

具体的な講義の項目

- イン트로ダクション
- データベースのモデル
- 関係データベース(関係代数、関係論理)
- 正規化
- 問い合わせ言語
- トランザクション処理
- データベースの内部構造
- データベースの問い合わせ処理
- その他(オブジェクト指向DB、アクティブDB等々)

2020/6/1

データベース (©横田)

249

他のデータベースのモデル

- オブジェクト指向データベース(OODB)
- オブジェクトリレーショナルデータベース(ORDB)
- 演繹データベース(DDB)
- 演繹オブジェクト指向データベース(DOOD)
- アクティブデータベース(ADB)
- XMLデータベース(XMLDB)
- RDFデータベース(RDFDB)
- データストリーム処理
- ...

2020/6/1

データベース (©横田)

250

RDF

- RDF: Resource Description Framework
- トリプル <Subject, Predicate, Object>で表現
 - Subject: 資源(Resource)の実体
 - Predicate: 実体のプロパティを示す
 - Object: プロパティの値(Resource か Literal)
- トリプルの集合でグラフ構造を構築
 - Subject と Object がノード
 - Predicate が辺

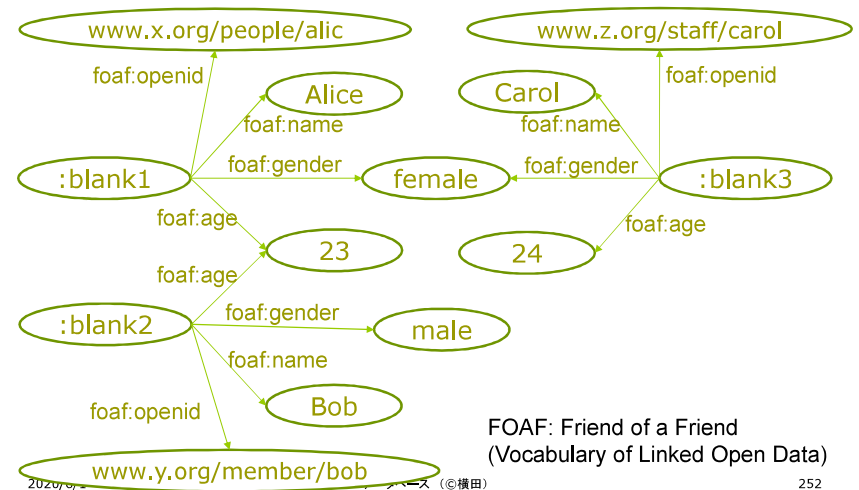


2020/6/1

データベース (©横田)

251

RDF によるグラフ構造の例



2020/6/1

データベース (©横田)

252

SPARQL Query Language

- 再帰的定義 (W3C)
 - SPARQL Protocol and RDF Query Language
- <S, P, O> のパターンマッチ
- Example

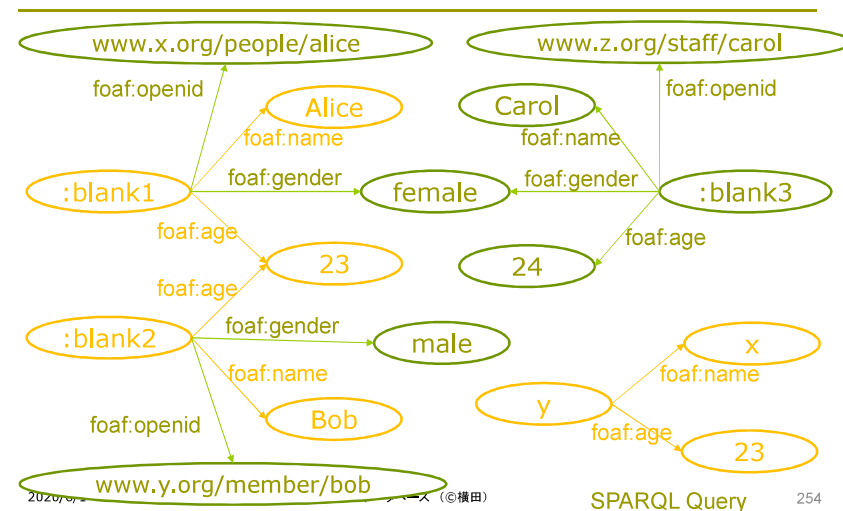

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?x
WHERE {?y foaf:name ?x
        ?y foaf:age 23}
```

2020/6/1

データベース (©横田)

253

Graph Match by SPARQL



2020/6/1

データベース (©横田)

SPARQL Query

254

Image of Self Join

Subject	Predicate	Object	Subject	Predicate	Object
:blank1	foaf:name	Alice	:blank1	foaf:name	Alice
:blank1	foaf:gender	female	:blank1	foaf:gender	female
:blank1	foaf:age	23	:blank1	foaf:age	23
:blank1	foaf:openid	www.x.org/...	:blank1	foaf:openid	www.x.org/...
:blank2	foaf:name	Bob	:blank2	foaf:name	Bob
:blank2	foaf:gender	male	:blank2	foaf:gender	male
:blank2	foaf:age	23	:blank2	foaf:age	23
:blank2	foaf:openid	www.y.org/...	:blank2	foaf:openid	www.y.org/...
:blank3	foaf:name	Carol	:blank3	foaf:name	Carol
:blank3	foaf:gender	female	:blank3	foaf:gender	female
:blank3	foaf:age	24	:blank3	foaf:age	24
:blank3	foaf:openid	www.z.org/...	:blank3	foaf:openid	www.z.org/...

2020/6/1

データベース (©横田)

255

ちょっと考えてみよう(12-1)

- 24歳の女性の名前とopenidを出力する SPARQL 問い合わせを記せ。

2020/6/1

データベース (©横田)

256

演繹データベース

- Deductive Database
- 一階述語論理とリレーショナルデータベースの融合
 - 一階述語論理の単位節を関係と見なす
- 高度な問い合わせ機能(再帰的な VIEW 機能)の提供
 - 停止性問題
 - 否定の取り扱い
 - 現実のアプリケーションが少ない

アクティブデータベース

- Active Database
- 従来のデータベースとアプリケーションの関係
 - データベースは指示されて動作:
 - 受動的
- データベースが状態によって能動的に動作
 - ルールベースシステムとデータベースシステムの融合
 - ECA (Event-Condition-Action)ルール
 - on insert(salary) [Event]
 - if salary > 1,000,000 [Condition]
 - then message('illegal insertion') [Action]

アクティブデータベース

- 演繹データベースも包含
 - 問い合わせをトリガーに新たな問い合わせを生成
- 各種のアプリケーション
- PostgreSQL でもルール記述が可能