

A) Beschreibung von Clustereigenschaften und Ausgabe passender CL-Algos

input: kann runde cluster, kann eingekreiste cluster, ..., kann brücken

output: einfache Aufzählung von CL-Algos

example: gegeben ein rundes, nicht zusammenhängendes Cluster, gesucht sind CL-Algos, die das gut Lösen können

```
SELECT ?algoname
WHERE {
    ?algo :can_circular ?circular.
    FILTER(?circular = circular value)
    ?algo :can_encircled ?encircled.
    FILTER(?encircled = encircled value)
    ?algo :can_bridged ?bridged.
    FILTER(?bridged = bridged value)
    ?algo rdfs:label ?algoname.
}
```

- Annahme, dass ein Algo alle festgelegten Clustereigenschaften erfüllen muss damit er aufgezählt wird!
- ist eigentlich nicht viel anderes als in C) !!

B) Finden von CL-Algos nach einer bestimmten Kategorie

input: categoryname

output: einfache Aufzählung von CL-Algos

example: gegeben Kategorie Centroid-based, Ausgabe aller passender CL-Algos die Centroid-based sind

```
SELECT ?categoryname ?algoname
WHERE {
    ?algo :has_category ?category.
    ?category dc:title ?categoryname.
    FILTER(?categoryname = "categoryname")
    ?algo rdfs:label ?algoname.
}
```

C) Finden von CL-Algos anhand von mehreren Eigenschaften

input: parameter value, noise value, dim value, speed value

output: einfache Aufzählung von CL-Algos

```
SELECT ?algoname
WHERE {
    ?algo :take_parameter ?param.
    FILTER(?param = "parameter value")
    ?algo :handle_noise ?noise.
    FILTER(?noise = noise value)
    ?algo :multi_dim ?dim.
```

```

FILTER(?dim = dim value)
?algo :speed ?speed.
FILTER(?speed = "speed value")
?algo rdfs:label ?algoname.
}

```

- Annahme, dass ein Algo alle festgelegten algorithmischen Eigenschaften erfüllen muss damit er aufgezählt wird!

D1) Finden von Literatur zu einem einzigen CL-Algo

input: algoname

output: Tabelle mit Algo und zugehörigen Papers

```

SELECT ?papername ?algoname
WHERE {
    ?algo rdfs:label ?algoname.
    FILTER(?algoname = "algoname")
    ?algo :described_in ?paper.
    ?paper dc:title ?papername.
}

```

	Algo
Paper1	x
Paper2	x

oder

Algos	Papers
Algo	Paper1, Paper2, ..

- was passiert wenn algo gar nicht existiert?
 - entweder absichern, sodass nur gültige Algos selektiert werden können
 - oder als Result "Algo existiert nicht" ausgeben
- was passiert wenn für algo keine papers gefunden werden können?
 - als Result unter Spalte Papers einfach keine Papers angeben

D2) Finden von Literatur zu mehreren CL-Algos

input: algoname1, algoname2, ..., algonameN

output: Tabelle mit Algo und zugehörigen Papers

```

SELECT ?papername1 ?algoname1 ?papername2 ?algoname2 ?papernameN ?algonameN
WHERE {
  ?algo1 rdfs:label ?algoname1.
  FILTER(?algoname1 = "algoname1")
  ?algo1 :described_in ?paper1.
  ?paper1 dc:title ?papername1.

  ?algo2 rdfs:label ?algoname2.
  FILTER(?algoname2 = "algoname2")
  ?algo2 :described_in ?paper2.
  ?paper2 dc:title ?papername2.

  ?algoN rdfs:label ?algonameN.
  FILTER(?algonameN = "algonameN")
  ?algoN :described_in ?paperN.
  ?paperN dc:title ?papernameN.
}

```

	Algo1	Algo2	AlgoN
Paper1	x		
Paper2			x
Paper3	x		
Paper4	x	x	

oder

Algos	Papers
Algo1	Paper1, Paper2, Paper3
Algo2	
AlgoN	Paper1, PaperN

- Annahme, dass man von vornherein die Anzahl der CL-Algos kennt, dass wird aber erst zur Laufzeit festgelegt! (Problem!)
- was passiert wenn algo gar nicht existiert?
 - entweder absichern, sodass nur gültige Algos selektiert werden können
 - oder als Result "Algo1 Algo4 existieren nicht" ausgeben und gleichzeitig zugehörige Zeilen nicht mit ausgeben!
- was passiert wenn für algo keine papers gefunden werden können?
 - als Result unter Spalte Papers einfach keine Papers angeben