

How to: Delete Rows From the Database

03/30/2017 3 minutes to read Contributors  all

In this article

- [Example](#)
- [Example](#)
- [See Also](#)

You can delete rows in a database by removing the corresponding LINQ to SQL objects from their table-related collection. LINQ to SQL translates your changes to the appropriate SQL `DELETE` commands.

LINQ to SQL does not support or recognize cascade-delete operations. If you want to delete a row in a table that has constraints against it, you must complete either of the following tasks:

- Set the `ON DELETE CASCADE` rule in the foreign-key constraint in the database.
- Use your own code to first delete the child objects that prevent the parent object from being deleted.

Otherwise, an exception is thrown. See the second code example later in this topic.

Note

You can override LINQ to SQL default methods for `Insert`, `Update`, and `Delete` database operations. For more information, see [Customizing Insert, Update, and Delete Operations](#).

Developers using Visual Studio can use the Object Relational Designer to develop stored procedures for the same purpose.

The following steps assume that a valid `DataContext` connects you to the Northwind database. For more information, see [How to: Connect to a Database](#).

To delete a row in the database

1. Query the database for the row to be deleted.
2. Call the `DeleteOnSubmit` method.
3. Submit the change to the database.

Example

This first code example queries the database for order details that belong to Order #11000, marks these order details for deletion, and submits these changes to the database.

C#

Copy

```
// Query the database for the rows to be deleted.
var deleteOrderDetails =
    from details in db.OrderDetails
    where details.OrderID == 11000
    select details;

foreach (var detail in deleteOrderDetails)
{
    db.OrderDetails.DeleteOnSubmit(detail);
}

try
{
    db.SubmitChanges();
}
catch (Exception e)
{
    Console.WriteLine(e);
    // Provide for exceptions.
}
```

Example

In this second example, the objective is to remove an order (#10250). The code first examines the `OrderDetails` table to see whether the order to be removed has children there. If the order has children, first the children and then the order are marked for removal. The `DataContext` puts the actual deletes in correct order so that delete commands sent to the database abide by the database constraints.

C#

Copy

```
Northwnd db = new Northwnd(@"c:\northwnd.mdf");

db.Log = Console.Out;

// Specify order to be removed from database
int reqOrder = 10250;

// Fetch OrderDetails for requested order.
var ordDetailQuery =
```

```
from odq in db.OrderDetails
where odq.OrderID == reqOrder
select odq;

foreach (var selectedDetail in ordDetailQuery)
{
    Console.WriteLine(selectedDetail.Product.ProductID);
    db.OrderDetails.DeleteOnSubmit(selectedDetail);
}

// Display progress.
Console.WriteLine("detail section finished.");
Console.ReadLine();

// Determine from Detail collection whether parent exists.
if (ordDetailQuery.Any())
{
    Console.WriteLine("The parent is present in the Orders collection.");
    // Fetch Order.
    try
    {
        var ordFetch =
            (from ofetch in db.Orders
             where ofetch.OrderID == reqOrder
             select ofetch).First();
        db.Orders.DeleteOnSubmit(ordFetch);
        Console.WriteLine("{0} OrderID is marked for deletion.", ordFetch.OrderID);
    }
    catch (Exception e)
    {
        Console.WriteLine(e.Message);
        Console.ReadLine();
    }
}
else
{
    Console.WriteLine("There was no parent in the Orders collection.");
}

// Display progress.
Console.WriteLine("Order section finished.");
Console.ReadLine();

try
{
    db.SubmitChanges();
}
catch (Exception e)
{
    Console.WriteLine(e.Message);
}
```

```
        Console.ReadLine();  
    }  
  
    // Display progress.  
    Console.WriteLine("Submit finished.");  
    Console.ReadLine();
```

See Also

[How to: Manage Change Conflicts](#)

[How to: Assign stored procedures to perform updates, inserts, and deletes \(O/R Designer\)](#)

[Making and Submitting Data Changes](#)