Okay, I understand. Here are the questions and concise answers for Section A:

1. Answer the following questions.

(a) Q: What is timestamp? How is the timestamp assigned to a transaction and to a data item?
A: A timestamp is a unique identifier assigned to a transaction or a data item, indicating its relative time of creation or access. Transactions are typically assigned timestamps when they begin. Data items can be assigned timestamps when they are created or last modified by a transaction.
(b) Q: Draw an entity set having one multivalued attribute and one composite attribute and then translate the entity set into relations.
A: (Imagine an ER diagram here)
* Entity Set: Person with attributes: Name (composite: FirstName, LastName), PhoneNumbers (multivalued), Address.
* Translation to Relations:
* Person (PersonID, FirstName, LastName, Address) - PersonID is the primary key.
* PersonPhone (PersonID, PhoneNumber) - (PersonID, PhoneNumber) is the primary key, PersonID is a foreign key referencing Person.
(c) Q: Define a schedule in transaction with an appropriate example.
A: A schedule is a chronological sequence of operations (read, write) performed by a set of concurrent transactions.
* Example:
* T1: read(A), write(A)
* T2: read(B), write(B)
* Schedule S: read$_1$(A), write$_1$(A), read$_2$(B), write$_2$(B)
(d) Q: Differentiate between Left Outer Join and Right Outer Join with a suitable example.
A:
* Left Outer Join: Returns all rows from the left table and the matching rows from the right table. If there's no match in the right table, NULLs1 are used for the right table's columns.
* Right Outer Join: Returns all rows from the right table and the matching rows from the left table. If there's no match in the left table, NULLs2 are used for the left table's columns.
* Example: (Imagine Tables Customers and Orders) A Left Outer Join would show all customers, even those with no orders. A Right Outer Join would show all orders, even if the customer information is missing.
(e) Q: What is the need of a data model in DBMS?
A: A data model provides a conceptual blueprint of the database structure, defining how data is organized, stored, and accessed. It helps in understanding the data requirements and facilitates communication between users, designers, and developers.
(f) Q: What do you mean by super key and candidate key?
A:
* Super Key: A set of one or more attributes that can uniquely identify a tuple in a relation.
* Candidate Key: A minimal super key, meaning no proper subset of it is also a super key. A relation can have multiple candidate keys.

(g) Q: Describe when a relation is in 2NF.
A: A relation is in 2NF (Second Normal Form) if it is in 1NF (First Normal Form) and every non-prime attribute (an attribute that is not part of any candidate key) is fully functionally dependent on each candidate key.
(h) Q: Differentiate between logical data independence and physical data independence.
A:
* Logical Data Independence: The ability to modify the logical schema (e.g., adding new attributes, changing relationships) without affecting the application programs.
* Physical Data Independence: The ability to modify the physical schema (e.g., changing storage structures, access methods) without affecting the logical schema or application programs.
(i) Q: Differentiate between Strong entity and Weak entity.
A:
* Strong Entity: An entity that has its own primary key and exists independently.
* Weak Entity: An entity that cannot be uniquely identified by its own attributes and depends on another (owner) entity for its existence and identification. It typically has a partial key and uses the primary key of the owner entity as part of its primary key.
(j) Q: Define entity integrity constraint.
A: The entity integrity constraint states that the primary key of a relation cannot contain null values. This ensures that each tuple in the relation can be uniquely identified.