Database

04 February 2022 15:30

.NET Data Provider: It is an add-on (nuget) package which enables a .NET application to consume relational data managed by a particular data-base system using SQL. It includes support for following objects

1. **Connection** - It opens a communication session with the database system using the information provided in a (connection) string.
2. **Command** - It executes an SQL statement on a database system using the Connection object.
3. **DataReader** - It fetches rows resulting from execution of a query statement using the Command object.

Entity Framework Core: It is an add-on package which enables a .NET application to consume relational data managed by a particular database system using LINQ queryable sets of uniquely identifiable *plain old CLR object* (POCO) type instances known as *entities*.
Under entity framework the *conceptual model* of application data is as specified by entities is associated with the *storage model* of that data using

1. **Mapping Conventions** - The Entity class is mapped to the database table whose name matches with the name of *DbSet<Entity>* type property in the *DbContext* derived class and its each property is mapped to the column with matching name in that table. The column mapped to *Id* or *EntityId* property is constrained to be the *primary-key* and the column mapped to *ParentEntityId* property is constrained to be the *foreign-key*.
2. **Data Annotations** - The conventional mapping is overridden by applying database schema specifying attributes to the entity class and its properties.
3. **Fluent API**: The conventional mapping is overridden by passing the meta-data of the entity class and its properties through along with database schema specifiers through a chain of methods exposed by model builder objects.

gRPC: It is a *light-weight*, *high-performance* generic *remote-procedure call* framework for implementing *cross-platform*, *service-oriented*

distributed applications using different high-level programming languages. It includes support for

1. **Protobuf** - It is a standard specified by Google consisting of an *interface definition language* known as Proto for describing service operations along with the structures of messages exchanged by those operations and a *binary buffer format* for serializing and deserializing those messages.
2. **Channel** - It allows the client which is a consumer of service operations to *connect* to the server which publishes those operation while enabling each of them to *stream* a protobuf message or a sequence of such messages to other using *HTTP/2*.