CS411 Database Systems Fall 2016 Final Examination Solutions

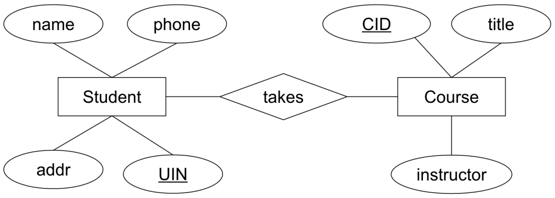
Problem 1

1. False. Age can be computed from birthdate, thus redundant.
2. 2. The root level has 1 node. The second level can hold 5 nodes. Each of the 5 nodes on the second level can index 4 records. So we can index 20 records with two levels already.
3. 2,000. The cost is $B(R)$ if R is clustered but no index exists on *department*, since we need to scan the entire relation.
4. False. The worst case scenario is when R is unclustered. It will take $T(R)$ to read R into memory, and $B(R)$ to sort it into runs. Then another $B(R)$ to merge. So the total cost is $T(R) + 2B(R)$.
5. False. All tuples with the same value of the key are clustered on as few blocks as possible.
6. True. Theta-join is equivalent to first a Cartesian product and then selection with the theta condition.
7. $YZ \to Y$ (Reflexivity rule)
   $X \to Y$ (Derived from $X \to YZ$ and $YZ \to Y$: Transitivity Rule)

   $YZ \to Z$ (Reflexivity rule)
   $X \to Z$ (Derived from $X \to YZ$ and $YZ \to Z$: Transitivity Rule)
8. False. In the absence of GROUP BY clause, SQL considers the whole relation as one group. For example, consider the query, SELECT COUNT(*) FROM Students HAVING COUNT(*) > 50. If number of students is less than 50, then this query shall return NULL. Otherwise, it will return the number of students.
9. False. The cost of the table-scan algorithm coupled with a filter step is $B(R)$, if R is clustered.
10. False. In an undo and a redo log, the commit records of transactions may be written at different times, and thus the results of recovery can be different. (You may refer to the practice problem in the corresponding class meeting.)
11. False. CPM means cost per mille (thousand) impressions.
12. While new technologies come and go, relational databases remain a strong foundation for data management.

Problem 2

1. The E/R diagram:



2. **E/R approach:**

   Company(<u>name</u>, addr)
   Product(<u>companyName</u>, <u>number</u>, type)
   Tablet(<u>companyName</u>, <u>number</u>, size)

   O-O approach:
   Company(<u>name</u>, addr)
   Product(<u>companyName</u>, <u>number</u>, type)
   Tablet(<u>companyName</u>, <u>number</u>, type, size)

   Null approach:
   Company(<u>name</u>, addr)
   Product(<u>companyName</u>, <u>number</u>, type, size)

Problem 3

1. CDA, CDB, CDE.

   Since C and D do not appear in any FD's right side, they must both be part of a key. Since CD is not a key, we then check CDA, CDB, and CDE– all of them are keys. We can stop here, since the remaining are all supersets of these keys.

2. R violates BCNF since A, BC and DE are not keys by they are left sides of the FDs.

3. Note: BCNF decomposition is not unique, one possible solution:

   R violates BCNF since A is not a key in the FD $A \to E$.
   We choose $A \to E$ and split the table into (AE), $(ABCD)$.
   $(ABCD)$ violates BCNF since BC is not a key, so we choose $BC \to A$ and split the table into (BCA), (BCD), both are in BCNF. So, the decomposition results in:

   $R1(A, E)$
   $R2(B, C, A)$
   $R3(B, C, D)$

4. No. In general, we cannot find a BCNF decomposition for R that is dependency preserving. For example if we decompose R into $R1(A, E)$, $R2(B, C, A)$ and $R3(B, C, D)$, $DE \to B$ will not be preserved.

Problem 4

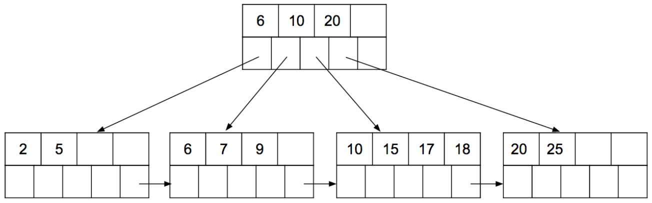
1.
   ```
   SELECT DepartmentHead
   FROM Departments D
   WHERE 0.5*NumberOfStudents <=
    (SELECT COUNT(DISTINCT E.NetId)
     FROM Students S, Enrollments E, Courses C
     WHERE S.NetId = E.NetId AND E.CRN = C.CRN AND
           S.Department = D.Department AND C.Department = D.Department)
   ```

2.
   ```
   SELECT Title, X
   FROM (SELECT CRN, COUNT(*) AS X
         FROM Enrollments E
         WHERE E.SCORE >
               (SELECT AVG(Score)
                FROM Enrollments E2
                WHERE E2.CRN = E.CRN)
         GROUP BY CRN) AS R, Courses C
   WHERE C.CRN = R.CRN
   ```

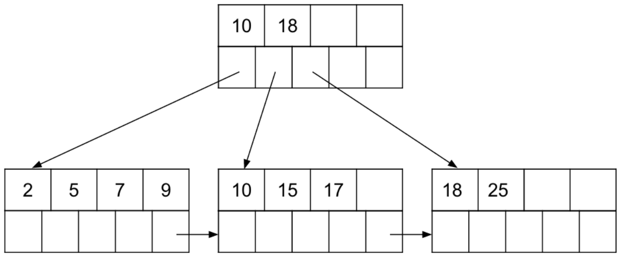
3.
   ```
   CREATE TRIGGER DeleteStudent
   AFTER DELETE ON Students
   REFERENCING OLD ROW AS OldTuple
   FOR EACH ROW
   DELETE FROM Enrollments WHERE NetId = OldTuple.NetId
   ```

Problem 5

1. (a) There are different ways to do this. You can get full credits as long you maintain B+tree properties.

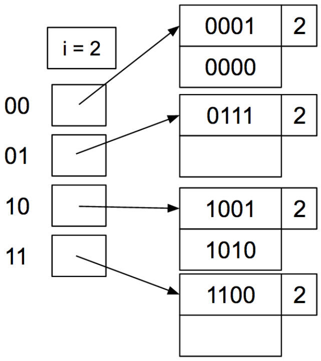


(b) There are different ways to do this. You can get full credits as long you maintain B+tree properties.



2. (a) The 4-bit hash values for 32 and 23 are 0000 and 0111 respectively.

   (b) The index becomes:



Problem 6

Given two clustered relations R and S, assume the size of the main memory is M blocks, the size of relation R is $B(R)$ blocks, and the size of relation S is $B(S)$ blocks. You should assume one buffer block is used to store outputs.

1. Given that one buffer block is used for outputs, we have: $10 + 10 * 15/(M - 2) <= 35$.
   Thus $M >= 8$.
   Since $B(R)$ is small in this case, you should receive full marks if you used $B(R) + CEILING(B(R)/(M - 2)) * B(S) <= 35$, and got $M >= 12$.
   Another acceptable solution is to use the larger relation S in the outer loop, and has $M >= 10$.

2. A one-pass algorithm is feasible since R can be completely stored in memory. Thus a two-pass algorithm is not necessary.
   $Cost = B(R) + B(S) = 25$.

3. Two-pass sort-merge join is not applicable since $MAX(B(R), B(S)) <= M^2$ is violated. Block-based nested-loop join is applicable. Two-pass hash-based join is not applicable since $MIN(B(R), B(S)) <= M^2$ is violated. So we can only use block-based nested loop join, and the cost is $B(R) + B(R) * B(S)/(M - 2) = 160$.

Problem 7

1. Here's the dynamic programming table.

Subquery	Size	Cost	Plan
AB	0.2*60*120 = 1440	0	(A)(B)
AC	0.2*60*20 = 240	0	(A)(C)
AD	0.2*60*50 = 600	0	(A)(D)
BC	0.2*120*20 = 480	0	(B)(C)
BD	0.2*120*50 = 1200	0	(B)(D)
CD	0.2*20*50 = 200	0	(C)(D)
ABC	0.2*240*120 = 5760	240	(AC)(B)
BCD	0.2*200*120 = 4800	200	(CD)(B)
ABD	0.2*600*120 = 14400	600	(AD)(B)

2. (a) To accommodate 50 different values of A, we need at least 50 tuples. 49 of these tuples can not satisfy the condition (because of the unacceptable values of A). Now, these 49 tuples can accommodate the 14 different values of B, 11 different values of C and 23 different values of D that also violate the condition. In other words, we can limit all unacceptable values of A, B, C and D within 49 tuples. The maximum selection size, therefore, is (1000 - 49) = 951 tuples.

   (b) We can have (i) $A = a_1$ and $B = b_2$ for all tuples where $C <> c_3$ and $D <> d_4$, and (ii) $C = c_3$ or $D = d_4$ for all tuples where $A <> a_1$ or $B <> b_2$. Thus, every tuple can satisfy the selection condition. The maximum selection size, therefore, is 1000.

   (c) We can have (i) $A = a_1$ for all tuples where $B <> b_2$, (ii) similarly, $B = b_2$ for all tuples where $A <> a_1$. Thus, every tuple can satisfy the selection condition. The maximum selection size, therefore, is 1000. Note that, projection does not affect number of tuples.

Problem 8

1.
```
INPUT(A)
READ(A, t1)
INPUT(B)
READ(B, t2)
IF (t1 > t2)
  WRITE(C, t2)
  WRITE(D, t1)
ELSE
  WRITE(C, t1)
  WRITE(D, t2)
OUTPUT(C)
OUTPUT(D)
```

2. We complete the table as follows:

Action	Undo Log	Redo Log	Undo-Redo Log
	⟨START T⟩	⟨START T⟩	⟨START T⟩
INPUT(A)			
READ(A, t1)			
INPUT(B)			
READ(B, t2)			
WRITE(C, t1)	⟨T, C, 6⟩	⟨T, C, 5⟩	⟨T, C, 6, 5⟩
WRITE(D, t2)	⟨T, D, 7⟩	⟨T, D, 10⟩	⟨T, D, 7, 10⟩
		⟨COMMIT T⟩	
OUTPUT(C)			
			⟨COMMIT T⟩
OUTPUT(D)			
	⟨COMMIT T⟩		

Problem 9 The soltion depends on if you choose to tell the truth or a lie or make a promise. Your choice.