

1 数据库设计规范

Phenix 对数据库的设计有一定要求，并与 Addin 代码生成工具、持久层引擎一起，协同实现应用软件的快速开发及产品化。

1.1 数据结构命名规范

数据结构的规范命名，一方面为 Addin 代码生成工具提供便利，一方面也为开发设计过程、调试维护过程，提供快速指引、望文生义的帮助。

1.1.1 表命名规则

规则：XXX_名称

解释：前缀为三个字符，建议为所属模块的缩写

1.1.2 字段命名规则

规则：XXX_名称

解释：前缀为三个字符，建议为所属表的缩写

特例：枚举类型、布尔类型字段，需加 “_FG” 后缀；主、外键字段，需加 “_ID” 后缀；其他具体要求见下文“约定字段命名”章节

1.1.2.1 主键命名规则

规则：PK_所属表名

1.1.2.2 外键命名规则

规则：FK_所属字段名

1.1.2.3 索引命名规则

规则：I_名称

1.1.2.4 视图命名规则

规则：XXX_名称_V

解释：前缀为三个字符，建议为所属模块的缩写

1.1.2.5 存储过程命名规则

规则：XXX_名称_P

解释：前缀为三个字符，建议为所属模块的缩写

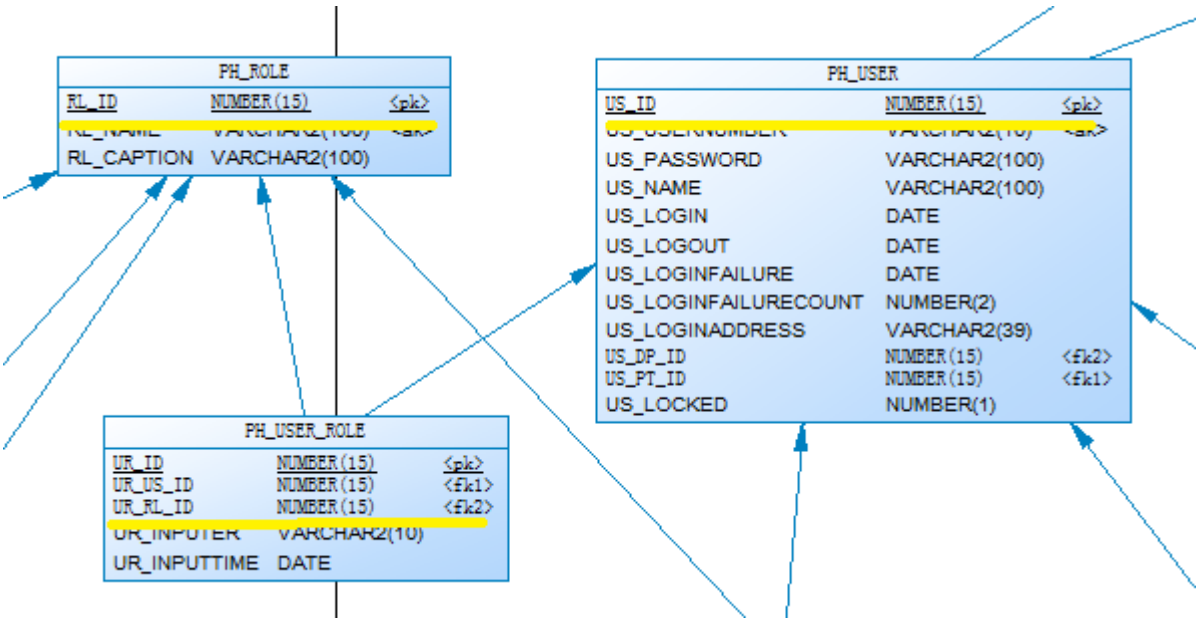
1.2 数据类型规则

数据类型尽量采用 Oracle、SQL_Server 通用的，以实现产品化要求：

ORACLE	SQL SERVER	业务类
--------	------------	-----

NUMERIC	NUMERIC	浮点: decimal 精度<10: int 精度>=10: long
VARCHAR	VARCHAR	String
DATE	DATETIME	DateTime
LONG RAW	IMAGE	byte[]
LONG	TEXT	String

1.3 主键的设计方法



主键的设计没有一个定论，各人有各人的方法。

如上图示例，Phenix 推荐采用无意义、非递增、NUMERIC(15)非空格式的主键。这样的设计，使得主、外键的关联关系变得非常简单、有规则，系统性能也可得到提高，开发过程也因 Phenix 自动化工具的配套支持而变得高效。

1.3.1 主键的必要性

每个表都应该具有主键，不管是单字段主键还是多字段主键。主键的存在就代表着业务对象的唯一性。主键用于主、从表的外键关联，代表着数据的完整性。如果不用主键，表记录的修改与删除操作会变得非常麻烦。

1.3.2 主键的无意义性

主键不应该具有实际的意义，这可能对于一些朋友来说不太认同。比如，订单表会有“订单编

号”字段，这个字段在业务逻辑中应该具有唯一性，具有唯一标识记录的功能。但是，具有实际意义的字段必定具有“意义更改”的可能性。比如，订单编号在正常流程下一切顺利，但是客户说“订单可以作废，并重新生成订单，而且订单号要保持原订单号一致”，这样的设计就将面临大幅重构的风险。因此，推荐新设一个字段专门用做主键，此主键在界面上不被显示出来、本身在业务逻辑中也不具有实际意义，仅唯一标识记录。

1.3.3 随机数主键

主键不应该定义为递增字段，这主要是为了预防数据库的迁移和合并，特别是复杂关联关系的表单的合并，无形中会增加产品的实施工作量。所以，推荐使用 Phenix 提供的 `Phenix.Core.Data.Sequence.Value`，为主键字段自动赋值。

主键一旦被定义成字段名后缀为“_ID”且字段类型为 `NUMERIC(15)`，即纳入 Phenix 的统一管理。比如，当 New 业务对象时会自动填充其映射的主键属性值为 `Phenix.Core.Data.Sequence.Value`，这个值由“当前时间+当前机器标识+当前进程 ID+随机数+序列号”组成，出现并发重复的概率极低。

如果仍然不放心，比如在本地多进程环境下做批量 New 业务对象作业的话，可以将这个业务类的 `Phenix.Core.Mapping.ClassAttribute` 标签属性 `SequenceFromServer = true`。这样，这个类的对象在 New 时，会从服务器上获取到主键值，出现重复的概率基本为 0。

采取本方法较之常规方法（从后台数据库获取），可节省数据库操作及跨物理域传递信息的来回时间的消耗。

1.4 外键的设计方法

与主键的设计一样，外键也要采用 `NUMERIC(15)` 非空格式。

从表的外键，表达了业务对象的两种关联关系，即组合关系（`CompositionDetail`）、聚合关系（`AggregationDetail`）。当业务对象在聚合关系下 Unlink 或直接赋 `null` 值给业务对象外键属性时，Phenix 会自动修正其为默认值（需要表结构设计中定义该字段缺省值（建议为-1），以代表无外键关联）。这样，组合关系的表结构应该保有物理外键，而聚合关系可剔除物理外键、且用缺省值代表无外键关联。当持久化的时候，组合关系会因有物理外键的约束条件而抛出异常给开发人员（需修正代码缺陷，即不应保存无外键关联的值），而聚合关系的从表可成功提交。

以上外键的设计方法，主要是为了配合数据库性能优化的要求。

需注意的是，剔除了物理外键的业务类外键字段，应用 `Phenix.Core.Mapping.FieldLinkAttribute` 标签显式声明其与主表的关联关系。

1.5 约定表字段命名

由定义在 `Phenix.Core.Mapping.CodingStandards` 类中的属性约定，决定了 Phenix 是如何：

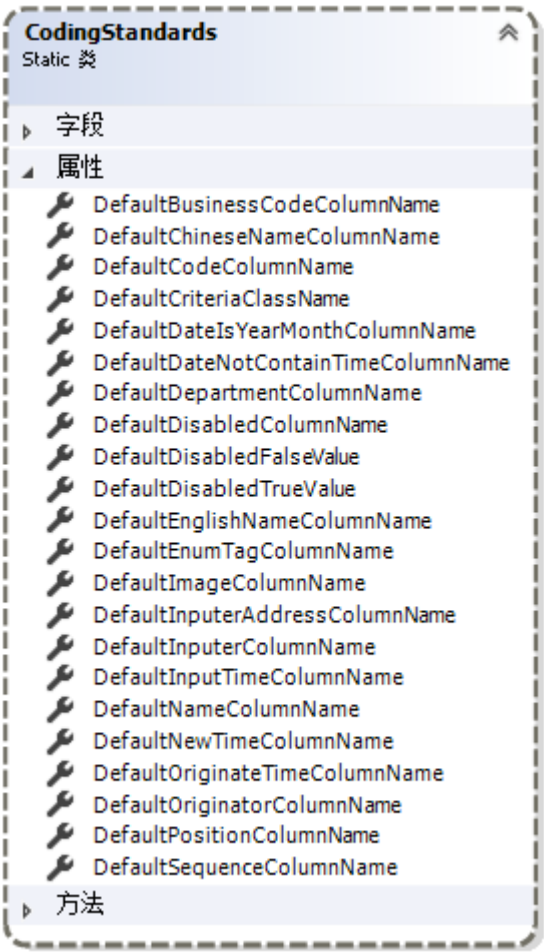
- 自动构建业务类上的映射关系代码（见“使用指南. 03. Addin 工具使用方法”的“初始化/编辑业务类”章节、“使用指南. 12. 业务结构对象模型”的“FieldAttribute”章节）；
- 自动识别枚举类型字段（见“使用指南. 12. 业务结构对象模型”的“枚举在业务结构中的使用方法”章节）；

约定：

- 1，枚举型表字段，字段名后缀为“_FG”，类型必须定义为 NUMERIC(2)；
- 2，布尔型表字段，字段名后缀为“_FG”，类型必须定义为 NUMERIC(1)；
- 3，在任何情形下，类似业务状态的枚举字段，都应处于一个有意义的状态值，字段必定不允许为空，且应将其缺省值设置为 0（对应枚举的初始状态值）；但是，业务对象的有些枚举字段，在某些业务场景下，无法赋值为清晰而有意义的枚举值，此时从模型设计和数据库性能考虑，建议仍定义为不允许为空，缺省值为-1（对应到枚举里为-1 的值（建议统一命名为 None），相当于 null 所代表的含义），举例如下：

```
TimeOfDay : int { None = -1, Morning, Afternoon, Evening }
```

- 强制数据输入规则（见“使用指南. 08. 业务对象公共接口的数据验证”的“通用的数据验证规则”章节）；



属性	含义	缺省字段名标识	字段类型
DefaultSequenceColumnName	关键字	字段名 ID、后缀为 “_ID”	NUMERIC (15)
DefaultNewTimeColumnName	新增时间	后缀为 “_NEWTIME”	DATE
DefaultBusinessCodeColumnName	业务码	后缀为 “SERIAL”	VARCHAR ()
DefaultOriginatorColumnName	制单人	后缀为 “_ORIGINATOR”	VARCHAR (10)
DefaultOriginateTimeColumnName	制单时间	后缀为 “_ORIGINATETIME”	DATE
DefaultInputerColumnName	录入人	后缀为 “_INPUTER”	VARCHAR (10)
DefaultDepartmentColumnName	录入人部门	后缀为 “_DEPARTMENT”	VARCHAR (10)
DefaultPositionColumnName	录入人岗位	后缀为 “_POSITION”	VARCHAR (10)
DefaultInputTimeColumnName	录入时间	后缀为 “_INPUTTIME”	DATE
DefaultInputerAddressColumnName	录入人地址	后缀为 “_INPUTERIP”	VARCHAR (20)
DefaultDisabledColumnName	禁用	后缀为 “_DISABLED”	NUMERIC (1)
DefaultEnumTagColumnName	枚举/布尔标志	后缀为 “_FG”	枚举: NUMERIC (2)

			布尔: NUMERIC(1)
DefaultCodeColumnName	代码	后缀为 “CODE”	VARCHAR()
DefaultNameColumnName	名称	后缀为 “NAME”	VARCHAR()
DefaultChineseNameColumnName	中文名	后缀为 “CHNNAME”	VARCHAR()
DefaultEnglishNameColumnName	英文名	后缀为 “ENGNAME”	VARCHAR()
DefaultDateNotContainTimeColumnName	日期(不包含时间部分)	后缀为 “DATE”	DATE
DefaultDateIsYearMonthColumnName	年月(当月的头天零时)	后缀为 “YEARMONTH”	DATE
DefaultImageColumnName	图片	后缀为 “IMAGE”	LONG RAW

注意：如果不希望被 Phenix 约束，可以将 `FieldAttribute.AllowReservedColumn = false`，因为缺省情况下是 `true` 的。