

## Module Review and Takeaways

The heart of an MVC web application is the model. This is because the model classes describe the information and objects that your web application manages. In this module, you have seen how to create your model, define relationships between the model classes, describe how to display, edit, and validate properties, and how to extend MVC model handling capabilities. You have also seen how to bind model classes to database tables by using Entity Framework and how to query object in the model by writing LINQ code.



**Best Practice:** If you have a pre-existing database for a web application, use Entity Framework in the database-first workflow to import and create your model and its classes.



**Best Practice:** If you want to create a new database for a web application and prefer to draw your model in a Visual Studio designer, use Entity Framework in the model-first workflow to create your model and its classes.



**Best Practice:** If you want to create a new database for a web application and prefer to write code that describes your model, use Entity Framework in the code-first workflow to create your model and its classes.



**Best Practice:** If you want to separate business logic from data access logic, create separate model classes and repository classes.

### Common Issues and Troubleshooting Tips

Common Issue	Troubleshooting Tip
The website cannot connect to or create a database.	

### Review Question(s)

**Question:** At the end of the first iteration of your project, you have a website that displays photos that users upload. However, during development, the database is empty and users must upload several photos to the site so they can test the functionality. Your manager wants you find some way to populate the database whenever it is deployed to the test server. How can you do this?