**Course Name: .Net 6 Core Entity Framework: How-To Guide For Professionals**

*NOTE: Mark the correct answers with Yellow highlight*

**Chapter Number 5 – IQueryable vs. IEnumerable**

1. What is the fundamental difference between IEnumerable and IQuerable Interfaces?

a. IEnumerable is used to access collections of data, whereas IQueryable is used to access single instances.

Reason – Incorrect; both are used to return collections of data from the back-end repository.

b. IEnumerable queries in-memory data (data returned from the database), while IQueryable queries are applied at source (in the database itself).

Reason – Correct; both use deferred execution however IQueryable requires a subsequent cast operation to invoke the query in the database and load the resultant dataset whereas IEnumerable returns all the data and then applies the where condition to the data in-memory.

c. IEnumerable enables retrieval of collections of parent data, whereas IQueryable is used to retrieve parent and child data.

Reason – Incorrect; the Include() method is used on both interfaces to identify child collection to include in the returned result set..

d. IQueryable is used to perform searches on data where IEnumerable requires keys to be provided.

Reason – Incorrect; both interfaces are used to search for data, that may or may not include key search parameters.

2. How do you cause data query to be performed in the database server and not in the application layer (for better performance)?

a. Request for data from the DbContext using IQueryable() as the returned result, and then cast those results to a list or enumerable type.

Reason – Correct; IQueryable executes and applies where conditions in the database server when data requested is cast to result collection via operation such as ToList(), ToListAsync(), AsEnumerable() etc. and then returned to the client application.

b. Use the ExecuteInDb() extension method on queries created using either IEnumerable or IQueryable interfaces.

Reason – Incorrect; using IQueryable is the only interface that executes the select query on the server side with all filters (where clause parameters).

c. Request for data from the DbContext using IEnumerable as the return result, and then iterate through the collection using the for..each mechanism.

Reason – Incorrect; IEnumerable perform filtering of the data once it is return from the database (re: in memory).

d. use the OnServer() override on the DbContext derived data service.

Reason – Incorrect; the DbContext does not have such a method; the usage of IQueryable or IEnumerable alone determines where the queries are executed.

3. When you have and IEnumerable supported list that contains multiple types of objects, how do you retrieve only the items that are strings from that list?

a.Trick question, IEnumerable collection can only contain one type of object at a time, and will throw an exception when this is violated.

Reason – Incorrect; IEnumberable can be a collection of any (non-generic) object type.

b. using For..each mechanism, where the type being iterated is declared as the desired type

Reason – Incorrect; the for…each mechanism would throw an exception when it encountered a type mismatch.

c. using the method Cast<int>() on the IEnumerable based type.

Reason – Incorrect, this will throw an exception when the object assigned the results of this operations is iterated (via for..each mechanism).

d. using the method OfType<string>() on the IEnumerable based list.

Reason – Correct; this method returns only the contained elements of a specific type.

4.Why is IQueryable more performant than IEnumerable?

a. IQueryable used deferred execution and which enables the query to be performed when the content is first accessed.

Reason – Incorrect; both IQueryable and IEnumerable use deferred exection.

b. IQueryable performs the where clause (data filtering) on the data server, whereas IEnumerable returns the entire dataset which is then filtered in the client application.

Reason – Correct; IEnumerable execute a select query on the server side, load data in-memory on a client-side and then filter data. Querying data from a database, IQueryable execute the select query on the server side with all filters.

c. IQueryable has been specially tuned to used indexes, whereas IEnumerable does not.

Reason – Incorrect; this is not true at all.

d. IQueryable is multi-threaded whereas IEnumberable is single threaded.

Reason Incorrect; there is no such thread distinction between the two interfaces.

5. Which interface enables paged result sets and what methods are used to achieve it?

a. both IQueryable and IEnumerable can return paged results, using the Skip() and Take() methods to select a page of data from the repository.

Reason – Incorrect; Skip() and Take() are only available on the IQueryable interfaces.

b. IEnumerable only, using methods Skip() and TaKe() to put a page of data from the repository.

Reason – Incorrect; Skip() and Take() are only available on the IQueryable interfaces.

c. IQueryable only, using methods Skip() and TaKe() to put a page of data from the repository.

Reason – Correct; IQueryable exposes both methods, and used together they can select a specific number of rows, starting at a specific row index, from an IQueryable result set.

d. nether; you must cast the return results to a List based collection first, and then apply the Skip() and Take() methods to access a specific page of search data.

Reason – Incorrect; there are no such methods on List based collections.