Part A.

Question 1. Discuss the differences between DDL and DML. What operations would you typically expect to be available in each language?

Data Definition Language or DDL statements are used to define the database structure or schema. Some examples:
1. CREATE - Used to create a new table in the database.
2. ALTER - Used to alter the structure of tables in the database.
3. DROP - Used to delete a table or an object in the database.
4. TRUNCATE - Used to remove all records from a table, including all spaces allocated for the records are removed.

Data Manipulation Language or DML statements are used for changing/manipulating/updating data within database tables or objects. Some examples:
1. SELECT - Used to retrieve data from a database.
2. INSERT - Used to insert data into a table.
3. UPDATE - Updates existing data within a table.
4. DELETE - Deletes all records from a table, the space for the records remain.

Question 2. Describe the difference between data security and data integrity.

Data security refers to the protection of data against unauthorized access or corruption and is necessary to ensure data integrity. Data integrity refers to the accuracy or consistency or validity.
Data integrity is a result of data security; however, data integrity refers only to the validity and accuracy of data rather than the act of protecting data. Data security, in other words, is one of several measures which can be employed to maintain data integrity, since unauthorized access to sensitive data may lead to corruption of records, data loss as well as loss of validity of data. So, data security plays an important role in maintaining data integrity.

Question 3. Describe the main characteristics of the database approach and contrast it with the file based approach.

According to the database approach instead of storing separate and disconnected files with redundant data, all data items are integrated with a minimum amount of duplication and stored in one location.
The database is no longer owned by one department but is a shared corporate resource. In contrast, each file in a file based system, belongs to a department.
The database holds not only the organization's operational data but also a description of this data. The definition of data is separated from the application programs.
Due to Data Abstraction, if new data structures are added or existing structures are modified then the application programs are unaffected, provided they do not directly depend upon what has been modified, which is difficult to achieve in File Based Systems. Finally, according to the database approach, data is represented as entities. An entity can be any object.
An entity is described using various descriptors called as attributes. If two entities are logically related, then there exists a relationship between them. While in a File Based System, all data is simple represented as text, and we need application programs to parse this data to identify objects, their attributes and the relationships, if any, between different objects.

Question 4. Provide a definition for a data administrator and a database administrator. What types of interactions would these two users of the database have?

The Data Administrator (DA) is responsible for the management of the data resource. The DA performs the flowing tasks: database planning, development and maintenance of standards, policies and procedures, and conceptual/logical database design. The DA is also responsible for consulting with and advising senior managers, ensuring that the direction of database development will ultimately support corporate objectives.

The Database Administrator (DBA) is responsible for the physical realization of the database. The DBA performs the flowing tasks: physical database design and implementation, security and integrity control, maintenance of the operational system, and ensuring satisfactory performance of the applications for users.

Interaction between the DA and DBA:

A DA you typically define a plan, set of goals and data requirements, and communicate these to the DBA, who would be responsible for implementing and enforcing them.

While the DBA would ensure that plan, goals and requirements defined by the DA are realizable or not. If not, the DBA, would inform the DA about the same. The DBA may also provide some suggestion on how the requirements could be updated or modified, so that they are more technically sound.

Question 5. Name three record-based data models. Discuss the main differences between these data models.

Record Based Models:

1. Hierarchical Model
2. Network Model
3. Relational Model

Hierarchical Data Model	**Network Data Models**	**Relational Data Models**
Based on the concept of trees.	Based on the concept of graphs.	Based on the concept of mathematical relations.
Data is represented as collections of records and relationships are represented by sets. However, the hierarchical model allows a node to have only one parent.	In the network model, data is represented as collections of records, and relationships are represented by sets.	In the relational model, data and relationships are represented as tables, each of which has many columns with a unique name
Records are organized into a tree-like structure where each child record has only one parent, whereas each parent record can have one or more child records	The records are organized as generalized graph structures with records appearing as nodes (also called segments) and sets as edges in the graph.	Records are organized as rows in tables. There is no explicit link between two relations, except if the same attribute exists in the two relations we can establish that a relationship exists.
adopt a navigational approach i.e., they specify how the data is to be retrieved.	adopt a navigational approach i.e., they specify how the data is to be retrieved.	adopt a declarative approach to database processing, i.e., they specify what data is to be retrieved.

Requires the user to have knowledge of the physical database being accessed	Requires the user to have knowledge of the physical database being accessed	user need not have knowledge of the physical database being accessed. Hence higher data independence.
Uses Procedural DML	Uses Procedural DML	Uses Non-Procedural DML

Question 6. What are the advantages of a relational database when compared to the file-based approach to storing data?

1. Control of Data Redundancy
   Traditional file-based systems waste space by storing the same information in more than one file. In contrast, the database approach attempts to eliminate the redundancy by integrating the files so that multiple copies of the same data are not stored. However, the database approach does not eliminate redundancy entirely, but controls the amount of redundancy inherent in the database.

2. Data Consistency
   By eliminating or controlling redundancy, we reduce the risk of inconsistencies occurring. If a data item is stored only once in the database, any update to its value must be performed only once and the new value is available immediately to all users. If a data item is stored more than once and the system is aware of this, the system can ensure that all copies of the item are kept consistent.

3. More information from the same amount of data
   With the integration of the operational data, it may be possible for the organization to derive additional information from the same data. This is not possible in file systems since there is no relationship amongst different files.

4. Sharing of Data
   Typically, files are owned by the people or departments that use them. On the other hand, the database belongs to the entire organization and can be shared by all authorized users. In this way, more users share more of the data.

5. Improved maintenance through data independence
   In file-based systems, the descriptions of the data and the logic for accessing the data are built into each application program, making the programs dependent on the data. In contrast, a DBMS separates the data descriptions from the applications, thereby making applications immune to changes in the data descriptions. This is known as data independence. The provision of data independence simplifies database application maintenance.

Question 7. What is concurrency control and why does a DBMS need a concurrency control facility?

Concurrency Control is the process of managing simultaneous operations on the database without having them interfere with one another.

Need for Concurrency Control:
A major objective in developing a database is to enable many users to access shared data concurrently. Concurrent access is relatively easy if all users are only reading data, as there is no way that they can

interfere with one another. However, when two or more users are accessing the database simultaneously and at least one is updating data, there may be interference that can result in inconsistencies.
Consider two transactions executing simultaneously. The system begins executing the first transaction until it reaches an I /O operation. While the I/O is being performed, the CPU suspends the first transaction and executes commands from the second transaction. When the second transaction reaches an I/O operation, control then returns to the first transaction and its operations are resumed from the point at which it was suspended. The first transaction continues until it again reaches another I /O operation. In this way, the operations of the two transactions are interleaved to achieve concurrent execution.
However, although two transactions may be perfectly correct in themselves, the interleaving of operations in this way may produce an incorrect result, thus compromising the integrity and consistency of the database.

Question 8. What is a transaction? Give an example of a transaction.

A transaction is a series of actions, carried out by a single user or application program, which accesses or changes the contents of the database. For example, some simple transactions for a University Database might be to add a new member of staff to the database, to update the salary of a member of staff, or to delete a course from the register.

Question 9. What is meant by the term 'client-server architecture' and what are the advantages of this approach? Compare the client-server architecture with two other architectures.

Client–server refers to the way in which software components interact to form a system. As the name suggests, there is a client process, which requires some resource, and a server, which provides the resource. There is no requirement that the client and server must reside on the same machine. In practice, it is quite common to place a server at one site in a local area network and the clients at the other sites. The client (tier 1) is primarily responsible for the presentation of data to the user, and the server (tier 2) is primarily responsible for supplying data services to the client.

<u>Advantages of Client-Server Architecture:</u>

1. Increased performance – if the clients and server reside on different computers then different CPUs can be processing applications in parallel. It should also be easier to tune the server machine if its only task is to perform database processing.
2. Communication costs are reduced – applications carry out part of the operations on the client and send only requests for database access across the network, resulting in less data being sent across the network.
3. Increased consistency – the server can handle integrity checks, so that constraints need be defined and validated only in the one place, rather than having each application program perform its own checking.

Teleprocessing	File-Server Architecture	Client–Server Architecture
There is one computer with a single central processing unit (CPU) and many terminals.	In a file-server environment, the processing is distributed about the network, typically a local area network (LAN).	There is a client process, which requires some resource, and a server, which provides the resource.
All processing is performed within the boundaries of the same physical computer. User terminals are typically 'dumb' ones, incapable of functioning on their own. They are cabled to the central computer.	The file-server holds the files required by the applications and the DBMS. However, the applications and the DBMS run on each workstation, requesting files from the file-server when necessary.	The client is primarily responsible for the presentation of data to the user, and the server is primarily responsible for supplying data services to the client
This architecture placed a tremendous burden on the central computer, which not only had to run the application programs and the DBMS, but also had to carry out a significant amount of work on behalf of the terminals	There is a large amount of network traffic. A full copy of the DBMS is required on each workstation. Concurrency, recovery, and integrity control are complex.	The server acts as a single point of failure. Traffic congestion is a problem when many simultaneous clients send requests to the same server.

Question 10. What is a Transaction Processing Monitor? What advantages does a TP Monitor bring to an OLTP environment?

A Transaction Processing Monitor is a program that controls data transfer between clients and servers to provide a consistent environment, particularly for online transaction processing (OLTP). A Transaction Processing Monitor, or TP Monitor, is a middleware component that provides access to the services of many resource managers and provides a uniform interface for programmers who are developing transactional software.

Advantages:

1. Transaction routing The TP Monitor can increase scalability by directing transactions to specific DBMSs.
2. Managing distributed transactions, The TP Monitor can manage transactions that require access to data held in multiple, possibly heterogeneous, DBMSs.
3. Load balancing The TP Monitor can balance client requests across multiple DBMSs on one or more computers by directing client service calls to the least loaded server.
4. Increased reliability The TP Monitor acts as a transaction manager, performing the necessary actions to maintain the consistency of the database, with the DBMS acting as a resource manager.