**· XML（树形结构）**

-语法：

1. 需有根和关闭标签

2. 对大小写敏感

3. 必须正确嵌套

4. 属性加引号

-XML架构：

1. 统一结构便于程序间信息交换

2. 统一结构的必要性

3. 是W3C标准

4. 用于XML

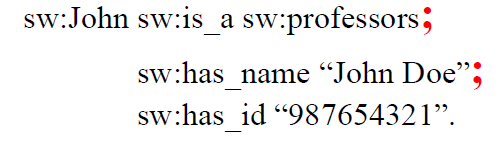
5. 满足此架构的XML文件是有效且优美的

-缺点：语义信息不明，树形结构难以融合

**· RDF（主谓宾三元组结构）使用TURTLE语法**

主语：空节点或资源 谓语：资源 宾语：空节点、资源或文本（文本type）

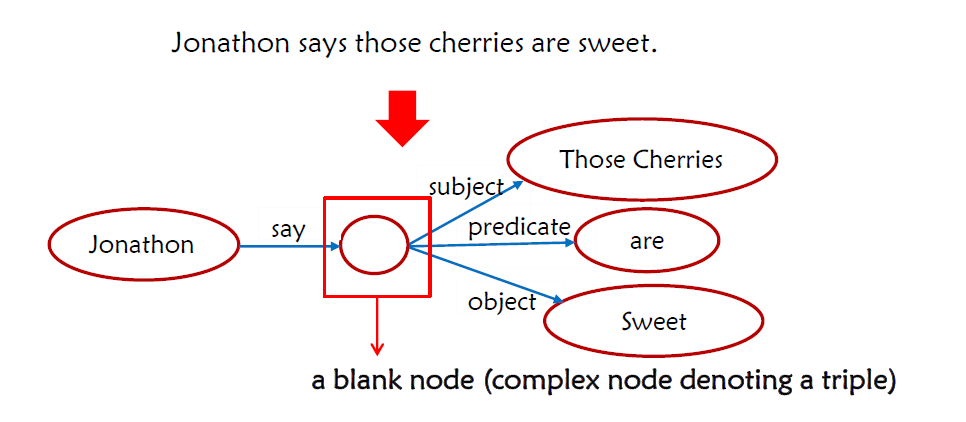
-RDF对比XML的优点：

1. 便于扩展

2. 可以为属性节点添加属性描述

-RDF三元组样例（右侧）分开需要用句号

**· RDFS（RDF的拓展，加入vocabulary）**

-特性：

1. 解释的一致性

2. 可以指定值域与定义域

3. 可以对隐含信息进行推理

4. 提供可以复用的定义

-优点：（与XML相比）

1. 语义清晰（显式赋类，附加语义）

2. 属性和类的层次性

3. 可以引用文本外的信息

图表 1利用空节点做句意拆解

-缺点：

1. 对定义域和值域没有属性值限制

2. 缺少节点数量限制

3. 属性特征描述能力不足

4. 难以表示等价

**· OWL（引入本体概念）**

-本体：一个共享概念化的一种形式化的显式规约。

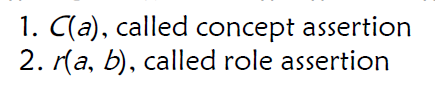
-特性：

1. 表达能力更强

2. 支持逻辑推理

3. 基于RDF语法，变量自主设定力度大

4. 具有一个特殊的“functional properties”，表示多对一关系；inverse表示的一对多

-描述逻辑：TBox（概念） 和 ABox（实例）

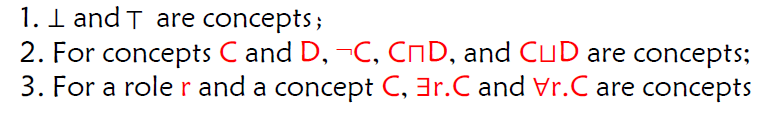
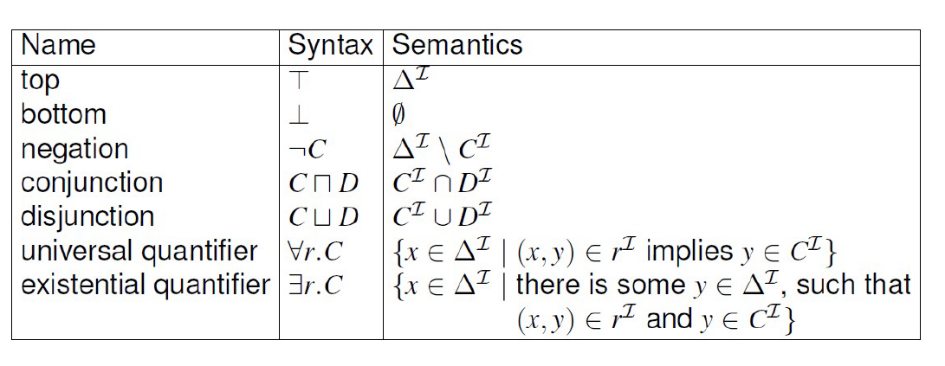
1. 易于理解

2. 标准表达

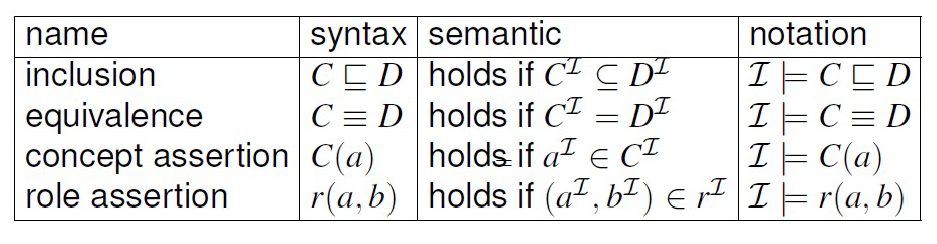
3. 表达能力强

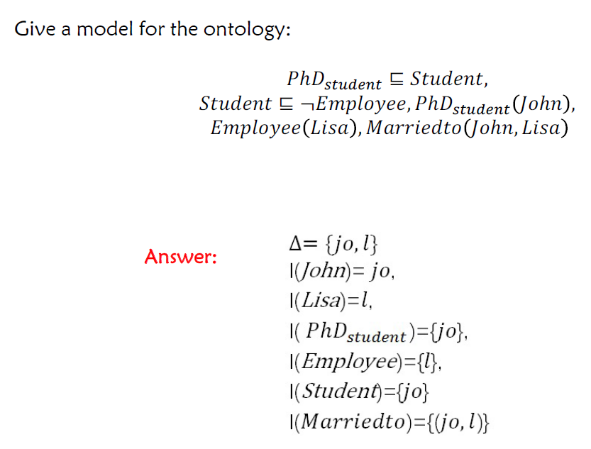
图表 2ABox的关系语法

4. 可以自动推理

以下是描述逻辑的语法：

图表 3ALC的关系语法（TBox有包含关系）



-逻辑建模：检查一致性

**· 基于图的知识推理**

-逻辑推理：

-前向推导：用于专家系统，使用规则向下搜索

-后向推导：用于蕴含内容查询以及问题回答，规则重写查询

-用于检查一致性（满足一致性的就是模型）

-统计推理：基于嵌入的推理

图表 4正确建模举例

-transE：(head, relation, tail)三元组

使用判断优劣，越小越真

L1是曼哈顿距离，L2是欧氏距离

-特性：

1. 向量表示三元组

2. 基于统计的模型

3. 只能表示一对一关系，不能表示一对多/多对一/多对多和对称关系

**· 构建知识图谱：从异构数据中抽取整理数据**

-RDB2RDF：将关系型数据库映射为RDF模型

-方案：

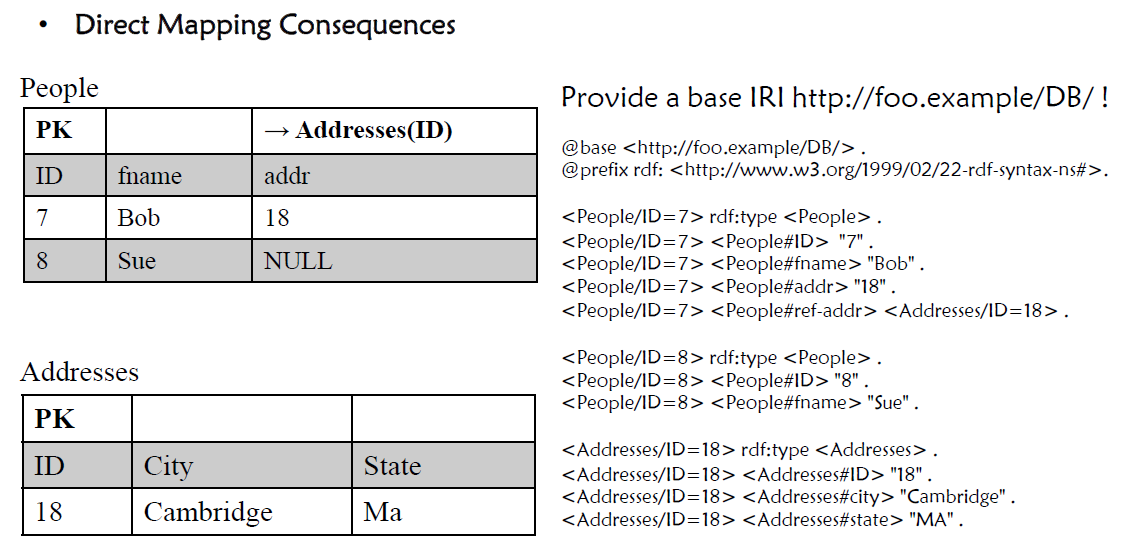
1. 直接映射（完全自动化的，一列一类）

2. R2RML（需要人工设定映射）

-直接映射：简单直观的变换

-IRI用于表征：1. 表格内的类 2. 每行的资源 3. 每个单元格的资源属性 4. 外键引用

-如下为映射样例：



-R2RML：使用外部自定义映射

-需要的内容：

1. 数据库表格

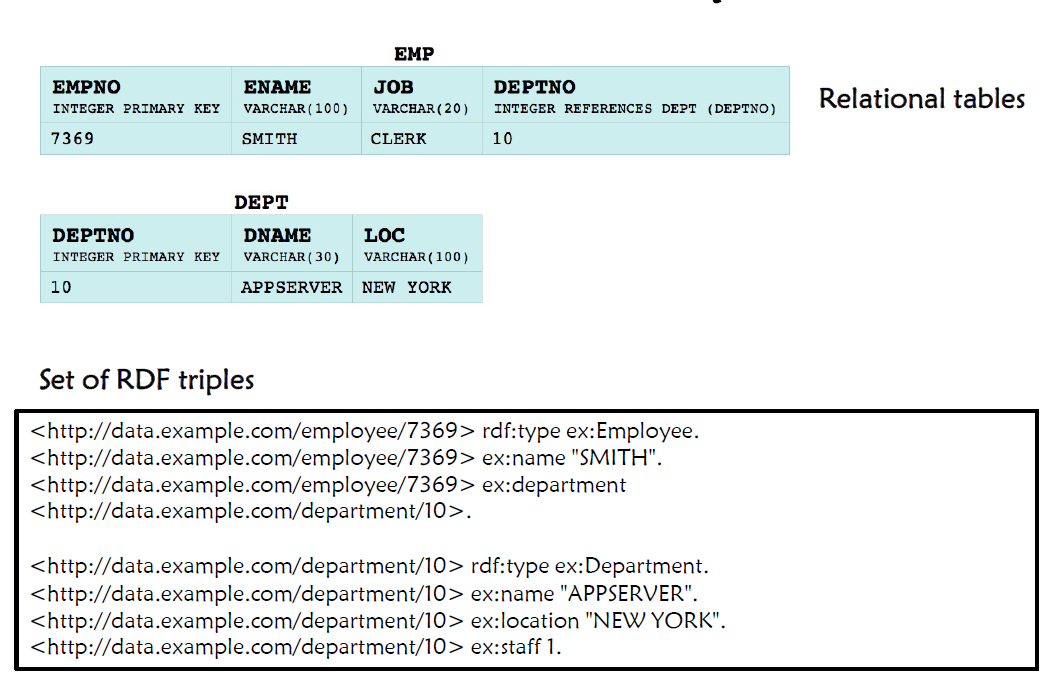
2. 数据库视图/SQL查询样式语句，即“R2RML视图“

-triples map包含内容：

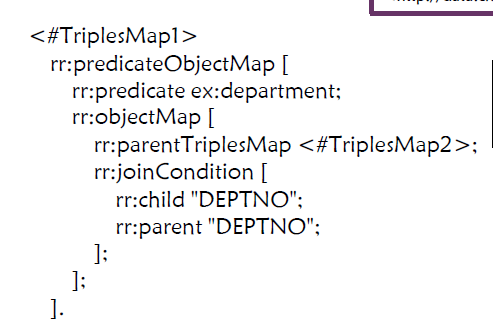
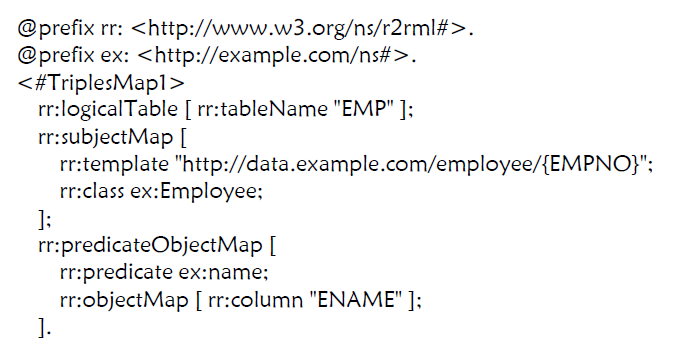
1. 逻辑表格名称

2. 主语映射（主键）

3. 多个谓语宾语映射



映射到view（左）和带外键的映射（右）：



注意：parent是被引用的表格，joincondition表示该项相同则join。

**· 关系抽取的三步骤**：Entity Linking->Column Typing->Relation Extraction

-实体链接：将单元格内字符串mention映射到给定知识库中的指代实体上

-步骤：候选生成->实体消歧

-候选生成：辨别可能的指代实体

-方法：

1. 基于字典的：使用wiki页面进行相关性打分

2. 基于字符串的：Levenshtein-增删改距离/Jaccard-

3. 基于语法的

-实体消歧：从可能的实体中选择最大置信度的作为指代实体

-方法：

1. 局部消歧：只用给定mention以及候选实体的上下文，不拓展到全局

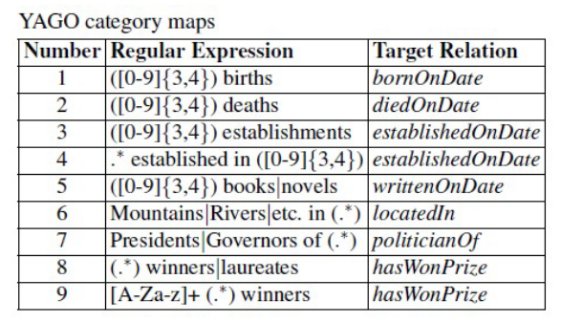
2. 全局消歧：使用候选实体全局的上下文关联

-列分类：将每一列转化为一类，表头为类型，在该列下的单元格都属于该类

-关系抽取：表头为关系，在该列下对应的单元格有行-关系->列

**· 半结构化数据生成知识图谱**：Fact Extraction->Type Inference->Taxonomy Induction

-作用：线上数据为半结构化的，从真实世界数据中生成关系图谱

-事实抽取：提取主谓宾三元组

-方法：1. 基于Infobox 2. 基于类别栏（正则表达式，右侧）

-类别推理：提取实体的类别从属信息

-方法：

1. 基于Infobox

2. 基于类别栏（启发式的）

-方法：选取概念性的语句，进行遍历

-问题：1. 语言结构原因某些语言不适用

2. 无法捕捉语义关联导致的实体和类别错配

3. 基于文本：Type Extraction->Type Disambiguation

-Type Extraction：使用文本的第一句话（简介）提取is-a

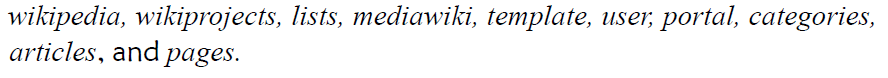
-Type Disambiguation：使用Wiki搜索接口

-分类学归纳：分类学即包含概念实体以及独立实体，表示类别从属关系的有向无环图， 这里为由在线的百科推导类别，删减非is-a的关系

-Wiki分类系统的问题：从属关系不严谨

-步骤：预清理->基于语法处理->关联性处理->Lexico-Syntactic处理->推理

-预清理：移除包含以下字符串的类别，为is-refined-by的类别



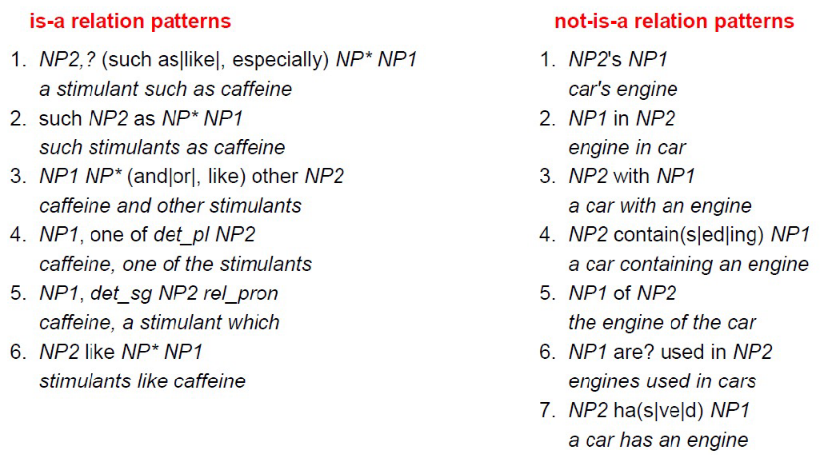


-基于语法处理：具有同一句法定义且位于根的分类对保留，反之删除



-关联性处理

-正则表达式匹配（下图）



-推理：根据传递性

**· 非结构化数据生成知识图谱**

-任务：

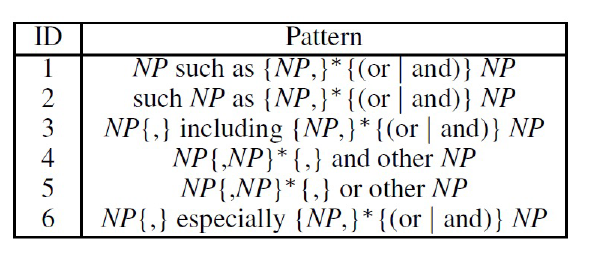
1. Is-a关系抽取（利于建立分类）

2. 术语提取（利于建特定领域内的知识图谱）

-Is-a关系抽取：

-方法：1. 基于模式的 2. 相似性

-基于模式的：Hearst Patterns



-术语提取：术语在某一领域内有极强的表意作用

-特性：

1. 与命名实体识别、关键词抽取等任务相关

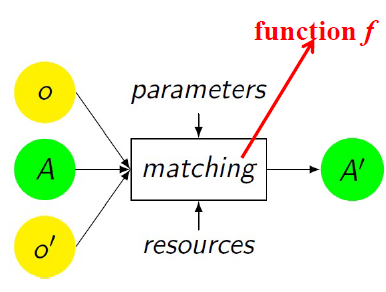
2. 与领域高度关联

3. 本体构建、文本提取、知识图谱构建等任务的关键

-基于统计的TF-IDF：

TF-IDF=

TF是本篇文章内的术语出现次数，IDF的log内部分是总文本数除以包含该术语的文本数加一（smoothing），TF-IDF是针对于同一文本中的不同词进行排序，而不是对不同文本进行排序。



**· 知识图谱对齐**：本体匹配+实例匹配

-本体匹配：在不同本体之间寻找类或属性之间的相关性

-优点：

1. 从多个局部本体构建全局本

2. 本体之间的知识复用

3. 解决异构的问题

4. 可以进行多分布的资源查找

-基于元素层次的方法：忽略实例间关系单独分析

-基于字符串的：

1. 前缀/后缀

2. N-Gram：

-基于结构的方法：

-基于图的：将本体转化为图结构，具有相似语义的节点邻居节点相似度高

-基于分类学的：在图中只考虑is-a关系的邻居节点计算相似度

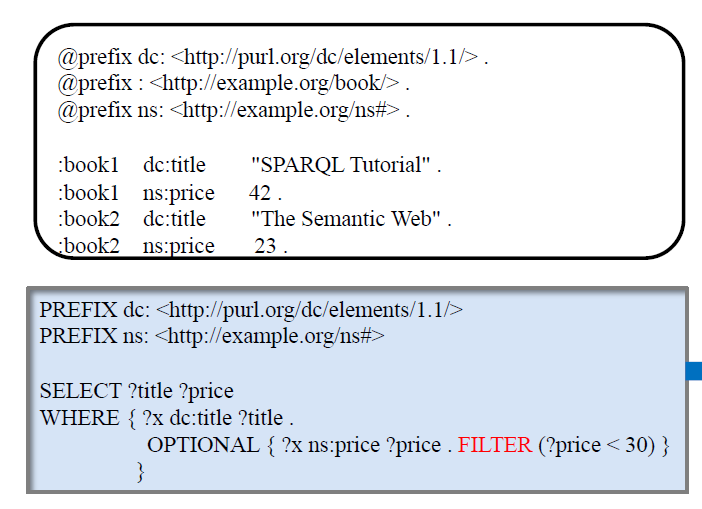
-基于实例的：对比两类下的实例相似度

-实例匹配：寻找并匹配相同物体在知识库中的的不同实例

**· 知识图谱查询**

-SPARQL：通过描述子图的模式进行匹配获取结果

-基础的结构：?x foaf:name ?name. 注意一句查询结束后后面要加句号

 -查询样例（右侧）

-WHERE限制：1. OPTIONAL：可有可无，多个并行需拆分

2. FILTER：直接跟在查询语句后限制本句

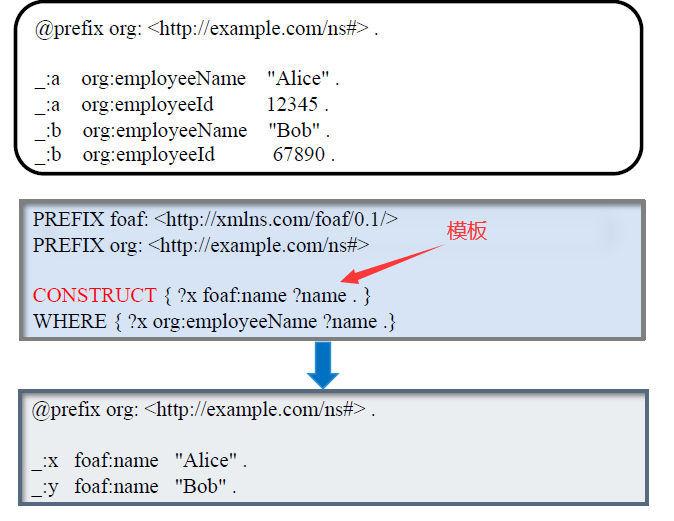
3. UNION：“或”，需要并列查询

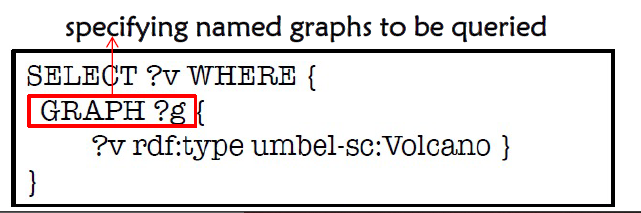
4. BINDINGS：构建新变量

5. BOUND：存在该值

-SELECT限制：1. DISTINCT：只取不重复的

2. A AS B：将A的值赋给B，取B

 -查询子图：满足具有该三元组条件的图



-构建新图CONSTRUCT需要给模板（右图）

-Cypher：高效、描述性强的知识图谱增删改查

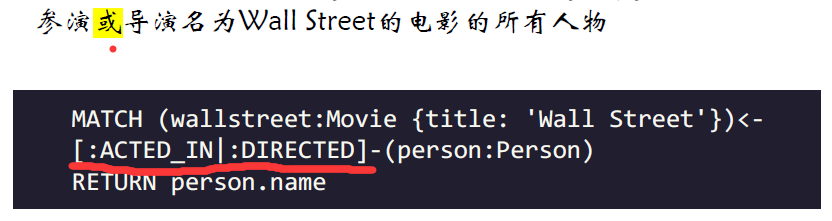
-格式：(n1{prop:})-[r{prop:}]-(n2{prop:})

-特性：1. 箭头可以带有方向，不确定时可以用- -

2. 不确定关系属性时可以直接忽略，用(n1{prop:})—(n2{prop:})查找

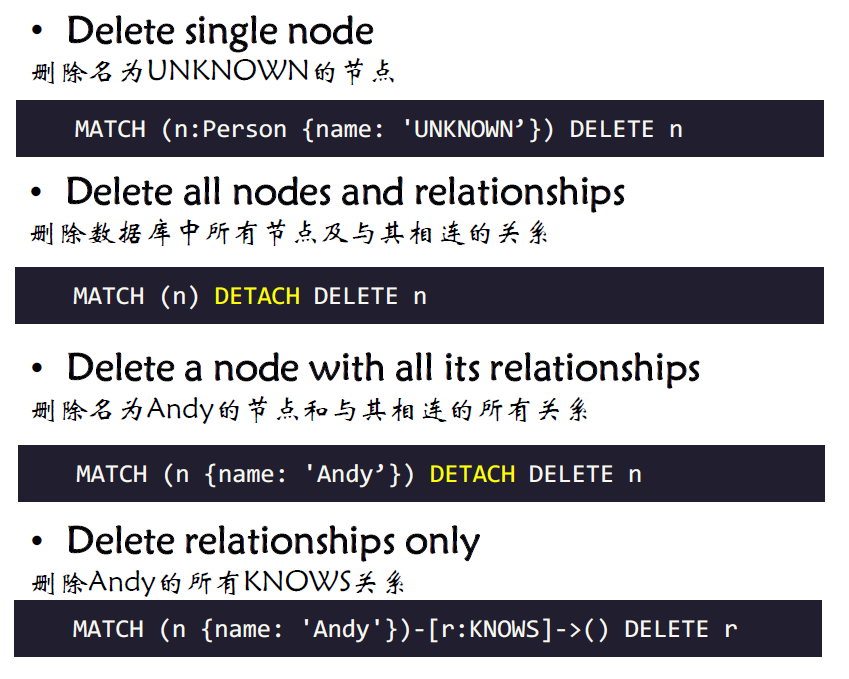
-语法：1. CREATE：创建节点/关系（三元组形式）

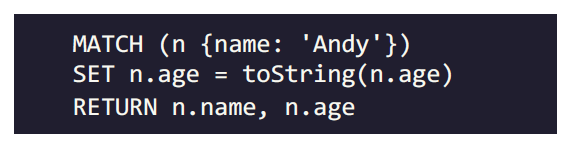
2. MATCH：查找满足条件的实例，用RETURN获取

 3. |：表达“或”（右图）

4. MATCH+CREATE：用查到的信息建立关系

5. DELETE：实例如右下图

 6. SET：设定值，可以修改属性（下图）

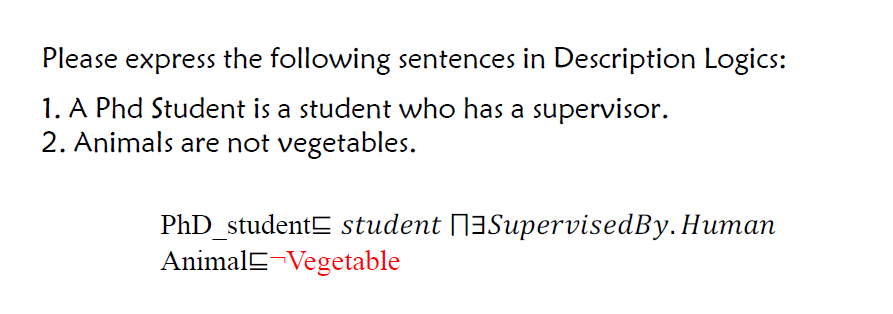
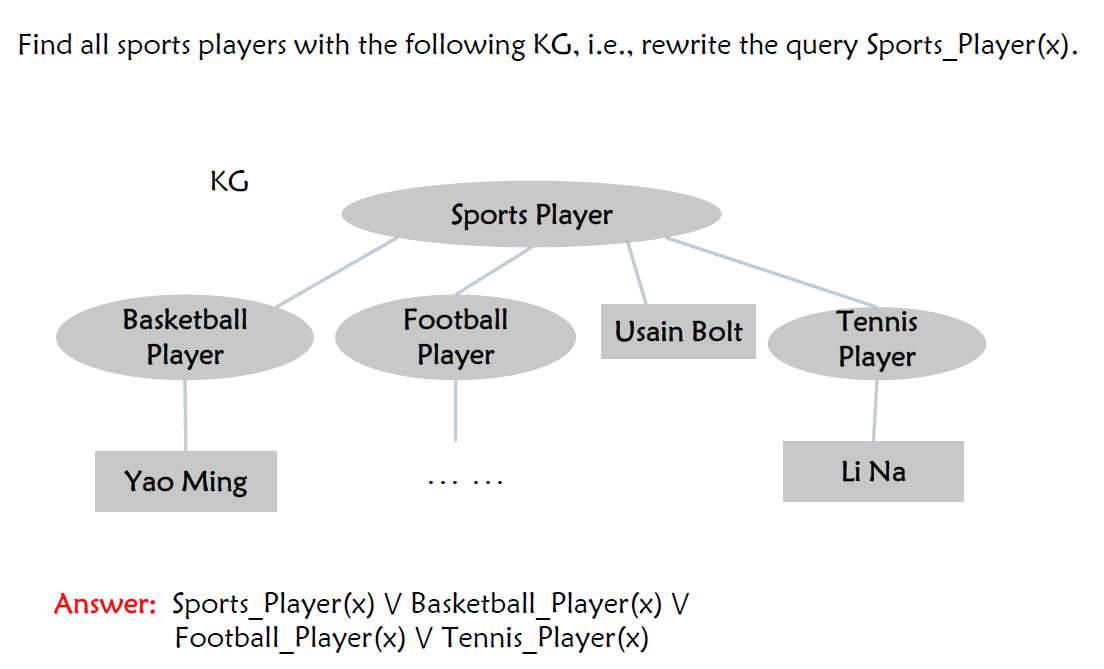


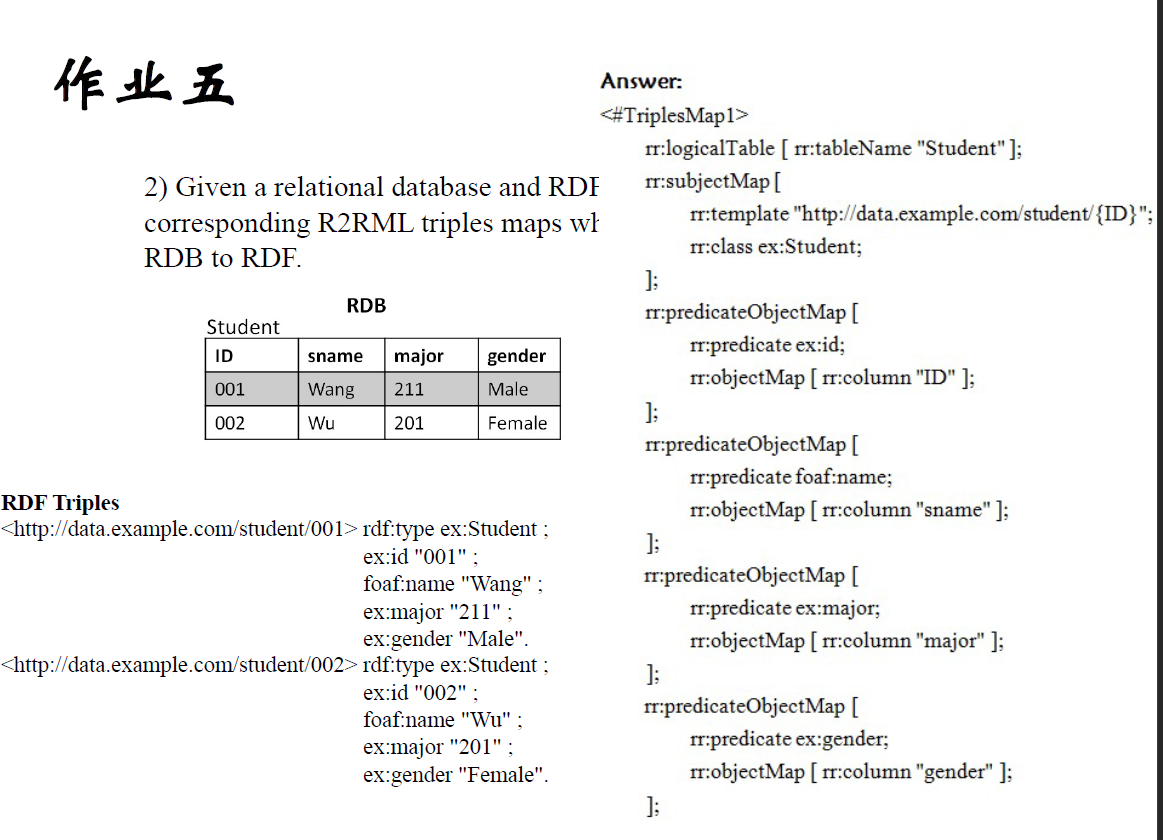
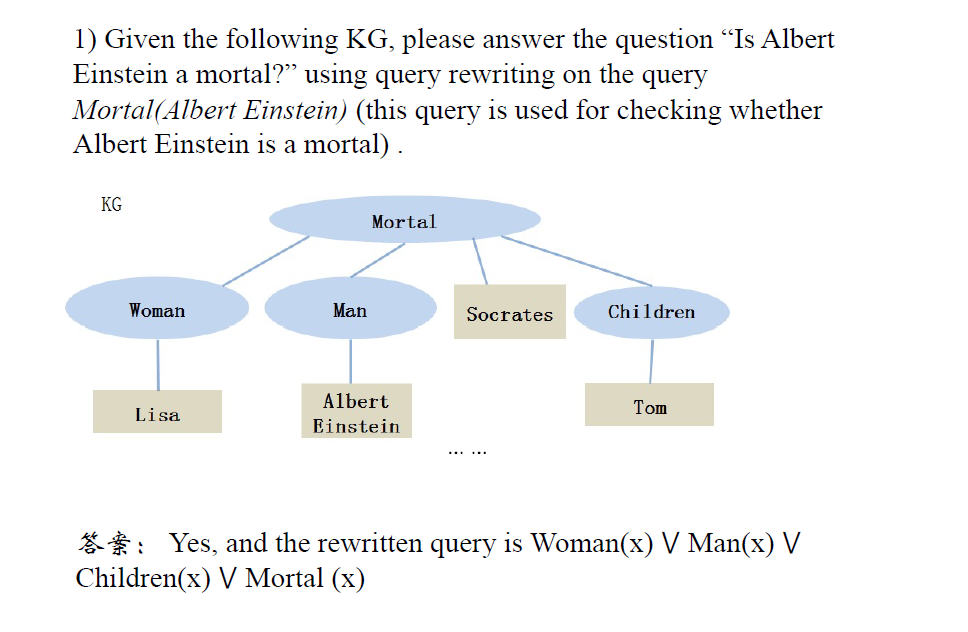
**· 后续补充**

1. rdf:type关系与is-a关系的区别：前者明确了实例属于该类，后者更偏重“属于”含义，包括实例类别信息或者子类包含信息。

2. 若未注明前缀则自行注明

**· 作业例题**

****

****