Typical Solution: Big Data and Associated Technologies

Exercise 1:

1. True (0,5 pt)

2. False (0,5 pt)
Explanaition: RDD stands for Resilient Distributed Dataset, not "Real-time". It is Spark's core abstraction for distributed data processing. (0,5pt)

3. False (0,5pt)
Explanation: These transformations are **lazy**, meaning they are not executed until an action (like count() or collect()) is called.

4. False (0,25pt)

Explanation: (1,75pt)

1. **Hadoop is a data processing framework**, not a database — it does not support structured query languages like SQL or provide indexing like traditional databases.
2. **It stores data in raw format (e.g., HDFS)** without a schema, and requires external tools (like Hive, Pig, or Spark) for data querying and analysis.

Exercise 2: (3pts)

Response a)
Explanation:

In Hadoop, the number of mapper instances is usually equal to the number of input splits, which are based on the HDFS block size.
To run exactly 5 mappers, we need to split the 10,000 MB file into 5 blocks.

Block size = Total file size / Number of desired blocks
= 10,000 MB / 5
= 2,000 MB

Exercise 3: (5pts)

1. db.stagaires.find({"groupe._id":1},{}).count()

2.

```
db.stagiaires.find({"groupe.filiere.modules":
        {$elemMatch:{
                                "examens.note":{$gt:10},
```

```
                                    "nom":"base de données"
                                    }}
})
```

3.

```
db.stagiaires.insertOne({
  _id: 2,
  nom: "new trainee",
  groupe: {
    _id: 1,
    nom: "g1",
    filiere: {
      _id: 1,
      nom: "filiere1",
      modules: [
        {
          _id: 1,
          nom: "module1",
          examens: [
            { type: "type1", note: 0 }
          ]
        }
      ]
    }
  }
})
```

4. db.stagiaires.deleteMany({ "groupe._id": 1 })

5.

db.stagiaires.find({"groupe.nom":"WFS"},
{"groupe.filiere.modules":1})

Exercise 4: (7pts)

MapReduce Program Logic
Mapper: (1pt)

- Reads each line
- Extracts date (the full date: 2016/01/01) and temperature (e.g. -1.2)
- Emits a key-value pair:

(date, temperature) (1pt)

Reducer: (1pt)

- For each date key, receives all temperature values
- Computes and emits the maximum temperature for that date
- Input Data Recap

Let's list all entries:
Temperature1.txt
2016/01/01,00:00,0
2016/01/01,00:05,-1
2016/01/01,00:10,-1.2

Temperature2.txt

2016/01/01,00:30,-0.5

2016/01/01,00:35,1

2016/01/01,00:40,1.5

Temperature3.txt

2016/01/01,00:30,-0.5

2016/01/01,00:35,1

2016/01/01,00:40,1.5

Mapper Output:

Each line emits:

Key = 2016/01/01

Value = temperature

So the emitted values will be: **(1pt)**

("2016/01/01", 0)

("2016/01/01", -1)

("2016/01/01", -1.2)

("2016/01/01", -1.5)

("2016/01/01", 0)

("2016/01/01", -0.5)

("2016/01/01", -0.5)

("2016/01/01", 1)

("2016/01/01", 1.5)

Reducer Logic: (1pt)

Key: "2016/01/01"

Values: [0, -1, -1.2, -1.5, 0, -0.5, -0.5, 1, 1.5]

Max temperature = 1.5

✅Final Output: (2 pts)

(2016/01/01, 1.5)

Explanation:

The mapper emits the date and corresponding temperature.

The reducer calculates the maximum temperature for that date.

Since all entries are from the same date, the result is a single line with the highest temperature on 2016/01/01.