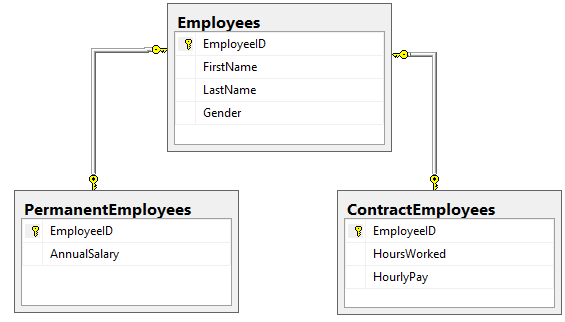
**Understanding Table Per Type Inheritance in Entity Framework**

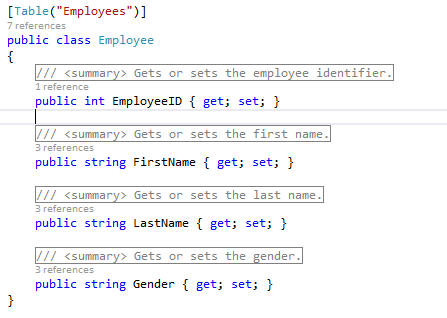
In Entity Framework, the Table Per Type inheritance is used in the way that each entity in the hierarchy of entities is mapped to a separate table in storage schema. It means, there is separate table in database to maintain data for each Entity Type. This article will demonstrate how we can make use of Table Per Type Inheritance of Entity Framework using Code First approach

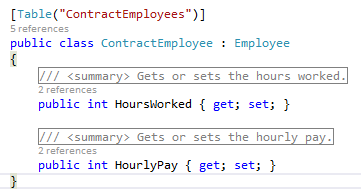


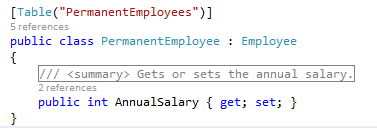
In TPT inheritance one database table per type is used to store data for the respective entity types in the inheritance hierarchy. Below are the steps to implement TPT hierarchy in Entity Framework

**Create Models**

Create model classes that will reflect the database hierarchy as shown in screenshot above.

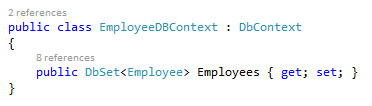






**Create DB Context Class**

Create another class that will inherit from DbContext class and add DbSet list property of Employee model



**Data Invocation**

In the code behind file, call should be made as below to fetch the data of respective employee type

employeeDBContext.Employees.OfType<PermanentEmployee>().ToList<Employee>()

employeeDBContext.Employees.OfType<ContractEmployee>().ToList<Employee>()

employeeDBContext.Employees.ToList()

Attached is the sample application that demonstrates how we can achieve TPT in ASP.Net MVC Web application.

ASP.Net MVC,MVC,TPT,Table Per Type

References

<https://msdn.microsoft.com/en-us/data/jj618293.aspx>

<http://blog.devart.com/table-per-type-vs-table-per-hierarchy-inheritance.html>