***SDA-Questions***

**Q: What is structure of your software?**

**Ans:** A software system's structure is a division of that system into a set of parts and the relations between those parts. Confusion often arises because of the many types of parts and the many important relations between them.

Software systems are built from units provided by the language. Some languages have functions and data; others have classes and objects. Some group these into modules. Some allow all of these types, trying to be everything to everyone. Some have none of these.

Programming is combining the language's available units, both within a type of unit (functions call other functions) and between types of units (functions are grouped into modules). This article explores several of these structural relationships, each of which is structured as a graph. Experienced programmers are necessarily good at thinking about graphs, though they may not use the technical term "graph". (Note: no experience with graph theory is necessary here; we won't do any formal mathematical analysis and the relevant ideas will be introduced from scratch.)

**Q: How crud operations are performed using MVC and rest?**

**Ans:**

Introduction

CRUD operation in MVC is the basic operations, where CRUD denotes create, read, update, and delete. But before understanding the basic operations of MVC, first, learn about MVC. MVC is the Model View Controller. MVC is a design pattern that is used to differentiate the data from business logic and presentation logic. It gives a pattern that helps in designing the web application.

1. Model Layer: MVC has a model component that deals with logic-related data. The model layer represents the information transferred between view and controller layers of data related to the business logic. For example, employee objects help fetch the employee information from the relevant table in the database, manipulate the data, and then update it back into the database.

2. View Layer: The view layer has the view components that deal with the User interface logic. As an illustration, an employee’s view components comprise the components, such as text boxes, radio buttons, drop-downs, check-boxes, etc. The view layer has the components that the end-user deals with.

3. Controller Layer: Controller is the interface between view layer components and model layer components. The controller controls the business logic. It receives the user input through the view layer and processes the information through the model layer.

CRUD and REST are two of the most popular concepts in the Application Program Interface (API) industry. REST was made to standardize the HTTP protocol interface between clients and servers and is one of the widely used design styles for web API. On the other hand, CRUD is an acronym used to refer to the four basic operations executed on database applications. Because both work on manipulating databases’ data, it’s easy to see why people have some confusion between them. This blog will discuss what REST and CRUD are, the basic principles that govern them, and their similarities and differences.

REST is an abbreviation for Representational State Transfer. It is a software architectural style that provides standards for computers on the web, dictating how the systems interact.[Roy Fielding](https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm), the founder of the REST protocol, defines it as “an abstraction of the architectural elements within a distributed hypermedia system.”

**Q: How request dispatcher is created in rest?**