Okay, here are the questions and concise answers for the remaining part of the section:

(b) Q: Discuss self-join with a suitable example.
A: A self-join is a join operation where a table is joined with itself. It's used to compare rows within the same table based on related attributes.
* Example: An Employee table with EmpID and ManagerID. A self-join can retrieve each employee along with their manager's information by joining Employee as E1 and Employee as E2 where E1.ManagerID = E2.EmpID.
(c) Q: List the problems that can arise due to bad database design.
A: Problems include data redundancy, update anomalies (insertion, deletion, modification), data inconsistency, difficulty in querying and reporting, increased storage space, and poor performance.
(d) Q: Compare and contrast aggregation and composition with suitable examples.
A:
* Aggregation: Represents a "has-a" relationship where the related entities can exist independently. The part can exist without the whole. (e.g., a Department has Employees; an employee can exist even if the department is dissolved). Represented by a hollow diamond in ER.
* Composition: A stronger form of aggregation representing a "part-of" relationship with exclusive ownership and coincident lifetime. The part cannot exist without the whole. (e.g., a Wall is part of a Room; if the room is destroyed, the wall ceases to exist as part of that room). Represented by a filled diamond in ER.
(e) Q: Consider the relation R(A,B,C,D,E,F,G,H) and set of FDs: F={A->BC, CD->E, E->C, D->AEH, AB->BD, DH->BC, AEG->G}. Is BCD->H a valid FD?
A: To determine if BCD->H is a valid FD, we need to check if H is in the closure of {B, C, D} under the given set of functional dependencies.
1. {B, C, D}+ initially = {B, C, D}
2. Using CD->E, {B, C, D}+ = {B, C, D, E}
3. Using E->C (already present)
4. Using D->AEH, {B, C, D}+ = {B, C, D, E, A, H}
Since H is in the closure of {B, C, D}, BCD->H is a valid functional dependency.
(f) Q: Distinguish between centralized and distributed databases.
A:
* Centralized Database: A database stored and managed in a single location. All users access this central database. Easier to manage but can have performance bottlenecks and a single point of failure.
* Distributed Database: A database where data is spread across multiple interconnected computers or sites. Offers improved scalability, availability, and fault tolerance but is more complex to manage and ensure data consistency.
(g) Q: Outline the significance of 'D' in ACID properties of a transaction.
A: 'D' in ACID stands for Durability. It ensures that once a transaction is committed, the changes made to the database are permanent and will survive subsequent failures (e.g.,

power outages, system crashes). The database system takes steps to persist these changes to stable storage.
(h) Q: Define data independence. List different types of data independence.
A: Data independence is the capacity to modify the schema at one level of the database system without affecting the schema at the next higher level.
* Types of Data Independence:
* Logical Data Independence: The ability to change the logical schema (conceptual schema) without affecting the application programs.
* Physical Data Independence: The ability to change the physical schema (internal schema) without affecting the logical schema or application programs.
(h) (ii) Q: Let P, Q and S be three different entities in that ER diagram comprising single-valued attributes, except for S, which has only one multi-valued attribute. Let R1 and R2 be two relationships between P and Q, where R1 is one-to-one while R2 is many-to-many. Let R3 and R4 be two relationships between Q and S, where R3 is many-to-one while R4 is many-to-many. Relationships R1, R2, R3 and R4 do not have any attributes of their own. How many minimum number of tables required to map this scenario into a relational model?
A:
* Entity P: 1 table
* Entity Q: 1 table
* Entity S (with one multivalued attribute): 2 tables (one for S's single-valued attributes and primary key, and another for the multivalued attribute with a foreign key to S).
* Relationship R1 (one-to-one): Can be merged into either the P or Q table (adding a foreign key).
* Relationship R2 (many-to-many): Requires 1 separate table with foreign keys from P and Q.
* Relationship R3 (many-to-one from Q to S): Can be implemented by adding a foreign key from Q to S's main table.
* Relationship R4 (many-to-many): Requires 1 separate table with foreign keys from Q and S's main table.
* Minimum number of tables: 1 (P) + 1 (Q) + 2 (S) + 1 (R2) + 1 (R4) = 6 tables. (R1 merged, R3 as foreign key).
(i) Q: List set theory based relational algebra operations with suitable examples.
A:
* UNION ($\cup$): Combines tuples from two relations, eliminating duplicates. (e.g., StudentsInCSE $\cup$ StudentsInECE gives all students in either CSE or ECE).
* INTERSECTION ($\cap$): Returns tuples that are present in both relations. (e.g., StudentsInCSE $\cap$ StudentsInDSA gives students in both CSE and DSA courses).
* SET DIFFERENCE (-): Returns tuples present in the first relation but not in the second. (e.g., AllStudents - GraduatedStudents gives currently enrolled students).
* CARTESIAN PRODUCT ($\times$): Combines every tuple of the first relation with every tuple of the second relation. (e.g., Students $\times$ Courses creates all possible pairings of students and courses).
(j) Q: Which of the following concurrency control protocols ensure both conflict serializability and freedom from deadlock?

A:
* I. 2-phase locking: Ensures conflict serializability but can lead to deadlocks.
* II. Time-stamp ordering: Ensures conflict serializability and is deadlock-free.
* Answer: B. II only