Assignment 1: Recap of Standard Database Issues (1 P.)

(a) Transactions

(i) What are the ACID properties? Define them and illustrate through examples how each property contributes to the reliability of database transactions.

(ii) Specify an example of a lost update, a phantom read, and a dirty read, by defining two transactions T_1 and T_2 and, for each problem, create an example schedule.

(iii) Describe the Two Phase Locking (2PL) protocol. How can it be modified to prevent cascading aborts?

(iv) Is the following schedule of read and write operations conflict serializable? Explain what conflict serializability means and how one can decide if a schedule is conflict serializable or not.

$$r_1(a)r_2(b)r_3(a)w_1(b)w_3(c)r_5(b)w_2(d)w_4(e)r_5(d)r_4(b)w_3(a)r_3(e)$$

(b) Relational Algebra and SQL

Consider the following relational schema, where $\rightarrow$ expresses a foreign key relationship:

Person(<u>PID</u>, firstName, lastName, age)

Car(<u>CID</u>, name, year, model, price)

Mechanic(<u>PID → Person.PID</u>, city)

Repairs(<u>PID → Person.PID</u>, <u>car → Car.CID</u>, date, reason, amount)

Owns(<u>owner → Person.PID</u>, <u>car → Car.CID</u>, purchaseDate)

(i) Given the following expressions in relational algebra, check their correctness. If they are incorrect, briefly state the reason.

- $\pi_{firstName,lastName}(\sigma_{year<2010}(\pi_{firstName}(Person) \cup \pi_{lastName}(Person)))$
- $Person \bowtie_{Person.lastName=Mechanic.city} \pi_{city}(\sigma_{city='K-Town'}(Mechanic))$

(ii) Translate the following queries into SQL and relational algebra expressions:

- What are the cities in which at least one car has been repaired because of a broken windshield?
- What are the first names and ages of mechanics who repaired a Mercedes S550 on April 15, 2015?
- List all cars whose price is lower than €5000 and whose sum of repairs cost is larger than their price.

(iii) What do the following SQL queries compute?

(1)
```
SELECT p.firstName, p.lastName, c.name, c.model
FROM Person p, Repairs r, Car c, Mechanic m,
(SELECT x.PID, max(x.amount) as expensive
            FROM Repairs x
            GROUP BY x.PID) as y
WHERE p.PID = r.PID
AND p.PID = m.PID
AND y.PID = r.PIDa
AND r.car = c.CID
AND y.expensive = r.amount
```

(2)
```
SELECT p.firstName, p.lastName, sum(r.amount) AS repairs
FROM Person p LEFT OUTER JOIN  Repairs r ON p.PID = r.PID
WHERE NOT EXISTS
(SELECT m.PID
FROM Mechanic m
WHERE m.PID = p.PID)
GROUP BY P.PID
```

(c) Result Sizes

(i) Given two relations R and S with $|R| = m$, $|S| = n$, and A is an arbitrary attribute of R, specify the min and max number of results for the following relational algebra queries:

- $R \ltimes S$
- $R \times S$
- $\sigma(R \bowtie S)$

(ii) Given the following histogram over attribute A of a relation R. A is of type integer.

Bucket Bounds	**Average Frequency**
$[-3, -1)$	16
$[-1, 1)$	46
$[1, 3)$	11
$[3, 7)$	89

- How many tuples does R contain in total?
- How many tuples of R have an A value strictly larger than 0?
- How many tuples of R have an A value in $[-1, 3]$?

Assignment 2: MapReduce Fundamentals (1 P.)

(a) Describe what the following MapReduce functions do.

(i)
```
map(nr, txt)
  words = split (txt, ' ')
  for(i=0; i< |words| - 1; i++)
     emit(words[i]+' '+words[i+1], 1)

reduce(key, vals)
  s=0
  for v : vals
    s += v
  if(s = 5)
    emit(key,s)
```

(ii)
```
map(nr, txt)
    words = split (txt, ' ')
    for(i=0; i < |words|; i++)
```

```
    emit(txt, length(words[i]))

  reduce(key, vals)
    s=0
    c=0
    for v : vals
      s += v
      c += 1
    r = s/c
    emit(key,r)
```

(iii)
```
  map(nr, txt)
    words = split (txt, ' ')
    for(i=0; i < |words|; i++)
      emit(sort(words[i]), words[i])

  reduce(key, vals)
    s=0
    for v : vals
      s += 1
    if (s >= 2)
      emit(key, vals)
```

(b) Consider the following combiner functions. Each one corresponds to the MapReduce functions given in the first part of the assignment. Would these functions influence the final result of the MapReduce functions and would they improve the performance of the tasks? Explain your answer.

(i)
```
combine(key, vals)
  s=0
  for v : vals
    s += v
  emit(key,s)
```

(ii)
```
combine(key, vals)
  s=0
  c=0
  for v : vals
    s += v
    c += 1
  emit(key, s/c)
```

(iii)
```
combine(key, vals)
  s=0
  for v : vals
    s += 1
  if (s >= 2)
    emit(key, vals)
```

(c) Log Analysis with MapReduce

The MapReduce framework is particularly suitable for processing log files. Given a purchase log file, structured like the one in the table bellow:

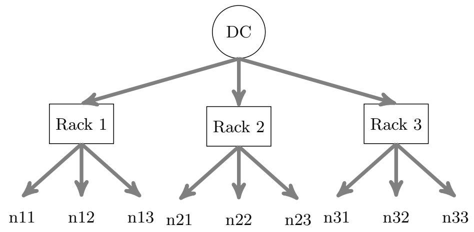
(i) Write a MapReduce job that will output all the usernames that have purchased more than 1000 items within 60 minutes starting on the hour (e.g., from 15:00-15:59).

(ii) Write a MapReduce job that will output Top 300 usernames by the number of purchased items for each Category.

Pid	Username	Date	Time	Item	Category
100	bruce	11/2/2013	19:22:00	Bat	Toys
101	clark	11/2/2013	13:45:00	Cape	Clothes
102	tony	11/2/2013	13:23:00	Robot	Tech
103	peter	11/9/2013	16:11:00	Suit	Business
104	jessica	11/2/2013	19:09:00	Jacket	Clothes
105	steve	11/9/2013	22:09:00	Baseball cap	Sports

Assignment 3: Rackaware Replica Placement in a DFS (1 P.)

Consider the following topology of nodes and racks in a data center (DC). Assume that two nodes within the same rack can send (read/write) data with a network bandwidth of 10 000 Mbit/s, whereas two nodes from different racks only have a bandwidth of 1000 Mbit/s available.



The probability that the entire DC breaks down (e.g., in a disaster) is 1/20, the probability that a specific rack (switch) breaks down is 1/10, the probability that a specific node breaks down is 1/4. For simplicity, we assume that the probability that a failure happens is independent of other failures.

Obviously, if the DC breaks down the entire nodes are unavailable, likewise, if a rack breaks down the enclosed nodes become unavailable, too. One failing node does not affect other nodes. Thus, you can interpret the directed edges in the topology as a "affects" relation.

(a) **Fault Tolerance:** Describe two different **valid** replica placement strategies (having only one replica on a node) and compute for each of them the probability that none of the replicas is available. Consider for the computation of the probability the case of the above topology and 2 replicas of a block (so there are 3 blocks in total).

(b) **Write Cost:** Consider a file that is split into 3 blocks of each 128MB size. Compute for both strategies you devised in part (a) the time it takes to create all replicas (following the replication principle "one after the other" as mentioned in the lecture). Discuss the pros and cons of the write/replication costs with respect to the fault tolerance.