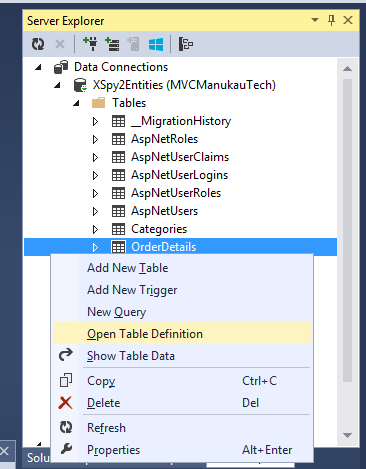
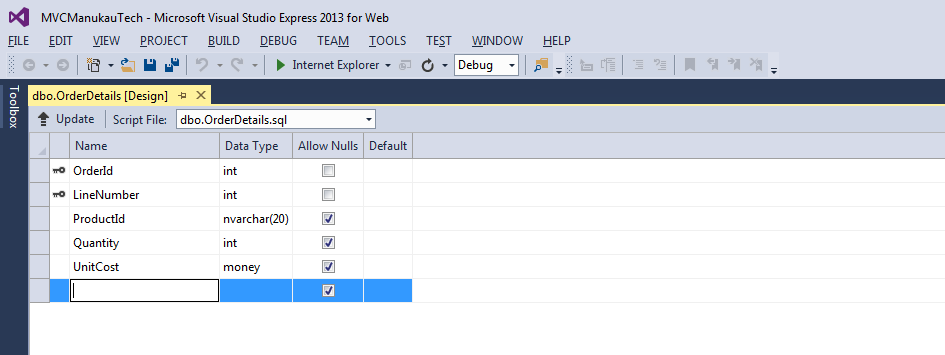
XSpy - howto cope with database design changes eg add a field.

Add field "Discount" to table "OrderDetails".  
Class discussion - students vote to make this a decimal fraction value.  
Eg a 10 percent discount will appear as 0.1  
a 15 percent discount will appear as 0.15

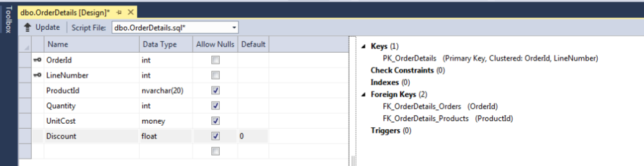
----------------------  
Start in "Server Explorer" panel. Open the database connection. Select table "OrderDetails".  
Select "Open Table Definition" to open its designer.



Add new field …



"Discount".   
"Allow Nulls" to go with the existing convention here.  
(Note: If I have a choice I usually try to run without "Allow Nulls" as much as possible.)   
Type "float", Default value of 0.



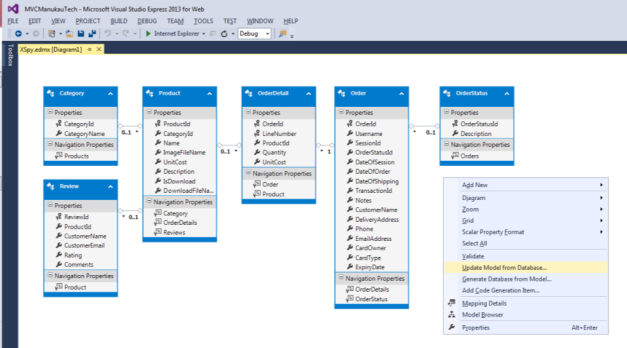
Click on "Update" (top left of this Design panel) then click "Update Database"

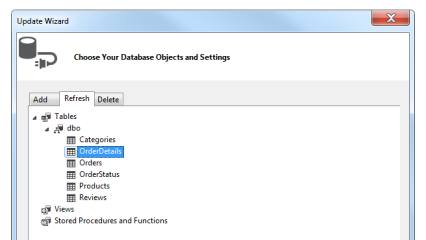
Check result in the "Data Tools Operations" panel which should appear at the bottom of Visual Studio.

We now need to mirror this change in the Model, Controller and View.

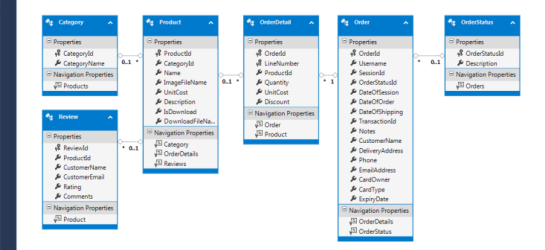
Back to the "Solution Explorer" panel and under "Models", open the .edmx file.

Right-click in diagram blank space and select "Update Model from Database"



This opens an "Update Wizard"  
Select Refresh 🡪 Tables 🡪 OrderDetails  


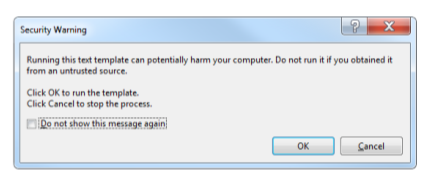
The new field should appear in the .edmx diagram. eg below see "Discount" added to "OrderDetail"



We have experienced a lot of challenges with this process.  
We inherited a database with fields named like "ProductID", "OrderID",   
and we wanted to change naming convention to "ProductId", "OrderId" etc  
Visual Studio appeared to be editing these names in its designer but they did not "stick" and in the meantime we had manually edited these names in the .edmx diagram. This worked until we added more fields.  
The "D-Fix" was to load the database "XSpy2" into "SQL Server Management Studio" as a full-on SQL database and edit the field names there. Then bring it back into this Visual Studio project.

Advice. If you run into lots of errors with this step, then it can work to delete the offending tables out of the visual diagram, then run "Update Model from Database" again, taking the first tab option "Add" and carefully selecting the table to re-import.  
We had another problem in that table "OrderStatus" originally imported as "OrderStatu" because of a default setting to rename these generated Model classes as singular. This "Artificial Intelligence" over-simplified this by cutting the "s" off. The fix there was to delete "OrderStatu" from the diagram, then run "Update Model from Database" with tab option "Add", select "OrderStatus" and clear a checkbox at the bottom of this popup window labelled "Pluralize or singularize generated object names".

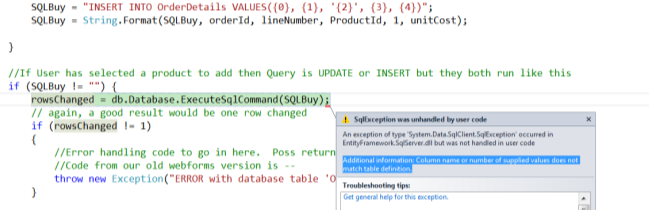
After any changes to our .edmx file/Models we need to build or run to compile the changes.  
Model changes give us this "Security Warning" dialog.   
Respond with OK - this seems to be common and normal edmx update behaviour.



Updating code to work with a new field.

Our approach was to test run and let the errors guide us to where code needs to change.

First crash was on adding a new product to the shopping cart.  
Crash on:



The villain here is the INSERT statement at the top of this screen shot.  
Look at the remarkably helpful error message.   
"Column name or number of supplied values does not match table definition."

Fix those 2 top statements by adding the extra "Discount" field as placeholder {5} with a default value of 0:

//140915 JPC added field Discount, add {5} and default value 0 here to handle Discount  
SQLBuy = "INSERT INTO OrderDetails VALUES({0}, {1}, '{2}', {3}, {4}, {5})";

SQLBuy = String.Format(SQLBuy, orderId, lineNumber, ProductId, 1, unitCost, 0);

Testing .. looks all good. Let me know if you find anything else!

Contact form for me is at   
<http://iafilm.blogspot.com>

John Calder  
Senior Lecturer in Computing  
Manukau Institute of Technology, Auckland South, New Zealand