

## 1. List down clearly the requirements to be satisfied for each of the first 3 normal forms.

The first three normal forms (1NF, 2NF, and 3NF) are used to ensure data integrity and eliminate data redundancy in a relational database. Here are the requirements that must be satisfied for each of the first three normal forms:

### a) First Normal Form (1NF):

- Atomicity: Each column in a table must hold only atomic (indivisible) values, i.e., values should not be further subdivided.
- No repeating groups: Each column in a table must hold only a single value, and not a group or a list of values.
- Unique column names: Each column in a table must have a unique name.

### b) Second Normal Form (2NF):

- Meets all requirements of 1NF.
- No partial dependencies: All non-key attributes must depend on the entire primary key, and not just part of it.

### c) Third Normal Form (3NF):

- Meets all requirements of 2NF.
- No transitive dependencies: Non-key attributes should not depend on other non-key attributes. In other words, there should be no indirect relationships between non-key attributes through other non-key attributes.

By satisfying the requirements of each of the first three normal forms, we can ensure that the data in a relational database is organized in a way that is free from data anomalies, redundancy, and inconsistencies. This helps to ensure that the data is accurate, consistent, and easy to maintain.

## 2. Compare the differences in database and file based approaches.

A file-based approach and a database approach are two ways of storing and managing data. Here are some key differences between the two approaches:

- **Data Storage:** A file-based approach stores data in individual files, typically in a hierarchical folder structure. A database approach stores data in tables within a database file.
- **Data Management:** In a file-based approach, each file is responsible for its own data management, including access control and data consistency. In a database approach, data management is centralized and handled by the database management system (DBMS).
- **Data Integrity:** In a file-based approach, data integrity is difficult to maintain, as there is no centralized control over data consistency. In a database approach, data integrity is ensured through the use of constraints, such as primary keys, foreign keys, and referential integrity.
- **Scalability:** In a file-based approach, adding new data or making changes to the structure of the data can be difficult and time-consuming, as each file must be updated individually. In a database approach, adding new data or modifying the structure of the data is relatively easy and can be done using SQL commands.
- **Access Control:** In a file-based approach, access control is limited to individual files, and it can be difficult to enforce data security and privacy. In a database approach, access control is centralized, and data security can be enforced through the use of user accounts and permissions.

- **Performance:** A file-based approach can be faster than a database approach for simple data retrieval and storage operations. However, as the volume of data and the number of users increase, a database approach can be more efficient and provide better performance.

Overall, a database approach offers several advantages over a file-based approach, including better data integrity, scalability, and access control. However, a file-based approach may be more suitable for small, simple data storage needs, such as storing text documents or images.

### 3. what is meant by "SQL is a 4GL" ?

SQL (Structured Query Language) is often referred to as a 4GL (Fourth Generation Language). A 4GL is a programming language that is designed to provide a high level of abstraction from the underlying hardware and software systems. It allows programmers to write code at a higher level of abstraction than lower-level languages, such as assembly or machine code.

SQL is considered a 4GL because it provides a high-level declarative language for manipulating relational databases. Programmers can use SQL to query, insert, update, and delete data from a database without having to worry about the underlying implementation details. SQL provides a set of built-in functions and operators that make it easy to perform complex database operations with relatively simple syntax.

In addition to being a 4GL, SQL is also a domain-specific language (DSL) for managing relational databases. It is optimized for managing structured data in tables and provides a range of features that make it easy to work with complex data structures. SQL is widely used for data analysis, reporting, and management in a variety of industries, including finance, healthcare, and e-commerce.

### 4. Define a Functional Dependency and explain their use in DB design.

In database design, a functional dependency is a relationship between two attributes or sets of attributes in a table. It describes how changes in the value of one attribute affect the value of another attribute. Specifically, a functional dependency states that if two rows in a table have the same value for one attribute, they must also have the same value for another attribute.

For example, suppose we have a table called "Employees" with the following attributes: employee\_id, name, department, and salary. In this table, there is a functional dependency between employee\_id and name. This means that if we know the value of employee\_id for a particular employee, we can determine their name. Conversely, if we know the name of a particular employee, we can determine their employee\_id.

Functional dependencies are important in database design because they help ensure data integrity and minimize data redundancy. By identifying and documenting functional dependencies, designers can create efficient and effective database structures that reduce data duplication and improve data consistency. In addition, functional dependencies can be used to create normalization rules, which are guidelines for organizing data in a way that minimizes data redundancy and maximizes data integrity. By following these rules, designers can create databases that are easier to maintain, modify, and extend over time. Overall, functional dependencies play a crucial role in ensuring the accuracy and reliability of database systems.

5. Discuss the advantages of using the database approach as compared to the traditional file processing approach.

The database approach offers several advantages over the traditional file processing approach:

- **Improved Data Integrity:** In a database approach, data is stored in a centralized location and is managed by a database management system (DBMS), which enforces data integrity rules such as referential integrity and data constraints. This ensures that data is accurate and consistent, which is difficult to achieve in a file-based approach.
- **Better Data Consistency:** A database approach provides a mechanism for ensuring that data is consistent across all instances of the data. Any changes made to the data are immediately reflected in all views of the data, reducing the risk of data inconsistencies and ensuring that everyone has access to the same version of the data.
- **Improved Data Security:** A database approach provides a centralized mechanism for controlling access to data, which is difficult to achieve in a file-based approach. Access to the database can be controlled at the user level, ensuring that only authorized users can access sensitive data.
- **Increased Scalability:** In a database approach, the data can be easily expanded or modified as needed, whereas in a file-based approach, expanding or modifying the data requires changes to be made to every file containing the data.
- **Improved Data Retrieval:** A database approach provides a more efficient and powerful mechanism for retrieving data than a file-based approach. Queries can be written in SQL, which is a powerful query language that can retrieve data from multiple tables at once.
- **Improved Data Integration:** A database approach allows for the integration of data from different sources and the sharing of data across multiple applications. This is difficult to achieve in a file-based approach, where data is typically stored in multiple formats and locations.

Overall, the database approach offers significant advantages over the traditional file processing approach, including improved data integrity, consistency, security, scalability, retrieval, and integration. These advantages have made the database approach the preferred approach for managing data in modern organizations.