

A. Database fundamentals

- Database is a collection of individual data items stored in a highly structured way that models reality. This data can be stored in a single location or distributed across a large network. A database engine performs common tasks on the data, known as
- CRUD, which stands for Create, Read, Update, and Delete

B. Database Application

- Database Management System (DBMS), such as Microsoft Access, SQL Server, or Oracle, is a program used to define, record, query, and manage the data in a database. The
- Structured Query Language (SQL) is a language used by the DBMS to interact with the database, allowing you to create tables, insert/update data, and retrieve information

C. Relational Structure

- In a relational database, data is stored in tables consisting of rows and columns.
 - Columns are also called attributes, values, or fields, and they store a single piece of information.
 - Rows are also called records, and each one is made up of a series of values that represent a single identity.
- A well-designed database table is more structured than a typical spreadsheet. The rules of relational structures state that each cell should contain a single value, columns should store a single type of information, and column names and rows must be unique.

D. Database Development Lifecycle & Design Principles

Database Development Lifecycle is a set of steps for creating a database solution. Skipping these steps can lead to "undesirable results and wasted effort".

To prevent data anomalies and ensure a robust database, careful design is essential. Key principles include:

- **Removing Duplicate Information:** Duplicates can lead to slow performance, maintenance issues, and inconsistencies. You can fix this by moving repeated data into a separate table.
- **Eliminating Inconsistent Data:** This involves correcting typographic errors or different entry styles. The solution is to create a new table for the redundant information with a clear name.
- **Breaking Data Down into Components:** Fields should be specific and descriptive to make the database flexible. For example, a single "Employee Name" field should be broken into "First Name" and "Last Name".
- **Preventing Data Conflicts:** This occurs when a value is calculated from other data. If one of the original values changes, the calculated value must also be updated.
- **Requiring Complete Information:** You should determine which fields are required during data entry to prevent users from leaving them blank and causing incorrect or missing data.
- **Maintaining a Consistent Structure:** Tables should be "open for extension and closed for modification" to handle multi-value fields. Instead of adding new columns like "Child 1," "Child 2," etc., you should create a separate table for the repeated information, such as a "Children" table linked to the "Employees" table.