Mapping Many-to-Many Relationships

Torben Jensen
Developer/Cloud Architect

Overview

Associates multiple records in one table with multiple records in another

Items and orders in Carved Rock data model

Direct Many-to-Many with skip navigations

Indirect Many-to-Many

Direct Many-to-Many Relationship

Simple to set up

Easy to use

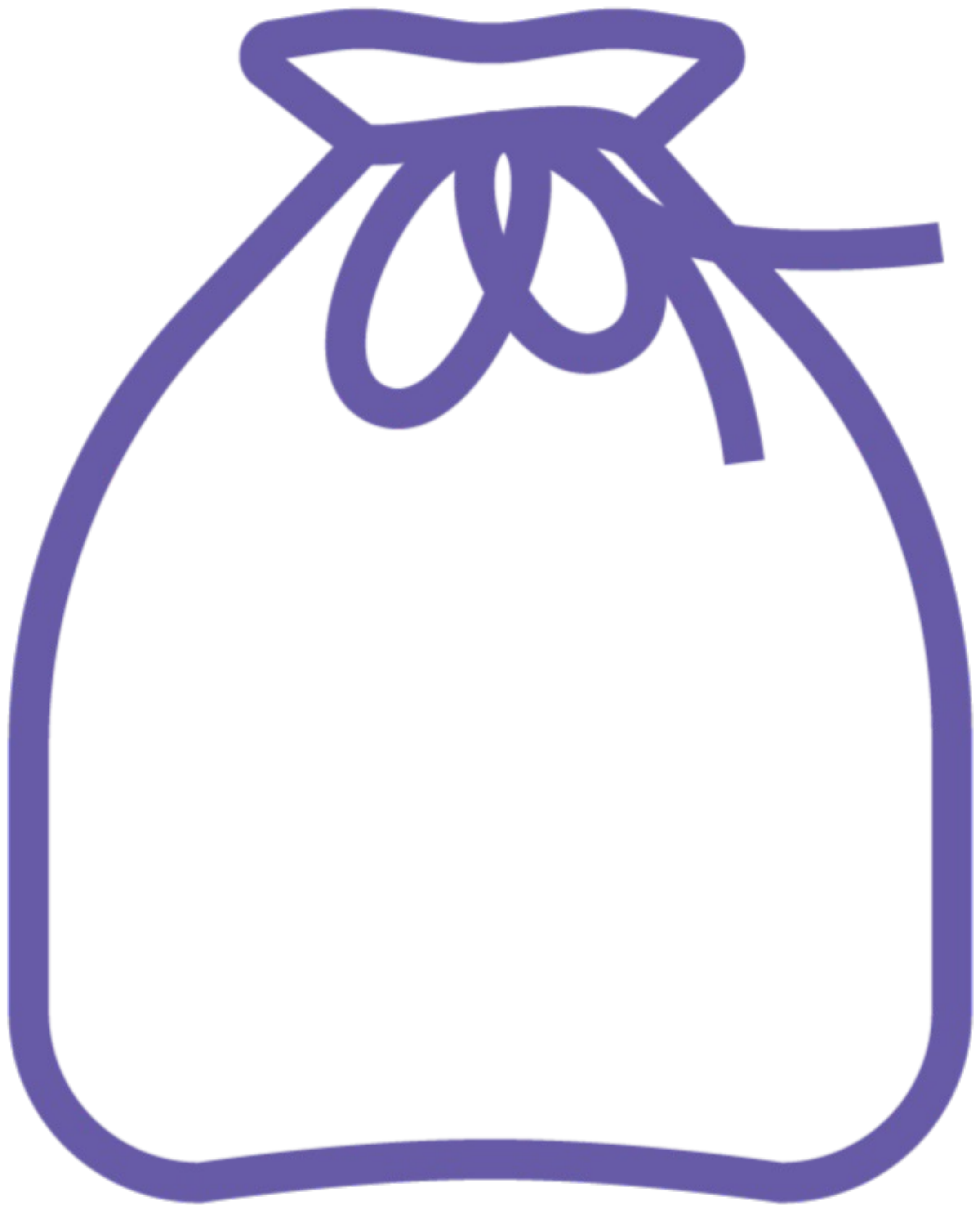
Natural from an object-oriented point of view

Database uses join table

Good for simple joins

Can’t add extra data

Skip Navigations without Payload



EF Core uses property bag entities under the covers

Which EF Core uses to create join entity

We can interact with hidden join entity

Demo

Fetching items in an order

Demo

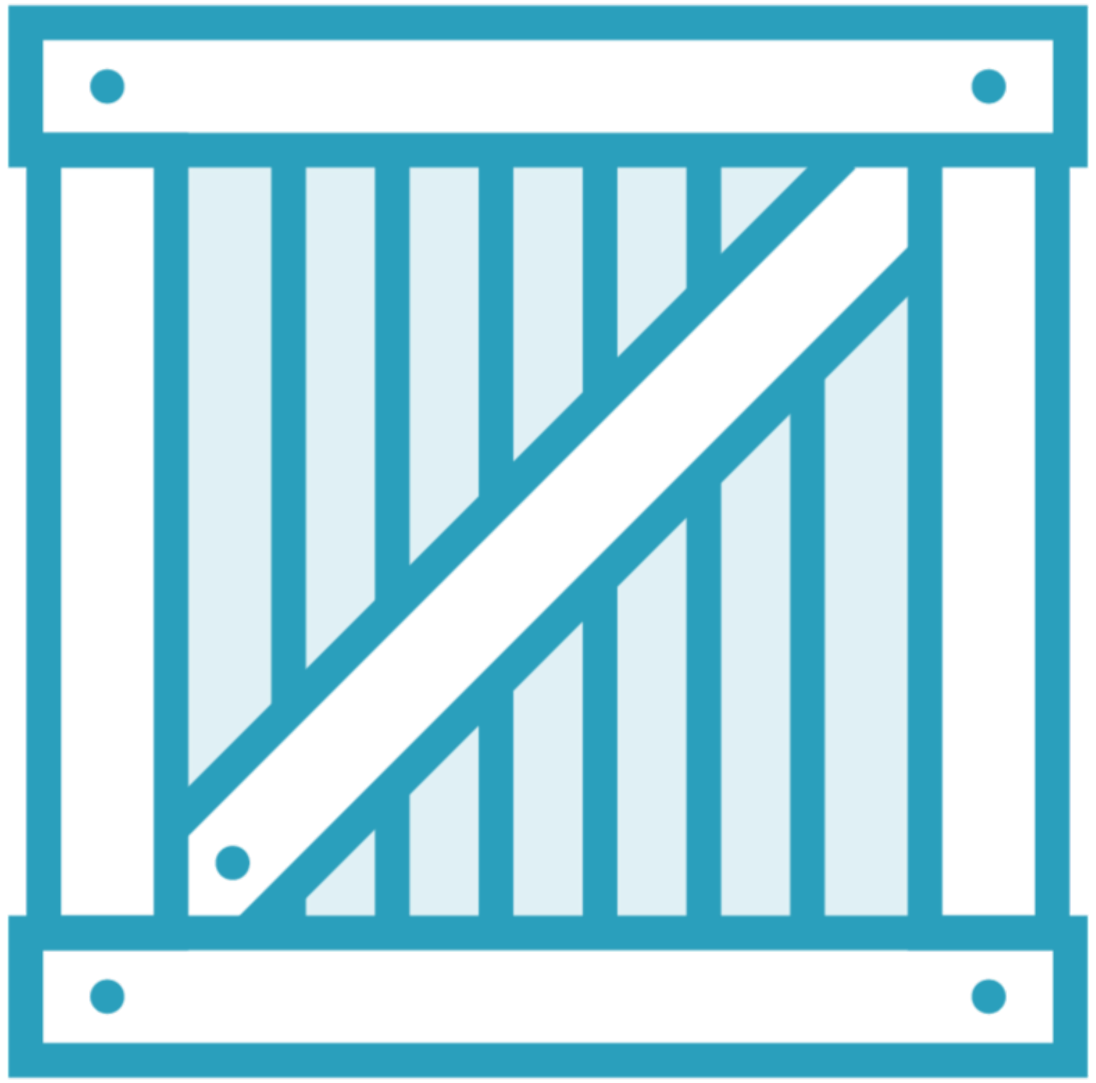
Interacting with the join entity

Property Bag Entity

```
modelBuilder
  .SharedTypeEntity<
    Dictionary<string, object>>
  ("Tag",
  entity =>
  {
    entity.Property<int>("Id");
    entity.Property<int>("CustomerId");
    entity.Property<string>("Nickname");
    entity.Property<int>("DiscountRate");
  });

DbSet<Dictionary<string, object>> Tags =>
  Set<Dictionary<string, object>>("Tag");
```

Skip Navigations with Payload



Augment join entity

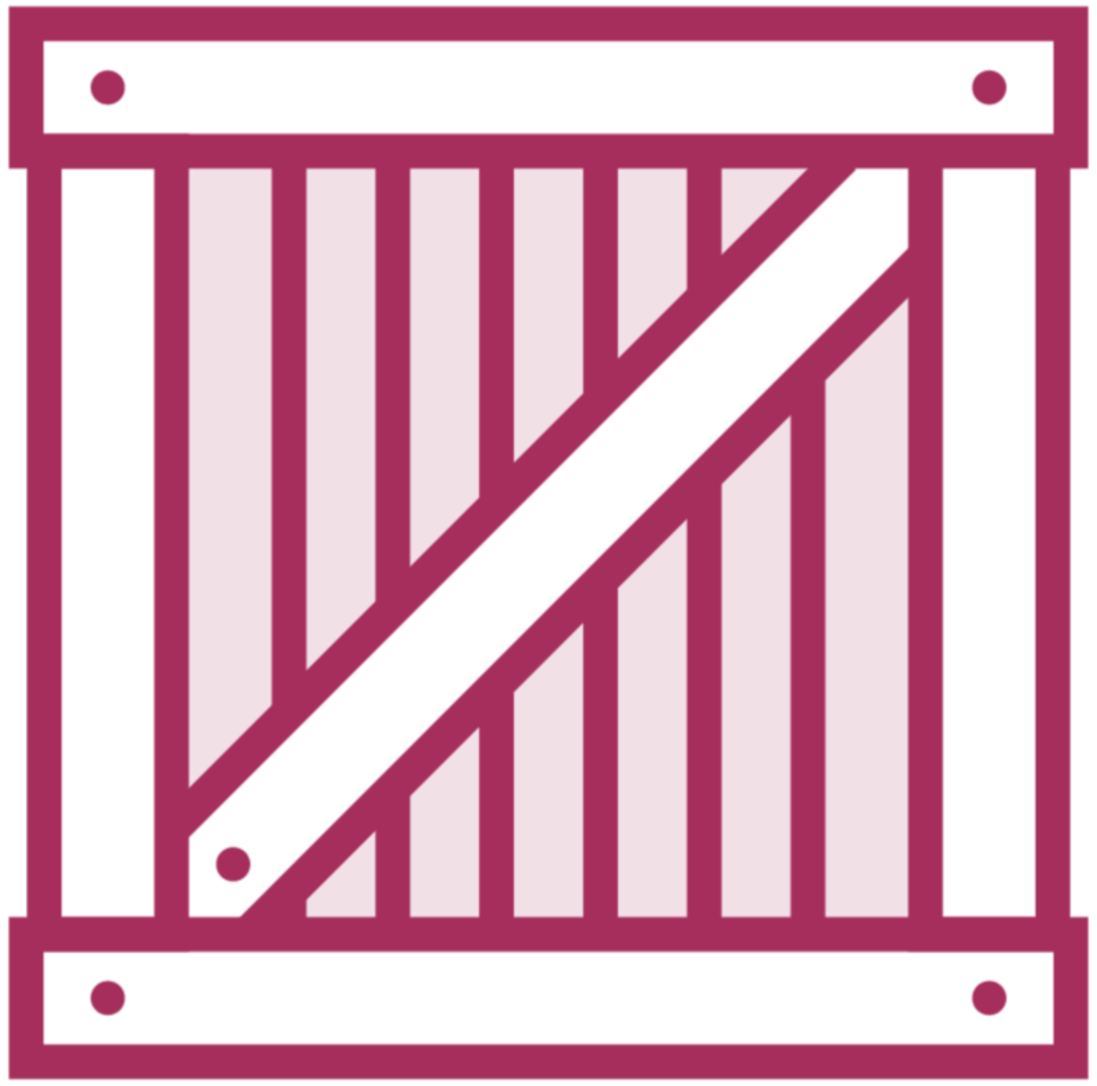
By adding a payload to it

We must create a class for the join entity

Set up relationship between items and orders

Database defined default values

Adding Payload to ItemOrder



Improve how we've related items and orders

Create ItemOrder entity

Add quantity to ItemOrder join table

EF Core allows this without re-wiring entire application

Only change code that needs to deal with the payload

Demo

Add payload to skip navigation

Create ItemOrder join entity

- Order date with default value
- Quantity

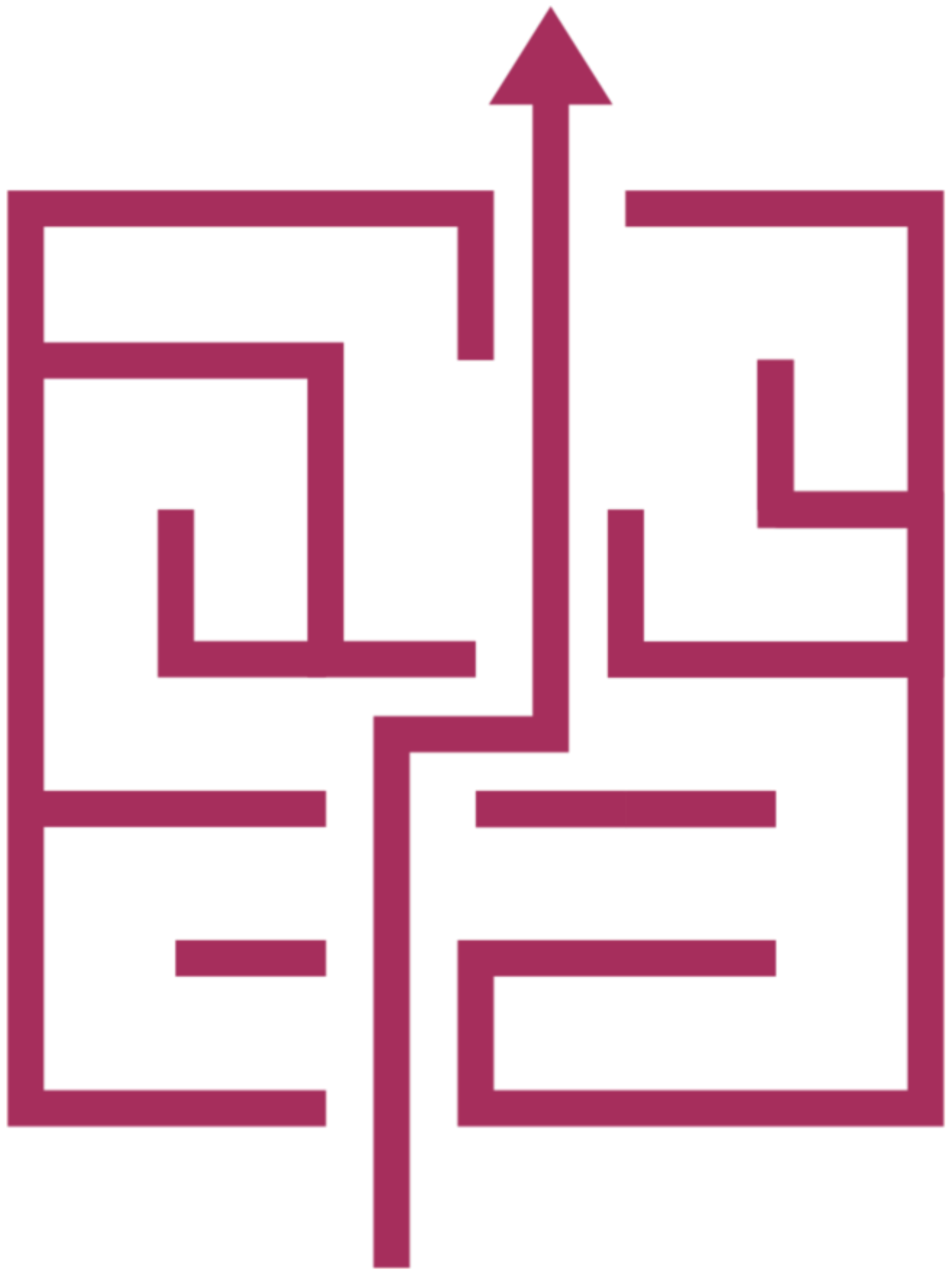
ItemOrder.cs

```
class ItemOrder
{
 Item Item { get; set; }
 int ItemsId { get; set; }
 Order Order { get; set; }
 int OrdersId { get; set; }
 DateTime OrderDate  { get; set; }
 int Quantity { get; set; }
}
```

Demo

Interacting with our new payload

What Did We Just Do?



Interact with join entity

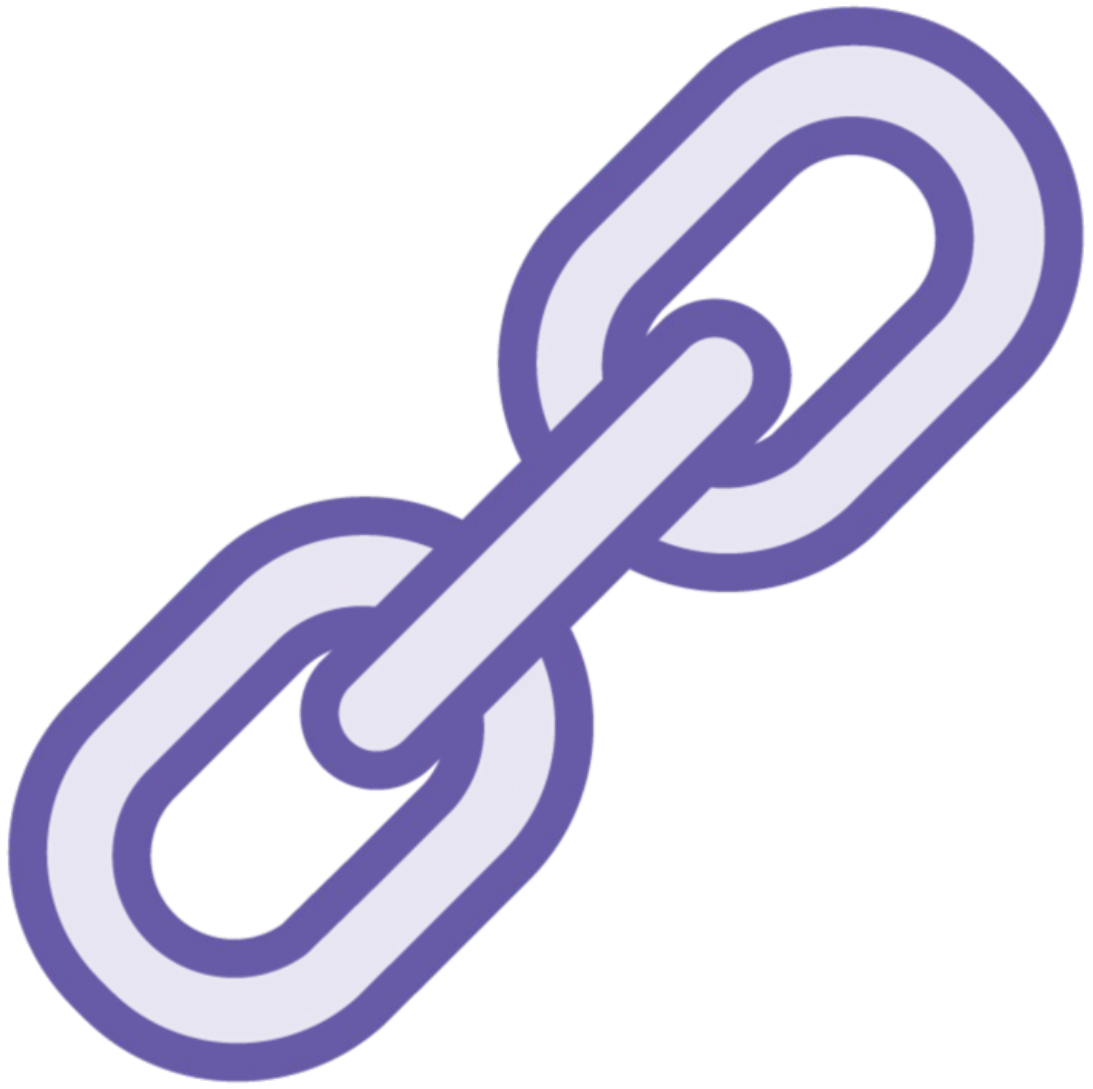
Created by EF Core to implement skip navigation

Property bag entities

Add payload to skip navigation

Fetch and update payload data

Indirect Many-to-Many Relationship



Replace skip navigations with navigation properties that use join entity

Code that interacts with payload data must be changed

Enables easier access to payload

Demo

Indirect many-to-many relationship

Many-to-Many

Item.cs BEFORE

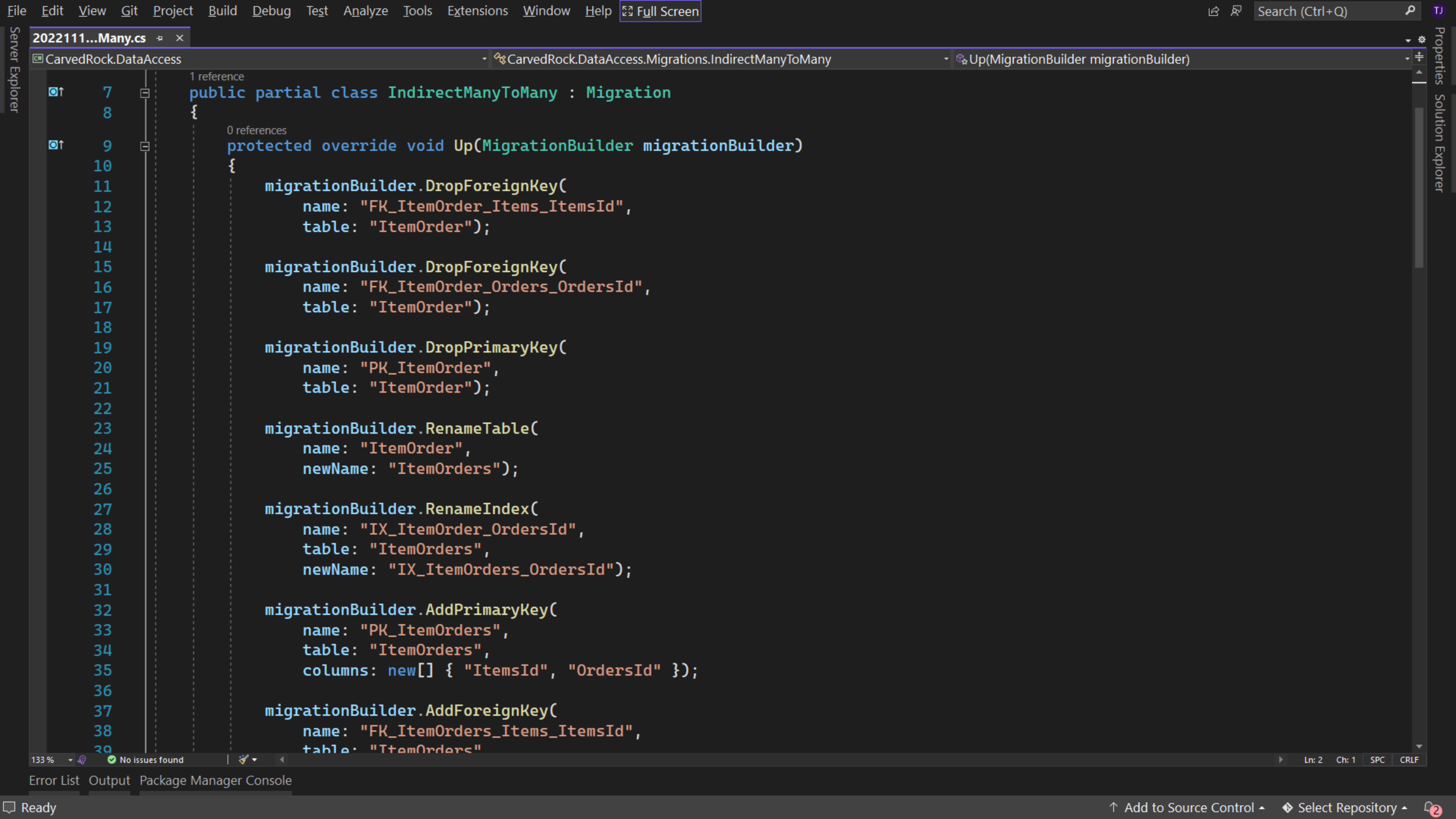
```
class Item
{
 int Id { get; set; }
 string Description { get; set; }
 decimal Price { get; set; }
 decimal PriceAfterVat { get; set; }
 float Weight { get; set; }

 ICollection<Order> Orders { get; set; }
}
```

Item.cs AFTER

```
class Item
{
 int Id { get; set; }
 string Description { get; set; }
 decimal Price { get; set; }
 decimal PriceAfterVat { get; set; }
 public float Weight { get; set; }

 ICollection<ItemOrder> Orders { get; set; }
}
```


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help Full Screen
Search (Ctrl+Q)
2022111...Many.cs
CarvedRock.DataAccess
CarvedRock.DataAccess.Migrations.IndirectManyToMany
Up(MigrationBuilder migrationBuilder)
Server Explorer
Properties
Solution Explorer
133 %
No issues found
Ln: 2 Ch: 1 SPC CRLF
Error List Output Package Manager Console
Ready
Add to Source Control
Select Repository

```
1 reference
public partial class IndirectManyToMany : Migration
{
    0 references
    protected override void Up(MigrationBuilder migrationBuilder)
    {
        migrationBuilder.DropForeignKey(
            name: "FK_ItemOrder_Items_ItemsId",
            table: "ItemOrder");

        migrationBuilder.DropForeignKey(
            name: "FK_ItemOrder_Orders_OrdersId",
            table: "ItemOrder");

        migrationBuilder.DropPrimaryKey(
            name: "PK_ItemOrder",
            table: "ItemOrder");

        migrationBuilder.RenameTable(
            name: "ItemOrder",
            newName: "ItemOrders");

        migrationBuilder.RenameIndex(
            name: "IX_ItemOrder_OrdersId",
            table: "ItemOrders",
            newName: "IX_ItemOrders_OrdersId");

        migrationBuilder.AddPrimaryKey(
            name: "PK_ItemOrders",
            table: "ItemOrders",
            columns: new[] { "ItemsId", "OrdersId" });

        migrationBuilder.AddForeignKey(
            name: "FK_ItemOrders_Items_ItemsId",
            table: "ItemOrders"
```

Demo

Interacting with the indirect many-to-many relationship

What Did We Just Do?

Transformed direct Many-to-Many into indirect mapping

Easier access to payload data

Code changes required

Demo

Create, update, and delete

What Did We Just Do?

Create

Update

Delete

Summary

Mapping Many-to-Many relationships

Direct Many-to-Many relationships

Interacting with join entities for skip navigations

Adding payload data to skip navigations

Indirect Many-to-Many relationship

Summary

Skip navigations for simple cases

Payload can be added

Direct Many-to-Many for more complex scenarios