34|DSE|aditi|1004|67|CCE|khushi|1005|67|DSE|abhi|1006

```
// Load the input data
val data = "34|DSE|aditi|1004|67|CCE|khushi|1005|67|DSE|abhi|1006"

// Split the data into chunks of four elements
val chunks = data.split("\\|").grouped(4).toList

// Convert data to DataFrame
val df = spark.createDataFrame(chunks.map(arr => (arr(0), arr(1), arr(2), arr(3))))
.toDF("marks", "department", "name", "regno")

// Filter DSE students and display their details
val dseStudents = df.filter($"department" === "DSE")
dseStudents.show()
```

CODE FOR PYSPARK:

```
import org.apache.spark.sql.{SparkSession, Row}
import org.apache.spark.sql.types._

// Create a SparkSession
val spark = SparkSession.builder()
.appName("StudentDetails")
.getOrCreate()

// Input data
val input = "34|DSE|aditi|1004|67|CCE|khushi|1005|67|DSE|abhi|1006"

// Split the input string by "|"
val data = input.split("\\|")

// Group data into chunks of four elements and add delimiter
val chunks = data.grouped(4).toList

// Define schema for DataFrame
val schema = StructType(Seq(
StructField("marks", StringType),
StructField("department", StringType),
StructField("name", StringType),
StructField("regno", StringType)
))
```

```
// Create RDD of Rows
val rowsRDD = spark.sparkContext.parallelize(chunks.map { case Array(marks, department,
name, regno) =>
Row(marks, department, name, regno)
})

// Create DataFrame from RDD and schema
val df = spark.createDataFrame(rowsRDD, schema)

// Filter DSE students
val dseStudents = df.filter($"department" === "DSE")

// Show DSE student details
dseStudents.show(false)

// Convert DataFrame to CSV format
val csvFormat = dseStudents.select("regno", "name",
"marks").collect().map(_.mkString(",")).mkString("\n")

// Display CSV format
println("regno,name,marks")
println(csvFormat)
```