



ENTERPRISE ARCHITECT

用户指南系列

信息工程

Author: Sparx Systems

Date: 13/11/2024

Version: 17.0

创建于  **ENTERPRISE
ARCHITECT**

目录

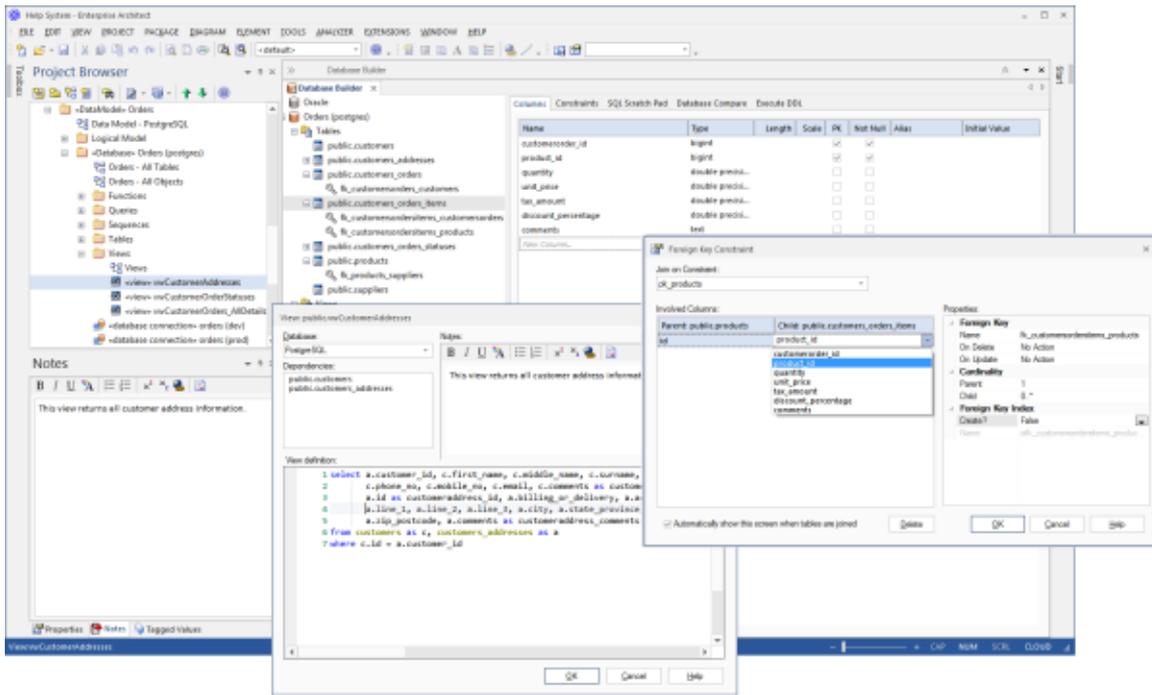
信息工程	4
开始	5
示例图表	7
使用数据模型类型	8
概念数据模型	9
实体关系图 (图表)	10
逻辑数据模型	14
物理数据模型	15
DDL变换	17
创建和管理数据模型	21
从模型模式数据模型	22
创建一个数据模型图表	24
示例数据模型图表	26
数据库生成器	27
打开数据库生成器	29
在数据库生成器中工作	31
列	35
创建数据库库表列	36
删除数据库库表列	38
重新排序数据库库表列	39
约束/索引	40
数据库约束库表/索引	41
主键	44
数据库索引	47
独特约束	50
外键	51
选择约束	54
库表触发器	56
SQL便签本	58
数据库比较	60
执行 DDL	66
数据库对象	69
数据库表	70
创建数据库库表	71
数据库库表栏目	73
创建数据库库表列	74
删除数据库库表列	76
重新排序数据库库表列	77
使用库数据库库表属性	78
设置数据库类型	79
设置数据库库表拥有着/架构	80
设置MySQL选项	81
设置 Oracle数据库库表属性	82
数据库约束库表/索引	83
主键	86
非聚集主键	89
数据库索引	90

独特约束	93
外键	94
选择约束	97
库表触发器	99
数据库视图	101
数据库过程	103
函数数据库	106
数据库序列	109
#数据库#	112
查询	112
创建操作容器	114
甲骨文包	116
数据库连接	117
管理 DBMS 选项	120
数据类型	121
DBMS 产品之间的映射数据类型	122
DBMS 产品转换为一个包	123
库表的数据类型转换	124
数据库数据类型	125
MySQL数据类型	127
Oracle数据类型	128
数据建模设置	129
数据建模Notations	130
DDL名称模板	134
导入数据库架构	136
生成数据库定义语言 (DDL)	139
为对象生成DDL	140
编辑 DDL模板	143
DDL 模板语法	145
DDL模板	146
用于 DDL 生成的基本模板	147
用于更改 DDL 生成的基本模板	150
DDL 宏	151
元素字段宏	152
列字段宏	155
约束字段Macros	156
DDL函数宏	158
DDL属性宏	163
模板中的 DDL 选项	169
DDL 限制	172
导入DDL脚本	173
支持数据库管理系统	174
更多信息	175

信息工程

设计、创建和管理概念、逻辑和物理数据模型

基于模型的系统开发的强大之处在于能够可视化、分析和设计系统的各个方面。能够与系统的其他模型一起查看和管理信息和数据提供了极大的清晰度并减少了出错的机会。Enterprise Architect广泛支持数据建模学科，从概念模型中的信息表示一直到数据库对象的生成。无论您是从UML模型生成数据库对象，还是将遗留DBMS逆向工程到模型中进行分析，该工具特征都将节省时间和宝贵的项目资源。



此图显示了数据库生成器接口，包括 DDL 生成和外键对话框。

Enterprise Architect支持概念、逻辑和物理层的信息建模。使用许多标准特征，这些模型可以相互连接，提供可追溯性。逻辑和物理模型也可以使用完全可变换引擎自动生成。可以使用方便的逆向工程功能导入、分析和比较遗留系统。

在本主题中，您将学习如何使用包括数据库构建器在内的丰富特征工具集来设计、创建、管理和可视化数据，包括将数据模型反向和正向工程到实时数据库。

数据库工具可用于创建和维护物理数据模型，并可以连接到正在运行的DBMS，因此您可以导入、生成、比较和更改实时数据库。

开始

信息建模师、数据建模师和架构师负责创建跨越多个抽象级别的组织信息模型，从概念到逻辑和物理。概念模型与技术无关，可用于与业务人员和领域专家的讨论，允许表示、讨论和同意领域中的基本概念。逻辑模型对概念模型进行了详细说明，增加了更多细节和精度，但通常仍然是技术中立的，允许信息分析师讨论并就逻辑结构达成一致。物理模型将技术特定数据应用到模型中，并允许工程师讨论并商定技术决策，为生成目标环境（例如数据库管理系统）做准备。

选择蓝图

Enterprise Architect将工具的大量特征划分为蓝图，确保您可以聚焦于特定任务并使用所需的工具，而不会受到其他特征的干扰。要使用数据建模特征您首先需要选择以下蓝图之一：

 <透视图名称> >数据库工程 >数据库工程

 <透视图名称> >数据库工程 >实体关系

设置蓝图可确保数据库工程图、其工具箱和蓝图的其他特征默认可用。

示例图表

示例图提供了对该主题的可视化介绍，并允许您查看在指定或描述数据模型定义方式时创建的一些重要元素和连接器，包括：表、视图、过程、序列、函数。

数据模型类型

可以从通常由业务人员创建或为业务人员创建的概念模型、业务和系统分析师使用的逻辑模型以及技术人员关心的物理模型（例如数据库工程师。在本主题中，您将学习如何管理所有三个级别的信息模型。

创建和管理数据模型

在本主题中，您将学习如何使用Enterprise Architect细节您的物理数据库模式。这包括使用数据库工具，它允许您通过 ODBC 连接与任意数量的实时数据库进行交互。

导入数据库架构

本主题将连接到实时数据库，包括生产，并将数据库逆向工程为模型创建表、视图、校正参考和过程测试，并展示您如何进行测试。自动创建数据库图，库表等元素可以与模型中A其他元素相关联，包括概念和逻辑模型、编程类测试等。

生成数据库定义语言（DDL）

在本主题中，您将学习如何通过直接从模型生成数据库定义语言代码来利用数据模型的强大功能。Enterprise Architect可以将代码生成到范围广泛的数据库管理系统中。

支持数据库管理系统

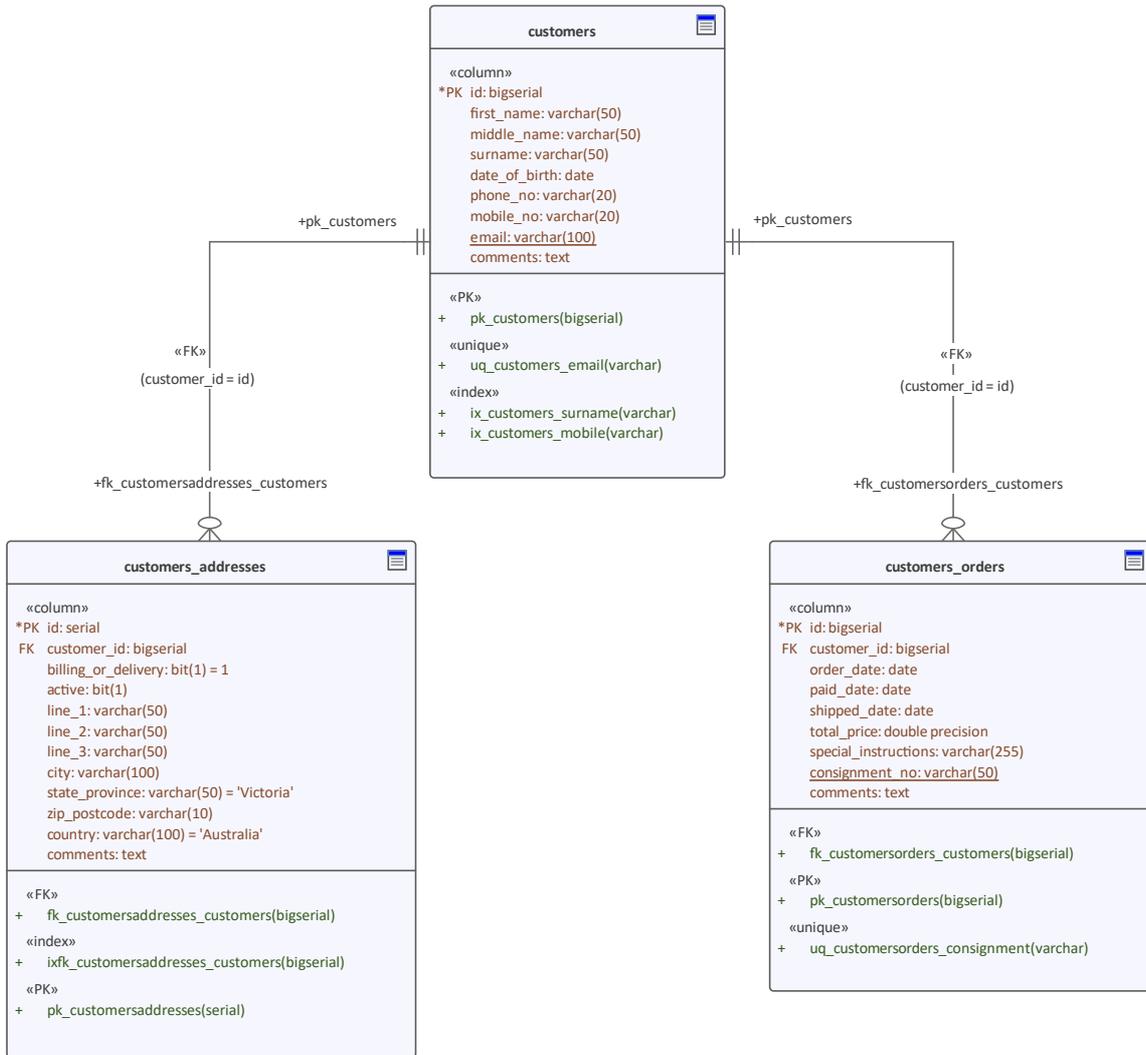
Enterprise Architect对大多数主流数据库管理系统 (DBMS) 都有丰富的支持。此特征允许使用导入特征来比较来自不同系统的模型以进行代码生成或分析。本主题列出了支持的 DBMS 和

更多信息

本部分提供了指向其他主题和资源的有用链接，您在使用数据建模工具特征时可能会发现它们很有用。

示例图表

使用Enterprise Architect的数据库工程特征，您可以创建丰富的对象模型，这些模型构成了从概念到逻辑到物理的任何抽象级别的数据模型。这些模型是通过从工具箱中添加库表和其他数据库对象或通过逆向工程和现有数据库来自一系列 RDBMS 的模型来创建的。数据库图A包含表、视图、过程、序列和函数。库表列注释为主键，外键使用专门的关联关系建模。在此示例中，用户创建了客户及其地址和订单的简单物理数据模型。



物理数据模型显示带有列和主键和外键的表。

使用数据模型类型

Enterprise Architect提供了许多特征来协助创建信息模型的过程，包括开发概念、逻辑和物理模型的能力，以及能够跟踪模型之间的基本概念的能力。可以为各种数据库系统开发物理模型，并且正向和逆向工程允许这些模型与实时数据库同步。

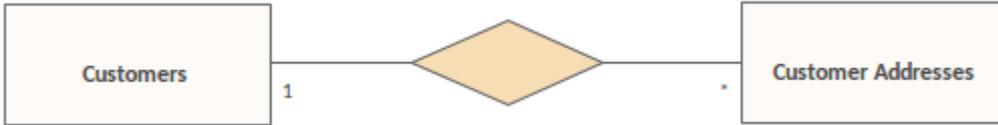
数据模型

类型	描述
概念数据模型	<p>概念数据模型，也称为领域模型，建立给定领域的基本概念和语义，并帮助将这些信息传达给广大利益相关者。</p> <p>概念模型也是项目分析阶段的常用词汇；它们可以在Enterprise Architect中使用实体关系或UML类模型创建。</p>
逻辑数据模型	<p>逻辑数据模型为概念模型元素添加更多细节并细化域的结构；它们可以使用实体关系或UML类模型来定义。</p> <p>逻辑数据模型的一个好处是它提供了物理模型和后续数据库实施的基础。</p> <p>实体-关系建模是一种抽象和概念化的数据库建模方法，用于生成例如关系数据库及其需求的模式或语义数据模型，在实体-关系图表（图表）中可视化。</p> <p>ERD 可帮助您构建概念数据模型，直至为目标 DBMS 生成数据定义语言 (DDL)。</p> <p>可以使用变换A逻辑模型转换为物理数据模型。</p>
物理数据模型	<p>Enterprise Architect中的物理数据模型帮助您可视化您的数据库结构并自动导出相应的数据库模式；您专门为此目的使用Enterprise Architect的UML配置文件进行数据建模。</p> <p>该概要文件提供了UML标准的有用扩展，将表和关系的数据库概念映射到类和关联的UML概念上；您还可以模型数据库列、键、约束、索引、触发器、参照完整性和其他关系数据库特征。</p> <p>由于Enterprise Architect帮助您可视化同一存储库中的每种类型的数据模型，您可以轻松管理每个抽象级别之间的依赖关系，以最大限度地提高可追溯性并验证系统实施的完整性。</p>

概念数据模型

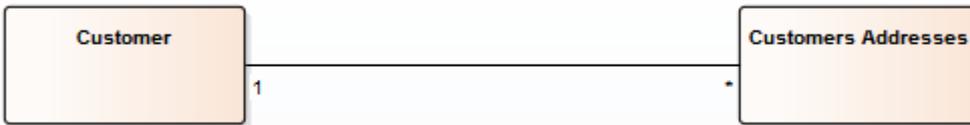
概念数据模型是数据模型A最抽象形式。由于其简单性，它有助于将想法传达给广泛的利益相关者。因此，特定于平台的信息，例如数据类型、索引和键，从概念数据模型中其它。其他实现细节，例如过程和接口定义，也被排除在外。

这是概念数据模型的示例，使用Enterprise Architect支持的两种表示法呈现。



显示一对多关系的实体关系图

使用实体关系 (ER) 表示法，我们将数据概念 “客户”和 “客户地址”表示为具有一对多关系的实体。我们可以使用UML类和关联来表示完全相同的语义信息。



显示相同的一对多关系的Unified Modeling Language图

是否使用UML或 ER 表示法来表示项目中的数据概念取决于相关利益相关者的经验和偏好。概念数据模型中说明的数据概念的详细结构由逻辑数据模型定义。

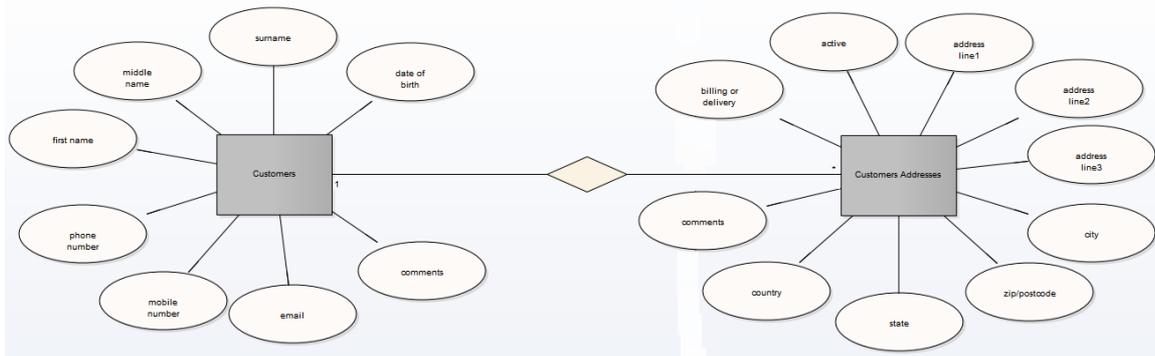
实体关系图 (图表)

根据在线维基百科：

实体关系模型 (ERM) 是数据的抽象和概念表示。实体-关系建模是一种数据库建模方法，用于以自上而下的方式生成系统 (通常是关系数据库) 及其需求的一种概念模式或语义数据模型。由该过程创建的图表称为实体-关系图、图表或图表。

Enterprise Architect图表的实体关系图

Enterprise Architect中的实体关系图基于 Chen 的 ERD 构建块：实体 (库表) 表示为矩形，属性 (列) 表示为椭圆 (连接到它们的实体)，关系之间表示为菱形连接器。



Enterprise Architect中的 ERD 技术可在从构建概念数据模型到为目标 DBMS 生成数据定义语言 (DDL) 的每个阶段提供帮助。

ERD 和 ERD 转换

Enterprise Architect使您能够通过使用与Enterprise Architect安装程序集成的MDG 技术快速简单地开发实体关系图。

实体关系图功能以以下形式提供：

- 实体关系图类型，通过“新图表”对话框访问
- 工具箱图表的一个实体关系图表页面
- “工具箱快捷方”菜单和快速链接器中的实体关系元素和关系条目

Enterprise Architect还提供转换模板，将实体关系图转换为数据建模图，反之亦然。

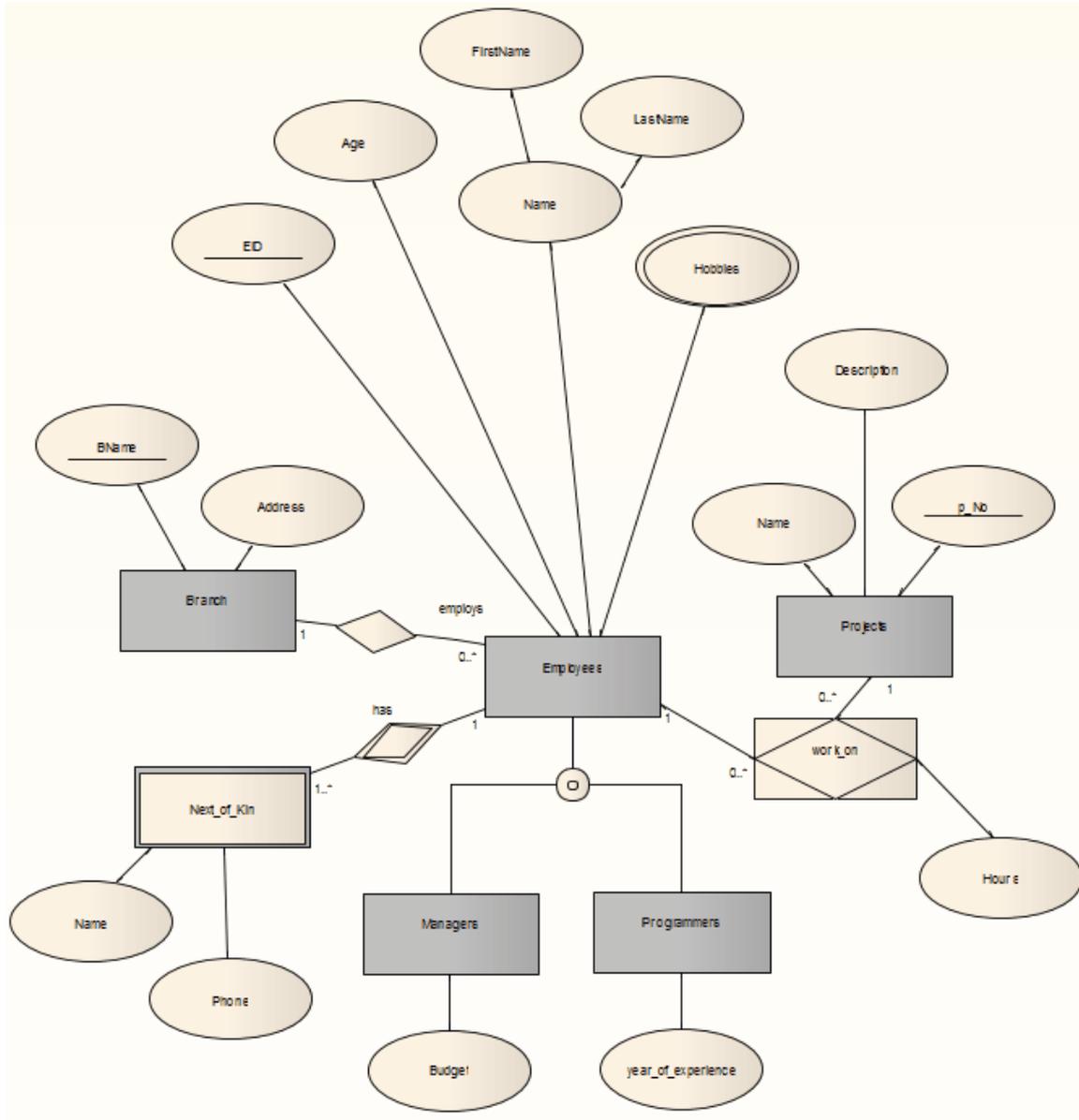
实体关系图表工具箱

您可以通过在工具箱的“查找图表图表”对话框实体指定“工具箱关系图”来访问“实体关系图表”工具箱

- ▾ ERD
 - Entity
 - Attribute
 - N-ary Association
- ▾ ERD - Relationships
 - Connector
 - Relationship
 - Disjoint
 - Overlapping

- 实体是唯一可识别的object或概念；关系连接器的 SourceRole 和属性定义中的 Multiplicity“属性可用 定义参与此关系的实体的基数
- 属性是实体或关系类型的属性
- N-ary关联表示一元（多对多递归）或三元关系，也可用于表示实体之间具有属性的关系；N-ary关联元素应始终位于连接器的目标端
- 连接器是实体和属性之间以及两个属性之间的连接器
- 关系是菱形连接器，表示实体之间有意义的关联
- Disjoint 和 Overlapping 表示超类实体和子类实体之间的关系

A的实体关系图



标记值

标注标记值可以修改一些实体关系图的组成部分，如下所示：

部件	标记值/笔记
实体	isWeakEntity 笔记：如果为真，则该实体为弱实体。
属性	属性类型 笔记：有四个有效选项：'normal'、'primary key'、'multi-valued'和'derived'
属性	通用数据类型 笔记：定义每个属性的通用数据类型。

属性	<p>dbmsDataType</p> <p>注记：为每个属性定义自定义的 DBMS 数据类型。此选项仅在 <i>commonDataType</i> 标签设置为 <i>na</i> 时可用。</p> <p>您必须首先通过 设置 > 参考 > 设置 > 数据库数据类型 功能区选项定义自定义类型。</p>
N-ary 关联	<p>是递归的</p> <p>注记：如果为真，N-ary 关联表示多对多递归关系。</p> <p>对于一对多和一对一的递归关系，我们建议使用普通关系连接器。</p> <p>有时您可能想要限制菱形关系连接器的伸展；只需选择一个关系连接器，右键单击以显示上下文菜单，然后选择 在光标处弯曲线 选项。</p>
关系	<p>弱</p> <p>注记：若为真，则关系为弱关系。</p>
脱离 重叠	<p>参与</p> <p>注记：有两个有效的选项，'partial' 和 'total'。</p>

注记

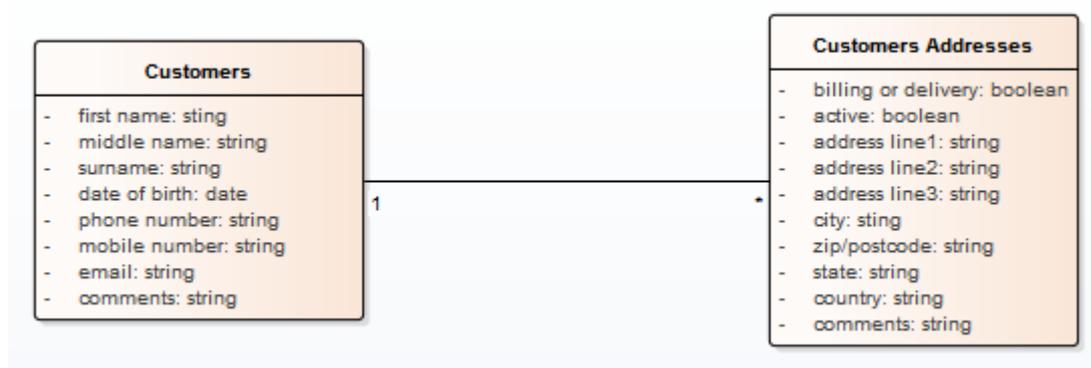
- Enterprise Architect 的企业版、统一版和终极版支持实体关系图

逻辑数据模型

逻辑数据模型有助于定义系统中数据元素的详细结构以及数据元素之间的关系。它们细化了概念数据模型引入的数据元素，并形成了物理数据模型的基础。在Enterprise Architect中，逻辑数据模型通常使用UML类表示法表示。

示例

此图是逻辑数据模型的简单示例。逻辑模型向概念模型添加了细节，但没有达到指定将使用的数据库管理系统的级别。



概念数据模型与库表建模客户及其地址。

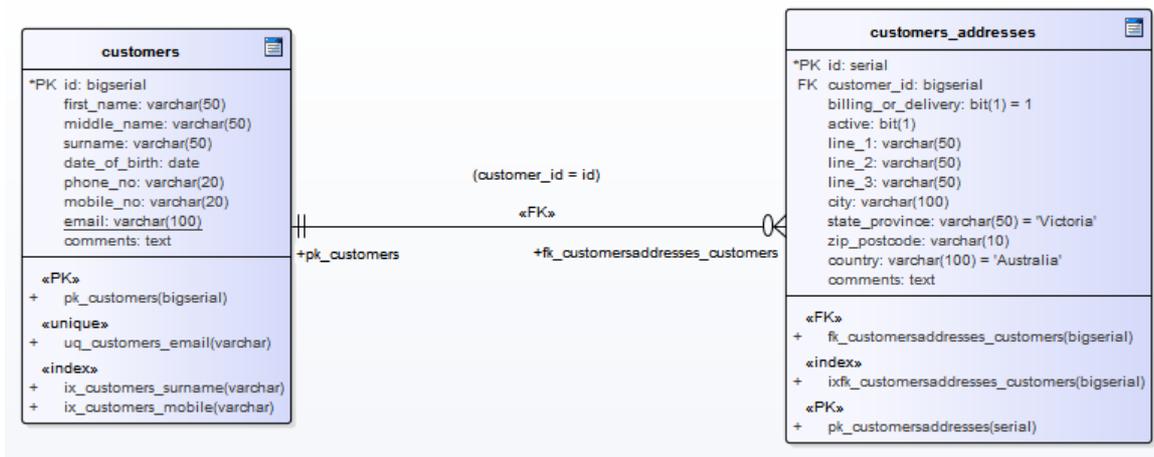
注册数据元素“客户”和“客户地址”包含UML属性；名称和通用数据类型保持平台独立。与特定DBMS实现相关的平台特定数据类型和其他元数据由物理数据模型定义。

物理数据模型

物理数据模型直观地表示由关系数据库模式实现A数据结构。除了提供数据库结构的可视化抽象之外，定义物理数据模型的一个重要好处是您可以从模型中自动导出数据库模式。这是可能的，因为物理数据模型捕获的元数据的丰富性及其与数据库模式的各个方面的紧密映射，例如数据库表、列、主键和外键。

示例数据模型

此示例显示了一个物理数据模型，可用于自动生成数据库模式。每个库表由一个UML类表示；库表列、主键和外键使用UML属性和操作建模。该模型演示了信息工程连接器样式的使用。



符号

示例模型是使用Enterprise Architect的用于数据建模的UML配置文件定义的；表之间的关系使用默认的信息工程符号。

信息工程是Enterprise Architect支持帮助数据建模者识别关系中的基数的三种符号之一。您可以通过选择 设计>图表>管理>属性”功能区选项，单击 连接器”页面并在 连接器符号”下拉列表中选择所需的选项来更改符号。您很可能会将符号更改为 IDEFX1，但 UML2. 1符号也可用。

默认数据库管理系统

在创建物理数据模型之前，建议您为项目设置默认 DBMS。设置默认 DBMS 可确保在图表上创建的所有新数据库元素都自动分配默认 DBMS。

如果未设置默认 DBMS，则会在未分配 DBMS 的情况下创建新表，这会限制Enterprise Architect正确模型物理对象的能力。例如，Enterprise Architect无法确定列的正确数据类型列表。

您可以使用以下方法设置默认 DBMS 类型：

- '开始>外观>首选项>首选项>源代码工程>代码编辑器'，或
- '设置>参考>设置>数据库数据类型或
- '开发>数据建模>数据类型或
- 代码生成工具栏中的第二个数据输入字段

注记：通过数据库生成器建模时，默认的 DBMS 是在模型级别定义的（作为 <<Database>>包的标记值'DBMS'）而不是在项目级别，从而在项目涉及时允许更大的灵活性多个 DBMS。

DDL变换

DDL 转换将逻辑模型转换为结构化的数据模型，以符合支持的 DBMS 之一。目标数据库类型由模型中设置为默认数据库的 DBMS 确定（参见数据库数据类型帮助主题， 设置为默认值”选项）。然后可以使用数据模型自动生成运行语句以在系统支持的数据库产品之一中运行。

DDL 转换使用并演示了中间语言对许多特定于数据库的概念的支持。

概念

概念	影响
库表	将一对一映射到类元素上。 转换支持 “多对多”关系，创建汇合库表。
柱子	一对一映射到属性。
首要的关键	列出所有涉及的列，以便它们存在于类中，并为它们创建主键方法。
外键	A特殊的连接器，其中源和目标部分列出了所有涉及的列，以便： <ul style="list-style-type: none"> • 列存在 • 目标类中A匹配的主键，并且 • 转换创建适当的外键

MDG 技术自定义默认映射

针对新的、用户定义的 DBMS 的 DDL 转换需要MDG 技术来将 PIM 数据类型映射到新的目标 DBMS。

为此，创建一个名为 “UserDBMS Types.xml”的MDG 技术.xml 文件，将 UserDBMS 替换为添加的 DBMS 的名称。将文件放在 EA\MDGTechnologies 文件夹中。MDG 技术文件的内容应具有以下结构：

```
< MDG .技术版本1 “1.0”>
<Documentation id="UserdataTypes" name="Userdata Types" version=" 1 .0" notes="类型for UserDBMS"/>
<代码模块>
<CodeModule 语言="用户数据" 注释="">
<代码选项>
<CodeOption name="DBTypeMapping-bigint">BIGINT</CodeOption>
<CodeOption name="DBTypeMapping-blob">BLOB</CodeOption>
<CodeOption name="DBTypeMapping-boolean">TINYINT</CodeOption>
<CodeOption name="DBTypeMapping-text">CLOB</CodeOption>
...
</代码选项>
</代码模块>
</代码模块>
```

</ MDG .技术>

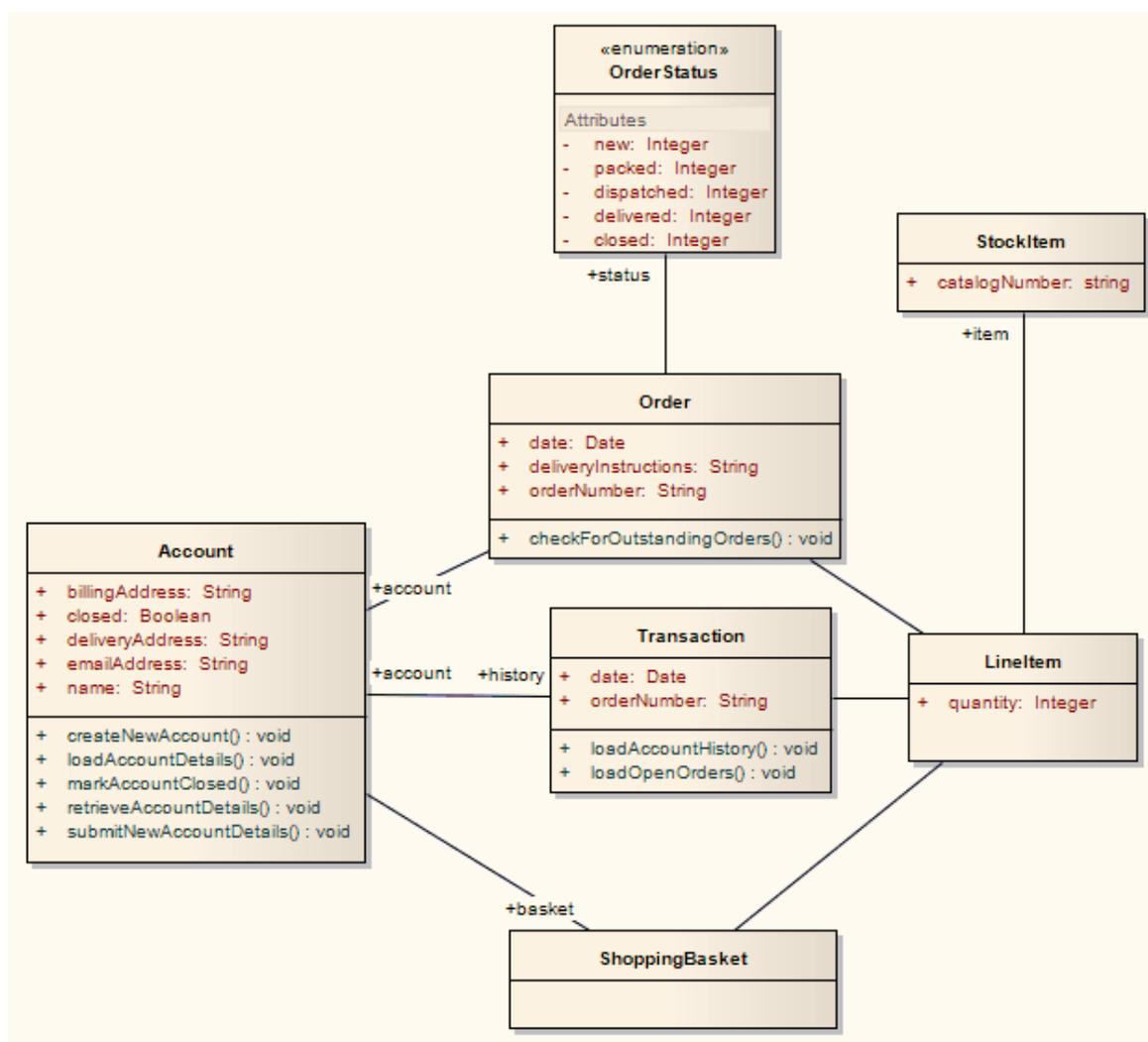
例如，“text”是映射到新 DBMS 的“CLOB”数据类型的公共类型（如 数据库数据类型”对话框中所列）。

注记

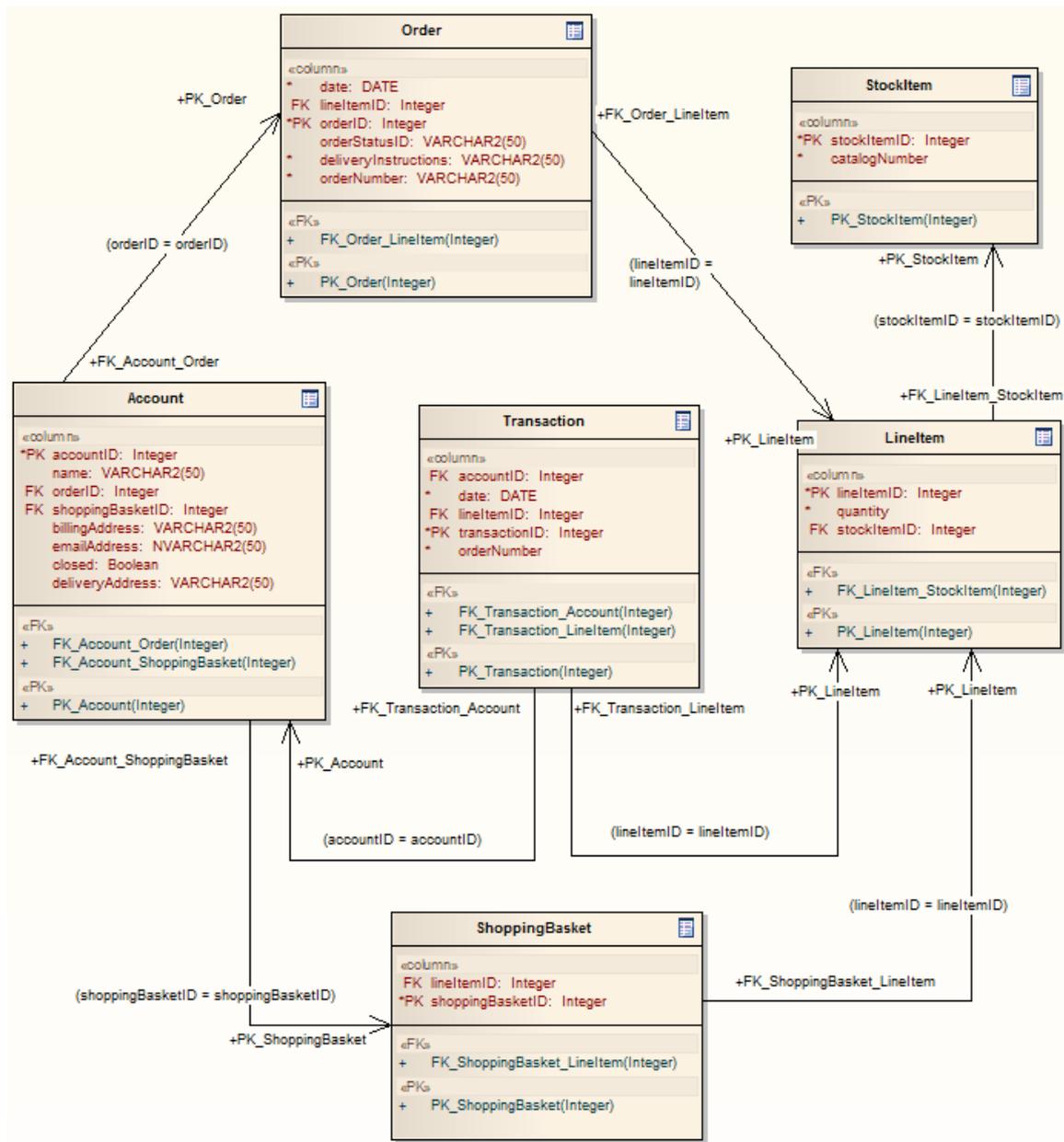
- 您可以在转换后定义逻辑模型中未描述的 DBMS 特定方面，例如存储过程、触发器、视图和选择约束；查看 *Physical Data* 模型帮助

示例

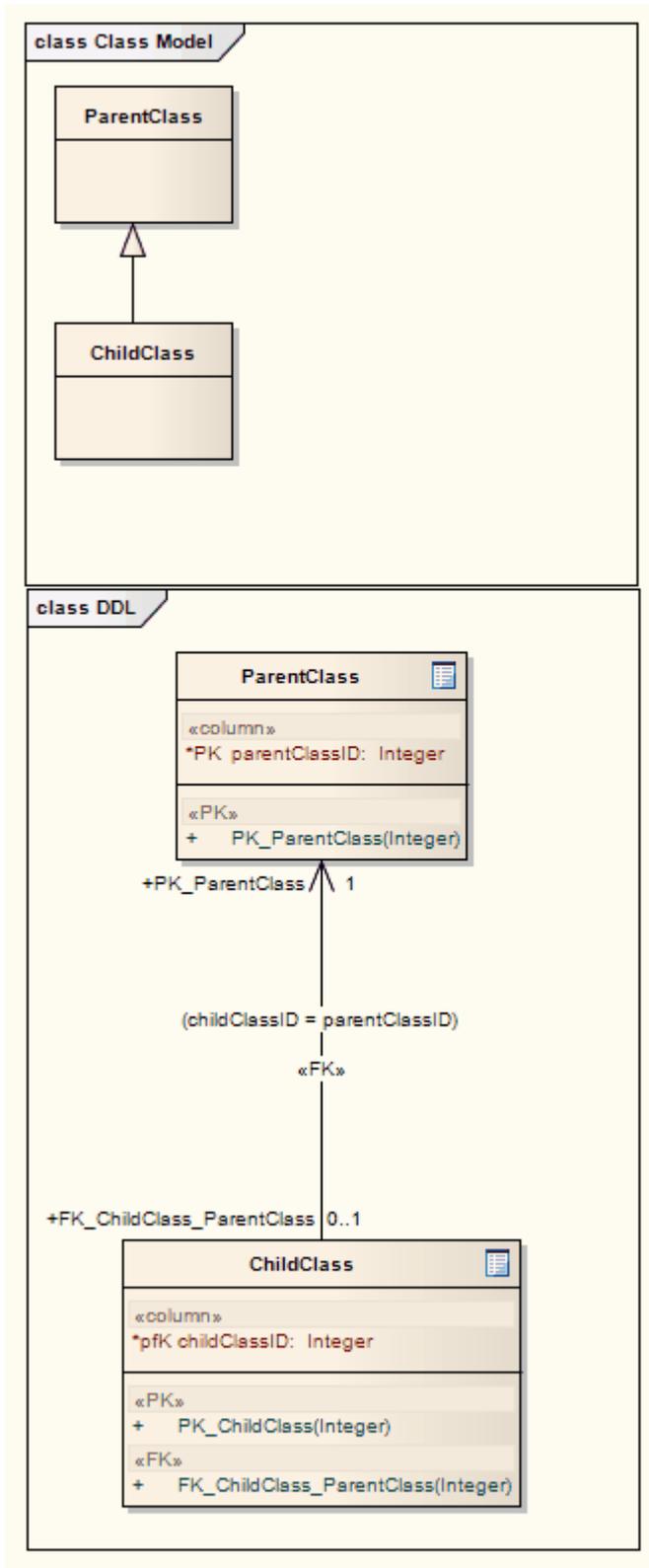
PIM 元素



改造后成为PSM元素



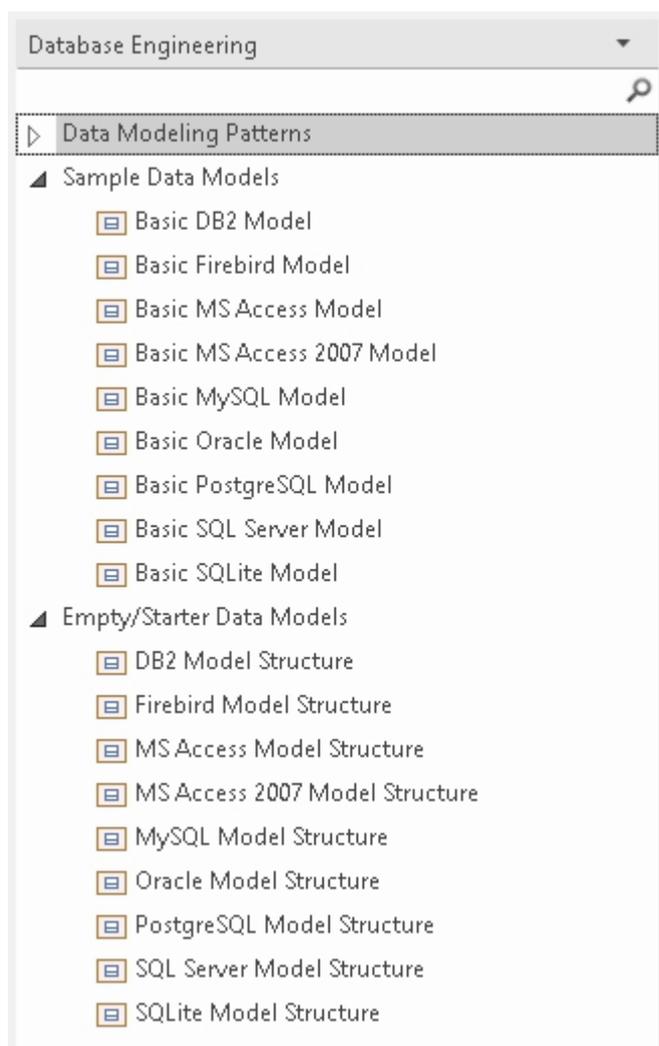
泛化是通过为子元素提供父元素的外键来处理的，如图所示。不支持向下复制继承。



创建和管理数据模型

Enterprise Architect是一个功能齐全的数据库建模平台，使用户能够在所有阶段使用他们的物理数据模型，从设计到实时数据库的实施，适用于各种数据库管理系统，例如 Microsoft #

服务器、甲骨文、PostgreSQL 和MySQL。



此图显示了用于一系列 RDBMS 的数据库设计的启动模型向导模式。

从模型模式数据模型

创建数据建模工作空间的最简单方法是使用预定义的数据库模型模式，可通过模型构建器获得。Enterprise Architect为系统支持的每个DBMS 提供了一个模式。

访问

使用此处概述的任何方法显示模型构建器。

在模型构建器对话框中，选择“数据库工程”蓝图。

功能区	开始> 个人>模型构建器 设计>包>模型生成器
上下文菜单	右键点击包 模型 (模式库)
键盘快捷键	Ctrl+Shift+M
其它	浏览器窗口标题栏菜单 模型 (图案库)

创建数据模型

字段/按钮	行动
添加到包	显示所选根包的名称。
技术	点击“数据库”。
名称	如有必要，展开模式的数据库工程组。 单击您在模型中支持的每个数据库管理系统的复选框。
全部	单击此按钮可选择所有数据库工程模型类型和实体关系图的复选框，以将它们全部包含在模型中。
没有任何	单击此按钮可清除所有选中的复选框，以便您可以单独重新选择某些复选框。
确定	单击此按钮可将您正在建模的每个数据库管理系统的包和图表添加到浏览器窗口。

每个数据建模模式提供什么

- A模型图
- 报表工件元素) (A汇总图上快速记录数据模型)

- 每个逻辑模型和物理模型的A包
- 在物理模型包中，预定义的子包层次结构，一个用于被建模的 DBMS 支持的每种object类型（例如表、视图、过程和函数）；这些会在添加数据库对象时自动组织它们
- 工作区的 DBMS 类型
- A所有者
- A包中的数据建模图，连接符设置为 IDEF1X

注记

- 创建数据建模工作区后，您可以通过以下两种方式之一开始开发您的模型：
 - 通过数据库构建器，这是一个支持数据库建模器的专用视图
 - 通过浏览器窗口和图表，这是可能适合用户的传统方法经验丰富的UML建模人员

创建一个数据模型图表

要对关系模型的结构进行建模，您可以使用数据建模图，它是扩展类图。当您打开数据建模图时，工具箱图表其中包含图表元素：

- 库表
- 视图
- 过程
- 序列
- 函数
- 关联和
- 连接数据库

访问

使用此处列出的任何方法显示“新图表”对话框。

功能区	设计>图表> 添加图表
上下文菜单	右键单击包 添加图表 右键单击元素 添加 添加图表
键盘快捷键	Ctrl+插入
其它	浏览器窗口标题栏菜单 新图表

创建数据建模图

字段/按钮	行动
包	默认为在浏览器窗口中选择的包的名称，或者，如果父项是元素，则为包含该元素的包的名称。 如果您将图表直接添加到包中并注意到它不是正确的包，请单击  按钮并浏览正确的包。
家长	如果要将图表添加到元素，则此字段会显示元素名称。
图表	该字段默认为父包或元素的名称。 如果需要，请使用您的首选名称改写默认名称。
选择自	单击此标题并选择最适合您工作区域的蓝图组和蓝图或工作空间（在本例中，“信息工程 >数据库模型”）。 从面板中列出的选项中，单击“扩展”。
图表类型	点击“数据建模”。

确定	<p>单击此按钮以创建图表。</p> <p>图表视图显示空白图表，“数据建模模型”页面显示在图表工具箱中。</p> <p>将元素和连接器从工具箱拖到您的图表上，以创建您的数据模型。</p>
----	--

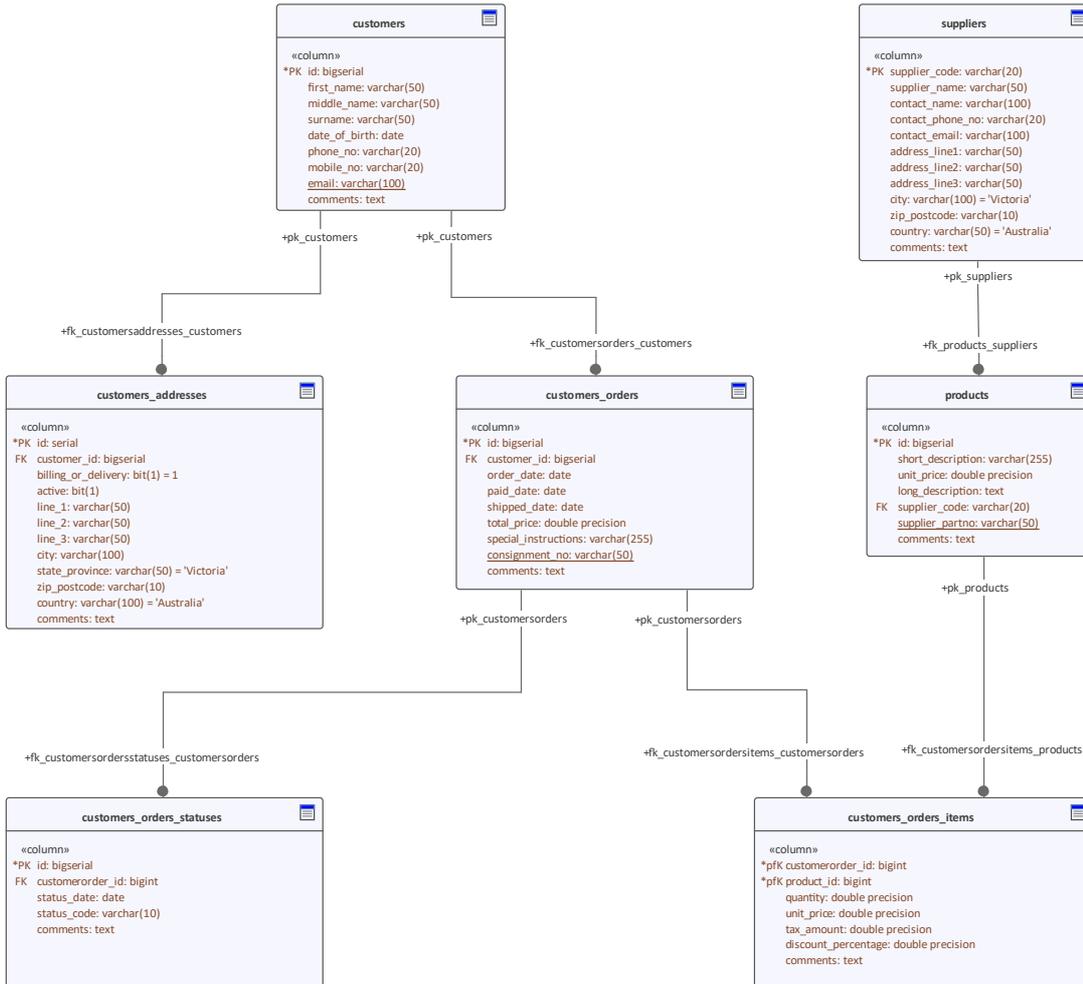
注记

- 所有新图表的默认图表连接器符号是信息工程，尽管许多数据建模者更喜欢符号 IDEF1X；要进行此更改，请选择 设计>图表>管理>属性>连接器”并单击 连接器符号”下拉列表中所需的选项

示例数据模型图表

此数据模型图示例显示了书店仓储系统的数据模型。库表是使用具有用于显示列名称和类型的列隔间的构造型类建模的。主键和外键由列上的构造型表示。您可以在示例模型中更细节地检查此模型，该模型与Enterprise Architect一起安装，可从此功能区位置获得。

开始>帮助>帮助> 打开示例模型



数据建模图，带有隐藏的操作隔间，显示连接以指示外键关系的库表。

数据库生成器

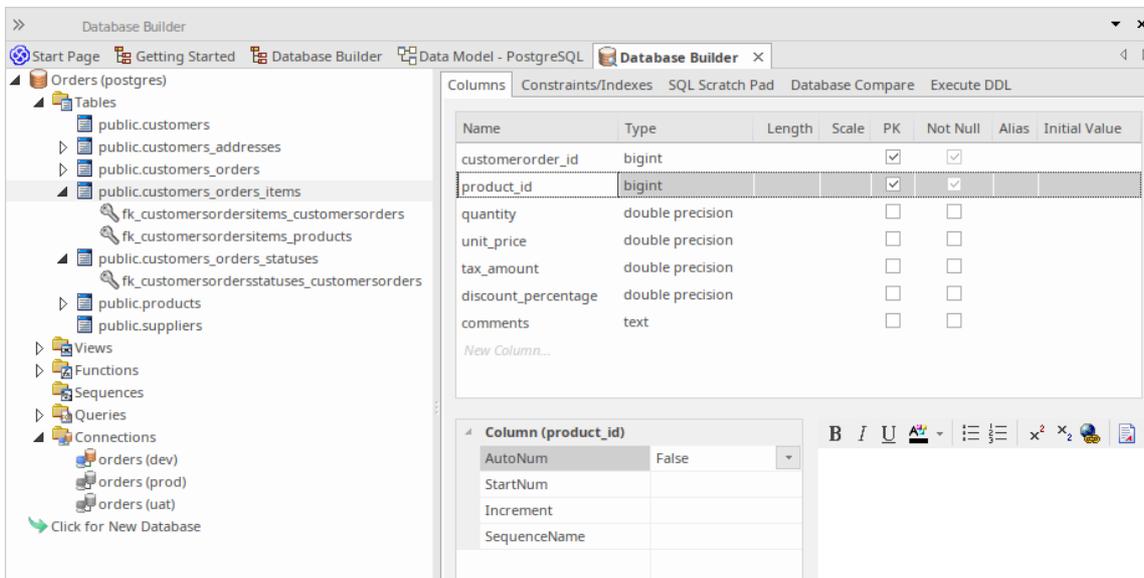


数据库是为数据建模器量身定制的界面；所有与数据库相关的建模任务都可以在一个位置执行。界面及其相关屏幕仅包括与数据建模相关的信息，从而简化和简化建模过程。

访问

功能区	开发 > 数据建模 > 数据库构建器
-----	--------------------

数据库生成器



此图显示了加载了“Orders (postgres)”数据模型的数据库构建器，它出现在示例模型中。

概述

数据库生成器的接口包括：

- 数据A树，列出当前存储库中所有已定义的数据模型
- A “列” 您可以通过它直接管理库表列的选项卡
- A ‘约束/索引’ 用于直接管理主键、外键和索引等库表约束的选项卡
- 可用于运行# SQL便签本# 针对实时数据库的查询
- A数据库比较”选项卡，显示数据模型和实时数据库之间的比较结果

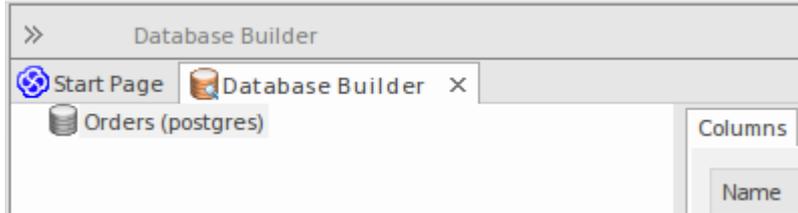
- 一个“执行 DDL”选项卡，您可以在其中立即针对实时数据库执行生成的 DDL
- 您可以使用数据库来：
- 创建、编辑和删除数据库对象（表、视图、过程、序列和函数）
 - 创建、编辑和删除库表约束（主键、索引、唯一约束、选择约束和触发器）
 - 创建、编辑和删除库表外键
 - 逆向工程数据库架构信息
 - 从建模数据库生成 DDL
 - 将实时数据库模式与建模数据库进行比较
 - 针对实时数据库执行生成的 DDL
 - 执行即席#
针对实时数据库的语句

注记

- Enterprise Architect 的企业统一版和终极版中提供了数据库构建器

打开数据库生成器

首次打开数据库生成器时，它会搜索整个项目，查找所有具有构造型 <<Data Model>> 的包，并将相应的数据模型作为根加载到树中。灰色A表示未加载数据模型的详细信息。



该图显示了具有单个卸载数据模型（称为“订单（postgres）”）的数据库构建器。

使用数据库生成器

您可以通过以下两种方式之一开始在数据库生成器中工作：

任务	行动
创建新数据模型	打开数据库生成器视图后，右键单击树的空白处并选择“新建数据模型”以调用模型生成器。
加载现有数据模型	打开数据库构建器视图后，通过以下任一方式加载任何定义的数据模型： <ul style="list-style-type: none"> • 右键单击名称并选择“加载”，或 • 双击名称

数据属性模型

在早期版本的Enterprise Architect中（在引入数据库生成器之前），数据建模器有必要在允许某些任务之前手动设置属性对象。例如，Enterprise Architect不允许在未首先为库表分配DBMS的情况下定义库表列。这是因为DBMS控制可用数据类型的列表。

为了提高效率和用户体验，数据库构建器在数据模型级别定义了许多属性的默认值，然后在创建新对象时自动应用这些默认值。

属性

选项	描述
数据库管理系统	定义对象：数据模型的<<Database>>包 定义为：标记值 详细信息：定义当前数据模型的DBMS 额外的信息： <ul style="list-style-type: none"> • 控制在数据库构建器树中为当前数据模型显示哪些逻辑文件夹

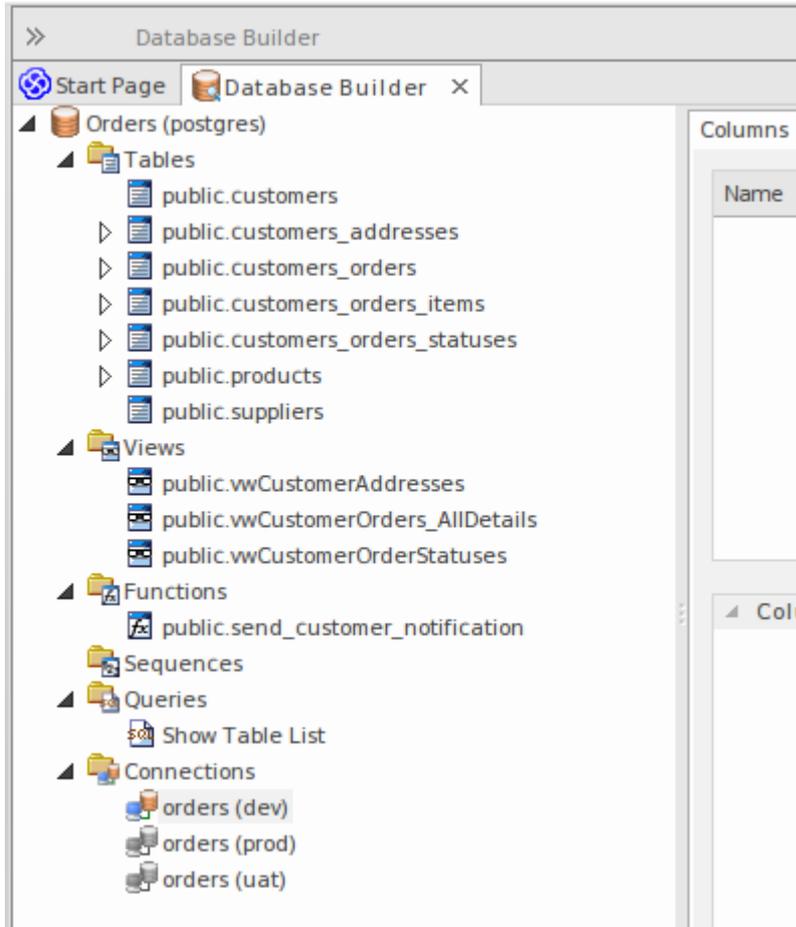
	<ul style="list-style-type: none">• 控制在数据库比较期间应用哪些 DBMS 规则• 自动分配给当前数据模型中创建的每个新数据库object
默认所有者	定义对象：数据模型的<<Database>>包 定义为：标记值 详细信息：定义当前数据模型的默认拥有者 额外的信息： <ul style="list-style-type: none">• 如果 DBMS 支持所有者/模式，则自动分配给当前数据模型中创建的每个新数据库object
默认连接	定义对象：数据模型的<<Database>>包 定义为：标记值 详细信息：(可选) 默认连接的名称 额外的信息： <ul style="list-style-type: none">• 每当加载数据模型时，都会检查 默认连接”属性；如果存在，该名称的连接将自动激活• 数据库工程模型模式没有为此属性定义值，只要用户将连接设置为默认值，就会创建或更新它

注记

- 如果打开数据库构建器时在浏览器窗口中选择一个数据模型，将自动加载该模型的详细信息

在数据库生成器中工作

加载数据模型后，数据库构建器会创建一组逻辑文件夹，每个文件夹对应当前 DBMS 支持的每种object类型。每个逻辑文件夹都填充有在数据模型的包层次结构中的所有该类型的对象（如浏览器窗口所示）。



在此图像中，数据模型'Orders (postgres)' 显示了用于表、视图、函数、序列、查询和连接的逻辑文件夹。值得注意的是，没有“过程”文件夹，因为 PostgreSQL 不支持数据库过程。

在数据库生成器树行动可用

大多数数据库生成器功能都可以通过上下文菜单访问。树中的每个object都有自己的组基于其类型和状态的独特菜单项。此表描述了可用的上下文菜单项并确定它们适用于哪些对象。

菜单选项	适用于/描述
新数据模型	适用于：空白空间 描述:打开模型生成器。
全部刷新	适用于：空白空间 描述：重新加载数据模型的完成列表。
加载	适用于：根 描述：加载数据模型的全部细节。

卸下	<p>适用于：根</p> <p>描述：卸载数据模型的全部细节。</p>
导入数据库架构	<p>适用于：Loaded根</p> <p>描述：使用当前活动连接作为实时数据库源打开“导入数据库模式”对话框。</p>
生成DDL	<p>适用于：已加载根、文件夹、库表、视图、过程、函数、序列、包</p> <p>描述：打开“生成DDL”对话框并选择当前object。</p>
显示差异	<p>适用于：加载根、文件夹、库表、视图、过程、函数、序列</p> <p>描述：将选定对象与当前活动连接进行比较。</p>
显示与选项的差异	<p>适用于：已加载根、文件夹、库表、视图、过程、函数、序列、包</p> <p>描述：将选定对象与当前活动连接进行比较，并根据指定的比较选项选择性地忽略一些差异。</p>
管理 DBMS 选项	<p>适用于：Loaded根</p> <p>描述：打开“管理 DBMS 选项”对话框，可用于更改分配的 DBMS 和拥有着多个对象。</p>
视图记录数	<p>适用于：库表、视图</p> <p>描述：构建并运行一个SELECT查询（格式化以适合元素的 DBMS）以显示所选库表或视图中的记录数。</p> <p>如果没有活动连接，系统会提示您选择一个。</p>
视图前 100 行	<p>适用于：库表、视图</p> <p>描述：构建并运行一个SELECT查询（格式适合元素的 DBMS）以显示所选库表或视图的前 100 行。</p> <p>如果没有活动连接，系统会提示您选择一个。</p>
视图前 1000 行	<p>适用于：库表、视图</p> <p>描述：构建并运行一个SELECT查询（格式适合元素的 DBMS）以显示所选库表或视图的前 1000 行。</p> <p>如果没有活动连接，系统会提示您选择一个。</p>
视图所有行	<p>适用于：库表、视图</p> <p>描述：构建并运行一个SELECT查询（格式化以适合元素的 DBMS）以显示所选库表或视图的所有行。</p> <p>如果没有活动连接，系统会提示您选择一个。</p>
属性	<p>适用于：已加载根、文件夹、库表、视图、过程、函数、序列、包、连接</p> <p>描述：打开选定object的标准“属性”对话框。</p>
在项目中查找浏览器	<p>适用于：已加载根、文件夹、库表、视图、过程、函数、序列、包、# 查询、连接</p> <p>描述：在浏览器窗口中查找选定object。</p>
刷新	<p>适用于：Loaded根</p> <p>描述：重新加载当前加载的数据模型的详细信息。当其他用户添加、更改或</p>

	删除对象或在数据库构建器之外执行更改时，这是必需的。
添加新的<type>	适用于：文件夹、库表、视图、过程、函数、序列、包、 # 查询、连接 描述：创建指定类型的新object。
克隆<名称>	适用于：文件夹、库表、视图、过程、函数、序列、包、 # 查询、连接 描述：制作所选object的新副本。当您选择此选项时，会显示一个提示，您可以在其中设置新object的名称和所有者。对于库表对象，您可以选择应该复制哪些现有约束（并为每个约束设置一个名称）以及应该复制哪些外键。对于基于 SQL 的对象，您可以对# 为新元素。
删除<名称>	适用于：库表、视图、过程、函数、序列、包、 # 查询、连接 描述：从存储库中永久删除选定object。
在 <name> 上添加新的外键	适用于：库表 描述：在选定的库表和另一个库表之间创建一个新关系，然后显示新关系的外键约束"屏幕。
# 物件属性	适用于：视图、过程、函数、序列 描述：打开'# 物件编辑器的屏幕。
编辑	适用于： # 查询 描述：加载# (在选定的元素中定义) 放入SQL便签本。
运行	适用于： # 查询 描述：加载# 在SQL便签本并运行它。 如果没有活动连接，系统会提示您选择一个。
设置为活动数据库连接	适用于：连接 描述：将选定的数据库连接标记为当前会话的活动连接。
设置为默认数据库连接	适用于：连接 描述：每次加载数据模型时，将选定的数据库连接标记为活动连接。
数据库连接属性	适用于：连接 描述：打开“数据库属性”屏幕，管理连接设置。

创建/编辑/删除数据库对象

本节列出的页面详细描述了细节使用数据库生成器的界面来创建和操作数据库表；但是，创建和操作基于 SQL 的数据库对象的过程记录在其他领域。有关详细信息，请参阅这些主题：

- [Database Views](#)
- [Database Procedures](#)
- [Database Functions](#)
- [Database Sequences](#)
- [Database Connections](#)

数据库构建器中的数据库连接

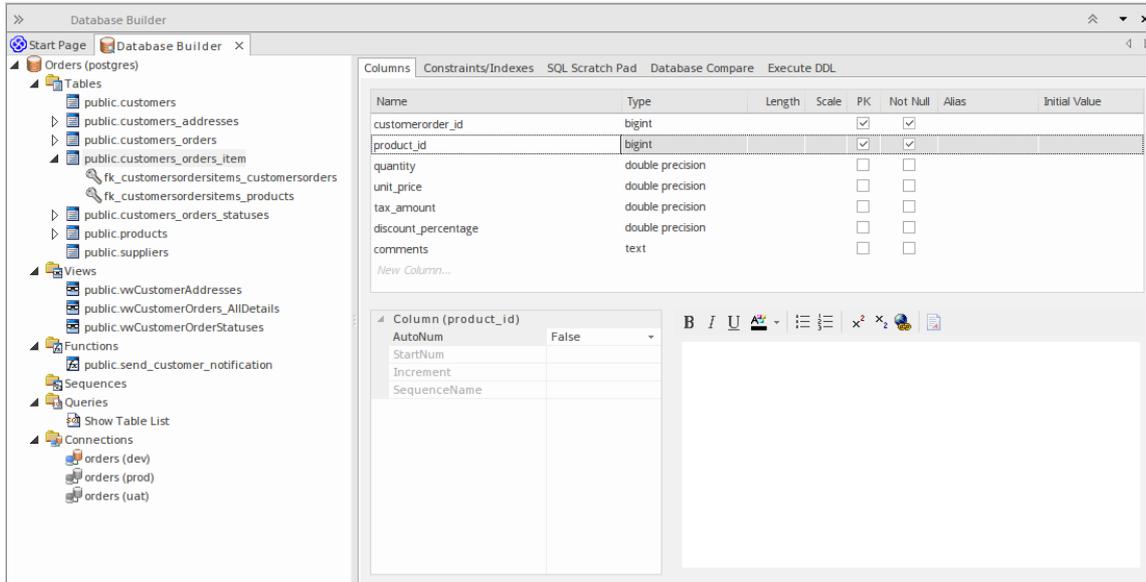
执行某些任务（例如“比较”或“执行 DDL”）时，数据库构建器需要一个活动的数据库连接。在给定时间，只能激活一个数据库连接（由彩色“数据库连接”图标指示，而其他数据库连接为灰色）。如果数据库连接当前未处于活动状态，并且您尝试执行需要连接的任务，则数据库构建器会根据定义的连接数执行以下操作之一：

- 0 连接 - 提示您创建连接，如果成功，则继续
- 1 连接 - 将其设置为活动并继续
- 2 个（或更多）连接——提示您选择一个，如果成功，继续

列

表是基本的数据库object，而列（及其属性）是最常修改的库表特征，由数据建模者更新和更改，因此“列”页面位于数据库构建器界面的第一页，非常方便。

一旦在数据库生成器的树中选择一个库表，“列”页面就会填充当前为该库表定义的列列表。然后，数据建模者可以直接在列表或网格中对主列属性进行更改。当数据建模者选择列表中的各个列时，列的扩展属性（和注释）立即显示在列表下方，允许修改这些扩展属性。



此图显示了显示对象树的数据库生成器界面和显示所选表的列的列选项卡。

注记

- 只有在数据库构建器的树中选择了库表项时，才会填充“列”页面

创建数据库库表列

数据库库表列A UML数据建模配置文件中表示为具有 <<column>> 构造型的属性。对于选定的库表，您可以在数据库构建器的“列”页面或“列和约束”屏幕上审阅现有列并创建新列。

您可以直接在“列”选项卡上的列列表中定义列详细信息。当您完成每个字段时，更改会自动保存。某些字段对您可以输入的数据有一定的限制，如此处所述。该选项卡还包含一个“属性”面板和一个“注记”字段，其中填充了所选列上的现有信息。您创建的每个新列都会自动分配一组默认值并添加到列表底部。

访问

功能区	开发 > 数据建模 > 数据库构建器 > 点击库表 > 列 > 右键 > 添加新列
上下文菜单	在图表中，右键单击所需的库表 特征 专栏 右击 添加新列
键盘快捷键	选择表 F9 Tab 键 (在“列”选项卡上设置输入聚焦) Ctrl+N

在库表中创建列

选项	行动
名称	用适当的列名文本改写默认名称。
类型	单击下拉箭头并为列选择适当的数据类型。 可用的数据类型取决于分配给父库表的 DBMS。
长度	(可选) 某些数据类型具有长度组件 - 例如，VARCHAR 具有定义可以存储的字符数的长度。如果数据类型没有长度组件，则禁用此字段。 如果该字段可用并且您需要定义多个字符，请在此处键入值。
规模	(可选) 某些数据类型具有比例组件 - 例如，DECIMAL 具有定义可以保留的小数位数的比例。如果数据类型没有比例组件，则禁用此字段。 如果该字段可用并且您需要定义比例，请在此处键入值。
PK	如果该列是此库表的主键的一部分，请选中该复选框。
不为空	如果该列禁止使用空值，请选中该复选框。 如果选中“PK”复选框，则该复选框被禁用。
别名	如果出于显示和文档目的需要，请输入该字段的替代名称。
初始值	如果需要，输入一个可用作此列默认值的值。
注记	类型在记录该列所需的任何附加信息中。 您可以使用字段顶部的注记工具栏来格式化文本。

列属性

库表的数据库管理系统的相应属性会自动显示在“属性”面板中（如果不可见，请展开“列（<名称>”分支）。

属性	数据库管理系统
自动编号（起始编号增量）	甲骨文 MySQL # 服务器 DB2 PostgreSQL 注记：如果您需要自动编号序列，请将此属性设置为True，并在必要时定义起始编号和增量。
生成	DB2 注记：将此附加属性设置为DB2中的自动编号，设置为“默认”或“始终”。
不代表	SQL服务器 注记：如果要阻止复制，请将此属性设置为True。
零填充	MySQL 注记：将此属性设置为True或False以指示字段是否填充为零。
未签名	MySQL 注记：将此属性设置为True或False以指示字段是否接受无符号数字。
长度类型	甲骨文 注记：设置该属性以定义字符语义为“None”、“Byte”或“Char”。

删除数据库库表列

对于选定的数据库库表，您可以在“列和约束”屏幕的“列”选项卡上审阅现有列并删除任何单个列。

访问

使用此处概述的方法之一来显示表格的列表，然后选择一列并将其删除。

当您选择“删除列<名称>”选项时，如果满足所有验证规则，则立即删除该列。

功能区	开发>数据建模>数据库生成器>点击库表>栏目>右键栏名>删除栏目<名称>
上下文菜单	在图表中，右键单击所需的库表 特征 专栏 右键单击列名 删除列<名称>
键盘快捷键	F9 使用'向上箭头'或'向下箭头'选择一列 Ctrl+D

注记

- 如果删除的数据库库表列涉及任何约束，它将自动从它们中删除

重新排序数据库库表列

如果您在数据库库表中定义了多个列，则可以更改它们的列出顺序。列表中的顺序是列在生成的 DDL 中出现的顺序。

访问

使用此处概述的方法之一来显示库表的列列表，然后选择一列并在列表中重新定位它。

功能区	开发 > 数据建模 > 数据库生成器 > 点击库表
上下文菜单	在图表中，右键单击所需的库表 特征 列
键盘快捷键	F9

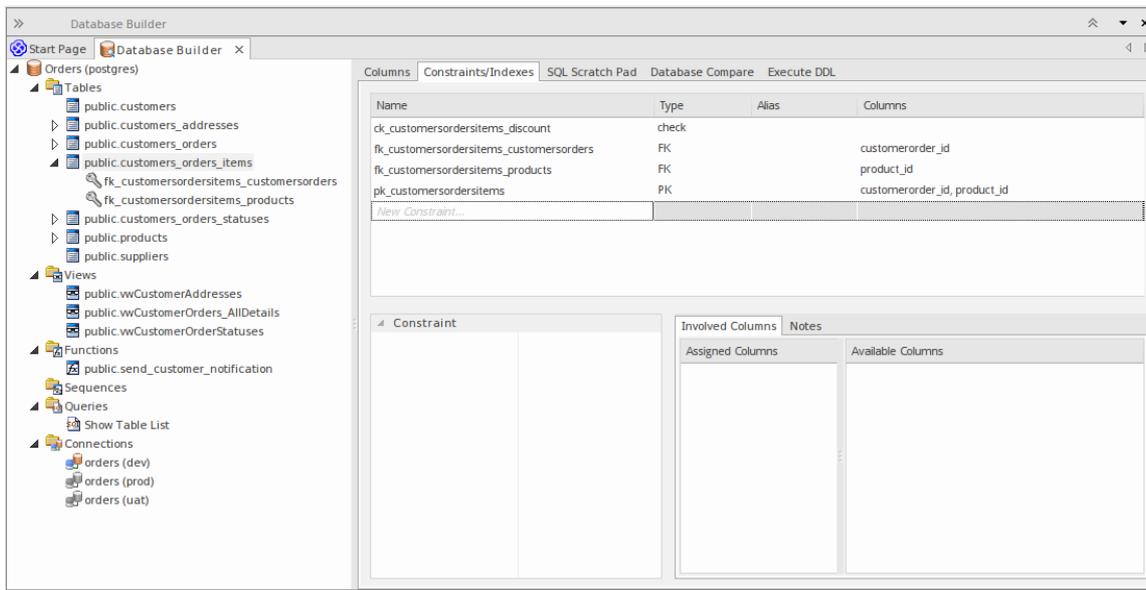
更改列顺序

节	行动
1	在“列”选项卡中，单击列表中所需的列名。
2	右键单击并选择： <ul style="list-style-type: none"> “上移列 <名称>”选项（或按 Ctrl+向上箭头）将列上移一位 “将列 <名称> 向下移动”选项（或按 Ctrl+向下箭头）将列向下移动一个位置 这些选项在“列”选项卡和图表中都会立即生效。

约束/索引

约束表是基本的数据库object，约束表和索引（及其属性）是数据建模人员更新和更改的第二频繁修改的库表特征，因此“约束表/索引”页面方便地位于数据库的第二页生成器的界面。

一旦在数据库构建器的树中选择了库表，“约束/索引”页面将填充当前定义的所选库表的约束和索引列表。然后，数据建模者可以直接在列表中对主要属性进行更改。当数据建模者在列表中选择单个约束或索引时，约束的扩展属性（和注释）立即显示在列表下方，允许修改这些扩展属性。



此图显示了显示对象树的数据库生成器界面和显示所选表的列的列选项卡。

注记

- “约束/索引”页面仅在选择数据库构建器树中的库表项时才会填充

数据库约束库表/索引

在Enterprise Architect中，库表约束和索引在同一屏幕上建模；它们统称为约束。数据库约束定义了强加于数据库行为的条件库表。他们包括：

- Primary Key - 唯一标识库表中的一条记录，由一列或多列组成
- 索引 - 提高对库表数据的检索和排序操作的性能
- 唯一约束- 唯一标识库表中一行的值的组合
- 外键 - 强制两个表之间的关系列（或列集合）
- 选择约束- 通过限制列接受的值来强制域完整性
- 库表触发器- #
或由于库表中的数据被修改而自动执行的代码

在Enterprise Architect中，您可以使用数据库构建器中专门设计的“约束/索引”页面或列和约束屏幕来定义和维护库表约束。

访问

功能区	开发 > 数据建模 > 数据库生成器 > 点击库表名称 > 约束/索引 右键 加新约束
上下文菜单	图中 右键单击库表 特征 约束/索引 右键 加新约束
键盘快捷键	点击库表：F9 > 约束/索引：Ctrl+N

创建一个约束

创建任何这些约束类型的过程是相同的，并且可以通过此处描述的一种方式来实现。

创建一个约束-使用上下文菜单或键盘

节	行动
1	A新的约束被自动创建并分配了默认名称约束 <i>n</i> （其中 <i>n</i> 是一个计数器）和一个“索引”的类型”。用您自己的约束名称改写默认名称。
2	如有必要，在“类型”字段中单击下拉箭头并选择适当的约束类型。
3	如果您愿意，请在“别名”字段中输入约束的别名。 “列”字段是只读的；它填充了您分配给“相关列”选项卡的列。

创建一个约束-改写模板文本

节	行动
1	在所选库表的 约束/索引 选项卡上，约束列表以模板文本新约束结束。 用适当的约束名称改写此文本，然后按 Enter 键。
2	新约束会自动创建并分配默认类型索引。 如有必要，在 类型 字段中单击下拉箭头并选择适当的约束类型。
3	如果您愿意，请在 别名 字段中输入约束的别名。 列 字段是只读的；它填充了您分配给 相关列 选项卡的列。

将列分配给约束

Primary Key、Foreign Key、Index、Unique 这些约束类型都必须至少分配一列；这定义了约束中涉及的列。

节	行动
1	在所选库表的 约束/索引 选项卡上，单击要为其分配列的约束。
2	可用列 面板列出了为库表定义的所有列。 对于要分配给约束的每一列，右键单击列名称并选择 分配列 <名称> 。 列名被传送到 分配的列 列表中。

从约束中取消分配列

节	行动
1	在所选库表的 约束/索引 选项卡上，单击要取消分配列的约束。
2	在 分配的列 列表中，右键单击要从约束中取消分配的列的名称，然后选择 取消分配列 <名称> 。 列名被转移到 可用列 列表中。

重新排序约束中的指定列

如果约束中有许多列，则可以通过将选定的列名一次在列表中向上或向下移动一个位置来重新排列序列。去做这个：

- 右键单击要移动的列名称并选择：
 - 向上移动列 **<名称>** (**Ctrl+向上箭头**) 或
 - 向下移动列 **<name>** (**Ctrl+向下箭头**)

删除约束

要删除不再需要的约束，请右键单击“约束/索引”选项卡上列表中的约束名称，然后选择“删除约束 <名称>”选项。如果满足给定约束类型的所有验证规则，则会立即从存储库中删除该约束以及所有相关关系（如果有的话）。

主键

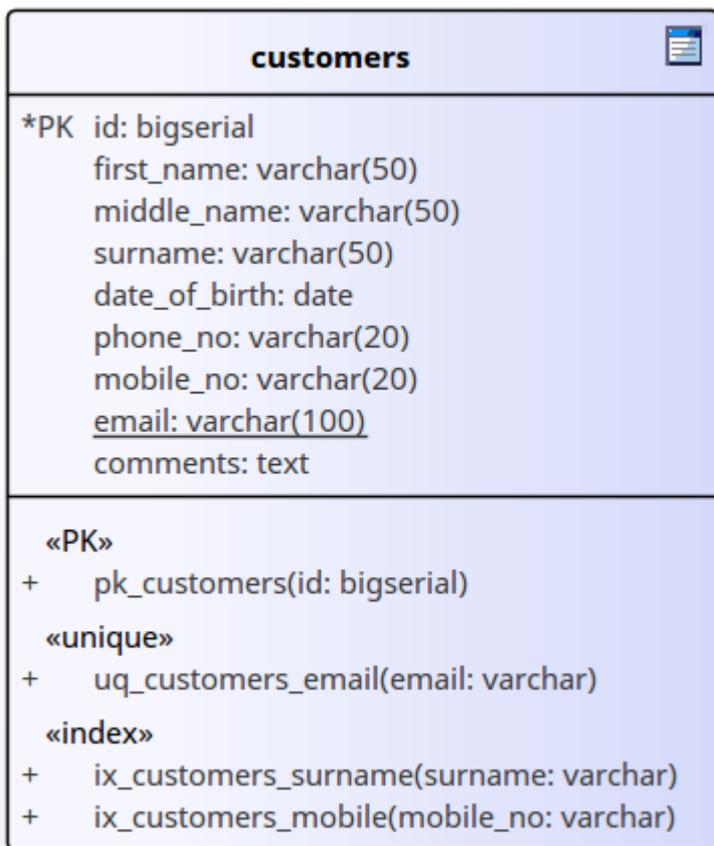
主键是唯一标识库表中每条记录A列（或列集）。A库表只能有一个主键。一些 DBMS 支持主键的附加属性，例如 Clustered 或 Fill Factor。

访问

功能区	开发 > 数据建模 > 数据库生成器 > 点击库表名称
上下文菜单	图中 右键单击库表 特征 约束/索引

创建主键

在Enterprise Architect中，您可以从“列”选项卡或“约束/索引”选项卡创建主键。无论哪种情况，当您将列添加到主键约束时，该列都会自动设置为“非空”。此外，任何包含库表元素的图表（假设设置了“显示限定词和可见性指示器”选项）都将针对列名显示“PK”前缀。在此图像中，请参见第一列“id: bigserial”。



创建主键 - 从列选项卡

节	行动
1	任何一个： <ul style="list-style-type: none"> 在数据库构建器中，单击具有一个或多个已定义列的库表，然后单击 列”选项卡，或 在图表上，单击库表并按 F9 显示 列”选项卡
2	对于要包含在主键中的每一列，选择 PK”复选框。 如果之前没有为当前库表定义主键约束，系统将使用主键名称模板创建一个新约束。

创建主键 - 从约束选项卡

节	行动
1	任何一个： <ul style="list-style-type: none"> 在数据库构建器中，单击包含一个或多个已定义列的库表，然后单击 约束/索引”选项卡，或 在图表上，单击库表并按 F10 以显示 约束/索引”选项卡
2	用主键名称改写新约束文本，按 Enter 键并单击 类型”字段下拉箭头，然后选择 PK”。
3	将所需的列分配给 PK 约束。
4	使用属性面板设置主键的扩展属性。 <ul style="list-style-type: none"> 填充因子是 0 到 100 之间的数值 Is Clustered 是一个布尔值，用于确定数据存储方式的物理顺序；对于大多数 DBMS，主键的 Is Clustered 属性默认为 True

从主键中删除列

您可以使用 列”选项卡或 约束/索引”选项卡从主键中删除列。

从主键中删除列 - 使用 列”选项卡

节	行动
1	任何一个： <ul style="list-style-type: none"> 在数据库构建器中，单击带有主键的库表，然后单击 列”选项卡，或 在图表上，单击库表并按 F9 显示 列”选项卡
2	针对要从主键中删除的每一列，取消选中“PK”复选框。 如果您已从主键约束中删除所有列并且不再需要主键，则必须手动删除它。

从主键中删除列 - 使用约束/索引选项卡

节	行动
1	任何一个： <ul style="list-style-type: none">在数据库构建器中，单击带有主键的库表，然后单击“约束/索引”选项卡，或在图表上，单击库表并按 F10 以显示“约束/索引”选项卡
2	根据需要取消分配 PK 约束上的列。

注记

- 警告：** Enterprise Architect假定主键约束至少分配了一列；但是，Enterprise Architect在建模期间不强制执行此规则
如果为主键未分配列的库表生成 DDL，则该 DDL 无效

数据库索引

将数据库索引应用于表以提高数据检索和排序操作的性能。一个库表可以定义多个索引；但是，每个索引都会在数据库服务器上增加开销（以处理时间和存储的形式）来维护它们，因为在库表中添加和删除信息

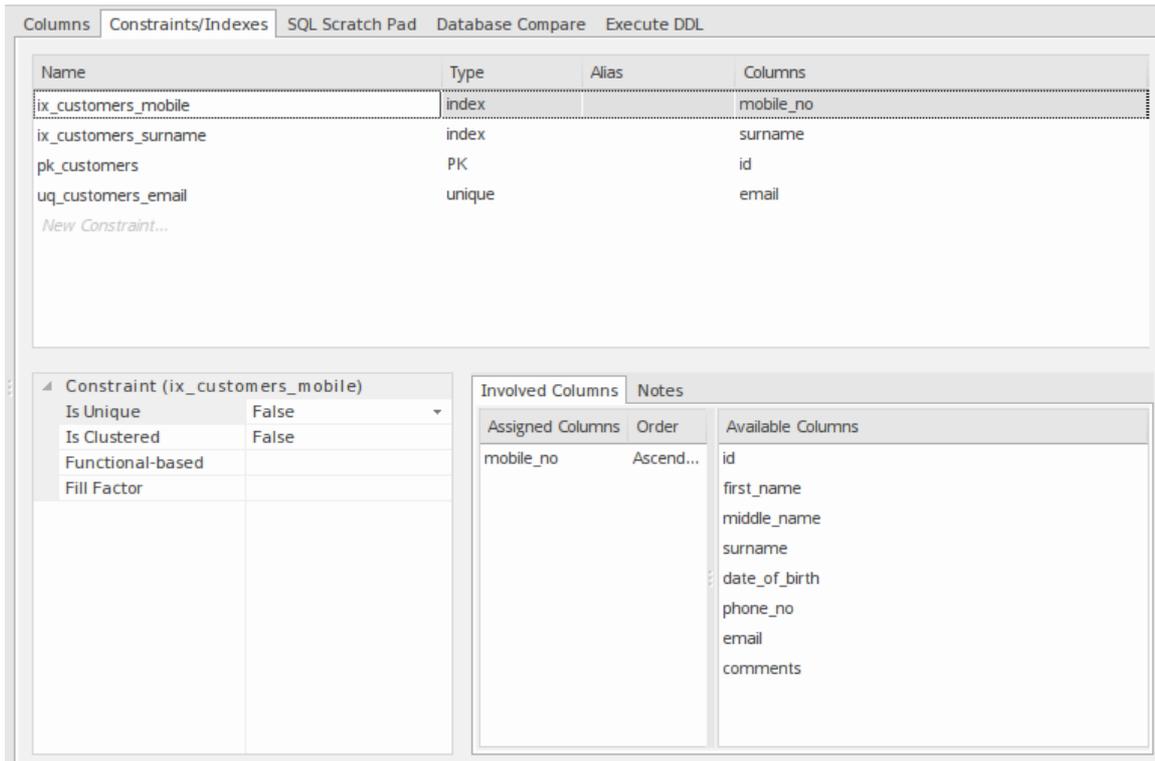
在Enterprise Architect中，索引被建模为原型操作。

一些 DBMS 支持特殊类型的索引； Enterprise Architect使用其他属性来定义这些，例如基于函数的、集群的和填充因子。

访问

功能区	开发 > 数据建模 > 数据库构建器 > 点击库表名称 > 约束/索引
上下文菜单	图中 右键单击库表 特征 约束/索引
键盘快捷键	点击库表：F9 > 约束/索引

处理索引



节	行动
1	在库表的“约束/索引”选项卡上，右键单击并选择“添加新约束”。 使用默认名称“constraint1”和“index”类型添加新约束。 用您喜欢的索引名称改写名称。

2	<p>将适当的列分配给索引。</p> <p>'Assigned Columns' 列表有一个附加的 'Order' 字段，用于指定每个已分配列存储在索引中的顺序（升序或降序）。您可以根据需要切换每列的顺序。</p> <p>此外，对于MySQL索引，“Len”字段将可见，您可以在其中定义部分索引；即，使用基于文本的字段的前导 <i>n</i> 个字符的索引。“Len”字段仅采用介于 0 和列的定义长度之间的整数数值。A 0（这是默认值）表示要索引整个列。</p>
3	<p>在 属性”面板中，审阅为当前 DBMS 定义的扩展属性的设置。</p>

附加属性

属性	描述
是独特的	<p>(True / False) 指示当前索引是否为 唯一索引”。唯一索引确保索引A（或列）不包含重复值，从而确保每一行具有唯一值（或索引由多列组成时的值组合）。</p>
是集群的	<p>(True / False) 指示当前索引是否为 聚集索引”。使用聚集索引，表的行在物理上以与索引中相同的顺序存储，因此每个表只能有一个聚集索引。默认情况下，一个表的主键是集群的。</p> <p>并非所有 DBMS 都支持聚集索引，因此“Is Clustered”索引属性仅对支持它的 DBMS 可见。</p>
是位图	<p>(True / False) 指示当前索引是否为 位图”索引。位图索引旨在用于具有相对较少唯一值的列（称为 低基数”列）并且物理上由每个唯一值的位数组（通常称为位图）组成。对于表中的每一行，每个数组都有一个位。</p> <p>考虑这个示例：在名为 性别”的列上创建位图索引，该列具有 男性”或 女性”选项。物理上，索引将由两个位数组组成，一个用于 男性”，一个用于 女性”。女性位数组在匹配行的值为 女性”的每个位中都有一个1。</p> <p>Is Bitmap 和 Is Unique属性是互斥的，所以当 Is Bitmap属性为True时，DDL 生成将忽略 Is属性。</p> <p>位图索引仅受 Oracle 支持；因此，此属性仅在对 Oracle 索引建模时可见。</p>
填充系数	<p>A介于 0 和 100 之间的数值，用于定义应用于数据的可用空间百分比。</p> <p>并非所有 DBMS 都支持填充因子，因此“填充因子”索引属性仅对支持它的 DBMS 可见。</p>
基于功能	<p>A #</p> <p>定义将被评估的函数/语句和索引结果的语句；例如： 降低（“字段”）</p> <p>并非所有 DBMS 都支持基于功能的索引，因此“基于功能的”索引属性仅对支持它们的 DBMS 可见，例如 PostgreSQL 和 Oracle。</p>
包括	<p>从当前表中标识非键列的逗号分隔列表 (CSV)。</p> <p>并非所有 DBMS 都支持索引的 包含”属性，因此该属性仅对支持它的 DBMS 可见。</p>

注记

- 警告：Enterprise Architect假定索引至少有一列分配给它们；但是，Enterprise Architect在建模期间不强制执行此规则
如果为定义了未分配列的索引的库表生成 DDL，则该 DDL 将无效，除非该索引是基于功能的
- 分配给基于功能的索引的任何列都将被忽略

独特约束

Unique约束在库表的所有行中强制一组字段的“唯一性”，这意味着库表库表中的任何两行不能在唯一约束的字段中具有相同的值。唯一约束与主键相似（因为它们也强制执行“唯一性”），但主要区别在于库表可以定义多个唯一约束但只有一个主键。

访问

功能区	开发 > 数据建模 > 数据库生成器 > 点击库表名称 > 约束/索引 > 右键单击 > 加新约束
上下文菜单	在图表或浏览器窗口 右键单击库表元素 特征 约束/索引
键盘快捷键	点击库表：F9 > 约束/索引：Ctrl+N

创建一个约束

节	行动
1	在“约束/索引”选项卡上，会自动创建一个新约束并为其分配默认约束名称和索引“类型”。使用将其标识为唯一约束的名称改写约束名称。
2	在“类型”字段中，将值从“索引”更改为“唯一”。

注记

- 警告：Enterprise Architect假定 Unique约束至少有一个列分配给他们；但是，Enterprise Architect在建模期间不强制执行此规则。如果为具有定义的唯一约束但未分配列的库表生成 DDL，则该 DDL 将无效。

外键

外键定义了A列（或列的集合），它强制两个表之间的关系。执行此关系以确保数据完整性是数据库服务器的责任。外键的模型定义由包含一组唯一数据的父（主）库表组成，然后在子（外）库表中引用该数据集。

在Enterprise Architect中，外键使用两个不同（但相关）的UML组件建模：

- 存储在子库表中A外键约束（具有 <<FK>> 原型的UML操作）
- 定义两者之间关系的关联连接器（<<FK>> 的刻板印象）表

创建外键

虽然外键的定义可能很复杂，但外键约束屏幕简化了外键的建模。此屏幕旨在帮助您选择要使用父库表中的哪个约束，并且会自动将子库表列与父库表中属于约束的列进行匹配。为了说明，这里分别描述了开发外键过程的不同方面，但整个过程应该是一个平稳的过渡。

A保存外键定义之前，必须满足许多条件：

- 两个表都必须定义匹配的 DBMS
- 父库表必须至少有一栏
- 父库表必须定义一个主键、唯一约束或唯一索引

创建外键 - 使用数据库

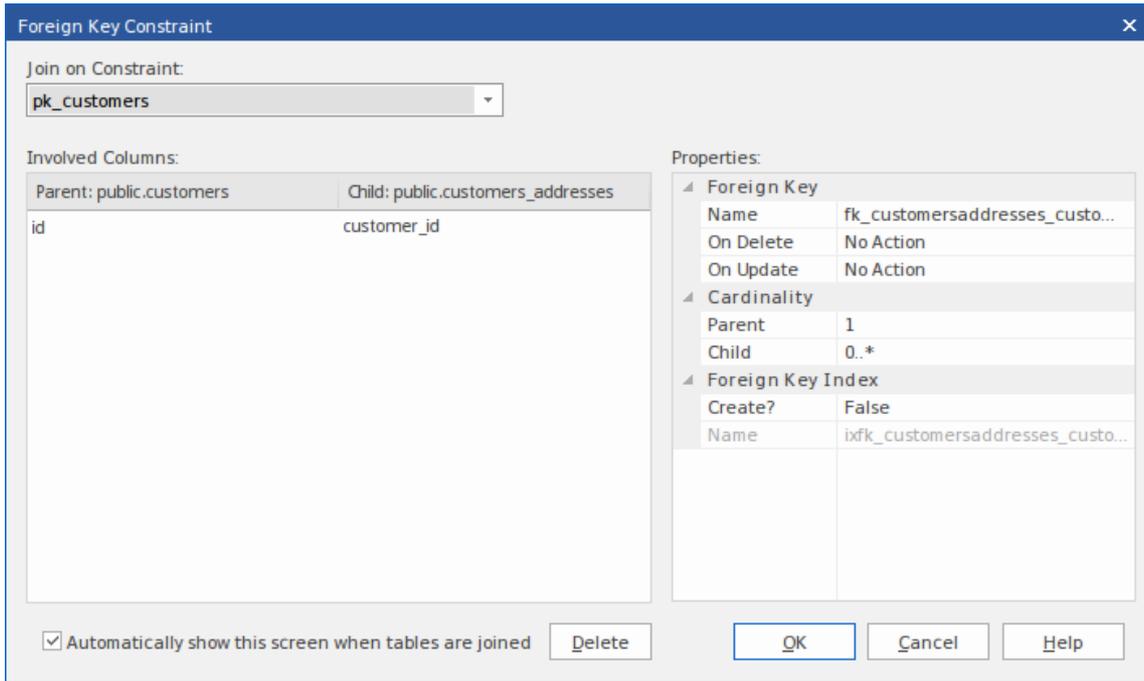
节	行动
1	在数据库构建器树中，右键单击子库表名称并单击 在 <表名称> 上添加新外键”。显示A对话框，列出所有可能的父表。
2	双击列表中所需的父库表名称或选择它并单击确定按钮。显示 外键约束”屏幕。

创建外键 - 在图表上使用关系

节	行动
1	在数据建模图中，找到需要的child（Foreign Key）库表和parent（Primary Key）库表。
2	在图表工具的“数据建模工具箱”页面中图表一个关联连接器。
3	单击子库表并将连接器绘制到父库表。
4	如果外键约束屏幕已设置为两个表连接时自动显示，则显示现在。否则，要么： <ul style="list-style-type: none"> • 双击连接器或 • 右键单击连接器并选择 外键”选项 外键约束屏幕显示。

外键约束屏幕

作为示例，此图像显示加载了“示例”详细信息的外键约束屏幕（如示例模型中所定义）。

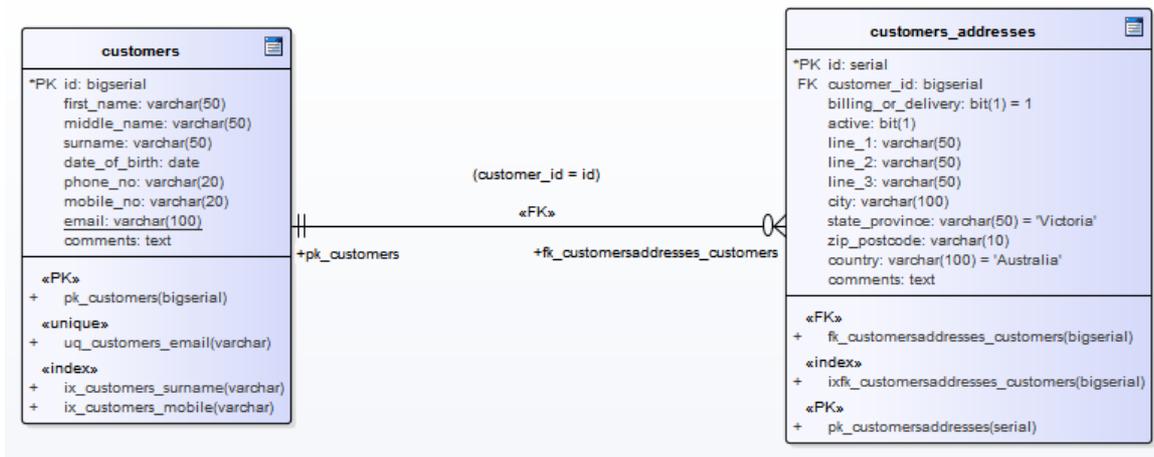


选项	行动
汇合关于约束	<p>这个组合框列出了父库表中所有定义的约束，这些约束可以用作外键的基础。（这些约束可以是主键、唯一约束或唯一索引。）</p> <p>默认选择列表中的第一个约束；如果这不是您想要的约束，请从组合框中选择正确的约束。</p> <p>当您选择约束时，它的列会自动列在“涉及的列”面板中的“父级：<tablename>”列下。</p>
涉及的列	<p>该列表分为两部分：左侧列出了所选约束中涉及的列，右侧列出了要与父列配对的子列。</p> <p>When a constraint is selected (in the '汇合 on constraint' field) the parent side is refreshed to display all columns assigned to the selected constraint.在子方面，系统将自动尝试将每个父列与子库表中的同名列匹配。如果子库表没有同名的列，则该名称的新列将被添加到列表中，用（*）标记表示将在库表中创建一个新列。</p> <p>但是，如果您想强制配对到现有的子库表列或具有不同名称的新列，请单击列名称字段，然后：</p> <ul style="list-style-type: none"> • 替换名称中的类型，或 • 选择现有列（单击下拉箭头并从列表中选择名称）
名称	<p>该字段定义外键约束的名称，默认为外键名称模板构造的名称。</p> <p>要将名称更改为默认值以外的名称，只需改写该值即可。</p>
关于删除	<p>选择删除父级数据时对子库表中的数据应采取的动作，以保持参照完整性。</p>

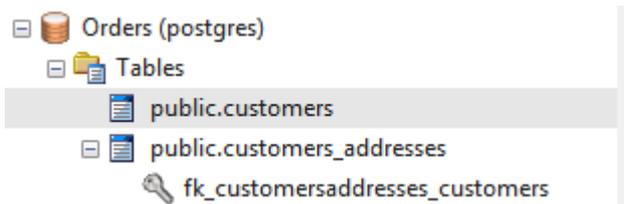
更新时	选择当父项中的数据更新时对子库表中的数据应采取的动作，以保持参照完整性。
家长	单击下拉箭头并在外键中选择父库表的基数。
子	单击下拉箭头并在外键中选择子库表的基数。
创建？	如果要在创建外键的同时创建外键索引，请将此属性设置为True。 外键索引的名称由外键索引模板控制，生成的名称显示在“创建？”下方的“名称”字段中场地。
加入库表时自动显示此画面	(对于图表建模)选中此复选框可在两个表之间创建关联时自动显示此屏幕。
删除	单击此按钮可删除当前选择的现有(已保存)外键。 将显示A提示以确认删除(以及外键索引的删除,如果存在)-单击是按钮。 删除外键会留下一个关联连接器,您可以对其进行编辑或删除(右键单击并选择“删除关联:到<库表名称>”)。
确定	单击此按钮以保存外键。

例子

此示例在图表中显示了简单的外键：



相同的外键将作为库表“customers.addresses”下的子节点显示在数据库构建器的树中。



选择约束

A选择约束通过限制列接受的值来强制域完整性。

访问

功能区	开发 > 数据建模 > 数据库生成器 > 点击库表名称 > 约束/索引 > 右键单击 > 加新约束
上下文菜单	图中 右键单击库表 特征 约束/索引 右键 加新约束
键盘快捷键	点击库表 : F9 > 约束/索引 : Ctrl+N

创建一个约束

节	行动
1	在列和约束屏幕的“约束/索引”选项卡上，会自动创建一个新约束并分配默认约束名称和索引的“类型”。 使用将约束标识为检查约束的名称改写约束名称，例如“CHK_ColumnName”（CHK_前缀是可选的）。
2	在“类型”字段中，将值从“索引”更改为“检查”。
3	在条件属性的“属性”面板中，键入# 将用作选择条件的语句；例如，第 1 列 < 1000。 如果条件很长，点击  按钮显示# 编辑器（带有语法高亮）。

删除一个约束选择

如果您不想保留检查约束，则：

- 在列表中右键单击它并选择“删除约束 <名称>”，或者
- 单击项目并按 Ctrl+D

约束被立即删除。

注记

- 分配给检查约束的任何列都将被忽略

库表触发器

库表触发器A

或者由于数据库中的数据被修改而自动执行的代码库表。触发器可高度定制，并且可以以多种不同方式使用；例如，它们可用于停止在工作时间执行某些数据库活动，或者在删除主库表中的记录时提供验证或在辅助表中执行删除。

在Enterprise Architect中，库表触发器被建模为一个定型操作并使用库表的“约束”屏幕进行管理。

访问

功能区	开发 > 数据建模 > 数据库生成器 > 点击库表名称 > 约束/索引 右键 加新约束
上下文菜单	图中 右键单击库表 特征 约束/索引 右键 加新约束
键盘快捷键	点击库表：F9 > 约束/索引：Ctrl+N

创建库表触发器

节	行动
1	在“约束/索引”选项卡上，会自动创建一个新约束并为其分配默认约束名称和索引“类型”。使用将约束标识为触发器的名称改写约束名称，例如 TRG_OnCustomerUpdate。（TRG_ 前缀是可选的。）
2	在“类型”字段中，将值从“索引”更改为“触发器”。
3	在 Statement 属性的“属性”面板中，输入完成# 将定义简单的语句（包括 CREATE 触发器）。 如果条件很长，点击  按钮显示# 编辑器（带有语法高亮）。
4	触发器时间和触发器事件属性当前是仅供参考的值，不用于 DDL 生成。

删除库表触发器

如果您不想保留触发器，则：

- 在列表中右键单击它并选择“删除约束 <名称>”，或者
- 单击项目并按 Ctrl+D

触发器立即被删除。

注记

- 分配给表触发器的任何列都将被忽略

SQL便签本

SQL便签本提供了一种机制来开发和运行ad-hoc #

针对实时数据库的查询。在开发数据模型时，您可能想要执行和测试 ad-hoc #

查询 DDL 脚本，或在实时数据库上运行查询；所有这些都可以在Enterprise Architect数据库生成器界面中实现。

SQL便签本要求数据库构建器具有与实时数据库的有效连接。此数据库连接在数据库构建器的“SQL便签本”、“数据库比较”和“执行DDL”选项卡之间共享。

Scratch Pad 包括：

- 提供导入、保存、执行和清除#
功能A工具栏#
查询
- 一个编辑器面板，您可以在其中创建或导入#
查询 - 此面板为当前数据模型提供基于 SQL 的语法突出显示
- 由两页组成A选项卡式面板，一页显示执行查询的结果，一页显示执行期间生成的任何消息

访问

打开数据库生成器窗口，然后显示“SQL便签本”选项卡。

功能区	开发 > 数据建模 > 数据库 > SQL便签本
-----	--------------------------

便签本工具栏



Scratch Pad Toolbar 上每个按钮的功能在此表中进行了描述，从左到右。

按钮	行动
运行#	执行# 当前显示在查询Pad 中。 为执行查询的输出选择“结果”和“消息”选项卡。
新的	清除# 查询编辑器字段，以便您可以输入新查询。
打开	加载一个# 从文件中查询。 A源文件浏览器显示，默认显示# 文件。单击文件名和打开按钮以在便笺簿中显示文件内容。
保存到# 查询	保存这个# 对# 的声明# 它来自的查询object。

保存到新# 查询	创建一个新的# 查询object并将此语句保存到该object。
保存到文件	将当前显示的查询保存到它来自的文件。 如果您从头开始创建查询，则会显示一个源文件浏览器，您可以在其中键入新文件名并单击“保存”按钮以保存查询。
保存到新文件	将当前显示的查询保存到新的 .sql 文件。 A源文件浏览器显示您在其中键入新文件名并单击“保存”按钮以保存查询。
清除	清除便笺本的内容。 Scratch Pad 中显示的任何内容都会保留在那里，直到您将其替换为另一个来自文件的查询或关闭模型查询。
切换注解	应用# 注释字符 '--' 到每个选定行的开头，或者，如果选定的行已被注释，则删除注释字符。或者，按 Ctrl+Shift+C。
语句分隔符	用于标记每个语句结束的字符中的类型。
帮助	在# 上显示帮助# 查询便笺。
查询描述	显示一个标签，提供当前# 的描述# , 是否有挂起的更改（由前导 * 表示），以及已加载的名称# 查询object或文件名。

注记

- SQL便笺本不会操纵您的#
无论如何，因此您必须对当前 DBMS 使用正确的语法
- 而SQL便笺本可以执行多个#
语句，并且每个语句的状态和消息都显示在“消息”列表中，一次只能在“结果”列表中显示一个SELECT语句的结果；所有后续的SELECT语句都将被忽略

数据库比较

数据库比较”选项卡提供了一种将当前数据模型 实时数据库进行比较的机制，并可选择同步任一方向的任何差异。使用 “Alter DDL” 语句执行 “推送”到实时数据库的差异，而从实时数据库导入的更改可以直接 “拉”到模型中。

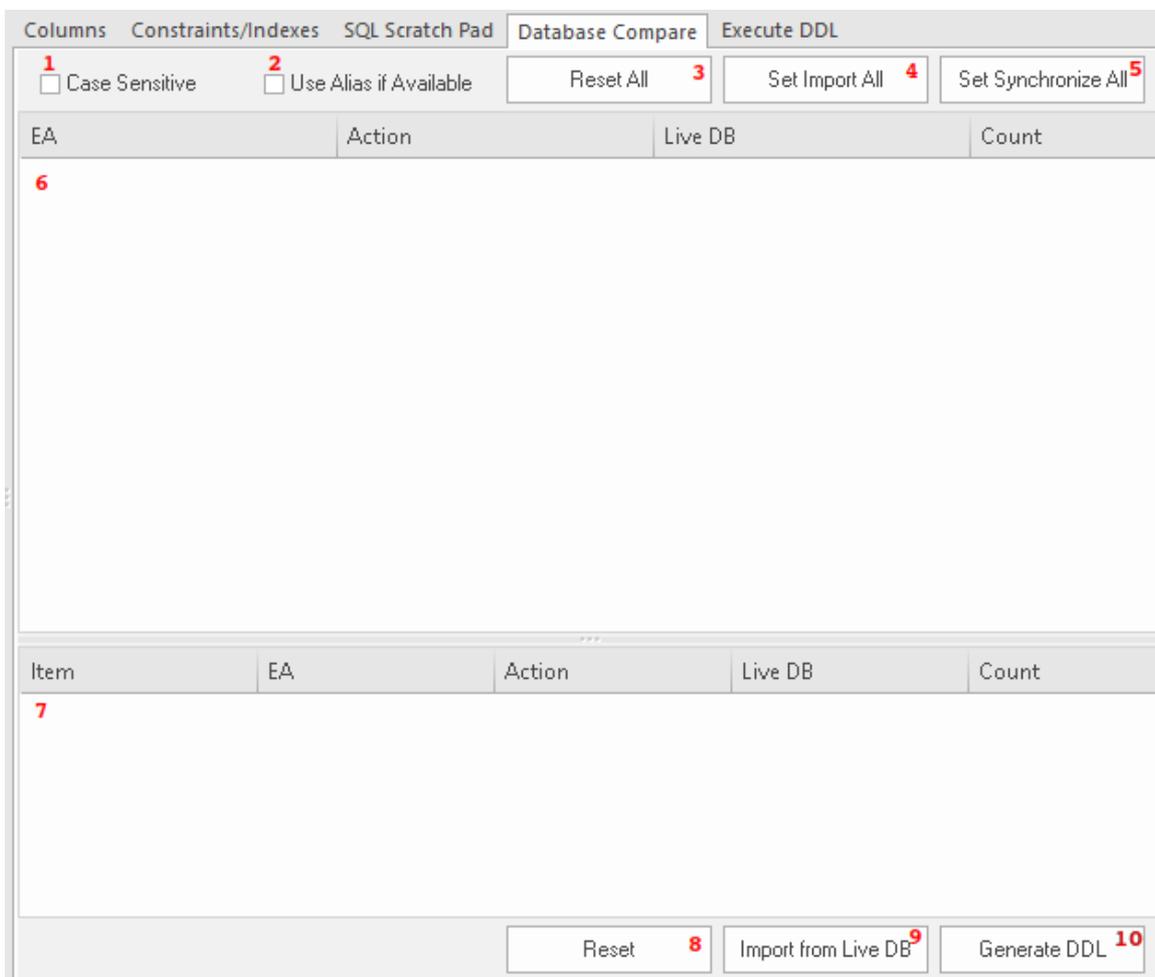
数据库比较功能要求数据库生成器与实时数据库建立有效连接。此数据库连接由数据库构建器的 “SQL便签本”、“数据库比较”和 “执行 DDL”选项卡共享。

访问

打开数据库生成器窗口，然后显示 “数据库比较”选项卡。

功能区	开发 >数据建模>数据库构建器 >数据库比较
-----	------------------------

DDL 比较选项卡



数据库比较’选项卡有许多控件，如此处所述。

编号及名称	描述

1 区分大小写	单击此复选框可使所有属性比较识别属性文本中字母大小写的差异。
2 使用别名 (如果可用)	单击此复选框以指示应使用任何已定义的别名而不是object名称 (在object和列级别)。
3 全部重置	单击此按钮可将所有对象的 行动”标志设置回默认值。
4 设置全部导入	单击此按钮可将所有检测到的差异的 行动”标志设置为<====; 也就是说, 使用实时数据库中的值更新模型。
5 设置全部同步	单击此按钮可将所有检测到的差异的 行动”标志设置为====>; 也就是说, 使用模型中的值更新实时数据库。
6 差异	审阅发现模型与实时数据库不匹配的对象列表。在此列表中选择个项目将填充 组件”列表。 (有关每列的详细说明, 请参阅差异列表表。)
7 个组件	审阅此模型和实时数据库之间不同的所选object的属性列表。 (有关每列的详细说明, 请参阅部件列表表。)
8 重置	单击此按钮可将当前object的所有属性的 行动”标志设置回默认值。
9 从 Live DB 导入	单击此按钮可将实时数据库中的所有属性值 (带有 <==== 的 行动”) 导入模型。
10 生成DDL	单击此按钮可为所有具有====> '行动'的对象生成'Alter DDL'语句, 并将语句发送到'Execute DDL'选项卡。

差异列表

柱子	描述
EA	显示模型中检测到一个或多个差异的每个object的名称。空白值表示模型中缺少object, 但在实时数据库中存在。
行动	<p>考虑到此object的差异, 默认为 无行动”作为要采取的行动。单击下拉箭头并选择特定操作。列表中可用操作的列表将取决于给定object是否在模型和实时数据库中配对。</p> <p>配对对象</p> <ul style="list-style-type: none"> 无行动- 不更新数据库或模型与此更改 ====> - 从模型更新数据库中的object <==== - 从数据库更新模型中的object 自定义 - 在下方面板中对每个项目设置不同的操作之前, 将项目设置为无行动 Unpair - 将配对的对象分开, 这样它们就不会相互比较或相互更新 <p>未配对的对象</p> <ul style="list-style-type: none"> Create < object name> - 在数据库或模型中创建缺失的数据库object, 视情

	<p>况而定</p> <ul style="list-style-type: none"> • 删除< object名称> - 从模型中删除object • Drop < object名称> - 从数据库中删除object • Pair with < object name> - 将数据库中的object与模型中的命名 (未配对) object配对, 以便比较它们之间的差异 <p>组件列表” (下部面板) 中的 行动”字段将根据该字段的选择进行更新。</p> <p>例如, 如果实时数据库有一个库表列 地址1”而模型没有, 则将object 行动”设置为 “==>” (从模型更新数据库中的object) 设置列 项移动”到 Drop行动”, 这将从实时数据库中删除该列。</p>
实时数据库	显示实时数据库中检测到一个或多个差异的每个object的名称。空白值表示该object存在于模型中, 但在实时数据库中缺失。
数数	显示在模型和实时数据库之间检测到的object (及其所有组件) 差异的总数。

部件列表

柱子	描述
物品	显示每个检测到的差异的组件名称或描述。差异分为三类：属性、列和约束, 采用树形结构。
EA	显示模型中检测到的给定组件的值。空白值表示模型中缺少该值, 但在实时数据库中不存在。
行动	<p>默认为与 差异”列表中object 行动”字段的设置相对应的操作, 以指示针对为组件检测到的差异要采取的操作。单击下拉箭头以选择替代操作; 列表中的可用选项取决于组件的类型和检测到的差异。</p> <ul style="list-style-type: none"> • 无行动——不更新数据库或模型 • ==> - 从模型更新实时数据库中的object • <=== - 从实时数据库更新模型中的object • 添加 <项目名称> - 在数据库或模型中创建缺少的项目, 视情况而定 • 删除<项目名称> - 从模型中删除项目 • Drop <item name> - 从实时数据库中删除该项目
实时数据库	显示活动数据库中选定组件的值。空白值表示该值存在于模型中, 但在实时数据库中缺失。
数数	显示在所选组件中检测到的模型和实时数据库之间的差异数量。

使用数据库比较

每当您执行比较时, Enterprise Architect都会从实时数据库和模型中读取定义, 然后尝试使用其名称 (和模式, 如果与当前 DBMS 相关) 将来自一个源的每个object与另一个源 配对”。

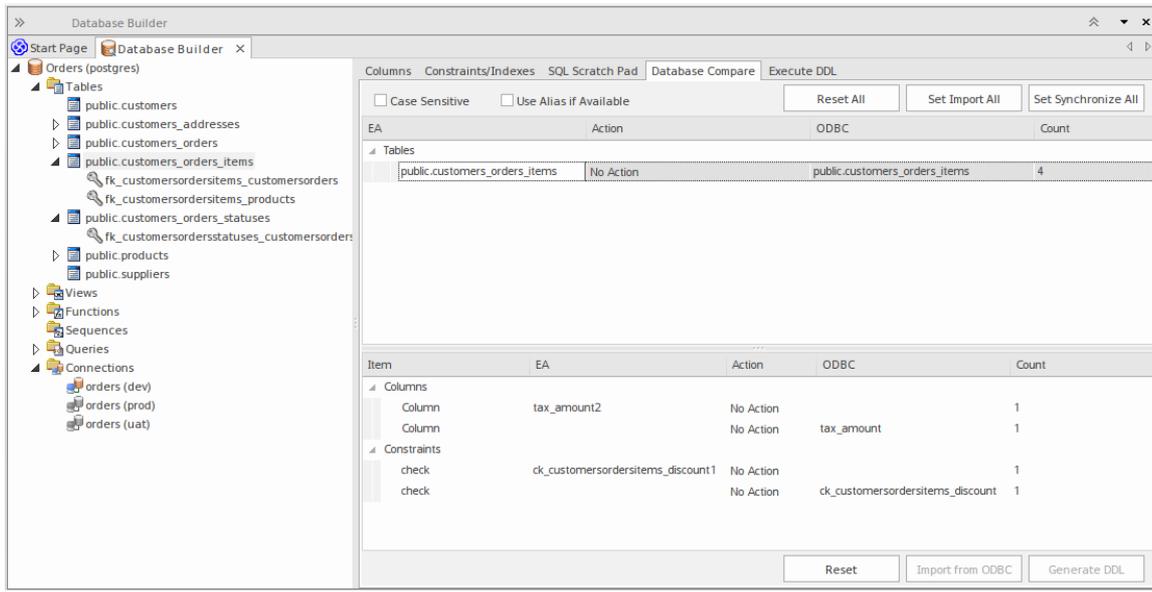
如果找到匹配项, 则object名称将显示在 EA”和”Live DB”列中, 默认操作为 无行动”。 计数”列表示为object及

其组件或属性找到的差异总数。

如果在系统之间未找到匹配项，则object名称显示在源列（“EA”或“Live DB”）中，而另一列为空白。在这种状态下，可以将object与不同名称的object配对；“行动”下拉列表将显示可用对象。如果进行了新配对，则会比较两个对象的定义是否存在差异，结果将显示在“组件”列表中，并选择默认操作“====>”。

如果您在object级别选择一个动作，这将为object的所有组件和属性设置匹配动作。但是，如果您在object级别选择“自定义”操作，您可以为每个组件确定不同的操作。

例如，列 (tax_amount) 和约束 (ck_customersordersitems_discount) 在库表 “public.customers_order_items” (在示例模型中) 中都被重命名，并执行了数据库比较；这张图片显示了发现的差异：



在图像中只有一个库表检测到差异 - 'public.customers_order_items'；选择此项会填充“组件”列表。从检测结果可以确定数据模型包含实时数据库不包含的列 (tax_amount2) 和检查约束 (ck_customerordersitems_discount1)，而实时数据库又包含列 (tax_amount) 和检查约束 (ck_customerordersitems_discount) 数据模型没有。

与选项比较

“选项比较”功能的工作方式与直接比较相同，只是提示您选择应执行的object/属性。这使您可以忽略当前不相关的特定差异。

这些库表描述了可以启用或禁用的不同比较。

所有对象，拥有着

比较	行动
拥有着	选择表示所有数据库对象的“拥有着”属性应在对象“配对”后进行比较。

库表选项

选项	行动
表	选择此父选项以启用所有库表比较选项。取消选择以禁用所有其他选项。然后，您将取消选择或选择列表中的特定选项。
库表-扩展属性	选择以指示应该比较表的扩展属性（例如 DB版本和表空间）。
库表-备注	选择表示应比较应用于表的备注。
列	选择此父选项以启用所有“列比较”选项。取消选择以禁用所有其他“列”选项。然后，您将取消选择或选择列表中的特定选项。
列 - 类型	选择以指示应比较库表列的数据类型名称。
列 - 大小	选择以指示应比较库表列的数据类型大小。
列 - 默认值	选择表示应比较库表列的默认值。
列 - 位置	选择以指示应比较库表列的位置。
列 - 非空	选择表示应该比较库表列的非空属性。
列 - 自动编号	选择以指示应比较库表列的自动编号属性（例如 AutoNum、StartNum 和 Increment）。
列 - 不匹配的列	选择以指示应比较模型和实时数据库之间不匹配的库表列。通常，这些列存在于一个系统中，但不存在于另一个系统中。
列 - 扩展属性	选择以指示应比较库表列的扩展属性（例如 Unsigned 和 Zerofill）。
专栏 - 备注	选择以指示应比较应用于库表列的备注。
约束	选择此父选项以启用所有“库表约束比较”选项。取消选择以禁用所有“库表约束”选项。然后，您将取消选择或选择列表中的特定选项。
约束-主键	选择以指示应比较与主键相关的属性。
约束-外键	选择以指示应比较与外键相关的属性。
约束-索引	选择以指示应比较与索引相关的属性。
约束-约束独特	选择以指示应比较与唯一约束条件相关的属性。
约束-约束选择	选择以指示应比较与选择约束相关的属性。
约束-库表触发器	选择以触发器应比较与库表相关的属性。
约束-无与伦比约束	选择以指示应比较模型和实时数据库之间不匹配的库表约束。通常这些是存在于一个系统中但不存在于另一个系统中的约束。
约束-扩展属性	选择以指示应比较库表约束的扩展属性（如填充因子和集群）。

约束-备注	选择以指示应比较应用于库表约束的备注。
-------	---------------------

注记

- 数据库比较功能目前可以对库表、视图、过程、函数和序列object类型进行比较

执行 DDL

执行 DDL”选项卡提供了一种机制，可以轻松地对实时数据库执行生成的 DDL 语句，并提供有关其成功的即时反馈，所有这些都 Enterprise Architect 界面内，无需其他产品。

Enterprise Architect 可以生成两种不同类型的 DDL 语句并将其发送到 执行 DDL”选项卡：

- 创建由生成 DDL 屏幕创建的 DDL 语句，以及
- 由数据库比较窗口创建的更改 DDL 语句

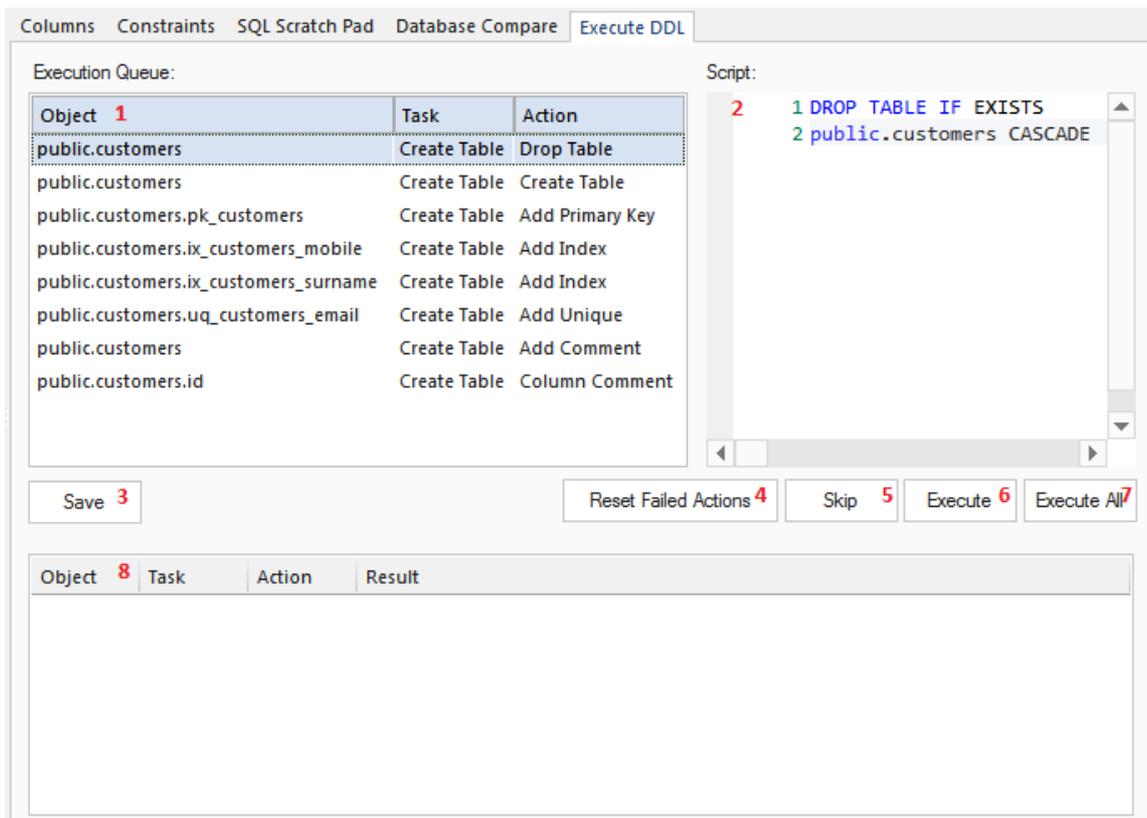
执行 DDL 功能需要数据库生成器与实时数据库建立有效连接。此数据库连接在数据库构建器的 SQL 便签本、数据库比较和 执行 DDL”选项卡之间共享。

访问

打开数据库生成器窗口，然后显示 执行 DDL”选项卡。

功能区	开发 > 数据建模 > 数据库 > 执行 DDL
-----	--------------------------

执行 DDL



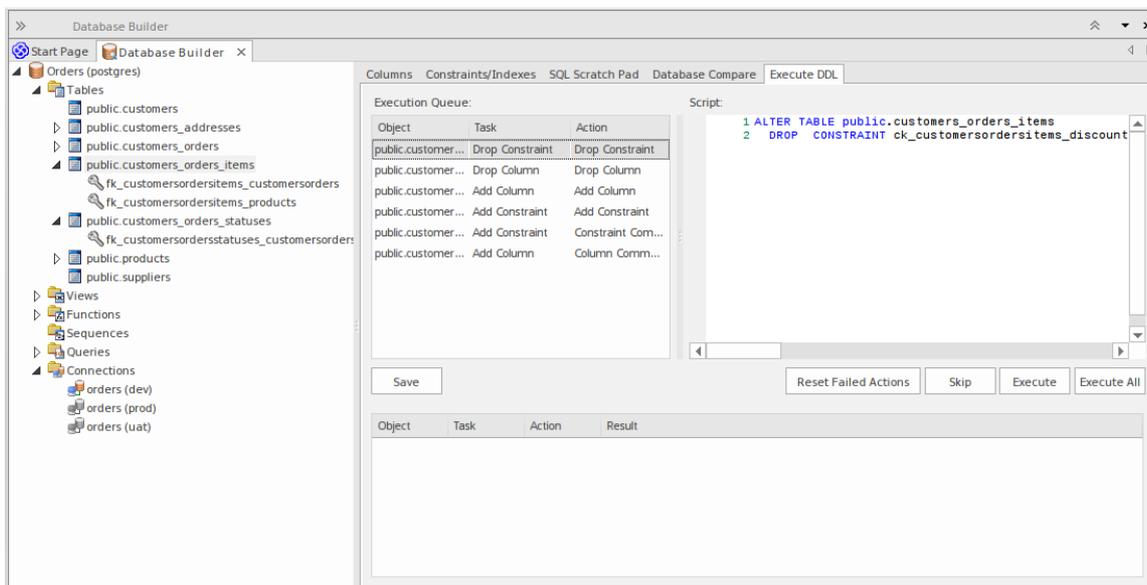
执行 DDL”选项卡具有以下字段和按钮：

字段/按钮	行动
1 执行队列	列出尚未执行的任务（每个都有关联的 DDL 语句）。该列表包含三列，指定

	所涉及object的名称、任务和正在执行的操作。 在列表中选择个项目将显示给定任务的关联 DDL 语句（在“脚本”字段中）。
2脚本	带有# A文本框# 语法高亮显示，显示所选任务的 DDL 语句。
3 保存	单击此按钮可将 执行队列”和 结果列表”中的所有单个 DDL 语句保存到单个文件中。
4 重置失败行动	单击此按钮可将任何失败或跳过的任务从 结果列表”重新排入 执行队列”底部。
5 跳过	单击此按钮可跳过 执行队列”中的下一个任务而不执行它。该任务将被移动到 结果列表”中，并且不会给出结果。 当您单击 重置失败的行动”按钮时，跳过的任务将与任何失败的任务一起返回到执行队列。
6 执行	单击此按钮以执行 执行队列”中的下一个任务。该任务从 执行队列”的顶部删除，并与执行结果一起添加到 结果列表”的末尾。
7 全部执行	单击此按钮以执行 执行队列”中的所有任务。当执行完成时， 结果列表”将显示每个单独任务的结果。
8 个结果列表	列出已执行的任务以及每个任务的执行结果。在此列表中选择个项目将在“脚本”字段中显示已执行的 DDL 语句。

示例

在前面关于数据库比较的部分中使用的示例中（当重命名列和约束时），如果使用默认值将数据模型更改 推送”到实时数据库中，则执行 DDL 屏幕将填充此处显示的详细信息。



总之，生成 DDL 以删除旧列和旧约束（任务“删除列”和“删除约束”），然后使用新名称创建列和约束（任务“添加列”和“添加约束”），最后每个人都应用了他们的评论/备注（任务“添加约束-约束注解”和“添加列-注解”）。

数据库对象

虽然表是关系数据库的基本组件并允许定义列、数据类型、键和索引，但在 RDBM 系统中还有许多其他重要的对象，包括：

- 视图-视图表示预定义查询的结果集；它们是从存储在一个或多个表（或其他视图）中的数据动态得出的
- 过程-一些 DBMS 产品实施以提供可以包含一个或多个#
例程的特征#
执行特定任务的语句，例如数据验证、访问控制或减少客户端和 DBMS 服务器之间的网络流量
- 函数——一些特征产品实现的特性，用于提供扩展数据库服务器功能的机制；each 是一个例程，可以接受参数、执行一个动作（例如复杂的计算）并将该动作的结果作为值返回
- 序列-一些 DBMS 产品实现的特征，以提供生成唯一值的机制 - 确保对它的每次调用都返回唯一值

UML本身并没有指定如何执行数据建模，但Enterprise Architect具有用于数据建模的完全集成的UML配置文件以及一系列内置于核心产品中的特征，这些特征将使数据建模变得容易。

该配置文件使用构造型和标记值将标准UML元素扩展为数据建模结构。这是通过将数据库object型添加到UML类来实现的；这样您就可以模型：

- 数据建模diagrams as extended UML类diagrams
- 表作为具有 <<table>> 构造型的UML类对象
- 视图作为具有 <<view>> 构造型的UML类对象
- 过程作为具有 <<procedure>> 构造型的UML类对象
- 函数作为具有 <<function>> 构造型的UML类对象
- 序列作为具有 <<dbsequence>> 构造型的UML类对象

您可以使用Enterprise Architect在您的数据库模型中快速创建和配置所有这些对象。

数据库表

表是关系数据库的基本组件，表示多行结构化数据元素（称为列）。输入到关系数据库中的每一项数据都由列中的值表示。

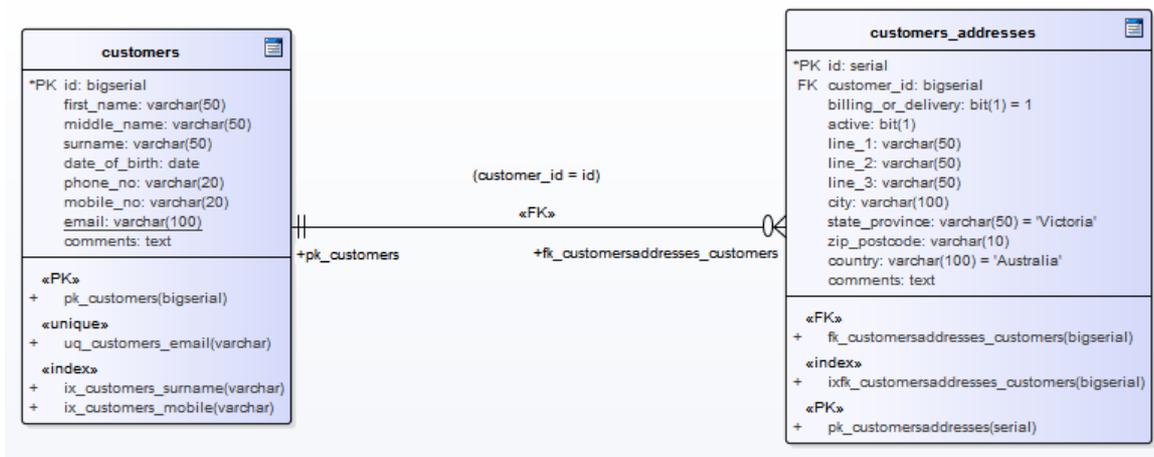
Enterprise Architect的用于数据建模的UML配置文件表示：

- 数据库表作为具有 <<table>> 构造型的UML类对象
- 库表列作为库表的UML属性，库表型为 <<column>>
- 主键作为库表的UML操作/方法，具有 <<PK>> 的构造型
- 外键作为库表的UML操作/方法，具有 <<FK>> 的构造型
- 索引作为库表的UML操作/方法，具有 <<index>> 的构造型
- Unique约束作为库表的UML操作/方法，具有 <<unique>> 的刻板印象
- 选择约束作为库表的UML操作/方法，具有 <<check>> 的刻板印象
- 库表作为库表表的UML操作/方法，具有 <<trigger>> 的触发器型

Enterprise Architect将库表的所有UML操作统称为约束，因此您用于维护库表的UML属性和操作的屏幕称为列和约束屏幕。

示例

Enterprise Architect中的物理数据模型图这个简单示例由两个由UML类表示的数据库表组成，名为customers和customer_addresses。



每个库表定义数据库列，使用为目标 DBMS（在本例中为 PostgreSQL）适当键入的UML属性。

笔记

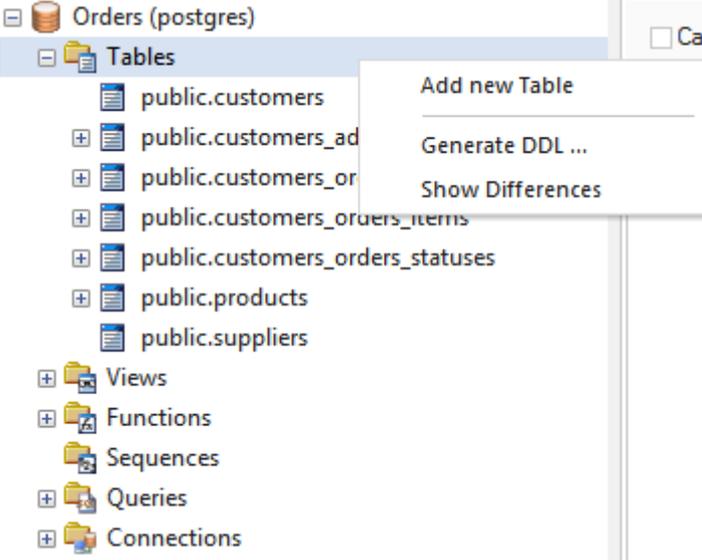
- 库表stereotype由每个类右上角的图标表示（参见数据建模Notation主题）
- 用于管理库表列的Enterprise Architect维护屏幕不允许您更改属性构造型，因为 <<column>> 是唯一有效的选项
- 可以隐藏示例表中显示的 <<column>> 构造型标签（参见数据建模注释主题）

创建数据库库表

数据建模的基础是在模型中创建数据库表。创建库表的三种方式：

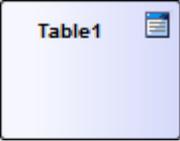
- 在数据库生成器中
- 在打开的数据模型图上
- 使用浏览器新元素选项

使用数据库添加数据库库表

节	行动
1	打开数据库构建器（ 开发 >数据建模>数据库构建器” ）。
2	加载或创建数据模型。
3	<p>右击表包，选择 “加新库表”。</p> 
4	用适当的库表名称改写默认名称，然后按 Enter 键。
5	双击库表元素，定义库表属性。

将数据库库表添加到图表

节	行动
1	创建和/或打开数据建模图。
2	将 “库表” 工具箱图标拖放到图表上。

	<p> Table</p> <p>这会生成一个新的库表元素：</p> <div data-bbox="279 280 459 421"> Table1</div>
3	双击库表元素，定义库表属性。

数据库库表栏目

在关系数据库中，库表列（有时称为字段）在库表的每一行中存储特定类型的单个数据值。库表列可以具有各种单独的属性，例如默认值或是否字段接受 Null 值。

A数据库库表列在UML数据建模配置文件中表示为一个原型属性；即，具有 <<column>> 构造型的属性。在 Enterprise Architect中，您可以使用专门设计的数据库构建器的“列”页面或“列和约束”对话框来定义和维护库表列。

创建数据库库表列

数据库库表列A UML数据建模配置文件中表示为具有 <<column>> 构造型的属性。对于选定的库表，您可以在数据库构建器的“列”页面或“列和约束”屏幕上审阅现有列并创建新列。

您可以直接在“列”选项卡上的列列表中定义列详细信息。当您完成每个字段时，更改会自动保存。某些字段对您可以输入的数据有一定的限制，如此处所述。该选项卡还包含一个“属性”面板和一个“注记”字段，其中填充了所选列上的现有信息。您创建的每个新列都会自动分配一组默认值并添加到列表底部。

访问

功能区	开发 > 数据建模 > 数据库构建器 > 点击库表 > 列 > 右键 > 添加新列
上下文菜单	在图表中，右键单击所需的库表 特征 专栏 右击 添加新列
键盘快捷键	选择表 F9 Tab 键 (在“列”选项卡上设置输入聚焦) Ctrl+N

在库表中创建列

选项	行动
名称	用适当的列名文本改写默认名称。
类型	单击下拉箭头并为列选择适当的数据类型。 可用的数据类型取决于分配给父库表的 DBMS。
长度	(可选) 某些数据类型具有长度组件 - 例如，VARCHAR 具有定义可以存储的字符数的长度。如果数据类型没有长度组件，则禁用此字段。 如果该字段可用并且您需要定义多个字符，请在此处键入值。
规模	(可选) 某些数据类型具有比例组件 - 例如，DECIMAL 具有定义可以保留的小数位数的比例。如果数据类型没有比例组件，则禁用此字段。 如果该字段可用并且您需要定义比例，请在此处键入值。
PK	如果该列是此库表的主键的一部分，请选中该复选框。
不为空	如果该列禁止使用空值，请选中该复选框。 如果选中“PK”复选框，则该复选框被禁用。
别名	如果出于显示和文档目的需要，请输入该字段的替代名称。
初始值	如果需要，输入一个可用作此列默认值的值。
注记	类型在记录该列所需的任何附加信息中。 您可以使用字段顶部的注记工具栏来格式化文本。

列属性

库表的数据库管理系统的相应属性会自动显示在“属性”面板中（如果不可见，请展开“列（<名称>）”分支）。

属性	数据库管理系统
自动编号（起始编号增量）	甲骨文 MySQL # 服务器 DB2 PostgreSQL 注记：如果您需要自动编号序列，请将此属性设置为True，并在必要时定义起始编号和增量。
生成	DB2 注记：将此附加属性设置为DB2中的自动编号，设置为“默认”或“始终”。
不代表	SQL服务器 注记：如果要阻止复制，请将此属性设置为True。
零填充	MySQL 注记：将此属性设置为True或False以指示字段是否填充为零。
未签名	MySQL 注记：将此属性设置为True或False以指示字段是否接受无符号数字。
长度类型	甲骨文 注记：设置该属性以定义字符语义为“None”、“Byte”或“Char”。

删除数据库库表列

对于选定的数据库库表，您可以在“列和约束”屏幕的“列”选项卡上审阅现有列并删除任何单个列。

访问

使用此处概述的方法之一来显示表格的列表，然后选择一列并将其删除。

当您选择“删除列<名称>”选项时，如果满足所有验证规则，则立即删除该列。

功能区	开发>数据建模>数据库生成器>点击库表>栏目>右键栏名>删除栏目<名称>
上下文菜单	在图表中，右键单击所需的库表 特征 专栏 右键单击列名 删除列<名称>
键盘快捷键	F9 使用'向上箭头'或'向下箭头'选择一列 Ctrl+D

注记

- 如果删除的数据库库表列涉及任何约束，它将自动从它们中删除

重新排序数据库库表列

如果您在数据库库表中定义了多个列，则可以更改它们的列出顺序。列表中的顺序是列在生成的 DDL 中出现的顺序。

访问

使用此处概述的方法之一来显示库表的列列表，然后选择一列并在列表中重新定位它。

功能区	开发 > 数据建模 > 数据库生成器 > 点击库表
上下文菜单	在图表中，右键单击所需的库表 特征 列
键盘快捷键	F9

更改列顺序

节	行动
1	在“列”选项卡中，单击列表中所需的列名。
2	右键单击并选择： <ul style="list-style-type: none"> “上移列 <名称>”选项（或按 Ctrl+向上箭头）将列上移一位 “将列 <名称> 向下移动”选项（或按 Ctrl+向下箭头）将列向下移动一个位置 这些选项在“列”选项卡和图表中都会立即生效。

使用库数据库库表属性

创建数据库库表后，您可以审阅其属性并检查 DBMS 和拥有着值是否正确。要显示库表的 属性”对话框，请双击 数据库生成器表”包中的库表名称或图表上的库表元素。

重要的

A先将 DBMS 分配给库表，然后才能在其中添加列。如果您使用数据库生成器，则数据模型的 DBMS 将自动应用于所有新表；但是，如果您通过其他方式（例如处理图表）添加了库表，则这是一个手动步骤。

任务

一旦定义了数据库库表属性，您就可以添加列了。

任务
为库表设置数据库类型——除了库表名称之外，为数据库库表设置的最重要的属性是数据库类型。
设置数据库库表Owner - 对于某些DBMS，所有表必须分配一个拥有着/架构；在Enterprise Architect中，此属性定义为名称为拥有着的标记值。
设置扩展选项- 一些 DBMS 具有仅与该 DBMS 相关的扩展选项。这些扩展属性被存储为标记值。

默认数据库管理系统

在创建物理数据模型之前，建议您设置默认 DBMS，它将自动应用于您在数据库构建器之外创建的新数据库对象。您可以通过以下方式之一设置默认 DBMS 类型：

- 选择 开始>外观>首选项>首选项>源代码工程>代码编辑器”，然后设置字段 默认数据库”
- 选择 设置 >参考> 设置 >数据库数据类型”，然后选择产品名称并选中 设置为默认值”复选框
- 在代码生成工具栏的第二个字段中设置 DBMS

设置数据库类型

为数据库库表设置的最重要的属性（在其名称之后）是数据库类型或 DBMS。选择的 DBMS 值将控制 Enterprise Architect 将如何确定：

- 如何显示库表名称（有或没有拥有着）
- 数据库建模时将应用哪组验证规则
- 创建列时可用的数据类型，
- DDL 生成中将使用哪组 DDL 模板

访问

在浏览器窗口或图表中选择一个库表，然后使用此处概述的任何方法，打开库表的“属性”对话框，显示“常规”选项卡，然后显示“主”子选项卡。

功能区	设计>元素>编辑器>属性对话框>常规>主
上下文菜单	右键单击库表元素 属性 特别行动 一般 主要的
键盘快捷键	Shift+Enter 一般 主要的
其它	双击库表元素 General 主要的

选项

字段/按钮	行动
数据库	单击下拉按钮并从列表中选择所需的数据库类型。
申请	单击“应用”按钮以保存任何待处理的更改。
确定	单击确定按钮以保存任何未决更改并关闭屏幕。

设置数据库库表拥有着/架构

对于某些 DBMS，必须为所有表分配一个拥有着/架构。在Enterprise Architect中，此属性在物理上定义为名称为拥有着的标记值。但是，提供了一个特殊的属性页面，可以帮助您轻松管理拥有着的属性。

访问

在浏览器窗口或图表中选择一个库表，然后使用此处列出的任何方法，打开库表的“属性”对话框，显示“常规”选项卡并显示“库表详细信息”子选项卡。

功能区	设计>元素>编辑>属性><<表>>
上下文菜单	右键单击库表元素 属性 特殊行动>常规>库表详细信息
键盘快捷键	Shift+Enter 综合 库表详情
其它	双击库表元素 'General' '库表详情'

设置数据库库表所有者

节	行动
1	在“拥有着者”字段中，输入库表的所有者或模式的名称。

设置MySQL选项

要在MySQL中使用外键，您必须将数据库库表类型声明为 InnoDB。

将库表类型声明为 InnoDB

节	行动
1	在库表中添加一个名为类型的标记值。
2	将 值"字段设置为 InnoDB"。

生成DDL

为这个库表生成 DDL 时，库表类型包含在#脚本。

为了允许更高版本的MySQL，可以以相同方式添加的其他库表选项包括：

标签	值 (示例)
引擎	InnoDB
字符集	拉丁语1
字符集	拉丁语1
整理	latin1_german2_ci

设置 Oracle 数据库库表属性

要设置额外的 Oracle 数据库库表属性，请使用库表的标记值。

设置属性

通过突出显示索引或约束操作并添加适当的标记值，可以将相同的属性添加到索引和约束中。

节	行动
1	使用 属性"库表的 属性/标签"列中提供的名称，将一个或多个标记值添加到库表库表中。
2	<p>为每个标签指定适当的值。 此属性库表的 值"列中提供了示例。</p> <ul style="list-style-type: none">• 缓存-NOCACHE• 数据库版本 - 9.0.111• 自由清单 - 1• 授予所有者 1 - SELECT• GRANT OWNER2 - 删除、插入、SELECT、更新• 首字母 - 65536• 初始化 - 1• 记录 - 记录• MAXEXTENTS - 2147483645• MAXTRANS - 255• MINEXTENTS - 1• 监控 - 监控• 所有者 - 所有者 1• 平行 - 不平行• 无PCT - 10• PCT 增加 - 0• PCT已使用 - 0• 同义词 - PUBLIC:TABLE_PUB;OWNER2:TABLE_OWNER2• 表空间 - MY_TABLESPACE• 临时的 - 是的

数据库约束库表/索引

在Enterprise Architect中，库表约束和索引在同一屏幕上建模；它们统称为约束。数据库约束定义了强加于数据库行为的条件库表。他们包括：

- Primary Key - 唯一标识库表中的一条记录，由一列或多列组成
- 索引 - 提高对库表数据的检索和排序操作的性能
- 唯一约束- 唯一标识库表中一行的值的组合
- 外键 - 强制两个表之间的关系列（或列集合）
- 选择约束- 通过限制列接受的值来强制域完整性
- 库表触发器- #
或由于库表中的数据被修改而自动执行的代码

在Enterprise Architect中，您可以使用数据库构建器中专门设计的“约束/索引”页面或列和约束屏幕来定义和维护库表约束。

访问

功能区	开发 > 数据建模 > 数据库生成器 > 点击库表名称 > 约束/索引 右键 加新约束
上下文菜单	图中 右键单击库表 特征 约束/索引 右键 加新约束
键盘快捷键	点击库表：F9 > 约束/索引：Ctrl+N

创建一个约束

创建任何这些约束类型的过程是相同的，并且可以通过此处描述的一种方式来实现。

创建一个约束-使用上下文菜单或键盘

节	行动
1	A新的约束被自动创建并分配了默认名称约束 <i>n</i> （其中 <i>n</i> 是一个计数器）和一个“索引”的类型”。用您自己的约束名称改写默认名称。
2	如有必要，在“类型”字段中单击下拉箭头并选择适当的约束类型。
3	如果您愿意，请在“别名”字段中输入约束的别名。 “列”字段是只读的；它填充了您分配给“相关列”选项卡的列。

创建一个约束-改写模板文本

节	行动
1	在所选库表的 约束/索引”选项卡上，约束列表以模板文本新约束结束。 用适当的约束名称改写此文本，然后按 Enter 键。
2	新约束会自动创建并分配默认类型索引。 如有必要，在 类型”字段中单击下拉箭头并选择适当的约束类型。
3	如果您愿意，请在 别名”字段中输入约束的别名。 列”字段是只读的；它填充了您分配给 相关列”选项卡的列。

将列分配给约束

Primary Key、Foreign Key、Index、Unique这些约束类型都必须至少分配一列；这定义了约束中涉及的列。

节	行动
1	在所选库表的 约束/索引”选项卡上，单击要为其分配列的约束。
2	可用列”面板列出了为库表定义的所有列。 对于要分配给约束的每一列，右键单击列名称并选择 分配列 <名称>”。 列名被传送到“分配的列”列表中。

从约束中取消分配列

节	行动
1	在所选库表的 约束/索引”选项卡上，单击要取消分配列的约束。
2	在“分配的列”列表中，右键单击要从约束中取消分配的列的名称，然后选择 取消分配列 <名称>”。 列名被转移到 可用列”列表中。

重新排序约束中的指定列

如果约束中有许多列，则可以通过将选定的列名一次在列表中向上或向下移动一个位置来重新排列序列。去做这个：

- 右键单击要移动的列名称并选择：
 - 向上移动列 <名称>” (Ctrl+向上箭头) 或
 - 向下移动列 <name>” (Ctrl+向下箭头)

删除约束

要删除不再需要的约束，请右键单击“约束/索引”选项卡上列表中的约束名称，然后选择“删除约束 <名称>”选项。如果满足给定约束类型的所有验证规则，则会立即从存储库中删除该约束以及所有相关关系（如果有的话）。

主键

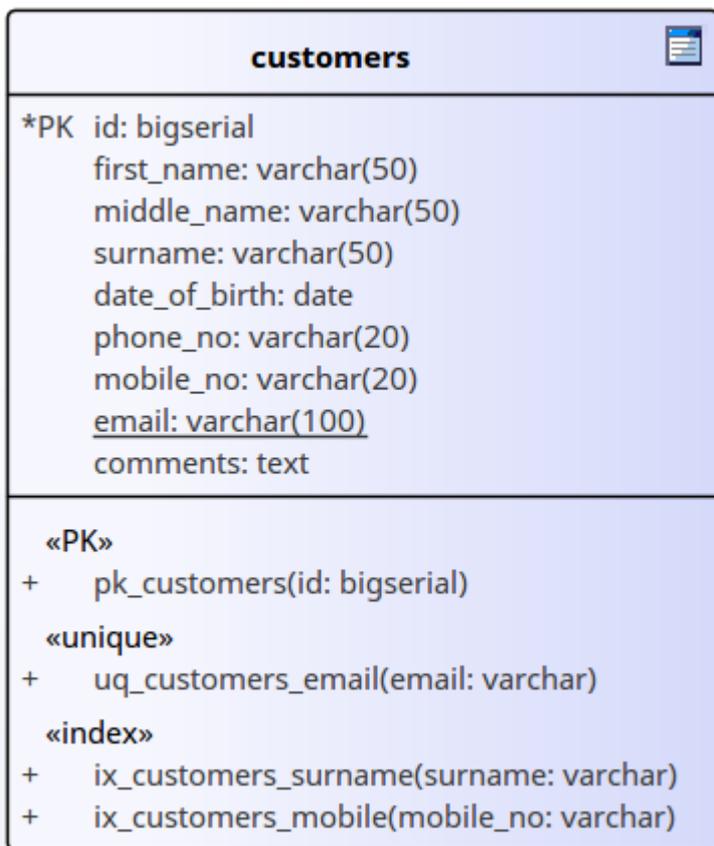
主键是唯一标识库表中每条记录A列（或列集）。A库表只能有一个主键。一些 DBMS 支持主键的附加属性，例如 Clustered 或 Fill Factor。

访问

功能区	开发 > 数据建模 > 数据库生成器 > 点击库表名称
上下文菜单	图中 右键单击库表 特征 约束/索引

创建主键

在Enterprise Architect中，您可以从“列”选项卡或“约束/索引”选项卡创建主键。无论哪种情况，当您将列添加到主键约束时，该列都会自动设置为“非空”。此外，任何包含库表元素的图表（假设设置了“显示限定词和可见性指示器”选项）都将针对列名显示“PK”前缀。在此图像中，请参见第一列“id: bigserial”。



创建主键 - 从列选项卡

节	行动
1	任何一个： <ul style="list-style-type: none"> 在数据库构建器中，单击具有一个或多个已定义列的库表，然后单击 列”选项卡，或 在图表上，单击库表并按 F9 显示 列”选项卡
2	对于要包含在主键中的每一列，选择 PK”复选框。 如果之前没有为当前库表定义主键约束，系统将使用主键名称模板创建一个新约束。

创建主键 - 从约束选项卡

节	行动
1	任何一个： <ul style="list-style-type: none"> 在数据库构建器中，单击包含一个或多个已定义列的库表，然后单击 约束/索引”选项卡，或 在图表上，单击库表并按 F10 以显示 约束/索引”选项卡
2	用主键名称改写新约束文本，按 Enter 键并单击 类型”字段下拉箭头，然后选择 PK”。
3	将所需的列分配给 PK 约束。
4	使用属性面板设置主键的扩展属性。 <ul style="list-style-type: none"> 填充因子是 0 到 100 之间的数值 Is Clustered 是一个布尔值，用于确定数据存储方式的物理顺序；对于大多数 DBMS，主键的 Is Clustered 属性默认为 True

从主键中删除列

您可以使用 列”选项卡或 约束/索引”选项卡从主键中删除列。

从主键中删除列 - 使用 列”选项卡

节	行动
1	任何一个： <ul style="list-style-type: none"> 在数据库构建器中，单击带有主键的库表，然后单击 列”选项卡，或 在图表上，单击库表并按 F9 显示 列”选项卡
2	针对要从主键中删除的每一列，取消选中“PK”复选框。 如果您已从主键约束中删除所有列并且不再需要主键，则必须手动删除它。

从主键中删除列 - 使用约束/索引选项卡

节	行动
1	任何一个： <ul style="list-style-type: none">在数据库构建器中，单击带有主键的库表，然后单击“约束/索引”选项卡，或在图表上，单击库表并按 F10 以显示“约束/索引”选项卡
2	根据需要取消分配 PK 约束上的列。

注记

- 警告：** Enterprise Architect假定主键约束至少分配了一列；但是，Enterprise Architect在建模期间不强制执行此规则
如果为主键未分配列的库表生成 DDL，则该 DDL 无效

非聚集主键

当您在某些 DBMS 中创建主键时 (例如#

服务器或ASA) , 它会在 'Is Clustered'属性设置为True的情况下自动创建。因此, 当您在Enterprise Architect数据模型中建立主键模型, 会发生相同的行为。

聚集索引通过按列物理组织数据, 为访问所涉及的列提供了改进的性能。每个库表只能有一个聚集索引。

在某些情况下, 您可能对分配给主键的列之外的列的性能更感兴趣, 因此您需要更改默认分配, 以便主键不聚集。

访问

功能区	开发 > 数据建模 > 数据库构建器 > 点击库表名称 > 约束/索引
上下文菜单	在图表或浏览器窗口 右键单击库表 特征 约束/索引
键盘快捷键	点击库表 : F9 > 约束/索引

将主键定义为非集群

随后, 您可以将模型库表的索引建模为集群。

节	行动
1	突出显示现有的主键约束。 主键属性显示在 属性"面板中。
2	对于 <i>Is Clustered</i> 属性, 在 值"字段中单击下拉箭头并将值更改为False。

数据库索引

将数据库索引应用于表以提高数据检索和排序操作的性能。一个库表可以定义多个索引；但是，每个索引都会在数据库服务器上增加开销（以处理时间和存储的形式）来维护它们，因为在库表中添加和删除信息

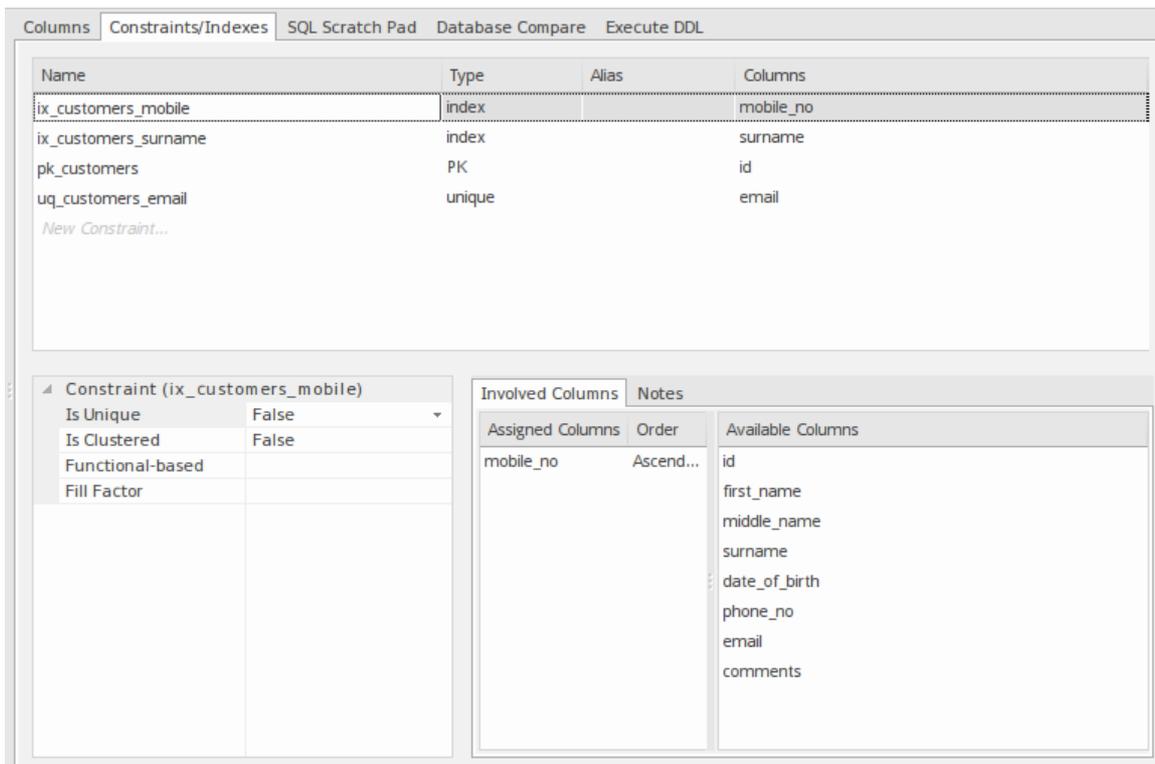
在Enterprise Architect中，索引被建模为原型操作。

一些 DBMS 支持特殊类型的索引； Enterprise Architect使用其他属性来定义这些，例如基于函数的、集群的和填充因子。

访问

功能区	开发 > 数据建模 > 数据库构建器 > 点击库表名称 > 约束/索引
上下文菜单	图中 右键单击库表 特征 约束/索引
键盘快捷键	点击库表：F9 > 约束/索引

处理索引



节	行动
1	在库表的“约束/索引”选项卡上，右键单击并选择“添加新约束”。 使用默认名称“constraint1”和“index”类型添加新约束。 用您喜欢的索引名称改写名称。

2	<p>将适当的列分配给索引。</p> <p>'Assigned Columns' 列表有一个附加的 'Order' 字段，用于指定每个已分配列存储在索引中的顺序（升序或降序）。您可以根据需要切换每列的顺序。</p> <p>此外，对于MySQL索引，“Len”字段将可见，您可以在其中定义部分索引；即，使用基于文本的字段的前导 <i>n</i> 个字符的索引。“Len”字段仅采用介于 0 和列的定义长度之间的整数数值。A 0（这是默认值）表示要索引整个列。</p>
3	<p>在 属性”面板中，审阅为当前 DBMS 定义的扩展属性的设置。</p>

附加属性

属性	描述
是独特的	<p>(True / False) 指示当前索引是否为 唯一索引”。唯一索引确保索引A（或列）不包含重复值，从而确保每一行具有唯一值（或索引由多列组成时的值组合）。</p>
是集群的	<p>(True / False) 指示当前索引是否为 聚集索引”。使用聚集索引，表的行在物理上以与索引中相同的顺序存储，因此每个表只能有一个聚集索引。默认情况下，一个表的主键是集群的。</p> <p>并非所有 DBMS 都支持聚集索引，因此“Is Clustered”索引属性仅对支持它的 DBMS 可见。</p>
是位图	<p>(True / False) 指示当前索引是否为 位图”索引。位图索引旨在用于具有相对较少唯一值的列（称为 低基数”列）并且物理上由每个唯一值的位数组（通常称为位图）组成。对于表中的每一行，每个数组都有一个位。</p> <p>考虑这个示例：在名为 性别”的列上创建位图索引，该列具有 男性”或 女性”选项。物理上，索引将由两个位数组组成，一个用于 男性”，一个用于 女性”。女性位数组在匹配行的值为 女性”的每个位中都有一个1。</p> <p>Is Bitmap 和 Is Unique属性是互斥的，所以当 Is Bitmap属性为True时，DDL 生成将忽略 Is属性。</p> <p>位图索引仅受 Oracle 支持；因此，此属性仅在对 Oracle 索引建模时可见。</p>
填充系数	<p>A介于 0 和 100 之间的数值，用于定义应用于数据的可用空间百分比。</p> <p>并非所有 DBMS 都支持填充因子，因此“填充因子”索引属性仅对支持它的 DBMS 可见。</p>
基于功能	<p>A #</p> <p>定义将被评估的函数/语句和索引结果的语句；例如： 降低（“字段”）</p> <p>并非所有 DBMS 都支持基于功能的索引，因此“基于功能的”索引属性仅对支持它们的 DBMS 可见，例如 PostgreSQL 和 Oracle。</p>
包括	<p>从当前表中标识非键列的逗号分隔列表 (CSV)。</p> <p>并非所有 DBMS 都支持索引的 包含”属性，因此该属性仅对支持它的 DBMS 可见。</p>

注记

- 警告：Enterprise Architect假定索引至少有一列分配给它们；但是，Enterprise Architect在建模期间不强制执行此规则
如果为定义了未分配列的索引的库表生成 DDL，则该 DDL 将无效，除非该索引是基于功能的
- 分配给基于功能的索引的任何列都将被忽略

独特约束

Unique约束在库表的所有行中强制一组字段的“唯一性”，这意味着库表库表中的任何两行不能在唯一约束的字段中具有相同的值。唯一约束与主键相似（因为它们也强制执行“唯一性”），但主要区别在于库表可以定义多个唯一约束但只有一个主键。

访问

功能区	开发 > 数据建模 > 数据库生成器 > 点击库表名称 > 约束/索引 > 右键单击 > 加新约束
上下文菜单	在图表或浏览器窗口 右键单击库表元素 特征 约束/索引
键盘快捷键	点击库表：F9 > 约束/索引：Ctrl+N

创建一个约束

节	行动
1	在“约束/索引”选项卡上，会自动创建一个新约束并为其分配默认约束名称和索引“类型”。使用将其标识为唯一约束的名称改写约束名称。
2	在“类型”字段中，将值从“索引”更改为“唯一”。

注记

- 警告：Enterprise Architect假定 Unique约束至少有一个列分配给他们；但是，Enterprise Architect在建模期间不强制执行此规则。如果为具有定义的唯一约束但未分配列的库表生成 DDL，则该 DDL 将无效。

外键

外键定义了A列（或列的集合），它强制两个表之间的关系。执行此关系以确保数据完整性是数据库服务器的责任。外键的模型定义由包含一组唯一数据的父（主）库表组成，然后在子（外）库表中引用该数据集。

在Enterprise Architect中，外键使用两个不同（但相关）的UML组件建模：

- 存储在子库表中A外键约束（具有 <<FK>> 原型的UML操作）
- 定义两者之间关系的关联连接器（<<FK>> 的刻板印象）表

创建外键

虽然外键的定义可能很复杂，但外键约束屏幕简化了外键的建模。此屏幕旨在帮助您选择要使用父库表中的哪个约束，并且会自动将子库表列与父库表中属于约束的列进行匹配。为了说明，这里分别描述了开发外键过程的不同方面，但整个过程应该是一个平稳的过渡。

A保存外键定义之前，必须满足许多条件：

- 两个表都必须定义匹配的 DBMS
- 父库表必须至少有一栏
- 父库表必须定义一个主键、唯一约束或唯一索引

创建外键 - 使用数据库

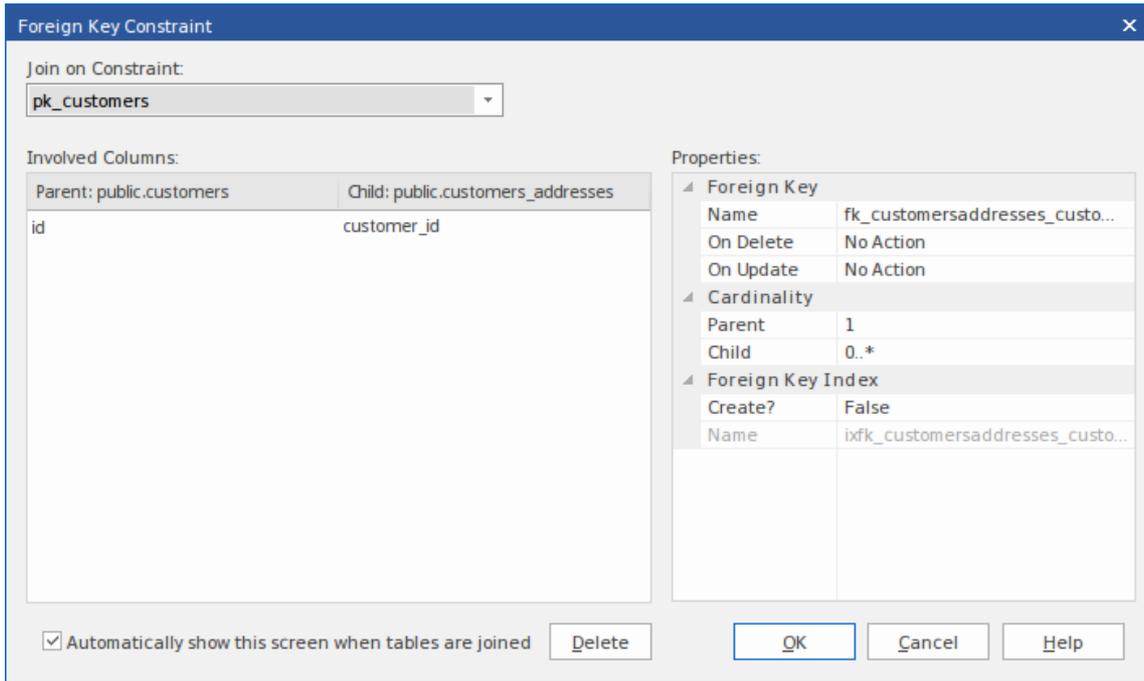
节	行动
1	在数据库构建器树中，右键单击子库表名称并单击 在 <表名称> 上添加新外键”。显示A对话框，列出所有可能的父表。
2	双击列表中所需的父库表名称或选择它并单击确定按钮。显示 外键约束”屏幕。

创建外键 - 在图表上使用关系

节	行动
1	在数据建模图中，找到需要的child（Foreign Key）库表和parent（Primary Key）库表。
2	在图表工具的“数据建模工具箱”页面中图表一个关联连接器。
3	单击子库表并将连接器绘制到父库表。
4	如果外键约束屏幕已设置为两个表连接时自动显示，则显示现在。否则，要么： <ul style="list-style-type: none"> • 双击连接器或 • 右键单击连接器并选择 外键”选项 外键约束屏幕显示。

外键约束屏幕

作为示例，此图像显示加载了“示例”详细信息的外键约束屏幕（如示例模型中所定义）。

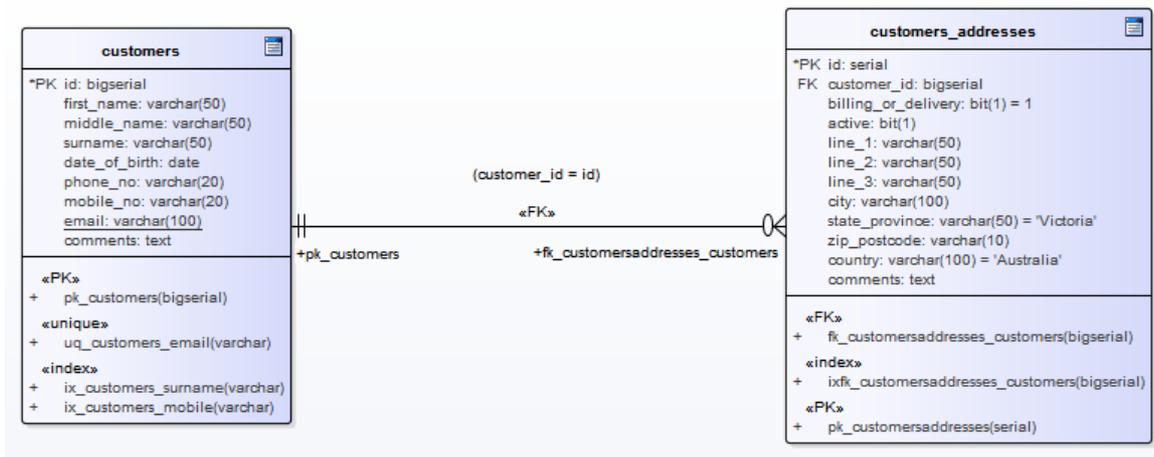


选项	行动
汇合关于约束	<p>这个组合框列出了父库表中所有定义的约束，这些约束可以用作外键的基础。（这些约束可以是主键、唯一约束或唯一索引。）</p> <p>默认选择列表中的第一个约束；如果这不是您想要的约束，请从组合框中选择正确的约束。</p> <p>当您选择约束时，它的列会自动列在“涉及的列”面板中的“父级：<tablename>”列下。</p>
涉及的列	<p>该列表分为两部分：左侧列出了所选约束中涉及的列，右侧列出了要与父列配对的子列。</p> <p>When a constraint is selected (in the '汇合 on constraint' field) the parent side is refreshed to display all columns assigned to the selected constraint.在子方面，系统将自动尝试将每个父列与子库表中的同名列匹配。如果子库表没有同名的列，则该名称的新列将被添加到列表中，用（*）标记表示将在库表中创建一个新列。</p> <p>但是，如果您想强制配对到现有的子库表列或具有不同名称的新列，请单击列名称字段，然后：</p> <ul style="list-style-type: none"> • 替换名称中的类型，或 • 选择现有列（单击下拉箭头并从列表中选择名称）
名称	<p>该字段定义外键约束的名称，默认为外键名称模板构造的名称。</p> <p>要将名称更改为默认值以外的名称，只需改写该值即可。</p>
关于删除	<p>选择删除父级数据时对子库表中的数据应采取的动作，以保持参照完整性。</p>

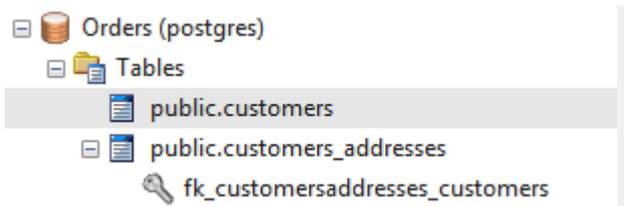
更新时	选择当父项中的数据更新时对子库表中的数据应采取的动作，以保持参照完整性。
家长	单击下拉箭头并在外键中选择父库表的基数。
子	单击下拉箭头并在外键中选择子库表的基数。
创建？	如果要在创建外键的同时创建外键索引，请将此属性设置为True。 外键索引的名称由外键索引模板控制，生成的名称显示在“创建？”下方的“名称”字段中场地。
加入库表时自动显示此画面	(对于图表建模)选中此复选框可在两个表之间创建关联时自动显示此屏幕。
删除	单击此按钮可删除当前选择的现有(已保存)外键。 将显示A提示以确认删除(以及外键索引的删除,如果存在)-单击是按钮。 删除外键会留下一个关联连接器,您可以对其进行编辑或删除(右键单击并选择“删除关联:到<库表名称>”)。
确定	单击此按钮以保存外键。

例子

此示例在图表中显示了简单的外键：



相同的外键将作为库表“customers.addresses”下的子节点显示在数据库构建器的树中。



选择约束

A选择约束通过限制列接受的值来强制域完整性。

访问

功能区	开发 > 数据建模 > 数据库生成器 > 点击库表名称 > 约束/索引 > 右键单击 > 加新约束
上下文菜单	图中 右键单击库表 特征 约束/索引 右键 加新约束
键盘快捷键	点击库表 : F9 > 约束/索引 : Ctrl+N

创建一个约束

节	行动
1	在列和约束屏幕的“约束/索引”选项卡上，会自动创建一个新约束并分配默认约束名称和索引的“类型”。 使用将约束标识为检查约束的名称改写约束名称，例如“CHK_ColumnName”（CHK_前缀是可选的）。
2	在“类型”字段中，将值从“索引”更改为“检查”。
3	在条件属性的“属性”面板中，键入# 将用作选择条件的语句；例如，第 1 列 < 1000。 如果条件很长，点击  按钮显示# 编辑器（带有语法高亮）。

删除一个约束选择

如果您不想保留检查约束，则：

- 在列表中右键单击它并选择“删除约束 <名称>”，或者
- 单击项目并按 Ctrl+D

约束被立即删除。

注记

- 分配给检查约束的任何列都将被忽略

库表触发器

库表触发器A

或者由于数据库中的数据被修改而自动执行的代码库表。触发器可高度定制，并且可以以多种不同方式使用；例如，它们可用于停止在工作时间执行某些数据库活动，或者在删除主库表中的记录时提供验证或在辅助表中执行删除。

在Enterprise Architect中，库表触发器被建模为一个定型操作并使用库表的“约束”屏幕进行管理。

访问

功能区	开发 > 数据建模 > 数据库生成器 > 点击库表名称 > 约束/索引 右键 加新约束
上下文菜单	图中 右键单击库表 特征 约束/索引 右键 加新约束
键盘快捷键	点击库表：F9 > 约束/索引：Ctrl+N

创建库表触发器

节	行动
1	在“约束/索引”选项卡上，会自动创建一个新约束并为其分配默认约束名称和索引“类型”。使用将约束标识为触发器的名称改写约束名称，例如 TRG_OnCustomerUpdate。（TRG_ 前缀是可选的。）
2	在“类型”字段中，将值从“索引”更改为“触发器”。
3	在 Statement 属性的“属性”面板中，输入完成# 将定义简单的语句（包括 CREATE 触发器）。 如果条件很长，点击  按钮显示# 编辑器（带有语法高亮）。
4	触发器时间和触发器事件属性当前是仅供参考的值，不用于 DDL 生成。

删除库表触发器

如果您不想保留触发器，则：

- 在列表中右键单击它并选择“删除约束 <名称>”，或者
- 单击项目并按 Ctrl+D

触发器立即被删除。

注记

- 分配给表触发器的任何列都将被忽略

数据库视图

数据库视图表示预定义查询A结果。与库表不同，视图是从一个或多个表（或其他视图）中的数据动态得出的。Enterprise Architect支持视图的定义，视图有没有这个语句：

创建视图 {viewName} As" 声

每当执行 DDL 生成时，系统将自动动态添加（如果缺少）。不定义此语句的优点是，当视图object被重命名时，视图定义"属性不必手动更新。

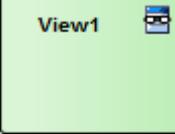
您可以创建数据库视图：

- 在数据库生成器中或
- 通过将数据建模工具箱中的 视图"图标拖到图表上

使用数据库生成器添加数据库视图

节	行动
1	打开数据库生成器。
2	加载或创建数据模型。
3	右键单击 视图"包并选择 加新视图"。
4	用视图的适当名称改写默认名称，然后按 Enter 键。
5	双击新视图，或右键单击它并选择' # 物件属性"。 ' # 显示 物件编辑器"对话框。

将数据库视图添加到图表

节	行动
1	打开您的数据建模图，如果需要，显示  工具箱数据建模图表工具箱项"对话框并指定"数据建模"）。
2	将 视图"图标拖到图表上。  View 这会生成视图元素： 

3	右键单击新的视图元素并选择'# 物件属性'。 '# 显示 物件编辑器'对话框。
---	--

#

物件编辑

'#

多个基于 SQL 的数据库对象（视图、过程、函数和序列）共享“物件编辑器”对话框；它帮助数据建模者管理基于 SQL 的 object 的各种属性。

选项	行动
数据库	如果已设置，则显示默认数据库类型。 如果未设置默认值，或者您想更改此视图的数据库类型，请单击下拉箭头并选择目标 DBMS 到模型。
依赖项	当前 object 所依赖 A 对象列表。“依赖项”列表显示： <ul style="list-style-type: none"> • 此视图与另一个库表或视图之间的每个视图连接器 • 父母“标记值”中的任何 object 名称（指定为 CSV 列表）
注记	如有必要，请在当前视图上输入评论。
定义	类型全# 视图定义。对于高达 12.1（编译 1227）的 Enterprise Architect 版本，这必须包括适用于目标 DBMS 的 CREATE_VIEW 语法（对于更高版本，这不需要）。例如： 创建视图“MyViewName”为 [查看定义] 代码编辑智能感知为基本的# 当前数据模型中所有对象的关键字、功能和名称。

数据库过程

数据库过程（有时称为存储过程或过程）是可以包含一个或多个的子例程#

执行特定任务的语句。它们可用于数据验证、访问控制或减少客户端和 DBMS 服务器之间的网络流量。广泛而复杂的业务逻辑可以嵌入到子程序中，从而提供更好的性能。

数据库过程类似于数据库函数。主要区别在于调用它们的方式 - 数据库函数的使用方式与# 中的任何其他表达式相同#

语句，而数据库过程必须使用 CALL 或 EXEC 语句调用，具体取决于 DBMS。

在Enterprise Architect中，数据库过程可以通过以下两种方式之一进行建模：

- 作为单个对象（默认方法）或
- 作为容器中的操作

从功能上讲，这两种方法会生成相同的 DDL。主要区别在于视觉 - 通过在一个容器中包含多个操作，您可以在图表上拥有更少的元素和更少的混乱。

单个对象

建模为单个对象的数据库过程是具有构造型 «procedure» 的UML类；您可以创建这些：

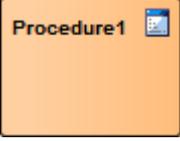
- 在数据库生成器中或
- 通过将数据建模工具箱中的“过程”图标拖到图表上

使用数据库添加数据库过程

节	行动
1	打开数据库生成器。
2	加载或创建数据模型。
3	右键单击过程包并选择“加新过程”。
4	用适当的过程名称改写默认名称，然后按 Enter 键。
5	双击新过程，或右键单击它并选择“# 物件属性”。 # 显示物件编辑器屏幕。

将数据库过程添加到图表

节	行动

1	打开您的数据建模图，如有必要，显示  工具箱数据建模图表查找工具箱项“对话框并指定“数据建模”）。
2	将 过程”图标拖到图表上。  Procedure 这会生成过程元素： 
3	右键单击新的过程元素并选择' # 物件属性”。 # 显示物件编辑器屏幕。

物件编辑

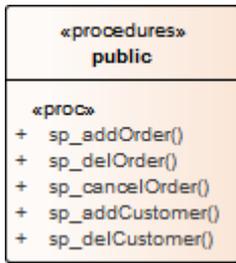
' #

多个基于 SQL 的数据库对象（视图、过程和函数）共享 物件编辑器”对话框；它可以帮助您管理基于 SQL 的 object 的各种属性。

选项	行动
数据库	如果已设置，则显示默认数据库类型。 如果未设置默认值，或者您想更改此过程的数据库类型，请单击下拉箭头并选择目标 DBMS 到模型。
笔记	如有必要，请输入对当前过程的评论。
定义	类型全# 过程定义，包括 CREATE PROCEDURE 语法。 代码编辑智能感知为基本的# 当前数据模型中所有对象的关键字、功能和名称。

在容器中操作

被建模为操作的数据库过程有一个容器 object，这是一个具有原型«procedures»（末尾带有's'）的UML类。每个数据库过程都是一个具有构造型«proc»的操作。系统提供专门的过程维护窗口，您可以通过该窗口过程管理定义为操作的数据库。



函数数据库

数据库函数为您提供了一种扩展数据库服务器功能的机制。数据库函数是接受参数、执行操作（例如复杂计算）并将该操作的结果作为值返回的例程。根据函数，返回值可以是单个值或结果集。

创建后，数据库函数可以用作#
中的表达式#
陈述。

在Enterprise Architect中，可以通过以下两种方式之一对数据库函数进行建模：

- 作为单个对象（默认方法）或
- 在容器中操作

从功能上讲，这两种方法会生成相同的 DDL。主要区别在于视觉 - 通过在一个容器中包含多个操作，您可以在图表上拥有更少的元素和更少的混乱。

单个对象

建模为单个对象的数据库函数是具有构造型 «function» 的UML类；您可以创建这些：

- 在数据库生成器中或
- 通过将函数图标从数据建模工具箱拖到图表上

使用数据库添加数据库函数

节	行动
1	打开数据库生成器。
2	加载或创建数据模型。
3	右键单击函数包并选择“加新函数”。
4	用函数的适当名称改写默认名称，然后按 Enter 键。
5	双击新函数，或右键单击它并选择“# 物件属性”。 # 显示物件编辑器屏幕。

将数据库函数添加到图表

节	行动
1	打开您的数据建模图，如有必要，显示  “工具箱数据建模图表查找工具箱项”对话框并指定“数

	据建模”)。
2	<p>将 函数”图标拖到图表上。</p>  <p>这会生成函数元素：</p> 
3	<p>右键单击新的函数元素并选择' # 物件属性”。</p> <p>#</p> <p>显示物件编辑器屏幕。</p>

#

物件编辑

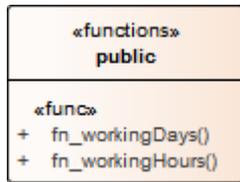
' #

多个基于 SQL 的数据库对象 (视图、过程和函数) 共享 物件编辑器”对话框；它可以帮助您管理基于 SQL 的 object 的各种属性。

选项	行动
数据库	<p>如果已设置，则显示默认数据库类型。</p> <p>如果未设置默认值，或者您想更改此函数的数据库类型，请单击下拉箭头并选择目标 DBMS 到模型。</p>
注记	<p>如有必要，请输入对当前函数的评论。</p>
定义	<p>类型全#</p> <p>函数定义包括 CREATE FUNCTION 语法。</p> <p>代码编辑智能感知为基本的#</p> <p>当前数据模型中所有对象的关键字、功能和名称。</p>

在容器中操作

被建模为操作的数据库函数有一个容器object，这是一个具有构造型«functions» (末尾带有's') 的UML类。每个函数都是一个具有构造型«func»的操作。系统提供专门的维护窗口，您可以方便地管理存储为操作的数据库函数。



数据库序列

序列是一些 DBMS 产品实现的特征，旨在为用户提供一种生成唯一值的机制 - 确保对它的每次调用都返回一个唯一值。当序列的结果用作主键时，这一点尤其重要。这些可以使用模式生成以加载到 DBMS 服务器上。

提供了序列，以便数据库用户不会被迫实现他们自己的唯一值生成器。并非所有 DBMS 产品都支持序列；那些不提供使用递增值初始化列的功能。

在Enterprise Architect中，可以通过以下两种方式之一对序列进行建模：

- 作为单个对象（默认方法）或
- 在容器中操作

从功能上讲，这两种方法会生成相同的 DDL。主要区别在于视觉 - 通过在一个容器中包含多个操作，您可以在图表上拥有更少的元素和更少的混乱。

单个对象

建模为单个对象的序列是具有构造型 «dbsequence» 的UML类；您可以创建这些：

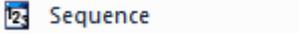
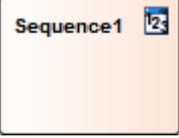
- 在数据库生成器中或
- 通过将“数据建模”工具箱页面中的“序列”图标拖到图表上

使用数据库构建器添加数据库序列

节	行动
1	打开数据库生成器。
2	加载或创建数据模型。
3	右键单击序列包并选择“加新序列”。
4	用适当的序列名称改写默认名称，然后按 Enter 键。
5	双击新序列，或右键单击它并选择“# 物件属性”。 '# 显示“物件编辑器”对话框。

将数据库序列添加到图表

节	行动
1	打开您的数据建模图，如有必要，显示  “工具箱数据建模图表查找工具箱项”对话框并指定“数据建模”。

2	<p>将 序列"图标拖到图表上。</p>  <p>这会生成序列元素：</p> 
3	<p>右键单击新的序列元素并选择' # 物件属性"。</p> <p>' #</p> <p>显示 物件编辑器"对话框。</p>

物件编辑

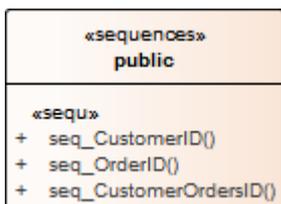
' #

多个基于 SQL 的数据库对象（视图、过程和函数）共享 物件编辑器"对话框；它可以帮助您管理基于 SQL 的 object 的各种属性。

选项	行动
数据库	<p>如果已设置，则显示默认数据库类型。</p> <p>如果未设置默认值，或者您想更改此序列的数据库类型，请单击下拉箭头并选择目标 DBMS 到模型。</p>
笔记	<p>如有必要，请在当前序列上输入注释。</p>
定义	<p>类型全#</p> <p>包括 CREATE SEQUENCE 语法的序列定义。</p> <p>代码编辑智能感知为基本的#</p> <p>当前数据模型中所有对象的关键字、功能和名称。</p>

在容器中操作

数据库Sequences 建模为操作有一个容器object，这是一个具有构造型«sequences»（末尾带有's'）的UML类。每个序列都是一个具有刻板印象«sequ»的操作。该系统提供了一个专用的维护窗口，建模者可以通过该窗口轻松管理定义为操作的序列。



#数据库#

查询

一个#

查询object提供了一种方便的机制来存储#

存储库中的语句，用于在实时数据库上重复执行。

一个#

在UML元素数据建模配置文件中以工件数据建模配置文件为代表的<查询元素查询>>。您可以创建这些元素：

- 在数据库生成器中或
- 通过拖动'# 数据查询数据建模工具'工具箱页面中的 图标"到图表中

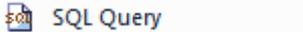
添加数据库#

使用数据库生成器查询

节	行动
1	打开数据库生成器。
2	加载或创建数据模型。
3	右键单击查询包并选择'加新#' '查询'。
4	用查询的适当名称改写默认名称，然后按 Enter 键。
5	右键单击新元素并选择 编辑”。 将显示 'SQL便签本'选项卡，您可以在其上创建# 查询语句。
6	当你完成# 声明，点击保存到# 工具栏中的按钮以保存对查询元素的更改。

将数据库函数添加到图表

节	行动
1	打开您的数据建模图，如果需要，显示  工具箱数据建模图表工具箱项"对话框并指定"数据建模")。

<p>2</p>	<p>拖动' # '查询'图标到图表上。</p>  <p>这会生成# 查询工件元素</p> 
<p>3</p>	<p>双击新元素并根据需要更新元素名称和其他属性。</p> <p>编辑元素的# 语句，访问数据库构建器，单击查询包中的元素并在“SQL便签本”选项卡上编辑查询。</p>

创建操作容器

虽然建模数据库函数、过程和序列的默认方法是将它们创建为单独的元素，但您也可以将每种类型的结构的数量表示为容器类的操作。您向类添加一个构造型，它指定：

- 类将包含的数据结构类型
- 将自动分配给类中创建的每个操作的构造型（对于给定的数据结构，操作只能是一个构造型）

访问

工具箱	将“类”图标拖到图表上
-----	-------------

创建容器类

节	行动
1	右键单击图表上的类元素并选择 设计>元素>编辑对话框>属性”选项。 显示元素 属性”对话框，显示 常规”选项卡。
2	在 名称”字段中，输入容器的适当名称。
3	在 构造型”字段（在对话框右边缘的表中）输入： <ul style="list-style-type: none"> • 数据库函数容器的“函数” • 存储过程容器的 程序” • 序列容器的“序列” 原型名称末尾的 s”很重要。
4	单击确定按钮保存设置并关闭对话框。

创建数据库结构作为类的操作

节	行动
1	单击图表上的类元素，然后按 F10。 将显示 数据库<Structure> 容器：<Classname>”对话框。
2	右键单击 函数”（ 过程”或 “列”）列表并选择 加新<结构>”。
3	在 名称”字段中，为操作键入适当的名称，例如： <ul style="list-style-type: none"> • fn_WorkDays • sp_AddOrder 或

	<ul style="list-style-type: none">• seq_AddressID
4	在“注记”字段中输入任何支持性评论或操作说明。 在“函数定义”字段（或“过程定义”或“序列定义”）中键入适当的文本。
5	重复步骤 2 到 4，直到您定义了所需的所有操作。
6	单击列表，然后单击关闭按钮关闭对话框并在图表和浏览器窗口中显示类中的操作。

甲骨文包

Oracle包是 Oracle DBMS 独有的数据库对象。它们是将逻辑相关的对象组合成一个定义的容器。包有两个部分——规范和主体。这：

- 规范部分声明了各种组件
- 体部分提供了组件的完整定义

包组件可以由类型、变量、常量、异常、游标和子程序组成。

在Enterprise Architect中，Oracle包被建模为具有 <<package>> 构造型的UML类。它有两个操作：

- 规格
- 体

对于这些操作中的每一个，完成#

语法包含在“初始代码”字段中。

创建 Oracle包

节	行动
1	将类元素添加到您的数据模型中。
2	打开元素的属性窗口，并在“构造型”字段中键入值“包”。
3	单击元素并按 F10，在“操作”页面显示特征窗口。 对于包规范，按 Ctrl+N 并创建一个名为“规范”且没有返回类型的操作。
4	属性窗口显示操作的属性；单击“代码”选项卡，然后在文本面板中输入整个包规范。
5	返回“操作”页面的特征窗口，对于包体，按 Ctrl+N 并创建一个名称为“体”且没有返回类型的操作。
6	在属性窗口上，单击“代码”选项卡，然后在文本面板中输入整个包体代码。

数据库连接

数据库object提供了A方便的方式来存储实时数据库的连接详细信息。Enterprise Architect支持定义许多不同的连接类型：

- MS Access
- 火鸟
- SQLite (在Enterprise Architect v16 中引入)
- 本机连接 (在Enterprise Architect v16 中引入)，以及
- ODBC

对于基于文件的连接 (MS Access、火鸟和SQLite)，您只需指定数据库文件的完整路径。对于本机连接，系统将提示您输入数据库服务器的连接详细信息。对于ODBC类型的连接，系统会提示您从计算机上的预定义ODBC DSN列表中进行选择。

创建数据库连接元素

A数据库元素在UML数据建模配置文件中表示为 < 一个工件元素与立体连接>>。您可以创建这些：

- 在数据库生成器中或
- 通过将“数据库连接”图标从“数据建模”工具箱页面拖到图表上

使用数据库添加数据库连接

节	行动
1	打开数据库生成器。
2	加载或创建数据模型。
3	右键单击 Connections包并选择“加新DB Connection”。
4	用适当的 Connection 名称改写默认名称，然后按 Enter 键。
5	双击新连接，或右键单击它并选择“数据库连接属性”。 将显示“数据库连接属性”对话框。

将数据库连接添加到图表

节	行动
1	打开您的数据建模图，如有必要，显示  “工具箱数据建模图表查找工具箱项”对话框并指定“数据建模”。
2	将“数据库连接”图标拖到图表上。

	 <p>Database Connection 这会生成数据库连接元素。</p>
3	<p>双击新元素。 将显示“数据库连接属性”对话框。</p>

连接数据库属性

选项	行动
DBMS类型	<p>单击相应类型的单选按钮：</p> <ul style="list-style-type: none"> MS Access基于文件的数据库 基于火鸟文件的数据库 基于 SQLite 文件的数据库 直接本机连接，或 基于 ODBC 的数据库 <p>保存密码？复选框仅对 ODBC 连接类型启用，并指示Enterprise Architect是否应存储所选 ODBC DSN 的密码。复选框默认选中；即，保存密码。虽然所有连接密码在保存之前都已加密，但有时数据建模者希望将访问权限限制为仅具有所需权限的用户。</p>
文件名/DSN	<p>如果您选择了MS Access或火鸟的“DBMS类型”，请输入或浏览物理文件的位置和名称。如果该文件尚不存在，它将被创建。</p> <p>如果您选择了 ODBC 的“DBMS类型”，请输入或选择定义的 ODBC DSN。根据 DBMS，可能会提示您输入其他详细信息，例如服务器、连接用户 ID 和密码。</p>
其它模式	<p>此字段充当模式过滤器，以限制对 ODBC 连接进行的查询返回的对象数。由于大量的系统对象，在此字段中输入一个值对于 Oracle 数据库来说尤其重要，因为它可以减少连接到数据库所需的时间。</p> <p>如果您需要输入多个要过滤的模式，请用逗号分隔它们。</p>
确定	<p>单击此按钮以保存您所做的更改。</p>

删除连接

如果不再需要连接，您可以将其删除，就像从数据库构建器、浏览器窗口或图表中删除任何其他元素一样。右键单击元素并选择相应的“删除<元素名称>”选项。

注记

- 建议在团队环境中工作（即，多个用户共享一个Enterprise Architect存储库）时，所有基于 ODBC 的数据库连接对象都定义为“无 DSN”，以便数据库连接object包含所有必要的详细信息，因此可以在所有用户之间共享，虽然本机连接这样做并且更容易设置
- 保存初始选择后，无法更改数据库连接object的 DBMS 类型

管理 DBMS 选项

使用“管理 DBMS 选项”对话框，您可以快速更改单个数据库object或单个包或包层次结构中的多个对象的 DBMS类型和/或拥有着。您还可以在所有还没有索引的外键上创建批量外键索引。

访问

功能区	设计>包>管理> DBMS选项 开发>数据建模>数据库生成器>在需要的数据库上右击 加载 右键单击根 管理 DBMS 选项
-----	---

选项

选项	行动
包	在您当前正在处理的浏览器窗口中显示包的名称。如有必要，单击  按钮并使用导航员窗口（ 查找包”对话框的一个版本）选择不同的包。
在子包中包含对象	选中此复选框以在所有子包中包含所有数据库对象。选择或取消选择此控件将立即刷新对象列表。
对象列表	此列表控件将显示当前包（或包层次结构）中所有对象的名称及其分配的 DBMS 和所有者。默认情况下，每当加载或刷新列表时，每个object都会选中其复选框。
全部	单击此按钮可选择 “对象列表”中所有取消选中的复选框。
没有任何	单击此按钮可取消选择 “对象列表”中所有选中的复选框。
更改数据库管理系统	如果要更改包中对象的分配 DBMS，请选中此复选框。为 “当前 DBMS”和“新 DBMS”字段提供值以继续。 “当前 DBMS”下拉列表包括选项 “All”，它将几个不同的 DBMS 值全部更改为新值。 注记：执行此函数时，所有库表列的数据类型会自动转换为与所选 DBMS 最接近的匹配；因此，您应该在运行该过程后对数据类型进行手动审阅。
更改拥有着	如果要更改 “对象列表”中选定对象的拥有着者，请选中此复选框。在 “当前拥有着拥有着”栏内指定当前拥有着者以继续。将 “新拥有着”字段留空将删除所有选定对象的拥有着属性。
在外键上创建索引	选中此复选框可在包中的所有外键上创建一个索引，其中一个尚不存在。
确定	单击此按钮开始更新过程。除非选择了列表中的至少一个object和其中一个更新选项，否则该按钮将被禁用。

数据类型

您在数据模型中定义的每个库表列都分配了一个数据类型，该数据类型指定了该列可以存储的信息类型。列的可用数据类型取决于为库表选择的 DBMS，因为每个 DBMS 都支持自己的数据类型列表。虽然每个 DBMS 都支持相同的基本类型，例如string、整数或十进制数，但每个 DBMS 以不同的名称调用它们并具有不同的属性。

每个Enterprise Architect存储库都包含许多标准 DBMS 产品的核心数据类型的定义。然而，由于数据类型从一个 DBMS 产品到另一个，从一个产品版本到另一个版本，Enterprise Architect为您提供以下工具：

- 为新版本的 DBMS 产品定义新数据类型
- 为新的非标准数据库产品定义数据类型
- 自动将数据类型从一种已定义的 DBMS 产品转换为另一种
- 在存储库之间导入和导出数据类型

DBMS 产品之间的映射数据类型

虽然建模物理数据模型提供了关于所有表及其列的大量细节，但这一级别的细节确实使更改目标技术或平台变得更加困难。例如，在将数据库逆向工程为物理数据模型之后，您必须在为新的 DBMS 产品生成模式之前重新映射数据类型。

Enterprise Architect为标准的、受支持的 DBMS 产品提供了一组默认映射，以帮助您自动化转换过程。

但是，您可能希望自定义默认映射以满足您的特定项目要求，或者当前未定义一种数据类型到另一种数据类型的映射。例如，在从一个 DBMS 平台迁移到另一个平台时，其中一个平台可能是非标准的，或者Enterprise Architect不支持。

访问

功能区	设置>参考>设置>数据库数据类型：Datatype映射
-----	-----------------------------

数据库数据类型映射

对每个要映射的数据类型重复此过程。

一旦您对数据类型映射感到满意，您就可以将单个表或整个表包转换为新的目标 DBMS 产品。

字段/按钮	行动
从产品名称	单击下拉箭头并选择要从中映射数据类型的 DBMS 产品。
为数据库定义的数据类型	显示产品的所有已定义数据类型，并在适当的情况下显示它们的大小和值。 单击要映射的数据类型 - 这必须具有定义的大小单位和值。 来自产品名称"字段下的 "Datatype "和 公共类型"字段显示此数据类型。
至产品名称	单击下拉箭头并选择要将数据类型映射到的 DBMS 产品。 此字段下的" Datatype "和 公共类型"字段显示 发件人"产品的字段中的值相对应的值。
尺寸	单击相应尺寸单位的单选按钮并在相应的数据字段中键入默认值。
节省	单击此按钮以保存映射。

DBMS 产品转换为一个包

使用 DBMS包映射器，您可以自动将数据库表包从一种支持的 DBMS 类型转换为另一种支持的 DBMS 类型。您还可以更改单个表的 DBMS 类型。

如果其中一种 DBMS 类型是非标准的，或者Enterprise Architect不支持，您应该检查数据类型从一种 DBMS 类型到另一种的映射是否已经定义。

访问

功能区	设计>包>管理> DBMS选项 开发>数据建模>数据库生成器>在需要的数据库上右击 加载 右键单击根 管理 DBMS 选项
-----	---

将一个包的 DBMS 数据类型映射到另一个 DBMS 的数据类型

字段/按钮	行动
在子包中包含对象	如果子包中的对象也需要更改，请选中该复选框。
更改数据库管理系统	选中复选框。
当前的数据库管理系统	单击下拉箭头并选择当前 DBMS。
新的数据库管理系统	单击下拉箭头并选择目标 DBMS。
确定	单击此按钮将所选包中的所有表映射到新的 DBMS。

库表的数据类型转换

一旦在Enterprise Architect图表上设置了数据库模式（通过 ODBC 导入或手动设置表），可以将 DBMS 更改为另一种类型，并为每个库表相应地映射列数据类型。

如果您从其他地方将少量表复制到项目中，则可以使用此过程，但如果您有很多表，您可以在其父包中一次转换所有表。

如果其中一种 DBMS 类型是非标准的，或者Enterprise Architect不支持，您应该检查数据类型从一种 DBMS 类型到另一种的映射是否已经定义。

将一个库表的 DBMS 类型映射到另一个 DBMS 类型

节	行动
1	双击图表中的库表元素。 库表 属性”对话框显示，“数据库”字段显示此库表的当前 DBMS。
2	要将数据类型映射到另一个 DBMS，请单击 数据库”下拉箭头并选择目标 DBMS。
3	单击应用按钮。 转换数据类型以匹配新 DBMS 的数据类型，这些数据类型反映在从该库表生成的任何 DDL 中。

数据库数据类型

使用Enterprise Architect的“数据库数据类型”对话框，您可以添加可用于特定 DBMS 的数据类型集。您可以：

- 识别正在使用的 DBMS，如果需要，将其设置为模型默认值
- 包括由更高版本的 DBMS 支持但尚未包含在Enterprise Architect中的任何新数据类型
- 删除任何先前添加的不再相关的数据类型
- 添加一个新的 DBMS 产品及其内置数据类型，例如，如果您想为Enterprise Architect本身不支持的 DBMS 产品创建物理数据模型

访问

功能区	设置 > 参考 > 设置 > 数据库数据类型或 开发 > 数据建模 > 数据类型
-----	---

管理数据类型

您可以使用“导出参考”和“导入参考”选项在Enterprise Architect模型之间传输这些数据库数据类型。

字段/按钮	行动
产品名称	单击下拉箭头并选择现有的 DBMS。 选择产品后，所有已定义的数据类型将显示在“已定义的数据库数据类型”列表中。
添加产品	如果您的 DBMS 未列出，请单击此按钮添加它。 将显示“输入”提示，您可以在其中键入 DBMS 名称；单击确定按钮将名称添加到下拉列表中。
设为默认	选中复选框以将选定的 DBMS 设置为数据库工程和建模的默认值。 设置默认数据库后，当您创建任何新的库表元素时，数据库类型会自动预设为此默认值。 您还可以在代码生成工具栏的第二个数据输入字段中设置默认数据库类型。
新的	单击此按钮可清除对话框中的数据类型字段，以便您可以定义其他数据类型。
Datatype	类型数据类型的名称。
尺寸	为所需大小选择适当的单选按钮，并在适当的情况下指定默认值和最大值： <ul style="list-style-type: none"> • 无 - 对于没有大小组件的数据类型，例如 INT • 长度 - 对于需要定义长度的单一大小的数据类型，例如 VARCHAR(10) • Precision & Scale - 适用于需要两个数值的数据类型，例如 DECIMAL(18,2)
公共类型	单击下拉箭头并选择每种数据类型的通用名称。这在库表的 DBMS 发生更改

	时使用。
节省	单击按钮立即将您的数据类型保存到存储库（并将其添加到 为数据库定义的数据类型“列表中”）。
为数据库定义的数据类型	此面板列出当前为选定 DBMS 定义的数据类型，系统提供的或用户定义的。
删除	在“Defined Datatypes for Databases”列表中选择一种数据类型，然后单击此按钮以删除该数据类型。
Datatype映射	如果您已将已定义数据类型的 DBMS 或技术更改为不受支持的 DBMS 类型或更改为不受支持的 DBMS 类型，请单击此按钮以定义如何将数据类型自动重新映射到新的 DBMS 或技术。

MySQL数据类型

MySQL支持 ENUM 和 SET 数据类型，必须将它们添加到Enterprise Architect模型中，然后才能将它们用作列的类型。

访问

功能区	设置 > 参考 > 设置 > 数据库数据类型
-----	------------------------

为MySQL添加 ENUM 和 SET 数据类型

稍后在列的“初始”字段中使用这些数据类型时，将值键入为逗号分隔的列表，格式如下：

('一二三')

如果一个值是默认值，请使用以下格式：

('一','二','三') 默认'三'

节	行动
1	将显示“数据库数据类型”对话框。
2	在“产品名称”字段中选择“MySQL”。
3	添加数据类型 ENUM 和 SET。

Oracle数据类型

Oracle 数据类型 NUMBER 和 VARCHAR 具有您可以模型的附加属性。

访问

功能区	设置 > 参考 > 设置 > 数据库数据类型
-----	------------------------

数据类型

数据类型	细节
数字	<p>NUMBER 数据类型需要精度和比例属性。</p> <p>当数据类型设置为NUMBER时，“Precision”和“Scale”字段显示在特征窗口的“Attributes”页面上；如果您在这些字段中输入信息，它会显示在您的图表上。</p> <p>例如：</p> <p>通过设置 'Precision' = 0 和 'Scale' = 0 创建 NUMBER</p> <p>通过设置 'Precision' = 8 和 'Scale' = 0 创建 NUMBER(8)</p> <p>通过设置 'Precision' = 8 和 'Scale' = 2 创建 NUMBER(8,2)</p>
VARCHAR	<p>Oracle VARCHAR2(15 CHAR) 和 VARCHAR2(50 BYTE) 数据类型可以通过添加标记值LengthType 值CHAR 或BYTE 来创建。</p>

数据建模设置

Enterprise Architect提供数据建模设置，可用于配置数据库系统在Enterprise Architect中建模的方式。其中包括定义数据建模语言的能力，这决定了连接器的显示方式，以及配置主键、外键和索引命名的设置。这些设置是全局的，将影响任何Enterprise Architect存储库。

访问

功能区	开始>外观>首选项>首选项>源代码工程>代码编辑器> DDL
-----	--------------------------------

DDL 编辑器

在此字段中，您可以浏览Enterprise Architect用于打开由其生成DDL 功能创建的文件的外部程序的完整执行文件路径和名称。如果您将此字段留空， Enterprise Architect将使用默认代码编辑器。

默认数据库

在此字段中，您选择将自动分配给在数据模型工作空间之外创建的数据库对象的帮助（请参阅模式模型模型主题创建数据）。

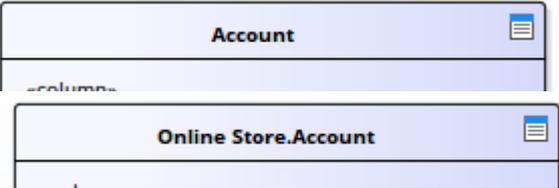
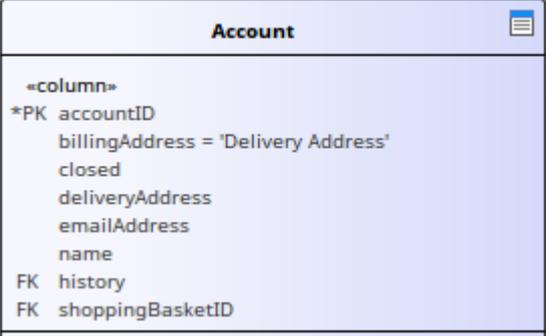
MySQL储存

在此字段中，您选择要分配给MySQL的默认存储引擎表；从MySQL v 5.5 开始，默认值为 InnoDB。

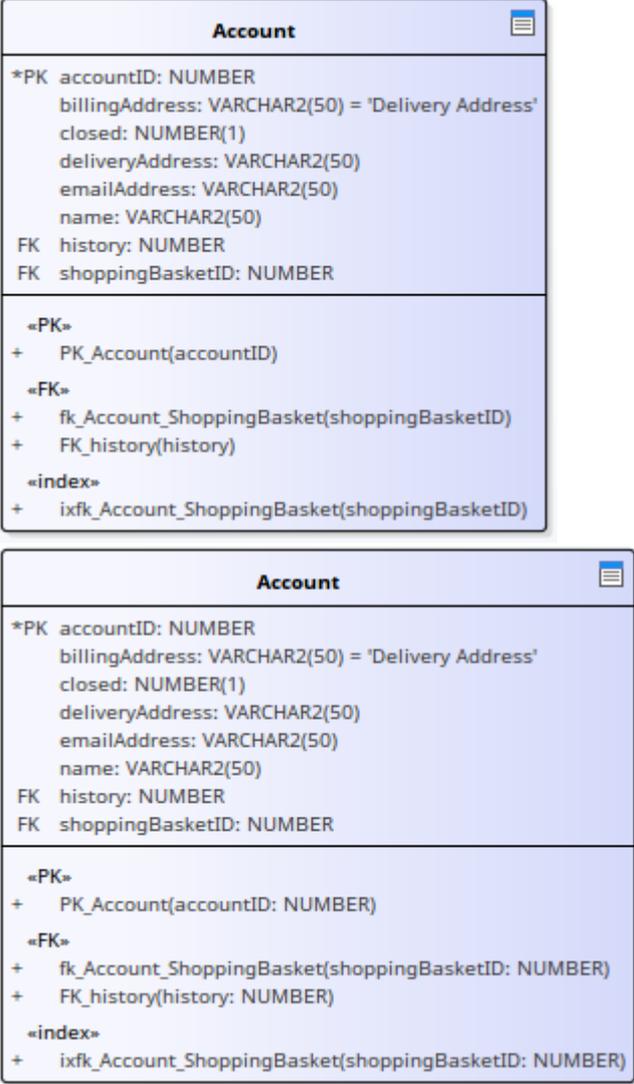
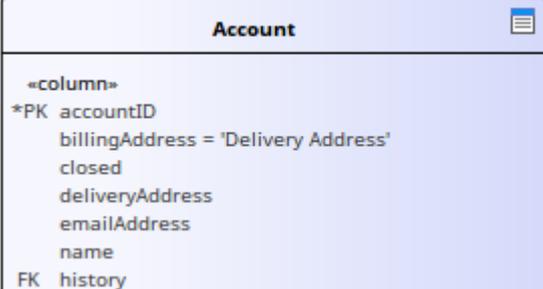
数据建模Notations

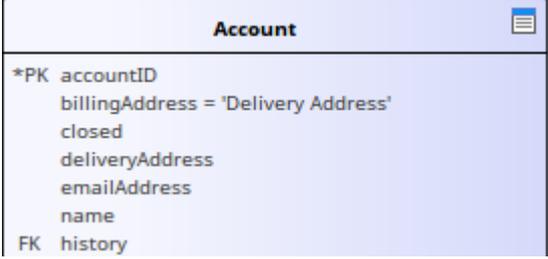
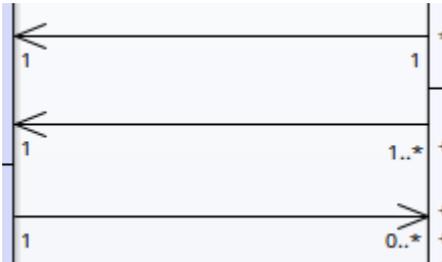
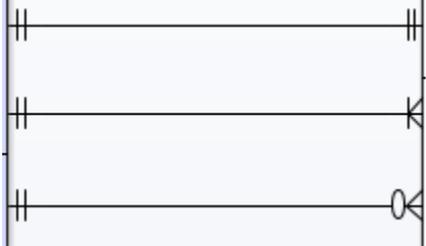
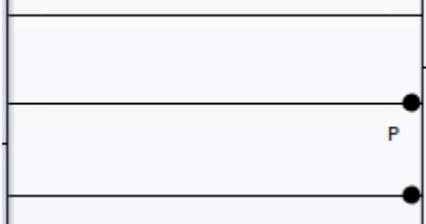
Enterprise Architect支持许多与数据建模相关的设置，这些设置会影响数据库对象在图表上的表示方式。此处描述了这些设置以及它们如何影响数据库对象的表示。

设置

环境	细节
构造型图标	<p>访问：'设计>图表>管理>属性>元素：使用构造型图标'</p> <p>默认值：True</p> <p>Enterprise Architect为原型对象的显示提供了图表级别的设置。当复选框被选中时，图表上的数据库对象将显示一个图标，代表它们的原型而不是原型名称。</p> 
显示数据模型拥有着	<p>访问：'设计>图表>管理>属性>元素：显示数据模型拥有着'</p> <p>默认值：True</p> <p>系统为所有者的显示提供了图表级别的设置。当复选框被选中时，当前图表上的数据库对象将以其全名 '{拥有着.}ObjectName' 显示。</p> 
显示列详细信息	<p>访问：'设计>图表>管理>属性>特征：显示属性详细信息'</p> <p>默认值：名称仅</p> <p>系统为库表列名和数据类型的显示提供了图表级设置。可用选项有：'仅名称'或'名称和类型'。</p> 

	<div data-bbox="520 197 1069 533"> <p style="text-align: center;">Account</p> <hr/> <p>«column» *PK accountID: NUMBER billingAddress: VARCHAR2(50) = 'Delivery Address' closed: NUMBER(1) deliveryAddress: VARCHAR2(50) emailAddress: VARCHAR2(50) name: VARCHAR2(50) FK history: NUMBER FK shoppingBasketID: NUMBER</p> </div>
<p>显示涉及的列详细信息</p>	<p>访问：'设计>图表>管理>属性>特征 显示参数详情' 默认值：仅类型 系统为库表约束中涉及的列的显示提供了图表级别的设置。可用选项有：“无”、“仅类型”、“仅名称”和“完整详细信息”。 在这些示例中，主键 (PK) 约束 PK_account“涉及列”accountID”。</p> <div data-bbox="520 772 1157 1339"> <p style="text-align: center;">Account</p> <hr/> <p>*PK accountID billingAddress = 'Delivery Address' closed deliveryAddress emailAddress name FK history FK shoppingBasketID</p> <hr/> <p>«PK» + PK_Account() «FK» + fk_Account_ShoppingBasket() + FK_history() «index» + ixfk_Account_ShoppingBasket()</p> </div> <div data-bbox="520 1355 1069 1892"> <p style="text-align: center;">Account</p> <hr/> <p>*PK accountID: NUMBER billingAddress: VARCHAR2(50) = 'Delivery Address' closed: NUMBER(1) deliveryAddress: VARCHAR2(50) emailAddress: VARCHAR2(50) name: VARCHAR2(50) FK history: NUMBER FK shoppingBasketID: NUMBER</p> <hr/> <p>«PK» + PK_Account(NUMBER) «FK» + fk_Account_ShoppingBasket(NUMBER) + FK_history(NUMBER) «index» + ixfk_Account_ShoppingBasket(NUMBER)</p> </div> <p style="text-align: right;">列2.png</p>

	 <p>Account</p> <p>*PK accountID: NUMBER billingAddress: VARCHAR2(50) = 'Delivery Address' closed: NUMBER(1) deliveryAddress: VARCHAR2(50) emailAddress: VARCHAR2(50) name: VARCHAR2(50) FK history: NUMBER FK shoppingBasketID: NUMBER</p> <p>«PK» + PK_Account(accountID)</p> <p>«FK» + fk_Account_ShoppingBasket(shoppingBasketID) + FK_history(history)</p> <p>«index» + ixfk_Account_ShoppingBasket(shoppingBasketID)</p> <p>Account</p> <p>*PK accountID: NUMBER billingAddress: VARCHAR2(50) = 'Delivery Address' closed: NUMBER(1) deliveryAddress: VARCHAR2(50) emailAddress: VARCHAR2(50) name: VARCHAR2(50) FK history: NUMBER FK shoppingBasketID: NUMBER</p> <p>«PK» + PK_Account(accountID: NUMBER)</p> <p>«FK» + fk_Account_ShoppingBasket(shoppingBasketID: NUMBER) + FK_history(history: NUMBER)</p> <p>«index» + ixfk_Account_ShoppingBasket(shoppingBasketID: NUMBER)</p> <p>列3.png</p>
<p>显示列构造型</p>	<p>访问：开始> Application > Preferences > Preferences > Objects: Show <<column>> stereotype</p> <p>默认值：True</p> <p>Enterprise Architect提供了一个全局级别的设置来控制<<column>> 原型是否显示在每个库表的列上方。因此，如果您愿意，可以隐藏构造型，考虑到构造型为 <<column>> 的属性是表的唯一有效选项。</p>  <p>Account</p> <p>«column» *PK accountID billingAddress = 'Delivery Address' closed deliveryAddress emailAddress name FK history</p>

	
连接器表示法	<p>访问：'设计>图表>管理>属性>连接器：连接器符号' 默认值：UML 2.1</p> <p>Enterprise Architect支持数据建模的三种图表符号：</p> <ul style="list-style-type: none">• UML 2.1 - 用于连接器的标准UML 2.1符号  <ul style="list-style-type: none">• 信息工程 - 信息工程 (IE) 连接方式  <ul style="list-style-type: none">• IDEF1X - 集成定义方法 IDEF1X 连接样式  <p>(这些是使用不同符号的相同三个连接器。) 数据建模模型图的默认符号是“信息工程”，而从工程创建的模型的默认模式是“数据库”。</p>

DDL名称模板

在数据建模过程中的不同时期，Enterprise Architect都需要自动生成库表约束。这些生成的约束的命名标准在DDL名称模板中定义和应用，您可以随时自由更改。这些名称模板是在存储库级别定义的，因此无论何时更改它们，存储库的所有用户都将使用新模板。

访问

功能区	设置>模型>选项>源代码工程：DDL名称模板 
-----	--

DDL名称模板

选项	行动
首要的关键	定义创建主键约束时使用的名称模板。
独特约束	定义创建唯一约束时使用的名称模板。
外键	定义创建外键约束时使用的名称模板。
外键索引	定义创建外键索引时使用的名称模板。
节省	单击此按钮以保存您定义的名称模板。

模板宏

在创建约束名称期间，这些已识别的宏将被名称文本替换。

宏	适用于
%表名%	首要的关键 独特约束 描述：由库表名称替换的string。
%列名%	由约束的列名替换的string。
%主表名%	外键 描述：被主（父）库表名称替换的string。
%foreigntablename%	由外部（子）库表名称替换的string。
%外键名%	外键索引

	描述：被外键名称替换的string。
--	--------------------

导入数据库架构

基于模型的工程的力量在于可视化、分析和设计系统各个方面的能力。能够与系统的其他模型一起查看数据库模式提供了极大的清晰度并减少了出错的机会。Enterprise Architect可以将 DBMS 模式及其对象逆向工程为多种不同标准下的模型，包括UML、信息工程和 IDEF 1X。支持范围广泛A数据库对象，包括表、视图、过程和函数。Enterprise Architect通过询问 DBMS 的信息模式并将定义导入UML对象来实现这一点。随着对 Live 数据库的修改，可以将更改同步到模型中。

一旦架构在Enterprise Architect中，数据库对象就可以追溯到其他元素，确保设计和架构的完整性。当系统以多个 DBMS 为目标时，这些都可以逆向工程到模型中，并且可以在这些模型之间比较元素和数据类型。复杂的报告引擎可以生成高质量的文档，包括数据字典、图表和关系，返回到其他模型，如架构和信息需求，并最终返回到业务目标和驱动程序。

可以通过数据库生成器（推荐）或 开发”功能区导入数据库架构信息。

导入数据库架构

节	行动
1	打开数据库生成器（开发 >数据建模>数据库生成器）
2	加载或创建数据模型。
3	<ul style="list-style-type: none"> • 右键单击数据库构建器中加载的数据模型并选择 “导入数据库模式”或 • 从功能区中选择 开发 >数据建模>导入” 将显示 “导入数据库架构”对话框，显示当前活动数据库连接的详细信息。

导入数据库架构对话框

选项	描述
数据库	此字段显示当前实时连接的描述，格式为： dbms.database_server.database_name 如有必要，单击  按钮并选择替代连接。
导入到	此字段显示将保存新对象的目标包。 如果要指定不同的包，请单击  按钮并选择替代包。
仅包括来自架构的对象	如果数据库类型支持多种模式（如# 服务器、Oracle、PostgreSQL 和 DB2 Express），您可以按模式过滤要从数据库中检索的对象。 此面板中会自动列出可用的模式。选中要包含在导入中的每个架构的复选框。 （您可以单击 “全部”按钮选择所有模式，或单击 “无”按钮清除所有选中的复选框。） 如果您怀疑模式列表在加载后可能已更改，则可以通过单击 重新加载模式

	”按钮刷新列表。
名称过滤器	<p>名称过滤器：“字段允许使用# 过滤对象# 适用于正在导入的模式的 DBMS 的通配符。 例如，对于 Oracle：</p> <ul style="list-style-type: none"> • LIKE 'A%' - 列出名称以字母 'A' 开头的对象 • NOT LIKE '%_%' ESCAPE '\' - 列出名称不包含下划线 () 的对象 • IN ('TABLE1','TABLE2') - 列出括号中包含名称的对象 • NOT IN ('TABLE1','TABLE2') - 列出名称未包含在括号中的对象 <p>注记只能输入一个过滤器。您不能使用 AND 子句添加第二个过滤器。 过滤不适用于MS Access</p>
过滤器选项	<p>过滤器选项”面板控制从数据库模式中读取的object类型和属性。在此屏幕上更改的值会保存到注册表中，以便在下一个工作会话中重新应用。此处简要描述了可用的选项；选中选项对应的复选框以激活它。</p> <p>表</p> <ul style="list-style-type: none"> • 表- 选择导入表 • 库表Primary Keys - 选择在表上导入主键定义 • 库表外键 - 选择在表上导入外键定义 • 库表索引 - 选择导入库表索引 • 唯一约束- 选择导入表上的唯一约束定义 • 选择约束- 选择导入表上的选择约束定义 • 库表触发器- 选择导入触发器定义表 • 库表属性-选择导入扩展库表属性 • 约束属性- 选择为表导入约束属性 • 长度语义 - 选择以在 Oracle string列上导入长度语义定义 <p>对象</p> <ul style="list-style-type: none"> • 视图- 选择导入视图 • 过程- 选择导入过程 <ul style="list-style-type: none"> - 作为操作- 选择导入过程作为单个类的操作（方法）；你可以通过数据库object容器 属性”对话框查看和编辑它们（该选项默认为到未选择，其中选定的项目作为单独的类导入） • 函数- 选择导入函数 <ul style="list-style-type: none"> - 作为操作- 选择导入函数作为操作（默认为未选中） • 序列 - 选择以导入序列 <ul style="list-style-type: none"> - 作为操作- 选择将序列导入为操作（默认为未选中） • 包-选择导入Oracle包 <p>高级</p> <ul style="list-style-type: none"> • 系统对象 - 选择导入系统表、视图等系统对象 <p>警告：使用过程函数和序列的 作为操作”选项，如果对象已在一个设置下导入（选中或未选中），然后您更改设置并导入更多对象，则在第一个设置下导入的对象将被删除。</p>
同步	<p>选择适当的单选按钮以指示是否要更新现有的类，或者将数据库对象作为新对象导入。</p> <p>如果您选择 同步现有类”选项，还请选择适当的复选框以确定模型注释、列默认值和/或库表约束是保留还是被导入对象的注释、值和约束覆盖。</p>

导入到	<p>选择适当的单选按钮以指示是更新包和当前打开的数据模型图，还是只更新包。</p> <p>如果没有打开图表，则“仅包”单选按钮默认为选中，并且选项被禁用；如果打开的图表在选定的包中，您可以选择任一选项。</p>
导入	<p>单击此按钮开始导入。</p> <p>将显示“选择要导入的数据库对象”对话框，列出找到的所有符合选择条件的数据库对象。</p> <p>选中每个架构（或object类型）的复选框以自动选择该组中的所有对象或单独导入每个object。</p> <p>单击全部按钮以选择所有类型和对象，或单击无按钮以清除所有选定的复选框。</p> <p>选择所有要导入的对象后，单击确定按钮继续导入。</p>

注记

- 在窗口中，可以为 32 位或 64 位应用程序定义 ODBC DSN，因此必须注意确保为Enterprise Architect使用的所有 ODBC DSN 定义共享相同的架构。从Enterprise Architect版本 16 开始，这一点尤其重要，因为它现在可用于 32 位和 64 位版本。另一种解决方案（也是Sparx Systems推荐的）是使用本地连接，因为它们适用于两种架构。
- ODBC 连接应使用 DBMS 供应商提供的 ODBC 驱动程序，例如MySQL的MySQL ODBC 驱动程序和 Oracle 的 Oracle ODBC 驱动程序；不支持第三方供应商提供的驱动程序，包括适用于 Oracle 的 Microsoft ODBC 驱动程序
- 您可以从<http://www.ch-werner.de/sqliteodbc/>为 SQLite 导入合适的 ODBC 驱动程序
- 由于 SQLite 的限制，不能实现 SQLite库表和列注释的round；要在从 ODBC 导入时保留在 SQLite 数据模型中输入的注释，请取消选中“从 ODBC源导入数据库架构”对话框的“同步”部分中的“覆盖物件注释”复选框
- 如果为逆向工程设置 ODBC 连接，默认设置就足够了
- 数据建模数据类型列表被定义为静态数据（在每个存储库中），因此根据存储库的年龄，“资源”的“数据建模数据类型”部分可能会提供其他数据类型Sparx Systems网站上的页面

生成数据库定义语言 (DDL)

一旦定义了物理模型并对对象建模，Enterprise Architect可以为各种对象生成数据库定义语言 (DDL)，包括数据库表、视图、函数、序列和过程。这是一种节省时间的机制，并减少了在其他工具中手动执行此操作可能引入的错误。正向工程由一组模板管理，这些模板定义如何将UML构造转换为目标 DBMS 中的对象。为所有受支持的 DBMS 提供了标准模板，并且可以编辑这些模板以自定义 DDL 的生成方式。在不支持开箱即用的 DBMS 的情况下，可以使用现有模板作为起点和参考来创建一组新模板。

在正向工程 DDL 时，可以将输出定向到一个文件（或一系列文件，每个object一个）或 DDL 执行引擎。执行引擎允许您立即执行 DDL，通过活动连接定位实时数据库。如果您将输出定向到文件，您可以稍后在方便时对实时数据库执行 DDL。生成的文件可以使用代码编辑器打开，通过选择 F12、Ctrl+E 或 Alt+7，您可以在 Enterprise Architect 中查看 DDL。

为对象生成DDL

在创建数据库模型时，您可以为单个object、一包对象或完成数据模型生成 DDL。唯一的区别是调用生成 DDL 过程的方式。

访问

打开数据库生成器窗口，然后使用上下文菜单并选择“生成DDL”。

功能区	开发 > 数据建模 > 数据库构建器 > 点击object、包或数据模型节点：生成DDL
-----	--

生成选项卡

字段/按钮	行动
包	单击  按钮并使用导航员窗口（查找包”对话框的一个版本）浏览要为其生成DDL的包。 (注记：此字段可能不会在所有情况下都显示。)
包括所有子包	选中此复选框以将对象包含在“选择要生成的对象”列表中的子包中。
删除目标文件	将对象生成为单个文件时，完整的文件名与object一起存储，并显示在“选择要生成的对象”列表的“目标文件”列中。 单击此按钮可删除所有现有文件名并提示新文件名。
选择要生成的对象	此字段按显示的顺序显示将为其生成 DDL 的对象列表。如果您需要更改此顺序以解决object依赖关系，请单击要移动的object并单击   按钮将该object在序列中向上或向下移动一个位置。 选择要为其生成 DDL 的每个object。点击： <ul style="list-style-type: none"> • 全部按钮选择每个项目 • 无按钮清除所有选择 • 按住 Ctrl 键的同时选择多个对象中的每个对象，以选择多个单独的对象 • 按住 Shift 键时块中的第一个和最后一个对象，以选择块中的每个object
保存生成的订单	如果您更改了列出对象的顺序，请在单击生成按钮时选中复选框以保存新序列。
刷新	重新加载对象列表，将每个object恢复到以前的位置（如果object位置已更改）。
单个文件	如果要将生成的 DDL 保存到单个文件，请选择此单选按钮。 单击  按钮以浏览文件路径和文件名。
每个表的单独文件	如果要将为每个object生成的 DDL 保存到单独的文件中，请选择此单选按钮。

	<p>钮。</p> <p>当您单击生成按钮时，系统会依次提示您输入每个object的目标文件名（如果尚未指定）。</p>
生成到 DDL 执行引擎	<p>如果要将在 DDL 保存到执行引擎（数据库构建器的“执行 DDL”选项卡），请选择此单选按钮。</p> <p>功能执行引擎提供了执行生成的脚本并立即响应执行中的错误，而无需创建外部文件并将其加载到另一个工具中。</p> <p>如果数据库构建器打开，“生成到 DDL 执行引擎”是默认选项。</p>
生成	<p>单击此按钮以使用您选择的选项运行生成运行过程。</p>
视图	<p>如果您已将 DDL 生成到单个外部文件，请单击此按钮以查看输出。</p> <p>默认情况下，Enterprise Architect使用默认代码编辑器。但是，您可以在“首选项”对话框中定义一个替代的默认 DDL 编辑器（开始>应用程序>首选项>首选项>源代码工程>代码编辑器>DDL”）。</p>
关	<p>单击此按钮可关闭对话框。如果您没有生成 DDL，此按钮也会放弃为 object生成 DDL。</p>

选项选项卡

如果您不想执行它们启动的操作，请将这些标志中的任何一个设置为False。

团体	选项
库表生成选项	<p>表- 表示应生成库表元素的 DDL (*)</p> <p>Primary Keys - 表示应该为 Primary Keys 生成 DDL (\$)</p> <p>Foreign Keys - 表示应生成外键的 DDL (\$)</p> <p>Indexes - 指示应生成索引的 DDL (\$)</p> <p>Unique约束- 表示应生成 Unique约束的 DDL (\$)</p> <p>选择约束- 表示应生成选择约束的 DDL (\$)</p> <p>库表触发器-表示应该生成DDL for库表触发器(\$)</p> <p>库表属性-表示应该生成扩展表的属性 (\$)</p> <p>Length Semantics - 表示应该为 Oracle Length Semantic 生成 DDL (\$)</p>
物件生成选项	<p>-视图- 表示应生成视图元素的 DDL (*)</p> <p>过程- 指示应生成过程元素的 DDL (*)</p> <p>函数- 表示应该生成函数元素的 DDL (*)</p> <p>序列- 指示应生成序列元素的 DDL (*)</p> <p>包- 表示应生成 DDL for Oracle包元素 (*)</p>
格式化	<p>包括前/后查询- 表示生成的 DDL 应该包括# '_PreStatements' 和 '_PostStatements' 中定义的语句# 查询</p> <p>Include Owners - 指示生成的 DDL 应包含所有元素的架构/所有者</p>

	<p>Include Comments - 表示生成的 DDL 应包含任何注释</p> <p>Include Header Comments - 指示生成的 DDL 应包含任何标题注释 (#)</p> <p>物件Comments - 表示生成的 DDL 应包含任何object (如库表或视图) 的注释 (#)</p> <p>Include Column Comments - 指示生成的 DDL 应包含任何列注释 (#)</p> <p>生成DROP statements - 表示生成的 DDL 应该包含对象的 DROP 语句</p> <p>使用数据库- 表示生成的 DDL 应该包含数据库使用语句</p> <p>使用别名- 指示生成的 DDL 使用任何object或列别名</p> <p>约束与库表分离- 表示生成的 DDL 应将约束的创建定义为与库表定义分开的语句</p> <p>在列定义中包含 NULL - 表示生成的 DDL 应将 NULL 关键字应用于定义为可为空的每个列定义；也就是说，未选中“NOT NULL”标志的列（此选项仅适用于支持“NULL”语法的 DBMS）</p>
--	---

注记

- (*) - 如果您已指定为该类型的单个元素生成 DDL，则带有此标记的选项将自动设置为True；也就是说，如果您选择了一个库表并且您的“生成库表”选项为False，Enterprise Architect会将选项更改为True
- (\$) - 如果“表”选项设置为False，带有此标记的选项将被禁用
- (#) - 如果“包含评论”选项设置为False，则带有此标记的选项将被禁用
- 在Enterprise Architect的企业版、统一版和终极版中，如果启用了安全性，则必须具有“生成源代码和DDL”权限才能生成DDL
- 对于 PostgreSQL 数据库，您必须将 'Sequences' 选项设置为True以后用要创建的自动增量列
- 如果生成 Oracle 序列，则必须始终将“库表触发器”和“Sequences”选项设置为True，以便生成预插入触发器以选择下一个序列值来填充列；此外，在列属性中，将“属性”属性设置为True
- 您可以编辑系统用来生成 DDL 的 DDL 模板；这些存储在存储库级别，以便同一存储库的所有其他用户将自动使用更新的模板

编辑 DDL 模板

DDL 模板编辑器提供了在从数据模型生成 DDL 时更改系统使用的模板的能力。它应用了公共代码编辑器的功能，包括各种宏的智能感知代码。有关智能感知信息和公共代码编辑器，请参阅编辑源代码帮助。

访问

功能区	开发 > 数据建模 > 模板
-----	----------------

选择和编辑模板

选项	行动
语	单击下拉箭头并选择数据库类型（数据库管理系统）。
新数据库	单击此按钮可为非标准 DBMS 创建一组新模板。 将显示 输入”对话框，您可以在其中键入要为其创建模板的新 DBMS 的名称。 这将更新 语言”字段。
模板	显示所选模板的内容，并提供修改这些内容的编辑器。
模板	列出基本 DDL 模板，单击模板名称可显示和编辑模板内容；所选模板的名称突出显示。 “修改”字段表示您是否修改了系统最初提供的默认模板。
构造型覆盖	列出当前选定的基本模板。 “Modified”字段指示您是否修改了默认的原型模板。
加新自定义模板	单击此按钮以显示 创建新的自定义模板”对话框，您可以在该对话框中从下拉列表中选择模板类型，然后输入模板的名称。 模板类型成为名称的前缀；例如： 命名空间_MyDDLTemplate
加新Stereotyped Override	选择一个基本模板并单击此按钮以显示 新模板覆盖”对话框，以便为所选模板模板从下拉列表中，选择要应用覆盖模板的类和/或特征原型。
获取默认模板	单击此按钮以使用所选模板的默认版本刷新编辑器显示。（这不会删除模板的更改版本。）
节省	单击此按钮以使用模板面板的更新内容覆盖选定的模板。
删除	如果你已经覆盖了选中的模板，点击这个按钮可以删除被覆盖的模板，并用相应的默认 DDL 模板替换它。

注记

- 用户修改和用户定义的参考可以作为模板数据导入和导出 (参见共享参考主题)
- 数据库类型的任何用户定义模板都列在 “参考代码、DDL、变换和 CSV 模板” 表中的 “导出数据” 对话框中，由带有后缀导出的 DBMS 名称标识 - 如果不存在用户定义模板对于 DBMS，对话框中没有 DBMS 条目
- 您还必须为 DBMS 定义任何适当的数据类型，如果将模板导出为参考，您还必须导出 “模型数据类型-代码和 DDL” 表

DDL 模板语法

DDL模板是使用Enterprise Architect的代码模板框架编写的，但它们已被扩展以支持 DDL 生成。

DDL 模板开发

本节将讨论 DDL 模板开发的这些方面。

方面	也见
DDL模板	DDL模板
DDL 宏	DDL 宏
DDL函数宏	DDL函数宏
DDL属性宏	DDL属性宏
模板中的 DDL 选项	模板中的 DDL 选项

DDL模板

DDL 模板编辑器的操作方式与代码模板编辑器相同，只是 DDL 模板编辑器同时显示用于 DDL 生成的模板和用于更改 DDL 生成的模板。Alter DDL Generation 模板显示在列表底部，前缀为“DDL Diff”。

用于 DDL 生成的基本模板

DDL 模板框架由许多用于 DDL 生成的基本模板组成。每个基本模板为UML数据模型的特定方面生成 DDL 语句（或部分语句）。

模板

此表列出并简要描述了用于 DDL 生成的基本模板。

模板	描述
DDL选择约束	由 DDL库表约束模板调用以生成库表object的选择约束声明。
DDL专栏注解	通常由 DDL 创建库表Extras模板为每个库表列生成COMMENT ON 语句（或等效语句）。
DDL 列定义	由众多模板调用以构建语句以创建单个库表列，就像它出现在 CREATE TABLE 语句中一样。
DDL 列附加功能	通常由 DDL 创建库表Extras模板调用以生成每个库表列的任何扩展列属性。
DDL约束栏名称	由每个约束模板调用以检索当前约束中涉及的正确格式的列名。
DDL约束注解	通常由 DDL 创建库表Extras模板调用以为每个库表约束生成 COMMENT ON 语句（或等效语句）。
DDL 创建外键	由 DDL 创建库表约束 为库表object生成外键约束的模板。
DDL 创建函数	由 DDL脚本文件调用 为函数object生成 CREATE FUNCTION 语句的模板。
DDL创建包	由 DDL脚本文件调用 用于为包object生成 CREATE PACKAGE 语句的模板（仅限 Oracle）。
DDL 创建过程	由 DDL脚本文件调用 用于为过程生成 CREATE object语句的模板。
DDL 创建架构	目前未使用。
DDL 创建序列	由 DDL脚本文件调用 用于为序列object生成 CREATE SEQUENCE 语句的模板。
DDL 创建库表	由 DDL脚本文件调用 为库表object生成 CREATE TABLE 语句的模板。
DDL 创建库表约束	由 DDL脚本文件调用 为库表object生成库表约束和索引的模板。
DDL 创建库表Extras	由 DDL脚本文件调用 为库表object生成扩展库表属性的模板。
DDL 创建视图	由 DDL脚本文件调用 为视图object生成 CREATE VIEW 语句的模板。
DDL 数据类型	由 DDL 列定义模板调用，为库表列生成格式正确的数据类型语句。
DDL 落柱附加功能	由 DDL Drop库表Extras模板调用，为列扩展属性生成任何专门的删除语句。

DDL 丢弃外键	由 DDL脚本文件调用 模板生成语句以删除库表object的所有外键。
DDL 丢弃函数	由 DDL脚本文件调用 为函数object生成 DROP FUNCTION 语句的模板。
DDL 下降过程	由 DDL脚本文件调用 用于为过程object生成 DROP PROCEDURE 语句的模板。
DDL 丢弃序列	由 DDL脚本文件调用 用于为序列object生成 DROP SEQUENCE 语句的模板。
DDL Drop库表	由 DDL脚本文件调用 为库表object生成 DROP TABLE 语句的模板。
DDL Drop库表Extras	由 DDL脚本文件模板调用以生成用于 DROP库表object的所有扩展属性的语句。
DDL 拖放视图	由 DDL脚本文件模板调用以生成视图object的视图VIEW 语句。
DDL 外国约束	由 DDL库表约束模板调用以生成库表object的 ADD FOREIGN KEY CONSTRAINT 语句。
DDL 补助金	由 DDL 创建库表Extras模板调用以生成当前object的 GRANT 语句 (仅限 Oracle)。
DDL 索引	由 DDL库表约束模板调用以生成库表object的 CREATE INDEX 语句。
DDL 左环绕	用于定义用作object (或object组件) 名称左侧分隔符的字符 (或多个字符)。
DDL名称	<p>大多数模板使用它来提供格式化object (或object特征) 名称的通用方式。这个模板接受四个参数：</p> <ul style="list-style-type: none"> 物件位置 (值：EA 或 LIVE) 物件类型 (取值：OWNER、TABLE、VIEW、PROCEDURE、FUNCTION、SEQUENCE、PACKAGE、COLUMN、CONSTRAINT、CONSTRAINT_COLUMN、REFERENCE_TABLE、REFERENCE_COLUMN) 包括拥有着标志；控制名称是否应以拥有名称为前缀 (值：拥有着或 {blank}) 包括环绕标志；控制名称是否应由左右环绕字符分隔 (值：INCLUDE_SURROUND 或 {blank})
DDL Primary约束	由 DDL库表约束模板调用以生成库表object的 ADD PRIMARY KEY CONSTRAINT 语句。
DDL参考列名称	通常由 DDL名称模板调用以检索外键中涉及的格式正确的引用列名称。
DDL参考定义	由调用 DDL 外国约束模板为外键约束生成 ON DELETE/ON UPDATE 语句。
DDL 右环绕	用于定义用作object (或object组件) 名称右侧分隔符的字符 (或多个字符)。
DDL脚本文件	用于生成 DDL A顶级模板；所有其他模板都是从这个模板调用的。
DDL脚本头	由 DDL脚本文件调用 模板在每个 DDL 文件的开头添加标题注释。

DDL脚本Separator	由必须在生成的 DDL 中包含语句分隔符的所有模板使用。
DDL 语句团队	用于定义用作语句终止符的字符 (或多个字符)。例如,对于大多数 DBMS,分号(';')。
DDL 语句团队Alt	用于定义用作替代语句终止符的字符 (或多个字符)。例如,某些 DBMS 必须更改语句终止符,以免导致为基于 SQL 的对象 (如视图和过程) 生成的 DDL 语句出现问题。
DDL 同义词	由 DDL 创建库表Extras模板调用以生成 CREATE SYNONYMS 语句 (仅限 Oracle)。
DDL库表约束表	由 DDL 创建库表约束模板调用,为每个库表object生成库表约束和索引,同时考虑生成选项。
DDL库表级别注解	由 DDL 创建库表Extras模板调用以生成object的 COMMENT ON 语句 (或等效语句)。
DDL触发器	由 DDL库表约束 为库表object生成 CREATE TRIGGER 语句的模板。
DDL Unique约束	由 DDL库表约束模板调用以生成库表object的 ADD UNIQUE CONSTRAINT 语句。
DDL使用数据库	由 DDL脚本文件模板调用以在每个 DDL 文件的开头包含一个 USE DATABASE 语句。

用于更改 DDL 生成的基本模板

DDL 模板框架由许多用于 Alter DDL 生成的基本模板组成。每个基本模板根据检测到的动作生成行动语句，这些动作必须用于同步数据模型和实时数据库。

模板

此表列出并简要描述了用于 Alter DDL 生成的基本模板。

模板	描述
DDL 差异列	Enterprise Architect为检测到的每个库表列差异直接调用。
DDL Diff约束	由Enterprise Architect为检测到的每个库表约束差异直接调用。
DDL Diff库表	Enterprise Architect为检测到的每个库表差异直接调用。
DDL视图	Enterprise Architect针对检测到的每个视图差异直接调用。
DDL 差异过程	由Enterprise Architect为检测到的每个存储过程差异直接调用。
DDL 差异函数	由Enterprise Architect为检测到的每个函数差异直接调用。
DDL 差异序列	由Enterprise Architect为检测到的每个序列差异直接调用。

DDL 宏

字段替换宏提供对模型数据的访问。特别是，它们用于从以下位置访问数据字段：

- 数据库对象（如表和视图）
- 列
- 约束
- 约束专栏

字段替换宏根据 Camel 大小写命名。按照惯例，所有 DDL 宏都以“ddl”为前缀。

如果复选框/布尔值为真，则表示复选框或布尔值的宏返回string值 “T”。否则返回一个空string。

内部字段宏 - 字段

ddlAction 宏是“Alter DDL”模板中可用的内部 ，提供对Enterprise Architect内部字段的直接访问；它没有直接映射到任何存储的数据。

ddlAction 表示将实时数据库与当前存储库同步所必须执行的操作。例如， “创建库表”、 “删除库表”或 “更改拥有着”。

元素字段宏

此列表标识 DDL 模板中可用于访问元素级字段的宏，其中（在Enterprise Architect中）字段是可编辑的，例如“库表名称”和“库表别名”。

ddlFunctionAlias

函数 属性”对话框：主”选项卡：别名”文本域。

ddl函数名

函数 属性”对话框：名称”文本字段。

ddl所有者

{库表元素} 属性”对话框：{元素} 库表详细信息”选项卡：拥有着”文本字段。

ddlPackageAlias

'包'属性'对话框：'主'选项卡：'别名' 文本域。

ddlPackageName

'包'属性'对话框：'名称'文本字段。

ddlProcedureAlias

过程”属性对话框：主”选项卡：别名” 文本域。

ddlProcedureName

过程 属性”对话框：名称”文本字段。

ddlSchemaFunctionName

从实时数据库中读取的函数元素定义的名称。

ddlSchemaOwner

从实时数据库中读入的元素定义的 拥有着"属性。

ddlSchemaProcedureName

从实时数据库中读取的过程元素定义的名称。

ddlSchemaSequenceName

从实时数据库中读取的序列元素定义的名称。

ddlSchemaTableName

从实时数据库中读入的 库表名称"属性。

ddlSchemaViewName

从实时数据库中读取的元素视图定义的名称。

ddlSequenceAlias

“ 列”属性对话框： 主”选项卡： 别名”文本字段。

ddlSequenceName

序列 属性”对话框： 名称”文本字段。

ddlTableAlias

库表 属性”对话框： 主”选项卡： 别名” 文本域。

ddlTableDBMS

库表 属性”对话框： 主”选项卡： 数据库”下拉列表字段。

ddlTableLevelComment

库表 属性”对话框： 注释”文本字段。

ddlTableName

库表 属性”对话框： 名称”文本字段。

ddlViewAlias

视图 属性”对话框： 主”选项卡： 别名” 文本域。

ddlViewName

视图 属性”对话框： 名称”文本字段。

列字段宏

此列表标识 DDL 模板中可用于访问列相关字段的宏，其中（在Enterprise Architect中）字段是可编辑的，例如“列名称”和“列别名”。

ddl列名

列和约束”对话框：列”选项卡：名称”单元。

ddlColumnAlias

列和约束”对话框：列”选项卡：别名”单元。

ddlColumnComment

列和约束”对话框：列”选项卡：注释”文本字段。

ddlSchemaColumnName

从实时数据库中读入的列名称属性。

注记：此字段在Enterprise Architect中不可直接编辑。

约束字段Macros

此表列出了 DDL 模板中可用于访问约束相关字段的宏，其中（在Enterprise Architect中）这些字段是可编辑的，例如“约束类型名称”和“约束”。

ddlConstraintAlias

“Columns约束约束标签：别名”单元。

ddlConstraintColumnAlias

列和约束”对话框：约束”选项卡：“涉及的列：已分配”列表。

ddlConstraintColumnName

列和约束”对话框：约束”选项卡：“涉及的列：已分配”列表。

ddlConstraintComment

列和约束”对话框：约束”标签：注释”文本字段。

ddlConstraintName

列和约束”对话框：约束”标签：名称”单元。

ddlPKColumnCount

仅当当前约束具有主键类型时才相关，此宏将向主键返回分配列的计数。

列和约束”对话框：约束”选项卡：“涉及的列：已分配”列表。

ddlReferenceColumnAlias

仅当当前约束具有外键类型时才相关，此宏将从引用表返回列别名。

专 和约束”对话框：约束”标签：别名”单元。

ddlReferenceColumnName

仅当当前约束具有外键类型时才相关，此宏将返回引用表中的列名。

外键 约束”对话框：“涉及的列”列表：“父”列。

ddlReferenceTableAlias

仅当当前约束具有外键类型时才相关，此宏将返回引用表的别名。

库表 属性”对话框：“主”选项卡：“别名”文本域。

ddlReferenceTableName

仅当当前约束具有外键类型时才相关，此宏将返回引用表的名称。

外键 约束”对话框：“涉及的列”列表：“父”列标题。

ddlReferenceTableOwner

仅当当前约束具有外键类型时才相关，此宏将返回引用表的所有者。

外键 约束”对话框：“涉及的列”列表：“父”列标题。

ddlSchemaConstraintColumnName

从实时数据库中读取的当前约束中涉及的列名。

注记：此字段在Enterprise Architect中不可直接编辑。

ddlSchemaConstraintName

从实时数据库中读取的约束名称属性。

注记：此字段在Enterprise Architect中不可直接编辑。

ddlSchemaConstraintType

约束类型属性从实时数据库中读取。

注记：此字段在Enterprise Architect中不可直接编辑。

DDL函数宏

DDL函数宏提供了一种方便的方式来操作、检索或格式化与 DDL 生成相关的元素数据。这些宏以及代码函数宏可用于 DDL 模板。每个函数宏返回一个结果string，并以与代码模板函数宏相同的方式使用。

此处描述了可用的函数宏。所有参数都有一个字符串类型，并用方括号表示；即：
FUNCTION_NAME([param])。

DDL_DATATYPE_SIZE ([产品名称] , [数据类型])

以 DDL 语法返回当前列的完全格式化的数据类型。

参数

- productName - 当前库表分配的 DBMS，如#
服务器2012，Oracle 或 PostgreSQL
- datatype - 当前列的数据类型名称，例如 VARCHAR 或 INT

评论

在Enterprise Architect库表列中，数据类型使用影响 DDL 语法的类型（属性1或2）定义；这个函数宏在构建返回值时会考虑长度类型（和其他因素）。

DDL_GET_DEFINITION_PARAS ([定义])

从提供的函数/过程定义中返回参数的string表示形式。

参数

- 定义 -完成#
过程/函数的定义

评论

一些 DBMS（如 PostgreSQL）支持同一过程/函数名的多个定义。这些定义的不同之处仅在于它们的参数列表，因此要操作此类对象，DDL 必须指定名称和参数。这个函数宏使 DDL 模板能够提取参数，以便随后可以使用它们来识别单个对象。

DDL_INCLUDE_SQLQUERY([objectName])

返回#

SQLQuery object中定义的语句。

参数

- objectName - #
的名称#
当前数据模型查询中定义的object

评论

没有任何。

DDL_INDEX_SORT ([产品] , [列])

返回给定索引的排序顺序。

参数

- 产品 - DBMS (目前, 火鸟)
- columns - 索引中涉及的列名的 CSV

评论

该宏目前仅适用于火鸟索引。

DDL_RESOLVE_NAME ([productName], [name], [leftSurround], [rightSurround])

如果名称是当前 DBMS 的保留字, 则返回提供的分隔名称 (使用提供的左右字符)。

参数

- productName - 当前库表分配的 DBMS, 如# 服务器2012, Oracle 或 PostgreSQL
- name - object /列名
- leftSurround - 用于围绕名称的左侧字符; 例如, 单引号 {}
- rightSurround - 用于围绕名称的右侧字符; 例如, 单引号 {}

评论

某些 DBMS 的 DDL 语法要求保留字的名称以不同的方式分隔; 这个函数宏可用于安全地格式化 DB2 和火鸟的所有名称。

DDL_TABLE_TAGVALUE ([tagName])

返回当前库表的存储库版本中提供的标记名称的值。

参数

- tagName - 要检索的标记项的名称

评论

没有任何。

EXECUTE_CURRENT ([objectName], [actionName], [priority])

将当前模板的返回string添加到执行引擎的执行队列中。

参数

- 物件- 将在执行队列的“物件”列中显示的值, 表示正在更新的object的名称
- 行动- 将在执行队列的“行动”列中显示的值, 指示导致生成此语句的操作
- priority - 表示语句优先级的数值; 数字越大, 语句在队列中的位置越低

评论

这个函数宏可以在整个模板中的任何时候调用, 但直到结束才会执行。一旦模板完成, 它生成的 DDL 就会被发送到执行队列。

如果用户选择为文件生成 DDL, 则此函数宏无效。

EXECUTE_STRING ([objectName], [actionName], [priority], [ddlStatement])

将提供的 DDL 语句添加到执行引擎的执行队列中。

参数

- 物件- 将在执行队列的“物件”列中显示的值，表示正在更新的object的名称
- 行动- 将在执行队列的“行动”列中显示的值，指示导致生成此语句的操作
- priority - 表示语句优先级的数值；数字越大，语句在队列中的位置越低
- ddlStatement - 执行所需操作的单个 DDL 语句

评论

如果用户选择为文件生成 DDL，则此函数宏无效。

EXIST_STRING ([ddlStatement])

在执行引擎的执行队列中搜索提供的 DDL 语句，如果找到该语句，则返回“T”。

参数

- ddlStatement - 单个 DDL 语句

评论

没有任何。

GET_FIRST_SQL_KEYWORD([语句])

返回提供的第一个关键字#

陈述。

参数

- 声明 - #
陈述

评论

没有任何。

ODBC_TABLE_TAGVALUE ([tagName])

返回当前表的实时数据库版本中提供的标记名称的值。

参数

- tagName - 要检索的标记项的名称

评论

没有任何。

PROCESS_DDL_SCRIPT ([type], [parameter2], [parameter3], [parameter4])

为特定目的返回格式化string A通用函数宏。

参数

- type - 指定要采取的特殊行动
- parameter2 - 通用参数 2，每种类型都有不同的用途
- parameter3 - 通用参数 3，每种类型都有不同的用途
- parameter4 - 通用参数 4，每种类型都有不同的用途

评论

对于 Oracle 同义词，请使用以下参数：

- 类型= 同义词“
- parameter2 =表名；例如，TBL_EMPLOYEES
- parameter3 = 一个分隔的string值，用分号分隔，指定同义词所有者和名称之间用全冒号；例如，OE:EMPLOYEES;PUBLIC:PUB_EMPLOYEES;
- parameter4 = 语句终止符

返回结果

格式：

为 TBL_EMPLOYEES 创建同义词 OE.EMPLOYEES；

为 TBL_EMPLOYEES 创建公共同义词 PUB_EMPLOYEES；

REMOVE_LAST_SEPARATOR ([ddlStatement], [分隔符])

返回提供的 DDL 语句，删除了最后一个分隔符（如果存在）。

参数

- ddlStatement - 部分 DDL 语句
- separator - 应删除的分隔符

评论

在构建表示 DDL 语句的string时，通常的做法是在每个项目之后附加分隔符；但是，最后一项之后不需要分隔符，因此提供了此函数宏来删除尾随分隔符。

REMOVE_STRING ([ddlStatement])

从执行引擎的执行队列中删除提供的 DDL 语句。

参数

- ddlStatement - 单个 DDL 语句

评论

没有任何。

SUPPRESS_EXECUTE_CURRENT ([boolean])

启用/禁用对 EXECUTE_CURRENT 的后续调用A函数宏。

参数

- 布尔值 - True或False

评论

此标志的默认状态为False ; 也就是说 , 对 EXECUTE_CURRENT 的调用不会被忽略。

DDL属性宏

DDL属性宏提供了一种检索元素属性值（即标记值）的便捷方式。在数据建模的范围内，有两组属性：

- 内部属性（Enterprise Architect在其比较中识别和使用的那些）和
- 自定义属性

这些属性宏提供对针对各种属性定义的属性的访问。所有属性宏具有相同的语法，返回一个string并要求指定属性的名称。

语法：`propertyMacroName:"propertyName"`

内部性质

tableBoolProperty : 属性名称"

返回当前库表的存储库版本中的内部属性值的布尔表示（“T”或“”）。

参数

- `propertyName` - 要检索的属性名称

评论

没有任何。

表属性 : 属性名称"

返回当前库表的存储库版本中的内部属性值。

参数

- `propertyName` - 要检索的属性名称

评论

没有任何。

列属性 : 属性名称"

返回当前列的存储库版本中的内部属性的值。

参数

- `propertyName` - 要检索的属性名称

评论

没有任何。

columnBoolProperty : "属性名称"

返回当前列的存储库版本中的内部属性值的布尔表示（“T”或“”）。

参数

- propertyName - 要检索的属性名称

评论

没有任何。

约束属性：属性名称”

返回当前约束的存储库版本中的内部属性的值。

参数

- propertyName - 要检索的属性名称

评论

没有任何。

约束布尔属性：属性名称”

返回当前约束的存储库版本中的内部属性值的布尔表示（“T”或””）。

参数

- propertyName - 要检索的属性名称

评论

没有任何。

约束列属性：属性名称”

返回当前约束列的存储库版本中的内部属性的值。

参数

- propertyName - 要检索的属性名称

评论

没有任何。

constraintColumnBoolProperty：属性名称”

返回当前约束列的存储库版本中的内部属性值的布尔表示（“T”或””）。

参数

- propertyName - 要检索的属性名称

评论

没有任何。

视图属性：属性名称”

返回当前视图的存储库版本中的内部属性的值。

参数

- propertyName - 要检索的属性名称

评论

没有任何。

程序属性：属性名称"

返回当前过程的存储库版本中的内部属性的值。

参数

- propertyName - 要检索的属性名称

评论

没有任何。

功能属性：属性名称"

返回当前函数的存储库版本中的内部属性值。

参数

- propertyName - 要检索的属性名称

评论

没有任何。

序列属性：属性名称"

返回当前序列的存储库版本中的内部属性的值。

参数

- propertyName - 要检索的属性名称

评论

没有任何。

包属性：属性名称"

返回当前数据库包的存储库版本中的内部属性值。

参数

- propertyName - 要检索的属性名称

评论

没有任何。

odbcTableProperty:"propertyName"

返回当前库表的 ODBC 版本中的内部属性值。

参数

- propertyName - 要检索的属性名称

评论

没有任何。

odbcConstraintProperty:"propertyName"

返回当前约束的 ODBC 版本中的内部属性值。

参数

- propertyName - 要检索的属性名称

评论

没有任何。

用户定义的属性

表用户属性：属性名称"

返回当前库表的存储库版本中用户定义属性的值。

参数

- propertyName - 要检索的属性名称

评论

没有任何。

列用户属性：属性名称"

返回当前列的存储库版本中用户定义属性的值。

参数

- propertyName - 要检索的属性名称

评论

没有任何。

约束用户属性：属性名称"

返回当前约束的存储库版本中用户定义属性的值。

参数

- propertyName - 要检索的属性名称

评论

没有任何。

约束列用户属性：属性名称”

返回当前约束列的存储库版本中用户定义属性的值。

参数

- propertyName - 要检索的属性名称

评论

没有任何。

viewUserProperty:"propertyName"

返回当前视图的存储库版本中用户定义属性的值。

参数

- propertyName - 要检索的属性名称

评论

没有任何。

procedureUserProperty:"propertyName"

返回当前过程的存储库版本中用户定义属性的值。

参数

- propertyName - 要检索的属性名称

评论

没有任何。

函数用户属性：属性名称”

返回当前函数的存储库版本中用户定义属性的值。

参数

- propertyName - 要检索的属性名称

评论

没有任何。

序列用户属性：属性名称”

返回当前序列的存储库版本中用户定义属性的值。

参数

- propertyName - 要检索的属性名称

评论

没有任何。

模板中的 DDL 选项

DDL 生成选项宏为 DDL 模板访问生成选项提供了一种方便的方法。

此列表标识并简要描述了每个可用的选项宏。每个选项都有一个值 'T' 代表 true 或一个空string代表false。

ddlGenerateToExecuteEngine

将生成的 DDL 定向到执行引擎。

ddlOptionColumn评论

在生成的 DDL 中包含列注释。

ddlOptionGenerateCheck

在生成的 DDL 中包含选择约束。

ddlOptionGenerateDrop

在生成的 DDL 中包含 DROP 语句。

ddlOptionGenerateForeign

在生成的 DDL 中包含外键。

ddlOptionGenerateFunction

在生成的 DDL 中包含函数。

ddlOptionGenerateIndex

在生成的 DDL 中包含索引。

ddlOptionGenerateLength语义

(仅限 Oracle) 在生成的 DDL 中的文本列上包含长度语义语法。

ddlOptionGenerateNullable

如果在生成的 DDL 中没有将其标记为 NOT NULL 列，则针对每列包含关键字 NULL。

ddlOptionGeneratePackage

(仅限 Oracle) 在生成的 DDL 中包含包。

ddlOptionGeneratePrimary

在生成的 DDL 中包含主键约束。

ddlOptionGenerateProcedure

将过程包含在生成的 DDL 中。

ddlOptionGenerateSeparateConstraint

生成库表约束分别对 CREATE TABLE 语句；也就是说，使用 ALTER TABLE 语句。
注记：一些 DBMS 不支持所有条件下的单独约束。

ddlOptionGenerateSequence

在生成的 DDL 中包含序列。

ddlOptionGenerateTable

在生成的 DDL 中包含表。

ddlOptionGenerateTableProperty

在生成的 DDL 中包含扩展属性表。

ddlOptionGenerateTrigger

在生成的触发器中包含简单的库表。

ddlOptionGenerateUnique

在生成的 DDL 中包含唯一约束。

ddlOptionGenerateView

在生成的 DDL 中包含视图。

ddlOptionHeaderComments

在生成的 DDL 中包含标题注释。

ddlOptionTable评论

在生成的 DDL 中包含库表注释。

ddlOptionUseAlias

使用别名而不是生成DDL 屏幕上指定的所有对象（object组件）的名称。

ddlOptionUseDatabaseName

在每个生成的文件的开头包含 USE DATABASE 语句。

ddlUseAlias

使用别名而不是数据库生成器“数据库比较”选项卡上指定的所有对象（object组件）的名称。

DDL 限制

数据库管理系统 (DBMS) 的基本特征是允许通过结构化语言定义数据库对象；这种语言称为 DDL (数据定义语言, 或数据描述语言)。每个 DBMS 的 DDL 语法都是唯一的。虽然在所有 DBMS 中都有通用的 DDL 语句和关键字, 但存在一些差异, 要求每个 DBMS 在 Enterprise Architect 中有自己的一组 DDL 模板。

本页总结了每个受支持的数据库管理系统的主要限制。

MS Access

- 评论不能应用于 (或更改) 表、库表列、库表约束或视图, 因此 Enterprise Architect 忽略这些差异
- CREATE TABLE 语句不支持列默认值的定义, 因此 Enterprise Architect 将默认定义从所有生成的 DDL 中排除; 但是, 它确实突出了比较逻辑中的默认差异
- 通常 DDL 中的 object 名称可以用方括号 ([]) 括起来, 以便它们可以包含空格和其他非标准字符, 但是 CREATE VIEW DDL 语句不支持方括号表示法; “创建视图” DDL 模板用下划线 (_) 字符替换所有空格

MySQL

- 只有 MySQL 版本大于 5.5.3 时, 才能对 Indexes 和 Unique 约束应用注释
- 评论只能在创建时应用于索引和唯一约束, 因此更改索引或唯一约束的评论会导致约束被删除并重新创建
- 不支持选择约束; 虽然 MySQL DDL 引擎可以解析此类语句, 但它只是忽略它们
- 评论不能应用于 (或更改) 视图、过程或函数, 因此 Enterprise Architect 忽略这些差异

甲骨文

- 注释不能应用于 (或更改) 过程、序列或函数, 因此 Enterprise Architect 忽略这些差异

PostgreSQL

- 目前 Enterprise Architect 不支持函数参数, 因此任何通过名称引用函数的语句 (COMMENT ON 或 DROP) 都会失败, 因为它们必须使用函数名称和参数的组合

#

精简版

- 约束不能添加到现有的库表中; 必须删除和创建库表 (包括创建语句中的新约束)
- 任何 object 类型都不支持注释, 因此 Enterprise Architect 忽略所有注释差异

导入DDL脚本

此特征允许您从文件系统中的指定目录导入 DDL 脚本，以在Enterprise Architect模型中创建数据库模型对象。目录中所有文件扩展名与指定文件扩展名匹配的脚本都将被导入。

脚本文件将导入到当前选定的Enterprise Architect包中，按照 DDL 脚本的定义创建表、视图、列、约束、过程、函数、序列等。

访问

功能区	开发>数据建模>导入DDL
-----	---------------

导入DDL脚本对话框

目录	类型或浏览要导入的目录的名称。
进程子目录	选中此复选框可以同时包含子目录的内容。
数据库管理系统	从下拉列表中选择适用 DDL 脚本的 DBMS 类型。
文件扩展	类型或从下拉列表中选择要包含在导入中的文件扩展名。使用 ";" 分隔值。
导入	点击 导入 按钮开始导入。

支持数据库管理系统

Enterprise Architect内置了对各种数据库管理系统的支持，但它也提供了扩展产品以支持其他 DBMS 的灵活性。DDL 模板编辑器可用于定义如何为不受支持的 DBMS 生成 DDL，转换模板可用于为不受支持的 DBMS 定义到物理模型的新转换，并且可以为现有或新 DBMS 定义新数据类型。

Enterprise Architect为这些数据库管理系统提供建模结构和正向和反向工程数据库模式的能力：

- DB2 (*)
- 火鸟
- MS Access 97、2000、2003、2007、2013
- 女士#
服务器从 2005 年开始，所有版本，包括 Express 和 Azure #
数据库
- MariaDB
- MySQL v4、v5
- 从 9i 开始的 Oracle (所有版本)
- PostgreSQL (包括版本 12)
- SQLite
- Informix (#)
- 安格尔 (#)
- 基间 (#)
- Sybase Adaptive Server Anywhere (Sybase ASA) (#)
- Sybase Adaptive服务器企业(Sybase ASE) (#)

(*) - 仅在窗口和 Linux 环境中托管时与 DB2 兼容。

(#) - 不会对这些 DBMS 进行进一步开发，因为Enterprise Architect用户群不常用这些产品。这将使Sparx Systems能够将精力集中在广泛使用的其他数据库建模领域。

注记

- 要为特定 DBMS 执行数据建模，您的存储库中必须具有该 DBMS 的适当数据类型；您可以从Sparx Systems网站的“资源”页面下载最新的数据定义

更多信息

Sparx Systems Enterprise Architect为信息建模者、数据建模者和架构师提供了实用工具，用于创建跨组织内抽象级别的模型：概念、逻辑和物理。

概念模型：这些模型与技术无关，有助于与业务和领域专家进行讨论，以表达和商定基本领域概念。

逻辑模型：这些模型在保持技术中立的同时为概念模型增加了细节和精度，促进了信息分析师对逻辑结构的讨论。

物理模型：**T模型**将特定于技术的数据应用于模型，帮助工程师做出在目标环境（如数据库管理系统）中部署的技术决策。

版信息

Enterprise Architect的企业统一版和终极版中提供了数据库生成器。

