CS315: Principles of Database Systems, IIT Kanpur				**Endsem** (25 Nov 2024)
Name	DEEBO			**40 marks**
Roll No	24007	Dept.	DOGE	Page **1** of **4**

Instructions:

1. This question paper contains 2 pages (4 sides of paper). Please verify.
2. Write your name, roll number, department above in **block letters neatly with ink**.
3. Write answers neatly **with a blue/black pen** and **not pencil**. Don't overwrite/scratch MCQ.
4. **Hardcoding** attempts will get **no credit**.
5. For **questions marked with** ***, grading will be done by firing your answer as a **query to an SQLite DB**. If the query takes **too long to execute**, you may get **no marks** even if it (eventually) produces the right response.

Q1. Write **T** or **F** for True/False **in the box**. Also, **give justification**. **(6 x (1+3) = 24 marks)**

1	When entity sets with composite attributes e.g., *name*(*firstname*, *lastname*) are represented as SQL schemata, flattening (creating multiple columns) is done to satisfy 3NF. If **T**, give an example. If **F**, explain which NF is satisfied by flattening.	F

Flattening is done to satisfy 1NF since it tries to ensure that attributes are atomic. For example, a single *name* attribute would potentially be non-atomic and may present difficulties in executing queries, say sorting students by last name instead of first name.

2	If a relation doesn't satisfy 1NF, it cannot satisfy BCNF. If **T**, give a proof. If **F**, give the schema of a counterexample relation satisfying BCNF but not 1NF and explain.	F

Consider a table stu(rollno, name) where the name attribute is non-atomic as in part 1 above. This table has only one non-trivial FD, namely rollno → name and rollno is indeed a superkey. Thus the table is in BCNF but not in 1NF due to the presence of a non-atomic attribute name.

3	For a relation $R(A, B, C)$, if A isn't a superkey, then R cannot satisfy the FD $A \to B$. If **T**, give a proof. If **F**, fill-in the table below giving a counterexample. Your counter example must use exactly 3 rows and the cells must contain only integers (no nulls).	F

If **T**, give proof here

In the counterexample table, A isn't a superkey (it isn't unique) yet the table satisfies $A \to B$.

If **F**, give counterexample here

A	B	C
1	2	3
1	2	3
2	4	3

4	For a relation R satisfying the dependency $X \to Y$ where $X, Y \subseteq R$, if Y is a superkey then so is X. If **T**, give a proof using Armstrong's axioms. If **F**, give a counter example using 3 columns (specify what is X, Y), 3 rows, integer values in cells and no nulls.	T

If **T**, give proof here

Since Y is a superkey, we get $Y \to R$. Combining with $X \to Y$ using transitivity gets us $X \to R$ which means that X is a superkey as well.

If **F**, give counterexample here

A	B	C

5	For an SQLite relation $R(A, B, C)$ satisfying the dependency $A \to B$, the query SELECT COUNT(DISTINCT A) FROM R; must return the same value as the query SELECT COUNT(DISTINCT B) FROM R; If **T**, give a proof. If **F**, fill-in the table below giving a counterexample using exactly 3 rows, only integers in cells and no nulls.	F

If **T**, give proof here

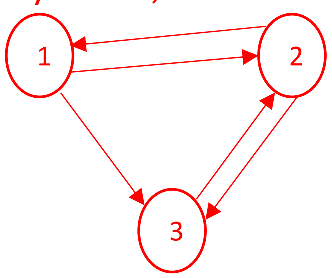
The counterexample table satisfies $A \to B$ but there are 2 distinct values for the attribute A but only 1 distinct value for the attribute B.

If **F**, give counterexample here

A	B	C
1	1	3
2	1	3
2	1	3

6	$R_i(X) \equiv$ transaction i is reading variable X, $W_j(Y) \equiv$ transaction j is writing to Y. Is the following schedule with 3 transactions, 3 variables conflict serializable? If **T**, show swaps and serialize. If **F** justify, say by drawing precedence graph with a cycle. $R_2(A), W_3(C), R_1(A), R_2(B), R_1(B), R_2(C), R_3(B), W_1(A), W_3(A), W_2(B)$	F

The operations on variable A (in order) are $R_2(A), R_1(A), W_1(A), W_3(A)$ which give rise to the precedences $2 \to 1, 2 \to 3, 1 \to 3$ due to read-write and $1 \to 3$ due to write-write conflicts.
The operations on variable B (in order) are $R_2(B), R_1(B), R_3(B), W_2(B)$ which give rise to precedences $1 \to 2, 3 \to 2$ due to read-write conflicts. There are no write-write conflicts on B.
The operations on variable C (in order) are $W_3(C), R_2(C)$ giving the precedence $3 \to 2$ due to a read-write conflict. There are no write-write conflicts on C.
Thus, the set of precedences are $2 \to 1, 2 \to 3, 1 \to 3, 1 \to 2, 3 \to 2$. This contains multiple cycles namely $1 \leftrightarrow 2, 2 \leftrightarrow 3$ and $2 \to 1 \to 3 \to 2$. Thus, the schedule is not conflict serializable.



Q2. Consider a relation $R(A,B,C,D,E)$ satisfying the FDs $A \to BC, CD \to E$ and $AD \to E$ that was decomposed into 3 relations $S(A,B), T(B,C,D), U(C,D,E)$. Execute the Chase algorithm giving only the initial and final states of the tableaus (exactly 3 rows in tableau). **(4+4+2=10 marks)**

Initial state of tableau before starting Chase

A	B	C	D	E
a	b	c1	d1	e1
a2	b	c	d	e2
a3	b3	c	d	e

Final state of tableau after finishing Chase

A	B	C	D	E
a	b	c1	d1	e1
a2	b	c	d	e
a3	b3	c	d	e

You may have noticed that the Chase algorithm indicates that the decomposition is not guaranteed to be lossless? However, Chase's failure does not mean that all decompositions are guaranteed to be lossy. Give an example of the relation R that demonstrates this. Your example must use exactly 2 rows, only integer values in the cells and no nulls. Also show the relations S, T, U that result from decomposing your relation R. Your S, T, U tables must also contain no nulls and exactly two rows. Your example relation R must satisfy all 3 dependencies i.e., $A \to BC, CD \to E$ and $AD \to E$. Your table R must result in a lossless decomposition i.e., $R = S \bowtie T \bowtie U$ where $\bowtie$ is the natural join.

Relation R that decomposes losslessly even if Chase failed

A	B	C	D	E
1	1	1	1	1
2	2	2	2	2

Relation S = SELECT A, B FROM R;

A	B
1	1
2	2

Relation T = SELECT B, C, D FROM R;

B	C	D
1	1	1
2	2	2

Relation U = SELECT C, D, E FROM R;

C	D	E
1	1	1
2	2	2

Deeba feels that Chase will succeed if R satisfies just one more FD in addition to the three FDs it already satisfies. Fill in boxes (one or more) next to FDs that will prove Deeba right. For example, select $A \to BCDE$ if the set of 4 FDs, namely $A \to BCDE, A \to BC, CD \to E$ and $AD \to E$ cause Chase to succeed.

- [ ] $A \to B$
- [x] $B \to A$
- [ ] $A \to BCDE$
- [ ] $BCDE \to A$

Q3*.** Given a table $R(A, B, C)$ with all columns taking integer values and no nulls anywhere, Deebo wants to write a conditional SQLite query (of the kind given on the right) to print **YES** if the table satisfies the dependency $AB \to C$ and **NO** otherwise. Complete the query by giving the Boolean expression for the **YES** case

```
SELECT CASE
  WHEN [Boolean expression]
    THEN 'YES'
  ELSE 'NO'
END;
```

Give only the Boolean expression and not the entire query. *Hint: put parenthesis around statements if comparing their results.* ***Note: evaluation will be purely DB query-based.*** (**6 marks**)

```
NOT EXISTS (
  SELECT *
  FROM R AS R1, R AS R2
  WHERE R1.A = R2.A
    AND R1.B = R2.B
    AND R1.C <> R2.C
)
```