

Bases de Dados

T28 - Dos Dados ao Conhecimento

Prof. Daniel Faria

Sumário

- Recapitulação Breve
- Modelo de Dados em Grafo
- A Web Semântica
- Modelação em OWL
- Modelar Grafos em SQL

Recapitulação Breve

Bases de Dados de Grafos

- Dados representados como nós , arestas (representando relações) e propriedades ou atributos
- **Labeled-property graph**
 - Conjunto de nós, relações, propriedades e rótulos (nomes). Quer os nós quer as suas relações são rotulados e podem ter propriedades representadas por pares chave-valor
- **RDF graph / Knowledge Graph**
 - Conjunto de nós e arcos sob o modelo de dados RDF

Resource Description Framework



- Padrão do W3C originalmente desenhado para modelar metadados
- Adotado como modelo de dados para descrição e partilha de dados em grafo
- Grafo dirigido composto por triplos, i.e. afirmações do tipo:
 - `sujeito predicado objeto`
- Originalmente serializado em XML, mas há serializações mais legíveis por humanos (JSON, Turtle)
- Modelo de dados simples e flexível mas com bastante poder expressivo
- A base de linguagens da web semântica como RDFS e OWL que adicionam uma camada semântica ao modelo de dados RDF

Triplestores

- Dados armazenados e consultados sob a forma de triplos RDF
- Consultas semânticas com o uso de linguagens especializadas (e.g. SPARQL)
- Visão centrada nos arcos do grafo RDF
- Dados podem ser vistos como linhas numa tabela de 3 colunas (sujeito, predicado, objeto)
- Bases de dados de grafos propriamente ditas têm funcionalidades adicionais tais como adjacência sem índice (i.e. os nós têm ponteiros diretos para os seus vizinhos)

Graph Query Languages

SQL (Relational Database)

```
SELECT p2.name FROM people p1 JOIN friend USING (person_id) JOIN people p2
ON (p2.person_id = friend_id) WHERE p1.name = 'Jack' ;
```

Cypher (Property Graph)

```
MATCH (p1:person {name: 'Jack'}) -[:FRIEND_WITH] - (p2:person)
RETURN p2.name
```

SPARQL (RDF Graph)

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name
WHERE { ?s foaf:name "Jack" ;
          foaf:knows ?o .
          ?o foaf:name ?name . }
```

Modelo de Dados em Grafo

Modelo de Dados em Grafo

- Um grafo é um modelo de dados **muito genérico**
- Um diagrama EA pode ser representado diretamente como um grafo:
 - Entidades são nós
 - Associações binárias são arestas
 - Atributos são propriedades dos nós ou arestas
 - Relações ternárias ou agregadas são nós ligados por arestas a cada uma das entidades envolvidas na associação

Modelo de Dados em Grafo

- Um grafo é um modelo de dados **muito genérico**
- Dados relacionais podem ser representado diretamente em RDF:

Customer		
Name	Phone	Address
Daniel	555 12 34	Some place
Rita	555 67 89	Another place



```
Daniel    rdf:type    Customer
Daniel    Name        "Daniel"^^xsd:string
Daniel    Phone       "555 12 34"^^xsd:string
Daniel    Address     "Some place"^^xsd:string
Rita      rdf:type    Customer
Rita      Name        "Rita"^^xsd:string
Rita      Phone       "555 67 89"^^xsd:string
Rita      Address     "Another place"^^xsd:string
```

Modelo de Dados em Grafo

- Também podemos representar grafos no modelo relacional:
 - Labelled-Property Graph:
 - nó (ID, nome, propriedade1, ...)
 - aresta (ID1, ID2, nome, propriedadeA, ...)
 - RDF Graph:
 - triple (subject, predicate, object)

Grafos vs. Relações

- **Relações:**

- Estrutura de dados rígida
- Mais eficiente para consultas sobre um tipo de objeto (tabela)
- Menos eficiente interligar dados (joins) com seletividade elevada
- Menos eficiente para classificação (organização hierárquica de dados)

- **Grafos:**

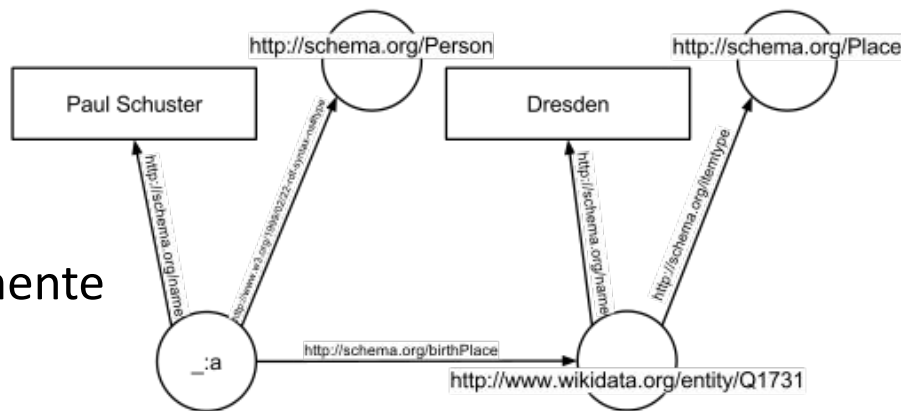
- Estrutura de dados flexível
- Mais eficiente para consultas sobre um objeto (nó)
- Menos eficiente interligar dados com seletividade baixa

A Web Semântica

A Web Semântica

- **Objetivo:**

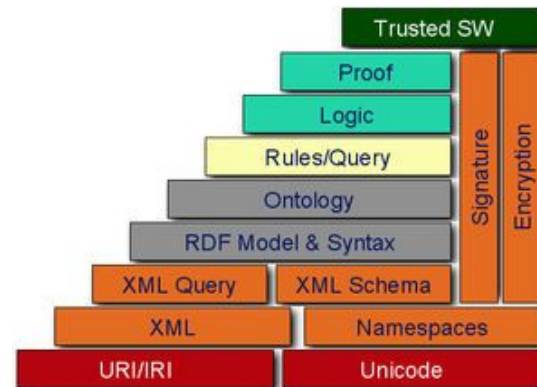
- Tornar os dados na Internet interpretáveis por máquinas
 - Representando explicitamente o seu significado
- Em vez de (apenas) hiperligações entre documentos, temos (também) hiperligações entre termos e as suas definições
 - *Web of documents* → *web of data*



A Web Semântica

- **Tecnologias:**

- Resource Description Framework (**RDF**):
 - Intercâmbio de dados
- Web Ontology Language (**OWL**):
 - Representação de conhecimento
- Simple Knowledge Organization System (**SKOS**)
 - Definição de thesaurus e taxonomias
- **SPARQL** (SPARQL Protocol and RDF Query Language)
 - Consulta de dados RDF

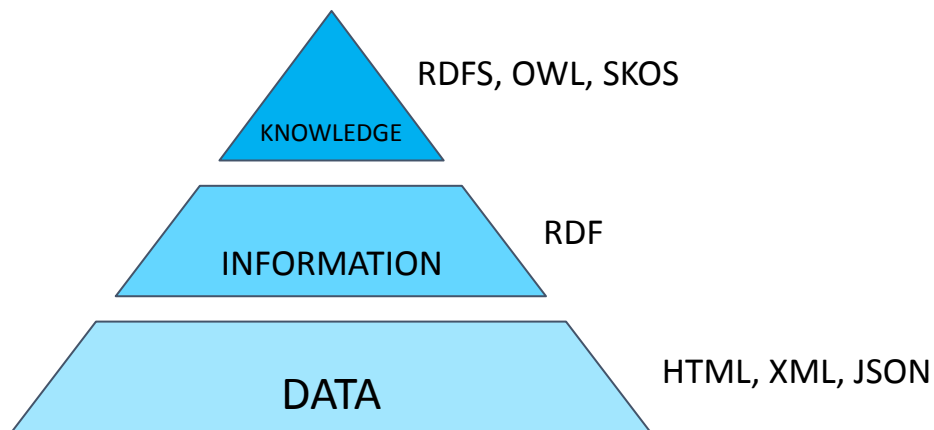


A Web Semântica



- **Linked Data:** elementos de dados (em páginas HTML ou em dados publicados na web) identificados e referenciados por Internationalized Resource Identifiers (IRIs)
- **Vocabulários:** providenciam o contexto semântico dos dados, por meio de classificação e lógica (OWL) ou de organização e terminologia (SKOS), tornando possível a sua interpretação automática
- **Inferência:** definições dos dados por meio de regras (OWL) permitindo raciocinar sobre eles e inferir conhecimento não explícito

A Web Semântica



Sintaxe RDF

- Podemos declarar triplos contendo:
 - Apenas elementos não controlados
 - Como predicado a propriedade `rdf:type`
 - Como objeto o tipo `rdf:Property` (i.e., declarando uma propriedade que pode ser usada em predicados)
 - Como objeto o tipo `rdf:List` (permitindo declarar recursivamente uma lista em triplos, com *first* e *rest*)
 - Como objeto o tipo `rdf:Statement` (para fazer reificação, i.e. atribuir um IRI a um triplo para que possa ser sujeito de outros triplos)

Sintaxe RDFS

- Podemos declarar triplos contendo:
 - Como objeto uma de várias categorias:
 - `rdfs:Resource`
 - `rdfs:Literal`
 - `rdfs:Class`
 - `rdfs:Datatype`
 - Como predicado `rdfs:subClassOf` ou `rdfs:subPropertyOf`
 - Como predicado `rdfs:domain` ou `rdf:range` de uma propriedade
 - Como predicado `rdfs:label`

Sintaxe OWL

- Separação de tipos de propriedades: *annotation*, *data* (atributos) e *object* (relações)
- Declaração explícita de indivíduos / instâncias
- Mais tipos de relações: equivalência, disjunção, união, interseção
- Restrições sobre propriedades: reflexividade, transitividade, etc
- Definições lógicas de classes (e.g. restrições existenciais, universais e de cardinalidade de propriedades)

Ontologia

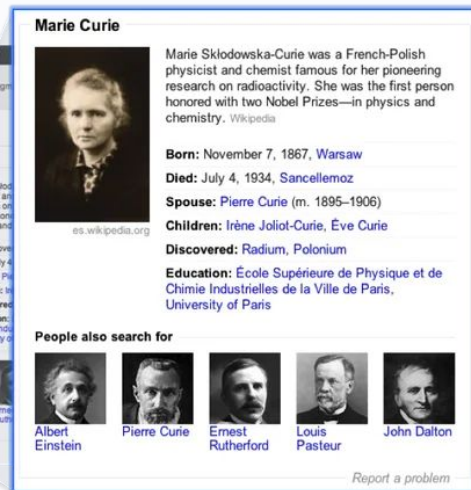
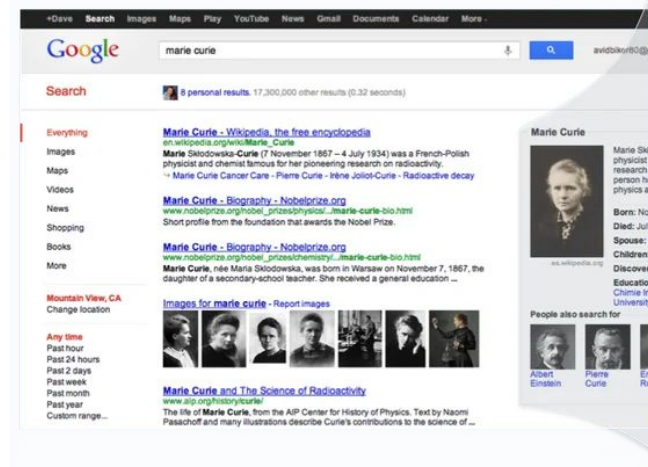
- **Definição:** Especificação da conceptualização de um domínio (i.e. modelo de conhecimento) tipicamente formalizada em OWL, que pode ou não incluir dados (indivíduos)
- **Aplicações:** Padronização de terminologia, classificação, integração semântica de dados (e.g. sistemas de informação geográfica), inferência
- **Exemplos:**
 - SNOMED-CT (medicina)
 - Gene Ontology (biologia)
 - FAO geopolitical ontology (geopolítica)

Grafo de Conhecimento

- **Definição:** Repositório de conhecimento que usa um modelo de dados em grafo (geralmente RDF) para integrar dados através de descrições interconectadas, frequentemente usando ontologias como esquemas
- **Aplicações:** Procura inteligente, mineração de dados
- **Exemplos:**
 - [Google Knowledge Graph](#)
 - WordNet
 - GeoNames
 - DBpedia

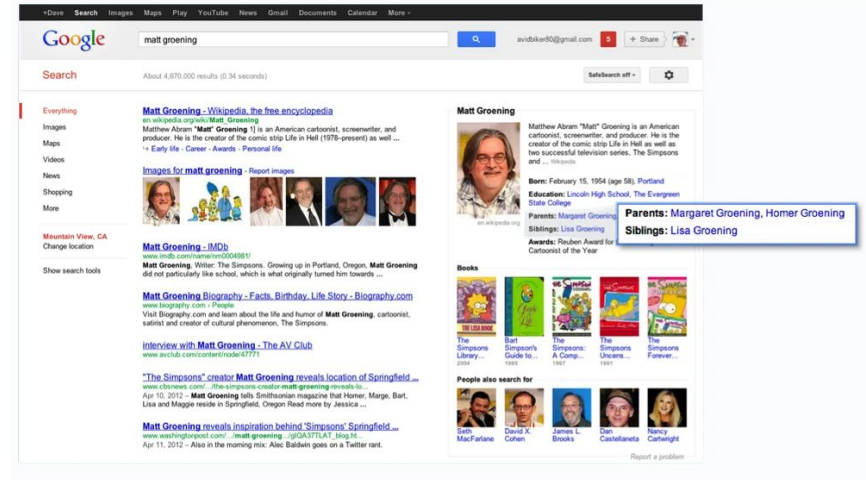
Procura Inteligente

- Integração de Informação
 - Procura por categoria mais genéricas engloba resultados de categorias mais específicas
 - Sumário de informação relevante

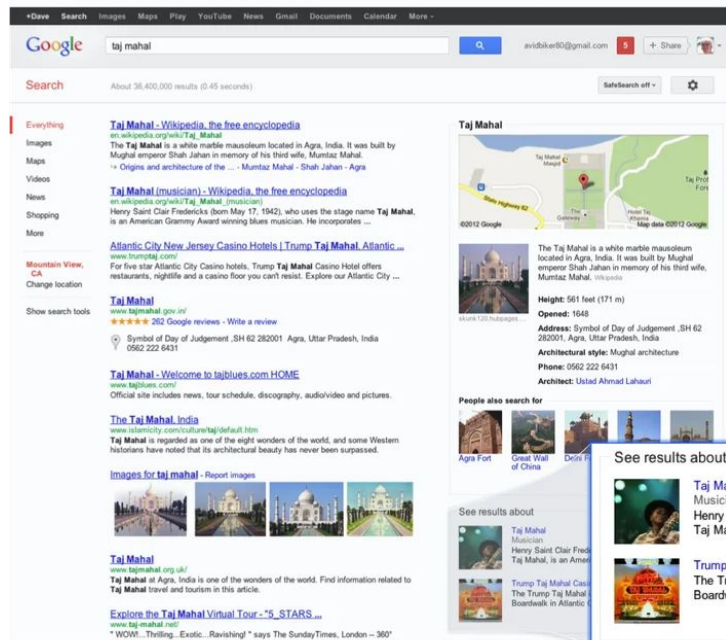


Procura Inteligente

- Linked Data & Related Searches
 - Permitem fazer *follow up* da busca num click, sem nunca abrir outra página web



Procura Inteligente



- Entity resolution:
 - Identificar e separar significados diferentes do mesmo termo de busca
 - Permitir ao utilizador filtrar por um dos significados

Procura Inteligente

- Schema.org: ontologia ligeira usada para classificar documentos na web
 - Define:
 - Classes (e.g. documento, pessoa)
 - Propriedades (e.g. título, assunto, nome, emprego)
 - Criadores de conteúdos web podem enriquecer semanticamente as suas páginas com markup schema.org embebido no HTML (e.g. com RDFa ou Microdata) ou os dados disponibilizados em JSON-LD
 - Motores de busca podem usar markup schema.org das páginas para as classificar e indexar

Extensões HTML

- **RDFa e Microdata**: especificações W3C definindo atributos para embeber metadados ricos em HTML (ou no caso RDFa também XHTML e XML)

RDFa

```
<div vocab="http://schema.org/" typeof="Movie">
  <h1 property="name">Avatar</h1>
  <div property="director" typeof="Person">
    Director: <span property="name">James Cameron</span>
    (born <time property="birthDate" datetime="1954-08-16">August 16, 1954</time>)
  </div>
  <span property="genre">Science fiction</span>
  <a href="../movies/avatar-theatrical-trailer.html" property="trailer">Trailer</a>
</div>
```

Microdata

```
<div itemscope itemtype="http://schema.org/Movie">
  <h1 itemprop="name">Avatar</h1>
  <div itemprop="director" itemscope itemtype="http://schema.org/Person">
    Director: <span itemprop="name">James Cameron</span>
    (born <time itemprop="birthDate" datetime="1954-08-16">August 16, 1954</time>)
  </div>
  <span itemprop="genre">Science fiction</span>
  <a href="../movies/avatar-theatrical-trailer.html" itemprop="trailer">Trailer</a>
</div>
```

Modelação em OWL

Exemplo: Converter para OWL

Animal → Class

name → DataProperty

name → FunctionalProperty

dob → DataProperty

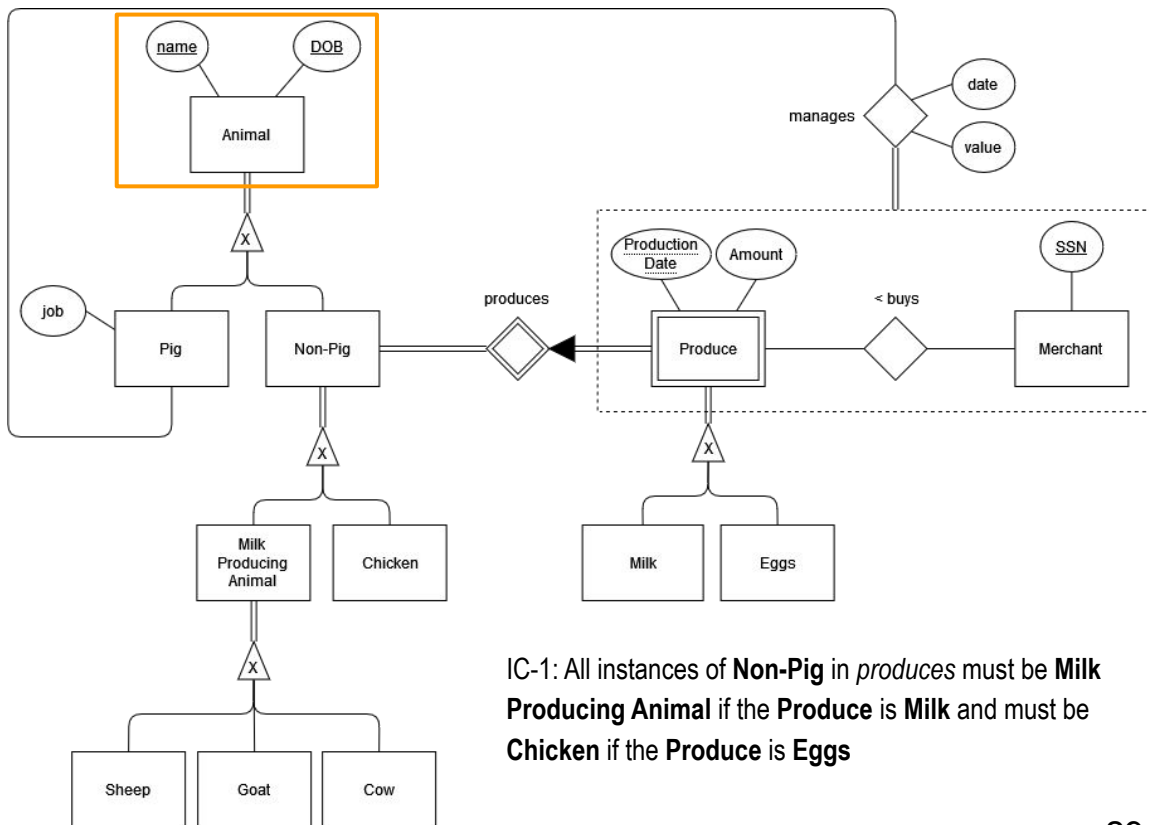
dob → FunctionalProperty

Animal → \exists name

Animal → \exists dob

Não existe o conceito de chave composta, mas chave simples pode ser declarada com:

InverseFunctionalProperty



Exemplo: Converter para OWL

Pig → Class

job → DataProperty

Pig → \exists job

NonPig → Class

Animal → Pig \cup NonPig

Disjoint → Pig, NonPig

MilkPrd → Class

Chicken → Class

NonPig → MilkPrd \cup Chicken

Disjoint → MilkPrd, Chicken

Sheep → Class

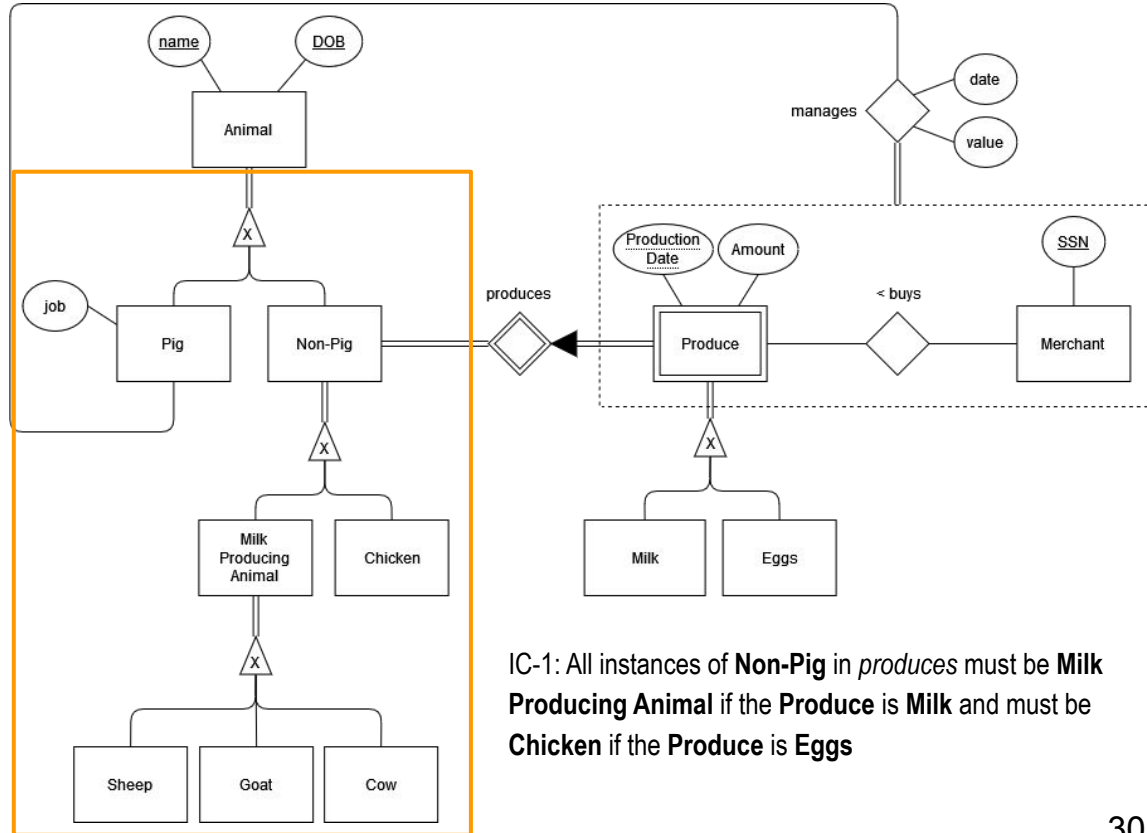
Goat → Class

Cow → Class

MilkPrd → Sheep \cup Goat \cup

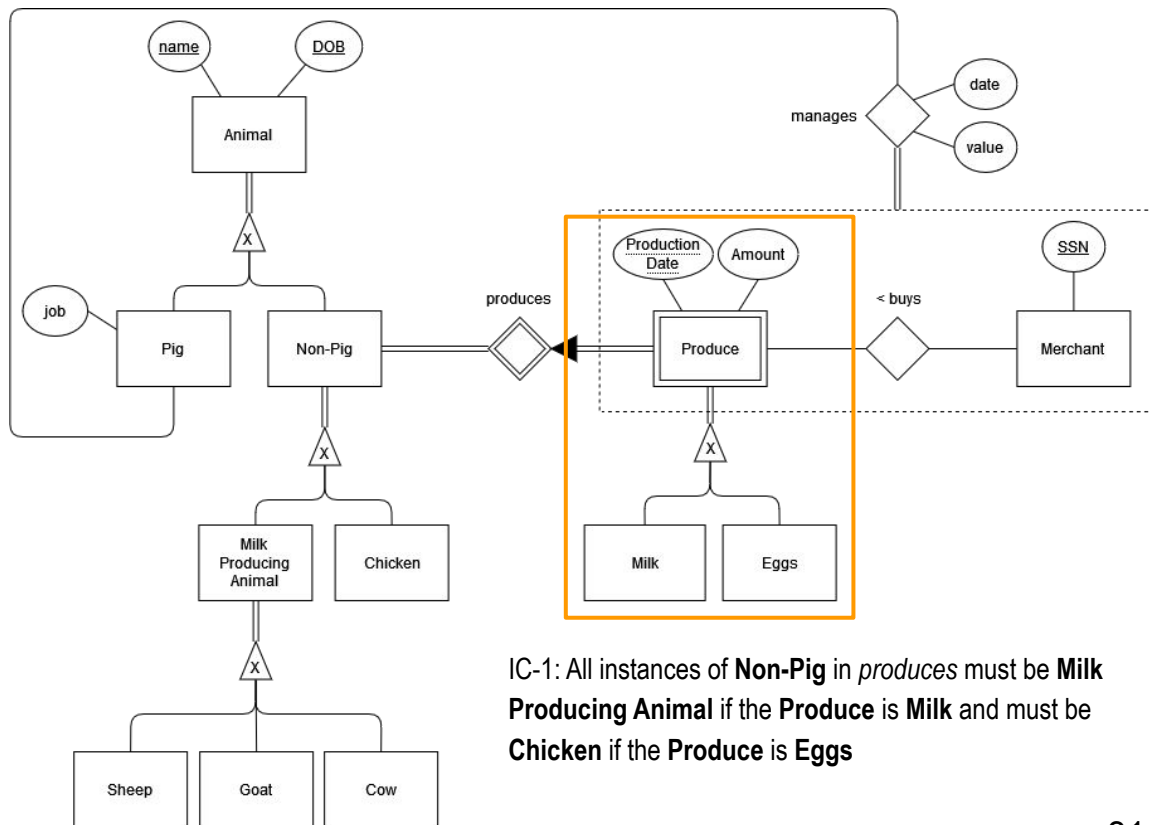
Cow

Disjoint → Sheep, Goat, Cow



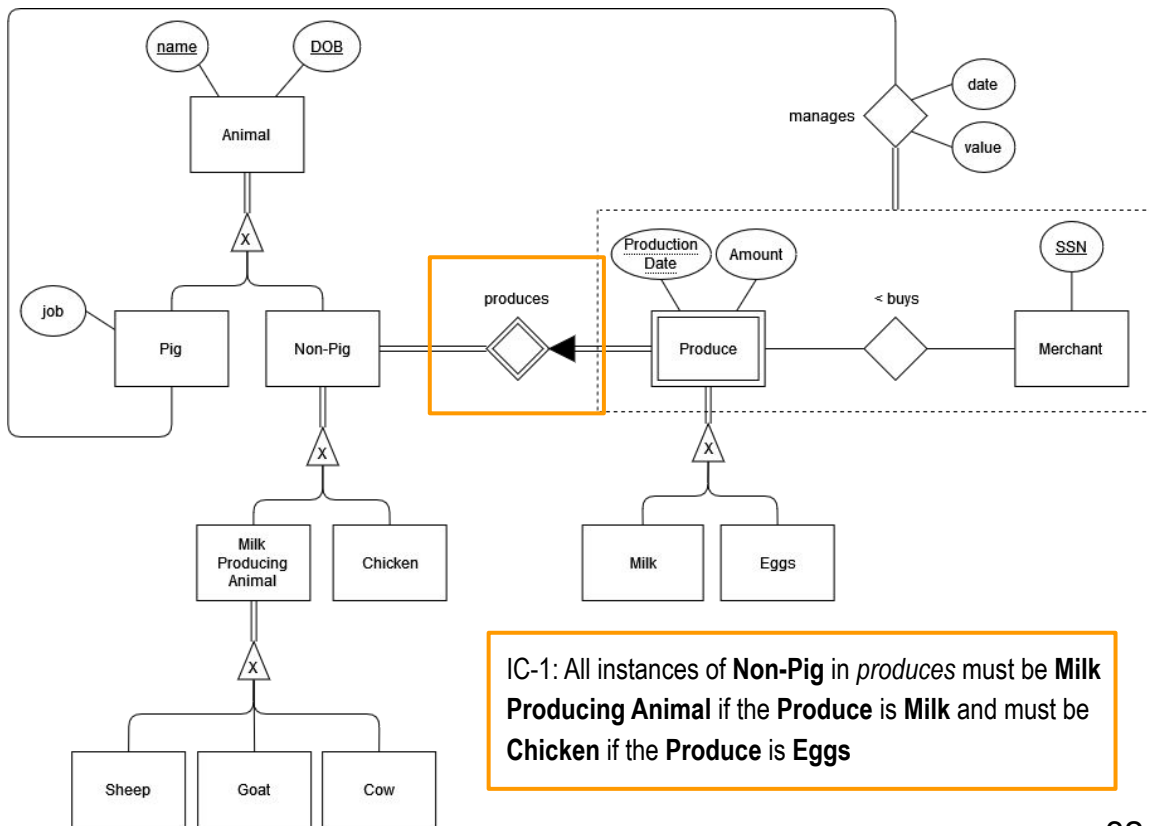
Exemplo: Converter para OWL

Produce → Class
prodDate → DataProperty
prodDate → FunctionalProperty
amount → DataProperty
amount → FunctionalProperty
Milk → Class
Eggs → Class
Produce → Milk \cup Eggs
Disjoint → Milk, Eggs



Exemplo: Converter para OWL

`produces` → `ObjectProperty`
`produces` →
`ReverseFunctionalProperty`
`produces` Domain `NonPig`
`produces` Range `Produce`
`NonPig` → \exists `produces`
`MilkProd` → \forall `produces` (`Milk`)
`Chicken` → \forall `produces` (`Eggs`)
`Produce` → \exists
`Inverse(produces)`



Exemplo: Converter para OWL

- **Dados:**

```
cow1 → Cow
cow1 name "gertrude"^^xsd:string
cow1 dob "2016-06-12"^^xsd:date
milk_912732 → Milk
cow1 produces milk_912732
milk_912732 prodDate "2022-07-21"^^xsd:date
```

Não definimos amount, mas OWL opera sob o princípio “open world”

Modelar Grafos em SQL

Grafos e Tabelas

- Maturidade de SQL oferece vantagens claras sobre graph databases no que toca a escrita de dados (transações)
- Modelo relacional vantajoso para consulta/análise em largura
 - Ideal para representar dados homogéneos com categorias planas
- Modelo em grafo vantajoso para consulta/análise em profundidade
 - Melhor para dados heterogéneos, ideal para classificação hierárquica

Grafos em SQL

- Modelação direta de RDF:
 - Mega tabela única: *triple (subject, predicate, object)*
 - Ineficiente para consultas em profundidade (requerem N auto-joins da tabela)
 - Não distingue esquema (i.e., definições de classes e propriedades) dos dados (i.e., declarações de indivíduos)
 - Não distingue tipos de dados

Grafos em SQL

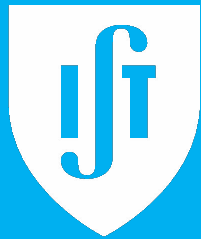
- Modelação tradicional:
 - *nó* (*nID*, *nome*)
 - *aresta* (*aID*, *nID1*, *nID2*, *nome*)
 - *propriedade* (*pID*, *prop*, *valor*), $pID \in \{nID, aID\}$
- Também ineficiente para consultas profundas
- Não distingue nós do esquema (classes) de nós dos dados (indivíduos)

Grafos em SQL

- Como otimizar consulta profunda?
 - Fecho transitivo do grafo: ligar cada nó a cada outro nó que ele vê, transitivamente
 - *graph_path(ID1, ID2, relation, distance)*
 - Fácil de implementar em SQL com algoritmo semi-naive
 - Começar com tabela de edges e $d=0$
 - Enquanto houver expansão, incrementar d e fazer auto-join com $g1.ID2 = g2.ID1$, $g1.distance = 1$ e $g2.distance = d$

Grafos em SQL

- Modelação híbrida:
 - Dados modelados sob modelo relacional (requer dados relativamente homogéneos e/ou poucas categorias diferentes)
 - Classes/categorias dos dados modeladas sob uma única tabela
 - Hierarquia de classes modelada sob tabela de relações com fecho transitivo



TÉCNICO LISBOA