

--	--	--

BDLE – Seconde Partie
Exemen réparti du 7 Septembre 2018
Durée : 2 Heures (au total)
Documents autorisés

Soit un extrait de la base MovieLens ci-dessous. Chaque ligne décrit un nuplet ayant le format

MovieID::Title::Genres

L'attribut Genres n'est pas atomique et contient un ensemble de genres séparés par des "|".

	Genre	NbFilms
1 : :Toy Story (1995) : :Animation Children's Comedy	Animation	1
2 : :Jumanji (1995) : :Adventure Children's Fantasy	Thriller	2
3 : :Grumpier Old Men (1995) : :Comedy Romance	Comedy	5
4 : :Waiting to Exhale (1995) : :Comedy Drama	Fantasy	1
5 : :Father of the Bride Part II (1995) : :Comedy	Romance	2
6 : :Heat (1995) : :Action Crime Thriller	Drama	1
7 : :Sabrina (1995) : :Comedy Romance	Children's	3
8 : :Tom and Huck (1995) : :Adventure Children's	Adventure	3
9 : :Sudden Death (1995) : :Action	Crime	1
10 : :GoldenEye (1995) : :Action Adventure Thriller	Action	3

films.txt

résultat.txt

Le but est d'extraire le nombre de films par genre qui, pour films.txt retourne résultat.txt.

Question 1 (2 points)

Compléter les instructions suivantes.

```
val data = sc.textFile(path+"films.txt")
```

```
scala> val tuples = data.
```

```
... .
```

```
... .
```

```
... .
```

```
tuples: org.apache.spark.rdd.RDD[(Int, String, String)] = ... .
```

```
scala> tuples.collect.foreach(println)
```

```
(1, Toy Story (1995), Animation)
```

```
(1, Toy Story (1995), Children's)
```

```
scala> val compteParGenre = tuples.
```

```
. . .
```

```
. . .
```

```
compteParGenre: scala.collection.Map[String, Long] = Map(Animation -> 1, ...
```

On s'intéresse maintenant à la formulation de requêtes Sparql en utilisant l'algèbre Dataset de Spark. On considère les triplets ci-dessous de la forme s, p, o , décrivent un sujet s est relié à un objet o via la propriété p .

n1,p1,n2	n2,p2,n3	n3,p3,n2
n3,p4,c	m1,p1,m2	m2,p2,m3
m3,p3,m2	m3,p4,c	c,p5,n4
c,p5,m4	n4,p6,m4	

triples.txt

Exprimer en **Dataset** les requêtes Sparql suivantes en considérant que les instructions ci-dessous sont déjà exécutées :

```
scala> case class Triple(sujet: String, prop: String, objet: String)
scala> val triples = sc.textFile("triples.txt").
  map(ligne => ligne.split(",")).
  map(tab => Triple(tab(0), tab(1), tab(2))).toDS()
```

Question 1 (1½ points)

```
select ?x ?z where { ?x p1 ?y. ?y p2 ?z. ?z p3 ?y }
```

```
scala> val . . .
```

```
. . .
```

```
. . .
```

```
. . .
```

```
. . .
```

```
. . .
```

```
. . .
```

```
scala> val res = . . .
```

le contenu de *res* est

n1	n3
m1	m3

Question 2 (1½ points)

```
select ?x1 ?x2 where { ?x1 p4 ?y. ?x2 p4 ?y. ?y p5 ?w1. ?y p5 ?w2. ?w1 p6 ?w2
} filter ( ?x1 != ?x2)
scala> val . . .
```

```
. . .
```

```
. . .
```

```
. . .
```

```
. . .
```

```
. . .
```

```
. . .
```

```
scala> val res = . . .
```

le contenu de *res* est

n3	m3
----	----