DATA MODELS

Lana Anvar

April 2023

Introduction

DATA MODELS

A data model is an abstract model that organizes elements of data and standardizes how they relate to one another and to the properties of real-world entities.

Relational Data Model

The relational model is a type of data model used to organize data in a tabular format. It consists of tables that are related to each other through common attributes. This model was initially described by Edgar F. Codd, in 1969. The relational data model is the widely used model which is primarily used by commercial data processing applications and in database management systems (DBMS) for handling large amounts of structured data.

For example, a bank might use a relational model to store customer information, with each customer being a row in a table, and each attribute (e.g. name, address, account number) being a column. Suppose we have a bank with three tables: customers, accounts, and transactions. The customers table would contain information about each customer, such as their name, address, and contact information. The accounts table would contain information about each account, such as the account number, account type, and balance. The transactions table would contain information about each transaction, such as the transaction date, amount, and the account involved.

Graph Data Model

The graph data model is a type of data model that represents data as a network of nodes and edges. This model is useful for handling complex data relationships, such as social networks or transportation systems. In a graph data model, nodes represent entities (e.g. people, places, things), and edges represent the relationships between them (e.g. friendships, routes). A graph data model is different from a relational model. A graph focuses on the relationships between information, whereas a relational model focuses on storing similar information in a list.

For example, Facebook uses a graph data model to represent its social network, with each user being a node and each friendship being an edge. Each node would have attributes such as the user's name, profile picture, and other personal information. Each edge would have attributes such as the type of connection (e.g. friend, family, colleague) and the date it was established.

Hierarchical Data Model

The hierarchical data model is a type of data model that organizes data in a tree-like structure. In the hierarchical model, segments pointed to by the logical association are called the child segment and the other segment is called the parent segment. If there is a segment without a parent is then that will be called the root and the segment which has no children are called the leaves. The main disadvantage of the hierarchical model is that it can have one-to-one and one-to-many relationships between the nodes. This model is commonly used in file systems and document management systems.

Taking the example of a company, the top-level node in the hierarchical data model would represent the company as a whole, and the next level of nodes might represent different departments, such as Sales, Marketing, and Finance. Each of these department nodes would have sub-nodes representing teams within the department, such as Sales Team 1, Sales Team 2, and so on. Each team node would have sub-nodes representing individual employees within the team.

INTERACTION MODELS

An interaction model is a design model that binds an application together in a way that supports the conceptual models of its target users. Interaction models are models used in software engineering and systems design to represent the interactions between different components of a system, or between the system and its users. These models are used to describe the behavior of a system and how its components interact with one another, and to help identify potential issues or areas for improvement in the system design. The different interaction models are:
1) Activity Diagram
2) Sequence Diagram
3) Data Flow Diagram

Activity Diagram

An activity diagram is a visual representation of a process or work-flow, showing the steps involved and the flow of control between them. Each step is represented by a node, and the flow of control between steps is represented by edges. Activity diagrams are commonly used in software development to model business processes or system workflows.

For example, an activity diagram might be used to model the process of ordering a product online. The diagram might include nodes representing steps such as "Browse Products", "Add to Cart", "Enter Payment Information", and "Place Order", with edges indicating the flow of control between steps.

Sequence Diagram

Sequence diagrams, commonly used by developers, model the interactions between objects in a single use case. They illustrate how the different parts of a system interact with each other to carry out a function, and the order in which the interactions occur when a particular use case is executed.
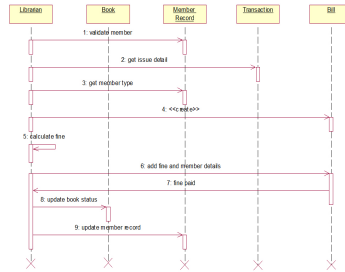
Figure 1: Library Management System

For example, a sequence diagram might be used to model the process of logging into a website. The diagram might include boxes representing the user, the web server, and the authentication system, with arrows indicating the messages exchanged between them, such as "User sends login credentials to server", "Server forwards credentials to authentication system", and "Authentication system sends approval or denial to server".

The sequence diagram below shows how the objects in the online library management system interact with each other to perform the function 'Borrowing book from a library'.