Structured API's session-5 :-

rdd vs dataFrame vs datasets

rdd

when we deal with rdd. we deal with low level code

map
filter
flatmap
reduceBykey

This low level code is not developers friendly.
rdd lacks some of the basic optimizations.

dataFrames :-

Spark 1.3 version
higher level constructs which make's the developers life easy.

challanges with dataFrames :-

1. DataFrames do not offer strongly typed code.
   type Error won't be caught at compile time rather we surprise at runtime
2. Developers felt that there flexibility (like to write has become limited. own anonymous function)

There was a way that the dataFrames can be converted to rdd.

df.rdd ⇒ whenever we want more flexibility and type sabety.

1. This conversion From Dataframes to rdd is not Seamless.
2. If we work with raw rdd by converting dataFrame to rdd. we will miss out on some of the major optimizations.

Dataset API :-

Dataset came into picture to address these challanges of DataFrame.

Spark 1.6 version.

1. Compile time safety
   with datasets type Errors are caught at compiler level.
2. we get more flexibility in terms of using lower level code.

Conversion from Dataframes to datasets is seamless.

we won't lose of any of the optimizations.

Before Spark 2 Both dataframes and datasets are 2 different things.

In spark 2 they merged these 2 into a single unified spark dataset API (Structured API)

→ DataFrame is nothing but a Dataset[Row]
Row is nothing but a generic type which will be bound at runtime.

In case of Dataframes the datatypes are bound at runtime

however Dataset [Employee]

the type will be bound at complile time.

Dataset [Row] → dataframe (type Error are caught at runtime)

Dataset [Employee] → dataset (Complile time type safety)

Q How to convert dataFrame to a dataset?

If we replace generic row with specific object then it becomes a dataset.