CI242 – Tópicos de Pesquisa em Informática

Gerenciamento de Dados em Larga Escala

Carmem Hara

Tópicos

- Gerenciamento de dados na nuvem
- Dados ligados e web semântica

Computação na Nuvem

- Por que?
  - A WEB está substituindo o desktop

    Google Gmail, Google Docs, Amazon, Flickr, Facebook, Twitter, YouTube
- Mudança de Paradigma:
  - Amazon Web Services
  - Windows Azure Platform
  - Google App Engine

Computação na Nuvem

É um modelo que proporciona acesso, através da rede, a um conjunto de recursos configuráveis (rede, servidores, armazenamento, aplicações e serviços), que são gerenciados pelo provedor do recurso, e que podem ser utilizados por clientes através de uma interface de serviço.

Computação na Nuvem

- Evolução dos conceitos de:
  - Virtualização: encapsulamento de características físicas do recurso e visão de multiplos recursos logicos sobre um mesmo recurso físico
  - Arquitetura orientada a serviço (SOA): baixo acoplamento entre o provedor e consumidor
  - Computação autônoma: auto-gerenciável, self-service e sob demanda
  - Computação como serviço público

Computação na Nuvem

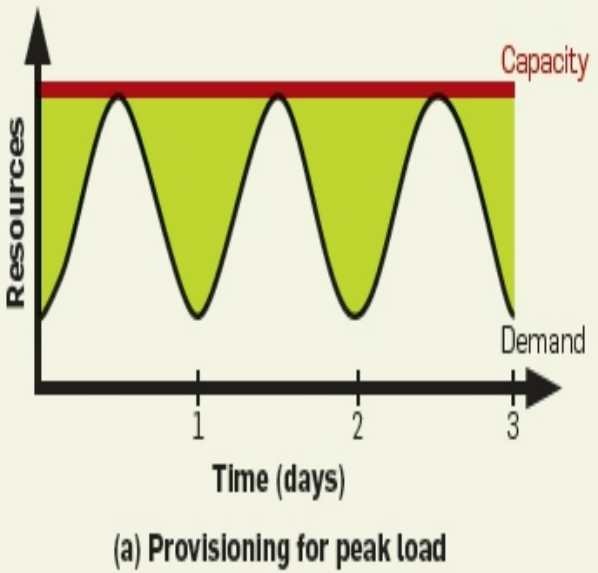
- Compartilhamento de recursos: CPU, armazenamento, banda de rede
- Disponibilidade, escalabilidade, elasticidade
- Gerenciamento, transferência de riscos
- Tolerância a falhas
- Baseado em computadores simples
- Pagamento pelo uso

Elasticidade

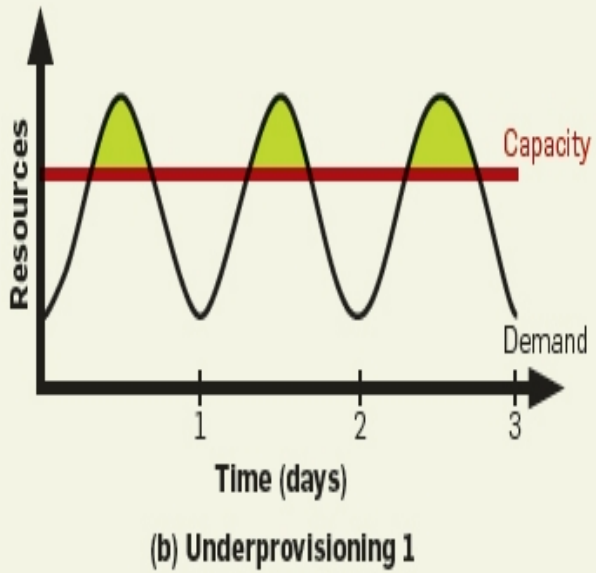
Data Center da Microsoft

- 1000 computadores usados por 1 hora custa o mesmo que 1 computador usado por 1000 horas
- Com paralelismo: resultados 1000 vezes mais rápido
- Exemplo: Animoto – carga de trabalho dobrou a cada 12 horas por 3 dias.

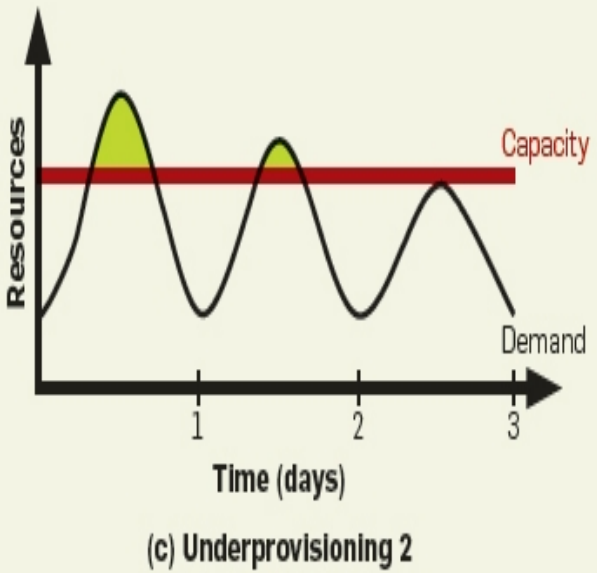
Utilização de Recursos



(a) Provisioning for peak load



(b) Underprovisioning 1



(c) Underprovisioning 2

Fonte: M. Armbrust et al, CACM'2010

- Provisionamento pela carga máxima pode causar subutilização.
- Provisionamento pela média pode aumentar o tempo de espera e perda de clientes.
- Dificuldade de prever a carga e variaçãao da carga no tempo.

Modelos de Operação

- Nuvem Pública

  computação como serviço público

- Nuvem Comunitária

  compartilhamento de recursos entre membros de uma comunidade com interesses comuns

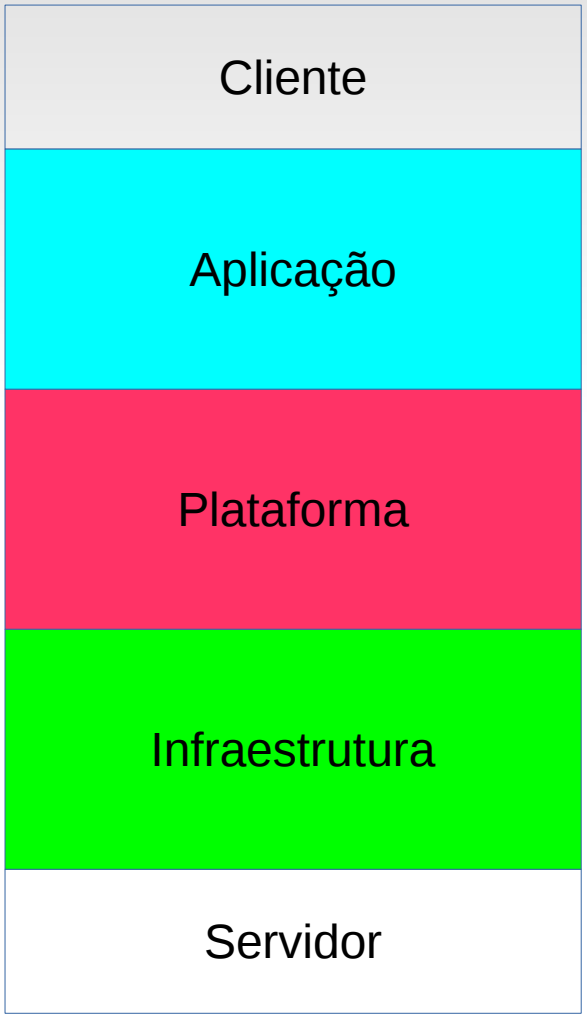
- Nuvem Privada

  virtualização de serviços em servidores locais

- Nuvem Híbrida

  combinação de abordagens

Paradigmas



SaaS (Software as a Service): oferece software como serviço, eliminando a necessidade de instalação e execução na máquina do cliente.

PaaS (Platform as a Service): oferece um conjunto de soluções como serviço para dar suporte a aplicações na nuvem. Ex: Google App Engine, MS Azure

IaaS (Infrastructure as a Service): oferece uma plataforma computacional, em geral um ambiente virtualizado, como serviço. Ex: Amazon EC2

Banco de Dados como SaaS

- Um serviço de armazenamento e busca de dados na Internet com:
  - Ilusão de recursos infinitos: *escalabilidade*
  - Custo mínimo de instalação
  - Pagamento pela utilização do serviço (volume de dados e de acessos)
  - Disponibilidade

Carga de Trabalho

- OLAP *(Online Analytical Processing)*
  - Análise de um grande volume de dados
  - Poucas atualizações
  - Processamento pesado

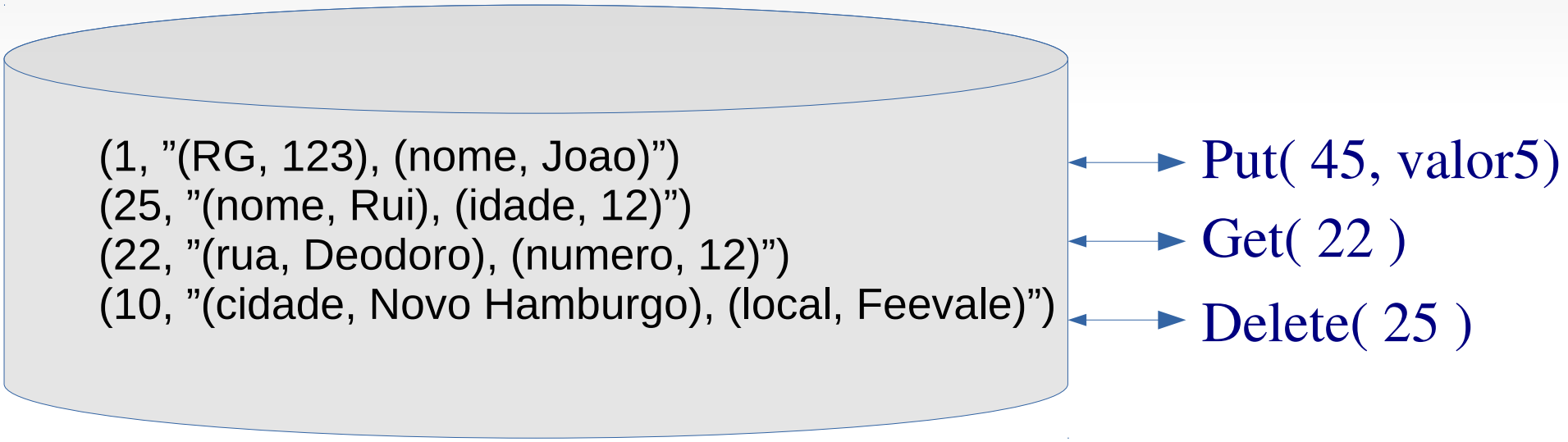
  - BD Paralelos
  - MapReduce - Hadoop

- OLTP (*Online Transaction Processing)*
  - Transações curtas que envolvem poucos dados
  - Grande volume de atualizações

  - BD Distribuídos
  - Repositórios chave-valor

Modelo Chave-Valor

A chave identifica um par e o valor associado é um BLOB.



Exemplo: Voldemort, Scalaris

Nível Lógico - Modelos

- Relacional
  - Interface: SQL
  - Ex: Amazon Relational Database Service (RDS)
- XML
  - Interface: Xquery
  - Ex: Sausalito

Movimento NoSQL

- NoSQL: ”Not Only SQL” ou ”Not Relational” ?
- Armazenamento distribuído e escalável
- Replicação de dados → tolerância a falhas
- Sem esquema ou com esquema extensível
- Interface simples, baseada em chamadas de operações simples
- Consistência fraca

Desafios

- Segurança e privacidade dos dados
  - Modelo
  - Nível em que deve ser implementado
  - Criptografia, políticas de segurança
- Modelos de dados distintos para aplicações distintas?
  - independência física e lógica
  - Qual o modelo / arquitetura de compartilhamento?
  - OLAP / OLTP

Desafios (cont.)

- Mapeamento entre o modelo lógico e físico
  - Particionamento e Localização dos dados
- Processamento de consultas e métodos de acesso (Indexação, otimização e tuning)
- Modelo de suporte a transações
  - Pelo repositório chave-valor
  - Componente do sistema gerenciador de banco de dados

Desafios (cont.)

- SGBDs multi-inquilinos
  - Grande quantidade de inquilinos
  - Carga variável ao longo do tempo – balancealmento de carga
  - Migração dinâmica de dados

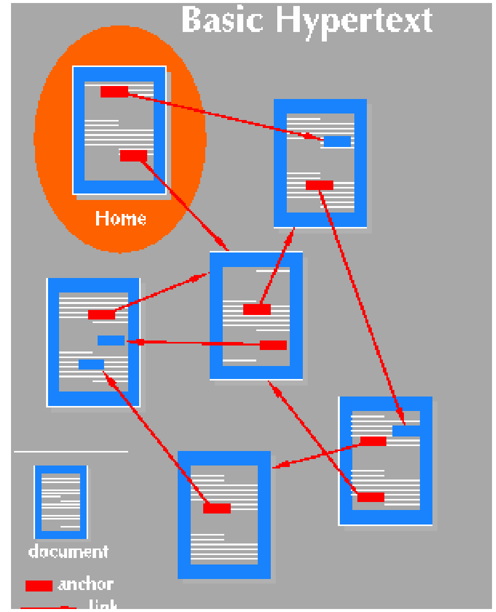
Gerenciamento de Dados na Nuvem

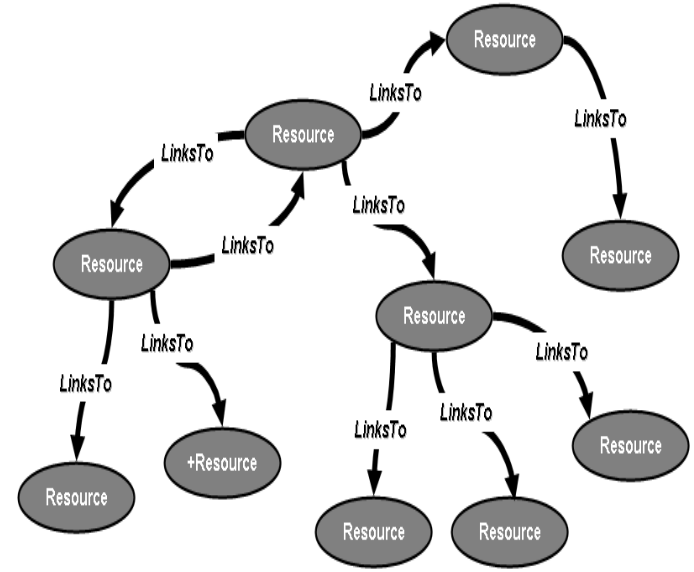
Um novo modelo de armazenamento de dados que apresenta diversos desafios para prover:

- Escalabilidade
- Elasticidade
- Consistência
- Facilidade de desenvolvimento de aplicações

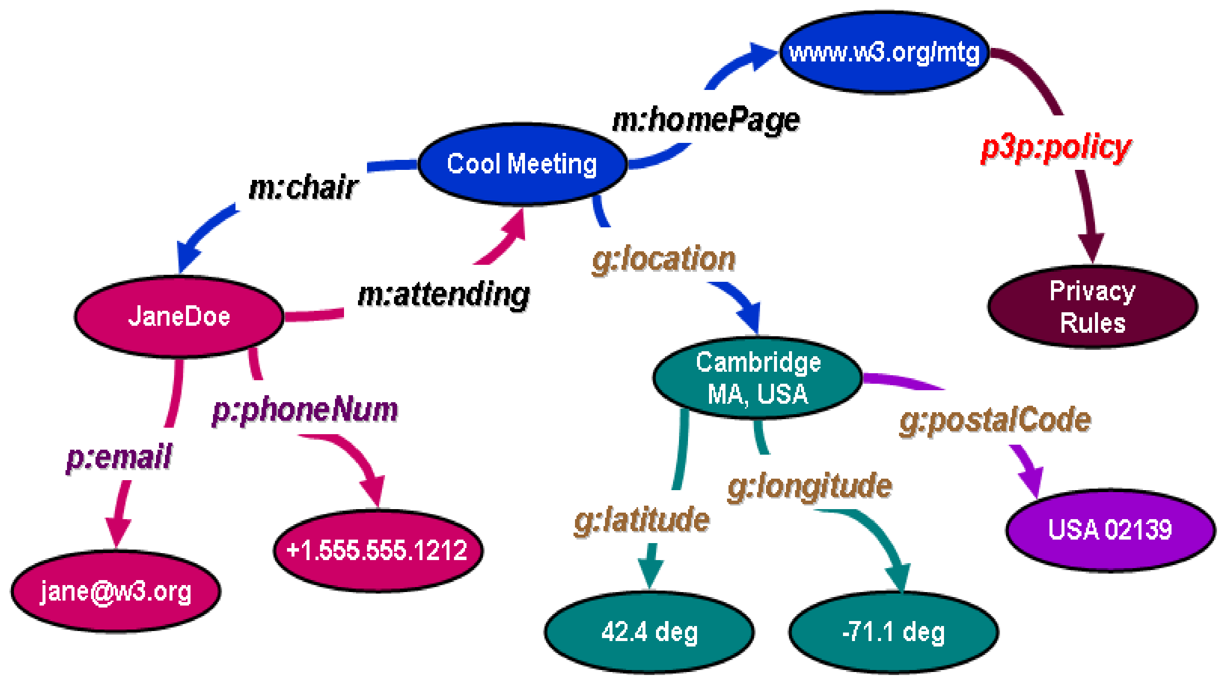
Dados Ligados e Web Semântica

Web of Documents



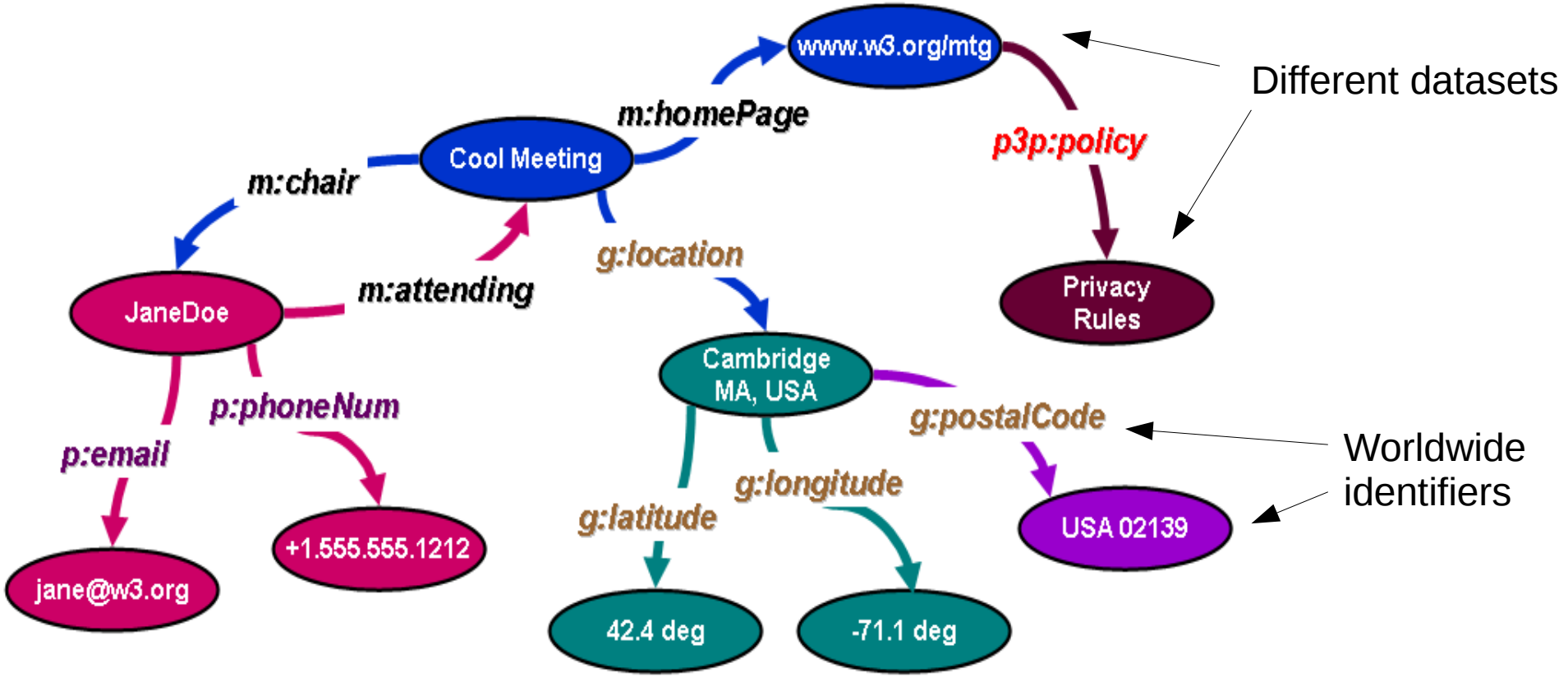


Web of Data



Web of Data = Semantic Web

Available data are easier for *machines* to: find, access and process



Linked Data

- Set of best practices for publishing and interlinking structured data on the Web
- Principles:
  - Use URIs as names for things
  - Use HTTP URIs to make them dereferenceable
  - Provide useful information using the standards (RDF, SPARQL).
  - Include links to other URIs to promote further discovery of knowledge
- Linked Data are provided in *RDF*

Large RDF Data Sets

Bio2RDF: bio and gene related data

- 10+ billion triples

DBPedia: data extracted from Wikipedia

- 3 billion triples

LOD: integration of various open data sources

- 31+billion triples

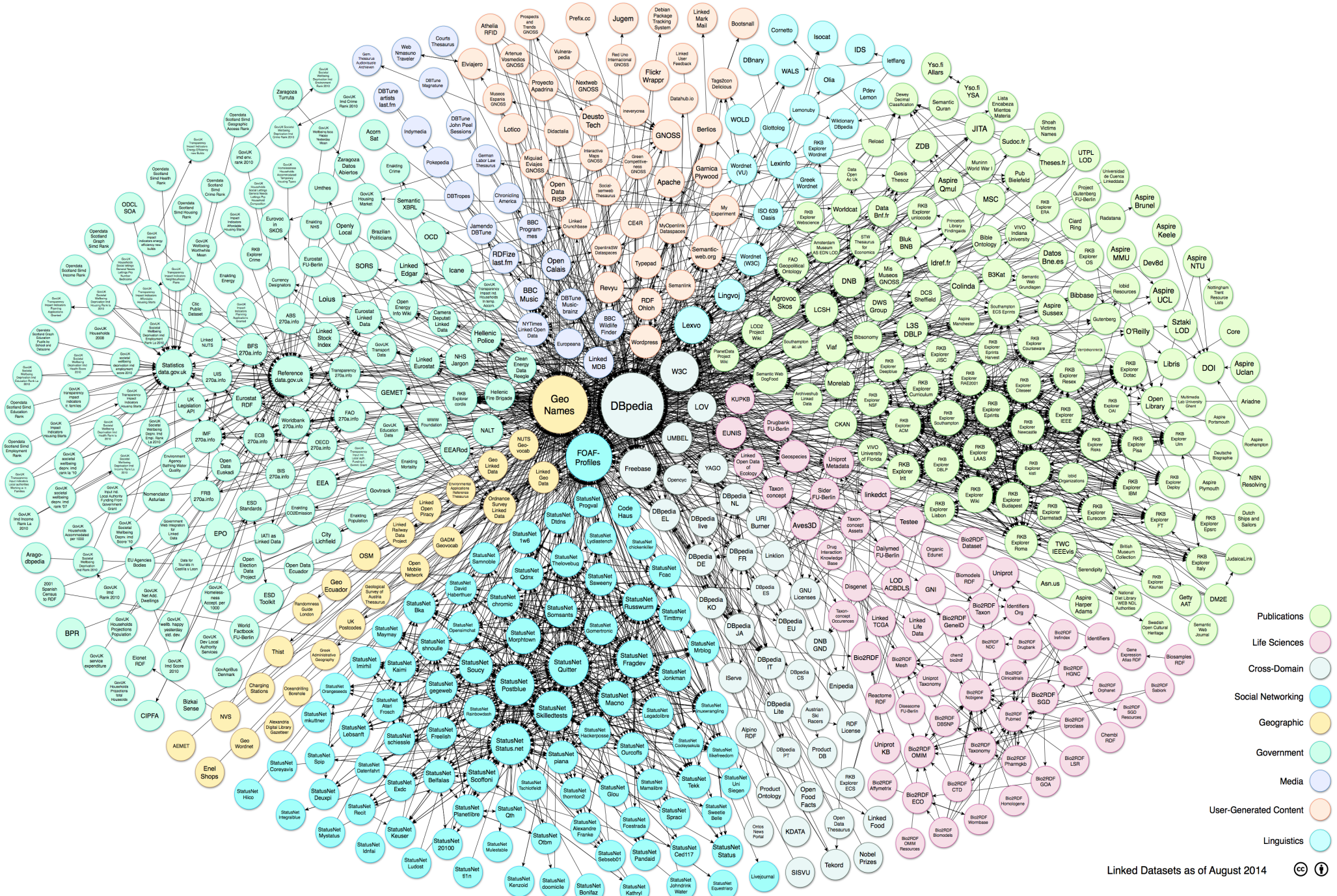
Data.gov: US open government data

- 6+ billion triples

LinkedGeoData: data from the OpenStreet

- 20 billion triples

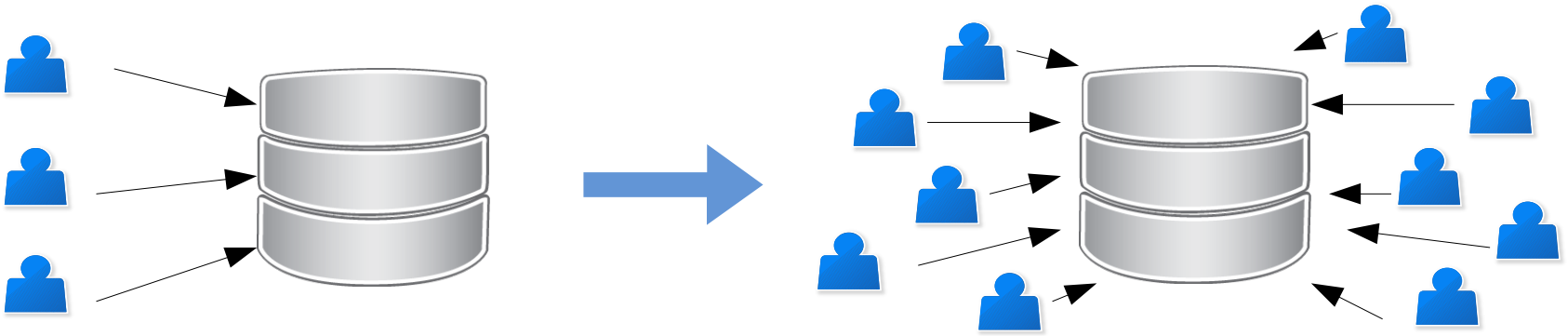
And it is getting bigger...



We need to provide scalability wrt :

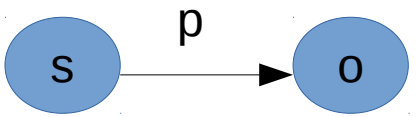
- Volume of data

- Volume of data requests (queries)



RDF (Resource Description Framework)

- Set of triples (s, p, o)

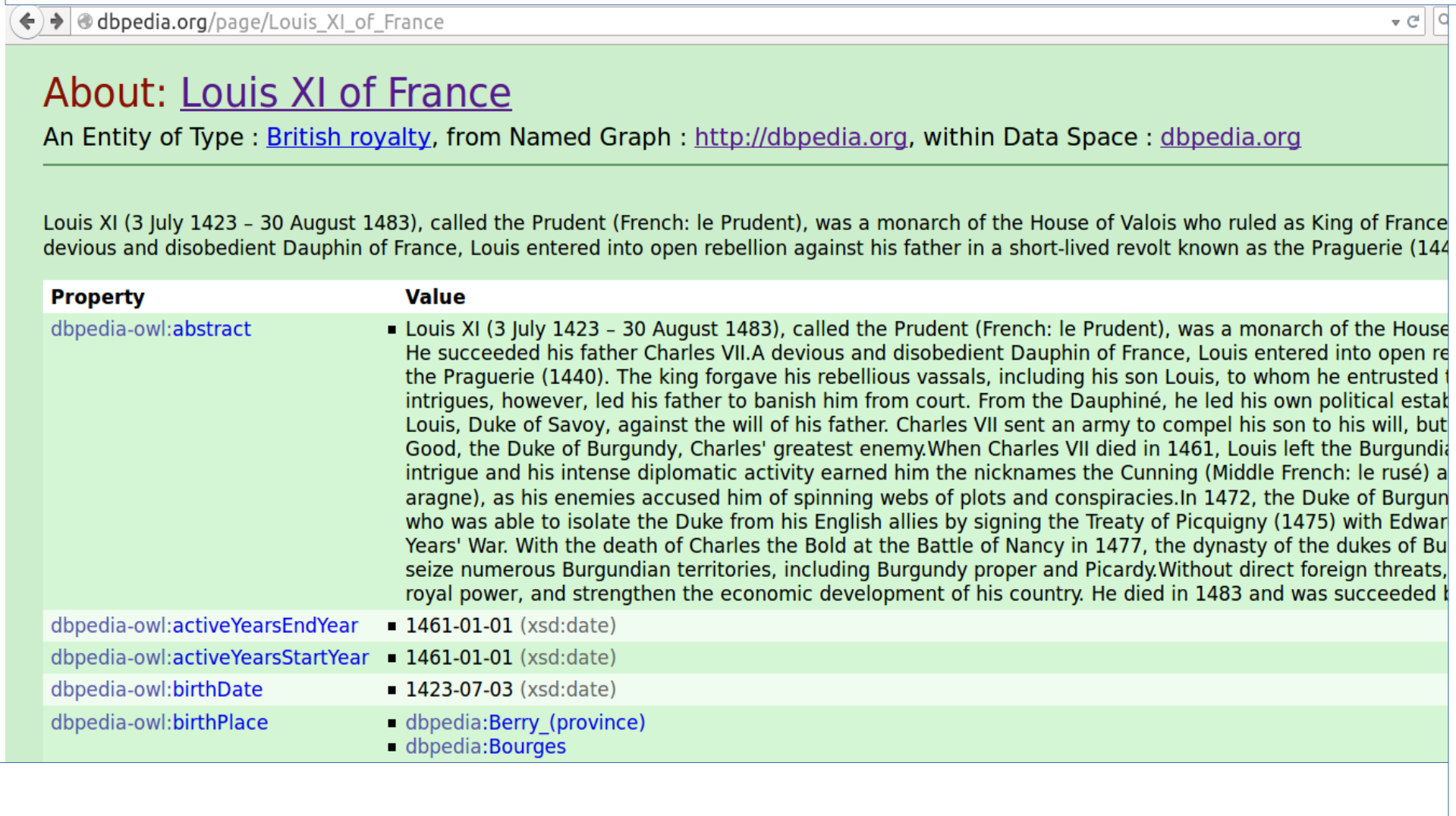


- Subject
  - IRI
  - Blank node
- Property/ Predicate
  - IRI
- Object
  - IRI
  - Blank node
  - Literal

- IRI: Internationalized Resource Identifier
  (generalization of URI)

Example of an IRI

http://dbpedia.org/resource/Louis_XI_of_France



RDF as a graph

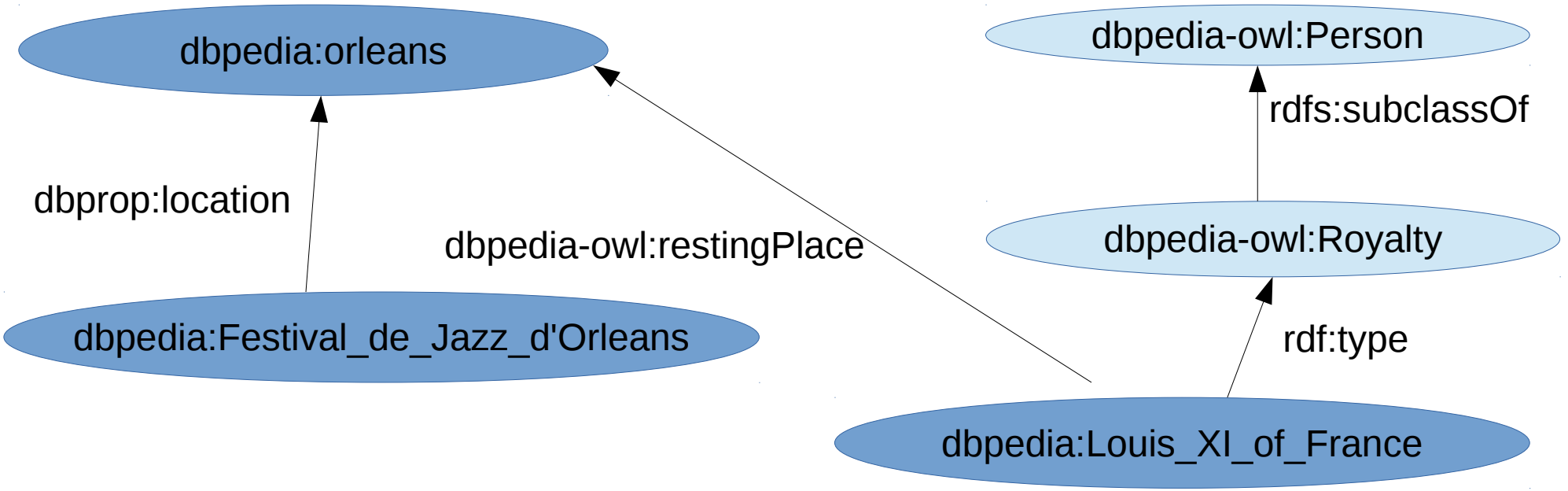
dbpedia: http://dbpedia.org/resource/

dbprop: http://dbpedia.org/property/

rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#

dbpedia-owl: http://dbpedia.org/ontology/

rdfs: http://www.w3.org/2000/01/rdf-schema#



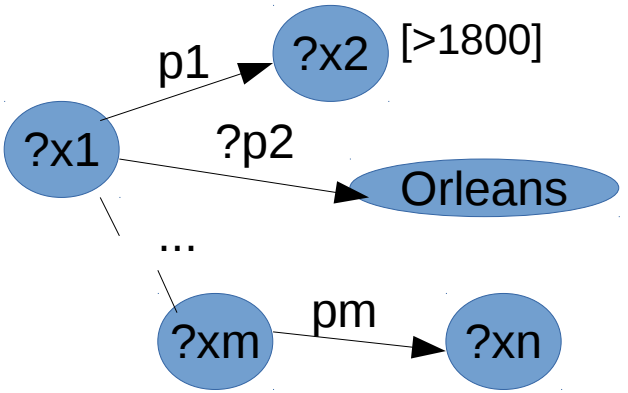
SPARQL Query Language

Problem: find a pattern in a graph

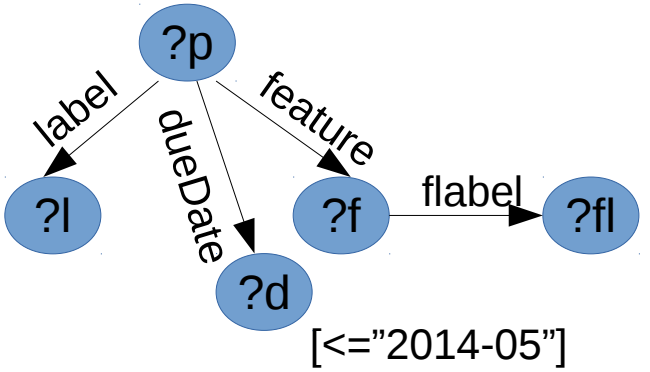
```
select ?x1, …, ?xn
where {?x1 p1 ?x2.
       ?x1 ?p2 Orleans.
       …
       ?xm pm ?xn .
filter ?x2 > 1800 }
```

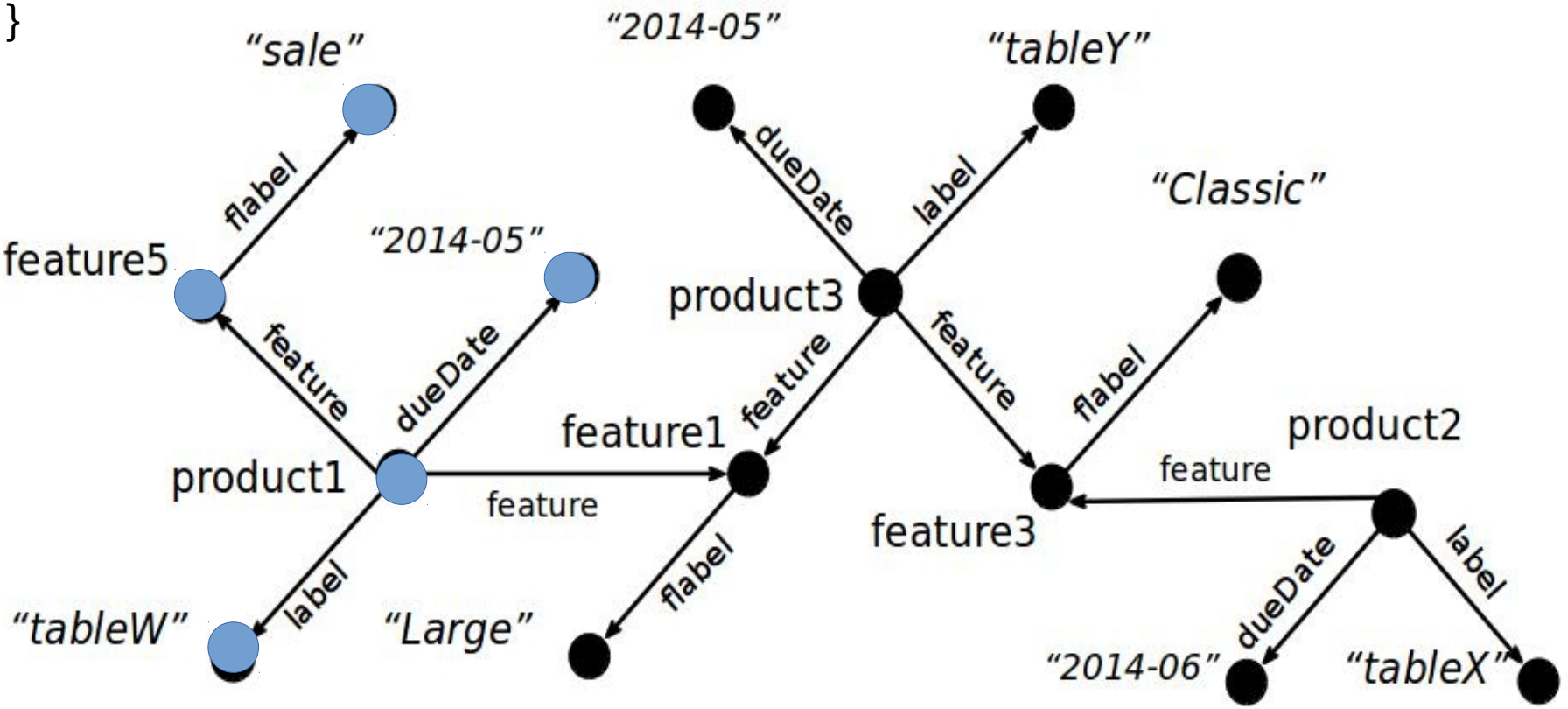
Result

Pattern

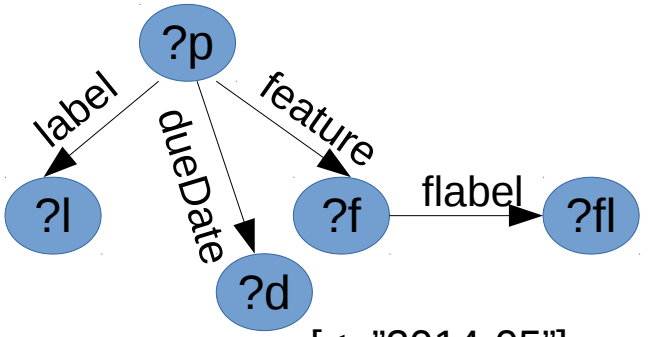


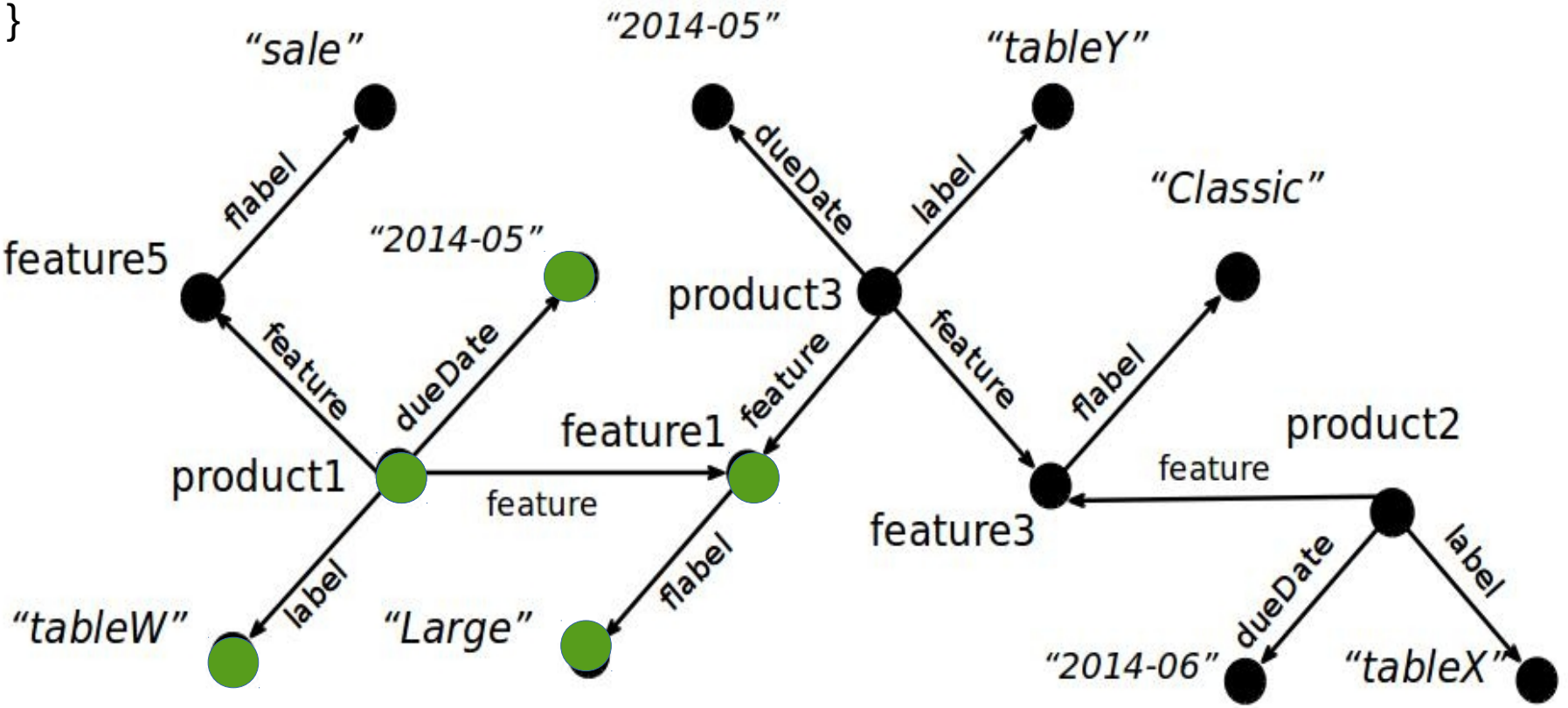
```
select ?l, ?fl
where {
   ?p label ?l .
   ?p dueDate ?d .
   ?p feature ?f .
   ?f flabel ?fl
  filter (?d <= “2014-05”)
}
```



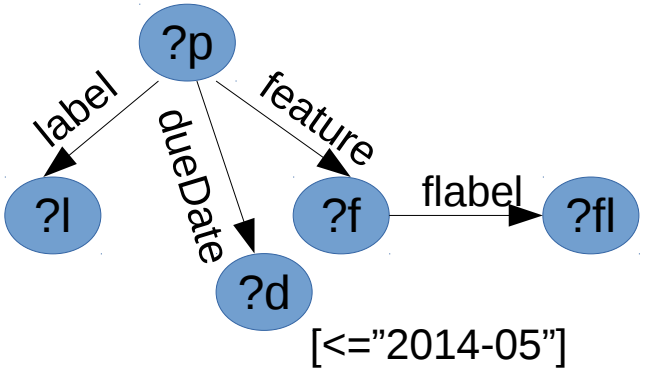


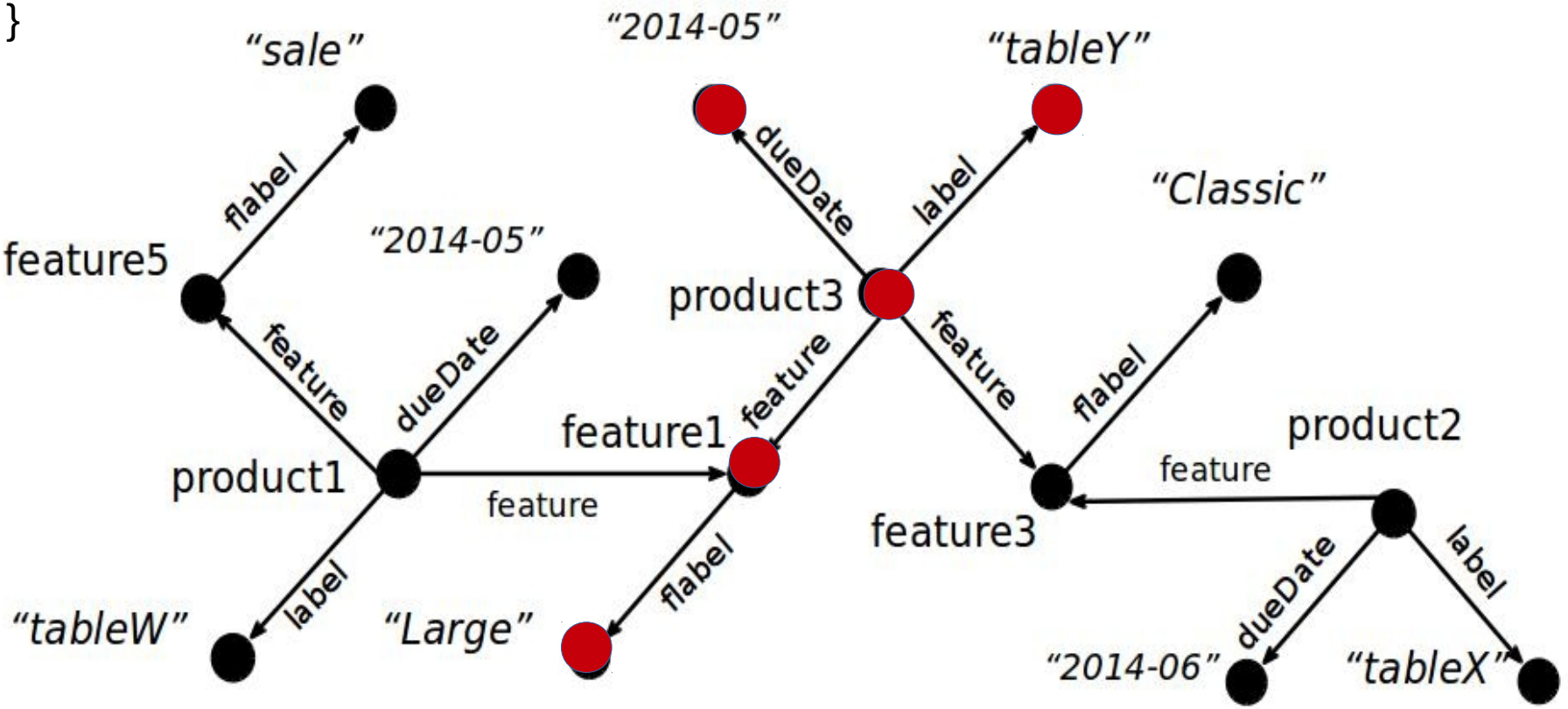
```
select ?l, ?fl
where {
   ?p label ?l .
   ?p dueDate ?d .
   ?p feature ?f .
   ?f flabel ?fl
  filter (?d = “2014-05”)
}
```



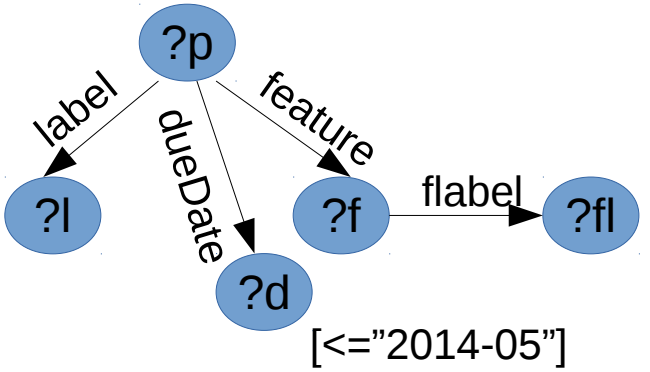


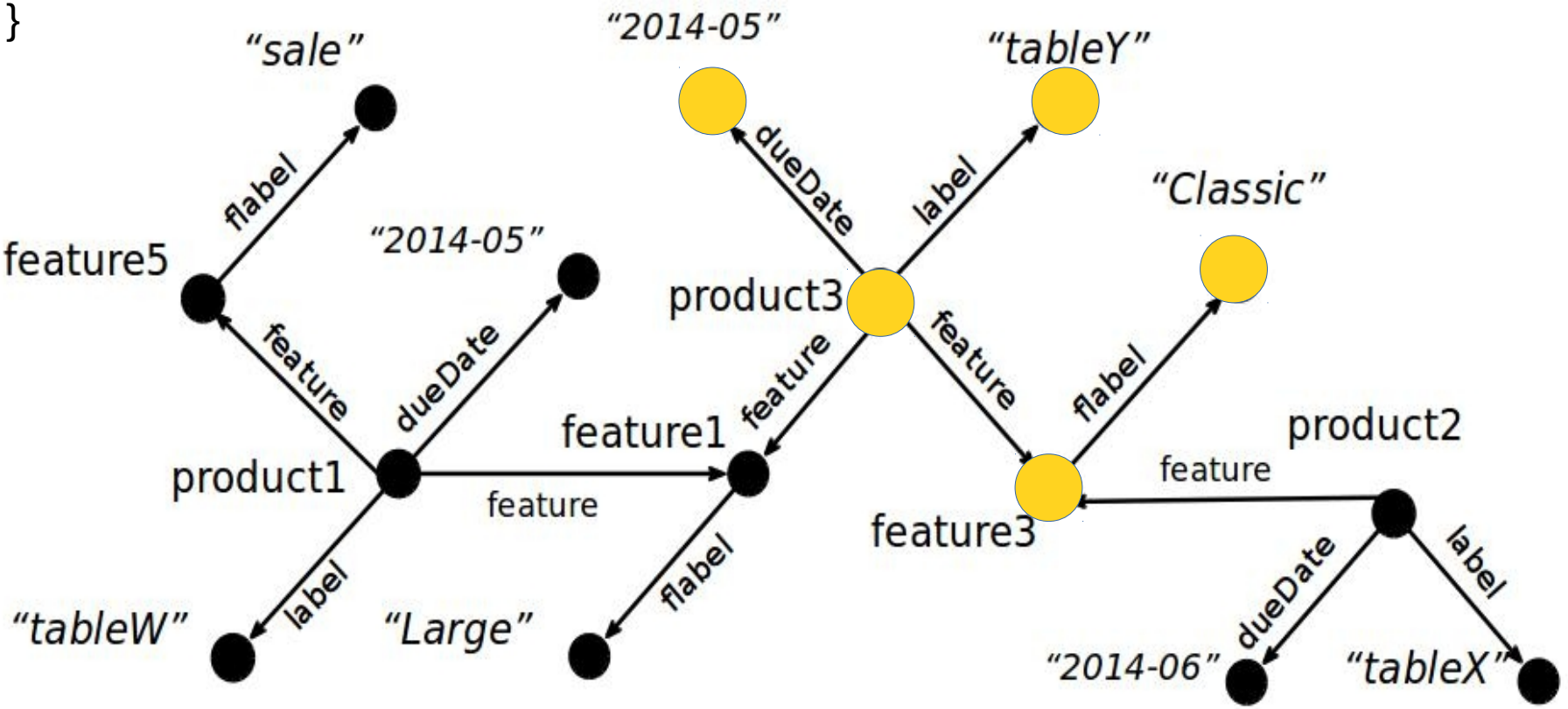
```
select ?l, ?fl
where {
   ?p label ?l .
   ?p dueDate ?d .
   ?p feature ?f .
   ?f flabel ?fl
  FILTER (?d = “2014-05”)
}
```

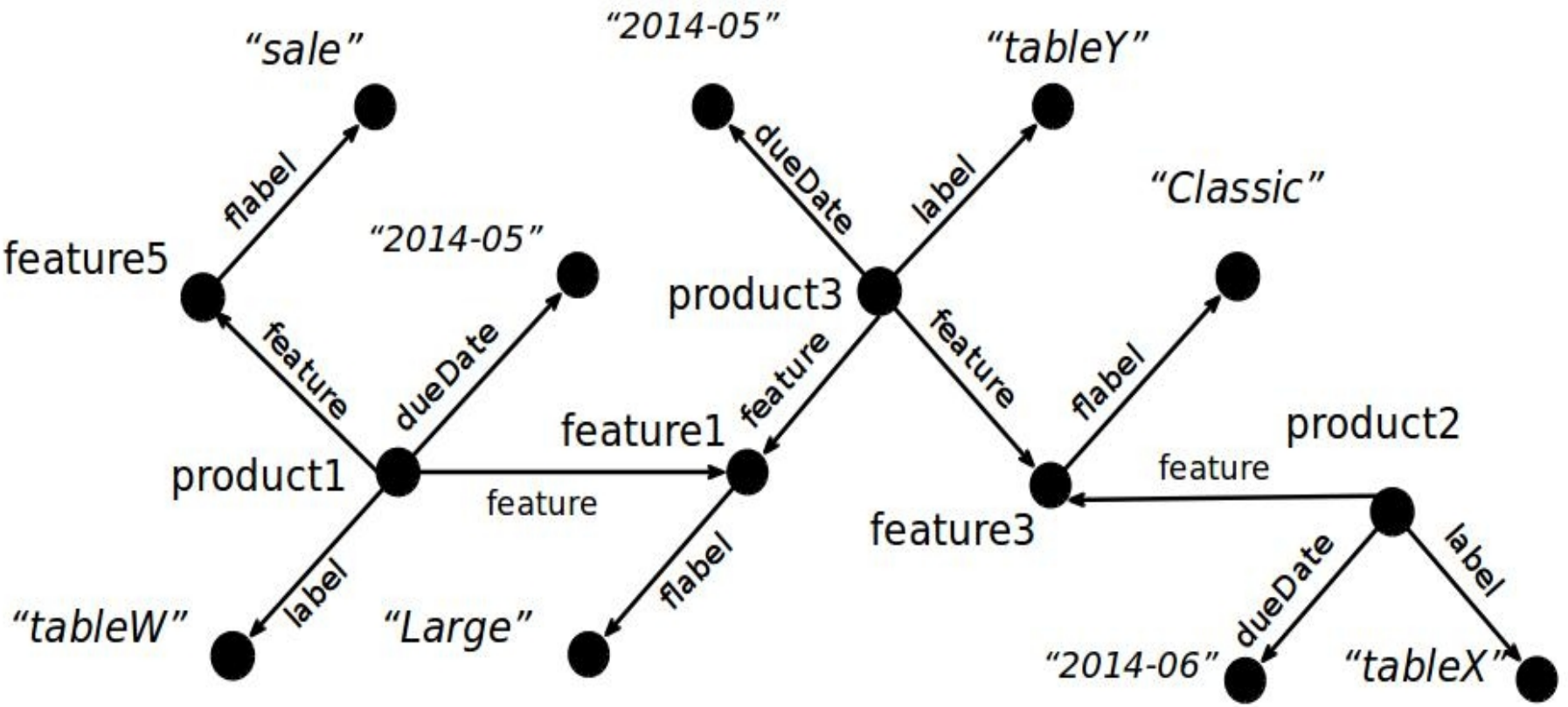


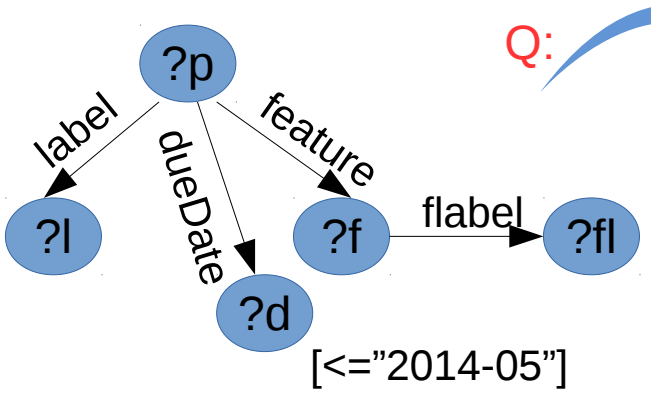


```
select ?l, ?fl
where {
   ?p label ?l .
   ?p dueDate ?d .
   ?p feature ?f .
   ?f flabel ?fl
  filter (?d <= “2014-05”)
}
```

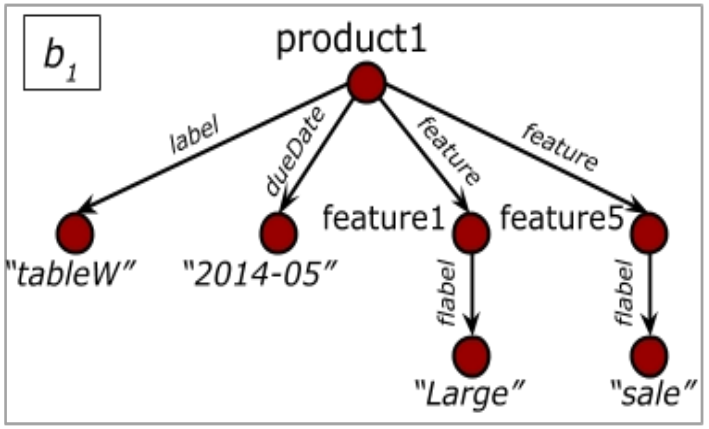


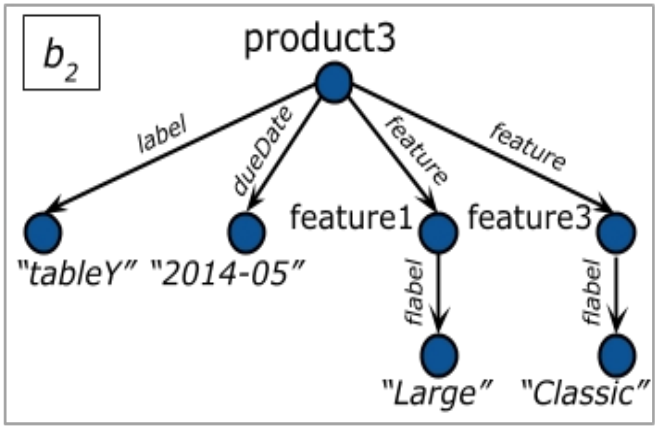



"sale"
"2014-05"
"tableY"
"Classic"
feature5
"2014-05"
product3
flabel
dueDate
label
feature
dueDate
feature
feature
flabel
feature1
product2
product1
feature
feature
feature3
label
dueDate
"tableW"
label
"Large"
flabel
"2014-06"
dueDate
"tableX"


?p
label
dueDate
feature
?l
?f
flabel
?fl
?d
[<="2014-05"]

Q: B(q):


b_1
product1
label
dueDate
feature
feature
feature1
feature5
"tableW"
"2014-05"
flabel
flabel
"Large"
"sale"


b_2
product3
label
dueDate
feature
feature
feature1
feature3
"tableY"
"2014-05"
flabel
flabel
"Large"
"Classic"

Research Problems

- Graph partition:
  - Allocation of subgraphs in different servers
  - Provides storage scalability
- Distributed SPARQL query processing
  - Provides processing scalability
- Integrity constraints for RDF
  - Guaratees data integrity
- Leverage relational databases to store RDF