INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, SRI CITY

END-SEM EXAMINATION – MONSOON 2025

Database Management Systems

CSE:UG2 /PC **Date: 24-11-2025**

Duration: 180 Mins (2:30-5:30 PM) **Max. Marks: 45**

Instructions: **Roll No:**_____

1. All questions are compulsory.
2. Write the answers neatly and clearly without any overwriting or corrections on the answer sheet; otherwise, an appropriate penalty will be imposed.
3. Attach the question paper with the answer sheet

Answer all the questions

1	i. Which layer ensures data independence? a) Physical layer b) View layer **c) Logical layer** d) Application layer ii. A weak entity is identified by: **a) Partial key** b) Primary key c) Foreign key d) Composite key iii. Which of the following belongs to Domain Relational Calculus? a) { t | P(t) } **b) { <x1, x2,…> | P(x1, x2,…)}** c) σ(condition)3 d) π(attribute) iv. The symbol → indicates functional dependency in the context of a relational database. Which of the following options is/are TRUE? **(answer: B, C, D)** a) $(X,Y) \rightarrow (Z,W)$ implies $(X \rightarrow (Z,W))$.	**[10 Marks]**

b) $(X, Y) \rightarrow (Z, W)$ implies $(X, Y) \rightarrow Z$.

c) $((X, Y) \rightarrow Z$ and $W \rightarrow Y)$ implies $((X, W) \rightarrow Z)$.

d) $(X \rightarrow Y$ and $Y \rightarrow Z)$ implies $X \rightarrow Z$.

v. Consider the relation R(P, Q, S, T, X, Y, Z, W) with the following functional dependencies.

PQ → X; P → YX; Q → Y; Y → ZW

Consider the decomposition of the relation R into the constituent relations according to the following two decomposition schemes.

D1 : R = [(P, Q, S, T); (P, T, X); (Q, Y); (Y, Z, W)]

D2 : R = [(P, Q, S); (T, X); (Q, Y); (Y, Z, W)]

Which one of the following options is correct?

a) D1 is a lossy decomposition, but D2 is a lossless decomposition.

b) <u>D1 is a lossless decomposition, but D2 is a lossy decomposition.</u>

c) Both are lossy decompositions

d) Both are lossless decompositions

vi. In a relational data model, which one of the following statements is TRUE?

a) <u>A relation with only two attributes is always in BCNF.</u>

b) If all attributes of a relation are prime attributes, then the relation is in BCNF.

c) Every relation has at least one non-prime attribute

d) BCNF decompositions preserve functional dependencies

vii. Consider a relational table R that is in 3NF but not in BCNF. Which one of the following statements is TRUE?

a) <u>R has a nontrivial functional dependency $X \implies A$, where X is not a superkey and A is a prime attribute.</u>

b) R has a nontrivial functional dependency $X \implies A$, where X is not a superkey and A is a non-prime attribute and X is not a proper subset of any key.

c) R has a nontrivial functional dependency $X \implies A$ where X is not a superkey and A is a non-prime attribute and X is a proper subset of some key.

d) A cell in R holds a set instead of an atomic value.

viii. Which component of a DBMS ensures atomicity and durability?

a) Concurrency control manager

b) Transaction manager

c) Recovery manager

d) Buffer manager

ix. A schedule is cascadeless if:

a) No transaction reads data written by another uncommitted transaction

b) No transaction writes data written by another uncommitted transaction

c) Transactions commit only after all other transactions commit

d) No transaction has to wait for another transaction

x. The process of choosing a suitable execution strategy for processing a query is known as _____ .

a) Query Parsing

b) Query Optimization

c) Query Execution

d) Query Translation

2

a. What is the difference between Specialization and Generalization design process in DBMS **[3 Marks]**

Aspect	Specialization	Generalization
Approach	Top-Down	Bottom-Up
Meaning	Dividing a broad entity into more specific sub-entities	Combining specific entities into a general entity
Direction	One → Many	Many → One
Focus	Distinguishing differences among entities	Finding common features among entities
Example	Employee → Teacher, Clerk, Manager	Car, Bike → Vehicle
Diagram symbol	Triangle pointing down to subclasses	Triangle pointing up to superclass
Usage	When entities need extra attributes beyond the general ones	When different entities share similar attributes

b. Explain Mapping Cardinalities in detail. **[4 Marks]**

	• One to one • One to many • Many to one • Many to many 2 marks for listing cardinalities and Please consider 2 marks for each explanation.	
3	Consider the universal relation R = {A, B, C, D, E, F, G, H, I, J} and the set of functional dependencies F = {{A, B}→{C}, {A}→{D, E}, {B}→{F}, {F}→{G, H}, {D}→{I, J}}. What is the key for R? Decompose R into 2NF and then 3NF relations. **Solution:** A→DE (given) => A→D and A→ D Since A→D and D→IJ (given) => A→IJ Using the union rule A→ ADEIJ, thus AB → ABDEIJ (augmentation) Also AB→C (given) => AB→ ABCDEIJ. Since B→F (given) and F→ GH (given), B→GH (transitivity) Thus AB→AGH holds. Also AB→AF holds from B→F (given) Finally, using the union rule AB→ABCDEFGHIJ. So AB is a key. This can also be determined by calculating AB^+ with respect to the set F. 2NF: R1 (A, B, C); R2 (A,D,E,I,J); R3 (B, F, G, H) 3NF: R1 (A, B, C); R2.1 (A,D,E); R2.2 (D, I, J); R3.1 (B, F); R3.2 (F, G, H)	**[6 Marks]**
4	Consider the relation DISK_DRIVE (Serial_number, Manufacturer, Model, Batch, Capacity, Retailer). Each tuple in the relation DISK_DRIVE contains information about a disk drive with a unique Serial_number, made by a manufacturer, with a particular model number, released in a certain batch, which has a certain storage capacity and is sold by a certain retailer. For example, the tuple Disk_drive ('1978619', 'WesternDigital', 'A2235X', '765234', 500, 'CompUSA') specifies that WesternDigital made a disk drive with serial number 1978619 and model number A2235X, released in batch 765234; it is 500GB and sold by CompUSA. Write each of the following dependencies as an FD: a. The manufacturer and serial number uniquely identifies the drive. b. A model number is registered by a manufacturer and therefore can't be used by another manufacturer. c. All disk drives in a particular batch are the same model. d. All disk drives of a certain model of a particular manufacturer have exactly the same capacity. **FD:** $Serial\_number, Manufacturer \rightarrow Model, Batch, Capacity, Retailer$	**[4 Marks]**

	FD: $Model \rightarrow Manufacturer$ FD: $Batch \rightarrow Model$ FD: $Model, Manufacturer \rightarrow Capacity$ *Each correct FD 1 marks*	
5	Consider the following relation: CAR_SALE(Car#, Date_sold, Salesperson#, Commission%, Discount_amt) Assume that a car may be sold by multiple salespeople, and hence {Car#, Salesperson#} is the primary key. Additional dependencies are: Date_sold → Discount_amt and Salesperson# → Commission% Based on the given primary key, is this relation in 1NF, 2NF, or 3NF? Why or why not? How would you successively normalize it completely? 1) Yes in 1NF 2) Not in 2NF, **partial dependency exists Salesperson# → Commission%** 3) Not in 3NF**, transitive dependency Date_sold → Discount_amt**	**[6 Marks]**

	Step 1: Achieve 2NF (Remove Partial Dependency) We isolate the dependency $\mathbf{Salesperson\#} \rightarrow \text{Commission\%}$. **1. SALES_COMMISSION (New Relation)** • **Attributes:** $\text{Salesperson\#}, \text{Commission\%}$ • **PK:** $\{\mathbf{Salesperson\#}\}$ **2. CAR_SALE_1 (Remaining Relation)** • **Attributes:** $\text{Car\#}, \text{Salesperson\#}, \text{Date\_sold}, \text{Discount\_amt}$ • **PK:** $\{\mathbf{Car\#}, \mathbf{Salesperson\#}\}$ --- **Step 2: Achieve 3NF (Remove Transitive Dependency)** We now work on CAR_SALE_1 and isolate the transitive dependency $\text{Date\_sold} \rightarrow \text{Discount\_amt}$. **1. CAR_DISCOUNT (New Relation)** • **Attributes:** $\text{Date\_sold}, \text{Discount\_amt}$ • **PK:** $\{\mathbf{Date\_sold}\}$ **2. CAR_SALE_2 (Final Fact Relation)** • **Attributes:** $\text{Car\#}, \text{Salesperson\#}, \text{Date\_sold}$ • **PK:** $\{\mathbf{Car\#}, \mathbf{Salesperson\#}\}$	
6	A large relation R occupies 520 file blocks on disk. The system has 11 buffer blocks available in main memory for external sorting. Using the external sort–merge strategy, answer the following: (i) Compute the number of initial runs formed during the sorting phase. (ii) Determine the degree of merging during the merge phase. (iii) Compute the number of merge passes required to completely sort the file. (iv) Find the total number of passes, including the initial sorting pass. Ans. **Each part carries 1 mark.** (i) Number of initial runs n_R Formula: $n\_R = \lceil b/n\_B \rceil$ Compute 520÷11(digit-by-digit):	**[4 Marks]**

	11×40=440→ remainder 520-440=80 11×7=77→ remainder 80-77=3 So 520÷11=47remainder 3, i.e. 47 3/11. Ceiling gives: (n_R=48) (ii) Degree of merging d_M Formula: d_M=min(n_B-1, n_R) Calculate n_B-1=11-1=10. Compare with n_R=48. So min(10,48)=10. (d_M=10) (iii) Number of merge passes n_P Formula: n_P=⌈〖log〗_(d_M) (n_R)⌉ Here d_M=10 and n_R=48. Reasoning: 10^1=10and 10^2=100. Since 10<48≤100, 〖log〗_10 (48)lies between 1 and 2. Therefore the ceiling is 2. (n_P=2) (iv) Total number of passes (including initial sorting pass) Total passes =1(initial sort pass) +n_P(merge passes) =1+2=3.	
7	Consider the three transactions T1, T2, and T3, and the schedules S1 and S2 given below. Draw the serializability (precedence) graphs for S1 and S2, and state whether each schedule is serializable or not. If a schedule is serializable, write down the equivalent serial schedule(s). (Check for both conflict and view serializability). T1: r1 (X); r1 (Z); w1 (X); T2: r2 (Z); r2 (Y); w2 (Z); w2 (Y); T3: r3 (X); r3 (Y); w3 (Y); S1: r1 (X); r2 (Z); r1 (Z); r3 (X); r3 (Y); w1 (X); w3 (Y); r2 (Y); w2 (Z); w2 (Y); S2: r1 (X); r2 (Z); r3 (X); r1 (Z); r2 (Y); r3 (Y); w1 (X); w2 (Z); w3 (Y); w2 (Y); **Ans: Each schedule carries 4marks.** **Schedule S1.** On X r3(X) (step 4) occurs before w1(X) (step 6) and one is a write ⇒ edge T3 → T1. On Z r1(Z) (step 3) occurs before w2(Z) (step 9) and one is a write ⇒ edge T1 → T2. On Y w3(Y) (step 7) occurs before r2(Y) (step 8) ⇒ edge T3 → T2. w3(Y) (7) also occurs before w2(Y) (10) ⇒ edge T3 → T2 (same edge). Precedence graph for S1 Edges:	**[8 Marks]**

T3 → T1
T1 → T2
T3 → T2 (redundant transitively via T3→T1→T2)

This graph has no cycle.

Conclusion for S1
Conflict-serializable: Yes.

A topological order consistent with edges is T3 → T1 → T2.
So an equivalent serial schedule is (T3, T1, T2). (This order is forced by the edges — T3 must precede T1 and T2; T1 must precede T2.)

View-serializable: Yes (every conflict-serializable schedule is view-serializable).
You can check reads and final-writers match the serial order (T3, T1, T2).

Schedule S2:
Find conflicts (by data item)
On X
r3(X) (3) before w1(X) (7) ⇒ edge T3 → T1.
On Z
r1(Z) (4) before w2(Z) (8) ⇒ edge T1 → T2.
On Y
r2(Y) (5) occurs before w3(Y) (9) and one is a write ⇒ edge T2 → T3.
w3(Y) (9) occurs before w2(Y) (10) ⇒ edge T3 → T2.
So between T2 and T3 we have both T2 → T3 and T3 → T2.

Precedence graph for S2

Edges:
T3 → T1
T1 → T2
T2 → T3
T3 → T2 (both directions between T2 and T3)

Because T1 → T2 → T3 → T1 (via T3 → T1) we have a cycle among all three (and specifically a 2-cycle between T2 and T3).

Conclusion for S2 (conflict)

Conflict-serializable: No — the precedence graph contains cycles (T1 → T2 → T3 → T1 and T2 ↔ T3).

Check view-serializability for S2

We also check view constraints briefly (reads-from and final-writer consistency):

	Final writers in S2: X final writer = T1 (w1 at step 7) Z final writer = T2 (w2 at step 8) Y final writer = T2 (w2 at step 10) Look at read-from relations: e.g. r2(Y) (step 5, in T2) reads the initial value (there is no earlier write to Y). For a serial order to produce the same read behavior, no transaction before T2 in that serial order may write Y. But T3 writes Y (and in the real schedule T3 writes later), and the final writer for Y must be T2 in the serial order (so T2 must come after any other Y-writer). These constraints conflict: r2(Y) wants T2 to be before any Y-writer, but final-writer being T2 forces T2 to be after other Y-writers. Thus there is no serial order that preserves the same read-from and final-writer relations. Hence S2 is not view-serializable either.	