RollNo: _______________ Name: _________________________

National University of Computer and Emerging Sciences, Lahore Campus

Course:	**Advanced Database Concepts**	**Course Code:**	**CS451**
Program:	**BS(Computer Science)**	**Semester:**	**Spring 2017**
Duration:	**3 hours**	**Total Marks:**	**50**
Paper Date:	**Tue 23-May-2017**	**Weight**	**40%**
Section:	**CS**	**Page(s):**	**7**
Exam:	**Final**		

Instruction/Notes: Scratch sheet can be used for rough work however, all the questions and steps are to be shown on this question paper. No extra/rough sheets should be submitted with question paper.
You will not get any credit if you do not show proper working, reasoning and steps as asked in question statements. Calculators are allowed.

Q1. *(4 points)* Assume a relation R (A, B, C) is given; R is stored as an ordered file (un-spanned) on non-key field C and contains 500,000 records. Attributes A, B and C need 5 byte of storage each, and blocks have a size of 2048 Bytes. Each A value occurs at an average 5 times in the database, each B value occurs 50 times in the database, and each C value occurs 50,000 times in the database. Assume there is no index structure exists.
Estimate the number of block fetches needed to compute the following queries (where C_a and C_c are integer constants):

a) SELECT B, C FROM R WHERE A = C_a;

b) SELECT B, C FROM R WHERE C = C_c;

RollNo: _______________ Name: __________________________

Q2. *(10 points)* Consider a disk with block size *B=512* bytes. A block pointer is *P=6* bytes long, and a record pointer is $P_R=7$ bytes long. A file has *r=100,000* EMPLOYEE records of fixed-length. Record length R is *115* bytes long and DEPTCODE field is *15* bytes long.

Suppose the file is ordered by the non-key field DEPTCODE and we want to construct a clustering index on DEPTCODE that uses block anchors (every new value of DEPTCODE starts at the beginning of a new block). Assume there are *500* distinct values of DEPTCODE, and that the EMPLOYEE records are evenly distributed among these values. Calculate:

a) The index blocking factor (bfr_i).
b) The number of first-level index entries (r1) and the number of first-level index blocks (b1).
c) The number of levels needed (x) if we make it a multi-level index.
d) The total number of blocks required by the multi-level index (b_i).
e) The number of block accesses needed to search for and retrieve all records in the file having a specific DEPTCODE value using the clustering index (assume that multiple blocks in a cluster are either contiguous or linked by pointers).

Q3. *(3 points)* Consider the student table:

<u>RollNo</u>	Name	Address	Gender	Age	Grade
1001	Khadija	Faisal Town	F	16	B
1002	Tahreem	Model Town	F	16	C
1003	Isbah	DHA	F	18	A
1004	Izaan	Model Town	M	18	B
1005	Alia	Faisal Town	F	20	A
1006	Tahreem	DHA	F	17	B
1007	Ismail	Johar Town	M	19	A
1008	Izaan	DHA	M	17	D

Find the selectivity (*sl*) of the condition to retrieve:

a) RollNo=1004

b) Gender='F'

c) Age=16

Q4. *(3 points)* Suppose that the most often used query on the Student database is:

SELECT StudentName, CourseCode, LetterGrade
FROM student S JOIN grade G ON S.RollNo=G.RollNo WHERE S.BatchId='2014';

On which column(s) would you create an index? Write down the column name(s) and one sentence why you choose the column(s).

RollNo: ________________ Name: ___________________________

Q5. *(4+6= 10 points)* Consider the schedule:

Op #	**T1**	**T2**	**T3**
1	Read(A)		
2		Read(C)	
3			Read(A)
4		Write(C)	
5			Read(B)
6	Read(B)		
7		Write(B)	

a) Draw the precedence graph of the schedule given above. In case it is conflict-serializable then list down all ***the equivalent serial schedules.*** Justify your answer.

b) Show that the schedule will be accepted/rejected by the below protocols. Provide proper reason and show your working.

i) The basic two-phase locking protocol (add locks to the transactions)

ii) The timestamp-ordering protocol (you have T1 < T2 < T3)

Q6. *(3+4+3= 10 points)* Figure below shows the log corresponding to a particular schedule at the point of a system crash for five transactions. Suppose that we use the immediate update (undo/redo) protocol with *checkpointing*. Describe the recovery process from the system crash.

Assume that the initial values of items are *A=100*, *B=200*, and *C=300*. Isolation level of all transactions is *READ COMMITTED*.

a) Identify which transactions need undo/ redo operation(s)?
b) Specify which operations in the log are redone (in correct order) and which are undone.
c) Write down the values of items A, B, and C after system recovery.

```
[start_transaction, T1]
[read_item, T1, B, 200]
[start_transaction, T2]
[read_item, T2, A, 100]
[write_item, T2, A, 100, 50]
[read_item, T2, B, 200]
[write_item, T2, B, 200, 120]
[commit, T2]
[start_transaction, T3]
[read_item, T1, A, 50]
[write_item, T1, A, 50, 20]
[read_item, T3, C, 300]
[write_item, T3, C, 300, 250]
[checkpoint]
[commit, T3]
[start_transaction, T4]
[read_item, T4, C, 250]
[start_transaction, T5]
[read_item, T5, C, 250]
[write_item, T5, C, 250, 210]
[commit, T5]
System crash
```

Q7. *(10 points)* Consider the bank database, and the following SQL query:

Customer (custID, custName, cnic, birthDate, address, …   )
Account(accNo, custID, accTitle, accType, openingDate, …   )
Transaction(tID, accNo, transType, amount, transDate, …   )

```
   SELECT C.cnic, A.accNo, A.Title, T.noOfTrans
 FROM customer C JOIN account A ON C.custID=A.custID JOIN (SELECT accNo, COUNT(*) AS noOfTrans
                                                  FROM transaction GROUP BY accNo) T ON A.accNo=T.accNo
```

Write an efficient relational-algebra expression that is equivalent to this query and draw the optimal query plan for this query.