Q 1. [15pts] For the following questions, clearly circle True or False.

1. A given physical query plan can correspond to exactly one logical query plan.  
   False

2. The RDBMS catalog is never useful for query optimization.  
   False

3. Pipeline parallelism usually helps increase parallelism on multicore machines.  
   True

4. It is sometimes possible to execute an entire SQL query on a table using only a given index structure on that table.  
   True

5. Sort-merge join is always a blocking physical operator in the context of pipelined query execution.  
   False

6. The selectivity of a conjunctive predicate is never higher than what is yielded by the independence heuristic assumption for costing plans.  
   False

7. Materialized views typically increase runtimes for query execution but help reduce storage space requirements for the database.  
   False

8. The network often becomes a bottleneck for data access in shared-disk parallel RDBMSs.  
   True
9. Parallel RDBMSs often help reduce query runtimes even when the database can fit entirely on a single node’s disk.
   True
10. Superlinear speedups are impossible in parallel RDBMSs.
    False
11. It is impossible to modify a parallel RDBMS to make it fault tolerant.
    False
12. Hive on Hadoop+MapReduce is effectively another RDBMS.
    True
13. Spark better exploits distributed memory than Hadoop+MapReduce.
    True
    False
15. NoSQL systems are called so because they offer more expressive declarative query languages than SQL.
    False

Q 2. [15pts] Clearly circle the correct answer for each of the following questions (only one option is correct).

1. [3pts] Which of the following data partitioning schemes is the most common in parallel RDBMSs?
   (a) Round-robin       (b) Hash-based       (c) Range-based       (d) Random
   ANSWER: (b) Hash-based

2. [3pts] Which of the following SQL aggregates cannot always be computed in just one pass over the dataset in a typical parallel RDBMS?
(a) AVG  (b) SUM  (c) VARIANCE  (d) MIN  (e) MEDIAN

ANSWER: (e) MEDIAN

3. [3pts] How many distinct hash functions does the “improved” parallel hash join implementation discussed in class need for the join phase?

(a) 1  (b) 2  (c) 3  (d) 4  (e) 5  (f) 6

ANSWER: (b) 2

4. [3pts] Which relational operator’s implementation in a parallel RDBMS does the implementation of Hadoop+MapReduce most closely resemble?

(a) $\sigma$  (b) $\pi$  (c) $\bowtie$  (d) $\gamma$  (e) $\rho$  (f) $\times$

ANSWER: (d) $\gamma$

5. [3pts] Which of the following SQL aggregates cannot be recomputed incrementally with just algebraic rewrites during materialized view maintenance?

(a) SUM  (b) COUNT  (c) AVG  (d) MIN  (e) MAX

ANSWER: (c) AVG

Q 3. [10pts] Are you still on social media? Given the following relational database schema, for each of the given relational algebra query over this database, which of the relational algebra queries listed under it are logically equivalent to it? Clearly circle the correct answer (only one is correct).

Person (ID, Name, Age)
Friends (ID1, ID2)

Person.ID is the primary key of Person. Both Friends.ID1 and Friends.ID2 are foreign keys referring to Person.ID.
1. [3pts] \( \pi\text{Name}(\sigma_{\text{Age}>20}(\text{Person})) \)

(a) \( \sigma_{\text{Age}>20}(\pi\text{Name}(\text{Person})) \)  
(b) \( \pi\text{Name, ID}(\sigma_{\text{Age}>20}(\text{Person})) \)

(c) \( \pi\text{Name}(\sigma_{\text{Age}>20}(\pi\text{Name, Age}(\text{Person}))) \)  
(d) \( \sigma_{\text{Age}>20}(\pi\text{Name, ID}(\text{Person})) \)

(e) \( \sigma\text{Name}(\pi_{\text{Age}>20}(\text{Person})) \)  
(f) \( \pi\text{Name}(\pi\text{Age}(\sigma_{\text{Age}>20}(\text{Person}))) \)

ANSWER: (c)

2. [3pts] \( \sigma_{(\text{Age}>20)\land(ID2=1234)\land(Name=\text{"Thanos"})\land(ID=ID1)}(\text{Person} \times \text{Friends}) \)

(a) \( \sigma_{(\text{Age}>20)\land(Name=\text{"Thanos"})}(\text{Person}) \times \sigma_{(ID2=1234)\land(ID=ID1)}(\text{Friends}) \)

(b) \( \sigma_{ID=ID1}(\sigma_{ID1=1234}(\text{Friends}) \times \sigma_{(\text{Age}>20)\land(Name=\text{"Thanos"})}(\text{Person})) \)

(c) \( \sigma_{(\text{Age}>20)\land(Name=\text{"Thanos"})}(\text{Person}) \bowtie_{ID=ID1}\sigma_{ID1=1234}(\text{Friends}) \)

(d) \( \sigma_{ID2=1234}(\text{Friends}) \bowtie_{ID=ID1}\sigma_{(\text{Age}>20)\land(Name=\text{"Thanos"})}(\text{Person}) \)

ANSWER: (d)

3. [4pts] \( \pi_{\text{Name, ID}}(\text{Person} \bowtie_{ID=ID2}\pi_{ID2}(\sigma_{\text{Name}=\text{"Thanos"}}(\text{Person} \bowtie_{ID=ID1}\text{Friends}))) \)
(a) \(\pi_{\text{Name}}(\pi_{\text{ID}_1,\text{ID}_2}(\pi_{\text{ID}}(\sigma_{\text{Name}="Thanos"}(\text{Person}))) \bowtie_{\text{ID} = \text{ID}_1} \text{Friends}) \bowtie \text{Person})\)

(b) \(\pi_{\text{Name},\text{ID}}(\text{Person}) \bowtie_{\text{ID} = \text{ID}_2} \pi_{\text{ID}_2}(\sigma_{\text{Name}="Thanos"}(\text{Person}) \bowtie_{\text{ID} = \text{ID}_1} \text{Friends})\)

(c) \(\pi_{\text{Name},\text{ID}}(\text{Person}) \bowtie_{\text{ID} = \text{ID}_1} \pi_{\text{ID}_1}(\sigma_{\text{Name}="Thanos"}(\text{Person}) \bowtie_{\text{ID} = \text{ID}_1} \text{Friends})\)

(d) \(\pi_{\text{Name},\text{ID}_2}(\text{Person}) \bowtie_{\text{ID} = \text{ID}_2} \pi_{\text{ID}_2}(\sigma_{\text{Name}="Thanos"}(\text{Person}) \bowtie_{\text{ID} = \text{ID}_1} \text{Friends})\)

**ANSWER:** (b)