NTNU

Department of IDI

Examination paper for TDT4150 Advanced Databases

Academic contact during examination: Orestis Gkorgkas

Phone: 98832238

Examination date: 15-05-2019

Examination time (from-to): 15:00 - 19:00

Permitted examination support material: Simple calculator

Other information: There are 10 Questions to be answered. You can answer in English or Norwegian. You do not have to translate any terminology

Language: English

Number of pages (front page excluded): 6

Number of pages enclosed: 7

Checked by:

___7-5-2019___ Orestis Gkorgkas_

Date Signature

Informasjon om trykking av eksamensoppgave

Originalen er:

1-sidig X **2-sidig** ☐

sort/hvit ☐ **farger** X

skal ha flervalgskjema ☐

Students will find the examination results in Studentweb. Please contact the department if you have questions about your results. The Examinations Office will not be able to answer this.

Question 1 (20 %).

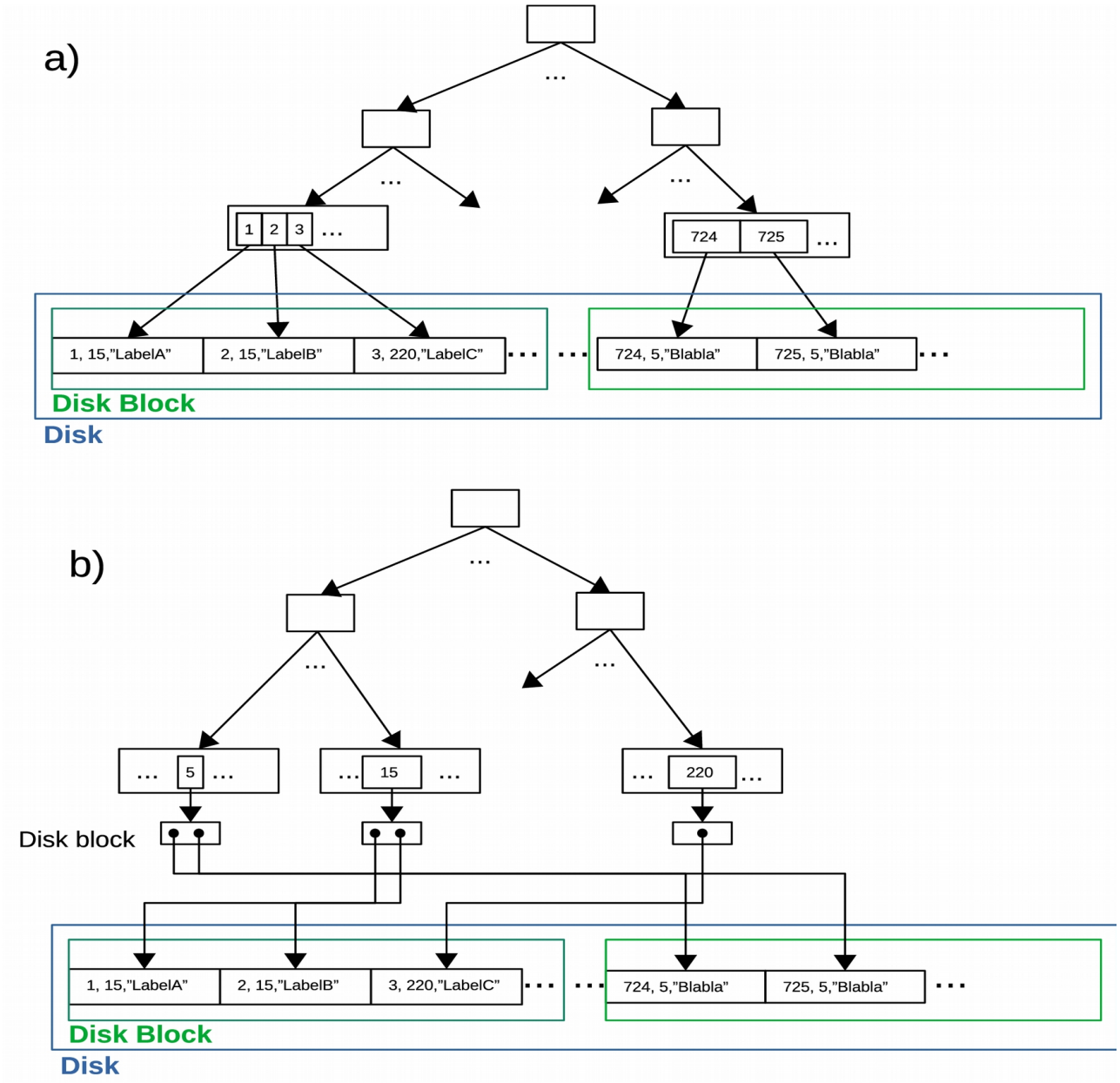
Let A be a relation in an Relational Database Management System (RDBMS) where:

- Table A has 3 fields (id,quantity, label) where id and quantity are integers and label is a string.
- id is the primary key of table A.
- Table A has 100 000 records.
- Each record has a size of 512 bytes.

The table is stored in a disk where each block has size 4096 bytes.

1.a) (1%)

Which of the following indices (a,b) is a primary index, and which is a secondary index.



1.b)(1%)

What is the blocking factor (tuples/block) for Table A?

1.c)(1%)

How many blocks does Table A occupy in the disk?

1.d)(2%)

How many I/Os (number of disk blocks to read/write) do we need to access the records with ids 1-8 if we are using a primary index. Assume that the non-leaf nodes of the primary index reside already in memory.

1.e)(3%)

Given that there are 10 000 records with quantity = 1, how many I/Os (number of disk blocks to read/write) do we need to perform for the following query in the worst case, if we have a secondary index on the attribute "quantity" ?

SELECT * FROM A WHERE A.quantity =1

1.f)(3%)

What is the average I/O cost of the following query if we perform a) a linear search, and b) a primary index search. Assume that the non-leaf nodes of the index reside already in memory. Explain shortly the result.

SELECT * FROM A WHERE A.id = c /* *c is a constant* */

1.g)(3%)

What is the average I/O cost of the following query if we perform a) a linear search, and b) a primary index search. Assume that the non-leaf nodes of the index reside already in memory. Explain shortly the result.

SELECT * FROM A WHERE A.id >= c AND A.id < c+40 /* *c is a constant* */

1.g)(3%)

What is the average I/O cost of the following query if we perform a) a linear search, and b) a secondary index search. Assume estimated attribute selectivity 1/10 and that the non-leaf nodes of the index reside already in memory. Explain shortly the result.

SELECT * FROM A WHERE A.quantity = c /* *c is a constant* */

1.h)(3%)

What is the average I/O cost of the following query if we perform a) a linear search, and b) a secondary index search. Assume estimated selectivity for the selection condition equal to 1/2 and that the non-leaf nodes of the index reside in memory. Explain shortly the result.

SELECT * FROM A WHERE A.quantity >= 100 and A.quantity <= 200

Question 2 (10%):

Let A be the table of Question 1 and B be a table with the same attributes (id,quantity,label). For both tables it holds that the record size is 512 bytes. Each disk block has a size of 4096 bytes.
Table A contains 100 000 records and table B contains 10 000 records.

Given the following joining query:

```
SELECT *
FROM A, B
WHERE A.label = B.label
```

2.a) (7%)

If neither A or B fit in memory what is the I/O cost (number of accessed disk blocks) of joining the tables with

a) Block Nested Loop (BNL)
b) Hash-join

You can ignore the cost of writing the result to the disk.

2.b) (3%)

If B fits in memory (but does not reside in memory yet) what is the cost of joining the tables using Hash-join. You can ignore the cost of writing the result to the disk.

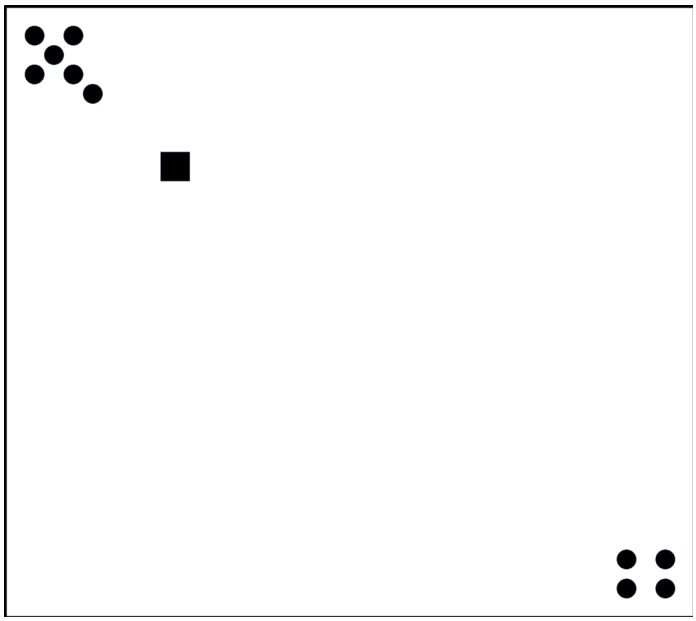
Question 3 (10%):

3.a) (5%)

What are the main advantages of an R-tree index when compared Grid-indices and Quad-tree indices?

3.b) (5%)

The following is a leaf node of an R-tree with M=10 and m=5. It already contains 10 items (circles). When we insert the new node (square) the node has to split. Given that the split is performed using the Quadratic-split algorithm, draw a possible split of the node.



Question 4 (5%):

Describe the structure of a leaf and a non-leaf node of a simple IR-tree.

Question 5: (15%)

Table A		
id	City	Price
1	Trondheim	1000
2	Drammen	3000
3	Trondheim	4000
4	Drammen	4000
5	Stavanger	5000
6	Oslo	6000
7	Oslo	7000
8	Bergen	7000
9	Stavanger	7000
10	Bergen	8000

Table B		
id	City	Price
a	Bergen	1000
b	Drammen	2000
c	Bergen	4000
d	Trondheim	4000
e	Oslo	4000
f	Bergen	6000
g	Bergen	6000
h	Oslo	7000
I	Trondheim	8000
j	Drammen	9000

Given the tables A, B above, we run the Rank-join algorithm for the following top-2 query.

SELECT * FROM A,B
WHERE A.city = B.city
ORDER BY (A.price + B.price) ASC
LIMIT 2.

Provide the top-2 result and explain when the algorithm will terminate.

Question 6 (15%):

Table A contains 5 records with 3 attributes (X,Y,Z) as shown below. For each attribute we have created and stored a list that is sorted according to the respective attribute.

Find the top-2 items with the highest score for the scoring function f = sum(X,Y,Z) using NRA.

- Assume **larger values are better** than smaller values.
- On each round of accesses (After accessing one tuple from each list) show the best and the worst score for each seen item.
- Be aware that you might find the top-2 result set without finding the exact score of all the items in the result-set.
- Show when the stopping condition has been satisfied.

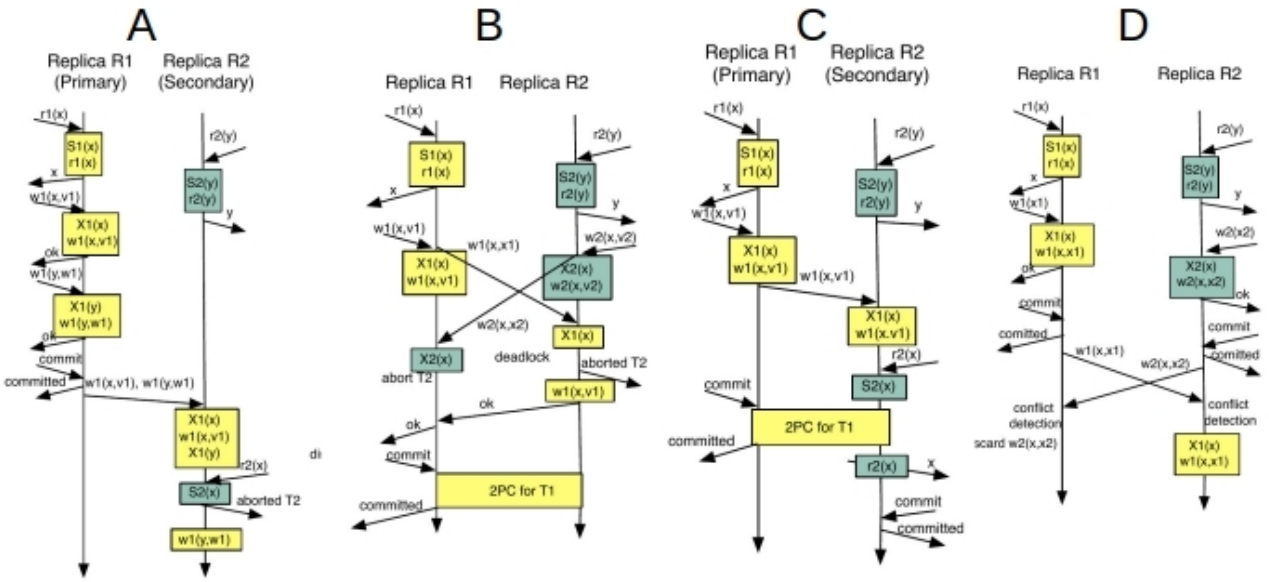
Table A			
id	**X**	**Y**	**Z**
a	3	1	2
b	2	5	5
c	3	4	1
d	5	4	4
e	1	3	1

Lists ordered by attribute value					
id	**X**	**id**	**Y**	**id**	**Z**
d	5	b	5	b	5
c	3	d	4	d	4
a	3	c	4	a	2
f	2	e	3	f	2
b	2	f	2	c	1
e	1	a	1	e	1

Question 7 (10%):

7.a) (3%)

Write the name of each of the following synchronization protocols.



7.b) (7%)

You are implementing two features for an e-commerce platform (such as komplett.no or ebay). The first feature is to give users the ability to write reviews for products they bought and to view reviews of other customers for the products they are browsing. The second feature is the payment/transaction feature where a customer is charged for a product he is buying. During this process a product is assigned to the user's account for delivery and the number of available products in the stock are updated. It is essential that no customer will ever be charged for a product that is not available and that a product will never be assigned to two customers. You have decided to use a distributed database for each of those features and that the data of the two features will be stored in different databases. Which synchronization protocol is more suitable for each of the features and why?

Question 8:

You are given:
the column: *attr*: <x,a,b,s,u,t,p,ø,j>
and the series of two consecutive cracking queries:

1. SELECT * FROM Table WHERE attr > 'i';
2. SELECT * FROM Table WHERE attr >= 'b' AND attr <= 't';

Describe:

1. Whether to use a 2-sided or 3-sided algorithm to crack the column for these queries, and why
2. The order of the column after each cracking query with the boundary between the different pieces explicitly shown
3. An ordered list of the (minimum) exchange/swap operations that transform the original column order into the final cracked order, as given by the 2-sided and/or 3-sided cracking algorithms.

Question 9 (5%):

9.a)

How do column oriented databases differ in comparison to typical database systems?

9.b)

Give 2 examples of how data could be compressed in a data column of such a system.

Question 10 (5%):

Describe shortly the 4 basic types of NoSQL databases.