January 26, 2017
Arnab Chakrabarti, Rihan Hai, Sandra Geisler

Implementation of Databases (WS 16/17)

Exercise 7

Due until February 7, 2017, 2pm.
Please submit your solution *in a single PDF file* before the deadline to the L^{2}P system!
Please submit solutions in groups of three students.

Exercise 7.1 (MapReduce) (15 pts)

Consider a document-oriented database with documents in the following form:

student

```
{  type : "student",
   sid  : 123,
   name : "...",
   gpa  : 3.45
}
```

dept

```
{  type  : "dept",
   did   : 234,
   name  : "..."
}
```

course

```
{  type  : "course",
   cid   : 456,
   name  : "...",
   did   : 345
}
```

enroll

```
{  type  : "enroll",
   cid   : 456,
   sid   : 123,
   grade : 1.7
}
```

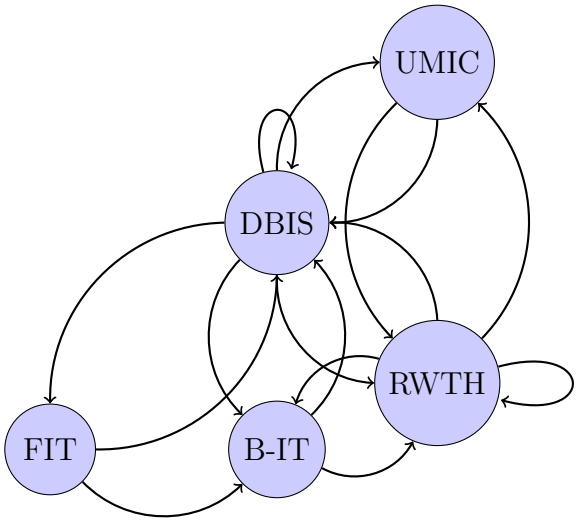
Define the following queries using the MapReduce pattern. The input to the map function is **one** document collection which contains all objects of the database, regardless of their type.

1. Compute the average gpa for all students with an SID less than 1.000.
2. For each course, compute the number of students enrolled in this course, and their average grade.
3. For each Computer Science course (a course from the department with the name 'CS'), get the course name and the average gpa of the students enrolled in the course.

Provide a sketch for the MapReduce implementation of these queries using some form of Java/-JavaScript pseudo code, similar to the syntax used in the lecture. Preferrably, implement the queries in MongoDB and test them with a sample collection.

Exercise 7.2 (PageRank) **(15 pts)**

Consider the following link graph of the websites UMIC (1), DBIS (2), RWTH Aachen University (3), B-IT Research School (4) and FIT Fraunhofer (5):



1. Write down the transition matrix which is defined as follows. The transition Matrix M has an element m_{ij} in row i and column j, where:
   (a) $m_{ij} = \frac{1}{r}$ if page j has a link to page i, and there are a total of $r \geq 1$ pages that j links to.
   (b) $m_{ij} = 0$ otherwise.

2. Suppose the initial page rank vector of the five pages is $\begin{bmatrix} \frac{1}{5} \\ \frac{1}{5} \\ \frac{1}{5} \\ \frac{1}{5} \\ \frac{1}{5} \end{bmatrix}$ What is the approximate final page rank vector and which page would be ranked the highest? *Hint:* you can achieve that by continuously multiplying the transition matrix to the vector until an approximate fixpoint is reached.

3. Define a data structure to represent documents (identified by a URL), their outgoing links, and their page rank in a JSON document. Sketch a MapReduce implementation of the Page Rank algorithm using some form of Java/JavaScript pseudo code, similar to the syntax used in the lecture. Preferably, implement the algorithm in MongoDB and test it with a sample collection.