05 - 模块 5：存储数据

模块 5：存储数据

模块简介（模块图块）：在此模块中，您将了解 Mendix 如何存储数据。

讲座 5.1：简介

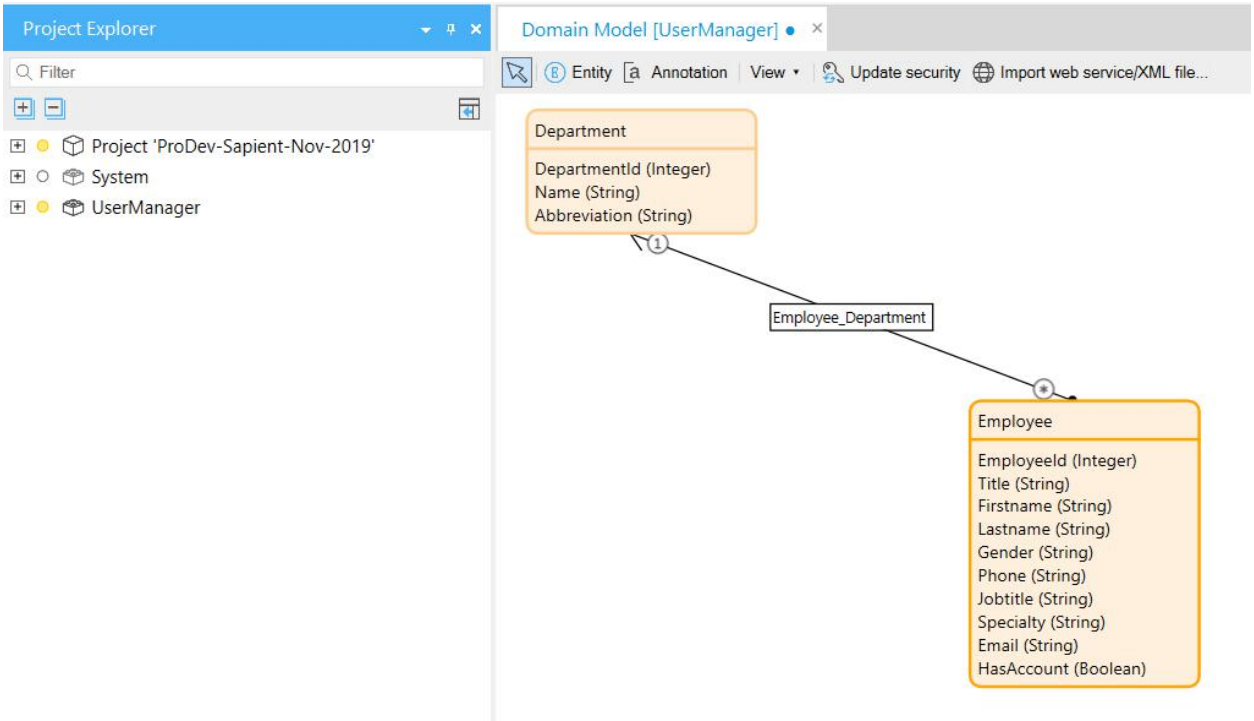
将 HR 数据配置导入 Mendix 后，现在需要将注意力集中在存储数据上，如此一来，即使 HR 服务关闭，也可确保 Summerhill 医院的用户能够正常访问数据。Mendix 将大多数数据存储在关系数据库中。对于 Summerhill 医院，我们将使用 Mendix Runtime 附带的内部数据库 *HSQL*。数字化创新团队计划在下一步提供更稳健的存储解决方案。幸运的是，Mendix 可使用最热门的 *RDBMS* 数据库作为后端，因此待数字创新团队选定数据库后，只需配置您的 Mendix 应用程序并切换到数据库即可。

在本模块中，您将了解更多以下内容：

- 实体
- 关联
- 永久
- 对象
- 在运行时实施 *ORM* 的详细信息

讲座 5.2：域模型概述

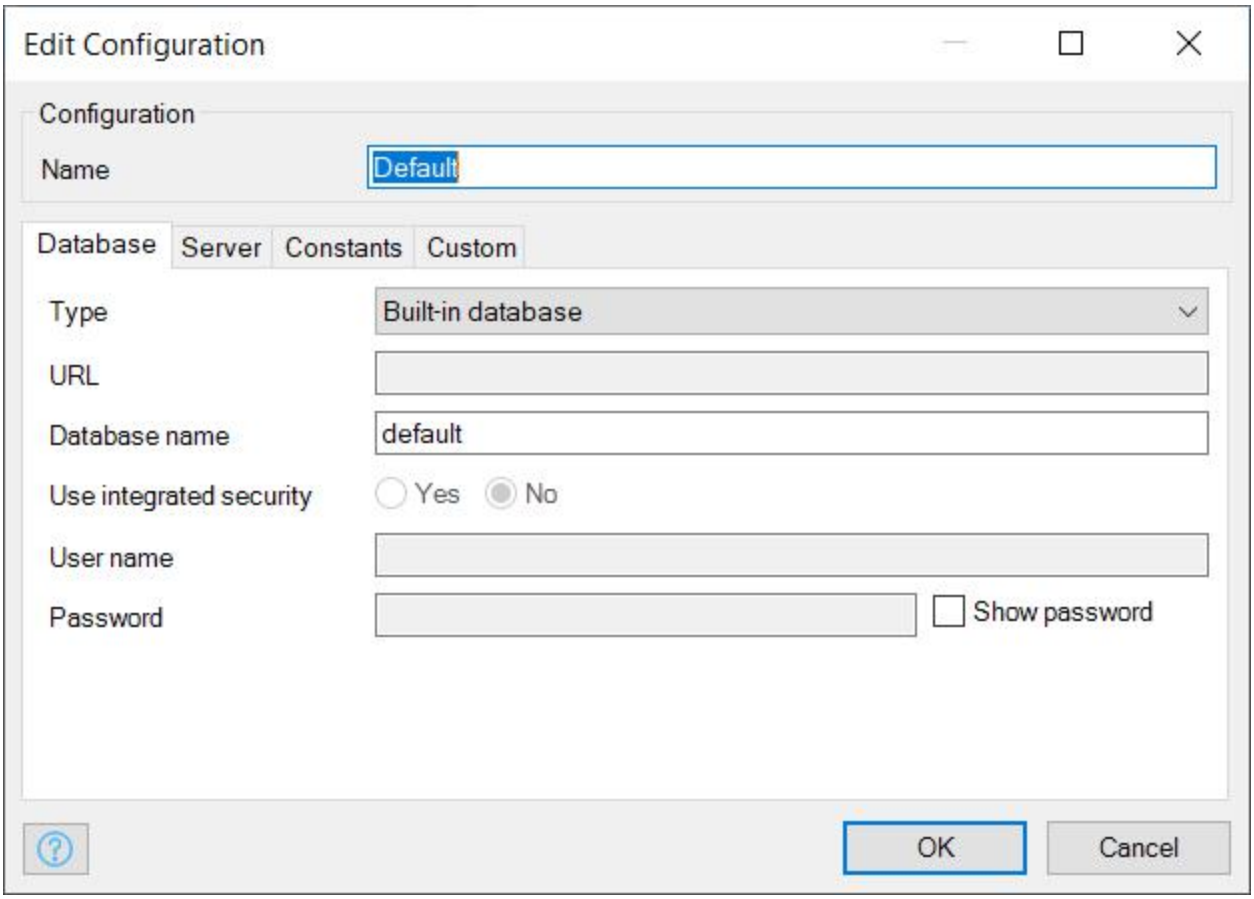
要在 Mendix 中存储数据，需使用域模型，一个包含在 Mendix 中的对象关系映射（即 ORM）。您可以把它比作 Hibernate、Entity Framework 或 Django。现在我们来认识 Mendix 构建数据的方式！



Mendix 域模型支持多个数据库。您可以选择内部数据库（适用于小型应用程序）或 Mendix 支持的任一企业数据库。具体包括：

- IBM DB2
- Microsoft SQL Server
- MySQL/MariaDB
- Oracle 数据库
- PostgreSQL

可从项目设置中为您的应用程序配置所使用的数据库。可通过编辑默认配置，在**项目 → 设置**下的**项目资源管理器**中找到它们。



配置完成后，Mendix 将处理数据库与您的应用程序之间的所有通信往来。这包括为您的应用程序创建和迁移数据库架构，以保留您的数据为前提。为此，Mendix 为您的数据库架构提供独立的版本控制系统。

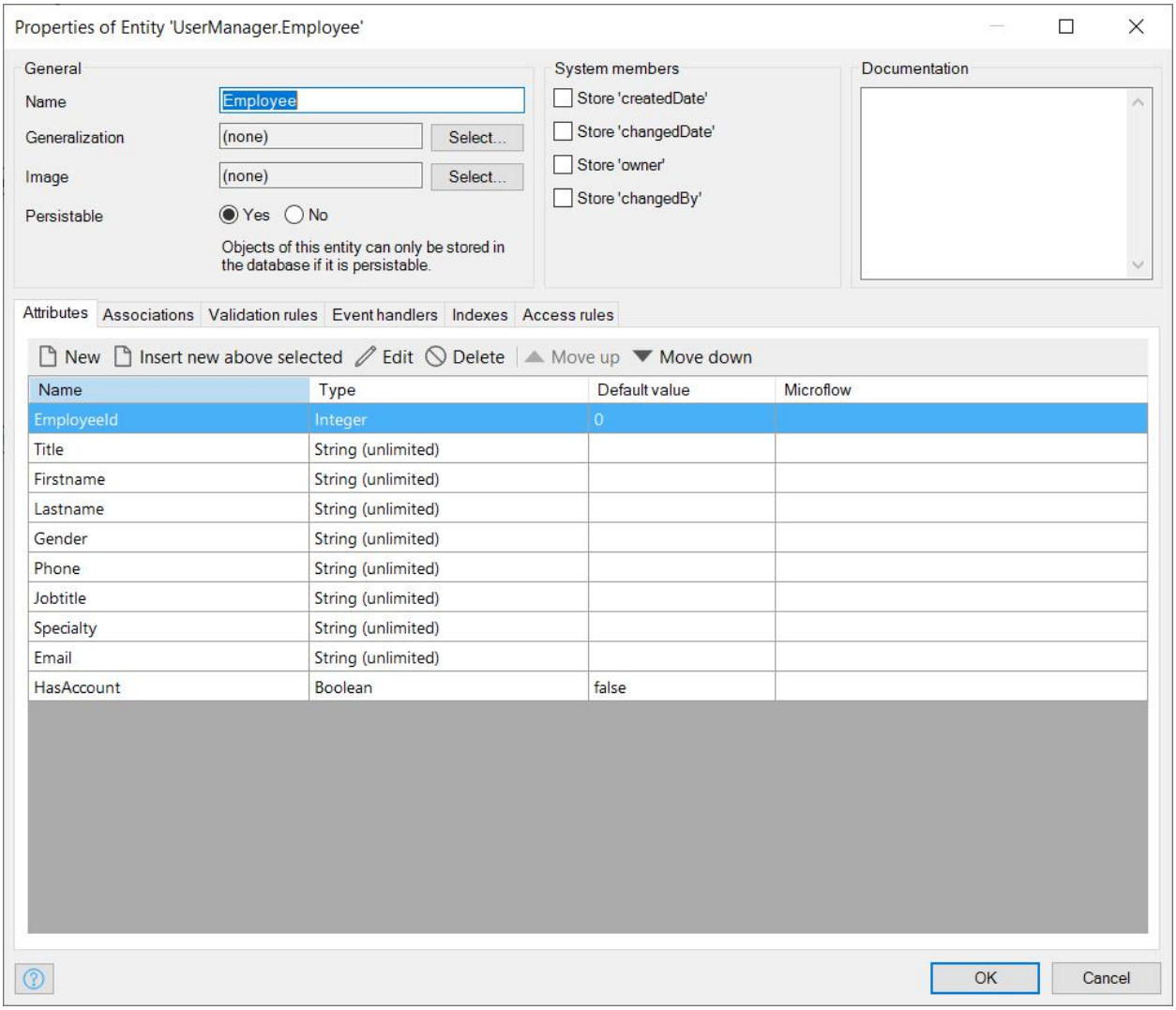
讲座 5.3：域模型实施详细信息

Mendix 将数据存储在实体中。这些实体代表真实世界中存在的各种元素，如客户、发票、订单等。实体通常包含属性，即描述和/或识别实体的特征。例如，客户实体具备名称、电子邮件地址和其他个人详细信息等属性。一个发票实体有一个 ID 号码、一个装运地址和一个账单地址。实体也可以与其他实体建立关系，称为关联。说明此类关系的示例可以是某位客户的订单或与订单对应的发票。

Mendix 实施支持类型发现反射，因此可以使用 *IMendixObject* 基类查询您所创建的每个实体。除其他外，这个类有 *getType()* 和 *getMembers()* 函数。您可以使用这些函数分别检索实体的类型和成员（属性和关联）。

每个成员都继承于 *IMendixObjectMember* 类，该类也有一个 *getType()* 函数。这可用于确定成员是基元还是引用类型（属性或关联）。如果是基元，则它的类型为 *IMendixPrimitive*，且您可以使用 *getType()* 和 *getValue()* 函数来确定它是什么类型的基元以及此特定对象的值。

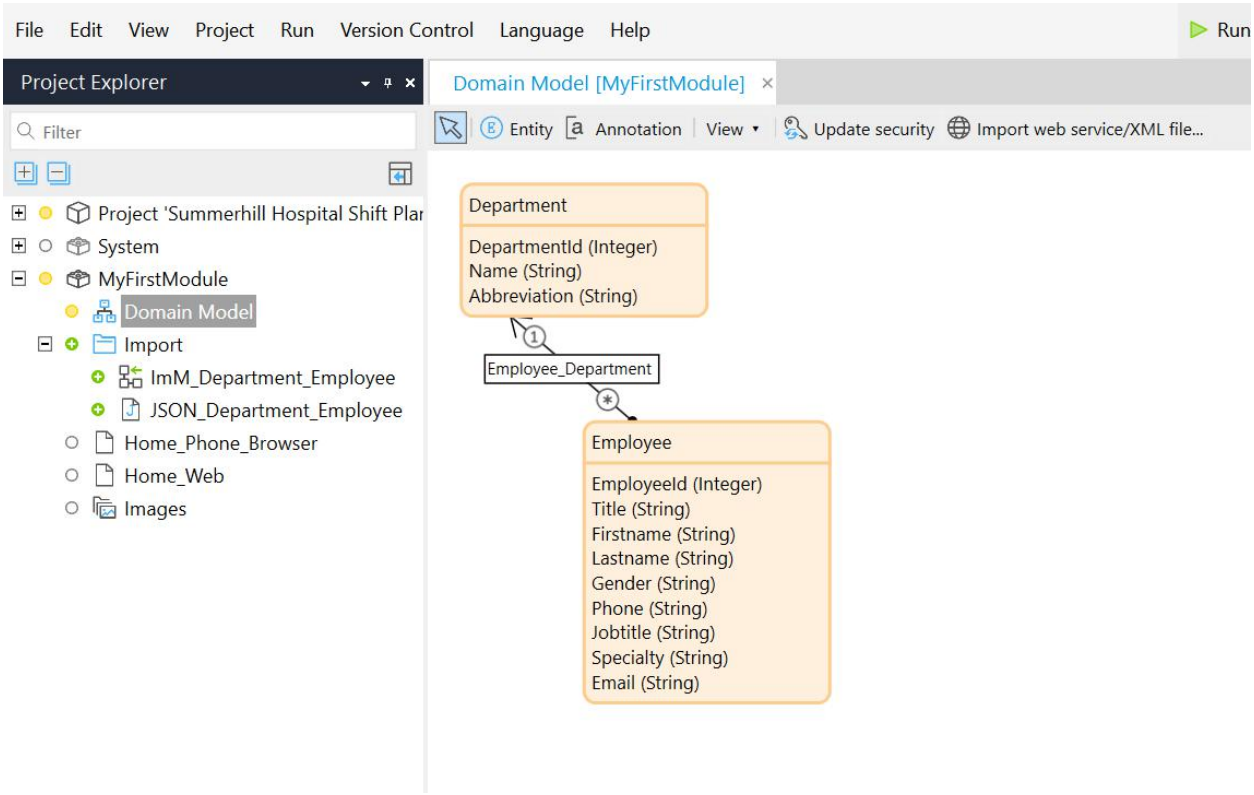
如您所见，所有这些元素都允许您通过 Mendix 中的 Java API 完全访问您的应用程序中的数据。这是 Mendix 在运行时用于解释模型的 API，这样您编写的任何 Java 代码都将成为应用程序中的一等公民。



实体具有多个可供您配置的选项，通常从命名开始。实体还可以具有用于继承的泛化。*在 Mendix*，我们使用类表继承实施继承。每个实体都将在数据库中获得自己的表。子表链接到其父表，因为它们共享相同的唯一 ID。从子类检索数据时，系统将遍历层次结构到父表以从父类检索属性，并且执行 JOIN 以获得实体的完整属性列表。

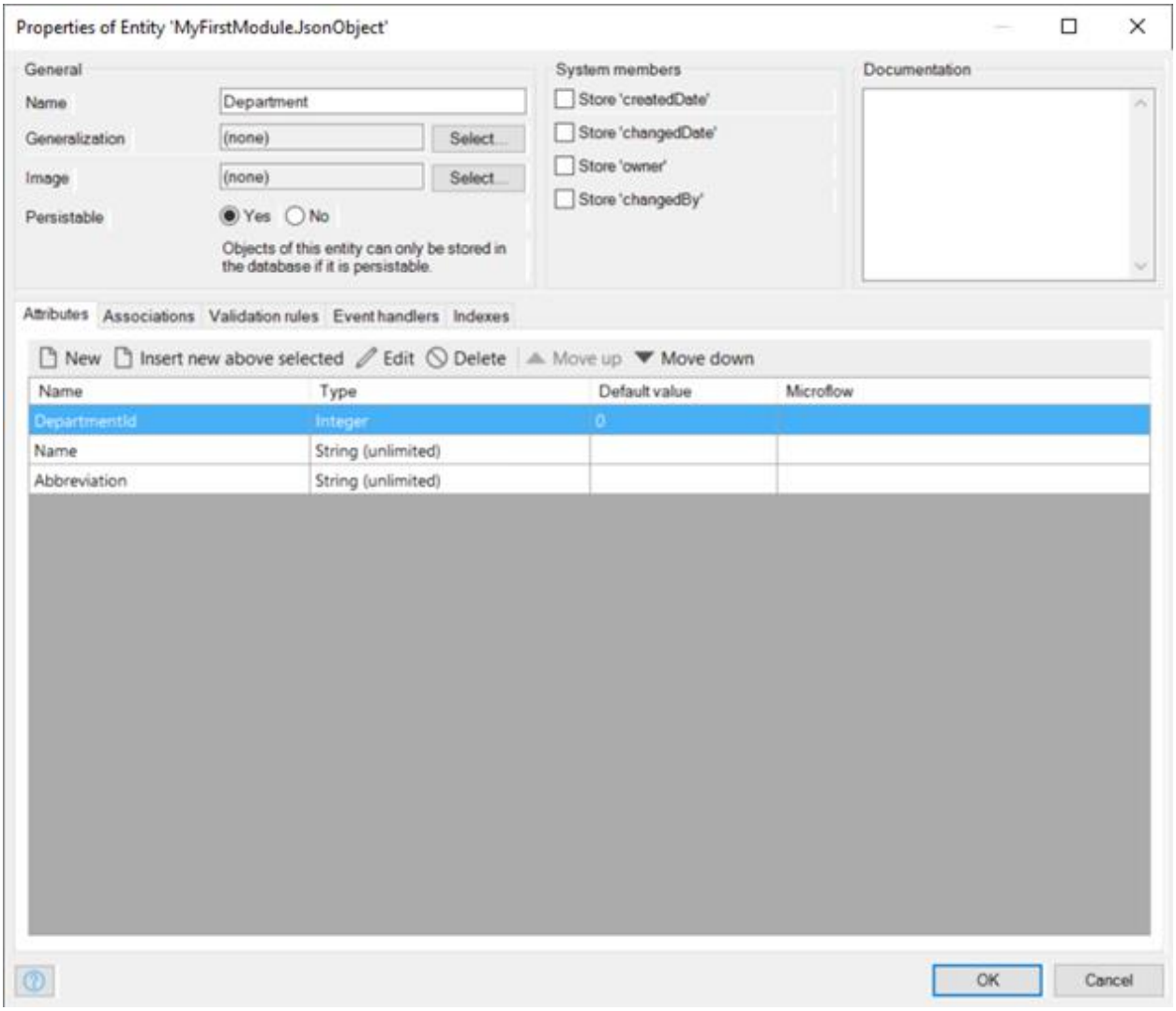
讲座 5.3.1：使实体永久化

现在，您已经掌握了所有的理论，可以确保数据能够存储在您的 Mendix 应用程序中。为此，您可以从项目资源管理器打开域模型窗口。

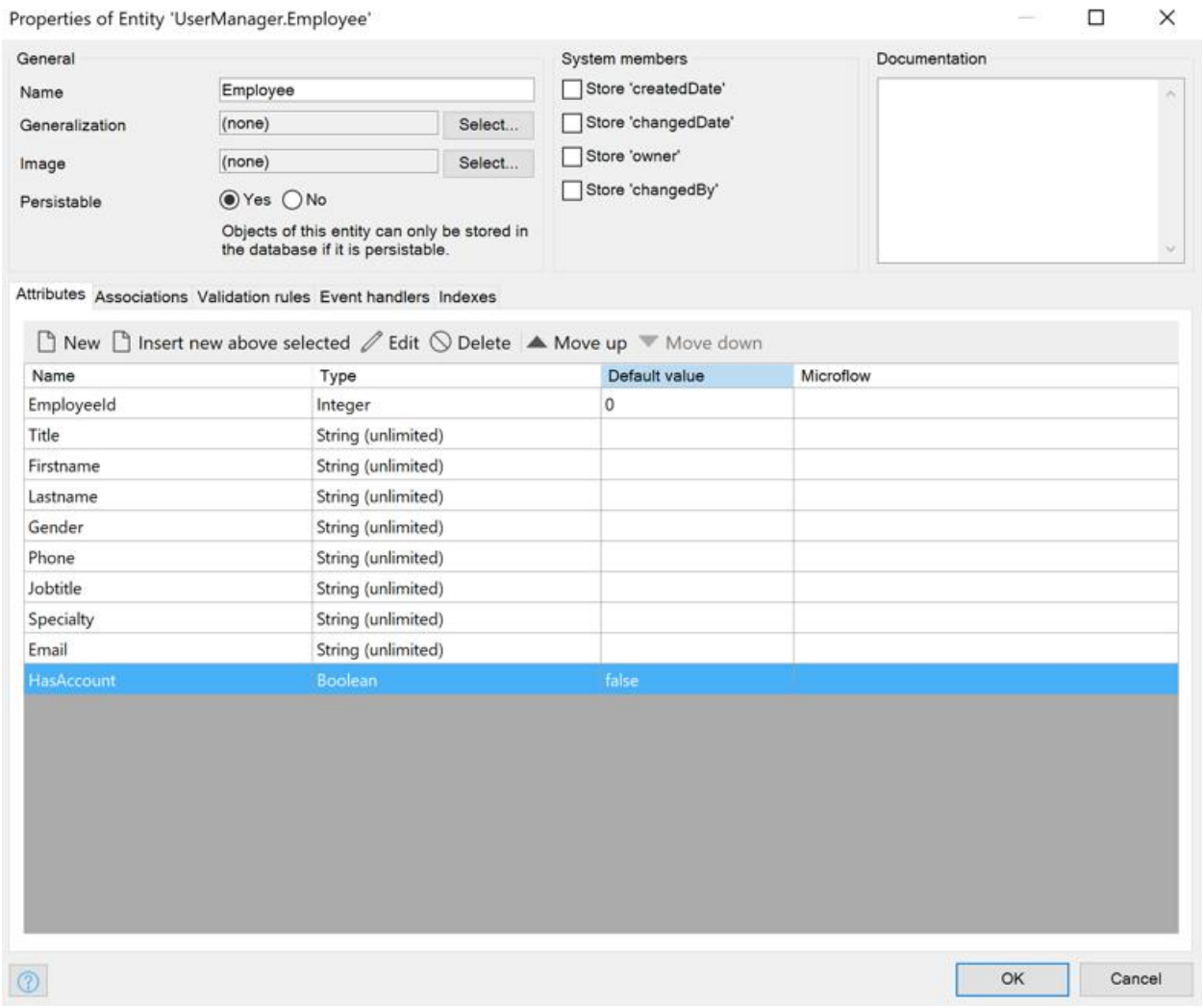


如您所见，域模型已包含实体。这些是由 Mendix 为导入映射而创建的。橙色表示这些实体为非永久实体。要将它们存储在数据库中，需使它们永久存在。这些实体将变为蓝色，即可轻松与橙色的非永久实体区分开来。

1. 双击**部门**实体，然后设为**永久**。



2. 对**员工**实体执行同样的操作。
3. 在**员工**实体中，单击**新建**以添加名称为 **HasAccount** 的**布尔**类型属性。您将在以后使用此功能来确定员工是否被允许登录到系统。



讲座 5.4：实体和属性

您可以使用实体执行其他一些操作，以任何需要的方式存储数据。您可以在实体属性中选择另一个实体进行泛化。这允许您从实体继承成员和设置，如前所述。还可以向实体添加系统成员，这些成员将自动保存特定的系统信息。Mendix 提供了 *createdDate*、*changedDate*、*changedBy* 和*所有者*作为系统成员。

域模型中的实体在属性中存储其数据。这些属性支持以下数据类型：

- 自动编号
- 二进制

- 布尔
- 日期和时间
- 十进制
- 枚举
- 散列字符串
- 整数
- 长整型
- 字符串

关于您之前操控的橙色和蓝色实体，应了解**永久实体**（蓝色）存储在数据库中，而**非永久实体**（橙色）仅存储客户端会话期间的数据。

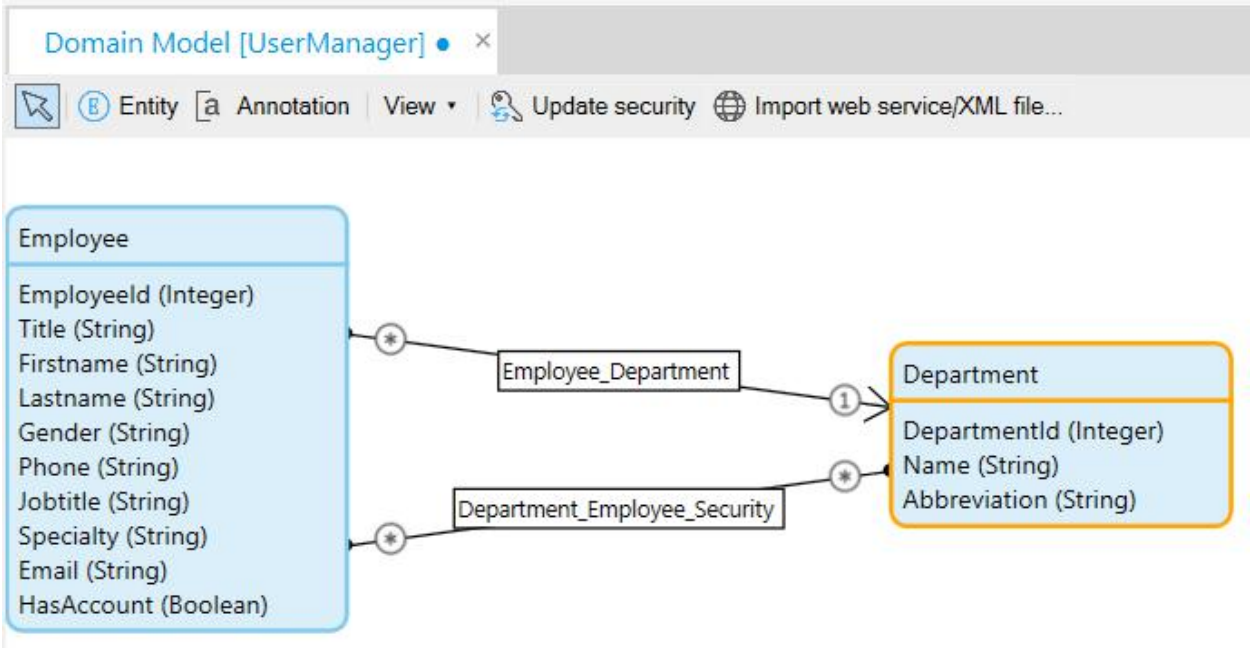
讲座 5.5：支持关系

现在我们来具体谈谈实体之间的关联类型。Mendix 支持常见的并查集：一对一、一对多和多对多关系。

所有关联都使用链接表实施。这便于在不同类型之间切换。此外，在过滤多个关联时，它可以节省大量时间，因为仅连接较小的链接表，而不是连接数据表。

由于链接表的信息相当少，因此在需要保存多对多关系的其他信息时，您通常会发现 Mendix 开发人员在中间使用“信息实体”。

域模型中的关联名称由 Mendix 自动生成，但可以编辑。自动生成的名称遵循最佳实践，默认情况下应使用，除非您需要在相同实体之间具有多个关联。在这种情况下，我们确实建议扩展关联名称，使其更具描述性。扩展此名称带着可识别的目的，描述应如何使用关联。例如，您可以在**部门**和**员工**之间建立一种关系。人员可能在部门工作，但您也希望允许经理在其部门内编辑权限，而无需创建单独的“经理”实体。解决此问题的一种方式是为此创建两个关联，并相应地调整关联的名称。例如，**Department_Employee_Security**。



当实体之间存在关联并且您更改一个或两个实体的名称时，Mendix 将自动重命名关联。

讲座 5.6：命名实体

还记得我们之前谈到的命名规则吗？我们也有一些用于**域模型**的命名规则。

大多数情况下，实体反映人们可以与之关联的真实对象或想法。因此，实体名称还应反映对象本身并确定其用途。实体名称为单数，因为当我们创建该实体的实例时，它们就是单数。例如，使用 **Customer**，而不是 **Customers**。此外，我们建议避免在实体名称中使用缩写、下划线、数学字符或任何其他特殊字符。实体名称还应使用帕斯卡命名法 (Pascal Case)，例如，**HousekeepingRecord** 或 **LogEntry**。

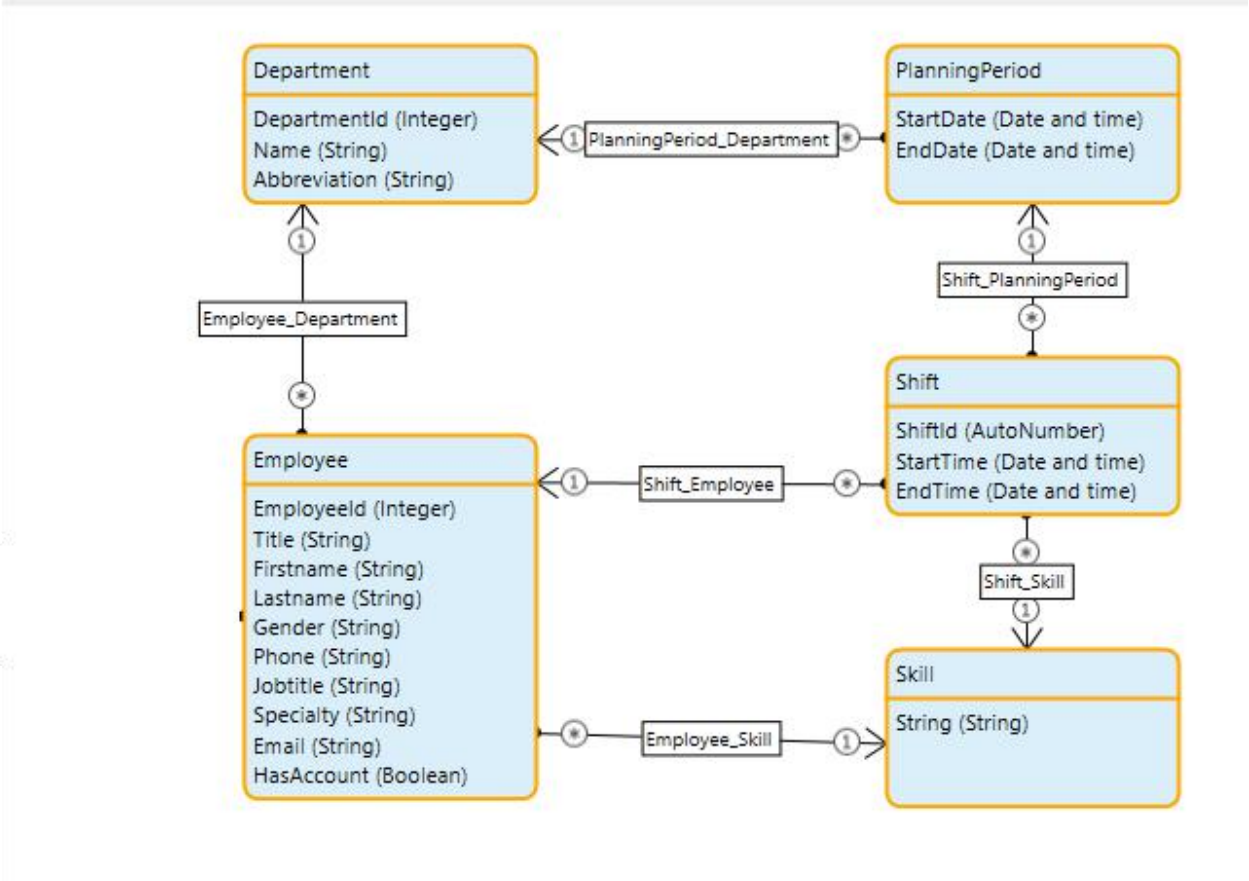
属性应反映所描述实体的可理解属性。建议尽可能避免在名称中使用缩写或特殊字符，但有一个例外。不反映与业务有关的数据，而只是出于技术原因需要的属性，应以下划线 (_) 开头。命名规则再次使用帕斯卡命名法 (Pascal Case)，例如 **FirstName** 或 **TelephoneNumber**。

您可以在此处找到有关实体命名规则的文档：https://docs.mendix.com/howto/general/dev-best-practices#3-2-domain-model

讲座 5.6.1：创建实体

您已经丰富补充了命名规则的知识，现在是时候亲手创建实体了。

1. 从域模型窗口的右上角，拖放实体。重复操作三次，创建三个不同的实体。
2. 双击新建实体以打开属性，指定名称为**技能**。
3. 单击**新建**添加属性。将其命名为**名称**，并将类型设为**字符串**。
4. 单击**确定**。
5. 对实体 **PlanningPeriod** 重复这一步骤，并赋予属性 **StartDate** 和 **EndDate**。确保将两个日期时间属性的**本地化**设置为**否**。
6. 对实体 **Shift** 重复这一步骤，并赋予属性 **StartTime**、**EndTime** 和 **ShiftId**。确保将两个日期时间属性的**本地化**设置为**否**。
7. 创建从**班次**到 **PlanningPeriod** 的 1-* 关联。
8. 创建从**班次**到**员工**的 1-* 关联。
9. 创建从**员工**到**技能**的一对*关联。
10. 创建从**班次**到**技能**的 1-* 关联。
11. 创建从 **PlanningPeriod** 到**部门**的 1-* 关联。



讲座 5.7：对象

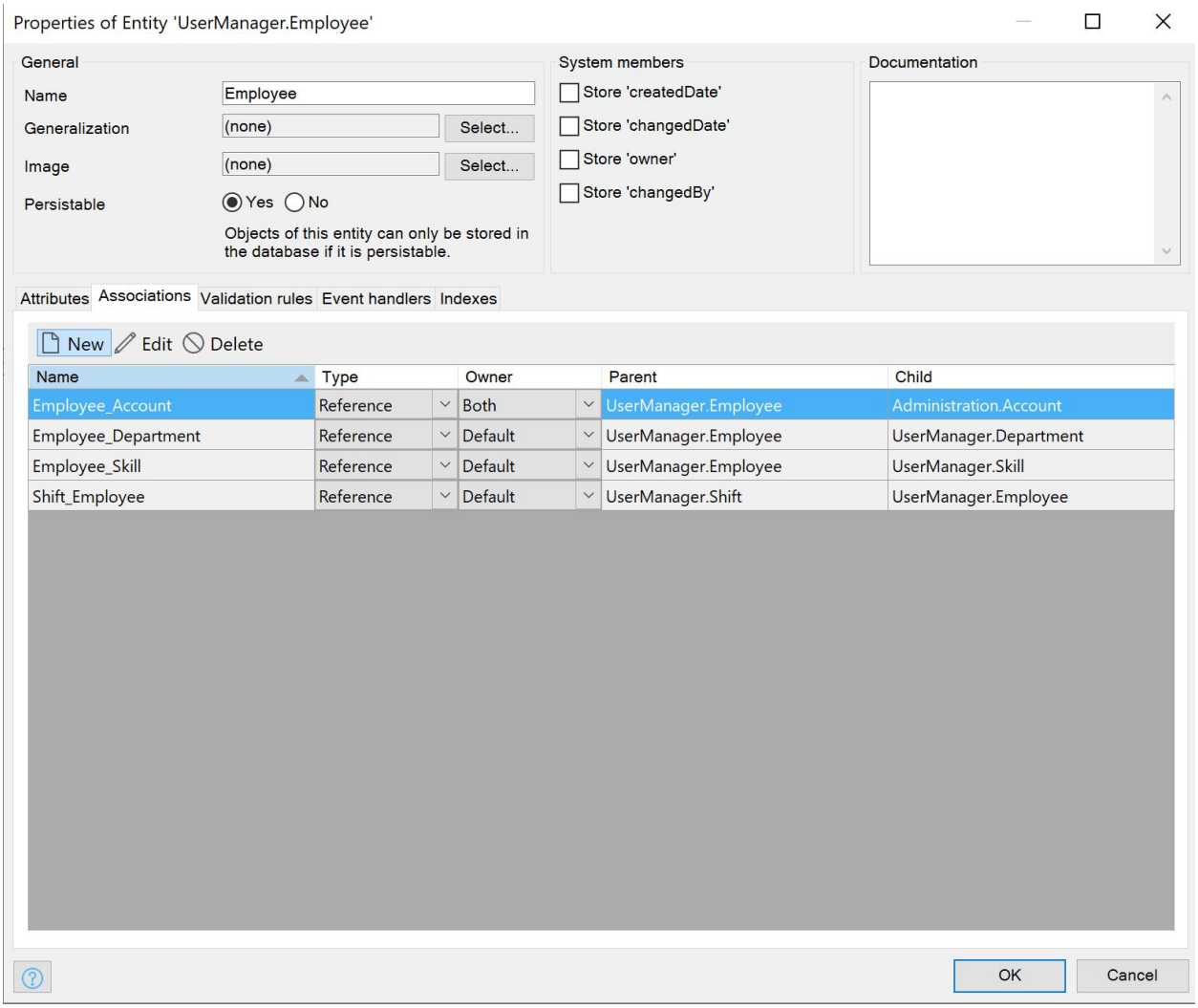
真正将数据添加到 Mendix 应用程序时，将创建**对象**。对象是基于实体的单数实例。输入到此应用程序的员工数据将占用对象。由于 Mendix 具有无状态运行时，因此，对于永久的已提交对象，此

对象存在于数据库中；对于非永久对象、永久但尚未提交的对象或客户端正在使用的永久对象，它们存在于客户端。运行时在执行操作时，内存中仅包含这些对象。

讲座 5.7.1：将员工关联到帐户

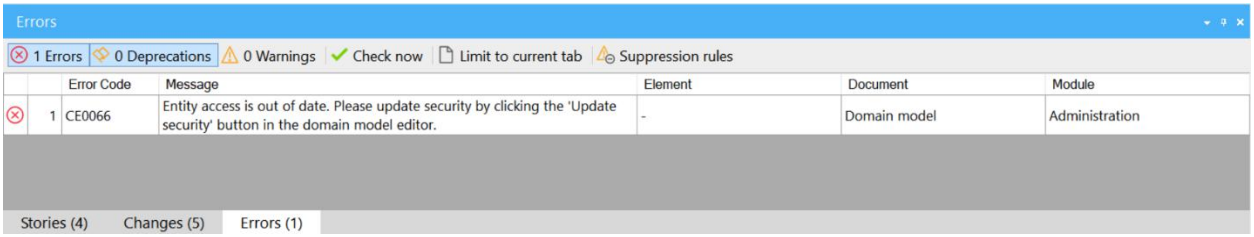
现在我们已经配置了员工实体，应将员工实体链接到在所有 Mendix 应用程序中默认安装的特殊实体，即**帐户**实体。如此一来，即可轻松地根据具体情况为需要访问仪表板的员工提供帐户。为此，我们可以创建一对一关系来完成这一操作，如图所示。

1. 从项目资源管理器中打开**域模型**。
2. 双击**员工**实体。
3. 单击**关联**选项卡。
4. 单击**新建**创建一个新的**关联。**
5. 通过扩展**应用商店模块**，然后是**管理**，选择**帐户**实体。
6. 将**所有者**设置为**两者**，从而将关联设为一对一关联。

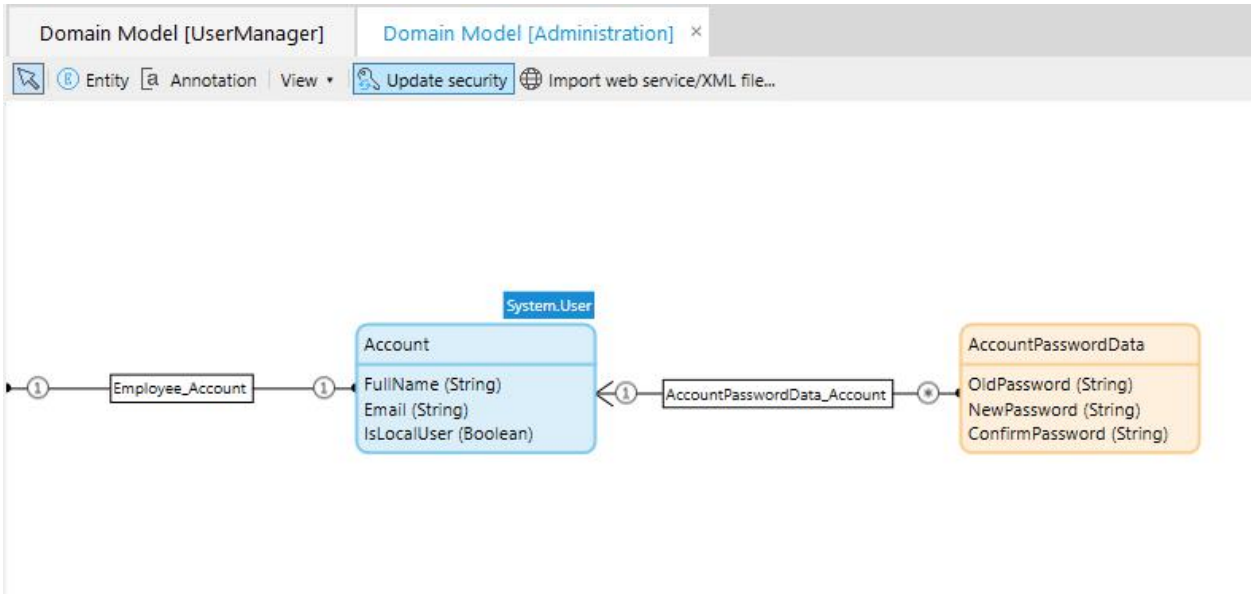


操作操作完成后，单击**确定**。

您已添加与**管理**模块的关联，该模块的实体访问现已过期。可在**错误**选项卡中查看该错误。

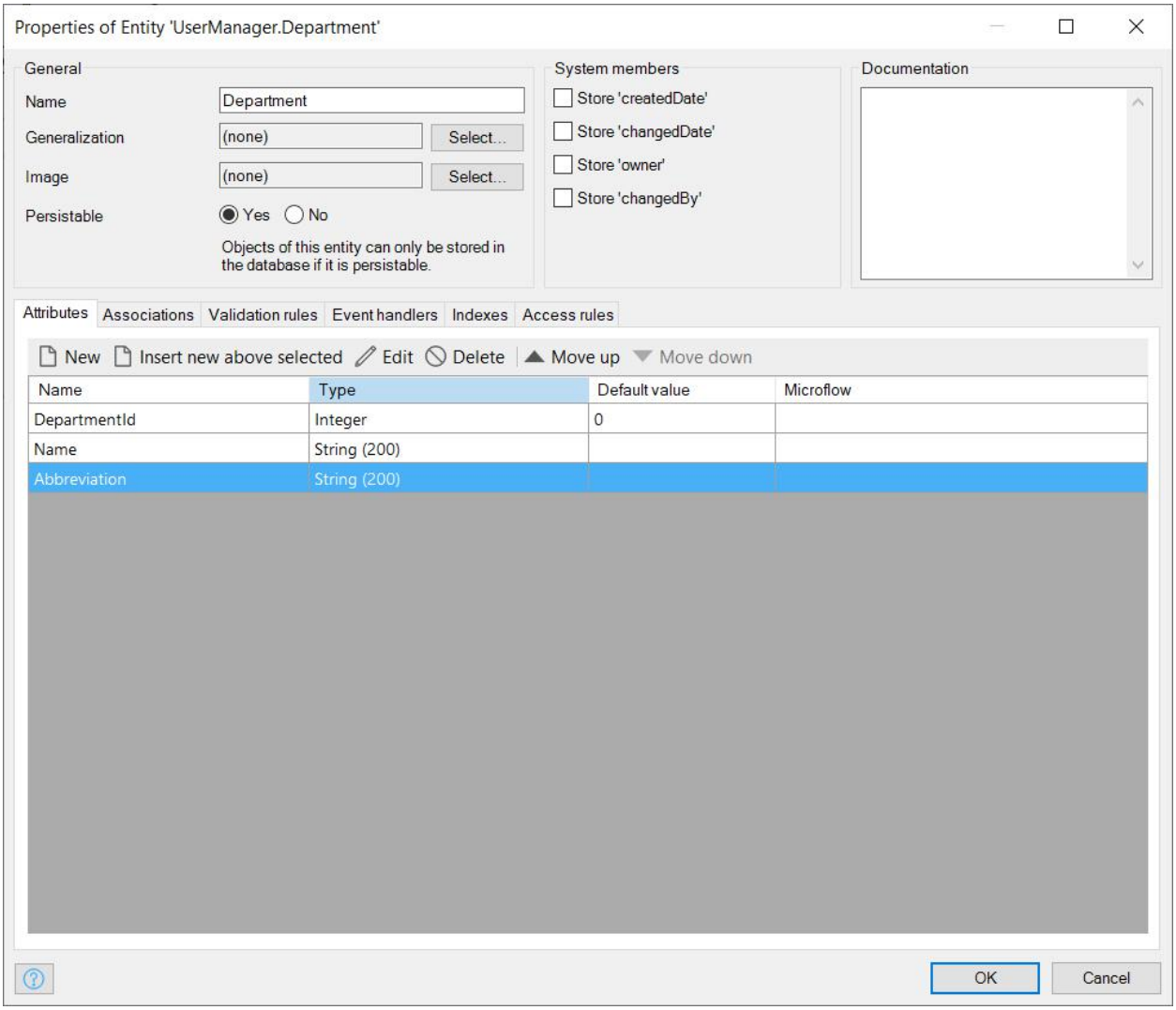


8. 双击错误选项卡中的“错误”以打开**管理**模块的域模型。
9. 单击**更新安全性**以修复错误。



我们都知道来自 **MockHrService** 的信息绝对不超过 200 个字符，因此，可调整实体**员工**和**部门**的属性大小。返回到 **UserManager** 中的域模型进行此操作。

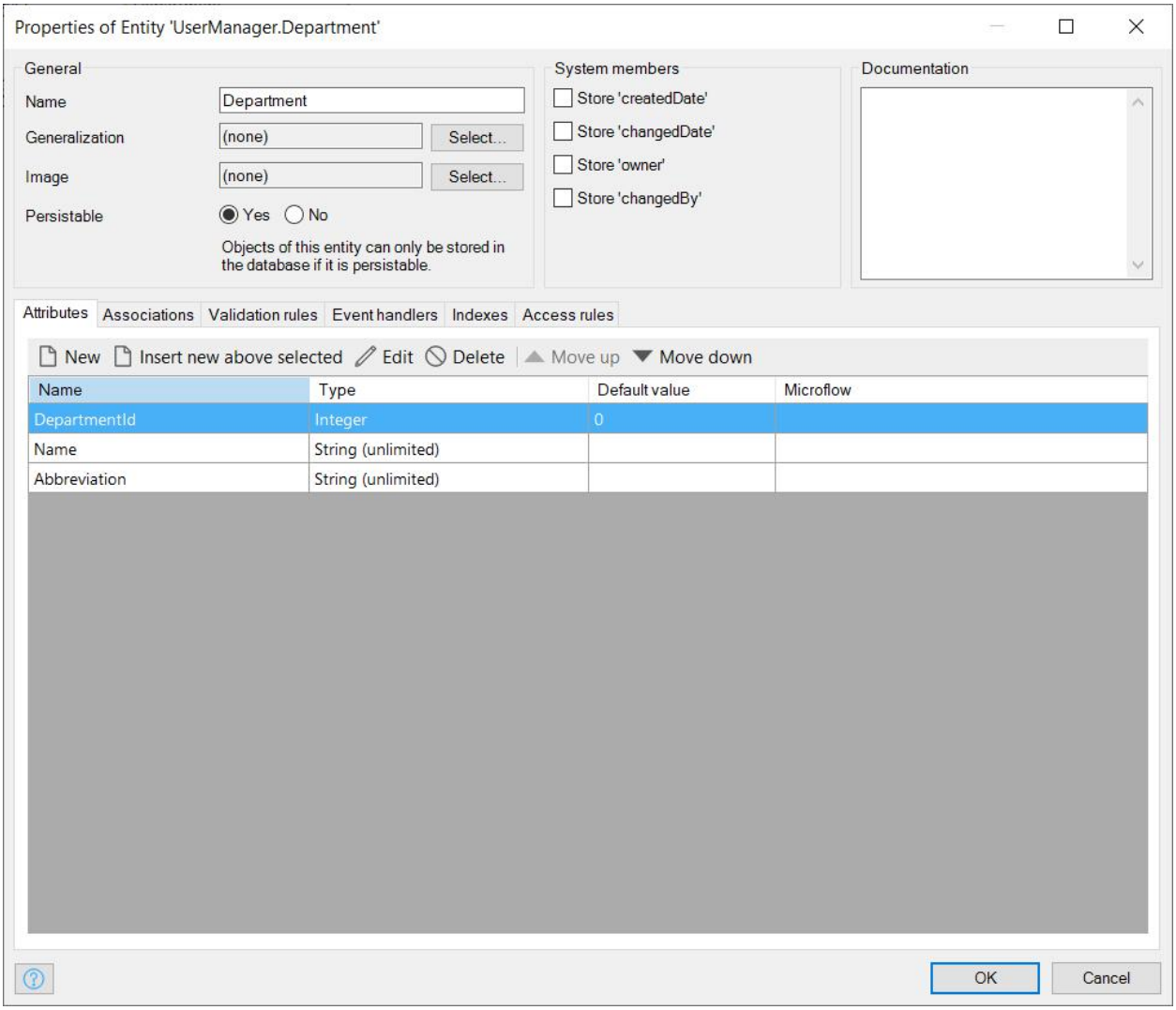
10. 双击**员工**实体以打开。
11. 单击**标题**属性，将长度从**无限**改为**有限**。
12. 对**字符串**类型的所有属性重复这一步骤。
13. 对**部门**实体重复这一步骤。



您已设置所有实体，且属性已设置为正确的大小，现在即可集中关注属性屏幕中的其他选项卡了。

讲座 5.8：其他实体元素

您可以配置实体的其他多个特征。每个特征在实体屏幕中都有自己的部分。



系统成员

单击复选框可激活系统成员。它们将添加到您的实体中，且将在可访问其他属性的任何位置可用。

验证规则

可将验证规则添加到属性。一个属性可以具有多个验证规则。规则可确保属性为**必需**、**唯一**、**等同于**某个值、位于特定**范围**内、与特定的**正则表达式**匹配或设有**最大值**。重要的是要记住，验证规则不能应用于**关联**。

事件处理程序

您可以使用事件处理程序触发以下对象事件前后的逻辑：

- 创建
- 提交
- 删除
- 回滚

索引

此选项卡可用于设置列索引，以利于更有效地搜索。

访问规则

要确保数据安全，您可以配置访问规则。这样可确保只有正确的用户有权访问敏感数据。我们将在本学习路径的安全模块中对此作深入的讲解。

总结

您现已创建了域模型！您已使用两种不同的方法来创建实体，并使用两种不同的方法来创建关联，甚至跨模块。您已设置永久性，了解到继承和可用的数据类型，也认识了受支持的**事件处理程序**和**验证规则**等各种配置选项。这些都需要大量的理论和练习积累，确保出色执行任务，完美收工。现在进入 Mendix 旅程的下一步，了解如何在您的应用程序中管理数据。