

技术方案

方案	描述
Aggregation关系	聚合关系：通过主业务对象GetAggregationDetail()管理从业务对象；主业务对象被删除时，从业务对象不会被级联删除，只会断开Link关系（外键字段值被设置为null）；如果不管理实体，则在Root类上定义Phenix.Core.Mapping.ClassDetailAttribute并将CascadingDelete = false。
Composition关系	组合（强聚合）关系：通过主业务对象GetCompositionDetail()管理从业务对象；主业务对象被删除时，从业务对象会被级联删除；如果不管理实体，则在Root类上定义Phenix.Core.Mapping.ClassDetailAttribute并将CascadingDelete = true；
primary key约束	在数据库中构建物理的primary key约束；一般表的结构采取的是唯一主键（ID NUMERIC(15)）方法，而有些在业务上肯定不会再被它表挂接的叶子级别表，就仅采取将外键字段作为主键的方法；
foreign key约束	在数据库中构建物理的foreign key约束
unique index约束	在数据库中构建物理的unique index约束
留痕：input信息	添加后缀为“_INPUTER”、“_DEPARTMENT”、“_POSITION”、“_INPUTTIME”、“_INPUTERIP”的字段；当增加、删除记录时会自动被记录下操作者信息；
虚拟删除：disabled信息	添加后缀为“_DISABLED”的NUMERIC(1)类型字段；当删除记录时不会被物理删除，仅改它的值为1（缺省为0）；正常Fetch时是不会下载的，除非显式申明；
ObjectRule	构建对象级别的规则类
EditValidationRule	构建属性有效性规则类
RangeCompareRule	构建比较范围规则类
LockProcess	利用过程锁，独占方式操作数据
OnInitializeNew	在业务对象新增初始化时实现
OnInsertingSelf	在业务对象新增提交前实现，可共用同一事务
OnInsertedSelf	在业务对象新增提交后实现，可共用同一事务
OnUpdatingSelf	在业务对象更新提交前实现，可共用同一事务
OnUpdatedSelf	在业务对象更新提交后实现，可共用同一事务
OnDeletingSelf	在业务对象删除提交前实现，可共用同一事务

OnDeletedSelf	在业务对象删除提交后实现，可共用同一事务
OnSelectedValueChanged	在业务对象被选或不被选之后实现
不允许手工新增：AllowAddItem return false	业务集合对象IBusinessCollection.AllowAddItem，控制UI手工操作权限（不是业务对象增删改权限）
不允许手工编辑：AllowEditItem return false	业务集合对象IBusinessCollection.AllowEditItem，控制UI手工操作权限（不是业务对象增删改权限）
不允许手工删除：AllowDeleteItem return false	业务集合对象IBusinessCollection.AllowDeleteItem，控制UI手工操作权限（不是业务对象增删改权限）
不允许手工设置：AllowSet return false	业务对象IBusinessObject.AllowSet，控制UI手工操作权限（不是业务对象增删改权限），等于不允许手工编辑和删除
不允许手工编辑：AllowEdit return false	业务对象IBusinessObject.AllowEdit，控制UI手工操作权限（不是业务对象增删改权限）
不允许手工删除：AllowDelete return false	业务对象IBusinessObject.AllowDelete，控制UI手工操作权限（不是业务对象增删改权限）
不允许手工编辑（属性）：禁用set语句	internal、private或注释掉业务对象中指定的属性的set语句
Fetch() 函数	“Fetch”是构建业务对象或业务集合对象的函数命名规范；如果自行编写类似功能的函数，也应该使用“Fetch”来命名它
New() 函数	“New”是新增业务对象或业务集合对象的函数命名规范；如果自行编写类似功能的函数，也应该使用“New”来命名它