

```

var NewCustomer = new customer {
    CustomerID = "DLEAP",
    CompanyName = "DevLeap",
    Country = "Italy" };
db.Customers.Add( newCustomer );

var oldCustomer = db.Customers.Single( c => c.CustomerID == "FRANS" );
db.Customer.Remove( oldCustomer );

```

كما هو واضح قمنا بعمل نسخة أو سجل من الكائن `customer` و اسمناه `newCustomer` و اعطيناه القيم المناسبة. ثم بعد ذلك استخدمنا دالة `Add` لإضافة السجل فى الجدول. ثم قمنا بالبحث عن السجل الذى قيمة `CustomerID` فيه تساوى `FRANS`. ثم باستخدام دالة `Remove` قمنا بمسحه من الجدول... لا يوجد أسهل من ذلك... انظر الى كود SQL الذى يتم توليده خلف الكواليس:

```

INSERT INTO [Customers](CustomerID, CompanyName, ...)
VALUES ("DEVLEAP", "DevLeap", ...)

```

```

UPDATE [Orders]
SET [CustomerID] = NULL
FROM [Orders]
WHERE ([OrderID] = @p1) AND ...

```

```

DELETE FROM [Customers] WHERE [CustomerID] = "FRANS"

```

أعتقد انك تعرف كود SQL جيداً و تدرك معانيه... اى انك لا تحتاج لشرح هذا الكود... ولو افترضنا فرضاً خيالياً انك لا تدرك معانيه, فلا تدع هذا يصيبك بالقلق, لقد وضعنا الكود هنا لعرضه فقط.

بخصوص حذف السجلات من الجداول يمكن استخدام `Remove` او `RemoveAll` فى المثال الاسبق استعملنا `Remove` و فى المثال التالى سوف نستعمل `RemoveAll`.

```

var order = db.Orders.Single( o => o.OrderID == 10248 );
db.Order_Details.RemoveAll( order.Order_Details );
db.Order.Remove( Order );
db.SubmitChanges();

```

تحديث قواعد البيانات

عند استخدام LINQ To SQL, فإن العديد من جمل SQL يتم توليدها و ارسالها الى قاعدة البيانات بطريقة غير مرئية بالنسبة للمستخدم. على الجانب الأخر فإن اوامر SQL التى تقوم بتعديل حالة البيانات فى القاعدة ترسل الى القاعدة عنا تقرر انت ذلك عن طريق استخدام دالة `SubmitChanges` الموجودة فى `DataContext`. كما هو موضح فى الكود التالى :

```

Northwind db = new Northwind( Program.ConnictionString );
var customer = db.Customers.Single( c => c.CustomerID == "FRANS";
customer.ContactName = "Marco Russo";
db.SubmitChanges();

```