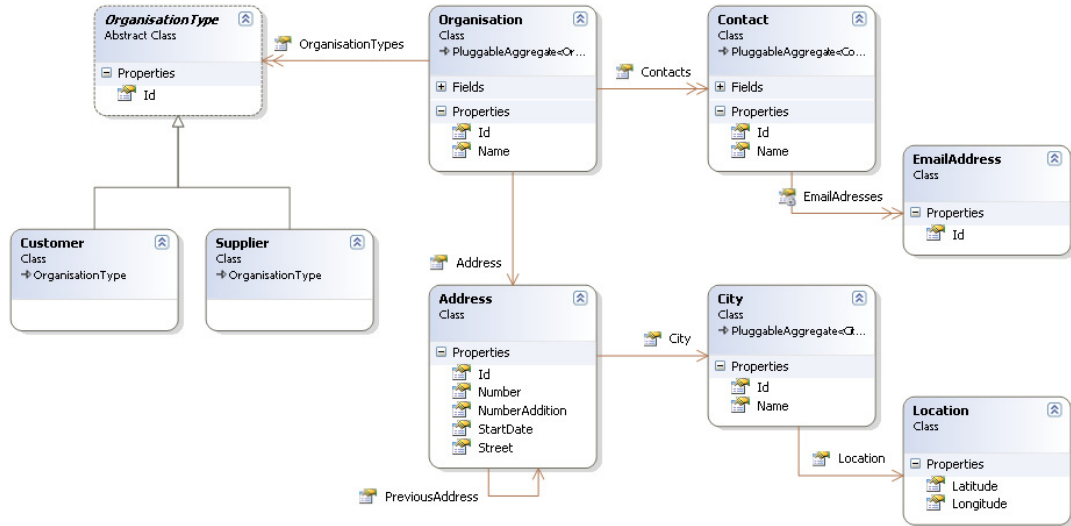


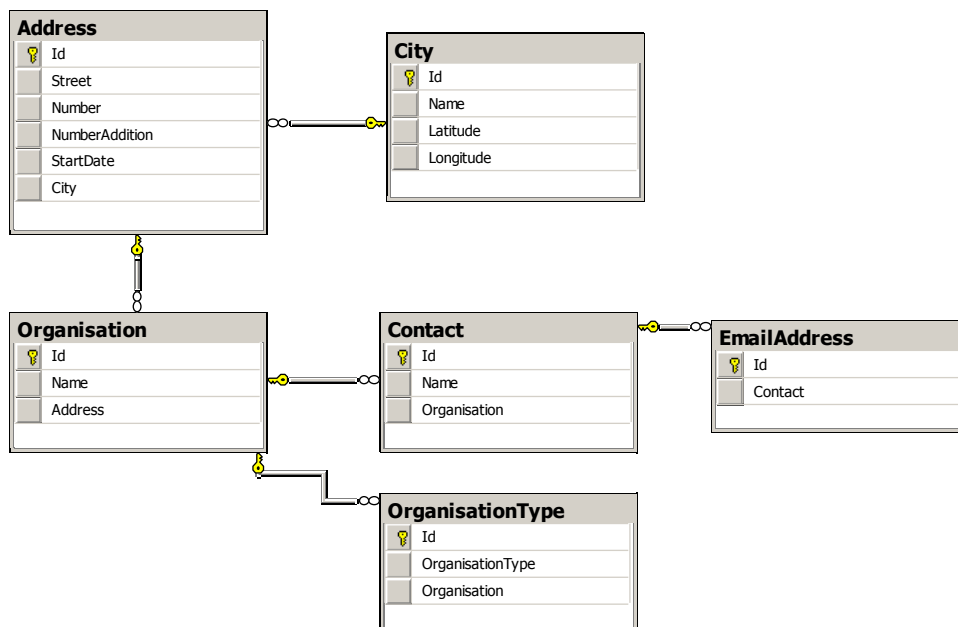
NHibernate Attribute Mappings

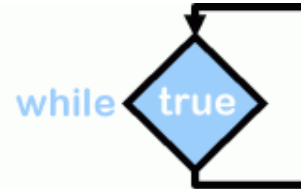
This reference sheet contains the most commonly used mappings using NHibernate Attributes.

Interesting patterns used are subclassing, one-to-many associations, many-to-one associations and component mappings.



The resulting database schema of the mapping (code on next page) is given below.





<pre>[Class] public class Organisation : PluggableAggregate<Organisation> { [Id(Name = "Id")] [Generator(1, Class = "native")] public int Id { get; set; } [Property] public string Name { get; set; } [ManyToOne] public Address Address { get; set; } private List<Contact> contacts; [Bag(Name = "Contacts", Cascade = "all", Lazy = true)] [Key(1, Column = "Organisation")] [OneToMany(2, ClassType = typeof(Contact))] public IList<Contact> Contacts { get { return contacts; } set { contacts = value != null ? value.ToList() : null; } } private List<OrganisationType> organisationTypes; [Bag(Name = "OrganisationTypes", Cascade = "all", Lazy = true)] [Key(1, Column = "Organisation")] [OneToMany(2, ClassType = typeof(OrganisationType))] public IList<OrganisationType> OrganisationTypes { get { return organisationTypes; } set { organisationTypes = value != null ? value.ToList() : null; } } }</pre>	<pre>[Class] public class Address { [Id(Name = "Id")] [Generator(1, Class = "native")] public int Id { get; set; } [Property] public string Street { get; set; } [Property] public int Number { get; set; } [Property] public string NumberAddition { get; set; } [ManyToOne] public City City { get; set; } [ManyToOne] public Address PreviousAddress { get; set; } [Property] public DateTime? StartDate { get; set; } }</pre>
<pre>[Class] public class City:PluggableAggregate<City> { [Id(Name = "Id")] [Generator(1, Class = "native")] public int Id { get; set; } [Property] public string Name { get; set; } [ComponentProperty] public Location Location { get; set; } }</pre>	<pre>[Class] public class Contact : PluggableAggregate<Contact> { [Id(Name = "Id")] [Generator(1, Class = "native")] public int Id { get; set; } [Property] public string Name { get; set; } List<EmailAddress> emailAddresses; [Bag] [Key(1, Column = "Contact")] [OneToMany(2, ClassType = typeof(EmailAddress))] IList<EmailAddress> EmailAddresses { get { return emailAddresses; } set { emailAddresses = value != null ? value.ToList() : null; } } }</pre>
<pre>[Class] [Discriminator(1, Column = "OrganisationType")] public abstract class OrganisationType { [Id(Name = "Id")] [Generator(1, Class = "native")] public int Id { get; set; } }</pre>	<pre>[Component] public class Location { [Property] public int Latitude { get; set; } [Property] public int Longitude { get; set; } }</pre>
<pre>[Subclass(DiscriminatorValue = "S", ExtendsType=typeof(OrganisationType))] public class Supplier:OrganisationType { }</pre>	