Cairo University
Institute of studies and statistical research
Computer and Information Sciences Department

Advanced Topics in Database Design & Implementation
CS604
June, 2012

Please answer all questions. Total pages: 2

Question 1: (23 points)

Consider the following universal relation R={A,B,C,D,E,F,G,H,I,J}, a set of functional dependencies F={{B,C}→{D},{C, E}→{F,G}, {B,E}→{H,I}, {B}→{J}, {I}→{A}} and a set of multivalued dependencies G={{J}—>>{H}, {J}—>>{A} } on the attributes of R.

a) Use the appropriate algorithm to decompose the relation R into 3NF with Dependency Preservation and Lossless join property.

b) Use the appropriate algorithm to decompose the relation R into 4NF relations with non-additive join property

c) consider the following decomposition for the relation R. D={R1,R2,R3,R4,R5}; R1={B,C,D,E}, R2={E,F}, R3={C,G}, R4={G,H,I}, R5={E,J,A}. Test whether the decomposition satisfies the lossless join property and the dependency preservation property.

Question 2: (22 points)

A *PARTS* file with *Part_no* as hash key includes records with the following *Part_no* values: 6,7,12,20,10,11,18,4,2.

a) Suppose that the file uses three buckets, numbered 0 to 2. Each bucket is one disk block and holds two records. Load these records into the file in the given order, using the *linear hashing* with the hash function h(K) = K mod 3.

b) Suppose that the field values are inserted in the given order in a B^+-tree of order p = 3 and $p_{leaf} = 2$; show what the final tree will look like.

c) Suppose that the field values are inserted in the given order in a B-tree of order p = 3; show what the final tree will look like.

Question 3: (11 points)

a) The following figure shows the log corresponding to a particular schedule at the point of a system crash for four transactions T1 , T2, T3, and T4. Suppose that we use the *immediate update protocol* with checkpointing. Describe the recovery process from the system crash. Specify which transactions are rolled back, which operations in the log are redone and which (if any) are undone, and whether any cascading rollback takes place.

	[start_transaction, T1]
	[read_item,T1,A]
	[read_item, T1,D]
	[write_item, T1,D, 20, 25]
	[commit,T1]
	[checkpoint]
	[start_transaction, T2]
	[read Item, T2,B]
	[writeitem, T2,B, 12,18]
	[start_transaction, T4]
	[read_item, T4,D]
	[write_item, T4,D, 25,15]
	[start_transaction, T3]
	[write_item, T3,C, 30,40]
	[read_item, T4,A]
	[write_item, T4,A, 30, 20]
	[commit,T4]
	[read_item, T2,D]
system crash →	[write_item, T2,D, 15, 25]

b) Determine whether each of the following transactions satisfies Basic, Conservative, Strict, or Rigorous Two-Phase Locking (justify your answer).

T1	T2	T3
read_lock (X); write_lock (Y); read_item (X); unlock (X); read_item (Y); Y:=X+Y; write_item (Y); unlock (Y); commit;	read_lock (X); read_item (X); write_lock (Y); unlock (X); read_item (Y); Y:=X+Y; write_item (Y); commit; unlock (Y);	read_lock (Y); read_item (Y); unlock (Y); write_lock (X); read_item (X); X:=X+Y; write_item (X); unlock (X); commit;

Question 4: (11 point)

Consider a transaction T with TS(T)=10 and an item X with read_TS(X) =11 and write_TS(X)=10. If transaction T issues a write_item(X) and a read_item(X) operations. Describe the execution of these operations according to the Basic Timestamp Ordering Algorithm.

Question 5: (11 points)

Consider schedules S1 and S2 below:

S1: rl(X); rl(Z); W1(X); r3(Z); r3(Y); W3(Z); r2(X); r2(Y); W2(Y); C2; W3(Y); C1;C3;

S2: rl(X); r3(Z); rl(Z); W1(X); r3(Y);C1; r2(X); W3(Z); W3(Y); C3; r2(Y); W2(Y);C2;

a) Determine whether they are *view equivalence* or not (justify your answer)

b) Determine whether each schedule is strict, cascadeless, recoverable, or nonrecoverable (justify your answer).

c) Draw the serializability (precedence) graphs for S1 and state whether the schedule is serializable or not. If a schedule is serializable, write down the equivalent serial schedule(s).

Question 6: (22 points)

a) There are many rules for transforming relational algebra operations into equivalent ones. State these rules.

b) Discuss the factors that influence physical database design.

Cairo University
Institute of studies and statistical research
Computer and Information Sciences
Department

Advanced Topics in Database Design & Implementation
CS604
January, 2012

Please answer all questions. Total pages: 4

Question 1: (16 points)

Consider the following universal relation R={A,B,C,D,E,F,G,H,I,J}, a set of functional dependencies F={{A,B}→{C},{B, D}→{E,F}, {A,D}→{G,H}, {A}→{I}, {H}→{J}} and a set of multivalued dependencies G={{I}—>>{H}, {I}—>>{J} } on the attributes of R.

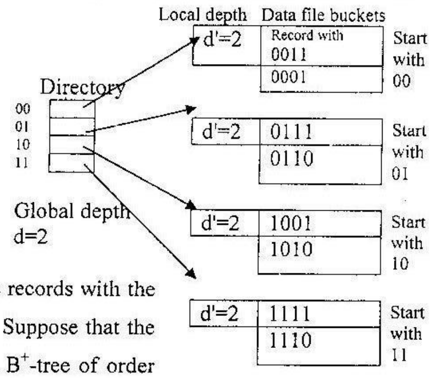
a) Use the appropriate algorithm to decompose the relation R into 3NF with Dependency Preservation and Lossless join property.

b) Use the appropriate algorithm to decompose the relation R into 4NF relations with non-additive join property

c) consider the following decomposition for the relation R. D={R1,R2,R3,R4,R5}; R1={A,B,C,D}, R2={D,E}, R3={B,F}, R4={F,G,H}, R5={D,I,J}. Test whether the decomposition satisfies the lossless join property and the dependency preservation property.

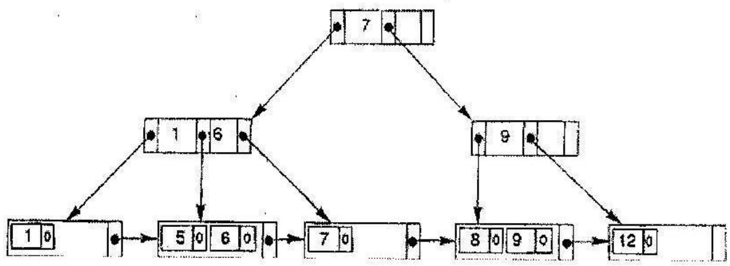
Question 2: (16 points)

a) Redraw the following extendible hashing after adding record with hash value start with 0101. Consider each bucket is one disk block and holds two records.



b) A *PARTS* file with *Part_no* as hash key includes records with the following *Part_no* values: 11,17,3,5,7,13,20,9,1. Suppose that the field values are inserted in the given order in a B^+-tree of order p=3 and p_{leaf} = 2; show what the final tree will look like.

c) Redraw the following B-tree after deleting 7 and 9.



Question 3: (8 points)

a) The following figure shows the log corresponding to a particular schedule at the point of a system crash for four transactions T1 , T2, T3, and T4. Suppose that we use the *immediate update protocol* with checkpointing. Describe the recovery process from the system crash. Specify which transactions are rolled back, which operations in the log are redone and which (if any) are undone, and whether any cascading rollback takes place.

b) Suppose that we use the deferred update protocol for the previous question. Show how the log would be different in the case of deferred update by removing the unnecessary log entries; then describe the recovery process, using your modified log. Assume that only REDO operations are applied, and specify which operations in the log are redone and which are ignored.

	[start_transaction, T1]
	[write_item, T1,D, 20, 25]
	[commit,T1]
	[checkpoint]
	[start_transaction, T2]
	[read_item, T2,B]
	[write_item, T2,B, 12,18]
	[start_transaction, T4]
	[read_item, T4,D]
	[write_item, T4,D, 25,15]
	[start_transaction, T3]
	[read_item, T3,B]
	[write_item, T3,B, 30,40]
	[read_item, T4,A]
	[write_item, T4,A, 30, 20]
	[commit,T4]
	[read_item, T2,D]
	[commit,T3]
system crash →	[write_item, T2,D, 15, 25]

Question 4: (8 point)

a) Consider a transaction T with TS(T)=5 and an item X with read_TS(X) =7 and write_TS(X)=5. If transaction T issues a write_item(X) and a read_item(X) operations. Describe the execution of these operations according to the Basic Timestamp Ordering Algorithm.

b) Determine whether transaction T1 satisfies Basic, Conservative, Strict, or Rigorous Two-Phase Locking (justify your answer).

c) Rewrite T2 to satisfy Strict two-phase locking and do not satisfy the Rigorous two-phase locking (justify your answer).

T2	T1
read_lock (X);	read_lock (X);
write_lock (Y);	read_item (X);
read_item (X);	unlock (X);
unlock (X);	write_lock (Y);
read_item (Y);	read_item (Y);
Y:=X+Y;	Y:=X+Y;
write_item (Y);	write_item (Y);
unlock (Y);	unlock (Y);
commit;	commit;

Question 5: (8 points)

Consider schedules S1 and S2 below:

S1: rl(X); r3(Z); rl(Z); W1(X); r3(Y);C1; r2(X); W3(Z); W3(Y); r2(Y); C3; W2(Y);C2;
S2: rl(X); rl(Z); r3(Y); W1(X); r3(Z); C1; W3(Z); W3(Y); r2(X); r2(Y); C3; W2(Y);C2;

a) Determine whether they are *view equivalence* or not (justify your answer)

b) Determine whether each schedule is strict, cascadeless, recoverable, or nonrecoverable (justify your answer).

Question 6: (14 points)

a) Can a nondense (sparse) index be used in the implementation of an aggregate operator? Why or why not?

b) Discuss the consideration for re-evaluating and modifying SQL queries.