CLASS TEST

S.No.: 01SKCS_ABCDEF_280425

MADE EASY

Leading Institute for IES, GATE & PSUs

Delhi | Bhopal | Hyderabad | Jaipur | Pune | Kolkata

Web: www.madeeasy.in | **E-mail:** info@madeeasy.in | **Ph:** 011-45124612

DBMS

COMPUTER SCIENCE & IT

Date of Test : 28/04/2025

ANSWER KEY

1.	(c)	7.	(b)	13.	(a)	19.	(c)	25.	(a)
2.	(b)	8.	(c)	14.	(a)	20.	(b)	26.	(b)
3.	(d)	9.	(b)	15.	(a)	21.	(a)	27.	(a)
4.	(d)	10.	(d)	16.	(a)	22.	(d)	28.	(c)
5.	(b)	11.	(d)	17.	(a)	23.	(c)	29.	(d)
6.	(d)	12.	(b)	18.	(a)	24.	(b)	30.	(c)

DETAILED EXPLANATIONS

1. (c)

Option (c) is (X ∪ Y) – [(X – Y) ∪ (Y – X)]

Let's take union as + and intersection as (·)

Now,

$\Rightarrow$ (X + Y) – [(X – Y) + (Y – X)]

$\Rightarrow \quad (X+Y)\cap\left[\overline{(X-Y)+(Y-X)}\right] \quad \left[\because\ R-S=R\cap\overline{S}\right]$

$\Rightarrow \quad (X+Y)\cap\left[\overline{(X\cap\overline{Y})+(Y\cap\overline{X})}\right]$

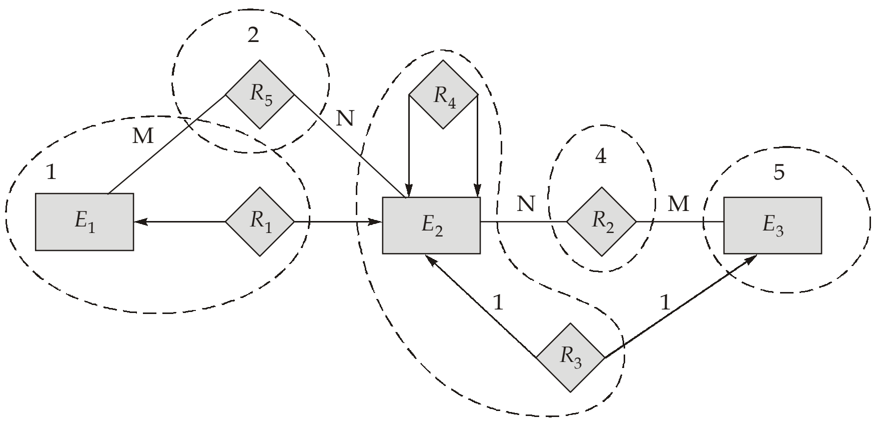
$\Rightarrow \quad (X+Y)\ \cap\left[(\overline{X}\cup Y)\cdot(X\cup\overline{Y})\right]$

$\Rightarrow \quad (X+Y)\cap\left[\overline{X}\,\overline{Y}+XY\right]$

$\Rightarrow$ XY + XY

$\Rightarrow$ X Y which is X ∩ Y

2. (b)



Thus total 5 tables are required.

3. (d)

A → C	B → C	A → B	B → A	A → B
A → B	B → A	A → C	B → C	A → C
B → A	A → B	B → A	C → A	B → A
B → C	A → C	B → C	C → B	B → C
C → A	C → A	C → A	A → B	C → A
C → B	C → B	C → B	A → C	C → B
⇓	⇓	⇓	⇓	⇓
{A → B, B → C, C → A}	{B → A, A → C, C → B}	{A → C, B → C, C → AB}	{B → AC, A → B, C → B}	{A → BC, B → A, C → A}
①	②	③	④	⑤

4. (d)

R(U V W X Y Z)

{U → V, W → X, Y → Z} Candidate key = {UWY}

U → V,

W → X Partial dependencies

Y → Z

So decomposition R_1(UWY), R_2(UV), R_3(WX) and R_4(YZ) is lossless and dependency preserving decomposition in BCNF.

5. (b)

Due to write operation, if the transaction will execute in non-serial manner, it may lead to non conflict serializable schedule.

6. (d)

An exclusive lock is released at the end to avoid cascading rollback problem and to ensure recoverability.

7. (b)

There is no instructor with the name "Samuel" in course.

So, all distinct sid's will be returned which are S_1 and S_2 only.

8. (c)

Undo: Using the log record it sets the data item specified in log record to add value.

Redo: Using the log record it sets the data item specified in log record to new value.

In the above log record transactions T_0 and T_1 are committed before the crash. So redo of T_0 and T_1 will be done and T_2 is not committed before the crash so undo (T_2) will be done.

9. (b)

- The key point in this question is not to update the balance twice.

Consider the following scenario:

Case-1:

Let balance = 19000 and t_1 is executed first then balance = 19000 × 1.7 = 32300.

Now t_2 will also be allowed to execute because balance is greater than 20000 which is wrong.

Case-2:

Now let t_2 execute first initial balance = 19000 t_2 will not be executed because balance < 20000.

So, t_1 will only update the balance.

So, option (b) is correct, t_2 should be executed first then t_1.

10. (d)

Disk block size = 5000 records

Block size = 10 records or 15 (keys + Pointers)

Sparse index at 1st level. So number of disk block at 1st level is number of block in database.

$$\text{Data base} = \frac{5000}{10} = 500 \text{ blocks}$$

$$1^{st} \text{ level} = \left\lceil \frac{500}{15} \right\rceil = \lceil 33.33 \rceil = 34 \text{ blocks}$$

$$2^{nd} \text{ level} = \left\lceil \frac{34}{15} \right\rceil = \lceil 2.26 \rceil = 3 \text{ blocks}$$

3rd level = 1 block
Total = [500 + 34 + 3 +1] blocks
= 538 blocks

11. **(d)**

Finding key or R.

Since B is a essential attribute:

$AB^+ \rightarrow ABCD$

$BD^+ \rightarrow ABCD$

$BC^+ \rightarrow BDAC$

So the set of candidate keys are {AB, BD, BC}.

R is decomposed as:

$R_1(CD)$	$R_2(AC)$	$R_3(BC)$
$C \rightarrow D$	$C \rightarrow A$	$\underline{BC} \rightarrow BC$ (Trivial)

Since in all the three relations LHS is key therefore it is in BCNF.

12. **(b)**

$${}^nC_{n-2} \Rightarrow \frac{n(n-1)(n-2)!}{(n-2)! \times 2} = \frac{n(n-1)}{2} \text{ Candidate keys.}$$

If every $(n - 2)$ attributes are candidate keys

Then every $(n - 1)$ and n attributes are also superkeys

$$\text{Total Super keys} = {}^nC_{n-2} + {}^nC_{n-1} + {}^nC_n = {}^nC_{n-2} + n + 1$$

$$= \frac{n(n-1)}{2} + n + 1$$

13. **(a)**

Format of B+ tree leaf node:

P(Keys + Record pointer) + Block pointer ≤ block size

$$P(10 + 5) + 7 \leq 1024$$

$$P(15) + 7 \leq 1024$$

$$P(15) \leq 1024 - 7$$

$$P \leq \left\lfloor \frac{1017}{15} \right\rfloor$$

$$P = 67$$

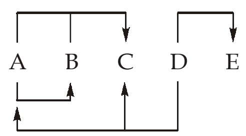
14. **(a)**

Candidate key = {CandidateNo, InterviewDate}
{RoomNo, InterviewDate, InterviewTime}

In FD_3 {StaffNo, InterviewDate} is non prime attribute but RoomNo is prime therefore Recruitment relation is in 3NF.

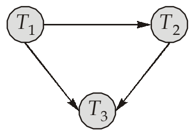
15. **(a)**

R(A, B, C, D, E)



$\{D\}^+ \rightarrow$ D, A, C, E, B
Hence D is the key.
R is decomposed as, R_1(A, C, D), R_2(D, E), R_3(B, C)
R_1(A, C, D): A $\rightarrow$ C, D $\rightarrow$ AC : Key $\rightarrow$ D
R_2(D, E): D $\rightarrow$ E : Key $\rightarrow$ D
R_3(A, B, C) : AB $\rightarrow$ C, A $\rightarrow$ B : Key $\rightarrow$ A
Combining R_1 and R_3 both have 'A' and 'C' an common AC is the superkey for R_3, so lossless.
Hence, it is dependency preserving and lossless.

16. (a)



$\because$ Schedule is conflict serializable.

T_1	T_2	T_3
$R(a)$		
$R(b)$		
$W(a)$		
	$W(b)$	
		$R(b)$
	$R(c)$	
$W(a)$		
	$W(a)$	
		$W(b)$
C_1		
	C_2	
		C_3

Schedule is recoverable.

17. (a)

If in 2PL, all exclusive locks are acquired by transactions only in increasing order then, there is a guarantee that atleast 1 process will complete its execution and free the locked data file for the other processes to continue their execution.

18. (a)

19. (c)

1. Bank {Branch_code, Contact_No}

   where {Branch_code, Contact_No} is primary key.

2. Bank {Branch_code, Bank_Name, Addr}

3. Branches {Branch_No, Branch_code} where {Branch_No, Branch_code} is primary key.

4. Bank_Branch {Addr, Branch_No, Name}.

5. Loan taken {<u>Loan No</u>, Amount, Type, Branch_No}.

   where Loan_No is primary key.

- Here we can't merge branches relation and Bank_Branch entity because foreign key "Code" is not the candidate key of bank entity, so we cannot combined these two.

20. (b)

(Upper query results those marks which are less than atmost 3 marks, i.e. 9.8, 9.7, 9.6) union (This will return empty since no marks greater than query other marks including itself i.e. 9.7 cannot greater than 9.7)

$$= (9.8, 9.7, 9.6) \cup (\phi)$$
$$= 9.8, 9.7, 9.6$$

21. (a)

Gate:

x	y
0	1
1	2
2	3
3	2
4	5

After the query (2, 3) is deleted but due to on delete cascade (1, 2), (0, 1), (3, 2) will be deleted additionally. So, 3 tuples extra deleted.

22. (d)

$$(AB)^+ = A\ B\ C\ D\ E\ F$$
$$(AC)^+ = C\ D\ E\ A\ F\ B$$
$$(AD)^+ = D\ E\ A\ F\ B\ C$$
$$(AF)^+ = A\ F\ B\ C\ D\ E$$
$$(AE)^+ = A\ E\ F\ B\ C\ D$$

There are 5 candidate keys are present for given relation.

23. (c)

Typical structure of a *B*-tree non-leaf node is follows :

P_1	B_1	K_1	P_2	B_2	K_2	–	P_{m-1}	B_{m-1}	K_{m-1}	P_m

in order for each *B*-tree node to fit in a single disk block the maximum value of p is.

$(p \times P_B)+((p-1) \times (P_D + K)) \leq B$

where disk block size $B = 256$ byte

block pointer $p = 7$ B

record (data pointer) $P = 10$ B

search field $K = 12$ B

$$(p \times 7) + ((p-1) \times (10+12)) \leq 256$$
$$7p + 22p - 22 \leq 256$$
$$29\,p \leq 278$$
$$p \leq \left\lfloor \frac{278}{29} \right\rfloor = 9$$

24. **(b)**

In S_2 : $w_3(y)$ is first and $r_2(y)$ appears second

Hence, c_2 should appear after c_3.

In recoverable [If T_j is reads a value written by T_j, the Ti must commit after T_j commits]

Schedules S_1 and S_3 are recoverable.

25. **(a)**

In order to recover the schedule, all the 10 transactions are required as the successive transaction is dependent on the data updated by previous transaction like T_2 is dependent on a variable which is updated by T_1. As T_1 fails, T_2 is also required to be recovered. Now if T_2 is recovered, T_3 is also recovered as T_3 depends on data updated by T_2. Likewise, it happens for all the transactions till T_{10}.

26. **(b)**

As R is referring to R_2 and S is primary key of R_2, $\pi_R(r_1) - \pi_S(r_2)$ will give empty relation or empty table as number of values in R column of table r_1 will always refer to of respective values in S column of r_2.

27. **(a)**

While converting this ER diagram into relational model, we get three relations

R_1: Parts (Part-id, P-id, Trainer-id, P-Name)

R_2: Product (P-id, P-Name)

R_3: Trainer (Trainer-id, Trainer-name)

28. **(c)**

(a) Recoverable :

$r_1(x), w_2(y), r_1(y), C_2, C_1$; [4]

(b) Casecadeless :

$w_1(x), w_2(x), C_2, w_1(x), C_1$; [2]

(c) Strict-recoverable :

$r_1(x), w_2(y), C_2, r_1(y), w_1(y), C_1$; [3]

(d) Non recoverable :

$r_1(x), w_2(y), r_1(y), C_1, C_2$; [1]

29. **(d)**

$\pi_B\left(R \underset{R \cdot B \neq S \cdot C}{\bowtie} S\right)$ which selects all foreign key values.

$\pi_C\left(R \underset{R \cdot B \neq S \cdot C}{\bowtie} S\right) - \pi_C(S) \equiv$ Always empty

All foreign keys values – All primary key values ≡ Always empty

30. **(c)**

$$= [2^4 - 1] \times 2^4$$
$$= 15 \times 16 = 240$$

■■■■