BD2 – 8/06/2015 – first part

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

1) Consider the following query

SELECT R.B, S:C, count(*), Avg(S.D)
FROM R, S
WHERE R.IdS = S.IdS and 10 < R.A < 20
GROUP BY R.B, S.C
ORDER BY S.C

Assume that R and S are stored as heap files. Assume that S.IdS is primary key and R.IdS is a foreign key that refers to S.
Assume that unclustered RID-sorted indexes are defined on attributes R.IdS, R.A, S.IdS, S.C.
Assume the following table for the optimization parameters of tables R and S, and of indexes on R.A and S.C.
The size of indexes R.IdS and S.IdS can be computed by assuming that each leaf may contain 500 RID's and ignoring the space needed to contain the value of IdS.
If you need Cardenas formula Φ(n,k), approximate it with min(n,k).

	NReg	NPag	NLeaf	NKey	Min	Max
R	200000	4000				
S	10000	1000				
Idx.R.A			440	100	1	100
Idx.S.C			24	1000	1	1000

a) Draw a logical access plan for the query
b) Compute NLeaf for Idx.R.IdS and Idx.S.IdS
c) Draw an efficient access plan for the query that uses no indexes and compute its cost
d) Compute the cost of an efficient access plan which is based on an IndexNestedLoop where R is the external relation, using all the indexes that are useful for this plan
e) Do you think that the plan in (c) is the most efficient plan for this query? Why?

2) Consider the following query on R(IdR, A, B, IdS*) and S(IdS, C) , where keys and foreign keys are defined as in exercise (1), where "X" is one attribute from R or from S (for example: R.IdS, R.A, S.C…)

SELECT DISTINCT ***X***, count(*)
FROM R, S
WHERE R.IdS = S.IdS and 10 < R.A < 20 and R.B = 30
GROUP BY R.IdS, R.B, S.C
ORDER BY S.C

a) May X be R.A? may it be R.B? More generally, which attributes may be substituted to X in order to produce a well-formed SQL query?

b) For each possible attribute that may be substituted to X, specify whether 'DISTINCT' is redundant or is necessary, explaining the answer

3) Consider the following log content. Assume that the DB was identical to the buffer before the beginning of this log.

(begin,T1) (W,T1,B,1,10) (begin,T2) (W,T2,C,1,20) (begin-ckp,{T1,T2}) (W,T2,C,20,30) (end-ckp) (begin,T3) (commit,T1) (begin,T4) (W,T4,A,1,4) (W,T3,B,10,20) (commit,T3)

a. Before starting this log, what was the content of A, B and C in the PS (Persistent Store)?
b. Immediately after the checkpoint completion, what was the content of A, B and C in the buffer? and in the PS?
c. 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? and in the PS?
d. At restart time, which transactions are put in the undo-list? Which in the redo-list?
e. List the operations that are undone, in the order in which they are undone
f. List the operations that are redone, in the order in which are redone
g. After restart is finished, what is the content of A, B and C in the buffer?
h. Undo and Redo are executed in the buffer or on the PS?
i. After restart is finished, what is the content of A, B and C in the PS?

BD2 – 8/06/2015 – second part

1) Assume that a system with no scheduler produces the following history:

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

a) Is this history serializable?
b) Is this history 2PL? If it is not, exhibit a history that may be produced by a strict 2PL scheduler if presented with the above operations in that order
c) Consider now a strict 2PL scheduler that adopts a wait-die deadlock detection protocol, and exhibit a history that may be produced by that scheduler. Assume that T_1 is the older transaction, followed by T_2 and T_3.

2) Consider an XML document that describes countries with their borders, populations, and, for each main spoken language, the name of the language and an estimate of people that speak that language

```
country*
    @id
    name
    population
    language*
            denomination
            numberofspeakers
    borderingCountry*
            @countryId
```

a) For each language, return (a) the total amount of people that speak that language (b) for each country where the language is the most spoken language, how many people speak that language and (c) for each country where the language is spoken, how many people speak that language (including all countries that already appear in "mostspokenin")

```
language*
    denomination
    mostspokenin*
            country
            numberofspeakers
    spokenin*
            country
            numberofspeakers
```

b) List the name of all pairs of countries such that the first is among the 'borderingCountry' of the second, but the second is NOT among the 'borderingCountry' of the first (of course, we are looking for inconsistent data)

3) Consider an ontology that describes countries and language, with classes Country, Language. MonoLanguageCountry, OnlyItalianSpoken with predicates:

IsSpoken: Country $\times$ Language
MostSpoken : Country $\times$ Language
CommonLanguage: Country $\times$ Country

Formalize the following assertions, trying not to confuse implications with double implications:

a) If in a country only one language is spoken, that the country is a MonoLanguageCountry
b) If there is a language that is spoken in both countries A and B, then A and B are connected by the CommonLanguage property
c) Is it possible to deduce from (b) that every country is connected to itself by CommonLanguage?
d) If, and only if, Italian is the only language spoken in a country, then the country belongs to OnlyItalianSpoken

Assume to have an RDF graph plus the four statements above. Keeping into account the fact that OWL is interpreted according to the Open World approach, is there a possibility that one may prove:

a) That a country belongs to OnlyItalianSpoken
b) That a country does not belong to OnlyItalianSpoken

Assume that a graph is given where England and Germany have no common language, and assume to add the following statement to the database:

ObjectPropertyAssertion(:CommonLanguage :England :Germany)

a) Does this statement generate a contradiction?
b) Is it possible to deduce from the above definition and this fact the symmetric fact:

ObjectPropertyAssertion(:CommonLanguage :Germany :England)