1. What is a Database? Explain with an example on why should we need a database.

A database is a structured collection of data that is organized for easy access, storage, and manipulation. It is like a big filing cabinet where you can store information in a way that makes it easy to find what you need. For example, a library might use a database to store information about all of its books, such as the title, author, publication date, and genre. This makes it easy for librarians and patrons to find the books they are looking for.

2. Write a short note on File base storage system. Explain the major challenges of a File-based storage system.

A file-based storage system is a way of storing data in individual files. Each file is like a separate container that holds a piece of information. For example, a document on your computer might be stored as a file. File-based storage systems are simple and easy to understand, but they have some major challenges.

* Data redundancy: The same data may be stored in multiple files, which can waste storage space and make it difficult to keep the data consistent.
* Data isolation: It can be difficult to find and access related data that is stored in different files.
* Data integrity: It can be difficult to ensure that the data in a file is accurate and up-to-date.
* Scalability: It can be difficult to add more data to a file-based storage system as it grows.

3. What is DBMS? What was the need for DBMS?

A DBMS (Database Management System) is a software application that is used to create, manage, and access databases. It provides a way to organize, store, retrieve, and manipulate data in a structured and efficient way. DBMSs were developed to overcome the challenges of file-based storage systems.

4. Explain 5 challenges of file-based storage system which was tackled by DBMS.

1. Data redundancy: DBMSs use schemas to define the structure of data and enforce data integrity rules, which helps to prevent data redundancy.
2. Data isolation: DBMSs use queries to access and manipulate data, which makes it easy to find and access related data.
3. Data integrity: DBMSs use transactions and locking mechanisms to ensure that data is accurate and up-to-date.
4. Scalability: DBMSs are designed to scale up to accommodate large amounts of data.
5. Security: DBMSs provide security features that can help to protect data from unauthorized access.

5. List out the different types of classification in DBMS and explain them in depth.

There are two main types of DBMS classifications:

* By data model:
  + Relational databases: Store data in tables with rows and columns.
  + NoSQL databases: Do not use a fixed schema and are more flexible than relational databases.
  + Hierarchical databases: Organize data in a tree-like structure.
  + Network databases: Organize data in a network of nodes and relationships.
  + Object-oriented databases: Store data in objects that encapsulate both data and code.
* By deployment:
  + Centralized databases: All data is stored in a single location.
  + Decentralized databases: Data is distributed across multiple locations.

6. What is the significance of Data Modelling and explain the types of data modeling

Data modeling is the process of creating a blueprint for a database. It involves defining the structure of the data, the relationships between different pieces of data, and the constraints that will be applied to the data. There are three main types of data modeling:

* Entity-relationship (ER) modeling: Identifies the entities (things) that need to be stored in the database and the relationships between those entities.
* Relational modeling: Defines the tables, columns, and relationships that will be used to store the data in a relational database.
* Object-oriented modeling: Defines the objects, classes, and relationships that will be used to store the data in an object-oriented database.

7. Explain 3 schema architecture along with its advantages.

A schema is a definition of the structure of a database. There are three main levels of schema in a DBMS:

* Internal schema: Defines the physical storage of data, such as how data is stored on disk.
* Conceptual schema: Defines the logical structure of data, such as the tables, columns, and relationships between tables.
* External schema: Defines the view of the data that is seen by users, such as the specific columns and tables that a particular user is allowed to access.

The advantages of using a three-schema architecture:

1. Data Independence: Changes made to one schema level (internal, conceptual, or external) do not necessarily require changes to the other levels. This allows for flexibility and avoids cascading updates throughout the entire system.

2. Modular Development: Different teams can work on different schema levels independently, improving development efficiency and speed.

3. Enhanced Security: Users only see the data they are authorized to access through their external schema, restricting unauthorized access to sensitive information.

4. Improved Performance: The internal schema can be optimized for specific storage and retrieval needs, leading to faster data access and better overall system performance.

5. Data Consistency and Integrity: Data integrity rules can be enforced at the conceptual schema level, ensuring consistency across all user views and preventing data anomalies.

6. Support for Database Evolution: The three-schema architecture allows for gradual changes to the database structure over time without impacting users or applications relying on the external schema.

7. Easier Maintenance: By separating concerns across different schema levels, maintenance and troubleshooting become more focused and less complex.

While the three-schema architecture offers significant advantages, it's important to note that it also introduces some complexity and overhead compared to simpler database designs. The decision of whether to use this approach depends on the specific needs and requirements of your project.