Nom :	**Prénom :**	*page 1*

BDLE – 5I852 - Examen réparti du 13 février 2015

Ex1 :	
Ex2 :	

Seuls les documents de cours et de TD sont autorisés – Durée : 2h.

Répondre aux questions sur la feuille du sujet dans les cadres appropriés. Utiliser le dos de la feuille précédente si la réponse déborde du cadre. Le barème est donné à titre indicatif. La qualité de la rédaction sera prise en compte. Ecrire à l'encre bleue ou noire. Ne pas dégrafer le sujet. Eteindre et ranger tout téléphone et autre appareil électronique.

Exercice 1. Manipulation de données avec Spark — 6 pts

On considère les données sur des films et les avis émis par des utilisateurs.

Avis (pseudo, film, étoile) // *film* est le numéro du film
Film (num, titre, année, catégorie)

Tous les attributs sont des nombres entiers (type Long) sauf le titre et le pseudo qui sont de type String.
Les données sont stockées dans les collections suivantes :

val avis :RDD[(String, Long, Long)]
val film : RDD[(Long, String, Long, Long)]

Rappel des opérations sur les collections :
c1.distinct : retourne un ensemble contenant les éléments de c1 sans doublons.
c1.filter(p) : retourne les éléments *e* tels que *e* est dans *c1* et *p(e)* est vrai.
c1.join(c2) retourne les éléments (k, (v, w)) tels que (k, v) est dans *c1* et (k, w) est dans *c2*.
c1.reduceByKey(f) : retourne les éléments (k, y) tels que pour l'ensemble des couples (k, v_i) dans *c1* ayant la clé k, on obtient y en appliquant *f* sur les v_i. On a y = f(...f(f(f(v_1, v_2), v_3), ...), v_n)
Remarque : les opérations join et reduceByKey ne sont applicables que sur des collections contenant des couples. Lorsque nécessaire, transformer préalablement les nuplets en couple à l'aide d'une opération map.

1) Que représente a ?

```
val a = avis.filter({case (pseudo, film, étoile) => étoile == 5}).map({case (pseudo, film, étoile) =>
film}).distinct
```

2) Que représente b3? Préciser aussi le type des éléments de b3.

```
val b1 = avis.map({case (pseudo, film, étoile) => (film, étoile) })
val b2 = film.map({case (num, titre, année, catégorie) => (num, catégorie) })
val b3 = b1.join(b2).map({case (num, (etoile, catégorie)) => (catégorie,1)}).reduceByKey( (v, w) => v+w)
```

```
b3
```

3) Exprimer *d* contenant les numéros d'utilisateurs ayant posté au moins 20 avis. Le type du résultat est RDD[Long].

```
Val d =
```

4) Exprimer *u* contenant les utilisateurs qui ont noté au moins un film qu'Alice a noté. Rmq, décomposer la réponse en exprimant d'abord *fa* les numéros de films qu'Alice a noté.

```
Val fa =


Val u =
```

5) Les données sont réparties sur les machines M1 à M10 en utilisant les fonctions de hachage h et h'

Mi contient les collections Ai et Fi : ($1 \leq i \leq 10$)

Ai : les avis tels que h(pseudo) = i

Fi : les films tels que h'(numéro) = i

Soit la requête R suivante : « Quelles personnes ont attribué 5 étoiles à un film de catégorie 1 ? ». R est une collection de pseudo, sans double. Le résultat de R peut demeurer réparti sur plusieurs machines (inutile de rassembler les données du résultat sur une seule machine).

Quelles données sont transférées entre les machines lors du calcul de R ? Expliquer brièvement les étapes. On ne demande **pas** le détail des expressions spark.

```
Sur Mi on calcule    _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

Puis on envoie _ _ _ _ _ _ _ _ _ _ _ _ vers _ _ _ _ _ _ _ _

Puis sur Mi on calcule  _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
```

6) On modifie la façon dont les avis sont répartis. Les avis sont maintenant répartis par numéro de film :

Ai : les avis tels que h'(film) = i

Les transferts sont-ils les mêmes que dans la question précédente ? Justifier.

Exercice 2. Stockage clé valeur 4 pts

On considère la base

Avis (<u>pseudo, film</u>, étoile, date) // *film* est le numéro du film

Film (<u>num</u>, titre, année, catégorie, nbavis) // *nbavis* est le nombre d'avis du film

1) Proposer des clés pour stocker les données dans kvstore. Une clé est formée de deux composantes : majeure et mineure. Répondre en séparant les deux composantes par un tiret (/ - /). Une composante est une liste de termes. Un terme en majuscule est un mot constant, un terme en minuscule est un littéral. Préciser la valeur associée à chaque clé.

Exemple de clé : /FILM/num/ - /TITRE/ La composante majeure est le mot « FILM » suivie d'un nombre égal au numéro du film. La composante mineure est le mot TITRE. La valeur associée à cette clé est le titre du film.

On précise les requêtes de l'application :

R1(f) : le pseudo des personnes ayant noté le film f.

R2(p) : les avis du pseudo p. Le résultat contient les attributs (film, étoile, date) des avis.

R3(c) : le titre des films de la catégorie c

R4(f) : le film f. Le résultat contient les attributs (titre, année, catégorie, nbavis)

R5 : le top 10 des numéros de films ayant le plus grand nombre d'avis

On sait que les paires ayant la même composante majeure sont stockées sur la même machine. Répondre de telle sorte que les clés permettent de répondre rapidement aux requêtes de l'application, en évitant d'accéder à plusieurs machines pour évaluer une requête.

2) On rappelle qu'une transaction ne peut modifier que des données ayant la même composante majeure. On veut traiter la transaction **T1** (u, f, e, d) qui ajoute un nouvel avis (l'utilisateur *u* attribue *e* étoiles au film *f* à la date *d*), et qui incrémente le nombre d'avis de f. Proposer des clés pour stocker les données de telle sorte qu'il soit possible de traiter T1.

3) Les données sont répliquées plusieurs fois (sur des machines différentes). On veut utiliser les répliques pour traiter T1 et R4 le plus rapidement possible. On propose 3 stratégies dénotées S1, S2 et S3 :

S1 : écriture : propagation synchrone vers toutes les répliques
lecture : lire une réplique quelconque

S2 : écriture : propagation asynchrone
lecture : lire le maître

S3 : écriture : propagation asynchrone
lecture : lire une réplique quelconque

Expliquer les avantages et inconvénients de chaque stratégie. Proposer une stratégie qui tient compte de la situation où un utilisateur exécute R4(f) juste après avoir invoqué T1 pour ajouter son avis sur le film f.