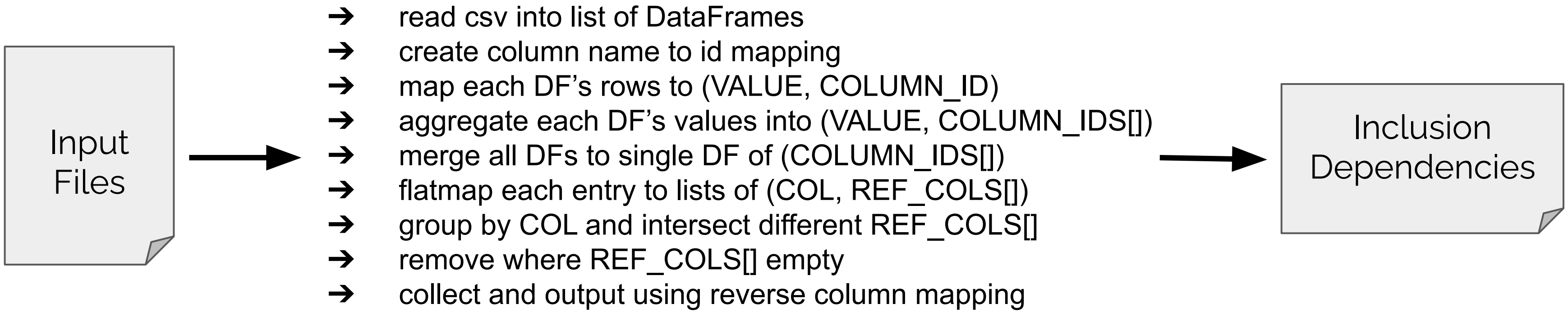
SINDY in Spark

mAKKAronis



Input Files

- ➔ read csv into list of DataFrames
- ➔ create column name to id mapping
- ➔ map each DF's rows to (VALUE, COLUMN_ID)
- ➔ aggregate each DF's values into (VALUE, COLUMN_IDS[])
- ➔ merge all DFs to single DF of (COLUMN_IDS[])
- ➔ flatmap each entry to lists of (COL, REF_COLS[])
- ➔ group by COL and intersect different REF_COLS[]
- ➔ remove where REF_COLS[] empty
- ➔ collect and output using reverse column mapping

Inclusion Dependencies

```
"C_CUSTKEY"      => 0,
"C_NATIONKEY"    => 1,
...
"S_COMMENT"      => ...
```

Column-ID-M
apping

```
val merged_dataframe = aggregated_dataframes
      .reduce(_ union _)
      .groupBy("_1")
      .agg(flatten(collect_set("_2")).alias("_2"))
      .drop("_1")
```

DataFrame Merge Logic