German University in Cairo
Faculty of Media Engineering and Technology

Bar code

Exam Solution

CSEN 604: Databases II (BI)

Spring 2014 Semester

Dr. WaelAbouelsaadat

Date: June 1st, 2014

Duration: 3 hours

Do **not** turn this page until you have received the signal to start.
In the meantime, read the instructions below carefully.

This exam consists of 7 questions (numbered 1 to 7) on 13 pages *(including this one and an aid sheet in the last page)*, printed on one side of the paper. When you receive the signal to start, please make sure that your copy of the examination is complete.

Answer each question directly on the examination paper, in the space provided, and **use the reverse side of the page for rough work**. If you need more space for one of your solutions, use the reverse side of the page and indicate **clearly** the part of your work that should be marked.

1. __________________ / 15 (Index Structures)
2. __________________ / 20 (Result Size Estimation)
3. __________________ / 8 (I/O Cost Estimation)
4. __________________ / 20 (Concurrency Control)
5. __________________ / 20 (Logs & Recovery)
6. __________________ / 10 (SQL Transactions)
7. __________________ / 7 (SQL Authorization)

________________ / 100 TOTAL

Question 1. IndexStructures [*15 marks total*]

a) [*7 marks*]Assume that you have the following table:

Item

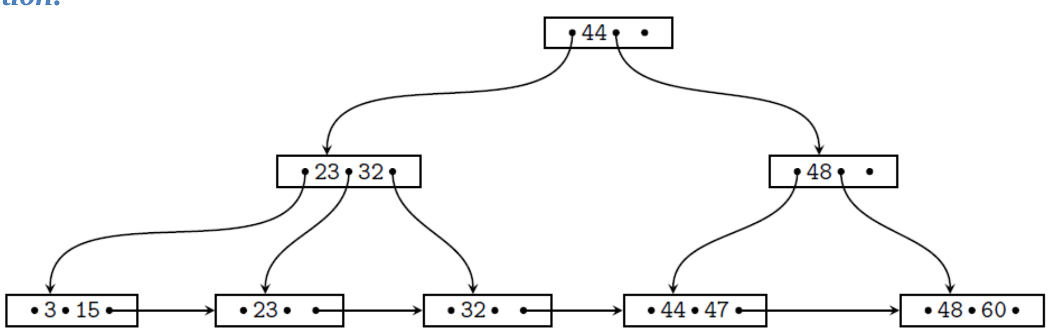
id	name	price
15	Shovel	13
44	Spate	23
3	Lawnmover	233
47	Lawnmover XL	499
48	Fertilizer	45
60	Sunflower seeds	3
32	Pine tree	299
23	Hop seeds	14

Create a B+tree for table *Item* on key *id* with *n* = 2 (up to two keys per node). You should start with anempty B+tree and insert the keys in the order shown in the table above. Write down the resulting B+treeafter each step.

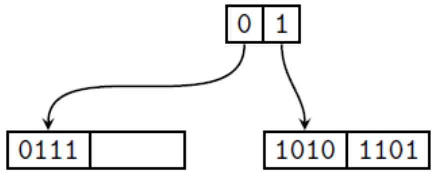
When splitting or merging nodes follow these conventions:

Leaf Split: In case a leaf node needs to be split, the left node should get the extra key if the keys cannotbe split evenly.
Non-Leaf Split: In case a non-leaf node is split evenly, the "middle" value should be taken from theright node.
Node Underflow: In case of a node underflow you should first try to redistribute and only if this failsmerge. Both approaches should prefer the left sibling.

Solution:



b)[*8 marks*] Consider the ***extensible Hash index*** shown below that is the result of inserting values 3, 4, and 5.



Each page holds two keys. Execute the following operations

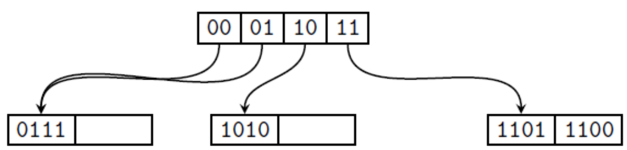
insert(0),insert(7),insert(6),insert(1),delete(5)

and write down the resulting index after each operation. Assume the hash function is defined as:

x	h(x)
0	1100
1	0001
2	0000
3	1010
4	1101
5	0111
6	1110
7	0000
8	1010

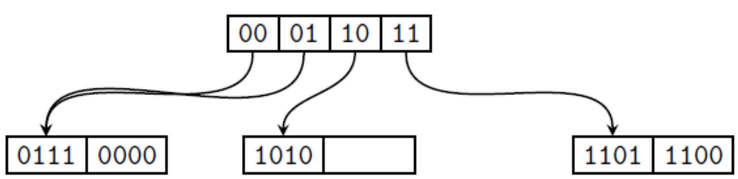
i. [*1 mark*] insert(0)

Solution:



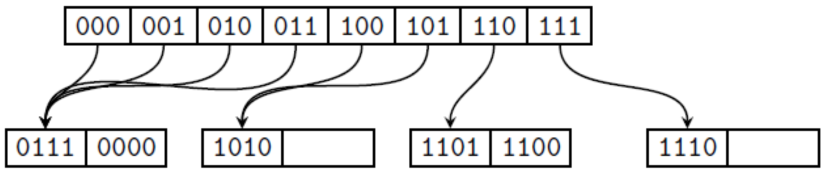
ii. [*1 mark*] insert(7)

Solution:



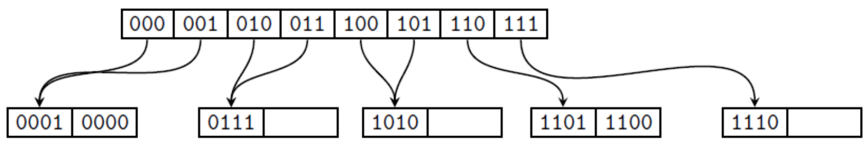
iii. [*2 marks*] insert(6)

Solution:



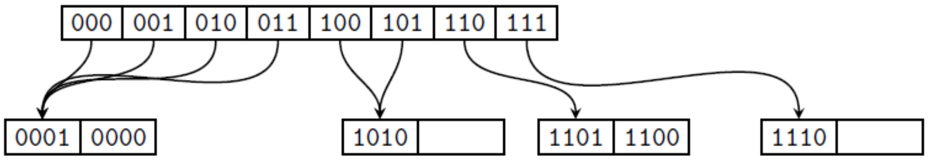
vi. [*2 marks*] insert(1)

Solution:



vii. [*2 marks*] delete(5)

Solution:



Question 2. Result Size Estimation [*20 marks total*]

Consider the following schema;

FruitJuice(brand, name, type, sugar)
Factory(brand, city, revenue)
Loc(city, state)

*FruitJuice.brand*is a foreign key toattribute*Factory.brand*.
Factory.city is a foreign key to attribute *Loc.city*

Given are the following statistics:

T(FruitJuice)	= 10,000	T(Factory)	= 400	T(loc)	= 2,000
V(FruitJuice, brand)	= 300	V(Factory, brand)	= 400	V(loc, city)	= 2,000
V(FruitJuice, name)	= 8,000	V(Factory, city)	= 50	V(loc, state)	= 50
V(FruitJuice, type)	= 10	V (Factory, revenue)	= 200		
V(FruitJuice, sugar)	= 10,000				

a) [*5 marks*] Estimate the number of result tuples for the query

$$\sigma_{type=\text{Mango}}(FruitJuice)$$

Solution:

T(q) = T(FruitJuice) / V(FruitJuice, type) = 10,000 / 10 = 1,000

b) [*5 marks*] Estimate the number of result tuples for the query

$$FruitJuice \bowtie Factory \bowtie city$$

Solution:

T(q) = T(FruitJuice) x T(Factory) x T(Loc) /
max(V(FruitJuice, brand), V(Factory,brand)) x max(V(Factory, city), V(Loc,city))

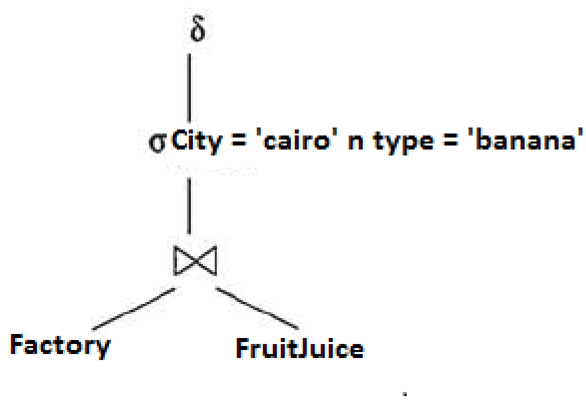
= 10,000 x 400 x 2000 / max(300,400) x max (50,2000) = 10,000

c) [*10 marks*] Suppose we want to find the list of factories making banana type in Cairo. The result set should contain unique tuples (i.e. filter out duplicates).

Note: students were informed during exam that there is a typo and should be type not brand.

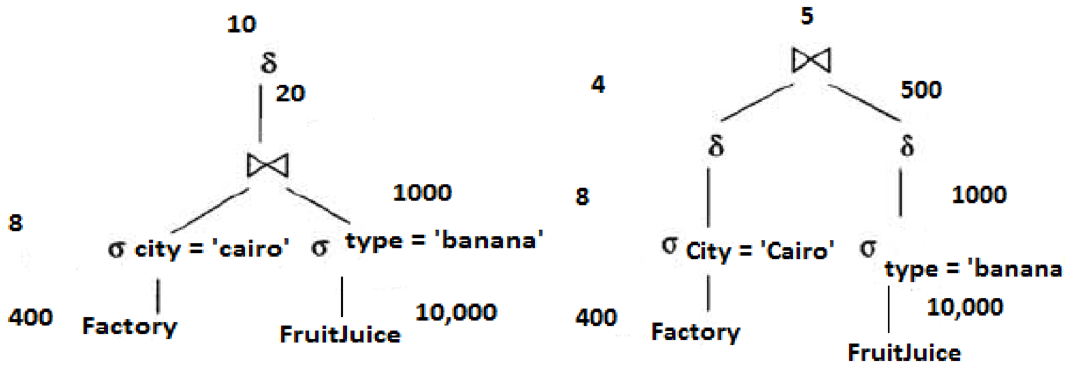
i. [*2 marks*] Draw a relational algebra tree that answer such a query.

Solution:



ii. [*4 marks*] Now, propose two different trees based on your answer to (i).Use equivalence rules whenever possible (mentioned in aid sheet).

Solution:



iii. [*2 marks*] Calculate the cost of each of the two trees in (ii)

Solution:

Check Tree.

v. [*2 marks*] Which tree would be considered as more efficient?

Solution:

The second one.

Question 3. I/O Cost Estimation [8 *marks total*]

Consider two relations *R* and *S* with *B*(*R*) = 3, 000, 000 and *B*(*S*) = 2, 000, 000. You have *M* = 101 memorypages available. Compute the minimum number of I/O operations needed to join these two relations using**block-nested-loop join**, **merge-join** (the inputs are not sorted), and **hash-join**. You can assume that thehash function evenly distributes keys across buckets. Justify you result by showing the I/O cost estimation foreach join method.

a) [*2 marks*] Nested-Loop Join

Solution:
S is smaller, thus, keep chunks of S in memory
(B(S) / M – 1) x (B(R) + min(B(S), (M-1))
= 20,000 x (3,000,000 + 100)
= 60,002, 000, 000 I/Os

b) [*3 marks*] Merge-join

Solution:
We can generate sorted runs of size 100 that means the number of sorted runs from R and S is low enough after two merge passes to keep one page from each run of both R and S in memory (3 runs for R and 2 runs for S). We need 3 merge passes for the sort, but can execute the last merge phase and join in one pass. 5 · (B(R) + B(S)) = 5 · (3,000,000 + 2,000,000) = 25,000,000 I/Os.

c) [*3 marks*] Hash-join

Solution:
We need 3 partitioning passes, because we can create 100 buckets. The bucket sizes of R and S after the third partitioning step will be 3 and 2. Thus, we can fit one bucket from R and one bucket from S into memory to join them. Cost is 3 · (B(R) + B(S)) = 3 · (3,000,000 + 2,000,000) = 15,000,000 I/Os.

Question 4.Concurrency Control [*20 marks total*]

a) [*8 marks*] Briefly explain why it is not necessarily desirable to execute multiple transactions as a serial schedule in a database system.

Solution:
The correctness principle tells us that every serial schedule will preserve consistency of the database state.

b) [*12 marks*] Consider the two separate schedules:

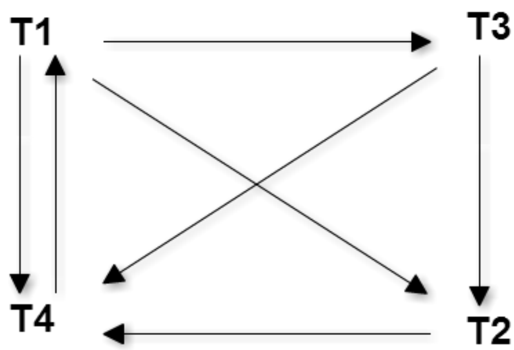
S_1: $w_1(X)$; $w_3(X)$; $w_2(X)$; $w_4(X)$; $r_4(Y)$; $w_1(Y)$

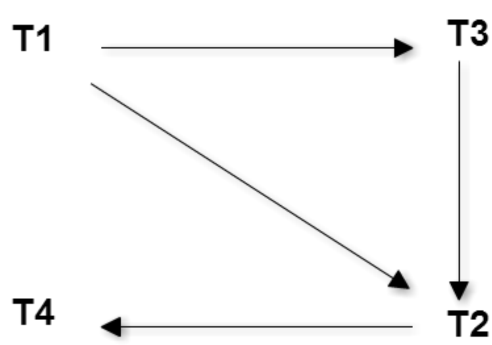
S_2 :$r_2(D)$; $r_1(E)$;$w_1(A)$; $r_3(A)$; $r_2(C)$;$w_2(A)$; $r_1(B)$;$w_1(B)$; $w_4(C)$; $w_3(B)$

Draw the precedence graph for schedule S1 and S2 (draw a separate graph for each). Is either of them conflict-serializable? Explain why.

Solution:
S1 is not conflict-serializable since the graph has a cycle.

S2 is conflict-serializable.





Question 5.Logs & Recovery [*20 marks total*]

Consider the following single transaction running in isolation:

T = r(A) r(B) w(A) w(B) r(C) w(C) w(D)

Say that the system crashes and when it restarts we have an opportunity to examine the log. A number of scenarios are given below. In each scenario, the logging method and log are given. For each scenario, determine which writes to data elements **must** be reflected in the database on disk, and which writes **must not** be reflected on disk. Indicate that a write to data element X is in one of these categories by writing "X" in the appropriate space, along with any other data elements that fit that category. If no data elements fit a category, simply write "empty" on the line.

a) [*5marks*] UNDO logging

log = <Start T><T, A, 5><T, B, 7><T, C, 2><T, D, 9><Commit T>

must: A,B,C,D　　　　must not: empty

b) [*5 marks*] REDO logging:

log = <Start T><T, A, 1><T, B, 2><T, C, 5><T, D, 0><Commit T>

must: empty　　　　must not: empty

c) [*5 marks*] UNDO/REDO logging with nonquiescent checkpointing:

log = <Start T><T, A, 5, 1><Begin Checkpoint (T)><T, B, 7, 2>

must: empty　　　　must not: C, D

d) [*5 marks*] UNDO/REDO logging with nonquiescent (fuzzy) checkpointing:

log = <Start T><T, A, 5, 1><Begin Checkpoint (T)><T, B, 7, 2>
<End Checkpoint><T, C, 2, 5><T, D, 9, 0><Commit T>

must: A　　　　must not: empty

Question 6.Transactions [*10 marks total*]

a) [5 *marks*] What are the four transactions isolation levels? Explain each in detail.

Solution:

• Read uncommitted: Transactions do not need to acquire any locks before reading data. Transactions may thus read data written by other transactions that have not yet committed. The value read may thus later be changed further or rolled-back. This problem is called the dirty read problem. This level of isolation also suffers from all the problems of the more restrictive isolation levels below.

• Read committed: Transactions must acquire shared locks before reading data. They may release these locks as soon as they read the data (short duration read locks). This level of isolation guarantees that the transaction never reads uncommitted data by other transactions. However, it doesn't ensure the data will not change until the end of the transaction. If a transaction reads the same data item twice, it can see two different values. This problem is called the non-repeatable read problem. This level of isolation also suffers from all the problems of the more restrictive isolation levels below.

• Repeatable read: Transactions must acquire long duration read locks on the individual data items that they read. This level of isolation provides all the guarantees of the read committed level. It also ensures that data seen by a transaction does not change until the end of the transaction: i.e., it provides repeatable reads. However, because locks are held on individual data items, transactions may experience the phantom problem. If a transaction reads twice a set of tuples that satisfy a predicate, it only locks the individual data items that match the predicate. If another transaction inserts a tuple that matches the predicate between the two read operations, that new tuple will appear as a result of the second read.

• Serializable: Transactions must acquire long duration read locks on predicates as well as on individual data items. This level of isolation protects against all the problems of the less restrictive levels. It ensures serializability.

b) [*5 marks*] Consider the following table Xbox_Games(name, price) and assume that these values already exist in the database

('ok_game', 40),
('good_game', 50),
('AWESOME_game', 60).

Given the following two transactions:

```
T1: BEGIN TRANSACTION
        S1: UPDATE  Xbox_Games SET price=22 WHERE name='ok_game'
        S2: INSERT INTO Xbox_Games VALUES ('BAD_Game', 0)
        S3: UPDATE Xbox_Games SET price=38 WHERE name='ok_game'
COMMIT;

T2: BEGIN TRANSACTION
        SET TRANSACTION ISOLATION LEVEL SERIALIZABLE
        S4: SELECT AVG(price) AS average_price FROM Xbox_Games
COMMIT;
```

Assume the above two transactions are hitting the DBMS at the same time. What are the possible values for average_price?

I. 50
II. 44
III. 37

a) I only.
b) II only.
c) I & II only.
d) I & III only.

Solution:

Answer is d) T2 must appear to be executed either completely before or completely after T1. If before, T2 produces the average of 40, 50, and 60 = 50. If after, T2 produces the average of 38, 50, 60, and 0 = 37.

Question 7.SQL Authorization [7 *marks total*]

Consider a database with relation R and users Alice, Bob, Carol, and Dave. Alice owns relation R. The following sequence of operations takes place:

Alice: GRANT SELECT ON R TO Bob WITH GRANT OPTION

Alice: GRANT SELECT ON R TO Carol WITH GRANT OPTION

Carol: GRANT SELECT ON R TO Bob WITH GRANT OPTION

Bob: GRANT SELECT ON R TO Dave WITH GRANT OPTION

Carol: GRANT SELECT ON R TO Dave

Dave: GRANT SELECT ON R TO Carol WITH GRANT OPTION

Alice: REVOKE SELECT ON R FROM Bob CASCADE

After these statements are executed, which of the following statements is true?

(a) Dave has the SELECT ON R privilege, but without the grant option.

(b) Dave has the SELECT ON R privilege with the grant option.

(c) Dave does not have the SELECT ON R privilege.

(d) Dave has the grant option for the SELECT ON R privilege, but does not have the privilege itself.

Solution:
(b) Dave retains the full privilege because of the route Alice -> Carol -> Bob -*gt; Dave.

End of Exam

Aid Sheet

$\sigma_{p1 \wedge p2}(R) = \sigma_{p1} [\sigma_{p2}(R)]$

$\sigma_{p1 \vee p2}(R) = [\sigma_{p1}(R)] \cup [\sigma_{p2}(R)]$

$\sigma_{p \wedge q}(R \bowtie S) = [\sigma_p(R)] \bowtie [\sigma_q(S)]$

$W = \sigma_{z=val}(R) \quad T(W) = \frac{T(R)}{V(R,Z)}$

$W = \sigma_{z \geq val}(R) \quad T(W) = f \times T(R)$

$$T(W) = [f \times V(Z,R)] \times \frac{T(R)}{V(Z,R)} = f \times T(R)$$

$W = R1 \bowtie R2 \qquad X \cap Y = A$

$$T(W) = \frac{T(R2)\ T(R1)}{\max\{V(R1,A), V(R2,A)\}}$$
