1. A Relational Database Management System (RDBMS) is a software program that allows you to create, update, and manage relational databases. A relational database is a type of database that organizes data into one or more tables with a unique key identifying each row. The most common language used to access RDBMS is SQL (Structured Query Language).

Here are some advantages of using a database management system over a file system:

Data redundancy and inconsistency: In a file system, data is stored in separate files, which can lead to data redundancy and inconsistency. In contrast, DBMS maintains a single repository of data that is defined once and accessed by many users. As there is no or less redundancy, data remains consistent.

Data sharing: Sharing data in a file system can be complex, whereas in DBMS, data can be shared easily due to a centralized system.

Data concurrency: Concurrent access to data means more than one user is accessing the same data at the same time. Anomalies occur when changes made by one user get lost because of changes made by another user. DBMS provides a locking system to stop anomalies from occurring.

Data searching: For every search operation performed on the file system, a different application program has to be written. While DBMS provides inbuilt searching operations. The user only has to write a small query to retrieve data from the database.

Data integrity: There may be cases when some constraints need to be applied to the data before inserting it into the database. The file system does not provide any procedure to check these constraints automatically. Whereas DBMS has the functionality to check the constraints on the data and allows user-defined data types

2. In a database management system, ACID properties are a set of four properties that ensure the reliability and consistency of a transaction. The acronym ACID stands for Atomicity, Consistency, Isolation, and Durability.

Atomicity: This property ensures that a transaction is treated as a single logical unit of work. Either the entire transaction takes place at once or doesn’t happen at all. There is no midway i.e. transactions do not occur partially. Each transaction is considered as one unit and either runs to completion or is not executed at all.

Consistency: This property ensures that the integrity constraints must be maintained so that the database is consistent before and after the transaction. It refers to the correctness of a database.

Isolation: This property ensures that multiple transactions can occur concurrently without leading to the inconsistency of the database state. Transactions occur independently without interference. Changes occurring in a particular transaction will not be visible to any other transaction until that particular change in that transaction is written to memory or has been committed.

Durability: This property ensures that once the transaction has completed execution, the updates and modifications to the database are stored in and written to disk and they persist even if a system failure occurs.

Overall, ACID properties provide a framework for ensuring data consistency, integrity, and reliability in DBMS.

3. Normalization is a process of organizing data in a database to minimize redundancy and eliminate undesirable characteristics like insertion, update, and deletion anomalies. It involves decomposing relations with redundant data into smaller, simpler, and well-structured relations that satisfy desirable properties. The main reason for normalizing relations is to remove anomalies that can cause data redundancy and integrity problems as the database grows. Normalization consists of a series of guidelines that help in creating a good database structure. The guidelines are called normal forms, and they apply to individual relations. A relation is said to be in a particular normal form if it satisfies constraints

4. Relational databases use Structured Query Language (SQL) to interact with the database. SQL is a comprehensive language for manipulating databases and is used in database systems such as PostgreSQL, MySQL, Microsoft SQL Server, and Oracle Database. SQL can be divided into several sublanguages to help us understand the different operations that can be performed on an SQL database. These sublanguages are:

Data Query Language (DQL): This sublanguage is responsible for reading or querying data from a database. In SQL, this corresponds to the SELECT statement.

Data Manipulation Language (DML): This sublanguage is responsible for adding, editing, or deleting data from a database. In SQL, this corresponds to the INSERT, UPDATE, and DELETE statements.

Data Definition Language (DDL): This sublanguage is responsible for defining the way data is structured in a database. In SQL, this corresponds to manipulating tables through the CREATE TABLE, ALTER TABLE, and DROP TABLE statements.

Data Control Language (DCL): This sublanguage is responsible for the administrative tasks of controlling the database itself, most notably granting and revoking database permissions for users. In SQL, this corresponds to the GRANT, REVOKE, and DENY commands, among others

5. A primary key is a column or a set of columns in a table that uniquely identifies each row in the table. A primary key column must contain unique values and cannot have null values. A table can have only one primary key, which may consist of single or multiple columns.  
A composite key is a combination of two or more columns in a table that can be used to uniquely identify each row in the table when the columns are combined 1. The uniqueness is guaranteed when the columns are taken together, but not individually. The columns that make up a composite key can be of different data types.  
For example, consider a voting table with fields QuestionID, MemberID, and Vote. The combination of QuestionID and MemberID can be used as a composite primary key to uniquely identify each row in the table   
In contrast, consider a customer table with fields CustId, CustName, CustCode, and CustMailId. Here, CustId can be used as a primary key to uniquely identify each row in the table

6. To create a table with a primary key, a column default value, and a column unique constraint in SQL, you can use the following syntax:

CREATE TABLE table\_name (

column1 datatype PRIMARY KEY,

column2 datatype DEFAULT default\_value,

column3 datatype UNIQUE

);

Here, table\_name is the name of the table you want to create. column1 is the name of the first column, which will be used as the primary key. datatype is the data type of the column. column2 is the name of the second column, which will have a default value of default\_value. column3 is the name of the third column, which will have a unique constraint.