var insertQuery = model.Into(model.PartName)

.Value(d => d.Name, datacontract.Name)

.Value(d => d.Description, datacontract.Description)

.Value(d => d.IsCover, datacontract.IsCover)

.Value(d => d.IsSheet, datacontract.IsSheet)

.Value(d => d.PictureMatching, datacontract.PictureMatching);

CallInsertQuery(partName, insertQuery);

Next:

//var periodicity = new MediumPublicationPeriodicity();

//var insertQuery = model.Into(model.MediumPublicationPeriodicity)

// .Value(d => d.MediumId, mediumDataContract.Id)

// .Value(d => d.PeriodicityType, mediumDataContract.Periodicity.PeriodicityTypeId)

// .Value(d => d.DayInterval, mediumDataContract.Periodicity.DayInterval)

// .Value(d => d.BasicFrequency, mediumDataContract.Periodicity.BasicFrequency)

// .Value(d => d.DayOfWeek, mediumDataContract.Periodicity.DayOfWeek)

// .Value(d => d.DateFrom, mediumDataContract.Periodicity.DateFrom)

// .Value(d => d.Mo, mediumDataContract.Periodicity.Mo)

// .Value(d => d.Tu, mediumDataContract.Periodicity.Tu)

// .Value(d => d.We, mediumDataContract.Periodicity.We)

// .Value(d => d.Th, mediumDataContract.Periodicity.Th)

// .Value(d => d.Fr, mediumDataContract.Periodicity.Fr)

// .Value(d => d.Sa, mediumDataContract.Periodicity.Sa)

// .Value(d => d.Su, mediumDataContract.Periodicity.Su)

// .Value(d=> d.IsEnabled, mediumDataContract.Periodicity.IsEnabled);

//CallInsertQuery(periodicity, insertQuery);

**Zmenovadlo 8. února 2019**

Pokud je nejaka entita abstraktni, instanci si vytvorime takto:

using (var model = CreateDbContext())

{

var motivlet = Motivlet.CreateInstance();

var insertQuery = model.Into(model.Motivlet)

.Value(d => d.OwnerId, contract.OwnerId)

.Value(d => d.CompanyBrandId, contract.CompanyBrandId)

.Value(d => d.ProductBrandId, contract.ProductBrandId)

.Value(d => d.ProductDetailId, contract.ProductDetailId)

.Value(d => d.CategoryId, contract.CategoryId)

.Value(d => d.RoleId, contract.RoleId)

.Value(d => d.MarketId, contract.MarketId)

.Value(d => d.TelcoId, contract.TelcoId)

.Value(d => d.GenderId, contract.GenderId)

.Value(d=> d.PlatformId, contract.PlatformId)

.Value(d => d.CreatedBy, userId);

CallInsertQuery(motivlet, insertQuery);

}

**Pokud nefunguje** CallInsertQuery(motivlet, insertQuery);  **staci pouzit normalni Insert. Entita musi implementovat rozhrani IAuditable. Ve zmenovadle to neimplementovala MotiveVersion a hlasilo mi to preteceni datumu ve sloupci Created.**

using (var model = CreateDbContext())

{

var motiveVersion = MotiveVersion.CreateInstance();

var original = model.MotiveVersion.First(d => d.Id == originalMotiveVersionId);

motiveVersion.MotiveId = original.MotiveId;

motiveVersion.ActiveFrom = activeFromDate;

motiveVersion.ActiveTo = originalActiveToDate;

motiveVersion.PrimaryMotivletId = original.PrimaryMotivletId;

motiveVersion.Name = original.Name;

motiveVersion.Campaign = original.Campaign;

motiveVersion.IsProduct = original.IsProduct;

motiveVersion.IsSponsorship = original.IsSponsorship;

motiveVersion.IsShow = original.IsShow;

motiveVersion.IsCoupon = original.IsCoupon;

motiveVersion.IsAnnouncement = original.IsAnnouncement;

motiveVersion.IsMedialPartnership = original.IsMedialPartnership;

motiveVersion.IsGeneralMeeting = original.IsGeneralMeeting;

motiveVersion.NotMatchingSubject = original.NotMatchingSubject;

Insert(motiveVersion, model);

}

**Insert z jedne tabulky na serveru do jiné bez tahani dat do aplikace.**

model.TempMotivlet.Into(model.Motivlet)

.Value(x => x.OwnerId, d => d.OwnerId)

.Value(x => x.CompanyBrandId, d => d.CompanyBrandId)

.Value(x => x.ProductBrandId, d => d.ProductBrandId)

.Value(x => x.ProductDetailId, d => d.ProductDetailId)

.Value(x => x.CategoryId, d => d.CategoryId)

.Value(x => x.RoleId, d => d.RoleId)

.Value(x => x.MarketId, d => d.MarketId)

.Value(x => x.GenderId, d => d.GenderId)

.Value(x => x.PlatformId, d => d.PlatformId)

.Value(x => x.TelcoId, d => d.TelcoId)

.Value(x => x.CreatedBy, d => d.CreatedBy) Spatne pokud tvorim nove motivlety

.Value(x => x.Created, d => d.Created)

.Value(x => x.Modified, d => d.Modified)

.Value(x => x.ModifiedBy, d => d.ModifiedBy).Insert();

Pozor na auditni stopu !

**Insert Into Select : Insert do nejake tabulky datama pozbiranymi selectem :**

using (var model = CreateDbContext())

{

model.Command.CommandTimeout = 120;

var query = from mtm in model.MotiveVersionToMotivlet

join temp in model.TempMotiveVersion on mtm.MotiveVersionId equals temp.Id

join ver in model.MotiveVersion on new { temp.MotiveId, temp.ActiveFrom, temp.ActiveTo } equals new { ver.MotiveId, ver.ActiveFrom, ver.ActiveTo }

select new { mtm.MotivletId, MotiveVersionId = ver.Id };

var queryable = query.Into(model.MotiveVersionToMotivlet)

.Value(x => x.MotiveVersionId, d => d.MotiveVersionId)

.Value(x => x.MotivletId, d => d.MotivletId)

.Insert();

}

Pozor na auditni stopu :

model.TempMotiveVersion.Into(model.MotiveVersion)

.Value(x => x.MotiveId, d => d.MotiveId)

.Value(x => x.ActiveFrom, d => d.ActiveFrom)

.Value(x => x.ActiveTo, d => d.ActiveTo)

.Value(x => x.PrimaryMotivletId, d => d.PrimaryMotivletId)

.Value(x => x.Name, d => d.Name)

.Value(x => x.Campaign, d => d.Campaign)

.Value(x => x.IsProduct, d => d.IsProduct)

.Value(x => x.IsSponsorship, d => d.IsSponsorship)

.Value(x => x.IsShow, d => d.IsShow)

.Value(x => x.IsCoupon, d => d.IsCoupon)

.Value(x => x.IsAnnouncement, d => d.IsAnnouncement)

.Value(x => x.IsMedialPartnership, d => d.IsMedialPartnership)

.Value(x => x.IsGeneralMeeting, d => d.IsGeneralMeeting)

.Value(x => x.NotMatchingSubject, d => d.NotMatchingSubject)

**.Value(x => x.CreatedBy, d => auditableIdentityProvider.UserIdentity.UserId)**

**.Value(x => x.Created, d => auditableIdentityProvider.CreatedStamp.DateTime)**

**.Value(x => x.Modified, d => auditableIdentityProvider.ModifiedStamp.DateTime)**

**.Value(x => x.ModifiedBy, d => auditableIdentityProvider.UserIdentity.UserId).Insert();**

Když tvorim nove verze musim pouzit AuditableIdentityProvidera

# Insert batch

Je rychly insert.

public void InsertTempVersions(List<TempMotiveVersion> tempVersions)

{

using (var model = CreateDbContext())

{

model.TempMotiveVersion.Delete();

model.InsertBatch(tempVersions);

}

}