1. List vs IEnumerable

Ans: IEnumerable describes behavior, while List is an implementation of that behavior. When you use IEnumerable, you give the compiler a chance to defer work until later, possibly optimizing along the way. IEnumerable is read-only and List is not. Many types other than List<T> implement IEnumerable such as an ArrayList. So one advantage is you can pass different collection types to the same function.

1. IEnumerable vs IQueryable

Ans: IQueryable queries out-of-memory data stores, while IEnumerable queries in-memory data. Moreover, IQueryable is part of .NET's System.LINQ namespace, while IEnumerable is in System.Collections namespace. IQueryable<T> extends the IEnumerable<T> so anything you can do with a “plain” IEnumerable<T>, you can also do with IQueryable<T>

1. Database First Approach vs Code First Approach

Ans: In Code First Approach we will first create entity classes with properties defined in it. Entity framework will create the database and tables based on the entity classes defined. So database is generated from the code. When the dot net code is run database will get created. In Database First Approach Database and tables are created first. Then you create entity Data Model using the created database.

Use CFA when the application is small and use DFA when the application is large. Data mapping and key creation are easier when using DFA because you don’t need to write any code.