14 Jan 2021

EAST WEST UNIVERSITY
Department of Computer Science and Engineering
B.Sc. in Computer Science and Engineering Program
Final Assessment (Online), Fall 2020 Semester

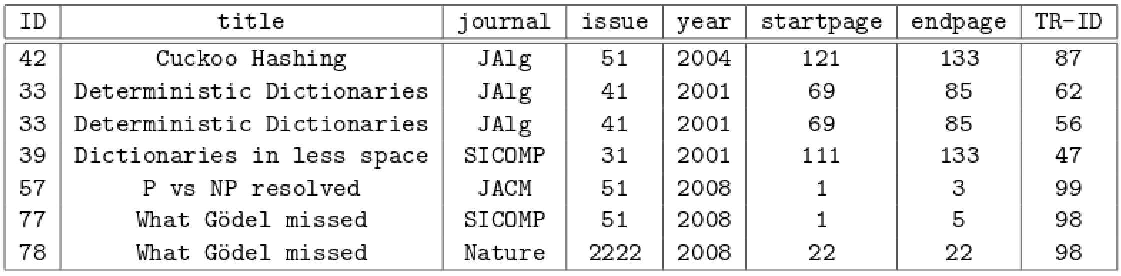
Course:	**CSE 302 Database Systems**
Instructor:	**Mohammad Rezwanul Huq, PhD, Associate Professor, CSE Department**
Full Marks:	**40 (20 will be counted for final grading)**
Time:	**1 Hour and 30 Minutes (including answer uploading time)**

Note: There are **6** (**SIX**) questions, answer ALL of them. Course Outcome (CO), Cognitive Level and Mark of each question are mentioned at the right margin.

1. Consider the following relation: [CO2, C3, Mark: 10]

Articles (ID, title, journal, issue, year, startpage, endpage, TR_ID)

It contains information on articles published in scientific journals. Each article has a unique ID, a title and information on the journal where it got published (name of journal, issue number, year and pages). Also, if results of an article is previously published in an internal paper known as a 'technical report' (TR), the id of this technical report can be specified. The following is an instance of the relation.



ID	title	journal	issue	year	startpage	endpage	TR-ID
42	Cuckoo Hashing	JAlg	51	2004	121	133	87
33	Deterministic Dictionaries	JAlg	41	2001	69	85	62
33	Deterministic Dictionaries	JAlg	41	2001	69	85	56
39	Dictionaries in less space	SICOMP	31	2001	111	133	47
57	P vs NP resolved	JACM	51	2008	1	3	99
77	What Gödel missed	SICOMP	51	2008	1	5	98
78	What Gödel missed	Nature	2222	2008	22	22	98

Answer the following questions based on the above relation.

a) How can ‘update anomalies' occur in relation Articles? Explain with an example.

b) What is the primary key of Articles relation?

c) Perform normalization into BCNF over the given relation and state the resulting relations. Justify the steps you have taken in detail.

2. Consider the following instance of the relation r. [CO2, C4, Mark: 6]

A	**B**	**C**	**D**	**E**	**F**
5	4	9	2	2	1
1	1	1	*	*	1
5	4	2	*	3	7
0	0	0	0	0	0
0	1	0	*	0	1
1	0	1	1	1	1
5	4	2	5	*	1
5	4	*	2	2	7

The values marked with * indicate that these values have been deleted mistakenly. You know that the relation r has the following functional dependencies.

C → D
AC → E

Find the values which were deleted mistakenly based on these two FDs. Put the values in their place, mark them with a circle and write the complete relation.

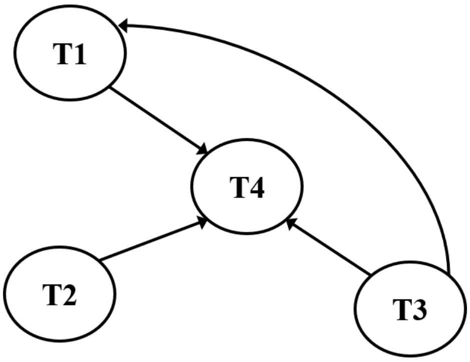
3. Consider the following schedule S. [CO3, C4, Mark: 10]

T1	**T2**	**T3**	**T4**
		R(X)	
	W(X)		
			W(X)
		W(Y)	
		R(Z)	
R(X)			
R(Y)			
	commit		
		commit	
commit			
			commit

Answer the following questions.
a) Is the given schedule S conflict-serializable? Justify your answer. If yes, find the serial schedule to which it is conflict equivalent with?
b) Is the given schedule S recoverable? Justify your answer.
c) Is the given schedule S cascadeless? Justify your answer. If no, rewrite the schedule without changing their precedence graph so that it becomes cascadelss schedule.

4. Consider the precedence graph of a concurrent schedule as shown in the following figure. [CO3, C6, Mark: 4]

The precedence graph has no cycle and therefore, it is a conflict-serializable schedule. **List** the possible serial schedules which are conflict-equivalent with the given concurrent schedule.



5. Consider the following instance of the Product table. [CO3, C3, Mark: 5]

Product

Product_no	**Product_name**	**Product_type**	**Unit_price**
P-1	Polo T-shirt	Clothing	1000.00
P-2	Men’s Watch	Fashion	2500.00
P-3	Gold Necklace	Jewellery	50000.00
P-4	Saree	Clothing	5000.00
P-5	Sunglass	Fashion	3000.00
P-6	Gold Ring	Jewellery	20000.00

An index helps queries to execute faster. One of the users wants to execute the following query.

```
SELECT Product_name, Unit_price
FROM Product
WHERE Product_type = ’Fashion’;
```

Answer the following questions.
a) On which attribute, you should create an index to execute the above query faster? Write the SQL statement to create the index.
b) What type of index you have created in question 5.(a)? Draw the index structure accordingly.

6. Assume that there is a relation r with 10000 tuples and 20 tuples can fit into 1 block (1 block = 4 KB). Also assume that, you have created a primary sparse index, which contains two search-key values (the smallest and the median) from each block of data file. 40 index entries can fit into 1 block. The main memory can hold 50 blocks. [CO3, C4, Mark: 5]

Answer the following questions.
a) What is the size of the data file in KB?
b) What is the size of the index file in KB?
c) Do you need multi-level indexing in this case? Justify your answer.