```java
static ArrayList<Pair> inclusionDependencies(WorkMessage dependent, WorkMessage referenced) {
    ArrayList<Pair> ans = new ArrayList<>();
    for (int colInd = 0; colInd < referenced.columns(); colInd++) {
        Set<String> indVals = new HashSet<>();
        for (int i = 0; i < referenced.rows(); i++) {
            indVals.add(referenced.elem(i, colInd));
        }
        int len = indVals.size();
        for (int colDep = 0; colDep < dependent.columns(); colDep++) {
            boolean depends = true;
            for (int i = 0; i < dependent.rows(); i++) {
                String cur = dependent.elem(i, colDep);
                indVals.add(cur);
                if (indVals.size() > len) {
                    indVals.remove(cur);
                    depends = false;
                }
            }
            if (depends) {
                ans.add(new Pair(dependent.getId(), referenced.getId(), dependent.col(colDep), referenced.col(colInd)));
            }
        }
    }
    return ans;
}
```

This is a Java function that finds inclusion dependencies between two "WorkMessage" objects, "dependent" and "referenced". An inclusion dependency between two columns exists when all values in the dependent column also appear in the independent column. The function uses a set "indVals" to store unique values of the referenced column. The loop over "colInd" and "colDep" iterates through all columns of the referenced and dependent tables, respectively. For each column, the function checks if all elements in the dependent column are included in the referenced column by adding the elements to the set and checking if the size of the set increases. If it does not, the function adds a pair of the dependent and referenced columns and their respective "WorkMessage" ids to the "ans" ArrayList. The function returns the "ans" ArrayList at the end, which contains all the inclusion dependencies found between the two "WorkMessage" objects.