentation
Course Content
Presentation
Class 6 Recording
Case Study I - Perfor...
Case Study II - Perfor...
Case Study Solution
In-Class Demo - Inpu...
Dataset
Dataset
Spark Sql Dataframe...
Personal Library
Presentation
69 / 93
Demo: Specifying Schema
# Import data types
from pyspark.sql.types import *
sc = spark.sparkContext
# Load a text file and convert each line to a Row.
lines = sc.textFile("examples/src/main/resources/people.txt")
parts = lines.map(lambda l: l.split(","))
# Each line is converted to a tuple.
people = parts.map(lambda p: (p[0], p[1].strip()))
# The schema is encoded in a string.
schemaString = "name age"
Step 3 : Now each of the Line/ Row of RDD is converted to a Tuple
edureka!
Copyright © edureka and/or its affiliates. All rights reserved.
31°C Haze
Search
ENG IN
23:20 01-06-2023


Course Classroom | Edureka
https://learning.edureka.co/classroom/presentation/651/6521/111156?tab=CourseContent
My Classroom > PySpark Certification Training Course
PySpark Certification T...
DataFrames and Spark SQL > Presentation
Course Content
Presentation
Class 6 Recording
Case Study I - Perfor...
Case Study II - Perfor...
Case Study Solution
In-Class Demo - Inpu...
Dataset
Dataset
Spark Sql Dataframe...
Personal Library
Presentation
70 / 93
Demo: Specifying Schema
# Import data types
from pyspark.sql.types import *
sc = spark.sparkContext
# Load a text file and convert each line to a Row.
lines = sc.textFile("examples/src/main/resources/people.txt")
parts = lines.map(lambda l: l.split(","))
# Each line is converted to a tuple.
people = parts.map(lambda p: (p[0], p[1].strip()))
# The schema is encoded in a string.
schemaString = "name age"
Step 4 : Now each of the Line/ Row of RDD is converted to a Tuple
edureka!
Copyright © edureka and/or its affiliates. All rights reserved.
31°C Haze
Search
ENG IN
23:20 01-06-2023


Course Classroom | Edureka
https://learning.edureka.co/classroom/presentation/651/6521/111156?tab=CourseContent
My Classroom > PySpark Certification Training Course
PySpark Certification T...
DataFrames and Spark SQL > Presentation
Course Content
Presentation
Class 6 Recording
Case Study I - Perfor...
Case Study II - Perfor...
Case Study Solution
In-Class Demo - Inpu...
Dataset
Dataset
Spark Sql Dataframe...
Personal Library
Presentation
71 / 93
Demo: Specifying Schema
fields = [StructField(field_name, StringType(), True) for field_name in schemaString.split()]
schema = StructType(fields)
# Apply the schema to the RDD.
schemaPeople = spark.createDataFrame(people, schema)
# Creates a temporary view using the DataFrame
schemaPeople.createOrReplaceTempView("people")
Step 5 : We will now generate a schema based on the Schema String we created in previous step
edureka!
Copyright © edureka and/or its affiliates. All rights reserved.
31°C Haze
Search
ENG IN
23:20 01-06-2023


e! Course Classroom | Edureka
https://learning.edureka.co/classroom/presentation/651/6521/111156?tab=CourseContent
My Classroom > PySpark Certification Training Course
PySpark Certification T...
DataFrames and Spark SQL > Presentation
Course Content
Presentation
Class 6 Recording
Case Study I - Perfor...
Case Study II - Perfor...
Case Study Solution
In-Class Demo - Inpu...
Dataset
Dataset
Spark Sql Dataframe...
Personal Library
Presentation
72 / 93
Demo: Specifying Schema
fields = [StructField(field_name, StringType(), True) for field_name in
schemaString.split()]
schema = StructType(fields)
# Apply the schema to the RDD.
schemaPeople = spark.createDataFrame(people, schema)
# Creates a temporary view using the DataFrame
schemaPeople.createOrReplaceTempView("people")
Step 6 : Once the RDD is created based on schema we will create a Data Frame
edureka!
Copyright © edureka and/or its affiliates. All rights reserved.
31°C Haze
Search
ENG IN
23:20 01-06-2023


Course Classroom | Edureka
https://learning.edureka.co/classroom/presentation/651/6521/111156?tab=CourseContent
My Classroom > PySpark Certification Training Course
PySpark Certification T...
DataFrames and Spark SQL > Presentation
Course Content
Presentation
Class 6 Recording
Case Study I - Perfor...
Case Study II - Perfor...
Case Study Solution
In-Class Demo - Inpu...
Dataset
Dataset
Spark Sql Dataframe...
Personal Library
Presentation
73 / 93
Demo: Specifying Schema
fields = [StructField(field_name, StringType(), True) for field_name in
schemaString.split()]
schema = StructType(fields)
# Apply the schema to the RDD.
schemaPeople = spark.createDataFrame(people, schema)
# Creates a temporary view using the DataFrame
schemaPeople.createOrReplaceTempView("people")
Step 7 : We will now create a Temporary View Table from the Data Frame
edureka!
Copyright © edureka and/or its affiliates. All rights reserved.
31°C
Haze
Search
ENG
IN
23:20
01-06-2023

Presentation 74 / 93

Demo: Specifying Schema

```
// SQL can be run over DataFrames that have been registered as a table.
results = spark.sql("SELECT name FROM people")

results.show()
```

Step 8 : We can now run the SQL query over the temporary view table that we created

edureka!

Presentation 75 / 93

Demo: Specifying Schema

```
// SQL can be run over DataFrames that have been registered as a table.
results = spark.sql("SELECT name FROM people")

results.show()
```

Step 9 : This is the action which will print all values from the table satisfying the query

edureka!

Presentation 76 / 93

Demo: Specifying Schema

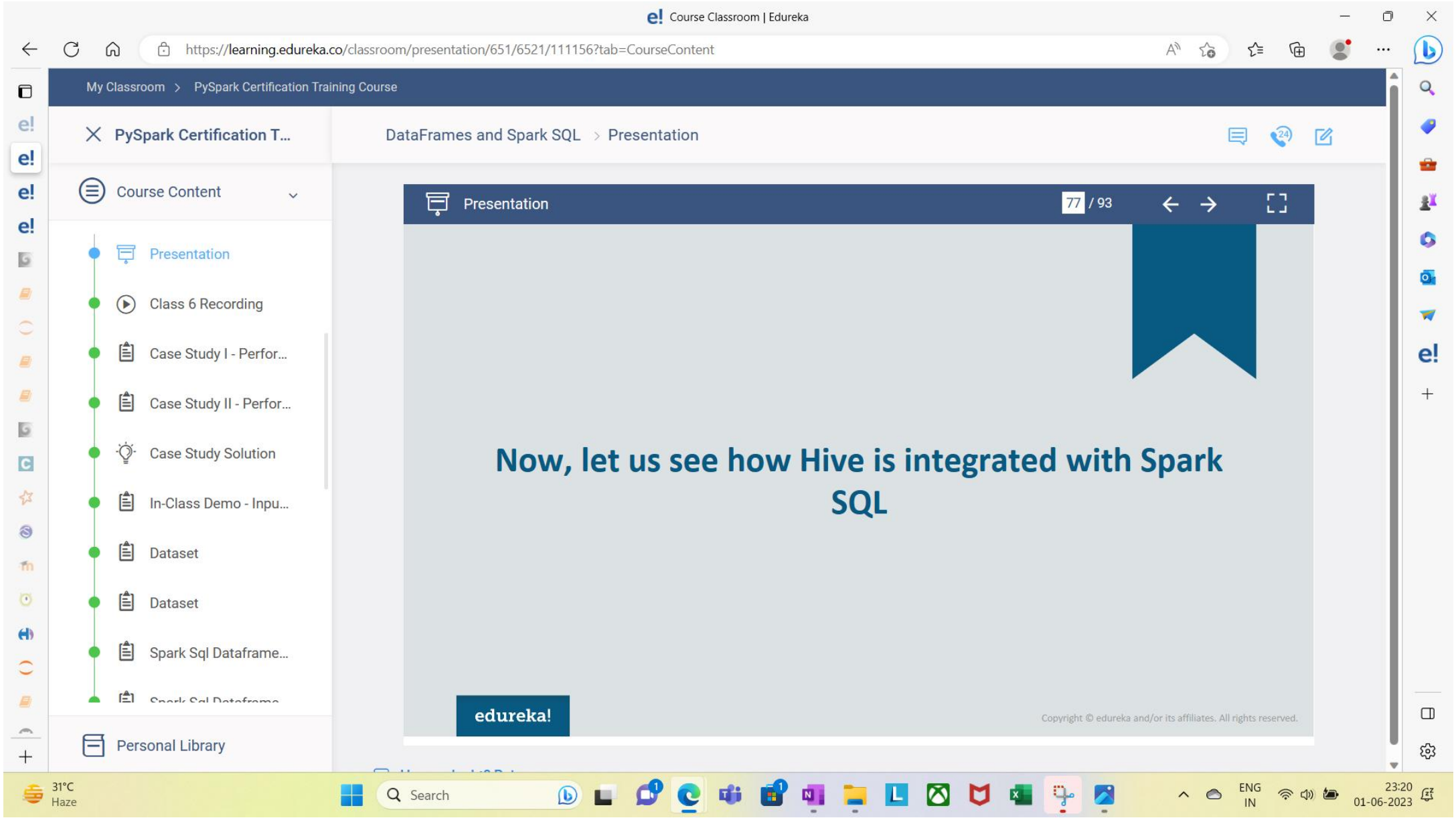
```
      /__ / .__/\_,_/_/ /_/\_\   version 2.1.0.cloudera2
         /_/

Using Python version 2.7.5 (default, Aug  4 2017 00:39:18)
SparkSession available as 'spark'.
>>> from pyspark.sql.types import *
>>> sc = spark.sparkContext
>>> lines = sc.textFile("/user/edureka_321047/sampleText")
>>> parts = lines.map(lambda l: l.split(","))
>>> people = parts.map(lambda p: (p[0], p[1].strip()))
>>> schemaString = "name age"
>>> fields = [StructField(field_name, StringType(), True) for field_name in schemaString.split()]
>>> schema = StructType(fields)
>>> schemaPeople = spark.createDataFrame(people, schema)
>>> schemaPeople.createOrReplaceTempView("people")
>>> results = spark.sql("SELECT name FROM people")
>>> results.show()
+---------+
|     name|
+---------+
|     John|
|    Rohit|
|    Parth|
|  Rishabh|
|    Daisy|
|    Annie|
| Sushmita|
| Kaivalya|
|    Varun|
|Shambhavi|
|  Johnson|
|      Riya|
|     Krish|
| Akanksha|
|    Rutuja|
+---------+

>>>
```

You can see the output that we get for the query

edureka!


Course Classroom | Edureka
https://learning.edureka.co/classroom/presentation/651/6521/111156?tab=CourseContent
My Classroom > PySpark Certification Training Course
PySpark Certification T...
DataFrames and Spark SQL > Presentation
Course Content
Presentation
Class 6 Recording
Case Study I - Perfor...
Case Study II - Perfor...
Case Study Solution
In-Class Demo - Inpu...
Dataset
Dataset
Spark Sql Dataframe...
Personal Library
Presentation
77 / 93
Now, let us see how Hive is integrated with Spark SQL
edureka!
Copyright © edureka and/or its affiliates. All rights reserved.
31°C
Haze
Search
ENG
IN
23:20
01-06-2023

Presentation 78 / 93

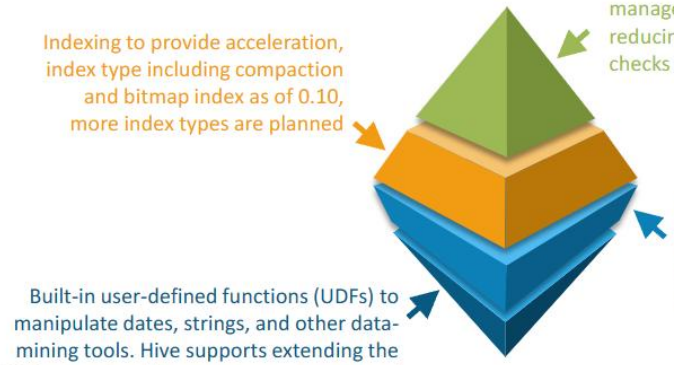
Understanding Apache Hive

1. **Apache Hive** is a data warehouse software project built on top of Apache Hadoop for providing data summarization, query and analysis
2. Hive gives an SQL-like interface to query data stored in various databases and file systems that integrate with Hadoop and fetches results
3. Traditional SQL queries must be implemented in the MapReduce Java API to execute SQL applications and queries over distributed data
4. Hive provides the necessary SQL abstraction to integrate SQL-like queries (HiveQL) into the underlying Java without the need to implement queries in the low-level Java API

edureka!

Presentation 79 / 93

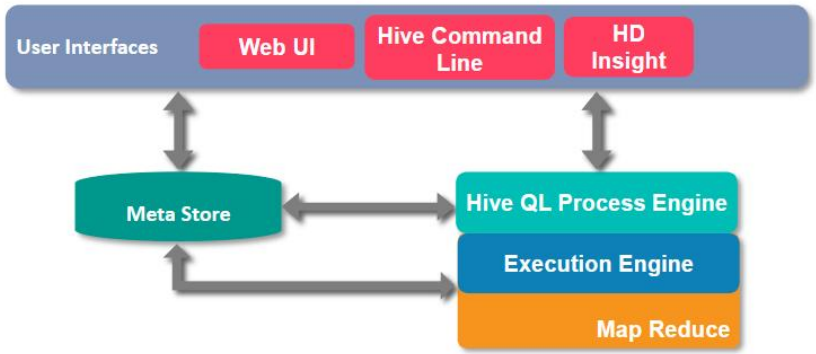
Features of Apache Hive



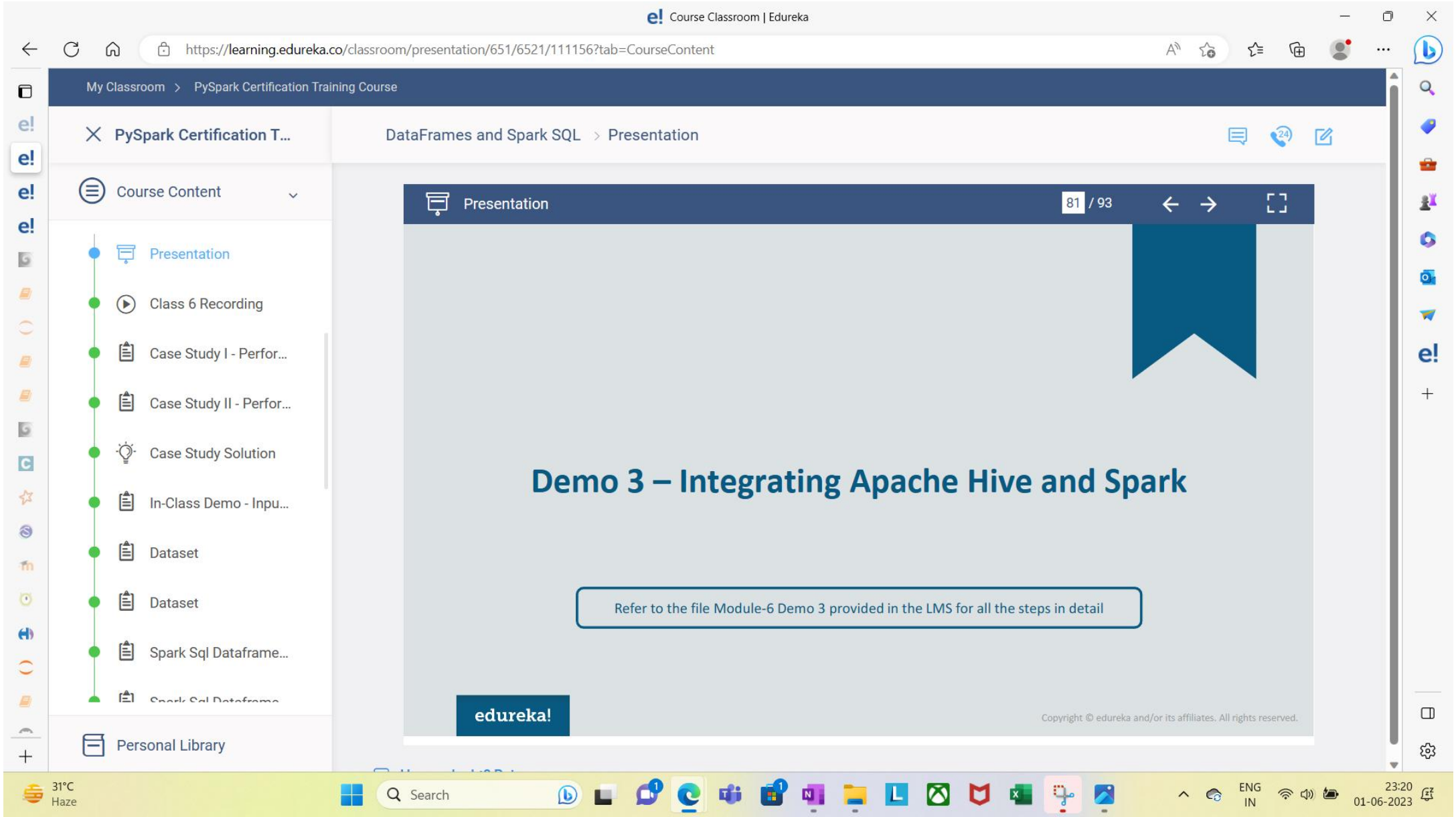
edureka!

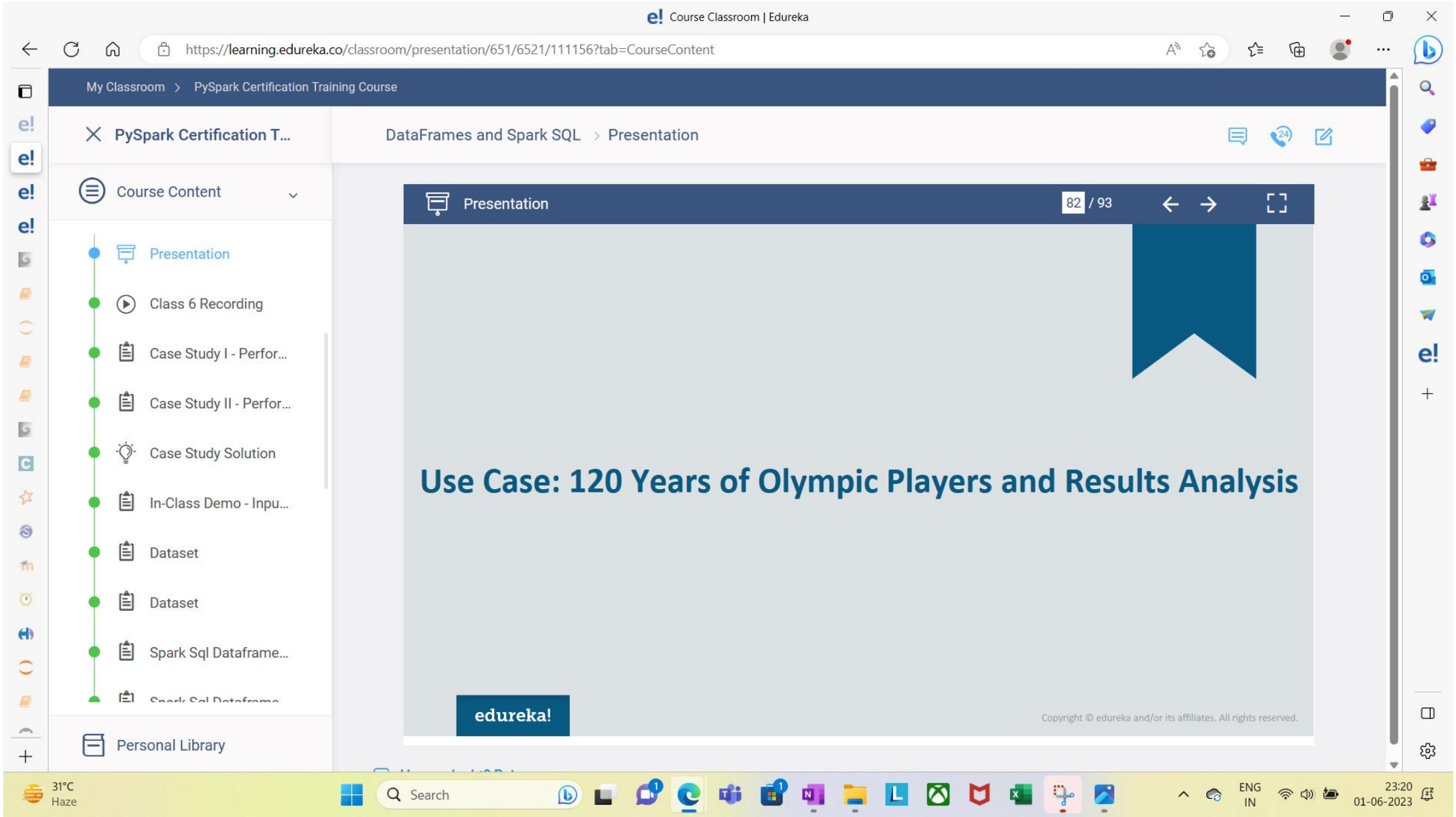
Presentation 80 / 93

Hive Architecture



edureka!


e! Course Classroom | Edureka
https://learning.edureka.co/classroom/presentation/651/6521/111156?tab=CourseContent
My Classroom > PySpark Certification Training Course
PySpark Certification T...
DataFrames and Spark SQL > Presentation
Course Content
Presentation
Class 6 Recording
Case Study I - Perfor...
Case Study II - Perfor...
Case Study Solution
In-Class Demo - Inpu...
Dataset
Dataset
Spark Sql Dataframe...
Personal Library
Presentation
81 / 93
Demo 3 – Integrating Apache Hive and Spark
Refer to the file Module-6 Demo 3 provided in the LMS for all the steps in detail
edureka!
Copyright © edureka and/or its affiliates. All rights reserved.
31°C
Haze
Search
ENG
IN
23:20
01-06-2023


Course Classroom | Edureka
https://learning.edureka.co/classroom/presentation/651/6521/111156?tab=CourseContent
My Classroom > PySpark Certification Training Course
PySpark Certification T...
DataFrames and Spark SQL > Presentation
Course Content
Presentation
Class 6 Recording
Case Study I - Perfor...
Case Study II - Perfor...
Case Study Solution
In-Class Demo - Inpu...
Dataset
Dataset
Spark Sql Dataframe...
Personal Library
Presentation
82 / 93
Use Case: 120 Years of Olympic Players and Results Analysis
edureka!
Copyright © edureka and/or its affiliates. All rights reserved.
31°C
Haze
Search
ENG
IN
23:20
01-06-2023

Presentation 83 / 93

Use Case: Problem Statement

- Analysis on sports data is very important
- It provides relevant insights into the data and helps us garner better results out of it
- One such important event is the Olympics Games
- The Olympic Games are held every four years, with the Summer and Winter Games alternating by occurring every four years but two years apart
- You will have to perform multiple computations on a given given set of data and provide insights on it
- We will now discuss the outputs that you have to provide from given Dataset and have a look at the kind of data we have

edureka!

Presentation 84 / 93

Use Case: Data Overview

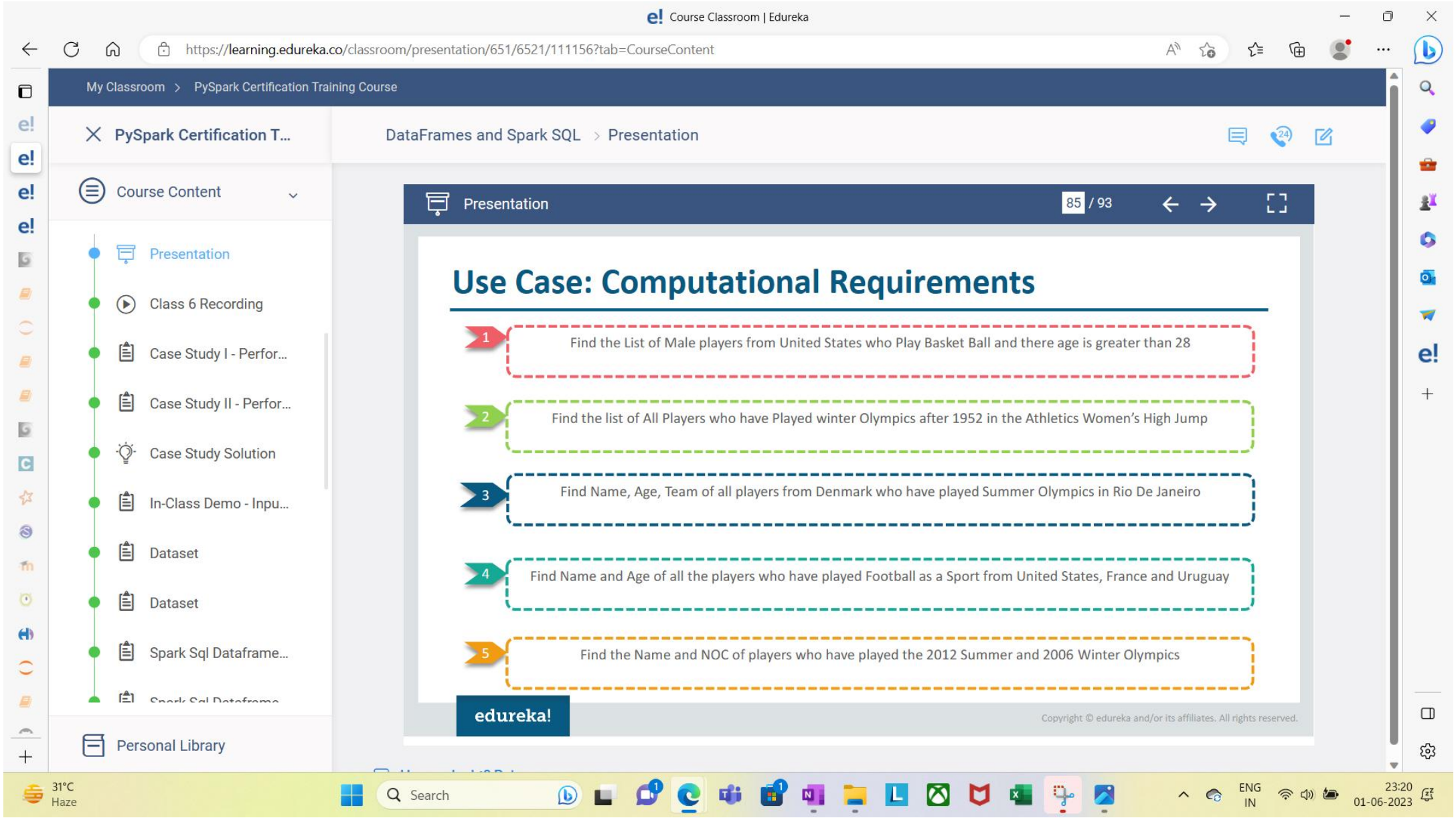
ID	Name	Sex	Age	Height	Weight	Team	NOC	Games	Year	Season	City	Sport	Event	Medal
1	A Dijiang	M	24	180	80	China	CHN	992 Summe	1992	Summer	Barcelona	Basketball	ll Men's Ba	NA
2	A Lamusi	M	23	170	60	China	CHN	012 Summe	2012	Summer	London	Judo	's Extra-Lig	NA
3	ar Nielsen	M	24	NA	NA	Denmark	DEN	920 Summe	1920	Summer	ntwerper	Football	ll Men's Fo	NA
4	Lindenau	M	34	NA	NA	mark/Swe	DEN	900 Summe	1900	Summer	Paris	ug-Of-War	Men's Tu	Gold
5	ne Jacoba	F	21	185	82	letherland	NED	988 Winte	1988	Winter	Calgary	eed Skating	Women's	NA
5	ne Jacoba	F	21	185	82	letherland	NED	988 Winte	1988	Winter	Calgary	eed Skating	Women's	NA

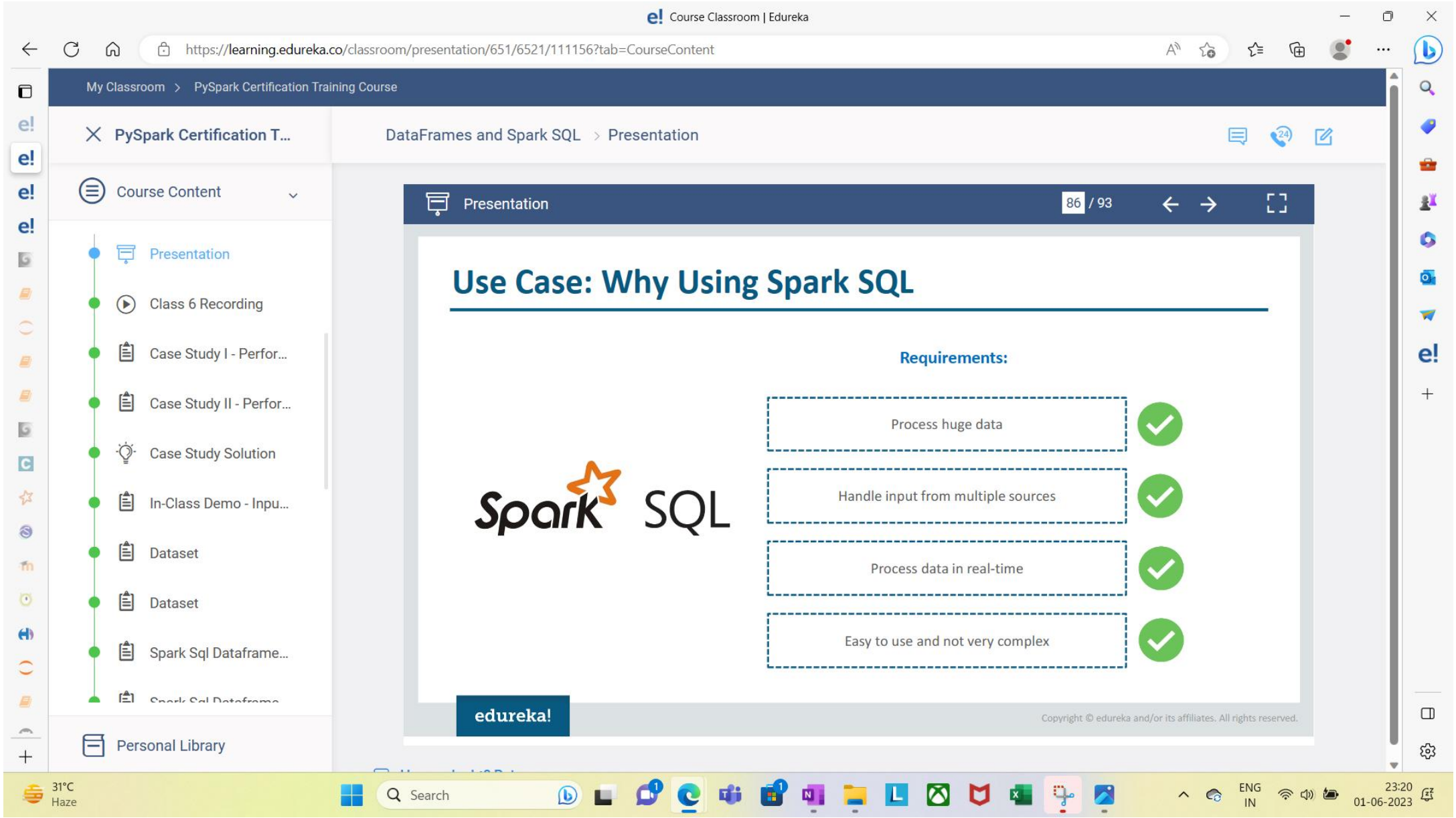
athlete_events

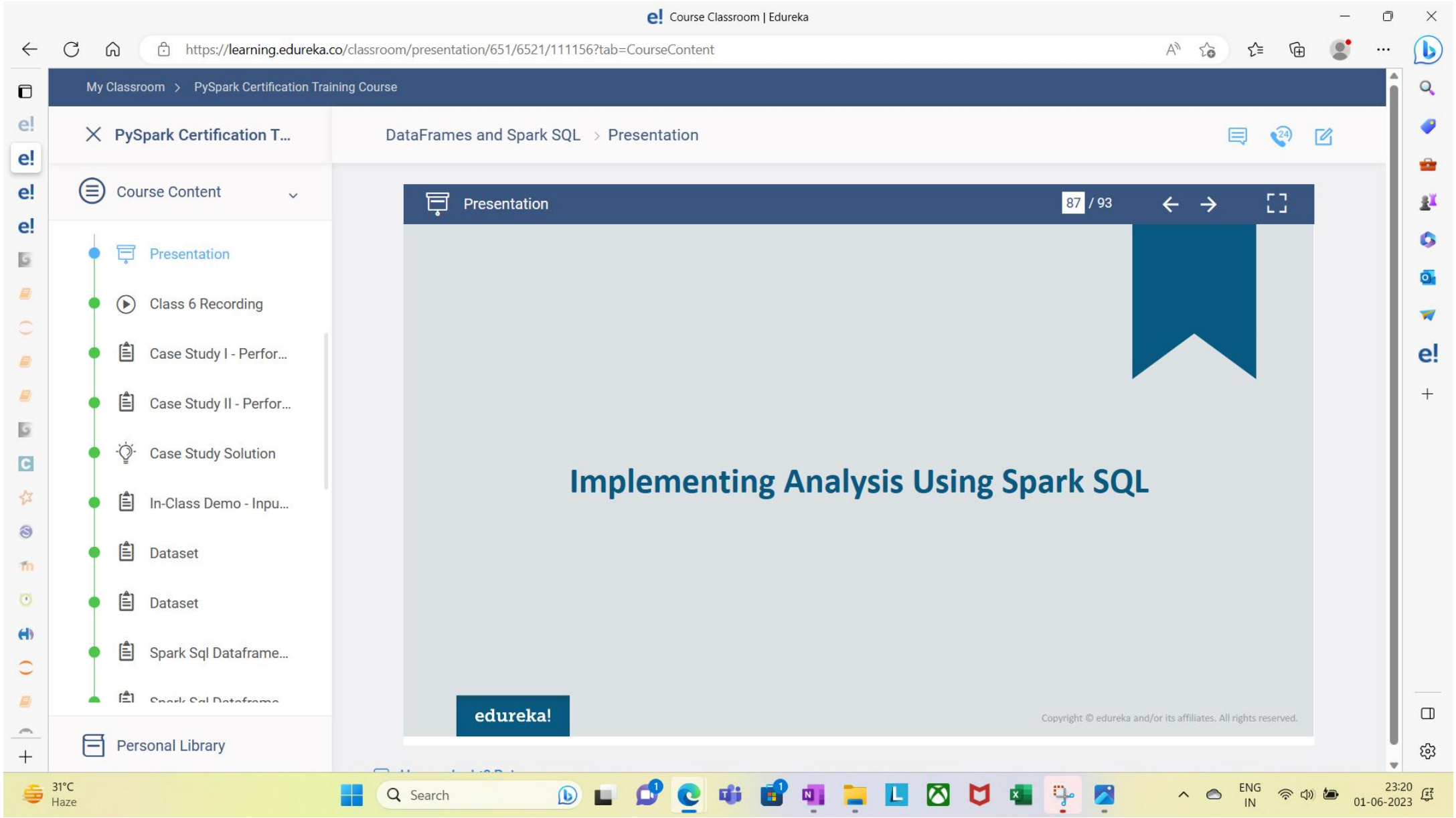
You can download the dataset from here

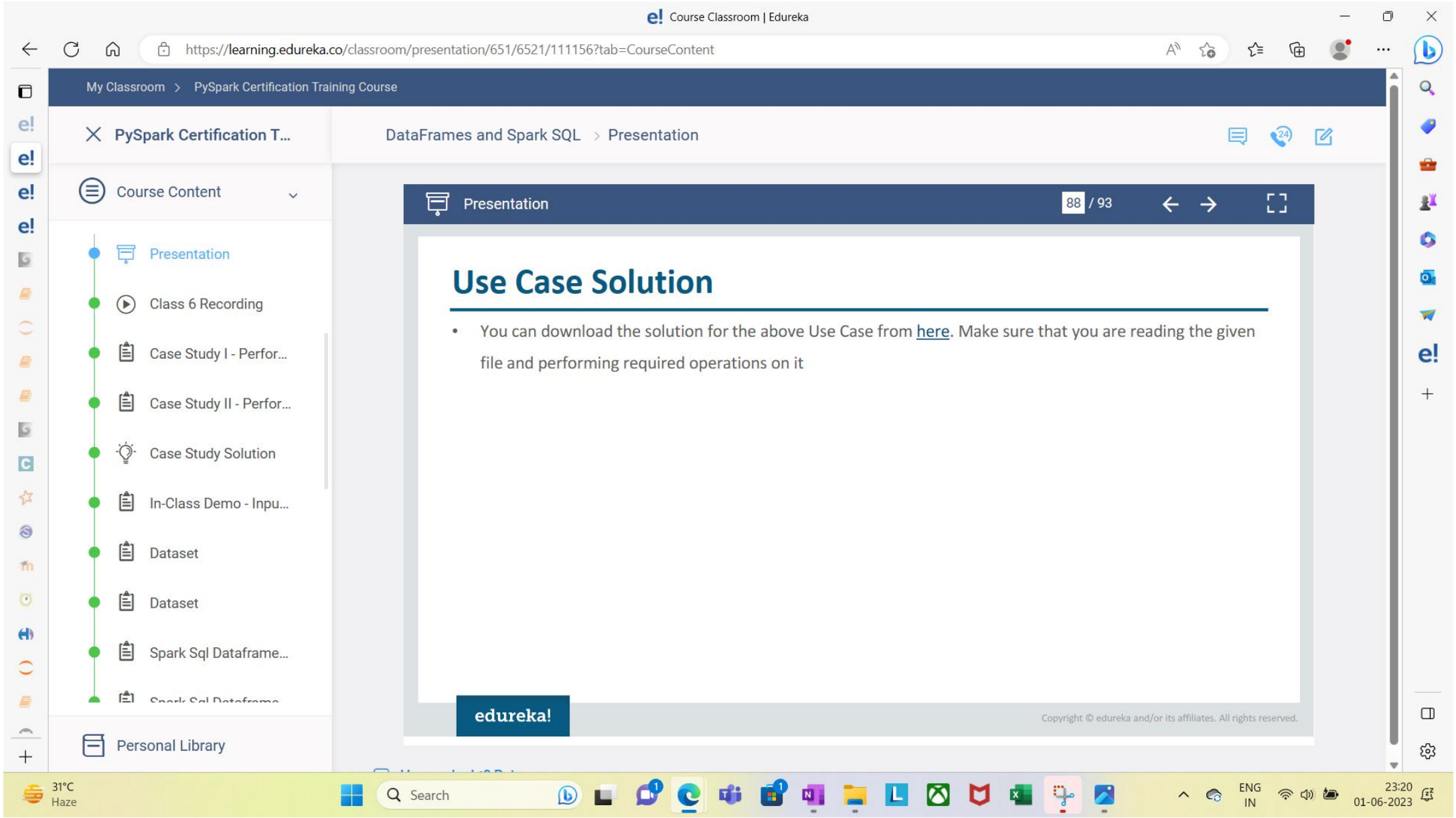
Given is the CSV file containing details of Players, Teams, Sports etc. You have to perform the analysis on this data and provide relevant outputs. Let us now discuss the various operations that you need to perform over this data set

edureka!


Course Classroom | Edureka
https://learning.edureka.co/classroom/presentation/651/6521/111156?tab=CourseContent
My Classroom > PySpark Certification Training Course
PySpark Certification T...
DataFrames and Spark SQL > Presentation
Course Content
Presentation
Class 6 Recording
Case Study I - Perfor...
Case Study II - Perfor...
Case Study Solution
In-Class Demo - Inpu...
Dataset
Dataset
Spark Sql Dataframe...
Personal Library
Presentation
85 / 93
Use Case: Computational Requirements
1 Find the List of Male players from United States who Play Basket Ball and there age is greater than 28
2 Find the list of All Players who have Played winter Olympics after 1952 in the Athletics Women's High Jump
3 Find Name, Age, Team of all players from Denmark who have played Summer Olympics in Rio De Janeiro
4 Find Name and Age of all the players who have played Football as a Sport from United States, France and Uruguay
5 Find the Name and NOC of players who have played the 2012 Summer and 2006 Winter Olympics
edureka!
Copyright © edureka and/or its affiliates. All rights reserved.
31°C Haze
Search
ENG IN
23:20 01-06-2023

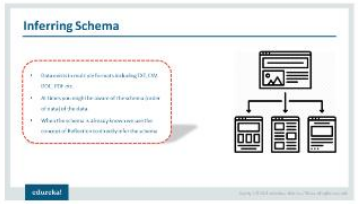

e! Course Classroom | Edureka
https://learning.edureka.co/classroom/presentation/651/6521/111156?tab=CourseContent
My Classroom > PySpark Certification Training Course
PySpark Certification T...
DataFrames and Spark SQL > Presentation
Course Content
Presentation
Class 6 Recording
Case Study I - Perfor...
Case Study II - Perfor...
Case Study Solution
In-Class Demo - Inpu...
Dataset
Dataset
Spark Sql Dataframe...
Personal Library
Presentation
86 / 93
Use Case: Why Using Spark SQL
Requirements:
Process huge data
Handle input from multiple sources
Process data in real-time
Easy to use and not very complex
Spark SQL
edureka!
Copyright © edureka and/or its affiliates. All rights reserved.
31°C
Haze
Search
ENG
IN
23:20
01-06-2023

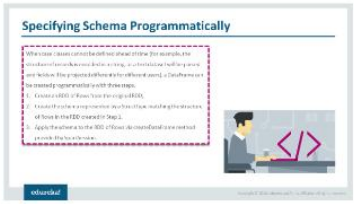

Course Classroom | Edureka
https://learning.edureka.co/classroom/presentation/651/6521/111156?tab=CourseContent
My Classroom > PySpark Certification Training Course
PySpark Certification T...
DataFrames and Spark SQL > Presentation
Course Content
Presentation
Class 6 Recording
Case Study I - Perfor...
Case Study II - Perfor...
Case Study Solution
In-Class Demo - Inpu...
Dataset
Dataset
Spark Sql Dataframe...
Personal Library
Presentation
87 / 93
Implementing Analysis Using Spark SQL
edureka!
Copyright © edureka and/or its affiliates. All rights reserved.
31°C
Haze
Search
ENG
IN
23:20
01-06-2023


Course Classroom | Edureka
https://learning.edureka.co/classroom/presentation/651/6521/111156?tab=CourseContent
My Classroom > PySpark Certification Training Course
PySpark Certification T...
DataFrames and Spark SQL > Presentation
Course Content
Presentation
Class 6 Recording
Case Study I - Perfor...
Case Study II - Perfor...
Case Study Solution
In-Class Demo - Inpu...
Dataset
Dataset
Spark Sql Dataframe...
Personal Library
Presentation
88 / 93
Use Case Solution
You can download the solution for the above Use Case from here. Make sure that you are reading the given file and performing required operations on it
edureka!
Copyright © edureka and/or its affiliates. All rights reserved.
31°C
Haze
Search
ENG
IN
23:20
01-06-2023

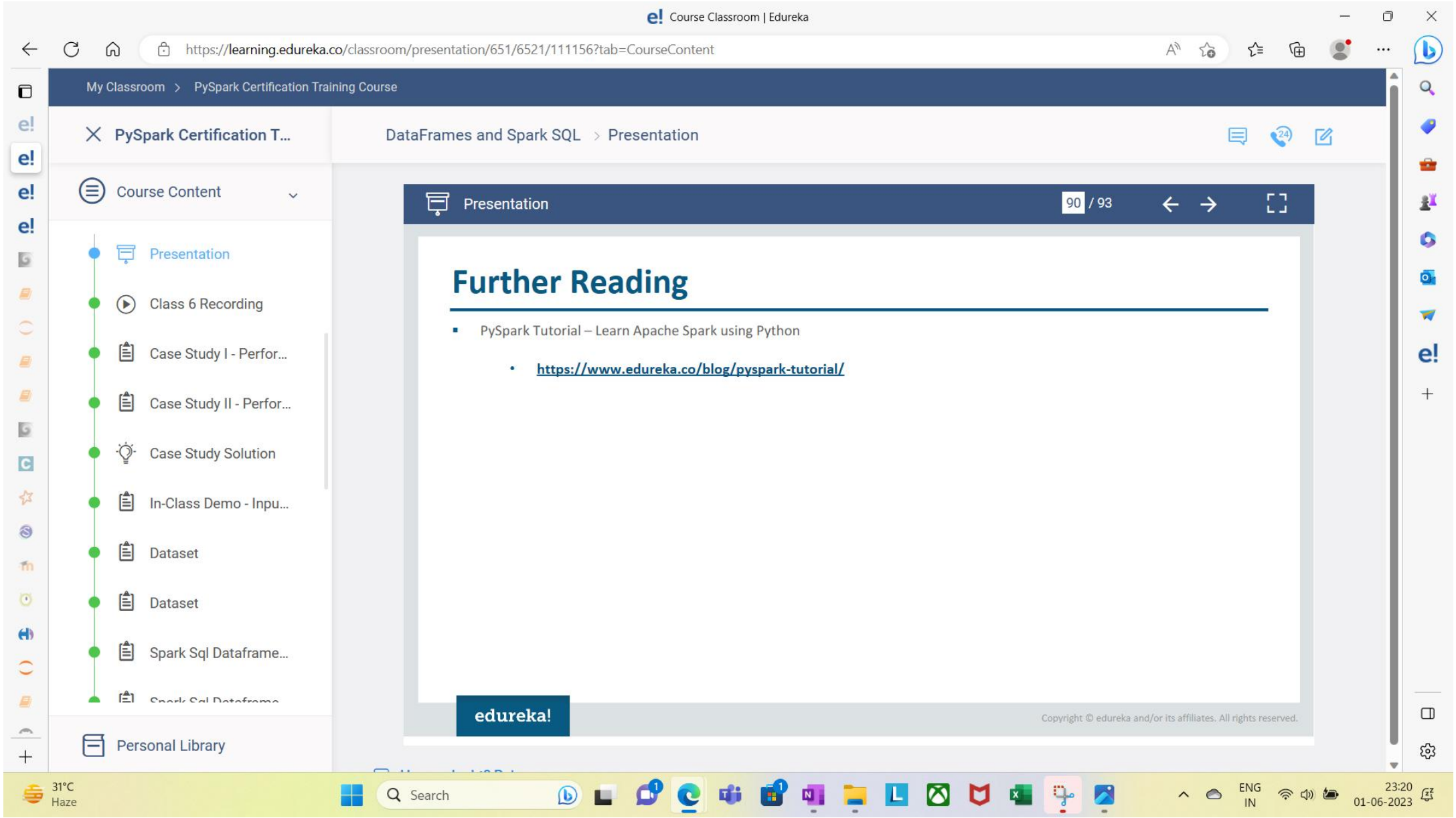
Presentation 89 / 93

Summary





edureka!


Course Classroom | Edureka
https://learning.edureka.co/classroom/presentation/651/6521/111156?tab=CourseContent
My Classroom > PySpark Certification Training Course
PySpark Certification T...
DataFrames and Spark SQL > Presentation
Course Content
Presentation
Class 6 Recording
Case Study I - Perfor...
Case Study II - Perfor...
Case Study Solution
In-Class Demo - Inpu...
Dataset
Dataset
Spark Sql Dataframe...
Personal Library
Presentation
90 / 93
Further Reading
PySpark Tutorial – Learn Apache Spark using Python
https://www.edureka.co/blog/pyspark-tutorial/
edureka!
Copyright © edureka and/or its affiliates. All rights reserved.
31°C
Haze
Search
ENG
IN
23:20
01-06-2023