Big Data Management (MDS/BDMA)

Final Exam

Family name:____________________________________Given name: ______________________

1. (20 points) Assuming the existence of a collection "events" in a **MongoDB cluster** whose optimizer does not work at all, if possible, **optimize the performance** of the following pipeline by modifying it without affecting its results at all. If not possible, briefly explain why.

```
db.events.aggregate([
  {$group: {
        _id: { type: "$event_type", success: "$success", team: "$team" },
        counter: { $count: {} }
      }
  },
  { $match: { "_id.type": "Attack", "_id.success": "True" } },
  { $project: { team: "$_id.team", counter: 1} }
  }
])
```

....

....

....

....

....

....

....

2. (20 points) Consider you have "$\oplus$" symbol for concatenation, "$prj_{a_{i_1}...a_{i_n}}(t)$" to get attributes "$a_{i_1}...a_{i_n}$" in t, and assume the "key" parameter k contains the PK of the table and the "value" v all the others. Given that and using any function "$f(l)$" you need over a list l of values, provide the pseudo-code of a single **MapReduce** job implementing the SQL query below. If not possible, briefly justify why.

   SELECT A, AVG(B) FROM T GROUP BY A HAVING COUNT(*)≥10;

Map(k,v)

....

....

....

....

Reduce(k, iv)

....

....

....

....

3. (20 points) Suppose you work in a project on a single dataset, but there are different studies being performed on it, requiring different queries. Chose a **correct fragmentation** of that dataset that minimizes the overall cost of the queries under the following assumptions:

   - The size of the dataset is seven columns (C_0 to C_6, where C_0 is the identifier of the rows in the dataset) and one hundred rows.
   - Fragments are retrieved completely (e.g., it is not possible to retrieve half a fragment).
   - The cost of a query can be simply estimated as the number of cells retrieved from the disk (i.e., columns multiplied by rows). No other cost needs to be considered.
   - The maximum number of fragments that can be defined is two.
   - There are three studies (all of them with the same frequency and relevance) that require one of the following queries each:
     1. SELECT C_0, C_1, C_2, C_3 FROM dataset;
     2. SELECT C_4, C_5 FROM dataset;
     3. SELECT C_5, C_6 FROM dataset;

   Give the kind of fragmentation, the fragments and the average cost of queries for the choice.

....

....

....

....

....

....

....

4. (20 points) Draw the result of the following Python code using **Spark dataframes**, assuming that "spark" variable contains an already running Spark session.

```
data = ({"A": 'a', "B": 1, "C": 1.1, "D": True},
{"A": 'b', "B": 2, "C": 2.2, "D": False},
{"A": 'c', "B": 3, "D": True},
{"A": 'd', "B": 4, "C": 4.4},
{"A": 'e', "B": 5, "C": 5.5, "D": True}
)
df = spark.createDataFrame(data)
df.replace(to_replace=[3,4,5], value=[66,88,None], subset=["B"])
df.show()
```

5. (20 points) Consider the following **PySpark code** and indicate a minimal change (i.e., do not modify more than three or four operations), so that it returns the department areas ("dArea") with departments in all cities where the employees working in those departments live. Optimizing the code is not necessary.

```
source1 = spark.read.format("csv").load("employees.txt", header='false', inferSchema='true', sep=";")
source2 = spark.read.format("csv").load("departments.txt", header='false', inferSchema='true', sep=";")
A = source1.toDF("eID","eName","eSalary","eCity","eDpt")[1]
B = source2.toDF("dID","dArea","dNumber","dStreet","dCity")
C = A.select(A.eCity.alias("city"))
D = B.select("dArea")
E = D.crossJoin(C)
F = B.select("dArea",B.dCity.alias("city"))
G = E.subtract(F)
H = G.select("dArea")
result = D.subtract(H)
result.show()
```

Example of Input:

Employees.txt

EMP1;RICARDO;250000;MADRID;DPT2
EMP2;EULALIA;150000;BARCELONA;DPT1
EMP3;MIQUEL;125000;BADALONA;DPT1
EMP4;MARIA;175000;MADRID;DPT3
EMP5;ESTEBAN;150000;BARCELONA;DPT4

Departments.txt

DPT1;DIRECCIO;10;PAU CLARIS;BARCELONA
DPT2;DIRECCIO;8;RIOS ROSAS;MADRID
DPT3;MARKETING;1;PAU CLARIS;BARCELONA
DPT4;MARKETING;3;RIOS ROSAS;MADRID

Expected output: "MARKETING"

. . . .

. . . .

. . . .

. . . .

[1] "eDpt" contains values in "dID"