En una aplicacion Web ASP.NET MVC 5 se pueden definir mas de una BD para ello hay que tener tantas clases DbConext como bases de datos se quiera tener en el sistema.

Al igual se tienen que definir en el fichero Web.config todos los connections strings de casa una de las bases de datos.

En web.config:

<add name="DefaultConnection" connectionString="Data Source=(LocalDb)\v11.0;AttachDbFilename=|DataDirectory|\user-roles.mdf;Initial Catalog=users;Integrated Security=True" providerName="System.Data.SqlClient" />

<add name="SportsStoreDb" connectionString="Data Source=(LocalDb)\v11.0;AttachDbFilename=|DataDirectory|\SportsStoreDb.mdf;Initial Catalog=aspnet-Primer-20160215114130;Integrated Security=True" providerName="System.Data.SqlClient" />

Notar que la propiedad name es distinta en cada definicion al igual que la definicion para Catalogo dentro de la propiedad connectionString.

Definicion de la Clase ProductDbContext para la BD "SportsStoreDb"

public class ProductDbContext:DbContext

{

public ProductDbContext()

: base("SportsStoreDb")

{

Database.SetInitializer<ProductDbContext>(new ProductDbInitializer());

}

public DbSet<Product> Products { get; set; }

public DbSet<Order> Orders { get; set; }

public DbSet<OrderLine> OrderLines { get; set; }

}

public class ProductDbInitializer : DropCreateDatabaseIfModelChanges<ProductDbContext>

{

protected override void Seed(ProductDbContext context)

{

new List<Product> {

new Product() { Name = "Kayak", Description = "A boat for one person",

Category = "Watersports", Price = 275m },

new Product() { Name = "Lifejacket",

Description = "Protective and fashionable",

Category = "Watersports", Price = 48.95m },

new Product() { Name = "Soccer Ball",

Description = "FIFA-approved size and weight",

Category = "Soccer", Price = 19.50m },

new Product() {

Name = "Corner Flags",

Description = "Give your playing field a professional touch",

Category = "Soccer", Price = 34.95m },

new Product() { Name = "Stadium",

Description = "Flat-packed 35,000-seat stadium",

Category = "Soccer", Price = 79500m },

new Product() { Name = "Thinking Cap",

Description = "Improve your brain efficiency by 75%",

Category = "Chess", Price = 16m },

new Product() { Name = "Unsteady Chair",

Description = "Secretly give your opponent a disadvantage",

Category = "Chess", Price = 29.95m },

new Product() { Name = "Human Chess Board",

Description = "A fun game for the family",

Category = "Chess", Price = 75m },

new Product() { Name = "Bling-Bling King",

Description = "Gold-plated, diamond-studded King",

Category = "Chess", Price = 1200m },

}.ForEach(product => context.Products.Add(product));

context.SaveChanges();

new List<Order>{

new Order() { Customer = "Alice Smith", TotalCost = 68.45m,

Lines = new List<OrderLine> {

new OrderLine() { ProductId = 2, Count = 2},

new OrderLine() { ProductId = 3, Count = 1},

}},

new Order() { Customer = "Peter Jones", TotalCost = 79791m,

Lines = new List<OrderLine> {

new OrderLine() { ProductId = 5, Count = 1},

new OrderLine() { ProductId = 6, Count = 3},

new OrderLine() { ProductId = 1, Count = 3},

}

}

}.ForEach(order => context.Orders.Add(order));

context.SaveChanges();

}

}

Definicion de la Clase ApplicationDbContext para la BD " DefaultConnection "

public class ApplicationDbContext : IdentityDbContext<ApplicationUser>

{

public ApplicationDbContext()

: base("DefaultConnection", throwIfV1Schema: false)

{

}

public static ApplicationDbContext Create()

{

return new ApplicationDbContext();

}

}