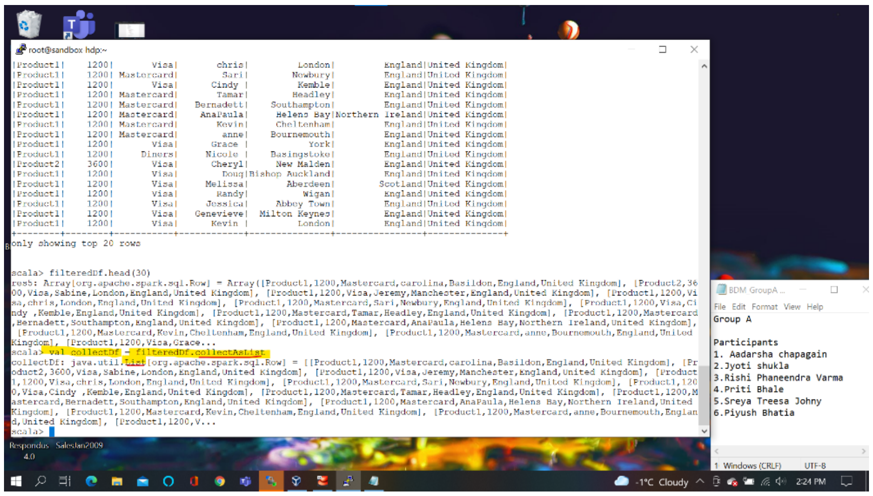
Group A
Lab 3D
SparkSql

Participants:
1. Aadarsha chapagain
2.Jyoti shukla
3.Rishi Phaneendra Varma
4.Priti Bhale
5.Sreya Treesa Johny
6.Piyush Bhatia

Part1: choose any 5 Data Frame operation

1. Collect As List

   val collectDf = filteredDf.collectAsList



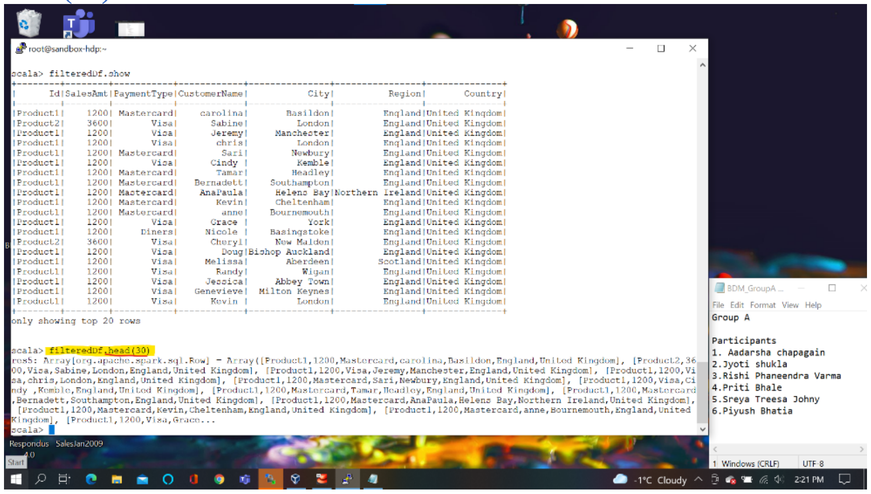
```
|Product1|    1200|       Visa|     chris|        London|          England|United Kingdom|
|Product1|    1200| Mastercard|      Sari|       Newbury|          England|United Kingdom|
|Product1|    1200|       Visa|    Cindy |        Kemble|          England|United Kingdom|
|Product1|    1200| Mastercard|     Tamar|       Headley|          England|United Kingdom|
|Product1|    1200| Mastercard| Bernadett|   Southampton|          England|United Kingdom|
|Product1|    1200| Mastercard|  AnaPaula|    Helens Bay|Northern Ireland|United Kingdom|
|Product1|    1200| Mastercard|     Kevin|    Cheltenham|          England|United Kingdom|
|Product1|    1200| Mastercard|      anne|   Bournemouth|          England|United Kingdom|
|Product1|    1200|       Visa|    Grace |          York|          England|United Kingdom|
|Product1|    1200|     Diners|   Nicole |   Basingstoke|          England|United Kingdom|
|Product2|    3600|       Visa|    Cheryl|    New Malden|          England|United Kingdom|
|Product1|    1200|       Visa|      Doug|Bishop Auckland|         England|United Kingdom|
|Product1|    1200|       Visa|   Melissa|      Aberdeen|         Scotland|United Kingdom|
|Product1|    1200|       Visa|     Randy|         Wigan|          England|United Kingdom|
|Product1|    1200|       Visa|   Jessica|    Abbey Town|          England|United Kingdom|
|Product1|    1200|       Visa| Genevieve| Milton Keynes|          England|United Kingdom|
|Product1|    1200|       Visa|    Kevin |        London|          England|United Kingdom|
+--------+--------+-----------+----------+--------------+-----------------+--------------+
only showing top 20 rows


scala> filteredDf.head(30)
res5: Array[org.apache.spark.sql.Row] = Array([Product1,1200,Mastercard,carolina,Basildon,England,United Kingdom], [Product2,36
00,Visa,Sabine,London,England,United Kingdom], [Product1,1200,Visa,Jeremy,Manchester,England,United Kingdom], [Product1,1200,Vi
sa,chris,London,England,United Kingdom], [Product1,1200,Mastercard,Sari,Newbury,England,United Kingdom], [Product1,1200,Visa,Ci
ndy ,Kemble,England,United Kingdom], [Product1,1200,Mastercard,Tamar,Headley,England,United Kingdom], [Product1,1200,Mastercard
,Bernadett,Southampton,England,United Kingdom], [Product1,1200,Mastercard,AnaPaula,Helens Bay,Northern Ireland,United Kingdom],
 [Product1,1200,Mastercard,Kevin,Cheltenham,England,United Kingdom], [Product1,1200,Mastercard,anne,Bournemouth,England,United
Kingdom], [Product1,1200,Visa,Grace...
scala> val collectDf = filteredDf.collectAsList
collectDf: java.util.List[org.apache.spark.sql.Row] = [[Product1,1200,Mastercard,carolina,Basildon,England,United Kingdom], [Pr
oduct2,3600,Visa,Sabine,London,England,United Kingdom], [Product1,1200,Visa,Jeremy,Manchester,England,United Kingdom], [Product
1,1200,Visa,chris,London,England,United Kingdom], [Product1,1200,Mastercard,Sari,Newbury,England,United Kingdom], [Product1,120
0,Visa,Cindy ,Kemble,England,United Kingdom], [Product1,1200,Mastercard,Tamar,Headley,England,United Kingdom], [Product1,1200,M
astercard,Bernadett,Southampton,England,United Kingdom], [Product1,1200,Mastercard,AnaPaula,Helens Bay,Northern Ireland,United
Kingdom], [Product1,1200,Mastercard,Kevin,Cheltenham,England,United Kingdom], [Product1,1200,Mastercard,anne,Bournemouth,Englan
d,United Kingdom], [Product1,1200,V...
scala>
```

```
Group A

Participants
1. Aadarsha chapagain
2.Jyoti shukla
3.Rishi Phaneendra Varma
4.Priti Bhale
5.Sreya Treesa Johny
6.Piyush Bhatia
```

2. Head

   Head is used to show top specified elements

   filteredDf.head(30)



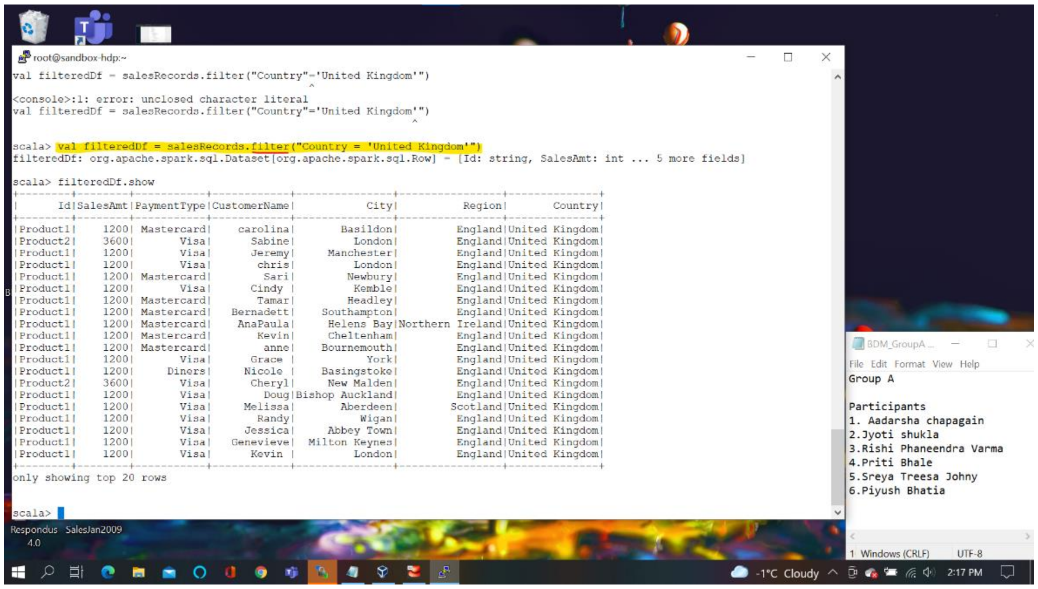
```
scala> filteredDf.show
+--------+--------+-----------+------------+---------------+-----------------+--------------+
|      Id|SalesAmt|PaymentType|CustomerName|           City|           Region|       Country|
+--------+--------+-----------+------------+---------------+-----------------+--------------+
|Product1|    1200| Mastercard|    carolina|       Basildon|          England|United Kingdom|
|Product2|    3600|       Visa|      Sabine|         London|          England|United Kingdom|
|Product1|    1200|       Visa|      Jeremy|     Manchester|          England|United Kingdom|
|Product1|    1200|       Visa|       chris|         London|          England|United Kingdom|
|Product1|    1200| Mastercard|        Sari|        Newbury|          England|United Kingdom|
|Product1|    1200|       Visa|      Cindy |         Kemble|          England|United Kingdom|
|Product1|    1200| Mastercard|       Tamar|        Headley|          England|United Kingdom|
|Product1|    1200| Mastercard|   Bernadett|    Southampton|          England|United Kingdom|
|Product1|    1200| Mastercard|    AnaPaula|     Helens Bay| Northern Ireland|United Kingdom|
|Product1|    1200| Mastercard|       Kevin|     Cheltenham|          England|United Kingdom|
|Product1|    1200| Mastercard|        anne|    Bournemouth|          England|United Kingdom|
|Product1|    1200|       Visa|      Grace |           York|          England|United Kingdom|
|Product1|    1200|     Diners|     Nicole |    Basingstoke|          England|United Kingdom|
|Product2|    3600|       Visa|      Cheryl|     New Malden|          England|United Kingdom|
|Product1|    1200|       Visa|        Doug|Bishop Auckland|          England|United Kingdom|
|Product1|    1200|       Visa|     Melissa|       Aberdeen|         Scotland|United Kingdom|
|Product1|    1200|       Visa|       Randy|          Wigan|          England|United Kingdom|
|Product1|    1200|       Visa|     Jessica|     Abbey Town|          England|United Kingdom|
|Product1|    1200|       Visa|   Genevieve|  Milton Keynes|          England|United Kingdom|
|Product1|    1200|       Visa|      Kevin |         London|          England|United Kingdom|
+--------+--------+-----------+------------+---------------+-----------------+--------------+
only showing top 20 rows


scala> filteredDf.head(30)
res5: Array[org.apache.spark.sql.Row] = Array([Product1,1200,Mastercard,carolina,Basildon,England,United Kingdom], [Product2,36
00,Visa,Sabine,London,England,United Kingdom], [Product1,1200,Visa,Jeremy,Manchester,England,United Kingdom], [Product1,1200,Vi
sa,chris,London,England,United Kingdom], [Product1,1200,Mastercard,Sari,Newbury,England,United Kingdom], [Product1,1200,Visa,Ci
ndy ,Kemble,England,United Kingdom], [Product1,1200,Mastercard,Tamar,Headley,England,United Kingdom], [Product1,1200,Mastercard
,Bernadett,Southampton,England,United Kingdom], [Product1,1200,Mastercard,AnaPaula,Helens Bay,Northern Ireland,United Kingdom],
 [Product1,1200,Mastercard,Kevin,Cheltenham,England,United Kingdom], [Product1,1200,Mastercard,anne,Bournemouth,England,United
Kingdom], [Product1,1200,Visa,Grace...
scala>
```

```
Group A

Participants
1. Aadarsha chapagain
2.Jyoti shukla
3.Rishi Phaneendra Varma
4.Priti Bhale
5.Sreya Treesa Johny
6.Piyush Bhatia
```

3. Filter

Filter is used to filter according to given criteria it is similar to where clause

Val filteresDf = salesRecords.filter(“Country = ”united kingdom”)

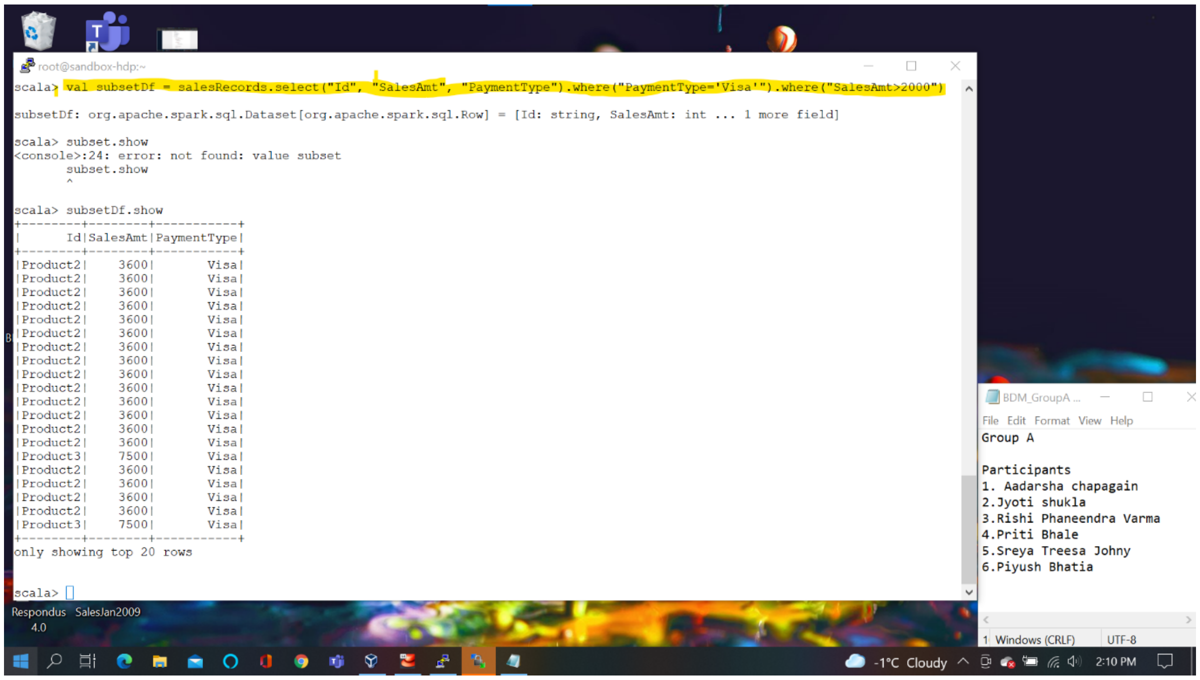


```
val filteredDf - salesRecords.filter("Country"-'United Kingdom'")
                                               ^
<console>:1: error: unclosed character literal
val filteredDf = salesRecords.filter("Country"='United Kingdom'")
                                                               ^

scala> val filteredDf = salesRecords.filter("Country = 'United Kingdom'")
filteredDf: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] - [Id: string, SalesAmt: int ... 5 more fields]

scala> filteredDf.show
+--------+--------+-----------+------------+----------------+----------------+--------------+
|      Id|SalesAmt|PaymentType|CustomerName|            City|          Region|       Country|
+--------+--------+-----------+------------+----------------+----------------+--------------+
|Product1|    1200| Mastercard|    carolina|        Basildon|         England|United Kingdom|
|Product2|    3600|       Visa|      Sabine|          London|         England|United Kingdom|
|Product1|    1200|       Visa|      Jeremy|      Manchester|         England|United Kingdom|
|Product1|    1200|       Visa|       chris|          London|         England|United Kingdom|
|Product1|    1200| Mastercard|        Sari|         Newbury|         England|United Kingdom|
|Product1|    1200|       Visa|      Cindy |          Kemble|         England|United Kingdom|
|Product1|    1200| Mastercard|       Tamar|         Headley|         England|United Kingdom|
|Product1|    1200| Mastercard|   Bernadett|     Southampton|         England|United Kingdom|
|Product1|    1200| Mastercard|    AnaPaula|      Helens Bay|Northern Ireland|United Kingdom|
|Product1|    1200| Mastercard|       Kevin|      Cheltenham|         England|United Kingdom|
|Product1|    1200| Mastercard|        anne|     Bournemouth|         England|United Kingdom|
|Product1|    1200|       Visa|      Grace |            York|         England|United Kingdom|
|Product1|    1200|     Diners|     Nicole |      Basingstoke|         England|United Kingdom|
|Product2|    3600|       Visa|      Cheryl|      New Malden|         England|United Kingdom|
|Product1|    1200|       Visa|        Doug|  Bishop Auckland|         England|United Kingdom|
|Product1|    1200|       Visa|     Melissa|        Aberdeen|        Scotland|United Kingdom|
|Product1|    1200|       Visa|       Randy|           Wigan|         England|United Kingdom|
|Product1|    1200|       Visa|     Jessica|      Abbey Town|         England|United Kingdom|
|Product1|    1200|       Visa|   Genevieve|   Milton Keynes|         England|United Kingdom|
|Product1|    1200|       Visa|      Kevin |          London|         England|United Kingdom|
+--------+--------+-----------+------------+----------------+----------------+--------------+
only showing top 20 rows

scala>
```

```
Group A

Participants
1. Aadarsha chapagain
2.Jyoti shukla
3.Rishi Phaneendra Varma
4.Priti Bhale
5.Sreya Treesa Johny
6.Piyush Bhatia
```

4. Select

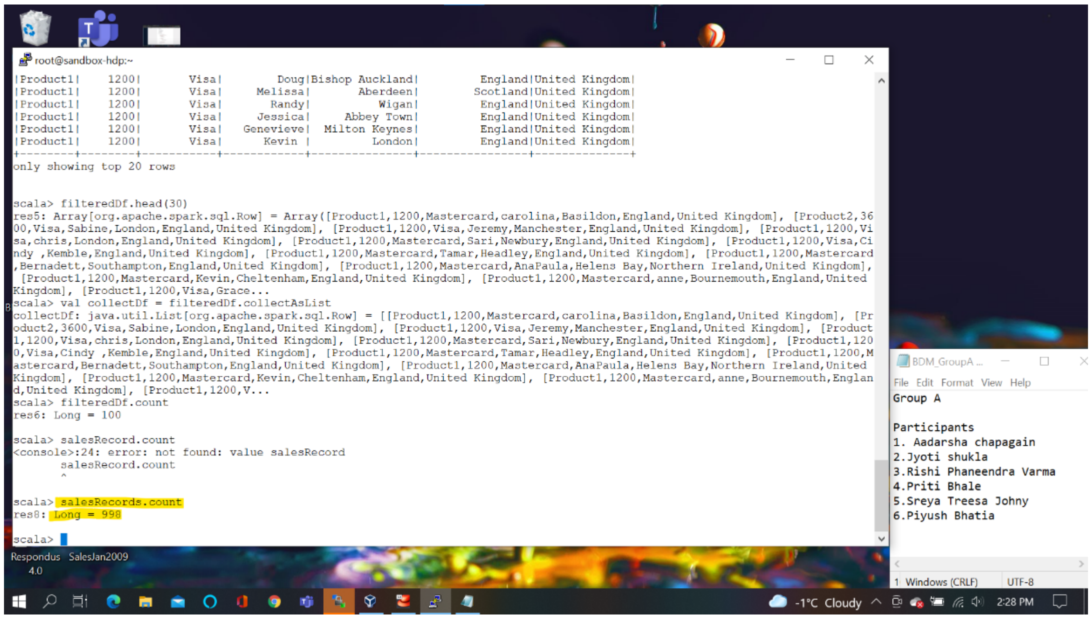
Select is used for projection



```
scala> val subsetDf = salesRecords.select("Id", "SalesAmt", "PaymentType").where("PaymentType='Visa'").where("SalesAmt>2000")

subsetDf: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [Id: string, SalesAmt: int ... 1 more field]

scala> subset.show
<console>:24: error: not found: value subset
       subset.show
       ^

scala> subsetDf.show
+--------+--------+-----------+
|      Id|SalesAmt|PaymentType|
+--------+--------+-----------+
|Product2|    3600|       Visa|
|Product2|    3600|       Visa|
|Product2|    3600|       Visa|
|Product2|    3600|       Visa|
|Product2|    3600|       Visa|
|Product2|    3600|       Visa|
|Product2|    3600|       Visa|
|Product2|    3600|       Visa|
|Product2|    3600|       Visa|
|Product2|    3600|       Visa|
|Product2|    3600|       Visa|
|Product2|    3600|       Visa|
|Product2|    3600|       Visa|
|Product2|    3600|       Visa|
|Product3|    7500|       Visa|
|Product2|    3600|       Visa|
|Product2|    3600|       Visa|
|Product2|    3600|       Visa|
|Product2|    3600|       Visa|
|Product3|    7500|       Visa|
+--------+--------+-----------+
only showing top 20 rows

scala>
```

```
Group A

Participants
1. Aadarsha chapagain
2.Jyoti shukla
3.Rishi Phaneendra Varma
4.Priti Bhale
5.Sreya Treesa Johny
6.Piyush Bhatia
```

5. Count



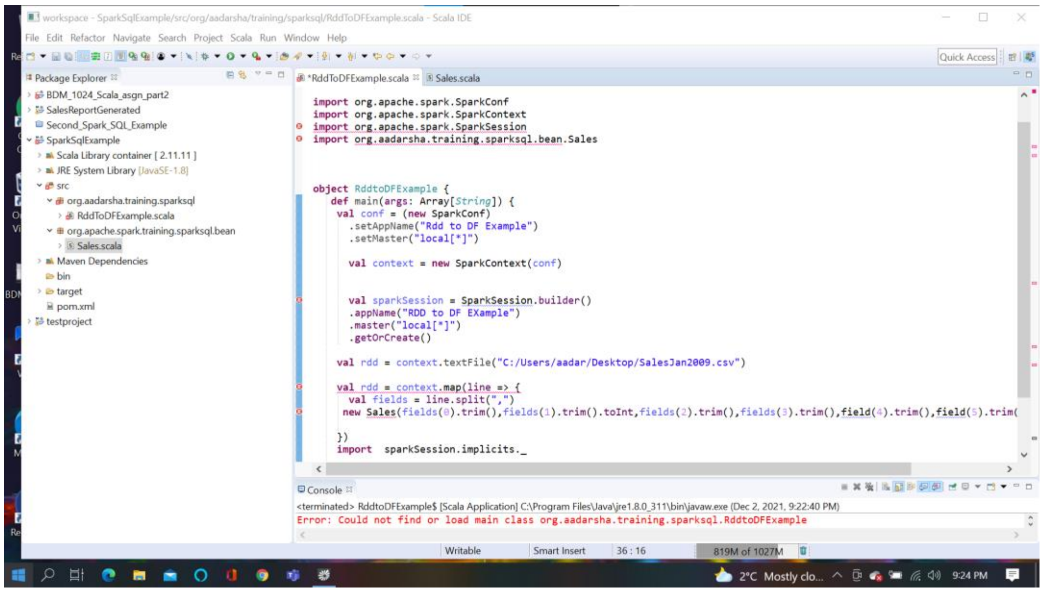
root@sandbox-hdp:~

```
|Product1|    1200|        Visa|        Doug|Bishop Auckland|          England|United Kingdom|
|Product1|    1200|        Visa|     Melissa|       Aberdeen|         Scotland|United Kingdom|
|Product1|    1200|        Visa|       Randy|          Wigan|          England|United Kingdom|
|Product1|    1200|        Visa|     Jessica|     Abbey Town|          England|United Kingdom|
|Product1|    1200|        Visa|   Genevieve|  Milton Keynes|          England|United Kingdom|
|Product1|    1200|        Visa|       Kevin |         London|          England|United Kingdom|
+--------+--------+-----------+------------+---------------+-----------------+--------------+
only showing top 20 rows


scala> filteredDf.head(30)
res5: Array[org.apache.spark.sql.Row] = Array([Product1,1200,Mastercard,carolina,Basildon,England,United Kingdom], [Product2,36
00,Visa,Sabine,London,England,United Kingdom], [Product1,1200,Visa,Jeremy,Manchester,England,United Kingdom], [Product1,1200,Vi
sa,chris,London,England,United Kingdom], [Product1,1200,Mastercard,Sari,Newbury,England,United Kingdom], [Product1,1200,Visa,Ci
ndy ,Kemble,England,United Kingdom], [Product1,1200,Mastercard,Tamar,Headley,England,United Kingdom], [Product1,1200,Mastercard
,Bernadett,Southampton,England,United Kingdom], [Product1,1200,Mastercard,AnaPaula,Helens Bay,Northern Ireland,United Kingdom],
 [Product1,1200,Mastercard,Kevin,Cheltenham,England,United Kingdom], [Product1,1200,Mastercard,anne,Bournemouth,England,United
Kingdom], [Product1,1200,Visa,Grace...
scala> val collectDf = filteredDf.collectAsList
collectDf: java.util.List[org.apache.spark.sql.Row] = [[Product1,1200,Mastercard,carolina,Basildon,England,United Kingdom], [Pr
oduct2,3600,Visa,Sabine,London,England,United Kingdom], [Product1,1200,Visa,Jeremy,Manchester,England,United Kingdom], [Product
1,1200,Visa,chris,London,England,United Kingdom], [Product1,1200,Mastercard,Sari,Newbury,England,United Kingdom], [Product1,120
0,Visa,Cindy ,Kemble,England,United Kingdom], [Product1,1200,Mastercard,Tamar,Headley,England,United Kingdom], [Product1,1200,M
astercard,Bernadett,Southampton,England,United Kingdom], [Product1,1200,Mastercard,AnaPaula,Helens Bay,Northern Ireland,United
Kingdom], [Product1,1200,Mastercard,Kevin,Cheltenham,England,United Kingdom], [Product1,1200,Mastercard,anne,Bournemouth,Englan
d,United Kingdom], [Product1,1200,V...
scala> filteredDf.count
res6: Long = 100

scala> salesRecord.count
<console>:24: error: not found: value salesRecord
       salesRecord.count
       ^

scala> salesRecords.count
res8: Long = 998

scala> 
```

BDM_GroupA

```
Group A

Participants
1. Aadarsha chapagain
2.Jyoti shukla
3.Rishi Phaneendra Varma
4.Priti Bhale
5.Sreya Treesa Johny
6.Piyush Bhatia
```

Slide 83:



workspace - SparkSqlExample/src/org/aadarsha/training/sparksql/RddToDFExample.scala - Scala IDE

Package Explorer
- BDM_1024_Scala_asgn_part2
- SalesReportGenerated
- Second_Spark_SQL_Example
- SparkSqlExample
  - Scala Library container [2.11.11]
  - JRE System Library [JavaSE-1.8]
  - src
    - org.aadarsha.training.sparksql
      - RddToDFExample.scala
    - org.apache.spark.training.sparksql.bean
      - Sales.scala
  - Maven Dependencies
  - bin
  - target
  - pom.xml
- testproject

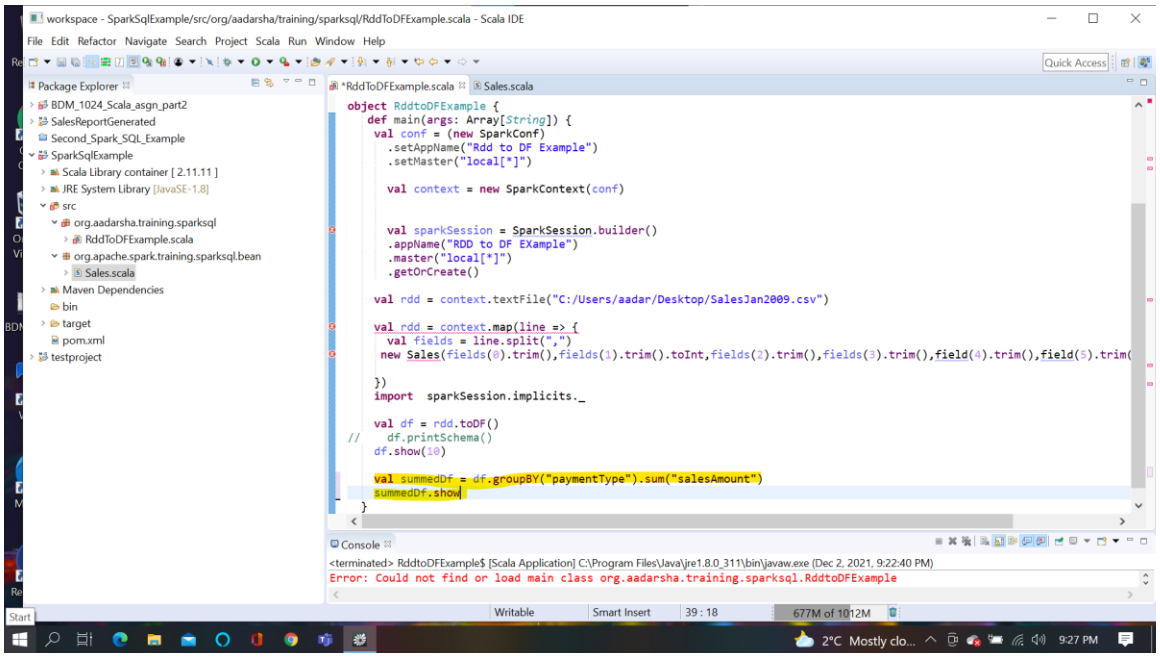
*RddToDFExample.scala | Sales.scala

```scala
import org.apache.spark.SparkConf
import org.apache.spark.SparkContext
import org.apache.spark.SparkSession
import org.aadarsha.training.sparksql.bean.Sales



object RddtoDFExample {
   def main(args: Array[String]) {
    val conf = (new SparkConf)
      .setAppName("Rdd to DF Example")
      .setMaster("local[*]")

      val context = new SparkContext(conf)


      val sparkSession = SparkSession.builder()
      .appName("RDD to DF EXample")
      .master("local[*]")
      .getOrCreate()

    val rdd = context.textFile("C:/Users/aadar/Desktop/SalesJan2009.csv")

    val rdd = context.map(line => {
      val fields = line.split(",")
     new Sales(fields(0).trim(),fields(1).trim().toInt,fields(2).trim(),fields(3).trim(),field(4).trim(),field(5).trim(

    })
    import  sparkSession.implicits._
```

Console

```
<terminated> RddtoDFExample$ [Scala Application] C:\Program Files\Java\jre1.8.0_311\bin\javaw.exe (Dec 2, 2021, 9:22:40 PM)
Error: Could not find or load main class org.aadarsha.training.sparksql.RddtoDFExample
```

Slide 85:



```
object RddtoDFExample {
  def main(args: Array[String]) {
   val conf = (new SparkConf)
     .setAppName("Rdd to DF Example")
     .setMaster("local[*]")

     val context = new SparkContext(conf)


     val sparkSession = SparkSession.builder()
     .appName("RDD to DF EXample")
     .master("local[*]")
     .getOrCreate()

   val rdd = context.textFile("C:/Users/aadar/Desktop/SalesJan2009.csv")

   val rdd = context.map(line => {
     val fields = line.split(",")
    new Sales(fields(0).trim(),fields(1).trim().toInt,fields(2).trim(),fields(3).trim(),field(4).trim(),field(5).trim(

   })
   import  sparkSession.implicits._

   val df = rdd.toDF()
//    df.printSchema()
   df.show(10)

   val summedDf = df.groupBY("paymentType").sum("salesAmount")
   summedDf.show
 }
```

```
<terminated> RddtoDFExample$ [Scala Application] C:\Program Files\Java\jre1.8.0_311\bin\javaw.exe (Dec 2, 2021, 9:22:40 PM)
Error: Could not find or load main class org.aadarsha.training.sparksql.RddtoDFExample
```