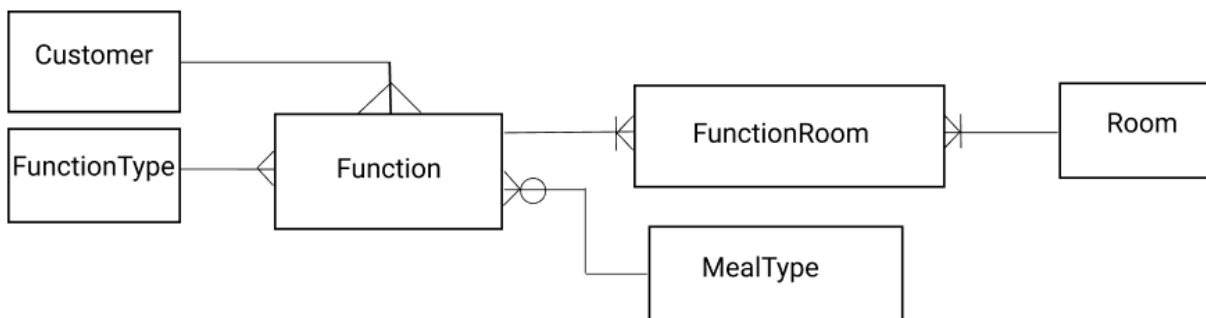


## Instructions

Continue with the ASP.NET Core MVC Catering Management application from Part 1. We will incorporate some additional Entities in the Model before going on, but there is a change!

Following discussions with the Client, they have requested a simplification that does away with the need for the FunctionDetail entity. Since almost all their business is made up of meetings, banquets and weddings, they want to keep it simple. On the very rare occasion that a function would have more than one component they want to just create an additional function record for the same client. As a result, we will eliminate the DurationDays property from the Function class and Date will become StartTime. We will also add a few new properties including EndTime, SetupNotes and the Boolean for Alcohol service. Of course, we will need to deal with some foreign key properties as well.



1. **(15 Marks)** In the Models folder, modify the Function entity and create classes for FunctionRoom, Room, and MealType. You will use the Fluent API to create the composite keys and Indexes, but use data annotations in the classes to implement the formatting and validation rules as listed below:

Field	Validation	Format/Notes
<b>Function Class (Revised – new properties shown in bold)</b>		
ID*		<i>Int Identity PK</i>
Name	String max length 120	
Lobby sign	String max length 60	
<b>SetupNotes</b>	<b>String max length 2000</b>	
<b>StartTime</b>	<b>Required, DateTime</b>	
<b>EndTime</b>	<b>DateTime</b>	If entered, EndTime comes after StartTime
BaseCharge	Required, Double	
PerPersonCharge	Required, Double	
GuaranteedNumber	Required, Int > 0	Default 1
SOCAN	Required, Double	Default \$50.00
Deposit	Required, Double	
DepositPaid	Boolean	Default False
NoHST	Boolean	Default False
NoGratuity	Boolean	Default False
<b>Alcohol</b>	<b>Boolean</b>	Default False
<b>MealTypeID</b>	<b>Int</b>	<i>Foreign Key</i>
CustomerID	Required, Int	<i>Foreign Key</i>
FunctionTypeID	Required, Int	<i>Foreign Key</i>

FunctionRoom Class			
FunctionID*	Required	Int PK, FK	Composite
RoomID*	Required	Int PK, FK	
Room Class			
ID*		Int Identity PK	
Name	Required, max length 80		
Capacity	Required Int > 0	Int	
MealType Class			
ID		Int Identity PK	
Name	Required, max length 80		

- a. Make sure that you supply user friendly error messages and display names.
  - b. All relationships should RESTRICT cascade delete except Function to FunctionRoom. If you delete a Function then all of the FunctionRoom records should be deleted as well.
2. **(5 Marks)** Modify your seed data and add new code to seed records in each entity.
  3. **(10 Marks)** Modify the controller and views for Functions to work with the new properties involved.
    - a. Don't forget that a property has been removed from the Function class!
    - b. Maintain protection against overposting attacks with the changes to properties in the Function class.

Note: You will **not** hand in Part 2A. This work is in preparation for the next steps in Part 2 on the MVC Catering Management application.

Note that controllers and views for Room and MealType will be added later on.