本体的概念最早起源于哲学领域， 指的是对客观存在系统的解释和说明。这句话出现在了几乎所有系统介绍知识图谱和本体的材料里。在很长一段时间里，以为这是一句废话，现在对这句话有了更多的体验。

一、前言

知识图谱的本体涉及很多具体概念，如：实体、关系、对象节点（资源）、数据节点（字面量）等。

所以向别人解释什么是本体时需要耗费非常多的精力，巴拉巴拉抛出一大堆概念，最后对方可能没听懂，或者听懂了但是人家根本就不关注这些细节。所以针对不同的听众可以有完全不同的说法。

当对方是市场人员或者客户，和对方提到”本体“两个字，仅仅是为了说明知识图谱构建工程需要做哪些事情。

比如：我们需要三周时间进行业务梳理和本体构建。

那么无论如何也绕不过什么是本体，要解释为什么要耗费这么长时间去构建本体。这种时候可以说的非常粗略：“本体是一个数据模型，这个模型用以约束知识图谱数据的组织方式”。

当时对方是技术人员或者产品人员，可以说：“本体可以理解为关系型数据库的ER模型”。

ER模型即“Entity-relationship model”，其实本体也是这两个概念，实体和关系。本体把名词概念称作一个实体，一个实体是一个节点，各个概念之间的联系称作关系，一条关系是两个相关节点之间的连线。

本体就是定义哪些名词概念成为实体节点和定义实体间关系的模型。如果对方是个Coder，也可以说本体模型类似类图，表达类与类之间的关系。

本体的一个实体就是一种类，本体的实例节点就是类的实例对象。本体的关系就是表达类之间的关系，当然本体的关系类型比类图的关系类型要多的多。

所以本体设计和传统的数据库或者数仓设计一样，需要强依赖于业务流程和业务需求。刚刚接触知识图谱和本体的时候，我曾错误的将本体设计和ER设计等同起来，甚至为了简便直接将ER模型当作本体模型使用。

本篇文章将会分享相关经验，通过举个小例子来讨论下本体设计和关系型数据库ER图的区别。

本体和知识图谱的构建流程可以查看本人在本站之前的文章进行交流：

知识图谱在风控的应用

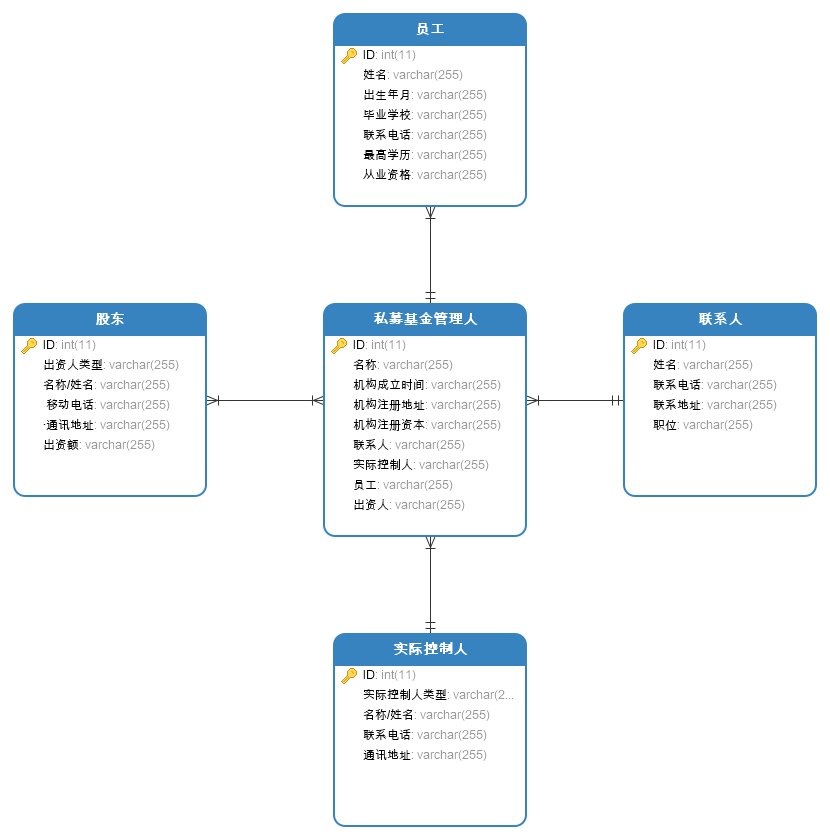
二、场景举例

拿私募基金业务为例，有如下简化版的数据结构。

私募基金管理人和其相关的股东、联系人、实际控制人、员工。根据相关规定：基金管理人的法律主体被限定为公司或合伙企业，自然人被排除在外。

基金管理人通常都会设定为公司形式，尤其是有限责任公司形式。其中股东和实际控制人可以为自然人，也可以为法人。

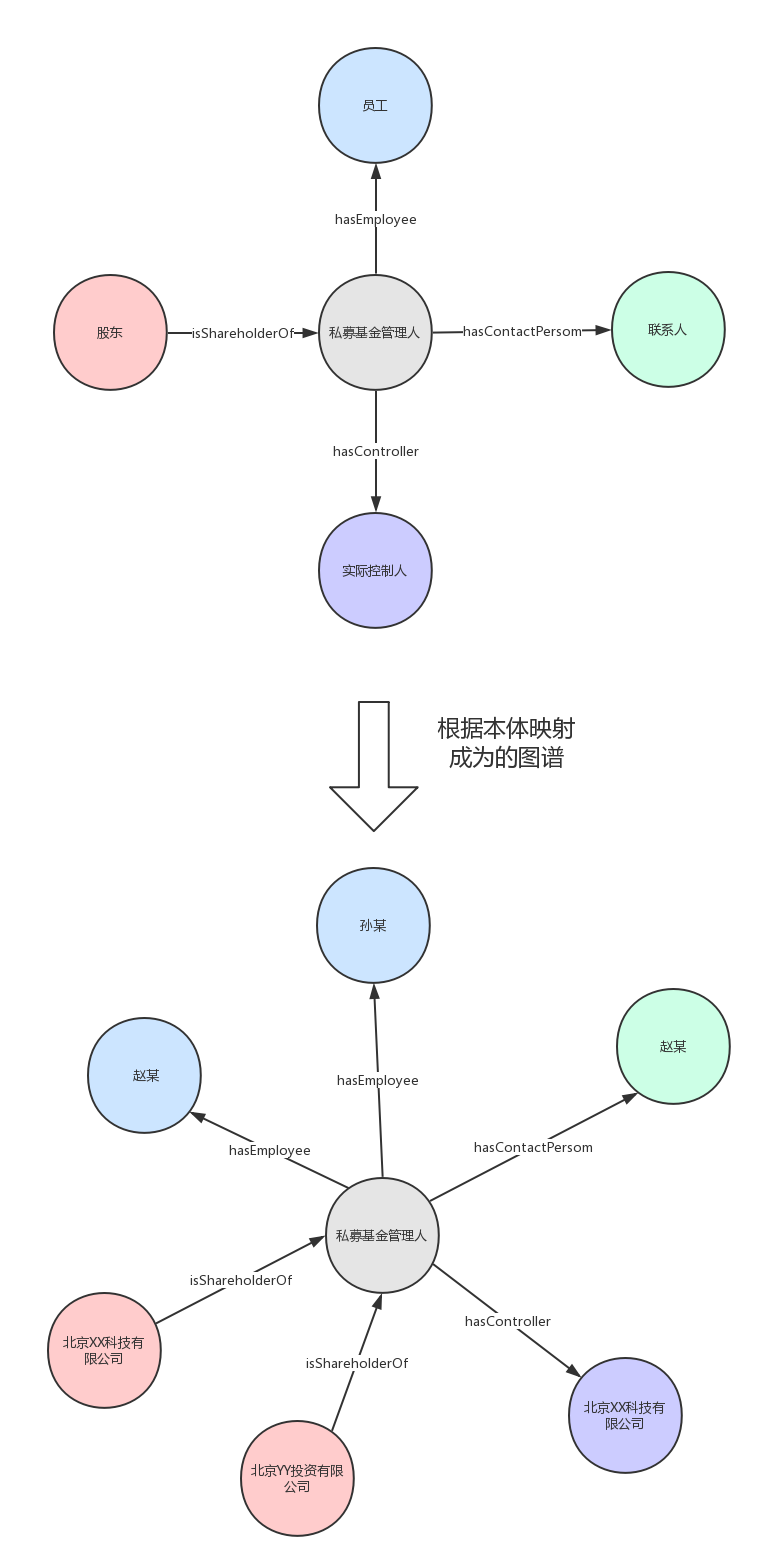
员工和联系人为自然人，一家私募基金管理人对应一个联系人和实际控制人，对应多个股东和公司员工。一个法人或自然人可以同时为股东和实际控制人，一个自然人可以同时作为一家私募基金管理人的员工和联系人。



三、本体设计

如果我们直接把ER模型转化成本体模型，再直接依据该本体进行数据映，可以得到相应的图谱如下。

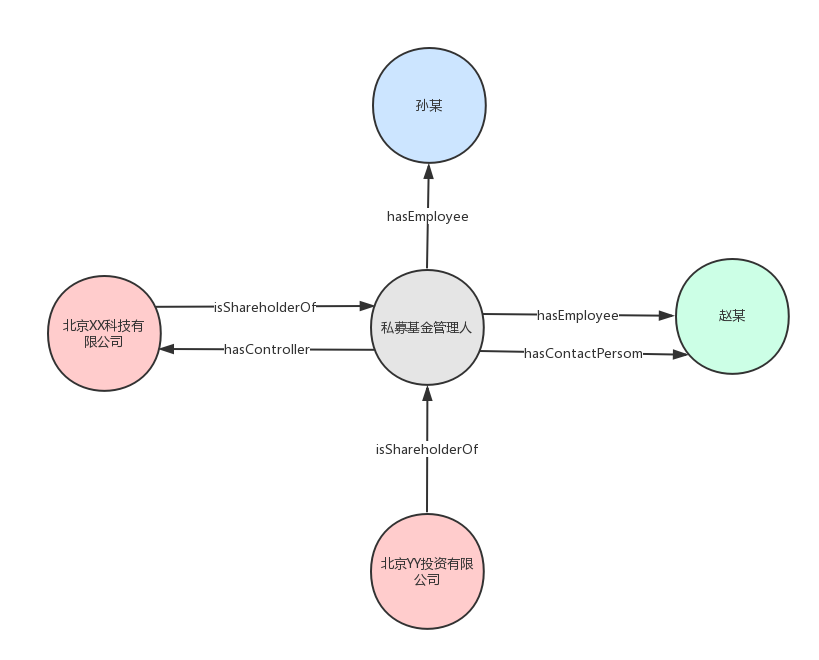
该图谱最大的问题在于：同一个人或者同一家公司会有多个节点，换句话说没有做节点融合。



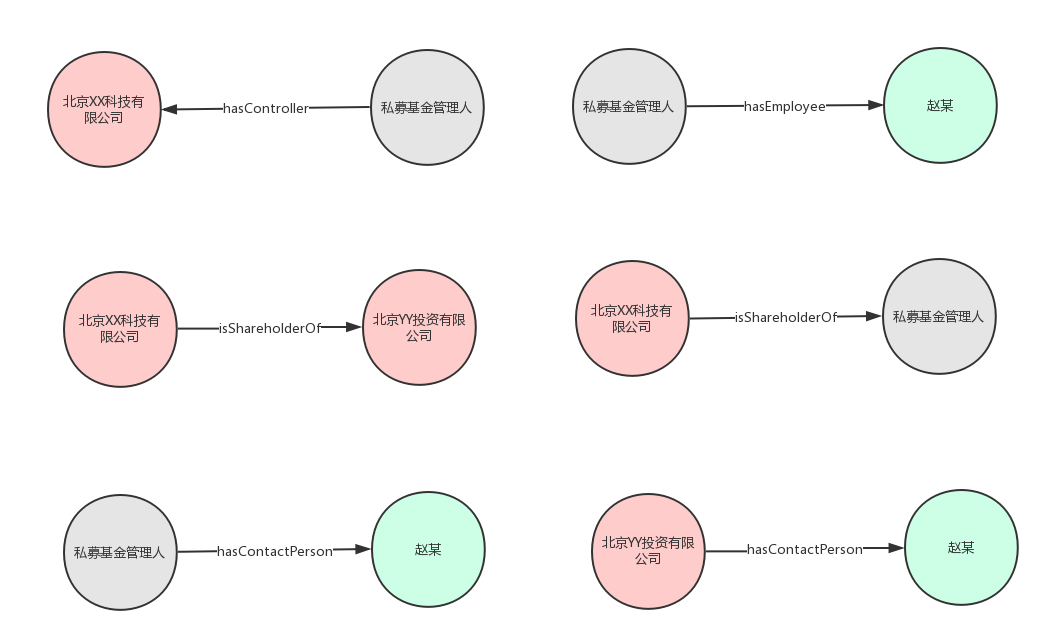
如上图所示：有两个相同的自然人节点——”赵某“，两个相同公司节点——“北京XX科技有限公司”。

这对于知识图谱的大部分应用场景来说是不合理的，在同一个图谱中，同一个实例不能属于两种类型，不能成为两个节点。

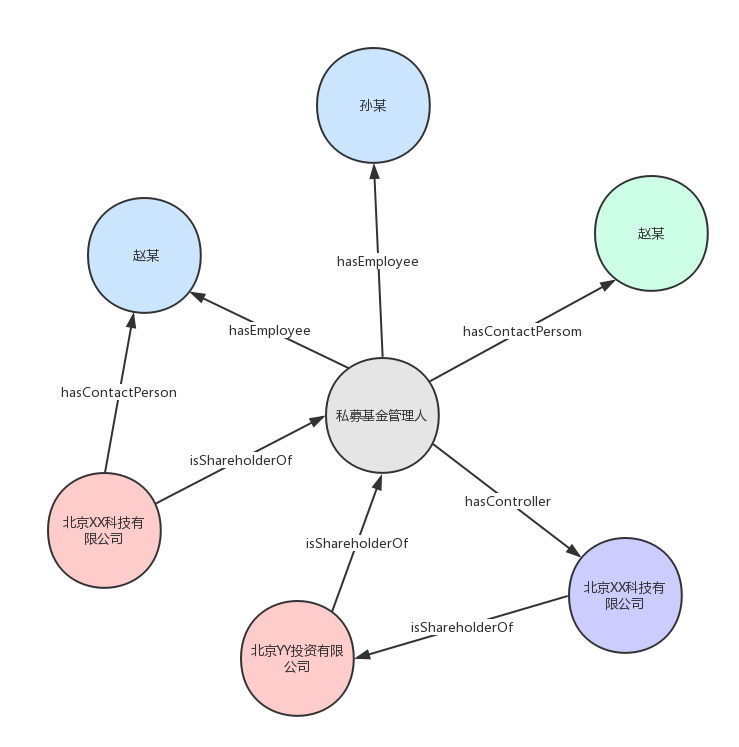
所以上述的知识图谱应该如下：



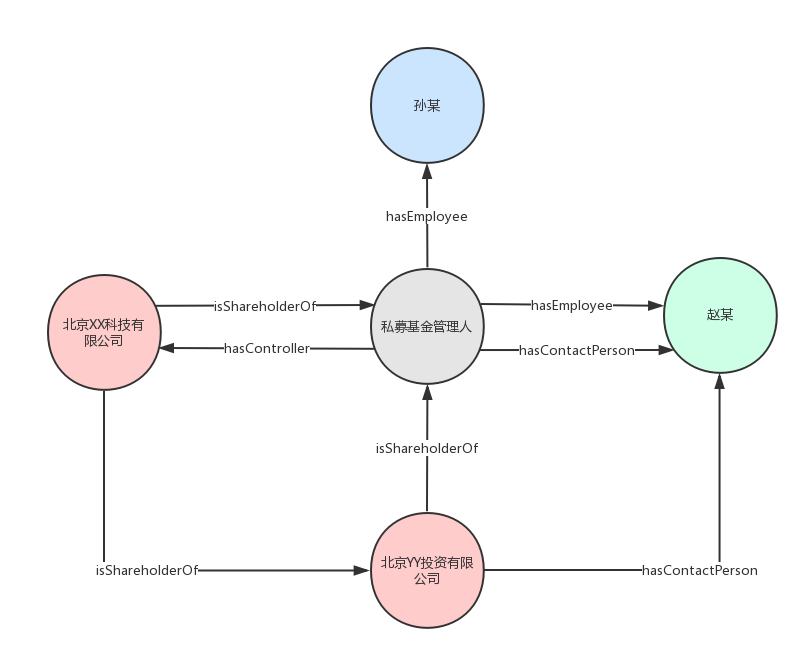
为什么同一个实例不能有不同的节点呢？从应用的角度，在更加复杂从的关系中，很难发现关键节点和业务关注的关系结构。



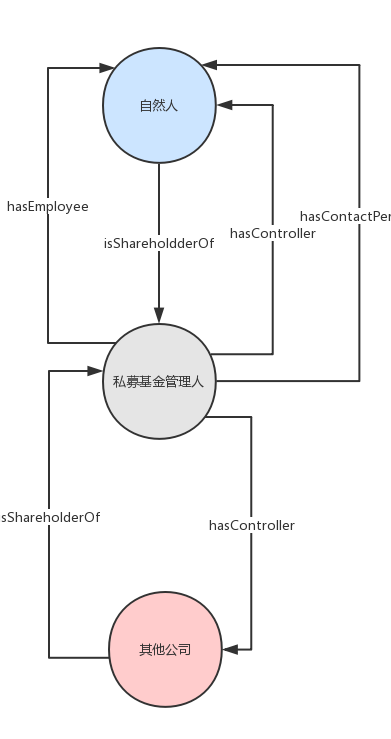
将上述关系以未作节点融合的图谱进行展示，仍旧很难发现多个节点之间存在的关系。



根据上述描述，如果采用进行实体融合后的图谱，则可以非常容易的发现该图谱中存在穿刺投资、持股方和被持股方拥有相同的联系人等结构。



所以由以上的图谱倒推得到一个更加合理的本体模型如下：



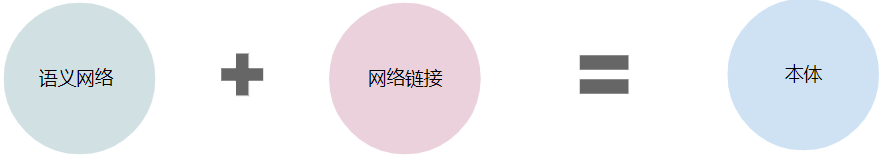
总结

”本体的概念最早起源于哲学领域， 指的是对客观存在系统的解释和说明“——这句话出现在了几乎所有系统介绍知识图谱和本体的材料里。

在很长一段时间里，本人也以为这是一句废话。现在对这句话有了更多的体验：

什么是客观世界，就是一个实例就只有一个。我作为一个自然人只有一个，所以反应在图谱里也只能有一个节点。但是我是作为”人“存在，还是作为“男人”存在，还是作为“员工”存在，是依赖于特定范围的业务需要。结合知识图谱的发展史，

知识图谱起源于语义网络和网络链接，本体的目标史对数据标准进行定义，使得图谱支持数据融合以及便于机器理解和展示。



本体模型的设计和其他数据模型的设计类似，没有一个绝对正确的设计，只能说哪个模型更加合理。

从以往经验看来：一个合理的本体模型大概要满足以下几点要求：

* 有效地支撑业务的分析和决策。
* 正确一致地展示数据信息。
* 拥有广泛的适用性，易于添加新的节点类型和关系