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

  // tuples of cell values and column name
  val cells = inputs.map(read).map(table =>
    table
      .flatMap(row =>
        row
          .schema
          .fieldNames.map(column =>
            (row.getAs[String](column), column)))
      .reduce((table_1, table_2) => table_1 union table_2)

    val groupedCells = cells.groupBy($"_1")
      .agg(collect_set($"_2").as($"_2"))
```

```
val inclusionLists = groupedCells
  .flatMap(cell => {
    val columns = cell.getAs[Seq[String]]("_2")
    columns.map(column => (column, columns.filter(column != _)))
  }).filter("size(_2) != 0")

val inds = inclusionLists
  .groupByKey(_. _1)
  .reduceGroups((group_1, group_2) =>
    (group_1._1, group_1._2.intersect(group_2._2)))

inds
  .sort("value")
  .map(group => (group._1, group._2._2.sorted.mkString(", ")))
  .foreach(ind => if(!ind._2.isEmpty) println(s"${ind._1} < ${ind._2}"))
```