**Chapter 1**

**Database Terms**

1. **Database-management system (DBMS)**  
   A software system that enables the creation, management, and manipulation of databases. Examples include MySQL, PostgreSQL, and Oracle.
2. **Database-system applications**  
   Software applications that utilize databases for storing, managing, and retrieving data. Examples include inventory systems, CRM software, and e-commerce platforms.
3. **File-processing systems**  
   Early systems for storing and managing data in files, often lacking integration, leading to data redundancy and inconsistency.
4. **Data inconsistency**  
   When different versions of the same data exist in different places due to poor data management practices.
5. **Consistency constraints**  
   Rules that ensure data remains accurate and consistent across the database. Examples include primary key constraints and foreign key constraints.
6. **Data abstraction**  
   The process of hiding low-level details and providing a simplified interface to interact with data, typically in levels (physical, logical, and view).
7. **Schema**  
   The overall structure or blueprint of a database, including tables, fields, and relationships.
8. **Instance**  
   A snapshot of the data in the database at a particular point in time.
9. **Physical Schema**  
   Describes how data is stored physically on storage devices.
10. **Logical Schema**  
    Describes the structure of the database logically, focusing on tables, fields, and relationships without considering physical storage.

**Data Models**

1. **Entity-relationship model (ER model)**  
   A conceptual data model that represents entities and the relationships between them using diagrams.
2. **Object-based data model**  
   Represents data in terms of objects, similar to object-oriented programming, where data and behavior are encapsulated together.
3. **Relational data model**  
   Represents data in tables (relations) with rows (tuples) and columns (attributes), widely used in modern databases.
4. **Semistructured data model**  
   Data model where the structure is flexible and not strictly defined, often seen in JSON and XML.

**Database Languages**

1. **Data-definition language (DDL)**  
   A set of SQL commands used to define the structure of a database, such as CREATE, ALTER, and DROP.
2. **Data-manipulation language (DML)**  
   A set of SQL commands used to modify data in the database, such as INSERT, UPDATE, and DELETE.
3. **Query language**  
   A language used to retrieve and manipulate data. SQL (Structured Query Language) is the most common example.

**Other Key Terms**

1. **Metadata**  
   Data that describes other data, such as schema definitions, field names, and data types.
2. **Application program**  
   Software that interacts with the database to perform specific tasks, such as data entry or report generation.
3. **Normalization**  
   The process of organizing data to minimize redundancy and improve data integrity.
4. **Data dictionary**  
   A repository that stores metadata, including information about tables, fields, and relationships in a database.
5. **Storage manager**  
   The component of a DBMS responsible for storing and retrieving data from storage devices.
6. **Query processor**  
   The component of a DBMS that interprets and executes queries, optimizing performance.
7. **Transactions**  
   A sequence of operations performed as a single unit of work to maintain database consistency.
8. **Atomicity**  
   Ensures that a transaction is either fully completed or not executed at all, maintaining integrity.
9. **Failure recovery**  
   Mechanisms for restoring a database to a consistent state after a failure, such as backups and logging.
10. **Concurrency control**  
    Techniques to manage simultaneous database access to ensure consistency and avoid conflicts.

**Architecture and Roles**

1. **Two- and three-tier database architectures**  
   Database system designs with layers such as client, application server, and database server to improve scalability and manageability.
2. **Data mining**  
   The process of discovering patterns and insights from large datasets using algorithms and statistical methods.
3. **Database administrator (DBA)**  
   The person responsible for managing, securing, and maintaining a database system.

Practice Exercises

**1.1 What are two disadvantages of database systems?**

1. **Complexity**
   * Database systems are complex to design, implement, and maintain. The use of sophisticated software and extensive design processes can lead to a steeper learning curve and longer development times.
2. **Cost**
   * Database systems can be expensive in terms of both software licensing and hardware requirements. Maintaining the system requires investment in skilled personnel, storage infrastructure, and ongoing support.

**1.2 List five ways in which the type declaration system of a language such as Java or C++ differs from the data definition language (DDL) used in a database.**

1. **Purpose**
   * **Java/C++**: Designed for defining data types within programs to facilitate computation and execution.
   * **DDL**: Designed for defining the structure and schema of databases (e.g., tables, columns, and relationships).
2. **Persistence**
   * **Java/C++**: Data types and objects are typically stored in volatile memory (RAM) and do not persist once the program terminates.
   * **DDL**: Database schemas and data persist on disk storage across sessions until explicitly modified or deleted.
3. **Data Integrity Constraints**
   * **Java/C++**: Type declarations are used for type safety during compilation and runtime, but they do not enforce constraints like primary keys or foreign keys.
   * **DDL**: Supports integrity constraints like primary keys, foreign keys, and uniqueness constraints to maintain data integrity.
4. **Scope of Definition**
   * **Java/C++**: Type declarations apply to variables, objects, and functions within the scope of a program.
   * **DDL**: Defines the structure of an entire database, affecting how data is stored, retrieved, and maintained.
5. **Modification**
   * **Java/C++**: Changes to data types require code recompilation.
   * **DDL**: Schema changes can often be made dynamically with commands like ALTER TABLE.