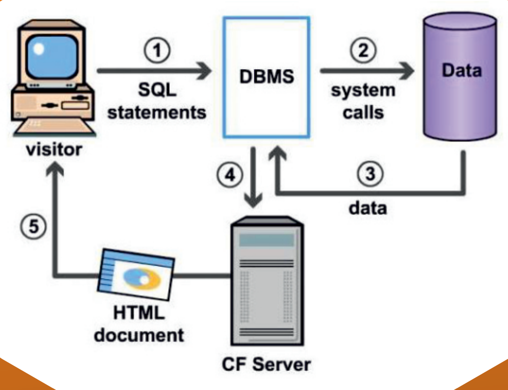
# GATE | PSUs

Since 2011 ACE

# COMPUTER SCIENCE & INFORMATION TECHNOLOGY

## Database Management Systems

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




HYDERABAD | AHMEDABAD | DELHI | BHOPAL | PUNE | BHUBANESWAR | BANGALORE | LUCKNOW
PATNA | CHENNAI | VISAKHAPATNAM | VIJAYAWADA | TIRUPATHI | KOLKATA

# Database Management Systems

*(Solutions for Text Book Practice Questions)*

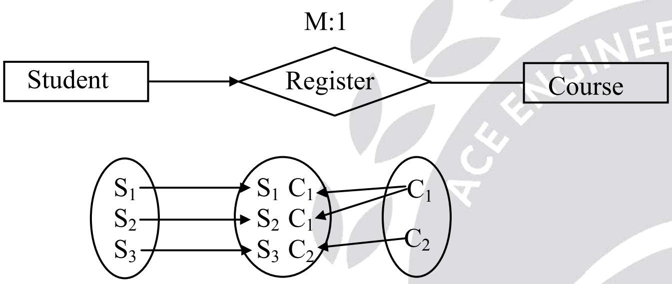
## 2. ER and Relational Model

**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:**

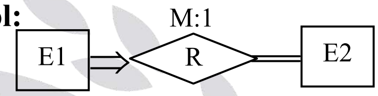
- Composite attribute is an attribute which is composed of other attributes.
- Multi valued attribute represented with double ellipse.
- Derived attributes represented with dotted ellipse

**05. Ans: (b)**

**Sol:** As there is a key constraint and partial participation from purchase to dealership, the E-R diagram represents "atmost one" relationship of car with dealership.

**06. Ans: (a)**

**Sol:**

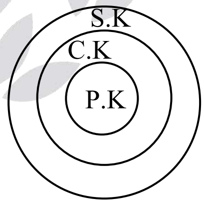



**07. Ans: 19.**

**Sol:** For (**StudentName**, **StudentAge**) to be a key, all the combinations of these two attributes must be unique.

**08. 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.

**09. Ans: (b)**

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

**10. Ans: (c)**

**Sol:** It violates referential integrity constraint as it is updating in foreign key but not in primary key.

**11. Ans: (c)**

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

**12. 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.

**13. Ans: 0**

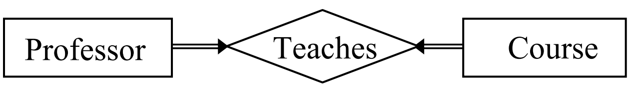
**Sol:** When <3, 8> is deleted, its related tuples in $T_2$ is (8, 3) and 3 is set to null. Hence the number of additional tuples to delete is 0

**14. Ans: (a)**

**Sol:** As the key constraint from professor, the maximum number of tuples possible in Teaches is number of tuples in professor.

**15. Ans: (a)**

**Sol:**




**16. Ans: (d)**

**Sol:** Student has 3 attributes and primary key attribute of department is added as foreign key because of M:1 relationship between student and department.

**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)**

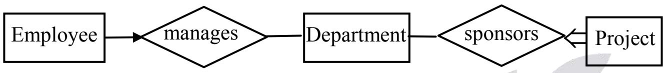
**Sol:** Strong entities $E_1$ and $E_2$ are represented as separate tables, in addition to that many to many relationship ($R_2$) must be converted as separate table by having primary key of $E_1$ and $E_2$ as foreign key. One to many relationship must be transferred to 'many' side table by having primary 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: (b)**

**Sol:**

- An Entity type is represented with one relation.
- Key attribute becomes primary key for the relation
- Composite attribute is represented with set of simple attributes
- Weak entity is represented always as a child table

## 3. Functional Dependencies

**01. Ans: (d)**

**Sol:** As 'BC' is key $BC \to A$ is satisfied

**02. Ans: (b)**

**Sol:** Based on the table values given in query and guidelines below, answer is b. YZ is having unique combination, and Y is also having unique values. Hence $YZ \to X$, $Y \to Z$ are possible.

**03. Ans: (c)**

**Sol:** In option 'C'; if a = 4 first and second tuples are with same AB=(4,2) but it's 'C' is 3 and 5 causing $AB \to C$ dependency violated

**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 → Y is said to be trivial iff Y ⊆ X.

**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: (c)**

**Sol:** $AC^+$ contains I then AC→ I dependency is possible.

**09. Ans: (a)**

**Sol:** D→E of F is not covered by G.

**10. Ans: (c)**

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

**11. Ans: (d)**

**Sol:** AB→C, A→BC both can be determined from remaining set of 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: (b)**

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

**14. Ans: (d)**

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

**15. 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

**16.**

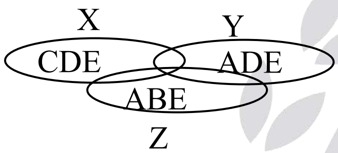
**Sol:**
$$\left.\begin{array}{l} A \rightarrow BC \\ AE \rightarrow H \\ C \rightarrow D \\ D \rightarrow G \\ E \rightarrow F \end{array}\right\} \text{Minimal set}$$

**17. Ans: 24**

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

**18. Ans: 16**

**Sol:**




$= 2^3 + 2^3 + 2^3 - 2^2 - 2^1 - 2^2 + 2^1$

$= 24 - 10 + 2$

= 16 super 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 are

F

AB

CB

**26. Ans: 6**

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

**27. Ans: 2**

**Sol:** D, AH

**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.**

**Sol:** 1. C.K = BD, Lossy, Dependency preserving

2. C.K = AB, CB, Loss-less,
   Not Dependency preserving
3. C.K = A, C, Loss-less,
   Dependency preserving
4. C.K = A, Loss-less,
   Not Dependency preserving
5. C.K = A, Lossy,
   Not Dependency preserving

**02. Ans: (b)**

**Sol:** R1= A, B

R2= B, C

R3= B, D

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

(R2 $\cup$ R3)$\cap$R1 = (B, C, D) $\cap$ (A, B) = B. B is a key in (B, C, D) as $B \to C$, $C \to D$. Hence it is lossless join but

$C \to D$ is not preserved.

**03.**

**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

$R_1$ (A D E I J) → $D^+$ $R_4$ (D I J), $R_5$ (A E D)

$R_2$ (B F G H) → $F^+$ $R_6$ (F G H), $R_7$ (B F)

{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.

**04.**

**Sol:** Candidate key: AC

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

(ACF) $R_3$

**05. Ans: (c)**

**Sol:** R is in 1NF as $A \to FC$ and $B \to E$ are partial dependencies

**06.**

**Sol:** (1) $C \to D$

$C \to A$

$B \to C$

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.

**07. 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.

**08. 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.

**09. Ans : (a)**

**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.

**10. Ans: (c)**

**Sol:** As store id can determine storelocation; here is a dependency

storeid → storelocation and it is a partial functional dependency because key of RECEIPT is (customerid, storeid).

Partial functional dependencies are not allowed in 2NF. To convert into 2NF, move storeid, storelocation attributes to new relation STORE and make storeid attribute as a foreign key in RECEIPT and primary key in STORE relations.

**11. 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.

**12. Ans: (a)**

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

**13. 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.

**14. Ans: (b)**

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

**15. 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 $C\rightarrow A$ is allowed in 3 NF (either left side is a key or right hand side is part of the key). $A\rightarrow C$ is allowed since it is prime attribute (an attribute that is part of any candidate key)

**16. Ans: (b)**

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

**17. 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 $X \rightarrow Y$, 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 $X \rightarrow Y$, it is allowed in BCNF if either X is a key or all determinants are keys Hence option (d) is incorrect.

**18. Ans: (c)**

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

## 5. Relational Algebra & Calculus

**01. Ans: (b)**

**Sol:** Relational Algebra eliminate duplicates always.

**02. Ans: (d)**

**Sol:** As $R_1 \bowtie R_2$ is based on A = C and B = D means it is selecting the common tuples from both $R_1$ and $R_2$, which is called $R_1 \cap R_2$.

**03. 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 $\phi$.

**04. Ans: (a)**

**Sol:** Common column between tables 'R' and 'S' is attribute B. In table 'R' B is primary key ($B\rightarrow A$, $A\rightarrow C$). 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').

**05. 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

**06. 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 $A_1 \subset A_2$.

**07. 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 A = a with S. Which is the optimized query.

**08. Ans: (b)**

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

**09. Ans: (d)**

**Sol:** The above query finds the Courses in which the male students are enrolled.

**10. 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.

**11. 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

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

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

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

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

**12. Ans: (c)**

**Sol:** Output variable should be sid, since it is division operation returns Sid of suppliers who supply all parts.

**13. 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.

**14. 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

**15. Ans: (c)**

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

**16. Ans: (c)**

**Sol:** P. duration = 3 months selects all projects of duration 3 months

T. pname = P.name selects project names in the output.

**17. Ans: (c)**

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

**18. Ans: 4**

**Sol:** Result of given query

| P | Q | U |
|---|---|---|
| 2 | b | 5 |
| 2 | b | 6 |
| 2 | c | 5 |
| 3 | d | 6 |

**19. Ans: 2**

**Sol:** Relational calculus eliminate the duplicates.

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

**20. 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: (b)**

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

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

**02. Ans: (c)**

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

**03. Ans: (c)**

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

**04. Ans: (c)**

**Sol:** Union operator eliminates the duplicates.

**05. Ans: (a)**

**Sol:** All the three queries return the same results.

**06. Ans: (b)**

**Sol:** In general a query with a having clause should also have a group by clause. <u>If you omit group by ( we can do it)</u> , all the rows not excluded by the where clause return as a single group. All the attributes in select clause must appear in Group by else it violates 1 normal form. But all attributes uses on group by clause need not appears in select clause. Hence p and s statements are correct.

**07. Ans: (b)**

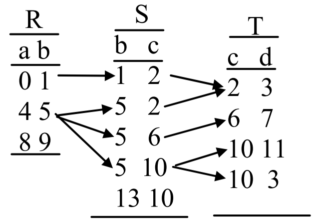
**Sol:** Select clause contains either aggregate function or the attributes that appear in group by clause.

**08. Ans : (c)**

**Sol:** If a is key; each group contains only one record and the having condition is always false the result is empty. If a→b does not hold on R then duplicate entries possible on 'a' and each group may contain more than one rows, therefore the result is non empty.

**09. 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.

**10. Ans: (c)**

**Sol:** Conditions *Student. Roll_number =Grades.Roll_number and Grade.grade=A* look for students with Grade =A, conditions *Courses.Course_number Grades.Course_number and Courses.Instructor =Korth* look for courses taught by Korth. Students who received B grades but taught by Korth will not be retrieved in the query. So it retrieves name of students who have got an A grade in at least one of the courses taught by Korth.

**11. Ans: 2**

**Sol:** It returns two rows.

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

**12. Ans: 5**

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

**13. 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 |

**14. Ans: 8**

**Sol:** Full outer join of R and S will give T relation. Here NULL entries are taken for R and S to include all missing instances of common attribute A while joining R and S.

**15. 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.

**16. 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.

**17. Ans: (b)**

**Sol:** Inner query finds managerid of manager who manages a department with highest budget and the outer query returns that employee name who is a manager

**18. Ans: (b)**

**Sol:** Innermost query retrieves bookno. of a book Database Systems.
next query retrieves rollno. who reserved Database Systems.
Hence outer query retrieves student details who have not reserved Database system book.

**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:** '=any' operator is same as 'in' operator

**21. Ans: (a)**

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

**22. 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.

**23. Ans: 7**

**Sol:** The output of the query is

| ta.player |
|---|
| Klose |
| Ronaldo |
| G muller |
| Fontaine |
| Pele |
| Klismann |
| Kocsis |

**24. 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 |

**25. Ans: (a)**

**Sol:** Inner query returns number of orders for each product, and the outer query returns pid's of products ordered by at least two customers.

**26. Ans: (b)**

**Sol:** Since $Q_1$ consists "not exists", it produces undesired results in certain conditions. Ex: If department = 4 and s.salary >= e.salary, or department = 4 and s.salary <= e.salary, condition is always failed hence inner query produces non empty set. This makes "not exists" condition true hence empId is selected for output. In $Q_2$, Inner query produces salaries for department 5, > Any operator Perfectly produces the desired results.

## 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: (d)**

**Sol:** Irrespective of failures, the changes made by a committed transaction must be permanent.

**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: (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.

**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: (b)**

**Sol:** T2 performs dirty read on T3, so T2 should commit after T3 is committed.

**09. Ans: (b)**

**Sol:** Transaction $T_2$ is reading the data item 'A' that was previously written by $T_1$. If $T_1$ fails after time instance 9 requires to rollback both $T_1$ and $T_2$, but rollback of $T_2$ is not possible as it is already committed. And the schedule is not- Recoverable.

**10. 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.

**11. Ans: (b)**

**Sol:** Every cascadeless schedule is recoverable but need not vice versa.

**12.**

**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) Not Conflict Serializable,
Views serializable through Thomas write rule, Serializable,
Recoverable,
Avoids cascading aborts,
Not strict

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

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

(f) Conflict Serializable,
View serializable,
Serializable,
Recoverable,
No need cascading aborts,
strict

**13. Ans: (a)**

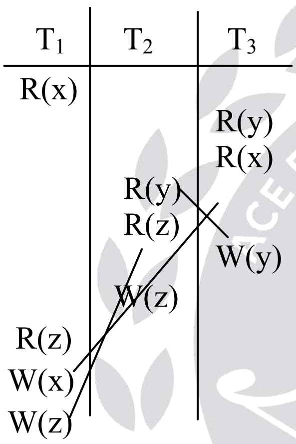
**Sol:** If a schedule is serializable, the topological order of a graph (precedence graph) yields a serial schedule.

**14. Ans: (a)**

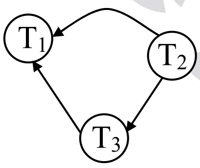
**Sol:** The above schedule is serializable to serial schedule $T_3$, $T_1$, $T_2$ by constructing the precedence graph.

**15. Ans: (a)**

**Sol:** S1:



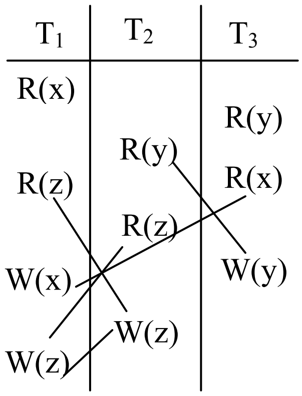

Precedence graph



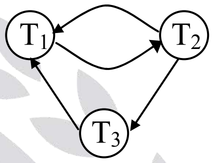

S1 is conflict serializable to $T_2 \rightarrow T_3 \rightarrow T_1$

| $T_1$ | $T_2$ | $T_3$ |
|---|---|---|
| R(A) | | |
| | W(A) | |
| | | R(A) |
| W(A) | | |
| | | W(A) |

S2:



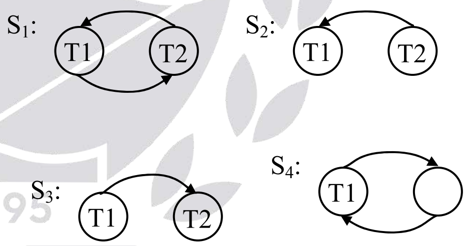

Precedence graph




S2 is not conflict serializable

**16. Ans: (a)**

**Sol:** Precedence graph for each of the schedule given is:




For a schedule, whose precedence graph contains cycles are said to be not conflict serializable.

**17. Ans: (d)**

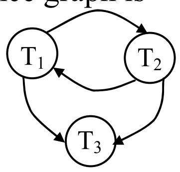
**Sol:** S1 and S2 are conflict equivalent to serial schedule T2, T3, T1.

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.

**18. Ans: (c)**

**Sol:** Precedence graph is




**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: (c)**

**Sol:** An older transaction requesting a data item held by an younger $T_x$ need to wait.

**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: (a)**

**Sol:** In the wait for graph we find a cycle from T31 – T30 – T29 – T31 results in a 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.

## 8. 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: 400**

**Sol:** Blocking factor = 512/20=25

Number of data blocks = 10000/25=400

In primary index, the number of index records= 400 which is number of blocks in multi level index.

**03. 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.

**04. Ans: (c)**

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

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

$23n \leq 530$

$n \leq 23.$

**05. 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

**06. 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).

**07. 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.

**08. 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')

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

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

$10n \leq 520$

$n \leq 52$

**09. Ans: (b)**

**Sol:** $n \times P + (n-1)k <= B$. Where n is order of the tree, P is block pointer, k is key value and B is block size.

Therefore $n \times 6 + (n-1)9 <= 1024$.

n = 1033/16 = 64 (approximately)

**10. Ans: 50**

**Sol:** Order of non-leaf node is

$(n\times 8) + (n-1)\ 12 \leq 1024$

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

$20n \leq 1036$

$n \leq 51$

maximum number of keys possible is :50

**11. Ans: (b)**

**Sol:** order of leaf node is $n(9+7)+6<=512$

$16n<=506$

$n <=31$

order of internal node is $n*6+(n-1)*9<=512$

$15n<=521$

$N<=34$

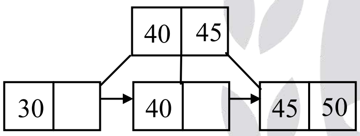
The maximum number of entries in the B+ tree leaf node order 'L' and internal node of order 'i' of 'l' levels is $\mathbf{i^{l} * L}$ that is $(34)^3 * 31$ entries

**12. 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.

**13. Ans: (d)**

**Sol:** The resultant tree after the insertion is

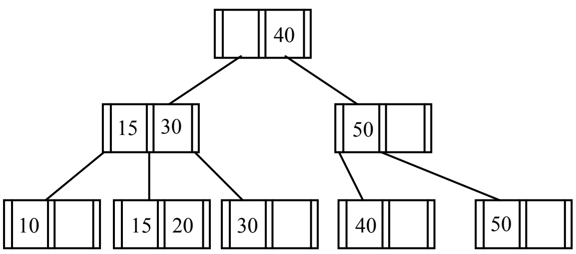



**14. Ans: (d)**

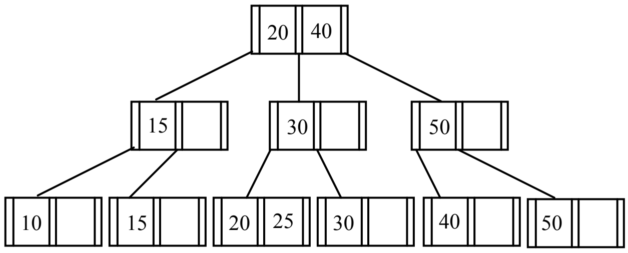
**Sol:** Deleting '10' from internal node requires 10 to be replaced with copy of 13.

**15. Ans: (a)**

**Sol:** Insert 15



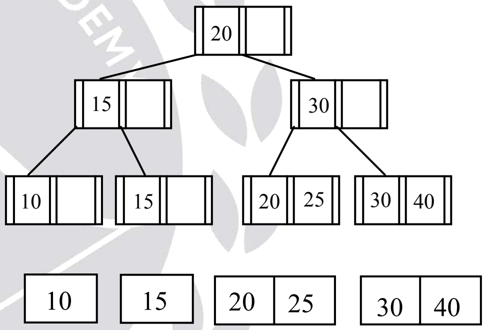

Insert 25




**16. 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