BD2 – July 7, 2016

Please feel free to answer this test in English, Italian, or any mixture

First Part (6 credits and 9 credits):

1. Consider a schema R(IdR, A, …, IdT*), T(IdT, B) and the following query

```
SELECT   R.A, Sum(T.B), Count(*),
FROM     T, R
WHERE    T.IdT = R.IdT and R.A < 100
GROUP BY R.A, T.IdT
```

Assume that R and T are stored as heap files. Primary keys are R.IdR and T.IdT, while R.IdT is a foreign key that refers to T.
Assume that an unclustered RID-sorted index is defined on R.IdT. Assume that every index on R has 20.000 leaves and every index on T has 4.000 leaves.
Assume the following table for the optimization parameters of tables R and T, and of index on R.IdT. If you need Cardenas formula Φ(n,k), approximate it with min(n,k).
Assume a buffer of 1000 pages.
Assume that every page is 4.000 bytes long, and that every attribute uses 4 bytes.

	NReg	NPag	NLeaf	NKey	Min	Max
R	10.000.000	100.000	20.000			
T	2.000.000	20.000	4.000			
Idx.R.A				100	0	10.000

a) Draw an access plan where IndexNestedLoop is used for the join, and T is the internal table (the one at the right hand side) and compute its cost
b) Draw an access plan where MergeJoin is used for the join, and compute its cost
c) Specify the conditions that allow group-by to be pushed below a join operator in the algebraic access plan
d) Do the conditions of (c) apply to this situation? That is, is it possible to execute the GroupBy before the Join?

2. Consider a schema R(IdR, A, …, IdT*), T(IdT, B) and the following query, and the parameters for R and T of question 1.

```
SELECT   *
FROM     T, R
WHERE    T.IdT = R.IdT
```

a) What is the cost of an access plan based on HashJoin, with T is the outer relation, under the following assumptions: B=100, B=1.000, B=10.000 (where B is the number of buffer pages

that can be used for the join)

b) What is the cost of an access plan based on MergeJoin, under the same assumptions.

3. Consider a query SELECT **X** -FROM-WHERE query (with no GROUP-BY clause). How is it computed the closure of set X in the result of the query (that is, how is it computed the set of attributes that are functionally determined by X in the result of the query)?

Second Part, BD2 and BSA (6 credits and 9 credits)):

4. Consider the following log content. Assume that the DB was identical to the buffer before the beginning of this log, and consider a undo-redo protocol

(begin,T1) (W,T1,A,1,20) (W,T1,B,1,20) (begin-ckp,{T1,T2}) (W,T1,A, 20, 40) (end-ckp) (commit,T1) (begin,T2), (begin,T3), (W,T3,B,20,40) (W,T2,C,1,60) (begin,T4) (W,T4,A,40,80) (commit,T2)

   a. Before starting this log, what was the content of A, B and C in the PS (Persistent Store)?
   b. Assume there was a crash at the end of the logging period. At crash time, what was the content of A, B and C in the buffer? What can be said about the content of A, B and C in the PS?
   c. At restart time, which transactions are undone? Which are redone?
   d. List the operations that are redone, in the order in which are redone
   e. After restart is finished, what is the content of A, B and C in the buffer?
   f. Undo and Redo are executed in the buffer or on the PS?
   g. After restart is finished, what is the content of A, B and C in the PS?
   h. Assume now a Redo-NoUndo protocol. In this case, what can be said about the content of A, B and C in the PS at crash time, that is, after the completion of (commit,T2)?

5. Assume that a system with no scheduler produces the following history, where we omit the commits:

   $r_1[A], w_2[B], w_3[A], r_3[B], r_2[C], r_1[C], c_1, w_2[B], c_2, r_3[A], c_3$

   a) Is this history serializable?
   b) Exhibit a history that may be produced by a strict 2PL scheduler if presented with the above operations in that order, assuming that each transaction commits immediately after its last operation. In case of deadlock, assume that a transaction is aborted and is restarted later
   c) Assume that the above list of operations is presented, in that order, to a scheduler that is based on Snapshot Isolation. Specify what would happen.

Second Part, BSA only (9 credits only):

6. Consider the following document that describes the location of the pieces of art of a Museum. Each piece is placed in a room for a period of time – between From and To. For simplicity, we assume here that From and To are just years where both extremes are included – from 2001 to 2003, for example, means tre years (2001, 2002, 2003).

Piece*
  @IdP
  Author
  Title
  Position*
    @IdRoom
    From
    To
Room*
  @IdR
  Name
  Building
  Capacity

a. Write an XQuery query that, for each room and each year, between 2000 and 2005, returns the names and the number of pieces in that room in that year (you can use (2000, 2001, 2002, 2003, 2004, 2005) to create an explicit list of numbers)) with the following structure

Room*
  Name
  OccupationByYear*
    Year
    NumberOfPieces
    Pieces*
      Author
      Title

b. Write and XQuery query that returns the @IdP of all pieces P such that there exists two Position subelement P1 and P2 for P such that the corresponding From-To sequences overlap (for example, P1 is 2001 – 2005 and P2 is 2003 – 2007 or is 2002-2004 or is 2000-2007)

7. Consider and RDF graph with classes Persons, Photos, Towns and predicates HasPerson, HasTown, HasFriend and ConnectedWith

HasPerson $\subseteq$ Photos $\times$ Persons
HasTown $\subseteq$ Photos $\times$ Towns
HasFriend $\subseteq$ Persons $\times$ Persons
ConnectedWith $\subseteq$ Persons $\times$ Persons

Formalize the following statements in OWL, paying extreme attention to the direction of the implication:

a. If there exists a chain C1 … Cn, with n>1, such that C1 HasFriend C2, C2 HasFriend C3 …, and Cn-1 HasFriend Cn, then ConnectedWith relates C1 and Cn (observe that C1 HasFriend C2 is already a chain with n>1)

b. The Transitive Closure of a relation R (TC(R)) is defined as the minimal transitive relation that includes R. Your formalization of (a) specifies that ConnectedWith is equal to TC(HasFriend) or does it specify a somehow different relationship?
c. Consider the two following theories, and specify whether they are equivalent:

    i. TransitiveObjectProperty(ConnectedWith)
    SubObjectPropertyOf(HasFriend ConnectedWith)

    ii. SubObjectPropertyOf(
        ObjectPropertyChain(HasFriend ConnectedWith)
        ConnectedWith)

d. Every photo has exactly one town
e. Assuming (d), formalize: If, and only if, one of the photos of person X has city Pisa, then X is in PhotoInPisa
f. Assuming (d), formalize: If, and only if, every photos of person X has city Pisa, then X is in PhotoOnlyInPisa

Consider an arbitrary RDF graph containing statements that relate persons, photos, and towns. Considering the fact that OWL has an open world semantics, and considering the directions of the implications, specify which of the following statements about a person X may be proved and which may never be proved. For those that may be proved give an example of and RDF graph supporting the proof, for those that may never be proved give an extremely brief reason for that

a. X is in PhotoInPisa
b. X is not in PhotoInPisa
c. X is in PhotoOnlyInPisa
d. X is not in PhotoOnlyInPisa