# Assignment 7

**UBIT -** **Due – May 7th 2014**

- **Print this assignment and write your answers inline.**
- **By submitting this assignment, you attest this work was done individually by you.**
- **Do NOT make any assumptions. Required information is provided.**

Q1: 60 Points (15 Each)

Consider a distributed parallel DBMS in which each relation is stored by horizontally partitioning its tuples across all disks.

*Employees(eid : integer, did : integer, sal : real)*

*Departments(did : integer, mgrid : integer, budget : real)*

The *mgrid* field of Departments is the *eid* of the manager. Each relation contains 20-byte tuples, and the *sal* and *budget* fields both contain uniformly distributed values in the range 0 to 1,000,000. The Employees relation contains 100,000 pages, the Departments relation contains 5,000 pages, and each processor has 100 buffer pages of 4,000 bytes each. The cost of one page I/O is $t_d$, and the cost of shipping one page between two sites is $t_s$; tuples are shipped in units of one page by waiting for a page to be filled before sending a message from processor *i* to processor *j*. There are no indexes, and all joins that are local to a processor are carried out using a sort-merge join. Assume that the relations are initially partitioned using a round-robin algorithm and that there are 10 processors.

***The Employees table is stored at Naples and all of Departments stored at Berlin.*** **The query is posed at Delhi.**

Consider the query:

**SELECT ***

**FROM Employees E, Departments D**

**WHERE E.eid = D.mgrid**

*You are told that only 1 percent of employees are managers.*

Find the cost of answering this query using each of the following plans:

1. Compute the query at Naples by shipping Departments to Naples; then ship the result to Delhi.

2. Compute the query at Delhi by shipping both relations to Delhi.

3. Compute the query at Berlin using Bloomjoin; then ship the result to Delhi.

4. Compute the query at Naples using Semijoin; then ship the result to Delhi.

   OR

4. Compute the query at Naples using Bloomjoin; then ship the result to Delhi.

(Space for answer)

(Space for answer)

Q2: 25 Points

Consider a situation wherein a query involves computation at 2 disparate processing center. Also assume that to maintain ACID property, the two sub-transaction need to interact with each other and ensure that they commit at the same time. If one of the transacation commits and the other doesn't there is violation of consistency. Assume that there was a downtime in the network and none of the participating nodes could interact with each other or with the coordinator. In which stage of 2PC the transactions have to be to potentially prevent progress of other transactions, before there is a time out due to network.

(Space for answer)

Q3. (15 Points)
Inspired by Dr. Kennedy's lecture on "Bloom Filters", the TAs decided to implement a converse of it and call it "Gloom Filters". Half way into the implementation, Dr. Kennedy said that filters with such a property is already present and ``caches" serve as one such example. The TAs and Dr. Kennedy get into an argument. So the question is do you agree whether the expected properties of "Gloom Filter" **mimic** the behaviour of a "cache". For this conceptual argument ignore the memory constraints of "cache" and the collision rate of hash functions in "Bloom Filter".

Justify your answer with a short explanation.

Property of Bloom Filter:

- Tells us whether an element is present in a set.
- False Positives are permitted but no False Negatives. (query returns either "possibly in set" or "definitely not in set").

Expected Property of Gloom Filter:

- Tells us whether an element is present in a set.
- False Negatives are permitted, but no False Positives (query always return "definitely in set" or may "not possibly be in set").

(Space for answer)