Advanced Database Systems, second intermediate test – 4 June 2018 – V1.1

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

1. 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,0,5) (begin,T2) (W,T2,B,0,20) (begin-ckp,{T1,T2}) (W,T1,A,5,10) (end-ckp) (begin,T3) (commit,T1) (W,T3,C,0,20) (commit,T3) (begin,T4) (W,T4,C,20,50) (commit,T4)

   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 put in the undo-list? Which in the redo-list?
   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? What is different between this answer and that of question (b)?

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

   $r_1[A], w_2[C], w_1[B], r_2[B], w_3[C], w_2[A], r_1[B], r_3[A]$

   a. Is this history c-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

3. Consider the following tables:

   Sales(Date,FKShop,FKCust,FKProd,UnitPrice,Q,TotPrice)
   Shops(PKShop,Name,City,State,Area)
   Customers(PKCust,Nome,FamName,City, State,Area)
   Products(PKProd,Name,SubCategory,Category,Price)

   With the following sizes:

   Sales: NRec: 100.000.000, Npag: 1.000.000;
   Shops: 500, 2; Customers: 10.000, 100; Products: 100.000, 10.000

   Assume a buffer of 1.000 pages. Consider the following queries:

```
SELECT P.PKProd, P.Name, P.Category, Sum(TotPrice)
FROM Sales S, Products P
WHERE S.FKProd=P.PKProd
GROUP BY P.PKProd, P.Name, P.Category
```

```
SELECT P.Category, Year(S.Date), Sum(TotPrice)
FROM Sales S, Customers C, Products P
WHERE S.FKCust=C.PKCust  AND   S.FKProd=P.PKProd
GROUP BY P.Category, C.Area
```

a. In the context of a traditional database, choose a physical organization to optimize both queries, and give a synthetic justification of your choice. Specify the primary organization of each table and which indices would you use. If the primary key of the Product and Customer relations must satisfy any property, specify this fact.
b. Show an access plan for the first query and compute its cost.
c. (Optional) How would the organization of (a) and the cost of (b) change if you added a condition AND Product.Category = 'Food', where 'Food' includes the 10% of the sales.

4. Answer the following questions. Please keep the answers short and WRITE IN YOUR BEST HANDWRITING.

   a. Column databases are better suited for OLAP applications than for OLTP applications. Why?
   b. (Really really optional) List some features of column databases, or some techniques that are used by some of them, that are related with the recent trends in the evolution of hardware architectures. Please be brief and write clearly.

5. Consider relations R(<u>K</u>,A, B)[NPag=1.000.000, NReg=100*NPag] and S(<u>K</u>,A, B)[NPag=100.000, NReg=100*NPag], both having K as their Key. Assume a Shared Nothing architecture, where the relations are horizontally partitioned and stored in nodes [n0,…,n99], and records are distributed using the function h1 applied to R.A for relation R, and the function h2 applied to S.B for relation S.
   a. Describe a hash-based algorithm to compute the intersection of R and S, that leaves each piece of the result on the disk of the node where it has been computed. Assume that every node has a buffer of 100 pages. When describing the actions in a node you may refer to standard hash based algorithms, such as hash-join or hash-based intersection.
   b. Assume that disks are able to read or write at a rate of 1.000 pages/s, and compute the cost of the I/O operations of each single node, including the initial read, the local intersection operation, the final local write. For each node specify how much data is read and written, and how much time is needed for that, assuming that, at each node, different i/o operations cannot overlap
   c. Optional: Assume that each node communicates with the network at a rate of 40 MBytes/s, that every page measures 4Kbytes, that the network itself has unlimited capacity, ignore the set-up time of the communications, and assume that, whenever a node sends a block of data the receiving node is ready to receive it. Compute the time of the communication phase, remembering that each node must both send and receive data, and that when a node is sending it cannot be receiving.

6. Consider a distributed DBMS composed by two separate nodes, US and EU, which stores the schema of exercise (1), where Area belongs to {'US','EU'). Assume that each node issues the following amount of operations per day.

Table	Queries per day	Updates per day

Sales	100	200
Shops	100	1
Customers	100	1
Products	100	5

Every query that is issued by a node n1 (US or EU) but also involves data that is *only* found on the other node incurs a cost QC, and every update that is issued by a node n1 (US or EU) but also involves data that is *also* found on the other node incurs a cost UC. The 90% of the US operations that involve a Sale regard a Sale whose shop has Area='US', and similarly for the operations on the EU node. The tables Customers and Products are uniformly accessed.

a. Define how the tables may be horizontally fragmented according to the above description. One fragment may include one entire table or a portion.
b. Specify how would you allocate the fragments on the two nodes, according to the usage model described, assuming QC and QU are similar. Ignore the issue of failure resilience.
c. (Optional) Compute the total communication cost per day according to the above description.