Program.cs

```
namespace JakubSzaredkoEFProducts
{
    class Program
    {
        public static void Main(string[] args)
        {
            Console.WriteLine("Enter a new product name");
            string productName = Console.ReadLine();

            ProductContext productContext = new ProductContext();
            Product product = new Product { ProductName = productName };
            productContext.Products.Add(product);
            productContext.SaveChanges();

            Console.WriteLine("\nList of all products stored in the database:");

            IQueryable<string> query = from prod in productContext.Products select
prod.ProductName;
            foreach (string pName in query)
            {
                Console.WriteLine(pName);
            }
        }
    }
}
```

Product.cs

```
namespace JakubSzaredkoEFProducts
{
    internal class Product
    {
        public int ProductID { get; set; }
        public string ProductName { get; set; }
        public int  UnitsOnStock { get; set; }
    }
}
```

ProductContext.cs

```
namespace JakubSzaredkoEFProducts
{
    internal class ProductContext : DbContext
    {
        public DbSet<Product> Products { get; set; }

        protected override void OnConfiguring(DbContextOptionsBuilder optionsBuilder)
        {
            base.OnConfiguring(optionsBuilder);
            optionsBuilder.UseSqlite("Datasource=ProductsDatabase");
        }
    }
}
```

ProductsDatabase

```
sqlite > .tables
Products __EFMigrationsHistory

sqlite> .schema Products
CREATE TABLE IF NOT EXISTS "Products" (
    "ProductID" INTEGER NOT NULL CONSTRAINT "PK_Products" PRIMARY KEY AUTOINCREMENT,
    "ProductName" TEXT NOT NULL,
    "UnitsOnStock" INTEGER NOT NULL
);

sqlite > PRAGMA table_info(Products);
0 | ProductID | INTEGER | 1 || 1
1 | ProductName | TEXT | 1 || 0
2 | UnitsOnStock | INTEGER | 1 || 0
```

Uruchomienie końcowego programu