Tanya Wang

1. **What is a Garbage collector? What are the different generations in the Garbage collector?**
   A feature of .NET runtime that automatically manages memory for objects in a program. It frees up memory space for objects that are no longer needed by the program to avoid memory leaks, which can then be used for other objects.

   Generation 0: This is for short-lived objects. The garbage collector checks this generation more frequently.
   Generation 1: For medium-lived objects that survived the first generation collection.
   Generation 2: This generation contains long-lived objects, such as those that last for the application's lifetime.

   The generational strategy optimizes memory management by focusing more on short-lived objects and less frequently on long-lived objects.

2. **What are Access Modifiers and how to use them?**
   Access modifiers control the visibility and accessibility of classes, methods, and other members**.**
   Public: Accessible anywhere.
   Private: Accessible only within the same class.
   Protected: Accessible within the same class and derived classes.
   Internal: Accessible within the same assembly.
   Protected Internal: Accessible within the same assembly or from derived classes.

3. **What are extension methods and give me examples of built in extension methods. Have you created an extension method in your project? If yes, how with examples?**
   Extension methods are static methods that allow you to "add" new methods to existing types (even if you don't have access to modify the original type) without modifying or inheriting from the type. They enable adding functionality to existing types in a clean, reusable way.

   An extension method is defined in a static class and the first parameter is preceded by the this keyword, indicating which type it is extending.

   Used String Extension Methods which is build in extension in the project:
   string str = "";
   bool isNullOrEmpty = string.IsNullOrEmpty(str);

   .IsNullOrEmpty(): Checks if a string is null or an empty string.
   .IsNullOrWhiteSpace(): Checks if a string is null, empty, or consists only of whitespace characters.

4. **What are lambda expressions, where have u used lambda expressions？**
   Lambda expressions are anonymous functions used to create delegates or expression tree types. They are typically used in LINQ queries.

5. **Write LINQ, GroupBy, Join, and Where conditions.**

```
var result = products.Where(p => p.Price > 50);
var grouped = products.GroupBy(p => p.Category);
var result = from p in products
             join c in categories on p.CategoryId equals c.Id
             select new { p.Name, c.Name };
```

6. **What does sealed keyword does in C# ?**
   The sealed keyword in C# prevents a class from being inherited
7. **What are ref vs out keywords?**
   ref: The variable must be initialized before it is passed.
   out: The variable doesn't need to be initialized before passing, but it must be initialized within the method.

8. **Can you explain about Normalization, especially 3NF and how it is helpful? Tell me scenarios where Denormalization is useful?**
   Normalization is the process of organizing data in a database. It includes creating tables and establishing relationships between those tables according to rules designed both to protect the data and to make the database more flexible by eliminating redundancy and inconsistent dependency.
   First normal form
   - Eliminate repeating groups in individual tables.
   - Create a separate table for each set of related data.
   - Identify each set of related data with a primary key.

   Second normal form
   - Create separate tables for sets of values that apply to multiple records.
   - Relate these tables with a foreign key.

   Third normal form
   - Eliminate fields that don't depend on the key.

   Denormalization: It's useful for performance optimization when reading data is more frequent than writing, such as in data warehouses.

9. **When do you use stored procedures and when do you use function? Differences between them?**
   Stored Procedures: Perform operations like inserts, updates, and deletions. They can return multiple result sets.
   Functions: Return a single value or table. They cannot perform insert, update, delete operations.

10. **Explain Primary Key, Foreign Key and Unique Key with differences?**
Primary Key: Uniquely identifies each record and cannot be null.
Foreign Key: Establishes a relationship between two tables.
Unique Key: Ensures all values in a column are unique but can have one null value.

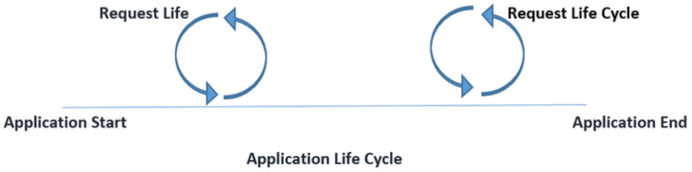
11. **What is the difference Between Union and Union all?**
Union: Removes duplicate rows.
Union All: Includes duplicates.

12. **Explain all the joins?**
Inner Join: Returns records with matching values in both tables.
Left Join: Returns all records from the left table and matched records from the right table.
Right Join: Opposite of left join.
Full Join: Returns all records when there's a match in either table.

13. **What are AddSingleton, AddScoped, AddTransient? What’s the difference?**
AddSingleton: A single instance is created and shared across the application.
AddScoped: A new instance is created per request.
AddTransient: A new instance is created each time the service is requested.

14. **Explain Dependency Injection**
It is a design pattern where objects are provided their dependencies rather than creating them internally.

15. **Explain MVC's application life cycle.**



MVC actually defined in two life cycles, the application life cycle, and the request life cycle.
The application life cycle, in which the application process starts the running server until the time it stops. and it tagged the two events in the startup file of your application.

16. **What will EF return IQueryable or IEnumerable?**
IQueryable<T>: When querying the database (e.g., using DbSet<T>). It represents a query that can be further composed and deferred until executed. This allows for better performance since SQL queries are executed on the database side.
IEnumerable<T>: When the query is executed and the data is loaded into memory. Once you use a method like ToList(), AsEnumerable(), or perform enumeration, the data is materialized and turned into IEnumerable.

17. **How would you see the SQL queries generated by Entity Framework? What tools or coding would you implement?**

18. **Explain the steps of code first approach with migrations**
    Define the data models (POCO classes) first, and EF will generate the corresponding database schema.
    - **Create Model Classes:** Define the entities (POCOs) that represent tables in the database.
    - **Define the DbContext**: Define the context to manage database access.
    - **Enable Migrations**: Open the Package Manager Console and run.
    - **Create an Initial Migration**: After modifying your models, create a migration
    - **Apply the Migration**: Apply the migration to the database
    - **Subsequent Migrations**: For future changes to the model, repeat steps”Create Initial” and “apply migration”.

19. **Why do you think we need an abstract class, why do you need an interface? Give few examples, where you have used abstract classes and interfaces in your project**
    Abstract Class:
    - Used when you want to provide some shared implementation across derived classes, but still need to enforce certain behaviors that must be implemented by derived classes.
    - It can have both abstract (without implementation) and non-abstract (with implementation) methods.
    Interface:
    - Used when you want to define a contract that multiple classes must implement without providing any implementation.
    - Promotes loose coupling and allows for multiple inheritance.

    I used abstract classes to share common behavior among classes that represent related entities (like animals or vehicles) and interfaces to enforce a contract across unrelated classes.

20. **What does static modifier do in C# ?**
    The static modifier in C# indicates that a member belongs to the type itself rather than to any specific instance of the class.
    Static classes: Cannot be instantiated and can only contain static members.
    Static methods and properties: Can be accessed without creating an object of the class.

21. **Write a query to create a view named that displays the total number for each product and their names and categories.**
    CREATE VIEW ProductSummary AS

```
SELECT p.ProductName, c.CategoryName, COUNT(*) AS Total
FROM Products p
JOIN Categories c ON p.CategoryId = c.CategoryId
GROUP BY p.ProductName, c.CategoryName;
```

22. **Add a check constraint to the quantity_sold column in the Sales table to ensure that the quantity sold is always greater than zero.**

```
ALTER TABLE Sales
ADD CONSTRAINT CHK_QuantitySold CHECK (quantity_sold > 0);
```