```scala
def read(name: String) = {
 spark.read
   .option("header", "true")
   .option("sep", ";")
   .option("ignoreLeadingWhiteSpace", "true")
   .option("ignoreTrailingWhiteSpace", "true")
   .csv(name)
}

// select each column and get domain
val columns = inputs.map(read)
 .flatMap(table =>
   table
     .columns
     .map(column =>
       table
         .select(column)
         .distinct()
         .repartition(numPartitions)
         .cache()))
```

```scala
// make all pairs of pairwise different columns (without same columns)
val pairs = columns
 .flatMap(column =>
   columns
     .filter(anotherColumn => anotherColumn != column)
     .map(anotherColumn =>(column, anotherColumn)))

// only keep pairs where second contains all values of first
val inds = pairs.filter(pair => pair._1.except(pair._2).isEmpty)

// select column names, group, format and print
inds
 .map(pair => (pair._1.schema(0).name, pair._2.schema(0).name))
 .groupBy(pair => pair._1)
 .map(group => (group._1, group._2.map(pair => pair._2).sorted.mkString(", ")))
 .toList.sortBy(group => group._1)
 .foreach(ind => println(s"${ind._1} < ${ind._2}"))
```