

1- Dependências:

{RowNR}->R

{DEP}->{CD, RD}

{CD}->{RD}

{ARR}->{CA, RA}

{CA}->{RA}

{Actype}->{ActypefullName,AG,AGfullName}

{ActypefullName}->{Actype,AG,AGfullName}

{AG}->{AGfullName}

{AGfullName}->{AG}

2- Algoritmo

```
para (todasCols como col) { //col=coluna determinante
```

```
    indexDefines=0;
```

```
    para (todasCols como colCompare){ //colCompare=coluna determinante
```

```
        se (depends(col,colCompare) && col!=colCompare){
```

```
            defines[indexDefines]=colCompare; //vetor das dependências da col
```

```
            indexDefines++;
```

```
        }
```

```
    }
```

```
}
```

```
boolean depends (int col,int colCompare){ //colCompare = coluna dependente
```

```
select distinct col, colCompare from table order by col; //dá uma seleção sem repetições e organizada pela coluna col (coluna que pretendemos testar se é determinante)
```

```
para (linha=0; linha<numLinhas; linhas+2){ //numLinhas=num de linhas do select anterior
```

```
    se (linha.col==(linha+1).col)**
```

```
        return false;
```

```
    }
```

```
return true;
```

```
}
```

```
/**Como os valores estão organizados pela coluna col (coluna que define a outra) e são eliminados os valores repetidos, se houver valores seguidos repetidos quer dizer que para o
```

mesmo valor de col vai haver diferentes valores de colCompare o que torna esta não dependente de col.