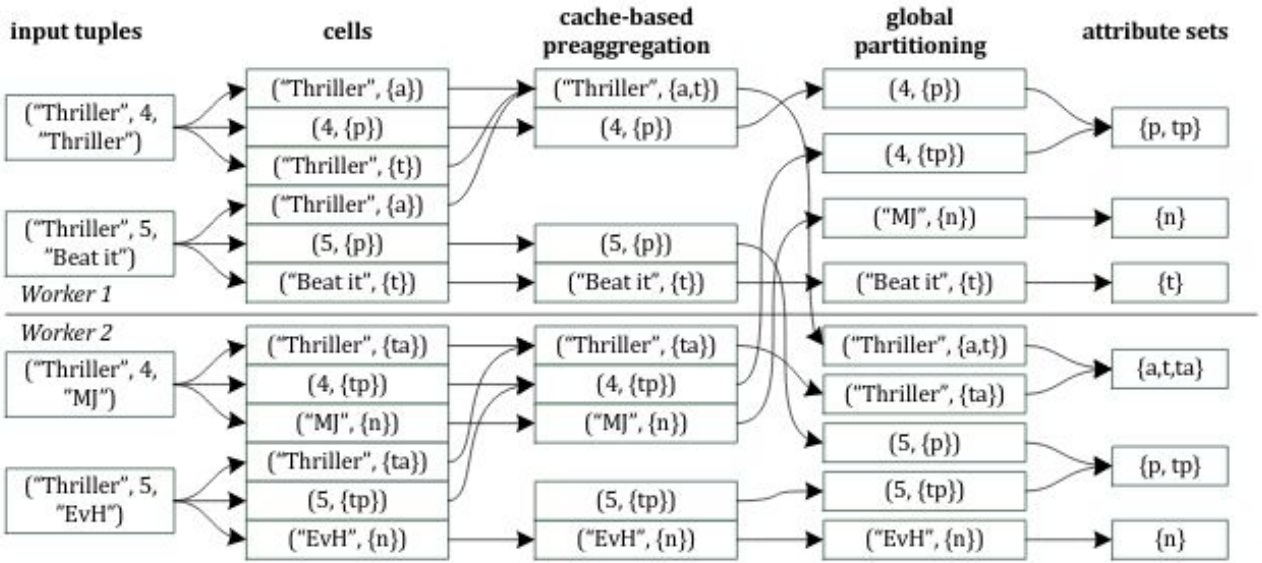
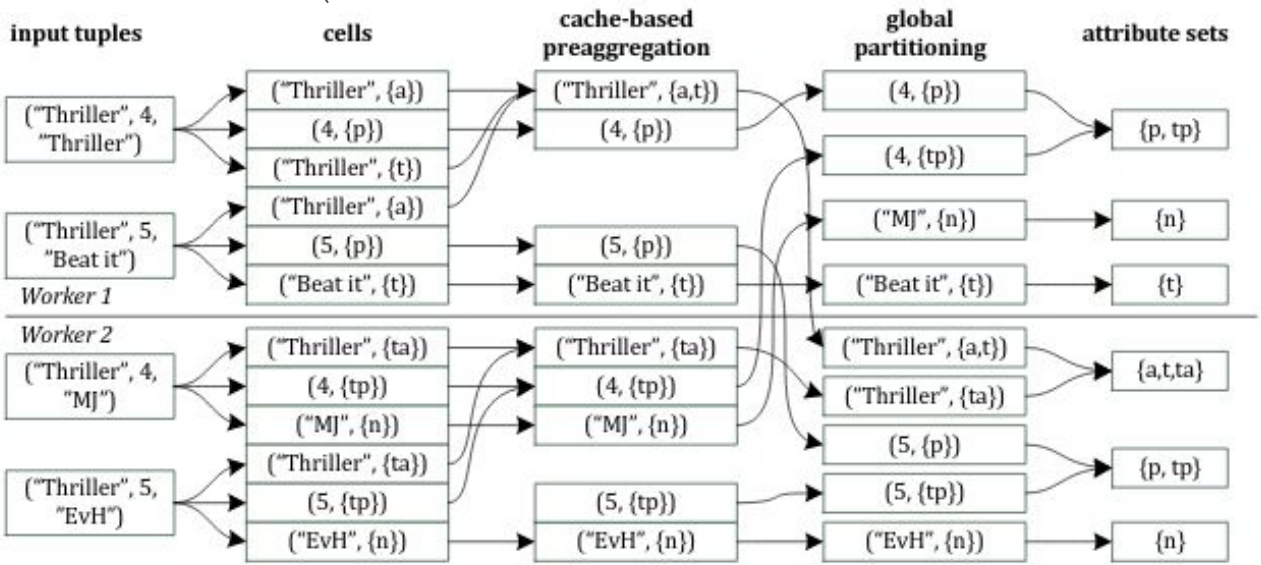
Team Lass es
es KRACHEN 😎🤣

es KRACHEN 😎🤣
Team Lass es

Implementation nach Sindy

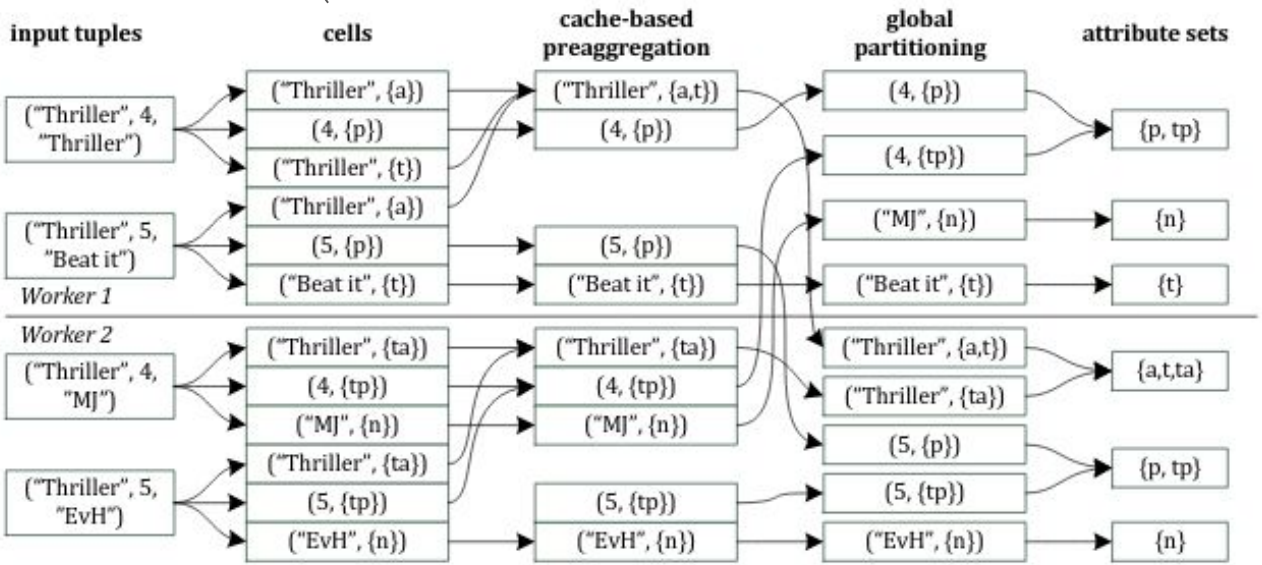


```scala
val values_to_columns_df = inputs.flatMap( file => {
  val df = reader.csv(file)
  df.columns.map( column => {
    df.select(column).map( row => ValueColumnPair(row.getString(0), column) )
  })
})
```



```
val values_to_columns_df = inputs.flatMap( file => {
  val df = reader.csv(file)
  df.columns.map( column => {
    df.select(column).map( row => ValueColumnPair(row.getString(0), column) )
  })
})
```

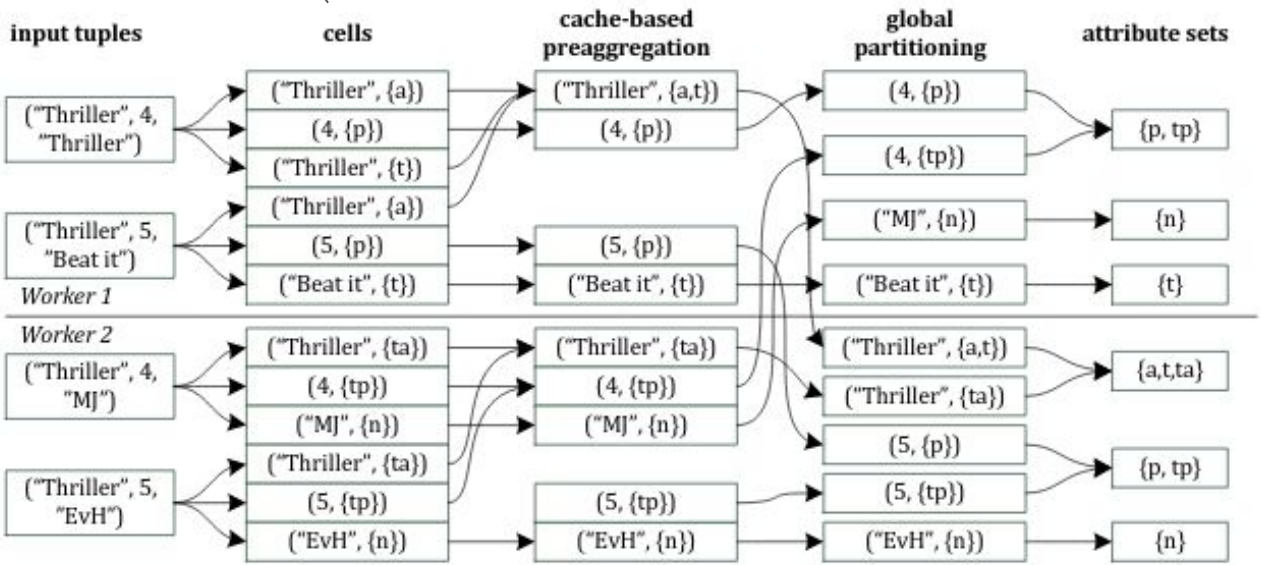
```
val unioned_values_to_columns = values_to_columns_df.reduce(_ union _)
```



```scala
val values_to_columns_df = inputs.flatMap( file => {
  val df = reader.csv(file)
  df.columns.map( column => {
    df.select(column).map( row => ValueColumnPair(row.getString(0), column) )
  })
})
```

```scala
val unioned_values_to_columns = values_to_columns_df.reduce(_ union _)
```

```scala
val columns_per_distinct_value = unioned_values_to_columns
  .distinct()
  .groupBy("value")
  .agg(collect_set("column"))
```

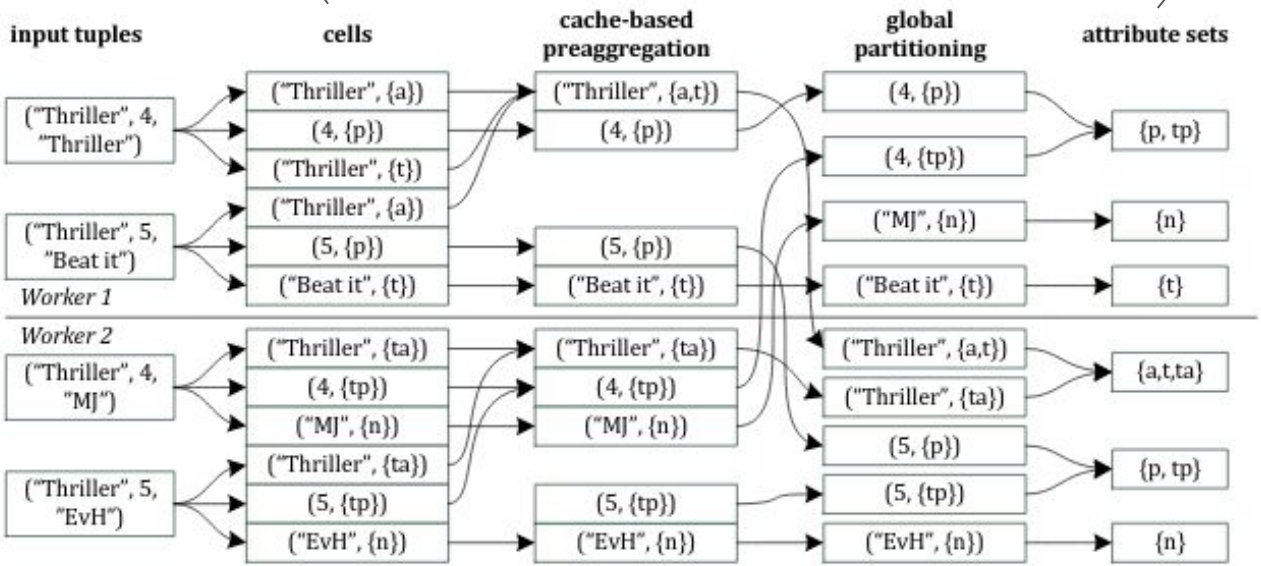


```
val values_to_columns_df = inputs.flatMap( file => {
  val df = reader.csv(file)
  df.columns.map( column => {
    df.select(column).map( row => ValueColumnPair(row.getString(0), column) )
  })
})
```

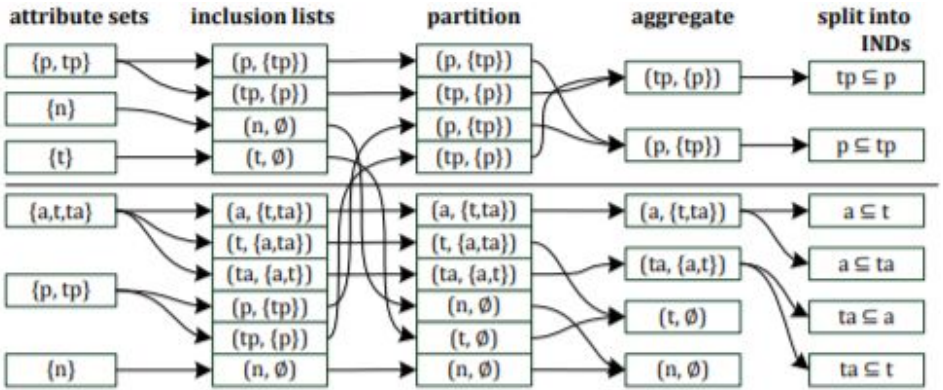
```
val unioned_values_to_columns = values_to_columns_df.reduce(_ union _)
```

```
val columns_per_distinct_value = unioned_values_to_columns
  .distinct()
  .groupBy("value")
  .agg(collect_set("column"))
```

```
// attribute sets as in the Sindy paper
val distinct_attribute_sets = columns_per_distinct_value
                                .select("collect_set(column)")
                                .distinct()
```



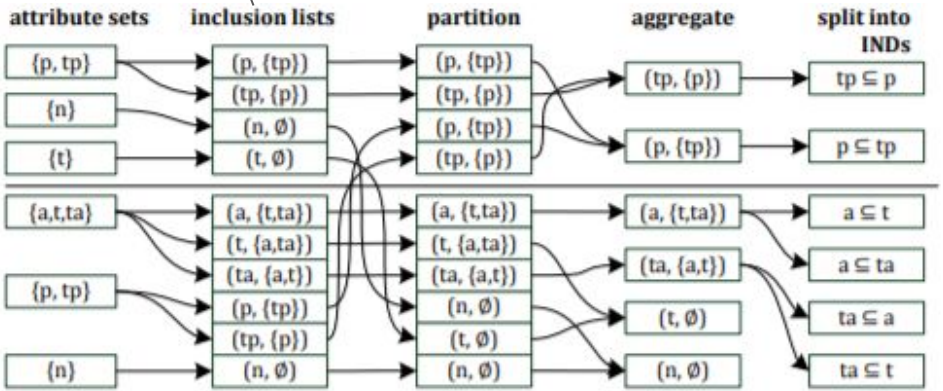
```
// attribute sets as in the Sindy paper
val distinct_attribute_sets = columns_per_distinct_value
                                .select("collect_set(column)")
                                .distinct()
```



```scala
// attribute sets as in the Sindy paper
val distinct_attribute_sets = columns_per_distinct_value
                                .select("collect_set(column)")
                                .distinct()
```

```scala
val inclusion_lists = distinct_attribute_sets.flatMap( row => {
  val list: Seq[String] = row(0).asInstanceOf[Seq[String]]
  val set = list.toSet

  set.map( element => {
    val reduced_set = set - element
    InclusionList(element, reduced_set)
  })
})
```



```scala
// attribute sets as in the Sindy paper
val distinct_attribute_sets = columns_per_distinct_value
                                .select("collect_set(column)")
                                .distinct()
```

```scala
val inclusion_lists = distinct_attribute_sets.flatMap( row => {
  val list: Seq[String] = row(0).asInstanceOf[Seq[String]]
  val set = list.toSet

  set.map( element => {
    val reduced_set = set - element
    InclusionList(element, reduced_set)
  })
})
```

```scala
val intersection_aggregator = new IntersectionAggregation()

val inclusion_dependencies = inclusion_lists
  .groupBy("column")
  .agg(intersection_aggregator(inclusion_lists.col("includedInColumns")).as("includedIn"))
```