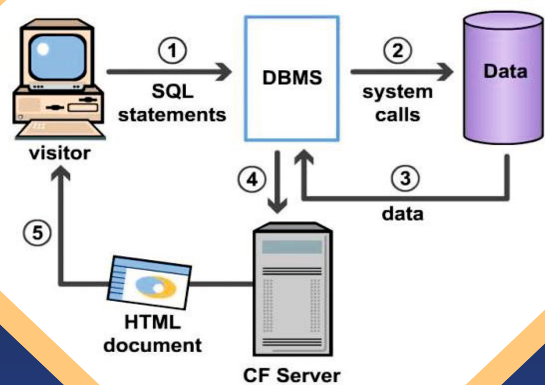
ACE Engineering Publications

SINCE 2011

GATE | PSUs



Computer Science & Information Technology

DATABASE MANAGEMENT SYSTEMS

Text Book: Theory with worked out Examples and Practice Questions

Database Management Systems

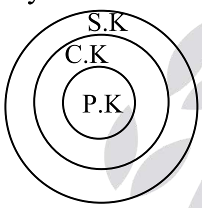
(Solutions for Text Book Practice Questions)

Chapter 2

ER and Relational Model

01. Ans: (b)

Sol: A superkey is one which contains a candidate key.



A key which contains a candidate key (primary key) VY then it will be considered as super key.

In options (a), (c) and (d) it contains candidate key (primary key) VY.

But option (b) doesn't contain candidate key (primary key) VY. So it is not super key.

02. Ans: (b)

Sol: All the values present in Foreign key must present in primary key of the referenced relation.

03. Ans: (c)

Sol: S1: A relation schema can have more than one foreign key references to more than one table.

S2: A foreign key in a relation schema R can be used to refer to tuples of R.

∴ Both statements S1 and S2 are false.

04. Ans: (c)

Sol: When parent is update, it requires child table to be updated simultaneously

05. Ans: (c)

Sol: 1. On removal of row (2,4), row (5,2) and (7,2) must also be deleted as they depend on value

2. On removal of row (5,2), row (9,5) must also be deleted as it depends on value 5.

06. Ans: (d)

Sol: Every value present in foreign key must present in primary key of the referenced relation but need not vice versa. Insertion of tuple (1,2) into R_2violates foreign key constraint and insertion (5,2) into R_3violates check constraint

07. Ans: (b)

Sol: Derived attribute is an attribute that derives its value from one or more attributes.

08. Ans: (b)

Sol: Hobby here represents multi valued attribute.

09. Ans: (c)

10. Ans: (c)

Sol: A State might have several Districts and each district belongs to one state.

11. Ans: (b)

Sol: Each phone is assigned to at least one, and may be assigned up to 10 Employees and Each employee is assigned at least one, but no more than 6 phones.

12. Ans: (a)

Sol: Cardinality ratio of A:B is one-to-many; A participates partially; B participates completely

13. Ans: (a)

Sol: Every entity in A is associated with exactly one entity in B

14. Ans: (a)

15. Ans: (a), (c) & (d)

Sol: Since is not an attribute of dept table, So it is wrong regarding to E-R diagram.

16. Ans: (b)

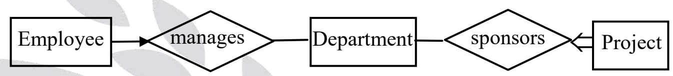
Sol: Strong entities E_1 and E_2 are represented as separate tables, in addition to that many to many relationship (R_2) must be converted as separate table by having primary key of E_1 and E_2 as foreign key. One to many relationship must be transferred to 'many' side table by having primary key of one side as foreign key. Hence we will have minimum of 3 tables.

17. Ans: (b)

Sol: Strong entities E1 and E2 are converted as separate tables. Since A23 is a multi valued attribute it should also be converted as separate table. Relationship R is transferred to 'm' side (E2).

18. Ans: 3

Sol: E-R model is

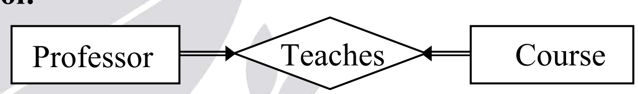


The minimum number of relations in relational model is 3.

1. (Employee, manages)
2. Department
3. (Project, sponsors)

19. Ans: (a)

Sol:



20. Ans: (b)

Sol: There are some tuples of course may not participate with any tuple of professor, then cid is a key for the relation.

21. Ans: (a)

Sol: (AR_1B) will be one table as there is total participation and key constraint.

(CR_2) will be the second table as there is a key constraint.

22. Ans: (d)

23. Ans: (b)

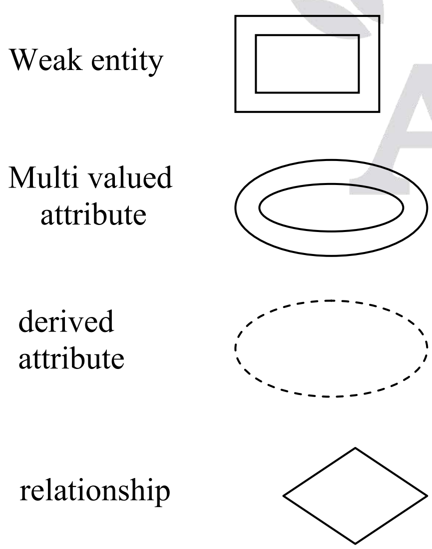
Sol: M, P are strong entities hence they must be represented by separate tables. M table is modified to include primary key of P side (i.e P1). N is weak entity, and it is modified to include primary key of P (i.e P1).

24. Ans: (a)

Sol: M and P are strong entities hence they must be represented as separate tables. To include R1, M table is modified to accommodate primary key of P side (i.e P1) as foreign key. N is weak entity, so modify N to accommodate primary key of P (i.e P1) as foreign key.
Therefore tables are (M1, M2, M3, P1), (P1, P2), (N1, N2, P1). So correct answer is (M1, M2, M3, P1).

25. Ans: (c)

Sol:



Chapter 3 Functional Dependencies

01. Ans: (a), (b) & (c)

Sol: Tuples 1 and 5 violates B → A dependency

02. Ans: (b)

Sol: p cannot uniquely determine q, because for the same value of p multiple values of r are obtained. So p → q is false. (a) and (d) are false hence. r → p is also false because unique value of r determines multiple values of p. So (c) is false. (b) is true because for unique values of q, unique value of p is determined and unique value of p determines unique value of r.

03. Ans: (a), (b), (c) & (d)

Sol: For any value of ‘a’ the dependency A→B is true

04. Ans: (a), (b)

Sol: $C^+ = \{C\}$
$E^+ = \{ABCDE\}$

05. Ans: (c)

Sol: AF^+ = AFDE not ACDEFG as given.

06. Ans: (b)

Sol: (A) closure of {A, C} = {A, C, E}; hence, {A, C} does not determine F.
(B) is true as {A, B, C} is a candidate key.
(C) closure of {A, B, D} = {A, B, D, E, F}; hence, {A, B, D} does not determine C.

07. Ans: (b)

Sol: CD^+ from functional dependencies (FDs) = **C**DE**A**B, it includes RHS attributes AC, so it can be derived from FDs BD^+ from functional dependencies (FDs) = BD only, RHS attributes CD are not included in the closure. Hence it cannot be derived BC^+ from functional dependencies (FDs) = BCDEA, it includes RHS attributes CD, so it can be derived from FDs AC^+ from functional dependencies (FDs) = ACBDE, it includes RHS attributes BC so it can be derived from FDs

08. Ans: (a), (b) & (c)

Sol: (a) P+ = PQR ∴ P → R possible

(b) PS+ = PSQRT ∴ PS → T possible

(c) PS+ = PSQRT ∴ PS→ Q possible

(d) R+ = R ∴ R → T is not possible

09. Ans: (c)

Sol: If every dependency of F is determined using G and every dependency of G is determined using F then F and G are said to be equivalent.

For option 'C'; both F and G are equal

10. Ans: (c)

Sol: D→C in set-2 and C→D in set-1 not covered by each other.

11. Ans: (a)

Sol: A → B, B → C, A → D is the irreducible set of the given FD's.

12. Ans: 5

Sol: AC → D can be eliminated, it can be derived from A → B and CB → D using augmentation and transitive rule.

A → B ⇒ AC → BC

⇒ AC → D

And remaining FD's are not possible to eliminate

∴ 5 FD's are there in minimal cover.

13. Ans: (d)

Sol: BC→A is inessential as it can be determined from the remaining set of dependencies.

14. Ans: (a)

Sol: As V → W, delete W from VW → X results in V → X

As V → X, delete X from Y → VX results in Y → V

The irreducible set is

V → W

V → X

Y → V

Y → Z

15. Ans: (a)

Sol: Only EC+ contains all attributes of the relation, then EC is key for R.

16. Ans: (b)

Sol: As 'K' is independent attribute, key is ABDK.

17. Ans: (d)

Sol: ABD^+ = A, B, C, D, E.

18. Ans: (b)

Sol: $ACEH^+$ contains all the attributes of R.

19. Ans: (d)

Sol: Closure of $AEH^+ = BEH^+ = DEH^+$ = A, B, C, D, E, H. If any closure includes all attributes of a table then it can become candidate key of the table. Closure of AEH, BEH, DEH includes all attributes of table. Hence they are candidate keys.

20. Ans: (d)

Sol: CD+ do not contains all the attributes or R

21. Ans: (a), (b) & (c)

Sol: From the given set of functional dependencies, we can see that (Reg No)+ = {Reg No, Name, Address, Phone, Class ID}. Thus, Reg No is a candidate key. According to FD2 and FD4, {Address, Phone} and Class ID, are also the candidate keys.

22. Ans: 24

Sol: $2^4 + 2^4 - 2^3 = 24$

23. Ans: (b)

Sol: A^+ = ABCEFGH

B^+ = ABCEFGH

E^+ = ABCEFGH

F^+ = ABCEFGH

All of the above attribute closures contain all attributes of R, except D. Hence the candidate keys are AD, BD, ED and FD. i.e, the number of candidate keys are 4.

24. Ans: 3

Sol: The candidate keys of a relation is: AB, AD and C.

25. Ans: (c)

Sol: A candidate key always determines any attribute of a relation. A superset of a candidate key is called super key and it can determine all the attributes of a relation.

Chapter 4 Normalization

01. Ans: (a)

Sol: Join between the tables of II is returning:

p	q	r
a	1	x
a	1	y
b	1	x
b	1	y

which is not the original table, hence the decomposition is lossy.

02. Ans: (c)

Sol: **Decomposition D1**

R1(PQST) R2 (PTX)R3(QY) R4(YZW)

R1∩R2 = PT+ = PTXYZW; determining attributes of R2

[(R1joinR2)∩R3] = Q + = QYZW; determining attributes of R3

[(R1joinR2joinR3)∩R4] = Y+ = YZW; determining attributes of R4

∴ The decomposition D1 is lossless.

Decomposition D2

R1(PQS) R2(TX) R3(QY) R4(YZW)

(R1joinR3joinR4) ∩R2 = ø

∴ The decomposition D2 is lossy.

03. Ans: (b)

Sol: R1= A, B

R2= B, C

R3= B, D

R2 ∩ R3 = B and it is key in R2 (B→C).

(R2 ∪ R3)∩R1 = (B, C, D) ∩ (A, B) = B.

B is a key in (B, C, D) as B→C, C→D. Hence it is lossless join but C→D is not preserved.

04. Ans: (a)

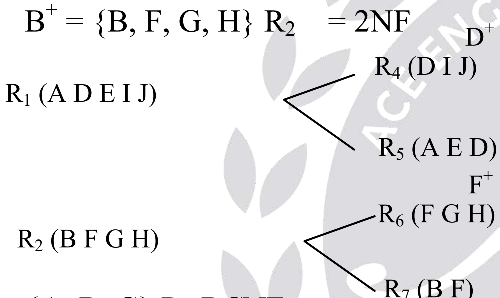
Sol: the decomposition is lossless but not preserving the dependencies B→D and E→G

05.

Sol: R is in 1NF ∴ decompose to 2NF

$A^+ = \{A, D, E, I, J\}$ R_1 = 2NF

$B^+ = \{B, F, G, H\}$ R_2 = 2NF



{A, B, C} R_3 BCNF

Then decompose into 2NF

R_1 (ADEIJJ)

R_2 (BFGH)

R_3 (ABC)

3NF also in BCNF

R_3 (ABC)

R_4 (DIJ)

R_5 (AED)

R_6 (FGH)

R_7 (BF)

AB^+ is key.

06.

Sol: Candidate key: AC

A^+ = (ABE) R_1, C^+ = (CD) R_2

(ACF) R_3

07. Ans: (c)

Sol: R is in 1NF as A→FC and B→E are partial dependencies

08.

Sol: (1) C → D

C → A

B → C

C.K: B, 2NF but not 3NF

(2) 2NF but not 3NF as no partial dependency CK: BD.

(3) R is in 3NF but not in BCNF

(4) C.K = A

(5) Candidate Keys = AB, CD, BC, AD

R is in 3NF but not in BCNF.

09. Ans: (a), (c)

10. Ans: (b)

Sol: S1 is true because it contains one attribute then no partial dependencies possible; hence a relation is always in 2NF.

S2 is false because partial dependencies are not allowed in 2NF and other higher normal forms beyond 3NF also satisfy the conditions required for 2NF.

11. Ans: (c)

Sol: for option (a): AB is key and B→C is a partial functional dependency and makes relation is in 1NF but not in 2NF.

for option (b): AB and AC are keys and satisfying the definition of 3NF, that is either LHS is super key or RHS is a prime attribute.

for option (c): A is key and B→C is transitive dependency hence the relation is in 2NF but not in 3NF.

for option (d): B and C are keys and satisfying the definition of BCNF, that is every LHS is a super key.

12. Ans: (c)

Sol: {A, B} = {A, B, C, D}. Hence {AB} is the primary key. A and B are prime attributes. In the FD AB→CD the determinant is a super key. In the FD D→B, the attribute B is a prime attribute. Hence R2 is in 3NF but not in BCNF.

13. Ans: (a)

Sol: Candidate keys of the relation are A, BC and E. As all determinants are keys, the relation is in BCNF.

14. Ans: (b)

Sol: To simply the process assume A=name, B=courseNo, C=RollNo, D=grade. Candidate keys are AB, CB. If we select AB as the primary key, then C→ A is allowed in 3 NF (either left side is a key or right hand side is part of the key). A→C is allowed since it is prime attribute (an attribute that is part of any candidate key)

15. Ans: (c), (d)

16. Ans: (b)

Sol: From the customer requirement, we can infer FD: Seniority → Salary and is considered as Transitive dependency. Hence the relation is in 2NF but not in 3NF and BCNF.

17. Ans: (b)

Sol: (Volume, Number) → Year is a partial functional dependency. So, the given relation is in 1 NF but not in 2 NF.

18. Ans: (a), (b) & (c)

Sol: Every dependency of R satisfies 3NF, that is if X→A is a functional dependency then either X is super key or A is a prime attribute. A relation in 3NF also satisfies 2NF and 1NF.

19. Ans: (d)

Sol: If relation consists only two attributes always it satisfies BCNF (no partial and transitive dependencies). Hence option (a) is correct.

If every key consists only one attribute, then there is no partial dependency hence option (b) is correct.

If there is dependency X → Y, it is allowed in 3 NF if either X is a key or Y is part of some key (prime attribute). Hence option (c) is correct.

If there is dependency X → Y, it is allowed in BCNF if either X is a key or all determinants are keys. Hence option (d) is incorrect.

20. Ans: (c)

Sol: A table is said to be in BCNF if it is already in 3 NF and all determinants are keys.

21. Ans: (a)

Sol: A relation with two attributes is always in BCNF

If all attributes of a relation are prime attributes then the relation is in 3NF

A relation need not to have non-prime attributes.

BCNF Decomposition may not preserves functional dependencies.

Chapter 5 Relational Algebra & Calculus

01. Ans: 2

Sol: Relational calculus eliminate the duplicates.

$\{T / \exists B \in \text{Book } (T.\text{Title} = B.\text{Title})\}$

02. Ans: (d)

Sol: Retrieve rollNo of students who have enrolled for exactly one course

03. Ans: (a)

04. Ans: (c)

Sol: Retrieve rollNo and name of students who have enrolled for no courses

05. Refer to PYQs **06. Refer to PYQs**

07. Ans: (b)

Sol: This relational algebra queries will be solved by using the select and project operator from the above mentioned relational schemas. At first it will select all the tuples from the relation whose name is Aditya using select operator then it will select phone number and address of 'Aditya' through the project operator

08. Ans: (a), (c)

Sol: The common tuples between the two tables comes as output. Hence intersection between the tables has been performed. So option (a) is correct. Again $R1 \cap R2 = R1 - (R1 - R2)$. Hence option (c) is also correct.

09. Ans: (b) & (d)

Sol: The given operation is equivalent to the intersection of Zone1 and Zone2. Hence, options (b) and (d) are correct.

10. Ans: (a)

Sol: $\Pi_B(r_1) - \Pi_C(r_2) = \phi$ is always true. Because 'B' is foreign key referencing 'C', so 'C' must be a primary key, 'B' cannot have a value that is not available in 'C'. Hence operation $\Pi_B(r_1) - \Pi_C(r_2)$ is always ϕ.

11. Ans: 16

Sol: The output of the above query contains 16 tuples

12. Ans: 8

Sol: The output of the query is

A	B	C	D	E	F	G	H
2	5	2	2	3	6	1	4
2	5	2	2	4	1	2	3
2	5	2	2	2	3	1	4
2	5	2	2	1	2	4	2
3	6	3	1	3	6	1	4
3	6	3	1	4	1	2	3
3	6	3	1	2	3	1	4
3	6	3	1	1	2	4	2

13. Ans: (a)

Sol: Common column between tables 'R' and 'S' is attribute B. In table 'R' B is primary key (B→A, A→C). In table 'S' B is foreign key so join is performed on attribute B. Therefore maximum tuples possible in the output is equal to rows in Table S (as it has less number of rows, provided B values are not repeated in table 'S').

14. Ans: (a)

Sol: R in r1 (P,Q,R) is foreign key with 2000 tuples references R (primary key) in r2 (R,S,T) with 2500 tuples. So natural matching rows are 2000

15. Ans: (d)

Sol:

empAge

empno	age
1	20
2	19
3	18

empAge

empno1	age1
1	20
2	19
3	18

The query return

empno
1
2

∴ The query returns Employee number of all employees whose age is not the minimum.

16. Ans: (c)

17. Ans: (d)

Sol: Minus operator indicates rows available in LHS table but not in RHS table. In this expression, LHS table produces all female students, RHS table consists students with less marks hence it produces names of all girl students with more marks than all the boy students.

18. Refer to PYQs **19. Refer to PYQs**

20. Ans: 2

Sol: The output of the query is a relation with schema Z and tuples { (3), (5) }

21. Ans: (a), (c)

Sol: As per the rules of division operation.

22. Ans: 4

Sol: The output of T_1 is:

courseName
CA
CB
CC

the output of T_2 is:

StudentName
SA
SC
SD
SF

23. Ans: 1

24. Ans: (b)

Sol: The result will have Q attribute values from S1 which are associated with all the pairs of P, R values present in S2 and are present in S1 as triples, hence outcome will have a relation with scheme (Q) and tuples {(2), (4)}. Note that 6 does not have the above property as only (1, 6, 2) is present in S1 but (2, 6, 4) is not present in S1.

25. Ans: (a) & (b)

Sol: $\Pi_{eId}(\Pi_{eId,bId}(Own)/\Pi_{bId}(Brand))$

$\Pi_{eId}(Own) - \Pi_{eId}((\Pi_{eId}(Own)\times \Pi_{bId}(Brand)) - \Pi_{eId,bId}(Own))$.

Chapter 6 Structured Query Language (SQL)

01. Ans: 2

02. Ans: (a), (b)

Sol: Option (c) is incorrect, since option (c) finds the students whose registration number is either 1 or 10 which also includes the tuple (1,ALEX, CIVIL).

03. Ans: (b)

Sol: The result of the query is

A	B	C
1	4	4
2	5	1
2	5	3
3	5	2

04. Ans: (d)

Sol: The query finding the tuples in r_1 but not in r_2

05. Ans: (c)

Sol: Union operator eliminates the duplicates.

06. Ans: (a)

Sol: All the three queries return the same results.

07. Ans: (a)

Sol: Table after the update query is

Rollno	Marks
456	70
457	53
458	63
459	null

Output of the average query is (70+53+63)/3 = 62

08. Ans: (c)

Sol: sum (rating)/count(0) is smaller value than avg(rating).

09. Ans: (b)

Sol: Select clause contains either aggregate function or the attributes that appear in group by clause.

10. Ans: 2

Sol: It returns two rows.

Student – Name	Sum(P.Marks)
Raj	4
Rohit	2

11. Ans: (A)

Sol: If a→b holds on a table, then a become a key and each group contains one record, hence the result is empty.

12. Ans: 3

Sol: The query finds the details of the staff of age below 40 and with at least one more staff of opposite gender having same supervisor.

The query find the details of the staff with name Alice, Mathew and Jane

13. Ans: (d)

Sol: R(A,B,C) will be

A	B	C
4	8	null
3	6	3
2	4	null
null	3	6
null	2	4

14. Ans: (c)

Sol:

R_1

A	B
1	5
3	7

R_2

A	C
1	7
4	9

$R_1 \bowtie R_2$

A	B	C
1	5	7
3	7	null
4	null	9

15. Ans: 1

16. Ans: 7

Sol:

Loan		
loan-number	**branch-name**	**amount**
L11	Banjara Hills	90000
L14	Kondapur	50000
L15	SR Nagar	40000
L22	SR Nagar	25000
L23	Balanagar	80000
L25	Kondapur	70000
L19	SR Nagar	65000

17. Ans: (b)

Sol: For the professor in the outer query, inner query finds min start date of all the professors of same dept.

18. Ans: (c)

Sol: The above query finds "Retrieve enames of all employees whose salary is maximum".

19. Ans: (b)

Sol: The condition is B>any (1,2,1,3,2,4) and the output will be 4.

20. Ans: (a)

Sol: The inner query returns all values of capacity and P1.capacity is true only for the maximum capacity.

21. Ans: (b)

Sol: The ALL keyword specifies that the search condition is TRUE if the comparison is TRUE for every value that the sub query returns. If the sub query returns no value, the condition is TRUE.

22. Ans: 2

Sol: The query finds name of those passengers whose age is above 65 and has some reservation for 'AC' class.

The output of the query is: <u>pname</u>

Rohan

Anil

23. Ans: (B)

Sol: **Not exists** operator returns true if the inner query returns empty set, and the query returns details of employee who do not work on any project.

24. Ans: 3

Sol: Inner query returns the output 0, and outer query returns true for each row.

25. Ans: 2

Sol:

Total	
name	**capacity**
Ajmeer	20
Bikaner	40
Churu	30
Dungargarh	10

Total – avg
capacity
25

The result of the query is: **<u>name</u>**

Bikaner

Churu

Chapter 7
Transactions & Concurrency Control

01. Ans: (d)

Sol: A: Atomicity C: Consistency
I: Isolation D: Durability

02. Ans: (b)

Sol: The data base system must be consistent before and after the transaction.

03. Ans: (c)

Sol: Atomicity property ensured by Transaction manager
Durability property ensured by Recovery manager
Isolation property ensured by concurrency control manager

04. Ans: (b)

Sol: The number of serial schedules are 2
The number of concurrent schedules are $= \frac{(5+3)!}{5!*3!} = 56$
Then, the total number of non serial schedules are = (number of concurrent schedules – number of serial schedules)
= 56 – 2 = 54

05. Ans: (d)

Sol: Transaction T_3 perform read on A, which is updated by T_1 and committed before T_1 does.

06. Ans: (c)

07. Ans: (c)

Sol: A recoverable schedule is one where for each pair of transactions T_i and T_j such that T_j reads a data item previously written by T_i, the commit operation of T_i appear before the read operation of T_j.

08. Ans: (a), (b)

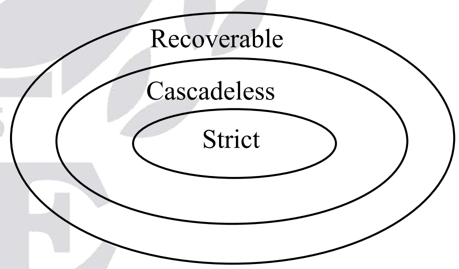
Sol: T2 performs dirty read on T3, so T2 should commit after T3 is committed (or) T3 commits before T2

09. Ans: (c)

Sol: A schedule is said to be strict if a value written by a transaction T is to be read or written by another transaction until either T commits or aborts.

10. Ans: (a), (c)

Sol:

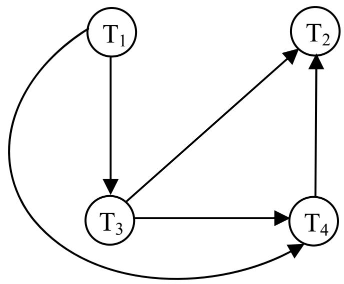


11. Ans: (a)

Sol: If a schedule is serializable, the topological order of a graph (precedence graph) yields a serial schedule.

12. Ans: (a)

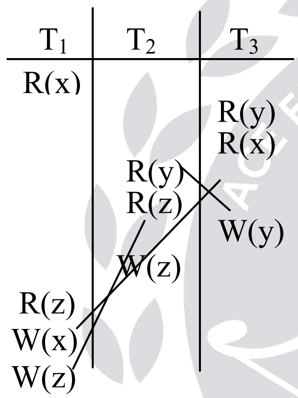
Sol: The precedence graph for the given schedule is



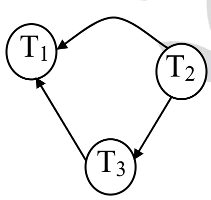
The topological sort of the graph is
$T_1–T_3–T_4–T_2$

13. Ans: (a)

Sol: S1:

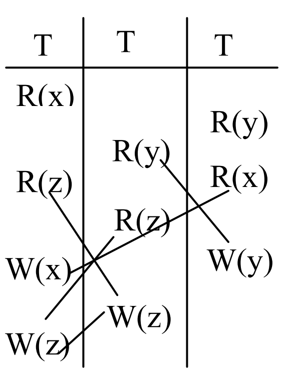


Precedence graph

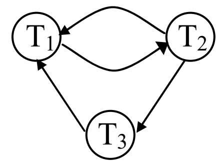


S1 is conflict serializable to $T_2 \rightarrow T_3 \rightarrow T_1$

S2:



Precedence graph



S2 is not conflict serializable

14. Ans: (d)

15. Ans: (d)

Sol:

T_1	T_2	T_3
R(A)		
	W(A)	
		R(A)
W(A)		
		W(A)

S1 and S2 are conflict equivalent to serial schedule T_2, T_3, T_1.

S3 is not conflict equivalent as 2RA, 3WA ($T_2<T_3$) and 3WA, 2WA ($T_3<T_2$) are the conflict operations. There is no serial schedule that satisfies both $T_2<T_3$ and $T_3<T_2$.

16. Ans: (a)

Sol: The two schedules are said to be conflict equivalent if all the conflicts in both the schedules are same.

All the conflicts in the given schedule are same in the schedule of option (a).

17. Ans: (a)

Sol: Draw the precedence graph of the transactions and observe that the graph has a cycle. So, the above schedule is not a conflict serializable schedule.

Next, we have to check whether the schedule is view serializable and the conflict operations should follow the time stamp order T2-T3-T1 and the order is violating. Therefore the schedule is neither view nor conflict serializable schedule.

18. Ans: (a)

Sol: In 2PL locking and unlocking being done in 2 phases

19. Ans: (c)

Sol: In strict 2 PL, all excusive locks (write locks) must be released after commit statements. Hence Option (c) is correct.

20. Ans: (b) & (c)

Sol: Wound-Wait Deadlock Prevention Scheme:

When TA1 requests data item held by TA2(older means smaller timestamp), two cases may arise.

- If TA1 older than TA2 then TA1 wounds TA2(TA2 will be aborted).
- If TA1 younger than TA2 then TA1 will wait to release the data item held by TA2.

In this case Transaction T_1 is older and T_3 is younger in respect of Transaction T_2. Hence options (b) and (c) are correct.

21. Ans: (c)

Sol: In wait-die deadlock prevention strategy and older transaction will wait for younger transaction to release the lock where as an younger transaction aborts if requesting a lock held by an older one.

In wound-wait deadlock prevention strategy a younger transactions need to wait for an older transaction to release the lock but an older transition requesting a lock held by an younger one preempts younger transaction to abort.

22. Ans: (d)

Sol: Figure1 is a wait-for graph without a cycle, so it doesn't shows possibility of deadlock, while Figure2 wait-for graph has cycle, so it shows possibility of deadlock.

23. Ans: (b)

Sol: 2 PL is based on locks and hence not free from deadlock but ensures conflict serializability. Timestamp ordering protocol is based on timestamps ensures conflict serializability and also free from deadlock.

24. Ans: (b)

Sol: The order of TimeStamp is $T_1 \rightarrow T_3 \rightarrow T_2$.

The timestamp ordering requires that all the conflicts to be processed in the order of their timestamps.

$W_2(B) - W_3(B)$ is violating time stamp order and the schedule is not possible under timestamp protocol, But allowed under Thomas Write Rule which ignores $W_3(B)$ called Obsolete Write.

25. Ans: (d)

Sol: When T_2 performs $R_2(A)$ and $TS(T_2)<W\text{-}TS(A)$ then $R_2(A)$ is rejected and rolled back. Therefore the above schedule is not allowed in both Basic timestamp protocol and Thomas write rule.

Chapter 8

Recovery Management System

01. Ans: (b)

Sol: Before T_i executes write (X) operation, a log record < T_i, X, V_1,V_2 > is written where V_1 is the value of X before the write (the old value) and V_2 is the value to be written to X (the new value).

02. Ans: (b)

Sol: In log based recovery we must perform Redo operation for those transactions that contains both <start> and <commit> log record.

We perform Undo operation for those transactions that contains only <start> but not <commit> log record.

Therefore we perform Redo of T_1 and Undo of T_2.

03. Ans: (b)

Sol: Here, T0 and T1 both are committed but T2 is not committed at the time of crash.

So, undo (T2) will be done first and the value of A is set to 300.

Transactions T0 and T1 need to be redone because log contains both the start and commit for both of them.

04. Ans: (c)

Sol: In the immediate database modification scheme, during recovery after a crash, a transaction needs to be redone if and only if both (T_i, start), (T_i, commit) are So, option (c) is correct.

05. Ans: (d)

Sol: In the deferred database modification scheme all modifications are recorded to the log, but all the writes are done after partial commit.

During recovery after a crash, a transaction needs to be redone if and only if both (T_i start) and (T_i commit) are present in the log.

In this scheme there is no need of any undo operation because all write operations are deferred until partial commit of that transaction.

If the system crash after step 6 then only redo of T1 is required but no action is required for T0 because partial commit of T0 is not done.

06. Ans: (a)

Sol: In the deferred database modification scheme, all modifications are recorded in the log, but all the writes are done after partial commit.

During recovery after a crash, a transaction needs to be redone if and only if both (T_i, start) and (T_i commit) are present in the log.

In this scheme, there is no need of any undo operation because all write operations are deferred until partial commit of that transaction.

So, option (a) is correct.

07. Ans: (d)

Sol: As per the process of transaction recovery.

08. Ans: (c)

Sol: As per the process of transaction recovery

09. Ans: (c)

Sol: In order to recover the database even on multiple crashes the same undo and redo list will be used.

10. Ans: (a)

Chapter 9 Indexing

01. Ans: (a)

Sol: Cluster index is created on a column with cluster of values (like year, age, branch where group of students fall under one group), so it is non-key but requires ordering.

02. Ans: (c)

Sol: **Primary index:** in a sequentially ordered file, the index whose search key specifies the sequential order of the file.

Secondary index: an index whose search key specifies an order different from the sequential order of the file.

Dense index : Index record appears for every search-key value in the file.

Sparse Index: contains index records for only some search-key values.

The indexing shown holds index records for some search key values and not all, hence it is sparse indexing. Moreover indexing on search key mentions sequential order of file. So it is primary indexing. Hence option (c) is correct.

03. Ans: (c)

Sol: A clustering index as the name suggests is created when the data can be grouped in the form of clusters of non-key and order. So, option (c) is correct.

04. Ans: (a), (b) & (d)

Sol: The first level index is an ordered file sorted on the candidate key.

The number of entries in the last level index is dependent of the block size.

The number of entries in the last level index is need not be always 1, may be more than one.

05. Ans: 698

Sol: Size of index record = 12 + 7 = 19

No. of index records = 1, 50, 000

Blocking factor $= \frac{4096}{15}$ =215 record/block

No. of index blocks = $\frac{1,50,000}{215}$

= 698.

06. Ans: (c)

Sol: Since block size is 2^{10} (=1024 bytes), $2^m \times 2^n = 2^{10}$. Only option (c) satisfies with m = 8 and n = 2.

07. Ans: (c)

Sol: $n*5+(n–1)*(10+8) \leq 512$

$5n+16n–18 \leq 512$

$23n \leq 530$

$n \leq 23.$

08. Ans: (c)

Sol: The order of the B-Tree is

$n* 6 + (n–1) * (9+7) \leq 512$

$6n + 16n – 16 \leq 512$

$22n \leq 528$

$n \leq 24$

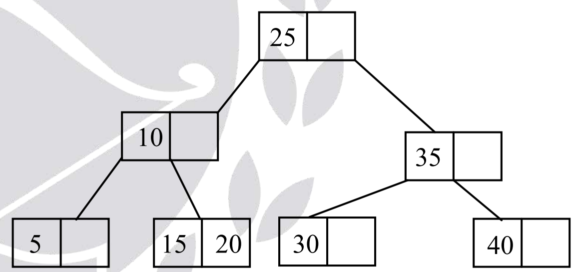
- Root has 1 node, 24 pointers and 23 index records
- Level 1 has 24 nodes, 576 pointer and 552 records
- Level 2 has 576 nodes, 13,824 pointer and 13248 records

09. Ans: (a)

Sol: Insertion of new key is leading to insertion of new node at all 4 levels, in turn it is leading to insertion of new root node. Hence the maximum number of nodes that could be created are 5

10. Ans: (c), (d)

Sol: The tree after key 40 inserted



11. Ans: (c)

Sol: 10 is replaced with largest key from the left subtree.

12. Ans: 5

Sol: The nodes to access all records with a "search key greater than or equal to 7 and less than 15" is (9), (5), (5, 7) (9, 11) and (13, 15).

13. Ans: 52

Sol: Key = 8, Block size = 512,

Block pointer = 2 bytes, the order of B^+ tree is maximum number of block pointers in it. (Let ‘n’)

$n * 2 + (n - 1)\, 8 \leq 512$

$2n + 8n - 8 \leq 512$

$10n \leq 520$

$n \leq 52$

14. Ans: (a)

Sol: $n * (k+P_r) + P \leq B$,

(Where n is order of the tree, P is block pointer, k is key value and B is block size).

$n*(9 + 7) + 6 \leq 1024$

$16n \leq 1018$

$N \leq 63$

15. Ans: 50

Sol: Order of non-leaf node is

$(n\times8) + (n-1)\, 12 \leq 1024$

$8n + 12n - 12 \leq 1024$

$20n \leq 1036$

$n \leq 51$

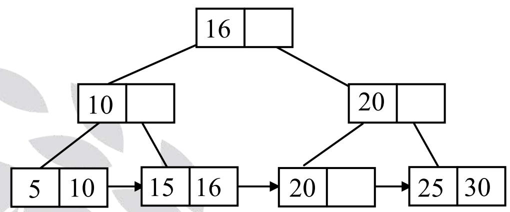
maximum number of keys possible is :50

16. Ans: (b)

Sol: We begin the traversal from the root and identify the leaf node (20, 22) as the possible node to insert. But this node is full because the order of leaf is 2. So we split this node to insert 15. The resulting nodes are (15, 20) and (22). So the number of leaf nodes is incremented by 1. The element 20 is moved to the internal node (22). The pair (20, 22) is the modified internal node. So the number of internal nodes doesn’t change.

17. Ans: 3

Sol: Resultant tree

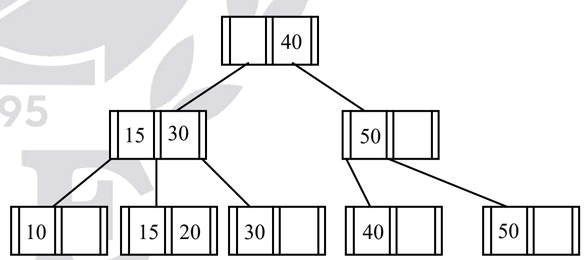


18. Ans: (d)

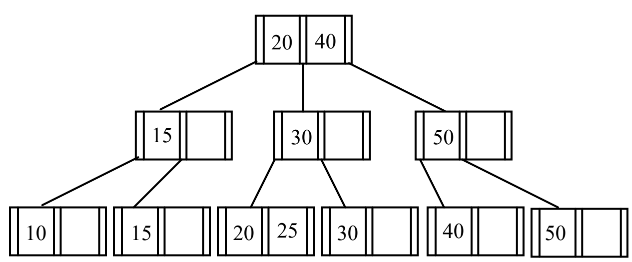
Sol: Deleting ‘10’ from internal node requires 10 to be replaced with copy of 13.

19. Ans: (a)

Sol: <u>Insert 15</u>



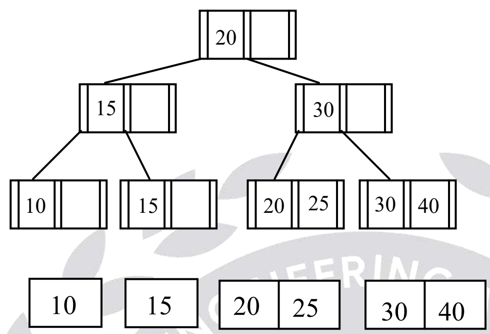
<u>Insert 25</u>



20. Ans: (a)

Sol: If we remove K50 in index node, there is no change in height as still root, index and leaf nodes exist.

Leaf nodes are now, after Delete 50, the B^+ is:



Hence Root now consists: 20

(i) is true, (ii) is true but (iii) is not true