Spark - Inclusion Dependency Discovery

Staircase Attack

Sindy *(implemented like in the "Scaling Out the Discovery of Inclusion Dependencies" paper)*

Spark: read all input paths as csv files into tables → repartition → create a buffer containing RDDs for each column of all tables with each cell having the (<value>, <columnname>) format → convert list buffer of RDDs to one RDD containing all cells

take just the distinct values to create the attribute sets {<columnname1, columnname2, ...>} ← reduceByKey so each element has the format (<value>, {<columnname1, columnname2, ...>})

flatMap on the attribute sets to create the inclusion lists by creating lists {attribute, {attributes - attribute}} → reduceByKey and apply intersection to aggregate the inclusion lists → filter out inclusion lists containing the empty set

use foreach to print each output in the desired output format
<dependent> < <referenced1>, <referenced2>, ... ← collect all output results ← sort lexicographically by key