**Chapter 12: How to use EF Core**

**Murach's ASP.NET Core MVC**

**MULTIPLE CHOICE**

1. Writing entity and database context classes before generating a new database is called

| a. | Database First development |
| --- | --- |
| b. | Initial Class development |
| c. | Code First development |
| d. | Entity Relationship development |

2. The \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ represent various components that a database represents, and the \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ allows communication with the database.

| a. | component properties - database property |
| --- | --- |
| b. | entity classes - database context class |
| c. | facade classes - dataset classes |
| d. | context classes - dataset component |

3. You can configure the creation of database tables in each of the following ways *except*

| a. | configuration by convention |
| --- | --- |
| b. | data attributes |
| c. | JSON plugins |
| d. | Fluent API |

4. Which of the following commands would you use in the Package Manager Console if you want to revert a database to a previous migration?

| a. | **Update-Database** |
| --- | --- |
| b. | **Add-Migration** |
| c. | **Drop-Database** |
| d. | **Scaffold-DbContext** |

5. If one customer can have many invoices, but each invoice can belong to only one customer, what type of relation exists between customers and invoices?

| a. | one-to-one |
| --- | --- |
| b. | one-to-many |
| c. | many-to-many |
| d. | primary-to-foreign |

6. If a table named Patient has a one-to-many relationsip with a table named Appointment, the Appointment table typically includes a \_\_\_\_\_\_\_\_\_\_\_\_\_ that relates it to the Patient table.

| a. | tuple identifier |
| --- | --- |
| b. | sub-primary key |
| c. | foreign key |
| d. | composite key |

7. To establish a many-to-many relationship between two entities, you typically use a

| a. | linking entity |
| --- | --- |
| b. | foreign entity |
| c. | composite entity |
| d. | context entity |

8. You can use the \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ method found in the Fluent API to help control how child rows are handled if a parent row is deleted.

| a. | OnFluent() |
| --- | --- |
| b. | OnRestrict() |
| c. | OnDelete() |
| d. | OnCascade() |

9. To avoid losing modifications of entity classes that are generated from a database, it’s common to use

| a. | code segmentation |
| --- | --- |
| b. | direct injection |
| c. | partial classes |
| d. | entity cloning |

10. When using LINQ to retrieve multiple rows from a database, the data that’s returned can be stored in an object with a data type of

| a. | IRunnable<T> |
| --- | --- |
| b. | IQueryable<T> |
| c. | IDataRetrievable<T> |
| d. | IEnumerable<T> |

11. To commit insert, update, and delete operations on a DbSet<T> object to the database, what method of the DbContext class must you call?

| a. | SaveChanges() |
| --- | --- |
| b. | Update() |
| c. | Commit() |
| d. | Disconnect() |

12. The concurrency option in EF Core that throws an exception whenever a row has been changed since its last retrieval is called

| a. | last in wins |
| --- | --- |
| b. | optimistic concurrency |
| c. | pessimistic concurrency |
| d. | refusal concurrency |

13. To begin encapsulating code for data access, it’s often helpful to begin by adding extention methods to the interface named

| a. | IRunnable<T> |
| --- | --- |
| b. | ILINQable<T> |
| c. | IEnumerable<T> |
| d. | IQueryable<T> |

14. Providing basic CRUD operations for each class and further abstracting their details with use of an interface is a common design pattern named the

| a. | requisition pattern |
| --- | --- |
| b. | observer pattern |
| c. | repository pattern |
| d. | chain of responsibility pattern |

15. The unit of work pattern provides a central class that allows several \_\_\_\_\_\_\_\_\_\_\_\_\_ objects to share a DB context.

| a. | requisition |
| --- | --- |
| b. | observer |
| c. | repository |
| d. | chain of responsibility |