CBCS SCHEME

USN | | | | | | | | | |

21CS53

Fifth Semester B.E. Degree Examination, Dec.2023/Jan.2024
Database Management Systems

Time: 3 hrs. Max. Marks: 100

Note: Answer any FIVE full questions, choosing ONE full question from each module.

Module-1

1 a. Define DBMS. Explain all the basic operations that can be performed by DBMS on a database. **(05 Marks)**

b. Explain the different users of a database system. **(10 Marks)**

c. Describe the 3-Schema Architecture. **(05 Marks)**

OR

2 a. Define the following terms:
i) Data model ii) Schema iii) Insurance iv) Canned transaction **(04 Marks)**

b. Describe the structural constraints of a database system with suitable example. **(10 Marks)**

c. Explain all the E-R diagram notations. **(06 Marks)**

Module-2

3 a. Explain the four relational model constraints. **(06 Marks)**

b. Explain all the steps of Relational database design using E-R to relational schema with a suitable example. **(06 Marks)**

c. Discuss the DIVISION operation of relational algebra. Find the Quotient for the following :

A =

SNO	DNO
S_1	P_1
S_1	P_2
S_1	P_3
S_1	P_4
S_2	P_1
S_2	P_2
S_3	P_2
S_4	P_2
S_4	P_4

B_1 =

PNO
P_2

B_2 =

PNO
P_2
P_4

B_3 =

PNO
P_1
P_2
P_4

Find i) A/B_1 ii) A/B_2 iii) A/B_3 **(08 Marks)**

OR

4 a. Explain the characteristics of a relational model. **(06 Marks)**

b. Explain all types of outer join operations in relational algebra. Demonstrate the advantage of outer join operation over the inner join operation. **(06 Marks)**

c. Considering the following schema
Sailors (<u>sid</u>, sname, rating, age)
Boats (<u>bid</u>, bname, color)
Reserves (<u>sid</u>, <u>bid</u>, day)
Write a relational algebra queries for the following :
i) Find the names of sailors who have reserved boat#103.
ii) Find the names of sailors who have reserved a red boat.
iii) Find the names of sailors who have reserved a red or green boat.
iv) Find the names of sailors who have reserved all boats. **(08 Marks)**

Module-3

5 a. Explain the basic data types available for attributes in SQL. (05 Marks)

b. Demonstrate the following constraints in SQL with suitable example:
i) NOT NULL ii) Primary key iii) Foreign key iv) Default v) Check. (10 Marks)

c. What are triggers? Explain with syntax and suitable example. (05 Marks)

OR

6 a. Explain the basic definition of a cursor and its usage with the help of a suitable example. (05 Marks)

b. What are Assertions? Assuming suitable company schema write an Assertion for the condition.
"The salary of an Employee must not be greater than the salary of the manager of the department that the employee works for". (05 Marks)

c. Referring to the below mentioned company schema. Write the SQL queries for the following:

Employee

Fname	Lname	Minit	Ssn	Bdate	Address	Sex	Salary	SuperSsn	Dno

Department

Dname	Dnumber	Mgr_Ssn	Mgr_start_date

Department_location

Dnumber	Dlocation

Project

Pname	Pnumber	Plocation	Dnum

Work_on

Essn	DNo	HRS

Defendant

Essn	Dependentname	Sex	Bdate

i) For each department retrieve the department number, the number of employees in the department and their average salary.

ii) For each project on which more than 2 employees work, retrieve the project number, the project name and the number of employees who work on the project.

iii) For each project, retrieve the project number, the project name and the number of employees from department no. 5 who work on that project.

iv) For each department that has more than 5 employees, retrieve the department number and the number of its employees who are making more than $40,000 salary.

v) Retrieve the names of an employees who have two or more dependents. (10 Marks)

Module-4

7 a. Explain the types of update anomalies with examples. (05 Marks)

b. Explain Armstrong's rules of inference. (05 Marks)

c. What is the need for normalization? Explain 1NF, 2NF and 3NF with examples. (10 Marks)

OR

8 a. Explain the informal design guidelines of a database. (06 Marks)

b. What is equivalence of sets of functional dependencies? Check whether the following sets of F.D's are equivalent or not.

$FD_1 = \{A \to B, B \to C, AB \to D\}$

$FD_2 = \{A \to B, B \to C, A \to C, A \to D\}$ (08 Marks)

c. Write an algorithm to find the closure of functional dependency 'F'. (06 Marks)

Module-5

9 a. Explain the desirable properties of a transaction. (06 Marks)

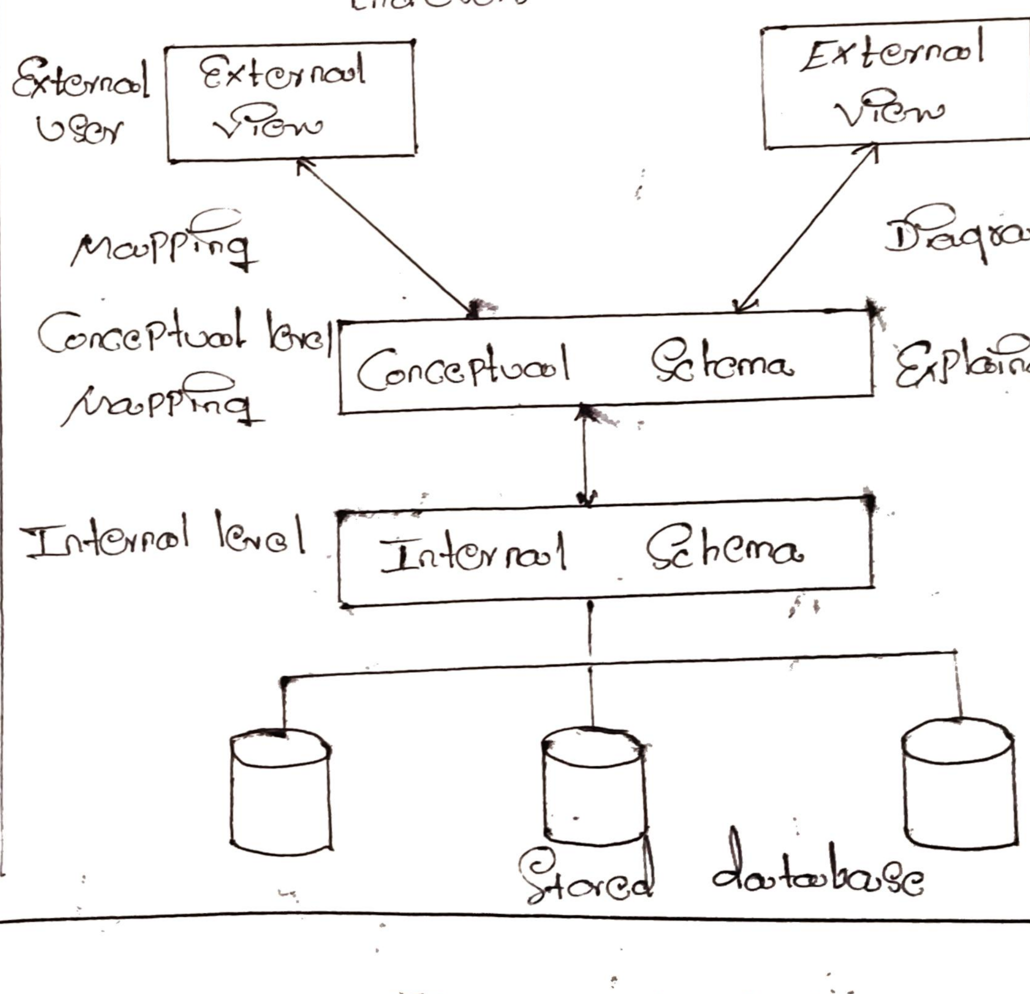
b. Explain with a neat diagram, the state transition diagram of a transaction. (06 Marks)

c. Explain two phase locking mechanism with suitable example. (08 Marks)

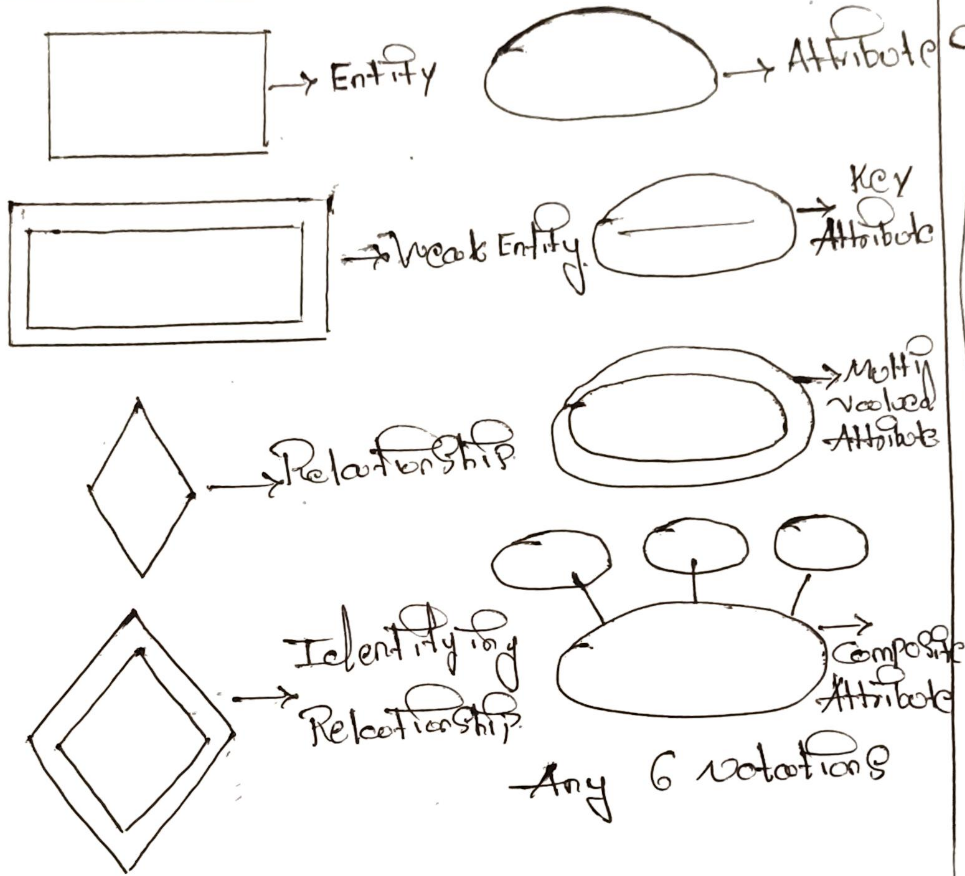
OR

10 a. Discuss on the database inconsistency problem. (10 Marks)

b. Explain Binary locks and shared locks with algorithms. (10 Marks)

Question Number	Solutions	Marks Allocated
1.a	DBMS is a Commercial S/w to Create & maintain the database. → Defn	1M
	Operations : Defining, Constructing, Manipulating, Sharing } → 4×1 = 4 operator	4M
1.b	Different users →	6M
	Actors on the Scene → 1.5 * 4 Database Administrators, Database designers, Endusers, applications Programmers.	
	Workers behind the Scene → 4M →	4M
	DBMS designers & implementers, Tool developers operators & maintenance personnel Endusers	
1.c		
	Diagram →	3M
	Explanation →	2M

Question Number	Solution	Marks Allocated
2.a	Data model → Collection of concepts to describe the database. Schema → description of the database Instance → Running operational copy of the database system. Canned Transaction → Constant querying and updating of the db Each defn 1M * 4	4M
2.b	Structural Constraints Cardinality Ratio → Specifies the maximum no of relationship instances that an entity can participate in → Defn → 1M Examples :- 1:N, 1:1, N:1, M:N → 1 Example Each 1x4=4	5M
	Participation Constraints & Existence Dependencies → Specifies the minimum number of relationship instances that each entity can participate in → Defn - 1M * Total participation } - 2M * Partial participation } - 2M	5M

Question Number	Solution	Allocated
2.c	 Any 6 notations	6M
3.a	i) Domain Constraint ⟶ 1M ii) Key Constraint ⟶ 1M iii) Entity Integrity ⟶ 1M iv) Relational Integrity ⟶ 3M	6M
3.b)	Each step should be explained with a suitable Example. 1) Mapping of Regular Entity Types. 2) Mapping of Weak Entity Types. 3) Mapping of Binary 1:1 Relationship Type 4) Mapping of Binary 1:N Relationship Type	6M

Question Number	Solution	Marks Allocated
	5) Mapping of Binary M:N Relationship type	
	6) Mapping of multivalued Attributes.	
	7) Mapping of N-ary relationship Types.	
3.c)	Division operation Defn → 2M	

Defn :- It is applied between 2 Relations. $R_1(Z) \div R_2(X)$ where R_1 is subset of R_2.

Let $Y = Z - X$. The result of division operation $T(Y)$ that includes a tuple t if tuples t_{R_1} appear in R_1 with $t_R[Y] = t$, and with $t_{R_2}[X] = t_s$ for every tuple t_2 in R_2 — 8M

i) A/B₁ =

S NO
S1
S2
S3
S4

→ 2M

ii) A/B₂ =

S NO
S1
S4

→ 2M

iii)

S NO
S1

→ 2M

Question Number	Solution	Marks Allocated
4a)	Characteristics $1.5 * 4 = 6$ i) ordering of tuples in a relation. ii) ordering of values within a tuple. iii) Values and null's in the tuple. iv) Interpretation (meaning) of a relation.	6M
4b)	**Defn of outer join** It is an operation where the users wants to keep all the tuples in R, or all those in S or all those in both R & S. i) Left outer join → 2M ii) Right outer join → 2M iii) Full outer join → 1M Demonstration of Advantage of outer join with Example → 1M	6M
4.c)	1) $T_1 \leftarrow \sigma_{\langle bid = 103 \rangle}$ (Reserves) $T_2 \leftarrow T_1 \bowtie_{sid = sid}$ Sailors → 2M RESULT $\leftarrow \pi_{\langle sname \rangle} T_2$ 2) $T_1 \leftarrow \sigma_{\langle color = red \rangle}$ Boats $T_2 \leftarrow T_1 \bowtie_{bid = bid}$ Reserves $T_3 \leftarrow T_2 \bowtie_{sid = sid}$ Sailors → 2M Result $\leftarrow \pi_{\langle sname \rangle} T_3$	

Question number	Solution	Marks Allocated
	3) $T1 \leftarrow \sigma_{\langle color = red \rangle \text{ OR } \langle color = green \rangle}(Boats)$ $T2 \leftarrow T1 \bowtie_{bid = bid} Reserves$ $T3 \leftarrow T2 \bowtie_{sid = sid} Sailors$ → 2M $Result \leftarrow \pi_{\langle sname \rangle} T3$	
	4) $T1 \leftarrow \pi_{\langle bid \rangle}(Boats)$ $T2 \leftarrow Sailors \bowtie_{sid = sid} Reserves$ $T3 \leftarrow \pi_{\langle sid, bid \rangle}$ → 2M $Result \leftarrow T3 \div T1$ $Final\ Result \leftarrow \pi_{\langle sname \rangle}(Result \bowtie_{sid = sid} Sailors)$	2M
5.a)	Numeric, Character strings, Bit - string Boolean, Date, Timestamp. Any 5 datatype 1*5 = 5	5M
5 b)	1) NOT NULL 2) Primary key 3) Foreign key 4) Default 5) Check Demonstration of Each Constraint with Syntax & Example 2M * 5 = 10	10M

Question number	Solution	Marks Allocated
5.c	Defn → 1M Example → 4M Create Trigger before_insert_Example Before Insert on table_name For Each Row Begin ---- Trigger Code End	5M
6.a)	Defn: It is a Mechanism that allows you to traverse the result set of a Query & Process each row one at a time. — 1M Example should include Cursor declaration opening a Cursor Fetching rows — 4M Processing & Manipulating rows. closing the Cursor	5M
6.b)	Defn - 2M Create Assertion Salary-Constraint Example:. Check (NOT Exists(Select * from Employee E, Employee M, Department D where. E.Salary > M.Salary AND E.Dept no = D.Number AND. D.MGRSSN = M.SSN)) — 3M	5M

Question number	Solution	Marks Allocated
6.c	(see below)	2*5 = 10M

```
1) Select Dno, Count(*), AVG (Salary)
   From Employee
   Group by Dno;                2x5 = 10 M

2) Select Pnumber, Pname, Count(*)
   from Project, works-on
   where Pnumber = Pno
   Group by Pnumber, Pname
   having Count(*) > 2;

3) Select Pnumber, Pname, Count(*)
   from Project, works_on, Employee
   where Pnumber = Pno and SSn = ESSn
        and Dno = 5
   group by Pnumber, Pname.

4) Select Dno, Count (*)
   from Employee
   where Salary > 40000 and Dno in
      (Select Dno from Employee
         Group by Dno
         having Count(*) > 5)
   Group by Dno.

5) Select Lname, Fname
   from Employee
   where (Select Count (*)
          from Dependent
          where SSn = ESSn) >= 2;
```

Question number	Solution	Marks Allocated
7a)	Insertion Anomary — Defn & Example 2M Deletion Anomary → Defn & Example 1M modification Anomary → Defn & Example 2M	5M
7.b)	if $X \supseteq Y$, then $X \rightarrow Y$ reflexive rule. $\{$if $X \rightarrow Y\} \models XZ \rightarrow YZ$ argumentation rule. $\{X \rightarrow Y, Y \rightarrow Z\} \models X \rightarrow Z$ traversitive rule. $\{X \rightarrow YZ\} \models X \rightarrow Y$ decomposition rule. $\{X \rightarrow Y, X \rightarrow Y\} \models X \rightarrow YZ$ $\{X \rightarrow Y, WY \rightarrow Z\} \models WX \rightarrow Z$	5M
7.c)	Normalization is a Process used in the database design to organise data and minimize redundancy while maintaining data Integrity. It helps designers to achieve 1) Eleminate the redundancy 2) Data Integrity —— 2M 3) Data Consistency —— 1NF States that relations within relations or relations as attributes values within tuples are disallowed. the Query attribute values Permitted are atomic values —— 2M	

Question number	Solution	Marks Allocated

Ex:

DNAME	DNUMBER	MGR-SSN	DLocation
Research	5	12345	{ABC, PQR, XYZ}
Administrd	4	12346.	Stafford
headQuaters	1	12347	Houston

DLocation	Dnumber
ABC	5
Stafford	4
Houston	1
PQR.	5
XYZ	5

DNAME	DNUMBER	MGR-SSN
Research	5	12345
Adminstd	4	12346
headQuote	1	12347

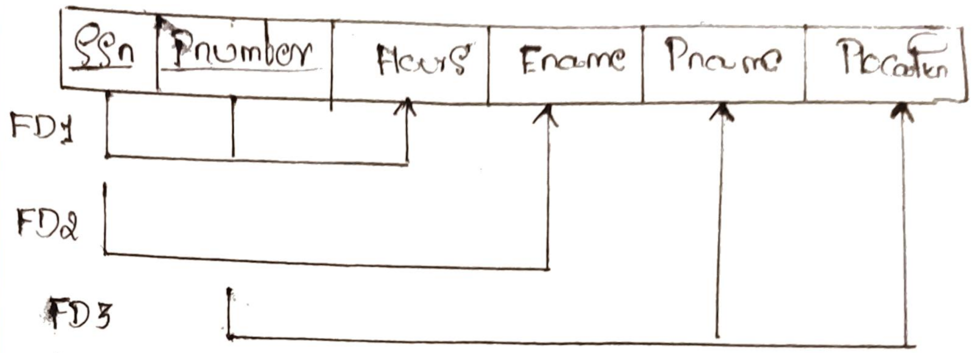
2NF : It is based on the Concept of ~~fr~~ functional dependancy.

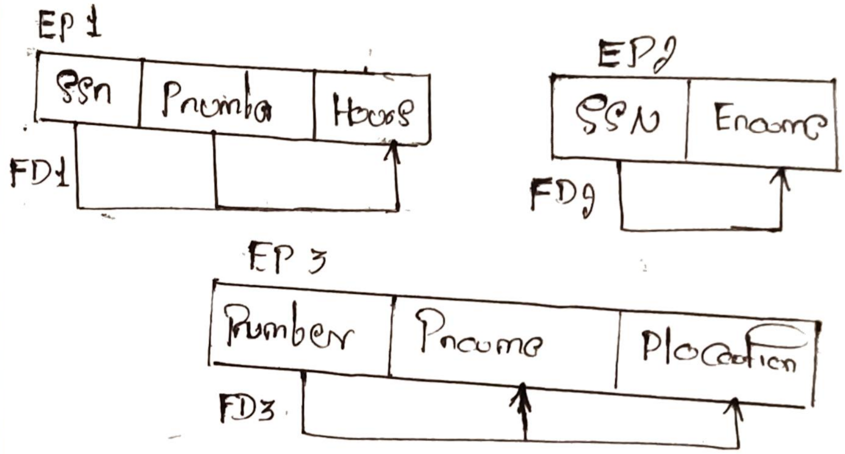
A functional dependency $X \rightarrow Y$ is a ~~fr~~ functional dependency if removal of any attribute A from X means the dependency does not held any more.

A relation Schema R is in 2NF is Every non-Prime attribute A is R.

Question Number	Solution	Marks Allocated
	Example:	

EMP_PROJ

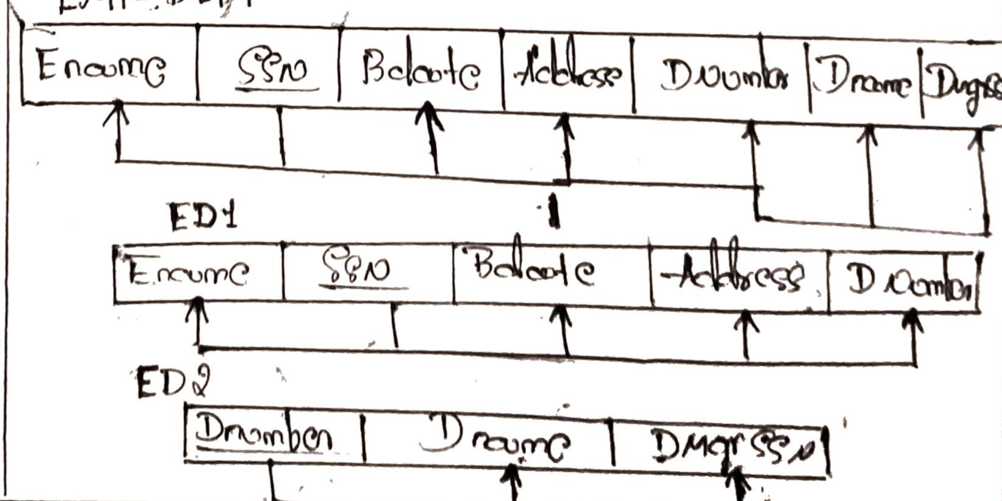




10M

3NF states that there should be no transitive dependency of a non key as the primary key — 3M

EMP_DEPT



Question Number	Solution	Marks Allocated
8.a)	**Guidence 1 :-** $1.5 * 4 = 6$ Design a Relation Schema so that it is Easy to Explain its meaning. Do not Combine the attributes from multiple Entity types & relationship types into a Single relation. **Guidence 2 :** Design the relation Schemas so that no insertion, deletion or modification anomalies are Present into a relation. **Guidence 3 :** As far as Possible avoid Placing the attributes in a base relation where values may frequently be null. **Guidence 4 :** Design the Relation Schemas So that they Can be joined with Equality Conditions on attributes that are appropriately related Pairs in a way that Guarantees no Spurious tuples are Generated.	6M
8.b	Defn → 2M Two Sets of functional dependencies E & F are Equivalent if $E^+ = F^+$. Therefore Equivalence means that Every FD in E Can be infered from F, and Every FD in F Can be infered from E.	

Question number	Solution	Marks Allocated
	Proof —— 6M Step 1 : Check wheter all FD's of FD1 are Present in FD2. A → B & B → C are common in FD1 & FD2 Lets check AB → D can be derived. For set FD2 $(AB)^+ = \{A, B, C, D\}$. It means that AB can functionally determine A, B, C & D So AB → D will also hold in set FD2. So, FD2 ⊇ FD1 true Step 2 : Checking whether all FD's of FD2 are present in FD1 A → B & B ⇒ are common. Let's check A → C can be derived or not For set FD1 $(A)^+ = \{A, B, C, D\}$ So A → ~~B~~ → C holds As all FD's in set FD2 also hold in set FD1, FD1 ⊇ FD2 is true. Step 3 : As FD1 ⊇ FD2 and FD2 ⊇ FD1 both are Equal.	8M

8.c) Determining X^+ the closure of X under F
Input: A set of FD's on relation schema R, and a set of attributes X, which is a subset of R.

$X^+ := X$;
repeat — 6M
old $X^+ := X^+$
for each functional dependency $Y \rightarrow Z$ in F do
if $X^+ \supseteq Y$ then $X^+ := X^+ \cup Z$;
until (X^+ = old X^+)

9.a) ~~Atomicity~~ with ~~Explanation~~ Explanation for each
Consistency 1.5 * 4 = 6 — 6M
Isolation
Durability

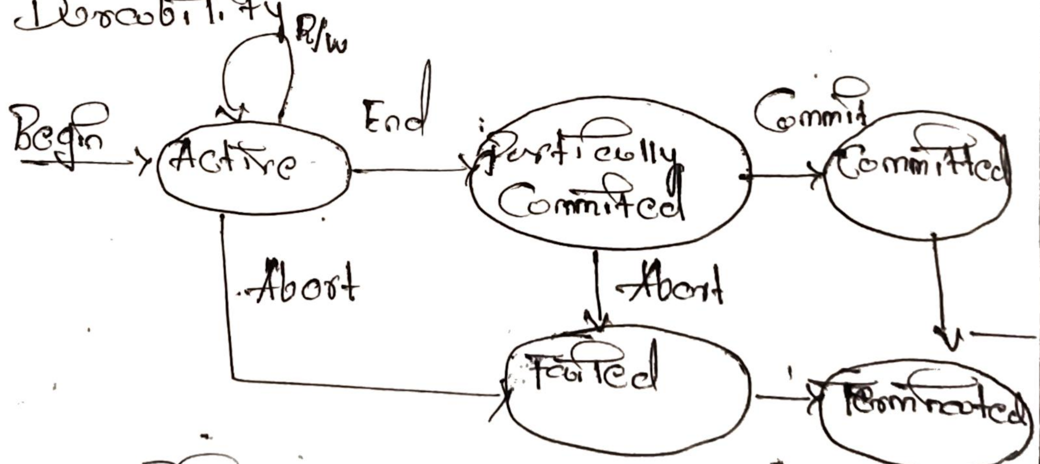
9.b)


Diagram → 4M
Explanation → 2M

Question Number	Solution	Marks Allocated
9.c	A transaction is said to follow two Phase following Protocol if all operations Precede the first unlock operation in transactions. Growing Phase → new locks are acquired Shrinking Phase → Existing locks are released. 4 Marks T1: read_lock(Y) read_item(y) unlock(y) write_lock(x) read_item(x) x = x+y write_item(x) unlock(x) T2: read_lock(x) read_item(x); unlock(x)); write_lock(Y); read_item(Y); y = x+y write_item(Y); unlock(Y) Two Transactions T1 & T2 without 2 Phase locking. 4 Marks T1': read_lock(Y) read_item(Y) write_lock(x) unlock(Y) read_item(x) x = x+y; write_item(x) T2': read_lock(x); read_item(x); write_lock(Y); unlock(x); read_item(Y); y = x+y write_item(Y);	8M

Question number	Solution	Marks Allocated
10.a)	T1 T2 Lost UPDate Problem T1: real = item (x) T1: x = x - n T2: real - item(x); T2: x = x + m; —— 1M T1: write - item (x) T1: real – item (y); T2: write_item (x); T1: y = y + n; T1: write . item (y);	
	T1 T2. T1: real = item (x); T1: x = x - n; T1: write . item (x); T2: real . item (x); T2: x = x + M; —— 1M T2: write - item (x); T1: real . item (y); —— temporary Update Problem	
	T1 T2 T2: sum = 0 T2: real (A) T2: sum = sum + A T1: real (x); T1: x = x - n; T1: write (x); —— 2M T2: real (x);	

10. b)	B: lock-item(x) if lock(x) = 0 then lock(x) ← 1 Else begin wait (until lock(x) = 0) and the lock manager wakes up the transactions. goto B. End Unlock-item(x) lock(x) ← 0 **Binary locks** — 4M	
	Shared locks. read-lock(x) B. if lock(x) = unlocked then begin lock(x) ← read no of read r(x) ← End Else if lock(x) = "read locked" — 3M no of reads = 1 Else begin wait until lock(x) = "unlocked" End. write-lock(x) B. if lock(x) = unlocked then lock(x) ← write locked. Else begin (until lock(x) And the lock manager wakes up its transaction goto B	10M