16. Databricks | Spark | Pyspark | Bad Records Handling | Permissive;DropMalformed;FailFast

5 seconds

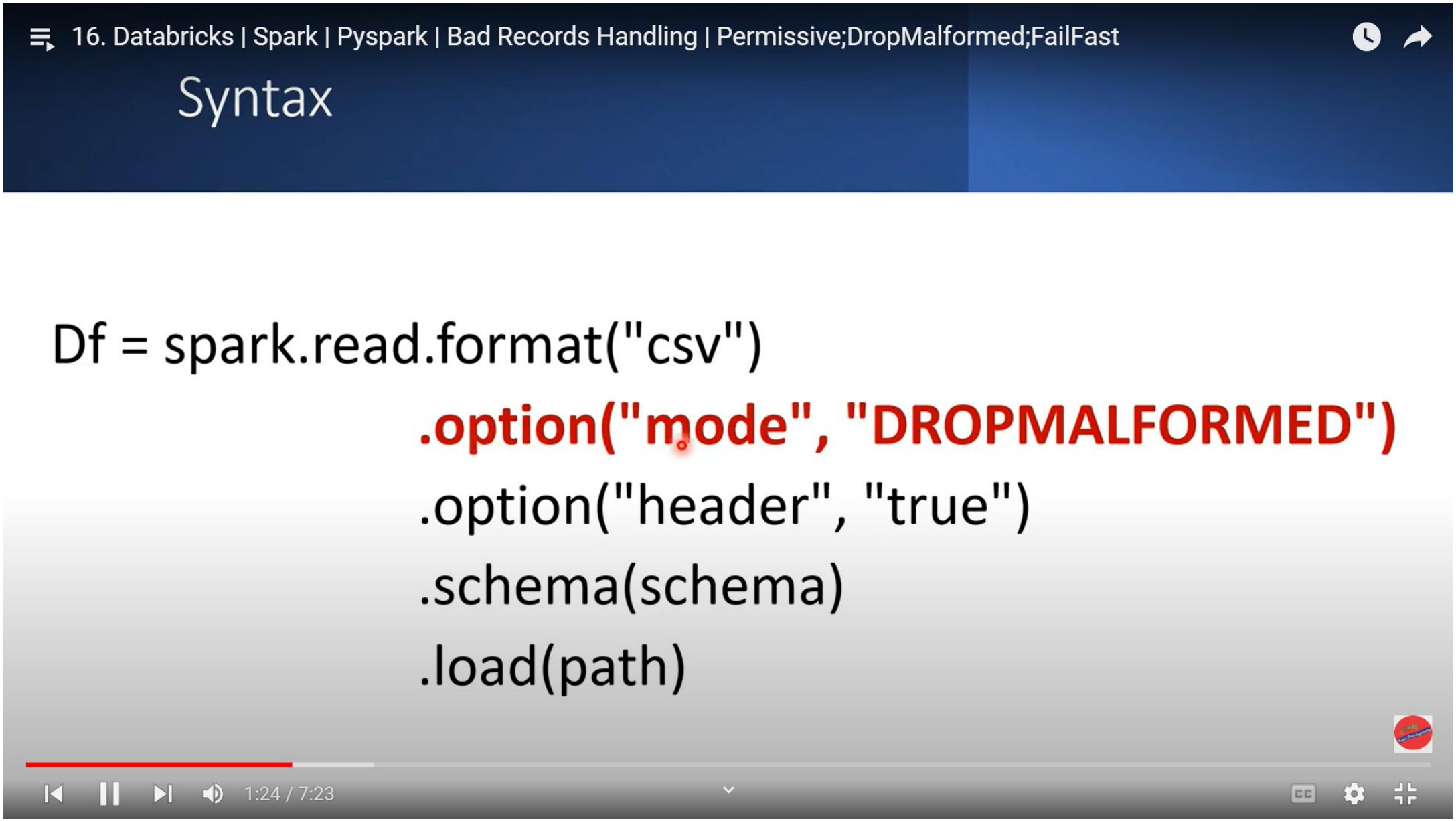
Raja's Data Engineering

Databricks| Pyspark|

Bad Records handling

0:06 / 7:23

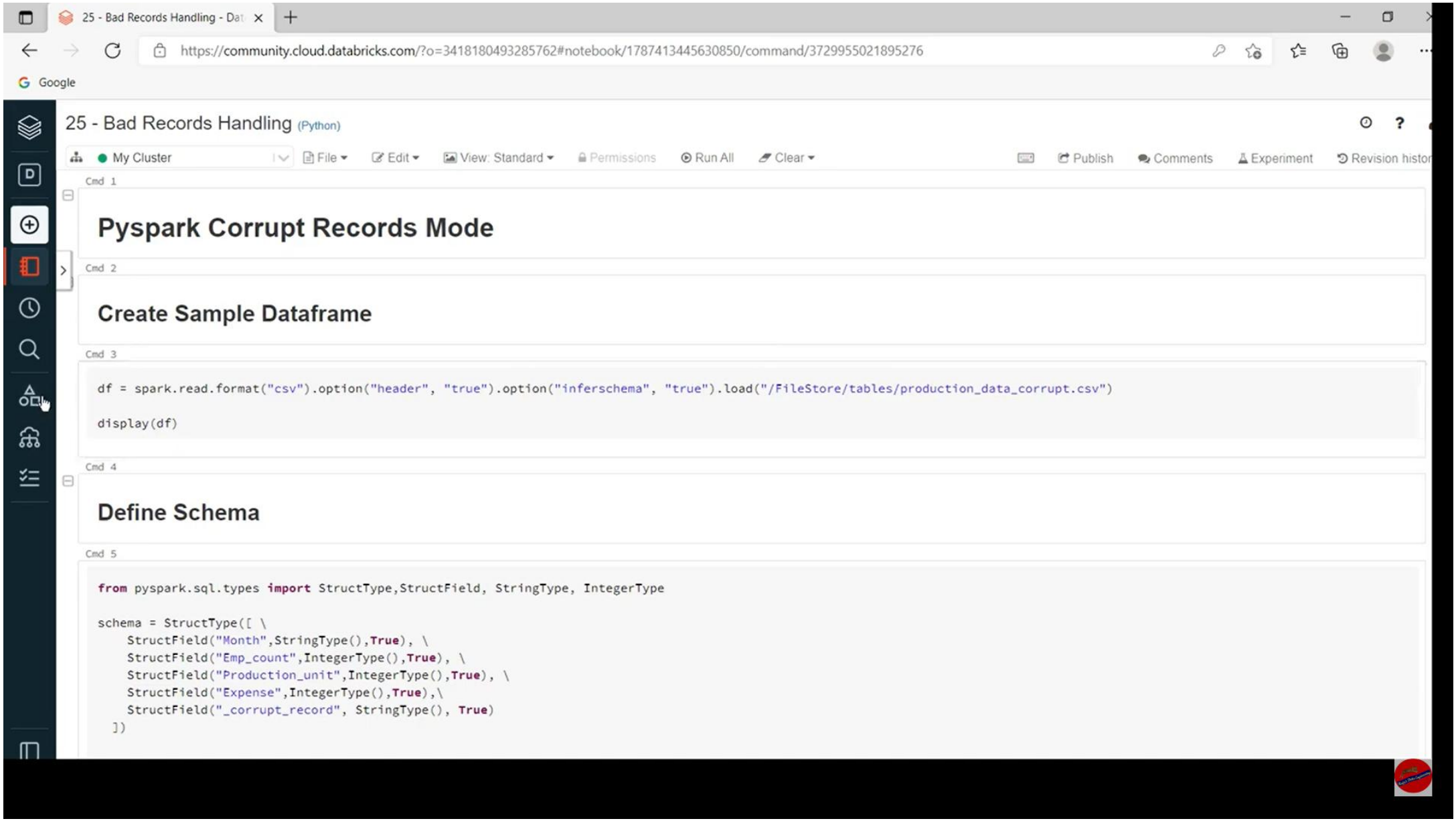

16. Databricks | Spark | Pyspark | Bad Records Handling | Permissive;DropMalformed;FailFast
Corrupt records handling
1. Permissive - Include Corrupt record in separate column
2. Drop Malformed – Ignore Corrupt Records
3. Fail Fast - Throw Exception if Corrupt Record
0:24 / 7:23

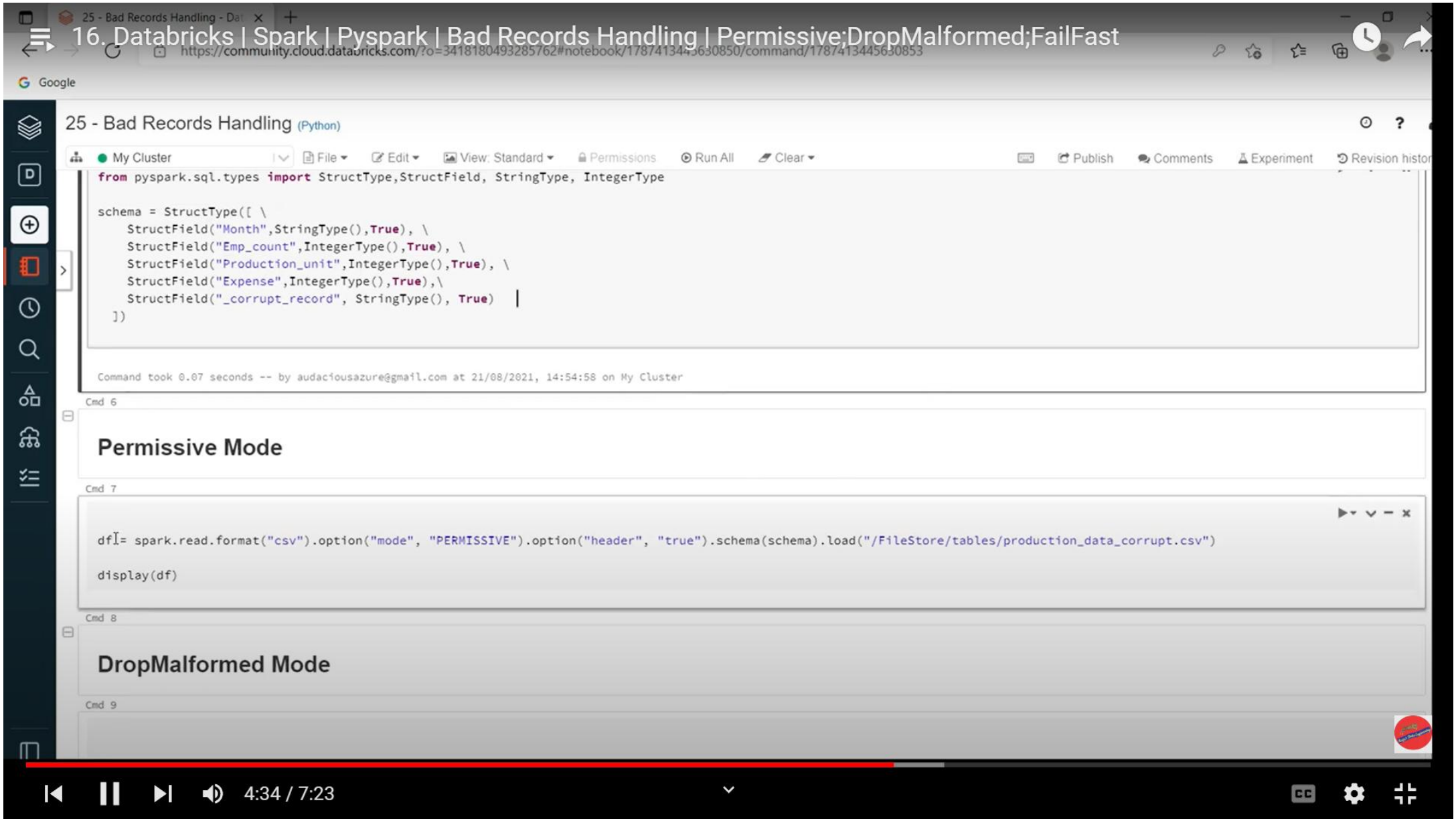

16. Databricks | Spark | Pyspark | Bad Records Handling | Permissive;DropMalformed;FailFast
Syntax
Df = spark.read.format("csv")
.option("mode", "DROPMALFORMED")
.option("header", "true")
.schema(schema)
.load(path)
1:24 / 7:23

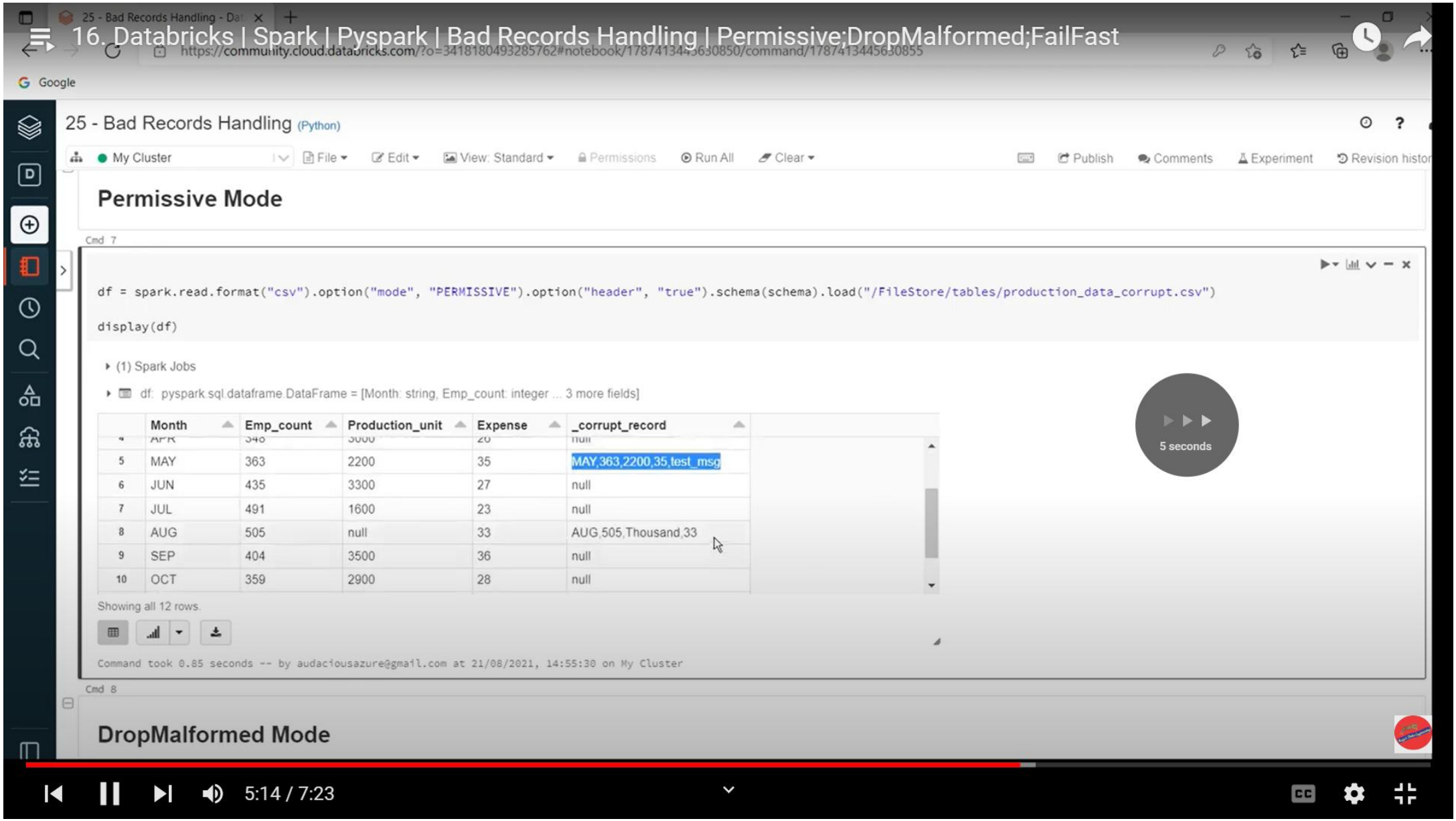
production_data_corrupt - Notepad

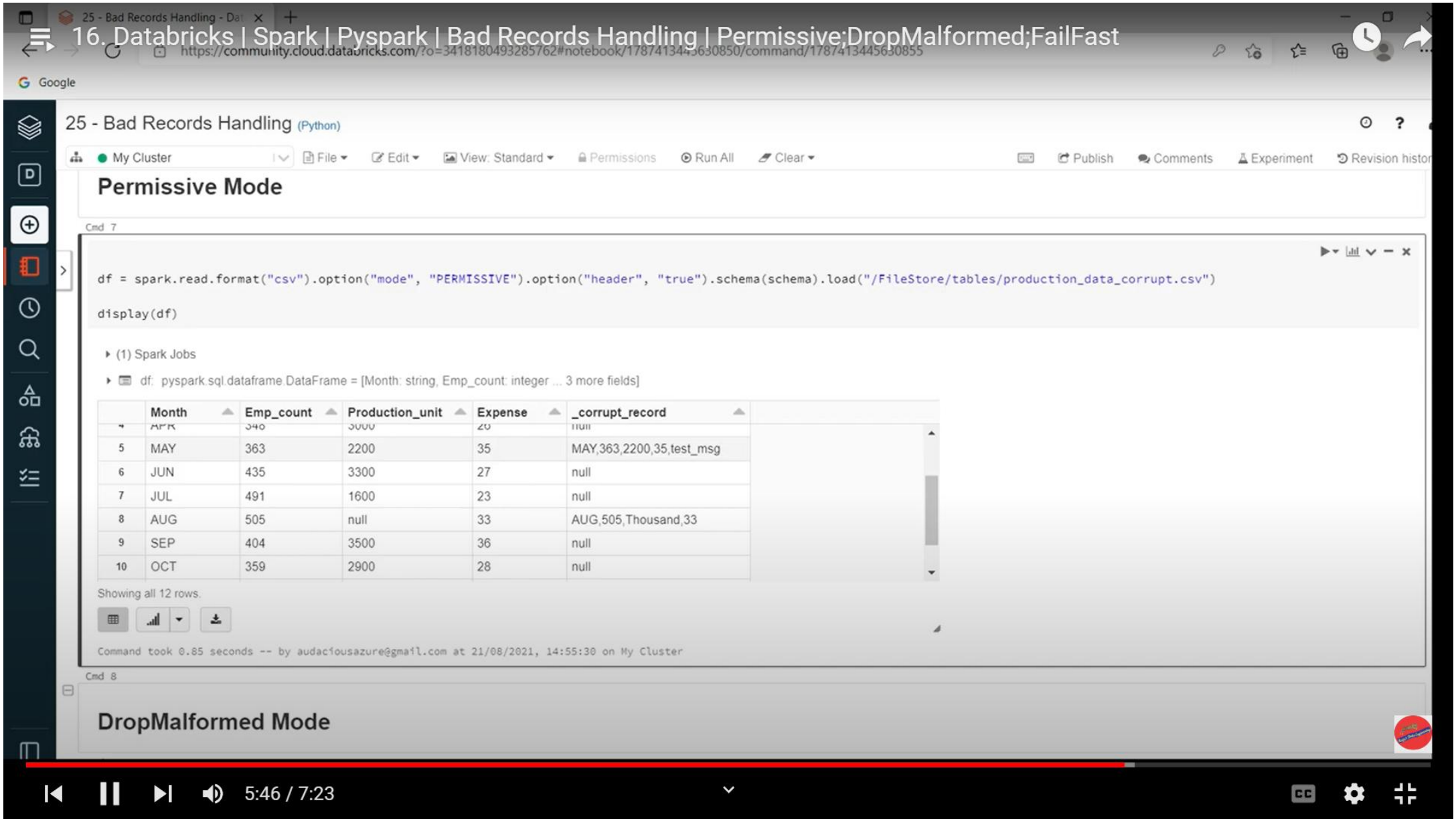
File Edit Format View Help

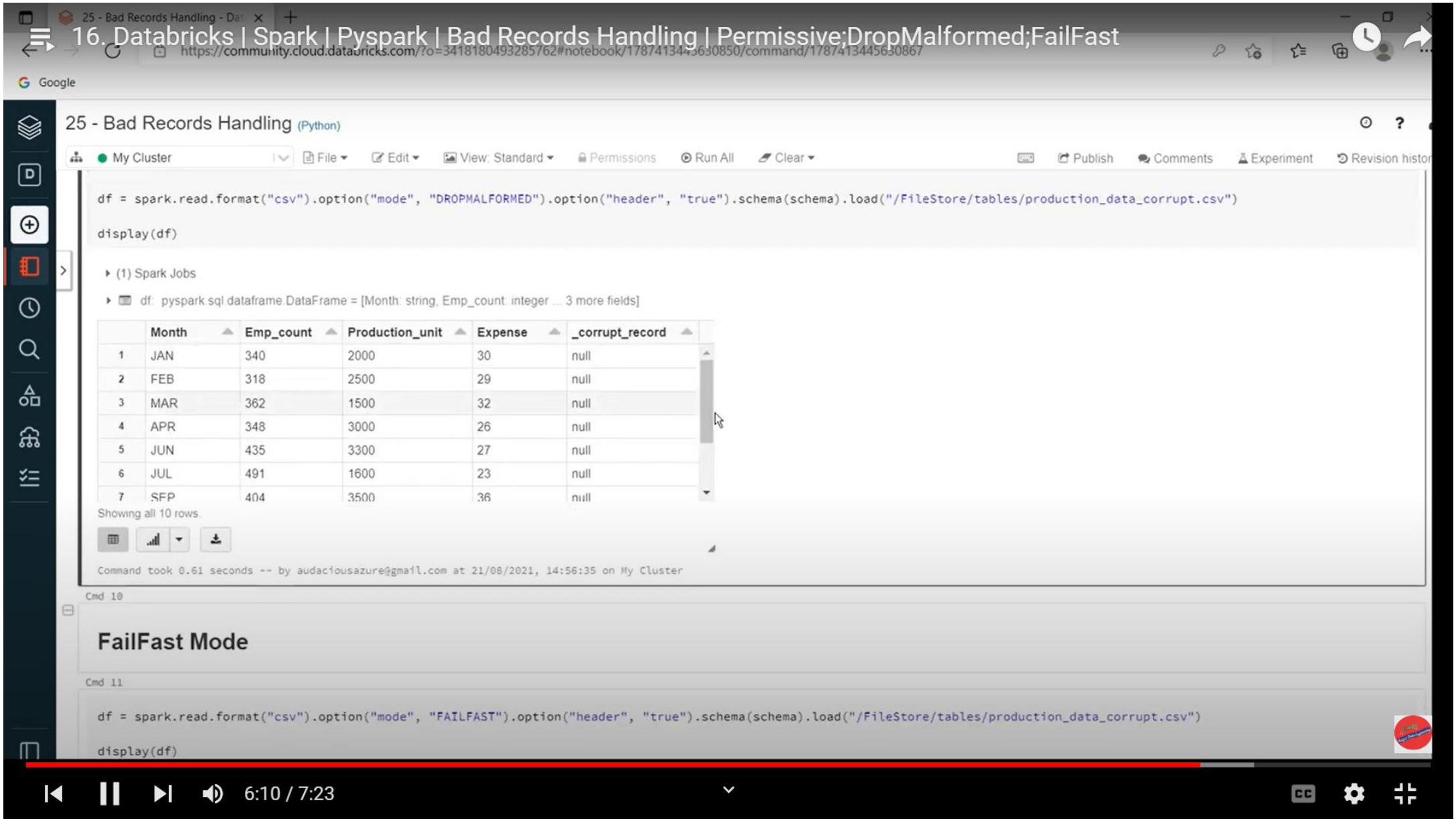
```
Month,Emp_count,Production_unit,Expense
JAN,340,2000,30,
FEB,318,2500,29,
MAR,362,1500,32,
APR,348,3000,26,
MAY,363,2200,35,test_msg
JUN,435,3300,27,
JUL,491,1600,23,
AUG,505,Thousand,33,
SEP,404,3500,36,
OCT,359,2900,28,
NOV,310,4000,25,
DEC,337,3400,31,
```


25 - Bad Records Handling - Dat
https://community.cloud.databricks.com/?o=3418180493285762#notebook/1787413445630850/command/3729955021895276
Google
25 - Bad Records Handling (Python)
My Cluster
File
Edit
View: Standard
Permissions
Run All
Clear
Publish
Comments
Experiment
Cmd 1
Pyspark Corrupt Records Mode
Cmd 2
Create Sample Dataframe
Cmd 3
df = spark.read.format("csv").option("header", "true").option("inferschema", "true").load("/FileStore/tables/production_data_corrupt.csv")
display(df)
Cmd 4
Define Schema
Cmd 5
from pyspark.sql.types import StructType,StructField, StringType, IntegerType
schema = StructType([\
StructField("Month",StringType(),True), \
StructField("Emp_count",IntegerType(),True), \
StructField("Production_unit",IntegerType(),True), \
StructField("Expense",IntegerType(),True),\
StructField("_corrupt_record", StringType(), True)
])


16. Databricks | Spark | Pyspark | Bad Records Handling | Permissive;DropMalformed;FailFast
25 - Bad Records Handling - Da
https://community.cloud.databricks.com/?o=3418180493285762#notebook/1787413445630850/command/1787413445630853
Google
25 - Bad Records Handling (Python)
My Cluster
File
Edit
View: Standard
Permissions
Run All
Clear
Publish
Comments
Experiment
Revision histor
from pyspark.sql.types import StructType,StructField, StringType, IntegerType
schema = StructType([\
StructField("Month",StringType(),True), \
StructField("Emp_count",IntegerType(),True), \
StructField("Production_unit",IntegerType(),True), \
StructField("Expense",IntegerType(),True),\
StructField("_corrupt_record", StringType(), True)
])
Command took 0.07 seconds -- by audaciousazure@gmail.com at 21/08/2021, 14:54:58 on My Cluster
Cmd 6
Permissive Mode
Cmd 7
df= spark.read.format("csv").option("mode", "PERMISSIVE").option("header", "true").schema(schema).load("/FileStore/tables/production_data_corrupt.csv")
display(df)
Cmd 8
DropMalformed Mode
Cmd 9
4:34 / 7:23


16. Databricks | Spark | Pyspark | Bad Records Handling | Permissive;DropMalformed;FailFast
25 - Bad Records Handling (Python)
Permissive Mode
Cmd 7
df = spark.read.format("csv").option("mode", "PERMISSIVE").option("header", "true").schema(schema).load("/FileStore/tables/production_data_corrupt.csv")
display(df)
(1) Spark Jobs
df: pyspark.sql.dataframe.DataFrame = [Month: string, Emp_count: integer ... 3 more fields]
Month | Emp_count | Production_unit | Expense | _corrupt_record
5 | MAY | 363 | 2200 | 35 | MAY,363,2200,35,test_msg
6 | JUN | 435 | 3300 | 27 | null
7 | JUL | 491 | 1600 | 23 | null
8 | AUG | 505 | null | 33 | AUG,505,Thousand,33
9 | SEP | 404 | 3500 | 36 | null
10 | OCT | 359 | 2900 | 28 | null
Showing all 12 rows.
Command took 0.85 seconds -- by audaciousazure@gmail.com at 21/08/2021, 14:55:30 on My Cluster
5 seconds
Cmd 8
DropMalformed Mode
5:14 / 7:23


16. Databricks | Spark | Pyspark | Bad Records Handling | Permissive;DropMalformed;FailFast
25 - Bad Records Handling (Python)
Permissive Mode
Cmd 7
df = spark.read.format("csv").option("mode", "PERMISSIVE").option("header", "true").schema(schema).load("/FileStore/tables/production_data_corrupt.csv")
display(df)
(1) Spark Jobs
df: pyspark.sql.dataframe.DataFrame = [Month: string, Emp_count: integer ... 3 more fields]
Month | Emp_count | Production_unit | Expense | _corrupt_record
5 | MAY | 363 | 2200 | 35 | MAY,363,2200,35,test_msg
6 | JUN | 435 | 3300 | 27 | null
7 | JUL | 491 | 1600 | 23 | null
8 | AUG | 505 | null | 33 | AUG,505,Thousand,33
9 | SEP | 404 | 3500 | 36 | null
10 | OCT | 359 | 2900 | 28 | null
Showing all 12 rows.
Command took 0.85 seconds -- by audaciousazure@gmail.com at 21/08/2021, 14:55:30 on My Cluster
Cmd 8
DropMalformed Mode
5:46 / 7:23


16. Databricks | Spark | Pyspark | Bad Records Handling | Permissive;DropMalformed;FailFast
25 - Bad Records Handling (Python)
My Cluster
File
Edit
View: Standard
Permissions
Run All
Clear
Publish
Comments
Experiment
df = spark.read.format("csv").option("mode", "DROPMALFORMED").option("header", "true").schema(schema).load("/FileStore/tables/production_data_corrupt.csv")
display(df)
(1) Spark Jobs
df: pyspark.sql.dataframe.DataFrame = [Month: string, Emp_count: integer ... 3 more fields]
Month Emp_count Production_unit Expense _corrupt_record
1 JAN 340 2000 30 null
2 FEB 318 2500 29 null
3 MAR 362 1500 32 null
4 APR 348 3000 26 null
5 JUN 435 3300 27 null
6 JUL 491 1600 23 null
Showing all 10 rows.
Command took 0.61 seconds -- by audaciousazure@gmail.com at 21/08/2021, 14:56:35 on My Cluster
Cmd 10
FailFast Mode
Cmd 11
df = spark.read.format("csv").option("mode", "FAILFAST").option("header", "true").schema(schema).load("/FileStore/tables/production_data_corrupt.csv")
display(df)
6:10 / 7:23