**Partie politique** :

LOAD CSV WITH HEADERS FROM "file:///DonateurUK.csv" as line return line

MATCH (r:Recipient) WHERE r.recipientName =~ ".\*Caroline Lucas.\*" RETURN r

MATCH (r:Donor) WHERE r.donorStatus = "Company" RETURN r

MATCH (d:Donor)-[:DONATED\_TO]->(r:Recipient), (d:Donor)-[:DONATED\_TO]->(r1:Recipient) WHERE r.recipientName =~ ".\*Labour \*" and r1.recipientName =~ ".\*Conservative.\*" RETURN d

MATCH (r:Donor) WHERE r.donorStatus = "Company" and r.Value > 10000 RETURN r (Convertir données en int avant)

MATCH (d:Donor)-[do:DONATED\_TO]->(r:Recipient) set do.amount = toInt(do.amount)

MATCH (d:Donor)-[do:DONATED\_TO]->(r:Recipient), WHERE d.donorStatus = "Company" and do.amount > 10000 RETURN d

MATCH (d:Donor)-[do:DONATED\_TO]->(r:Recipient) WHERE d.donorStatus = "Company" and do.amount > 10000 RETURN d

**Partie docteurs** :

Dans option, mettre : Enable multi statement query editor

CREATE CONSTRAINT ON (a:DOCTOR) ASSERT a.ID IS UNIQUE; USING PERIODIC COMMIT 2000

LOAD CSV WITH HEADERS FROM "file:///sunshine.anonymes.csv" as line FIELDTERMINATOR ','

MERGE (a:DOCTOR {ID: line.BENEF\_PS\_ID})

ON CREATE SET a.ORIGIN= line.ORIGIN, a.BENEF\_PS\_QUALIFICATION= line.BENEF\_PS\_QUALIFICATION, a.BENEF\_PS\_DEPARTEMENT=

line.BENEF\_PS\_DEPARTEMENT, a.totalDECL = toInt(line.DECL\_AVANT\_MONTANT) ON MATCH SET a.totalDECL = toInt(line.DECL\_AVANT\_MONTANT) + a.totalDECL;

Identifier le médecin qui a reçu le plus d'argent des compagnies pharmaceutiques.

match (p:PHARMA)-[i:IS\_LINKED\_TO]->(d:DOCTOR) where d.ORIGIN = "Médecin" and i.totalDECL is not null return d.ID, i.totalDECL ORDER BY i.totalDECL DESC LIMIT 1

Quelles sont les compagnies pharmaceutiques les plus généreuses.

match (p:PHARMA)-[i:IS\_LINKED\_TO]->(d:DOCTOR)

WHERE i.totalDECL IS NOT NULL

return p.NAME, d.ID, i.totalDECL

ORDER BY i.totalDECL DESC

LIMIT 5

Donnez le montant total des cadeaux donnés par les laboratoires, d'abord la valeur déclarée par les laboratoires puis par les praticiens de santé et enfin la somme dans la réalité

Labo : match (p:PHARMA)-[i:IS\_LINKED\_TO]->(d:DOCTOR)

return sum(i.totalDECL)

1. Donnez le nombre de contrat établi entre les laboratoires et les praticiens ainsi que le nombre de cadeaux offerts :

Nombre cadeaux offerts :

match (p:PHARMA)-[h:HAS\_DONATED]->(d:DOCTOR)

WHERE h.DECL\_AVANT\_NATURE = "CADEAU"

return p.NAME, d.ID, count(h.DECL\_AVANT\_NATURE)

nombre de contrats passés :

match (p:PHARMA)-[h:HAS\_DONATED]->(d:DOCTOR)

WHERE h.DECL\_CONV\_OBJET =~ ".\*CONTRAT.\*"

return p.NAME, d.ID, count(h.DECL\_CONV\_OBJET)

1. Médecins qui acceptent les cadeaux de différents laboratoires :

match (p:PHARMA)-[h:HAS\_DONATED]->(d:DOCTOR)

WHERE h.DECL\_AVANT\_NATURE = "CADEAU"

return p.NAME,collect(distinct d.ID), h.DECL\_AVANT\_NATURE

Code pdf précédent**:**

match (n:Personne{nom:"domingues"}),(m:Personne{nom:"mogenet"}) create (n)-[r:connait]->(m) return n,r,m

match (n) detach delete n