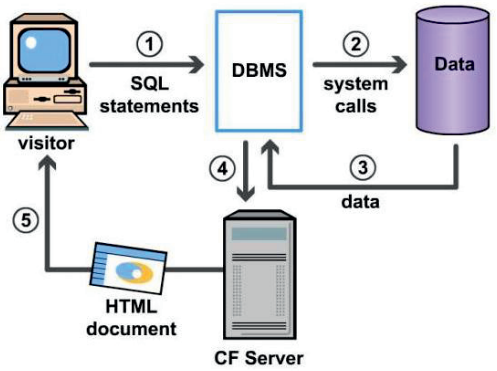
ACE x DEEP LEARN

GATE | PSUs

COMPUTER SCIENCE & INFORMATION TECHNOLOGY

Data Base Management Systems

(**Text Book :** Theory with worked out Examples and Practice Questions)



Hyderabad | Delhi | Pune | Bhubaneswar | Lucknow | Bengaluru | Chennai
Vijayawada | Vizag | Tirupati | Ahmedabad | Kolkata

2 Database Management Systems

(Solutions for Text Book Practice Questions)

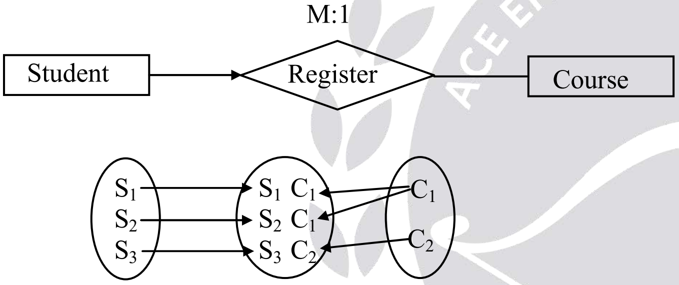
2. ER and Relational Model

01. Ans: (b)

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

02. Ans: (a)

Sol: The E-R model for the description is



03. Ans: (a)

Sol: As every specialized entity is subset of generalized entity, then the deletion of generalized entity requires the deletion of specialized entity.

04. Ans: (c)

Sol:

- Composite attribute is an attribute which is composed of other attributes.
- Multi valued attribute represented with double ellipse.
- Derived attributes represented with dotted ellipse

05. Ans : (c)

Sol:

- Participation of both department and employee is partial, so both options (a) and (d) are incorrect.
- The number on the cardinality specifies the maximum but not the minimum therefore option (b) is incorrect and option (c) is correct.

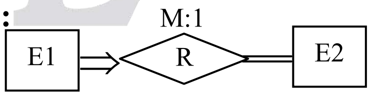
06. Ans: (b)

Sol: According to the definition, E-R Model is one which consists of set of entities and the relation among these entities. So Option (b) is correct and rests all are incorrect.

Option (a) represents Relational data model, Option (c) represents Network data model and Option (d) represents Hierarchical data model.

07. Ans: (a)

Sol:

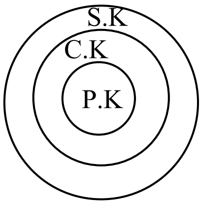


08. Ans: (a) & (b)

Sol: {12, Jessica} is a relation instance of the schema student (id#, emp name)

09. 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 option (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.

10. Ans: (b)

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

11. 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.

12. Ans: (c)

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

13. 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

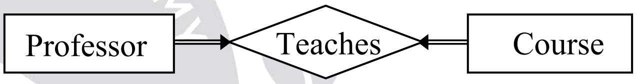
2. On removal of row (5,2), row (9,5) must also be deleted as it depends on value 5.

14. Ans: 0

Sol: When <3, 8> is deleted, its related tuples in T_2 is (8, 3) and 3 is set to null. Hence the number of additional tuples to delete is 0

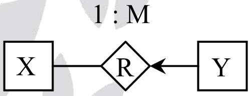
15. Ans: (a)

Sol:



16. Ans: (b)

Sol:



The ER model can be represented with two relations X and YR, therefore Y side include primary key of X side as Foreign key.

17. 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.

18. 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.

19. Ans: (b)

Sol: As we get key and participation constraint from course to registration, therefore the number of tuples in registration will be equal to the tuples in the course table.

20. Ans: (b)

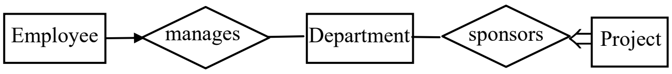
Sol: Strong entities E_1 and E_2 are represented as separate tables, in addition to that many to many relationship (R_2) must be converted as separate table by having primary key of E_1 and E_2 as foreign key. One to many relationship must be transferred to 'many' side table by having primary key of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

21. 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).

22. 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)

23. Ans: (c)

Sol: From both Employee and Department we have total participation and key constraint possible, Which can be converted into one relation in relational model.

24. 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).

25. 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).

26. Ans: (c)

3. Functional Dependencies

01. Ans: (a), (c), (d)

Sol: A particular employee id determines more than one department, hence employee id → department is invalid on the above table.

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: (d)

Sol: AC^+ =A, C, B, E, F, G

05. Ans: (c)

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

06. Ans: (c)

Sol: A functional dependency $X \to Y$ is said to be trivial iff $Y \subseteq X$.

07. Ans: (b)

Sol: CD^+ from functional dependencies (FDs) = **C**DE**AB**, 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) $P_+ = PQR \therefore P \to R$ possible

(b) $PS_+ = PSQRT \therefore PS \to T$ possible

(c) $PS_+ = PSQRT \therefore PS \to Q$ possible

(d) $R_+ = R \therefore R \to T$ is not possible

09. Ans: (c)

Sol: Every dependency of F can be determined using G and every dependency of F can be determined using F.

10. Ans: (c)

Sol: D→C in set2 and C→D in set1 not covered by each other.

11. Ans: (b)

Sol: E →A dependency of Y cannot be determined using X.

12. Ans: (d)

Sol: AB→C, A→BC both can be determined from remaining set of FD's.

13. Ans: 5

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

$A \to B \Rightarrow AC \to BC$

$\Rightarrow AC \to D$

And remaining FD's are not possible to eliminate

$\therefore$ 5 FD's are there in minimal cover.

14. Ans: (b)

Sol: As with C we determine B using the dependencies C → E and C → B, then attribute B can be dropped from X.

15. Ans: (d)

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

16. 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

17. Ans: 24

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

18. Ans: (b)

Sol: (no) + determines the entire attribute set of the relation. Hence n, o are prime, whereas m, p, q, r, s, t are non prime because they are not part of any candidate keys.

19. Ans: (a)

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

20. Ans: (b)

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

21. Ans: (d)

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

22. Ans: (b)

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

23. 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.

24. 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.

25. Ans: 3

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

26. Ans: 6

Sol: AB, AD, EB, ED, CB, CD.

27. 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.

28. 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

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: (a)

03. Ans: (b)

Sol: R1= A, B

R2= B, C

R3= B, D

R2 $\cap$ R3 = B and it is key in R2 (B$\rightarrow$C).

(R2 $\cup$ R3)$\cap$R1 = (B, C, D) $\cap$ (A, B) = B.

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

04. Ans: (c)

Sol: Decomposition D1

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

R1$\cap$R2 = PT+ = PTXYZW; determining attributes of R2

[(R1joinR2)$\cap$R3] = Q + = QYZW; determining attributes of R3

[(R1joinR2joinR3)$\cap$R4] = Y+ = YZW; determining attributes of R4

$\therefore$ The decomposition D1 is lossless.

Decomposition D2

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

(R1joinR3joinR4) $\cap$R2 = ø

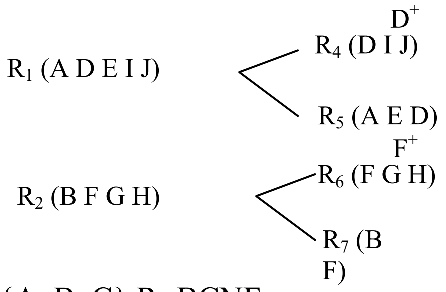
$\therefore$ The decomposition D2 is lossy.

05.

Sol: R is in 1NF $\therefore$ 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) $C \rightarrow D$
$C \rightarrow A$
$B \rightarrow C$
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: (d)

Sol: Relation R_1 satisfies A→B, B→C and C→AB dependencies and all the determinants are super keys. Hence the relation is in BCNF.

10. Ans: (a)

Sol: F = {QR → S, R→P, S→Q}
The decomposed relations Y(PR) and Z(QRS) satisfying the dependencies {R→P} and {QR→S, S→Q} respectively.
Relation Y is in BCNF but relation Z is not in BCNF **because** in S→Q ; S is not a super key. All the dependencies of relation X is satisfying on relations Y and Z.

11. Ans: (b)

Sol: As given client id and order id together is a key and it is possible to determine Firstname, Lastname of a client using his client id, then we have the dependency clientid → Firstname, Lastname which is a partial functional dependency. Hence the relation is in 1 NF and it is violating 2NF.

12. Ans: (a)

Sol: Primary key for the table is F1 F2, so F1→F3 and F2→F4 become partial dependencies therefore it is not even in 2 NF hence it is in 1 NF.

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: (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 superkey.

15. Ans: (b)

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

16. 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)

17. Ans: (b)

Sol: rollno, courseid is superkey in rollno, courseid→email, rollno is prime attribute in email→rollno.

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: (c)

Sol: 2NF deals with partial dependencies.

If X→Y allowed in 3NF then either X is superkey <u>**or**</u> Y is part of the key.

5. Relational Algebra & Calculus

01. 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

02. 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.

03. Ans: (b), (d)

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

04. 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.

05. 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').

06. 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

07. Ans: (a)

Sol: As relation 'r' need to satisfy both conditions F_1 and F_2, we replace the expression $\sigma_{F_1}\left(\sigma_{F_2}(r)\right)$ with $\sigma_{F_1 \wedge F_2}(r)$

Using A_1 only sufficient in the selection because $A_1 \subset A_2.$

08. Ans: (c)

Sol: 'A' is a column in relation R, then instead of joining all the tuples of R, with S we join only few tuples of R that satisfying the condition <u>A = a with S</u> which is the optimized query.

09. Ans: (b)

Sol: bal < 0 filter rows from account ∞ depositor from which we can operate on few rows to filter b city = “Agra”.

10. 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

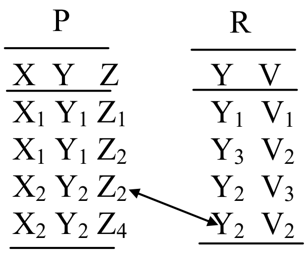
∴ The query returns Employee number of all employees whose age is not the minimum.

11. 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.

12. Ans: 1

Sol:



P

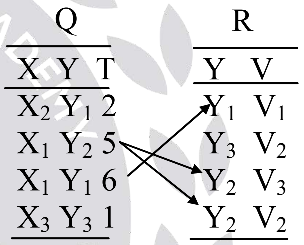
X	Y	Z
X_1	Y_1	Z_1
X_1	Y_1	Z_2
X_2	Y_2	Z_2
X_2	Y_2	Z_4

R

Y	V
Y_1	V_1
Y_3	V_2
Y_2	V_3
Y_2	V_2

Result of the expression

$$\prod_X \left(\sigma_{P.Y=R.Y\wedge R.V=V_2}{}^{(P\times R)}\right) \text{ is } \frac{X}{X_2}$$



Q

X	Y	T
X_2	Y_1	2
X_1	Y_2	5
X_1	Y_1	6
X_3	Y_3	1

R

Y	V
Y_1	V_1
Y_3	V_2
Y_2	V_3
Y_2	V_2

Result of the expression

$$\prod_X \left(\sigma_{Q.Y=R.Y\wedge Q.T>2}{}^{(Q\times R)}\right) \text{ is } \frac{X}{X_1}$$

The **result** of $(x_2) - (x_1) = x_2$

13. Ans: (a) & (c)

Sol: As per the rules of division operation.

14. Ans: (a)

Sol: Division operator is used to compare a value with all the values of other relation.

First expression returns sid’s of sailors who reserved all boats called Ganga, and its outer query returns those sailor names.

15. Ans: 4

Sol: The output of T_1 is: <u>courseName</u>

CA
CB
CC

the output of T_2 is: <u>StudentName</u>

SA
SC
SD
SF

16. Ans: (c)

Sol: The 'Λ' operator in tuple calculus will have same effect as the '$\cap$' intersection operator in relational algebra.

17. Ans: (c)

Sol: Use the generic expression of tuple relational calculus

18. Ans: (c)

Sol: Here, an existentially bound tuple variable e1 is used to make sure that if roll number and name of a student are returned by the query then the student definitely had enrolled for some course.

19. Ans: (c)

Sol: In negative queries TRC produces infinite results hence it is not considered as safe.

20. Ans: 2

Sol: Relational calculus eliminate the duplicates.

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

21. Ans: (a)

Sol: SQL, Relational algebra, tuple relational calculus and Domain relational calculus all is representing the same. i.e., all these expressions representing to find the distinct names of all students who score more than 90% in the course numbered 107.

6. Structured Query Language (SQL)

01. Ans: (d)

Sol: To search for a specified pattern in a column we use LIKE operator in the WHERE clause. The % sign represent zero or multiple characters.

Option (a) displays the name and loan number of customers whose names are exactly 'A'.

Option (c) displays the name and loan number of customers whose names ends with 'A'.

Option (b) gives error message, invalid relational operator for the word LIKES.

Since option (d) satisfies the requirements, it is correct.

02. Ans: (a), (b)

Sol: Option (c) is incorrect, since and 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)

$$\text{avg (marks)} = \frac{(10+0+30+0)}{4} = 10$$

06. Ans: (c)

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

07. Ans: (c)

Sol: Union operator eliminates the duplicates.

08. Ans: (a)

Sol: All the three queries return the same results.

09. Ans: (b)

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

10. Ans: (a)

Sol: If b→c hold result is empty as each group contains only one row.

11. Ans: (b)

Sol: The having clause selects only those names, which are duplicated. Therefore the guests is finding duplicate names of employees.

12. Ans: 2

Sol: It returns two rows.

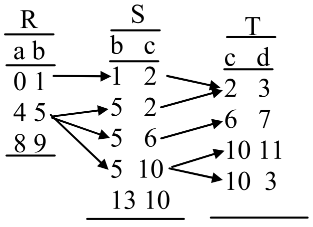
Student – Name	**Sum(P.Marks)**
Raj	4
Rohit	2

13. Ans: 5

Sol: For each student it returns one row as an output.

14. Ans: 5

Sol: Natural join is a join among the two relations with equality condition among all attributes having the same name.



R ⋈ S ⋈ T

a	b	c	d
0	1	2	3
4	5	2	3
4	5	6	7
4	5	10	11
4	5	10	3

Totally '5' tuples in the result.

15. 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

16. 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

17. Ans: (d)

Sol:

- Inner join returns the rows that have matching rows of both the relation.
- Left outer join returns all the rows from left side relation even if there is no matching row in the right side relation.
- Right outer join returns all the rows from right side relation even if there is no matching row in the left side relation.
- Full outer join returns all the rows from both the relation even if there is no matching row in the other relation.
- Query 4 returns a result, which is superset of Query1, Query2 and Query3.

18. Ans: (c)

Sol: Left outer join returns all rows from left side relation even if there is no matching row in the right side relation. In option (c), 'r' may contain some rows where B is greater than 5 and when joined with tuples of 'S' whose B is less than 5 results in some extra rows, when compared with result of Q.

19. Ans: (d)

Sol: Inner query finds the average salary of all the employee. Therefore the query finds, “for each department in the company, the number of female employees whose salary is greater than the average salary of employees in the company”.

20. Ans: (a)

Sol: The SOME clause holds true for greater than one condition

21. Ans: (b)

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

22. Ans: (a)

Sol: ‘=any’ operator is same as ‘in’ operator

23. Ans: (a)

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

24. 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.

25. Ans: 7

Sol: The output of the query is ta.player

Klose
Ronaldo
G muller
Fontaine
Pele
Klismann
Kocsis

26. 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: pname

Rohan
Anil

27. Ans: (c)

28. Ans: 2

Sol:

Total			**Total – avg**
name	**capacity**		**capacity**
Ajmeer	20		25
Bikaner	40		
Churu	30		
Dungargarh	10		

The result of the query is: **name**

Bikaner
Churu

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: (b)

Sol: The number of serial schedules are 2

The number of concurrent schedules are $= \dfrac{(5+3)!}{5!*3!} = 56$

Then, the total number of non serial schedules are = (number of concurrent schedules – number of serial schedules)

$= 56 - 2 = 54$

04. Ans: (d)

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

05. Ans: (a)

Sol: As $R_3(x)$ is dirty operations which read $W_1(x)$ and is committed before T_1. Hence schedule is non-recoverable.

06. 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.

07. Ans: (a), (b)

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

08. Ans: (c)

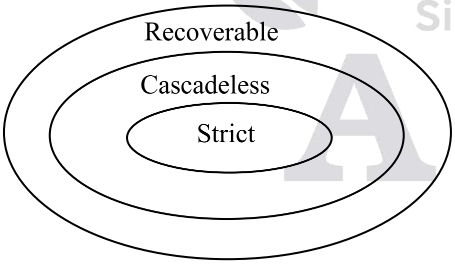
Sol: A schedule is said to be recoverable, let for each pair of transactions T_i and T_j such that T_j reads the value of a data item that was written by T_i and the commit operation T_j appears after the commit operation of T_i or the commit operation of Ti appears before the commit operation of Tj.

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.

Sol: (a) Not Conflict Serializable,
Not View Serializable,
Recoverable, Avoids Cascading aborts,
Not strict.

(b) Not Conflict Serializable,
Not View Serializable,
Not strict,
Recoverable, cascading aborts

(c) Conflict Serializable,
View Serializable,
Serializable,
Not Recoverable,
Not Avoid cascading aborts,
Not Strict

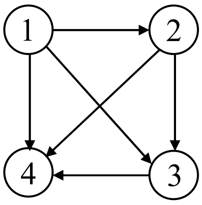
(d) Conflict Serializable,
View Serializable,
Serializable,
Recoverable,
Avoids cascading aborts, strict

12. Ans: (a)

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

13. Ans: (c)

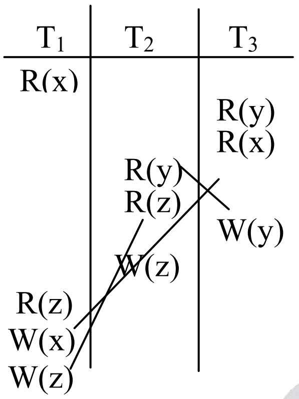
Sol: Precedence graph for the schedule is



The topological sort of the directed graph is $T_1 \rightarrow T_2 \rightarrow T_3 \rightarrow T_4$

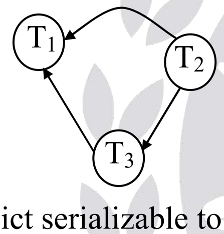
14. Ans: (a)

Sol: S1:



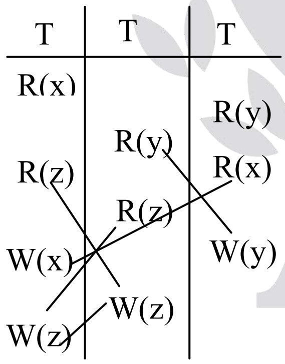
T_1	T_2	T_3
R(x)		
		R(y)
		R(x)
	R(y)	
	R(z)	
		W(y)
	W(z)	
R(z)		
W(x)		
W(z)		

Precedence graph



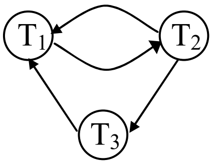
S1 is conflict serializable to $T_2 \rightarrow T_3 \rightarrow T_1$

S2:



T	T	T
R(x)		
		R(y)
	R(y)	
		R(x)
R(z)		
	R(z)	
W(x)		
		W(y)
	W(z)	
W(z)		

Precedence graph



S2 is not conflict serializable

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 T2, T3, T1.

S3 is not conflict equivalent as 2RA, 3WA (T2<T3) and 3WA, 2WA (T3<T2) are the conflict operations. There is no serial schedule that satisfies both T2<T3 and T3<T2.

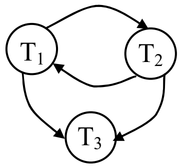
16. Ans: (a)

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: (c)

Sol: Precedence graph is



18. 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 scihedule 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.

19. Ans: (d)

Sol: Let T_i, T_j and T_k are three transactions, if we assume blind write of T_k is the last write operation, then blind writes of other two transactions appear in two ways i.e

T_i–T_j–T_k or

T_j–T_i–T_k

20. Ans : (c)

Sol: To perform W(y) T2 acquires Exclusive lock on y, which will be released only after commit in strict 2pl. Then request for T1 is processed only after commit of T2,.

21. Ans: (c)

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

22. 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.

23. 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 an 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.

24. 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.

25. 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.

26. 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.

27. 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.

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: (a), (b)

Sol: The immediate modification scheme allows updates of an uncommitted transaction to be made to the buffer, or the disk itself, before transaction commits.

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 used.

10. 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 transaction that contains only <start> but not <commit> log record.

Therefore we perform Redo of T_1 and Undo of T_2.

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: (d)

Sol: Number of blocks needed to store data file with 1000 records = 1000/5 = 200

Number of blocks needed to store dense file index = 1000/15= 67

Total blocks required = 200 + 67 = 267 blocks.

So, option (d) is correct.

05. Ans: (d)

Sol: Every block except the last one contains same number of records. No record should span across two blocks.

So the blocking factor of data file

$$= \left\lfloor \frac{\text{block size}}{\text{records Length}} \right\rfloor = \left\lfloor \frac{2048}{200} \right\rfloor = 10$$

The blocking factor of index file

$$= \left\lfloor \frac{\text{blockSize}}{\text{OderingKeyFieldSize} + \text{BlockPointerSize}} \right\rfloor$$

$$= \left\lfloor \frac{2048}{18+7} \right\rfloor = 102$$

06. Ans: 698

07. 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.

08. Ans: 3

Sol: Blocking Factor= 1024/100 =10

No. of data blocks = 30000/10 =3000

Blocking factor for the index file =1024/ 15 = 68

No. of first level blocks = 3000/68 = 45

No. of second level blocks = 45/68 = 1

Total number of blocks required to access a record by searching multilevel index = 2 + 1 = 3

09. Ans: (c)

Sol: $n*5+(n-1)*(10+8) \le 512$

$5n+16n-18 \le 512$

$23n \le 530$

$n \le 23.$

10. Ans: (c)

Sol: The order of the B-Tree is

$n* 6 + (n-1) * (9+7) \le 512$

$6n + 16n - 16 \le 512$

$22n \le 528$

$n \le 24$

- 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

11. 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

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: (b)

Sol:
- B^+ Tree is a **height** balanced search tree
- non leaf nodes have pointers to the next level nodes but not to the data records
- All the leaf nodes are connected with a pointer $\mathbf{P_{next}}$.
- All the key values in each node are kept in sorted order.

14. 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')

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

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

$10n \le 520$

$n \le 52$

15. Ans: (a)

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

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

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

$16n \le 1018$

$N \le 63$

16. Ans: 50

Sol: Order of non-leaf node is

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

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

$20n \le 1036$

$n \le 51$

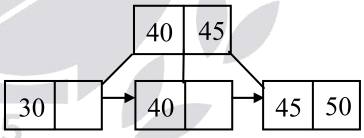
maximum number of keys possible is :50

17. Ans: (b)

Sol: All internal nodes, except the root, will have between n/2 and n children where 'n' is the order of the node. Since the maximum number of keys is 5, maximum number of children a node can have is 6. Number of keys in a node is n–1. Hence minimum number of keys = n/2 –1= 6/2 –1= 2.

18. Ans: (d)

Sol: The resultant tree after the insertion is

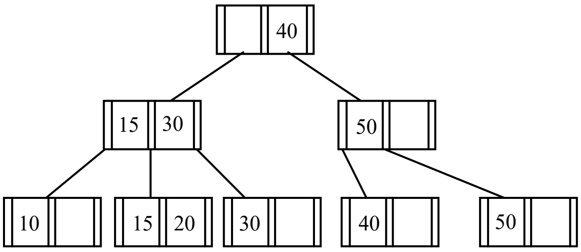


19. Ans: (d)

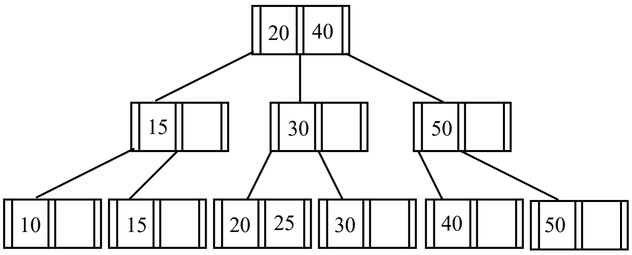
Sol: Deleting '10' from internal node requires 10 to be replaced with copy of 13.

20. Ans: (a)

Sol: Insert 15



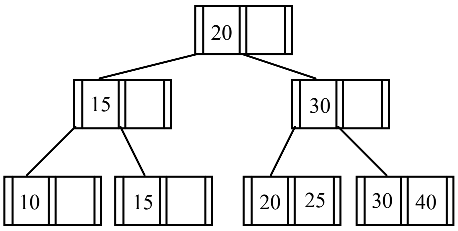
Insert 25

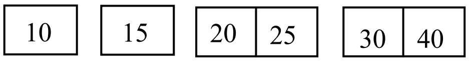


21. Ans: (a)

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