

# Performanzsteigerung datenbankgestützter RDF-Triple-Stores

Michael Martin

[martin@informatik.uni-leipzig.de](mailto:martin@informatik.uni-leipzig.de)

UNIVERSITÄT LEIPZIG

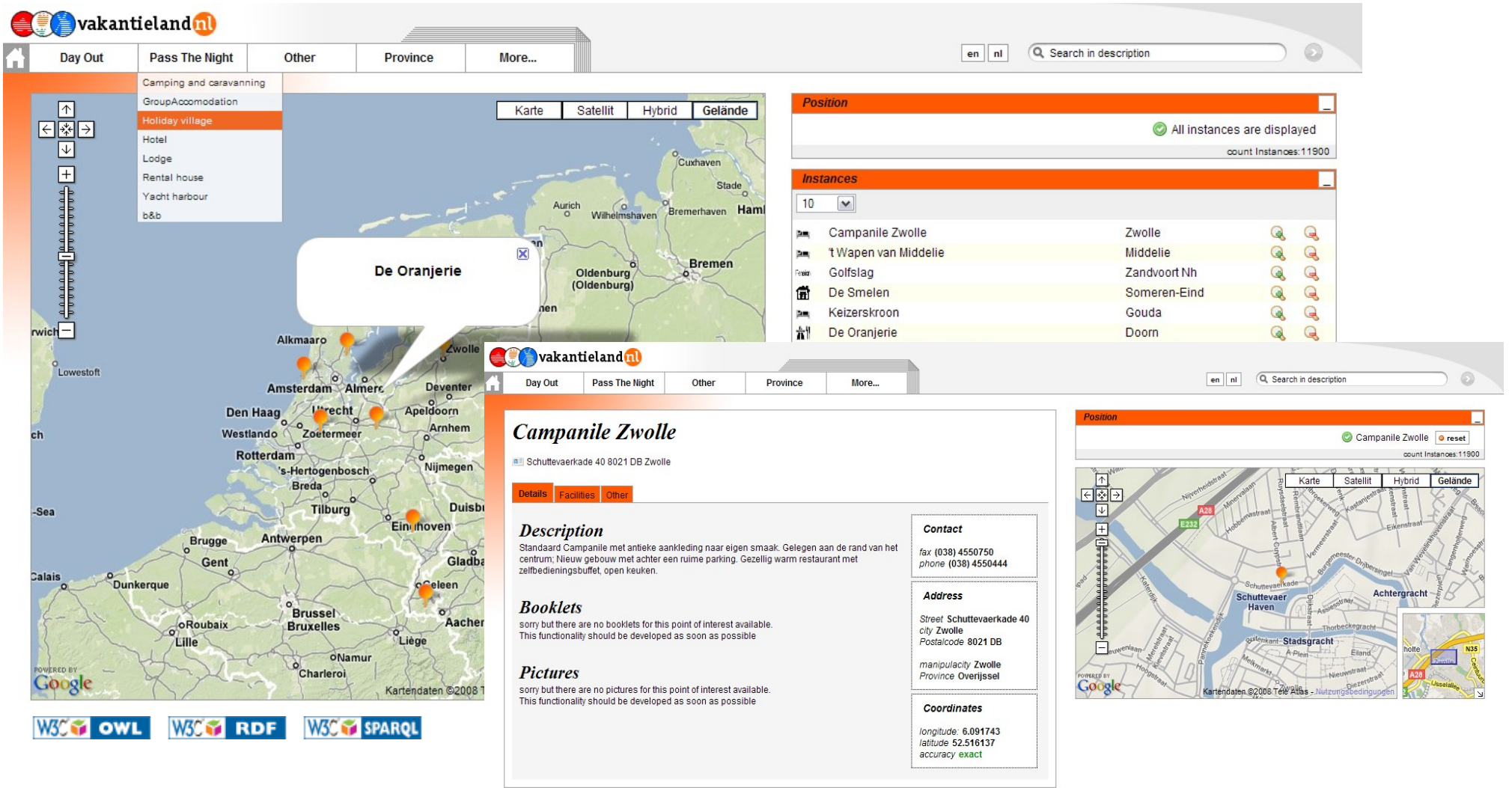
- Motivation
  - Entwicklung semantischer Web Applikationen
  - Problemstellung
- Vorstellung des Ansatzes zur Problemlösung
  - Formales Modell
  - Ansätze zum Caching
- Vorgehensbausteine

- Entwicklung von vakantieland.nl als semantische Web Applikation
- Darstellung von Informationen zu touristischen Zielen

#### Auszug statistischer Daten:

- ~ 20.000 touristische Ziele (point of interests)
- Abgebildet in ~1.000.000 Triple
- ~ 400 Klassen zur Klassifizierung und Separierung der Informationsbereiche der POI's

## Motivation: vakantieland.nl screenshots



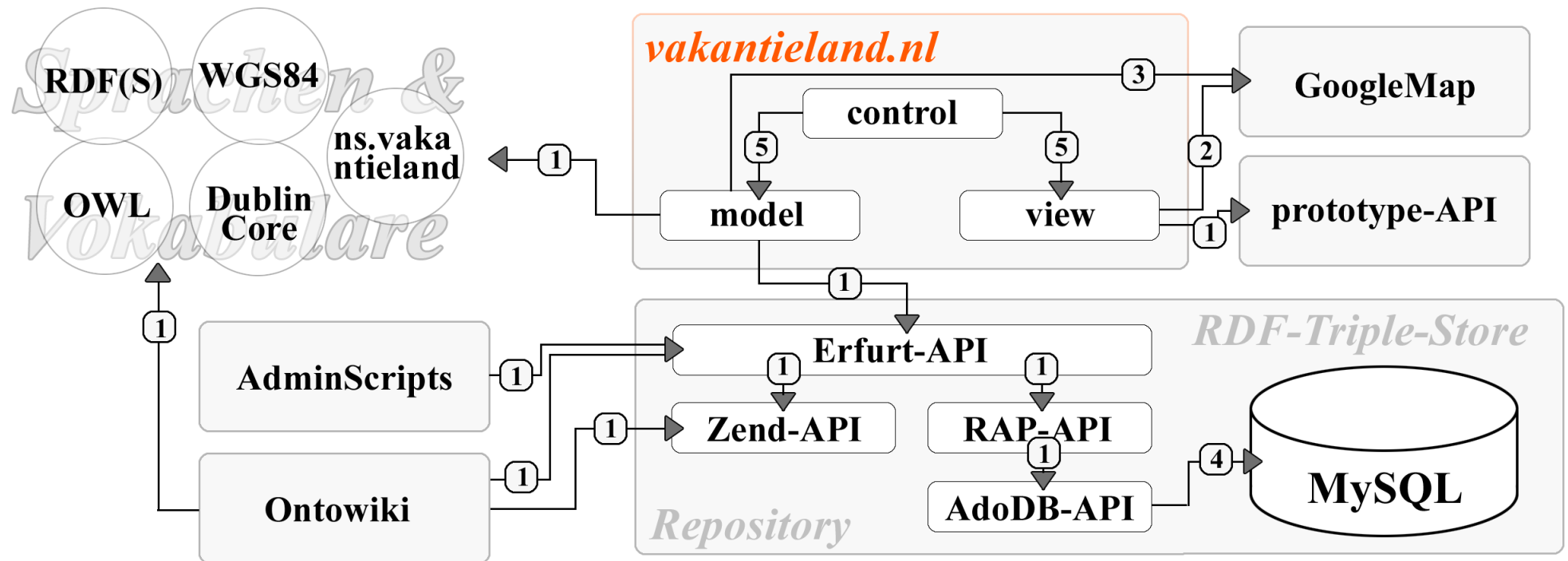
The image displays three screenshots from the website **vakantieland.nl**, illustrating its search and detail capabilities.

**Top Screenshot:** Shows the main search interface with a navigation menu (Day Out, Pass The Night, Other, Province, More...), a search bar, and a list of accommodation types (Camping and caravanning, Group Accommodation, Holiday village, Hotel, Lodge, Rental house, Yacht harbour, b&b). A map shows a location labeled "De Oranjerie".

**Middle Screenshot:** Shows a detailed view of "Campanile Zwolle". It includes a description, contact information (fax (038) 4550750, phone (038) 4550444), address (Street Schuttevaerkaade 40, city Zwolle, Postalcode 8021 DB), and coordinates (longitude: 6.091743, latitude: 52.516137, accuracy exact). It also shows a "Booklets" section with a message that no booklets are available.

**Bottom Screenshot:** Shows a map view of the location "Campanile Zwolle" with a "reset" button and a "count Instances: 11900" indicator.

## Motivation vakantieland.nl: Architektur



### Kantenlegende

1. uses    2. get Map View    3. get Coordinates    4. connect    5. access

- Maßnahmen um auf Performanzprobleme einzugehen

<b>Funktionen</b>	<b>Zeit zur Anzeige der Übersichtsseite</b>	<b>Zeit zur Anzeige der Detailseite</b>
<i>PS0</i>	$\approx 10s$	$\approx 6s$
<i>PS1</i>	$\approx 4,5s$	$\approx 3,5s$
<i>PS1, PS2</i>	$\approx 2,2s$	$\approx 1,5s$
<i>PS1, PS2, PS3</i>	$\approx 650ms$	$\approx 600ms$
<i>PS1, PS2, PS3, PS4</i>	$\approx 210ms$	$\approx 210ms$

Tabelle : Messungen von Zeiten zur Performanzsteigerung

- Implementierte performanzsteigernde Maßnahmen an den Anwendungsfall stark angepasst
- OntoWiki zur Ontologie-Administration müsste mittels Plugins die anwendungsfallorientierten Caching-Lösungen benutzen
- Spezielle Caching-Lösungen müssen bei Änderungen an Modelkomponente erweitert bzw. angepasst werden

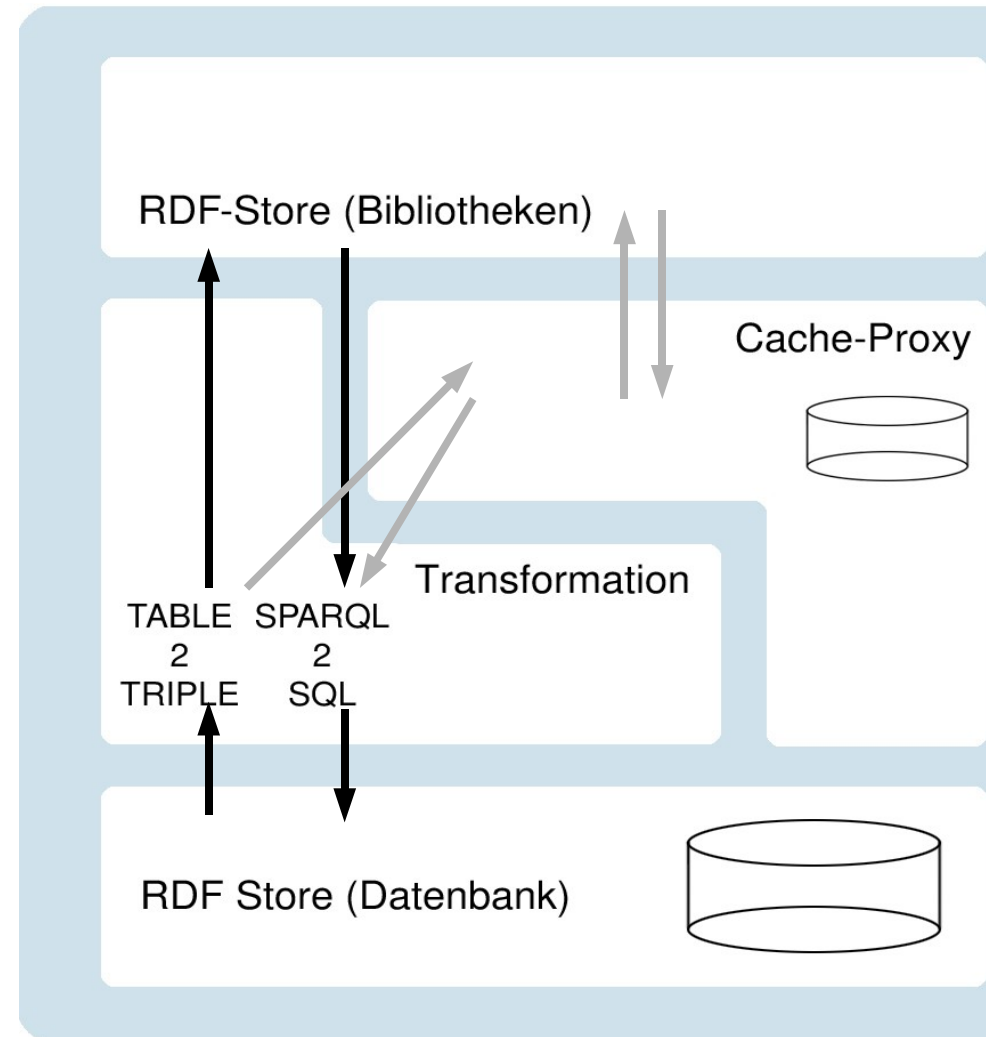
- Caching von Tripeln unabhängig von Anwendungsspezifika
- Erstellung eines formalen Modells zur Beschreibung der Caching-Methoden
- (Referenz-)Implementierung als Proxy-Schicht, die mit verschiedenen RDF-Triple-Stores interagieren kann



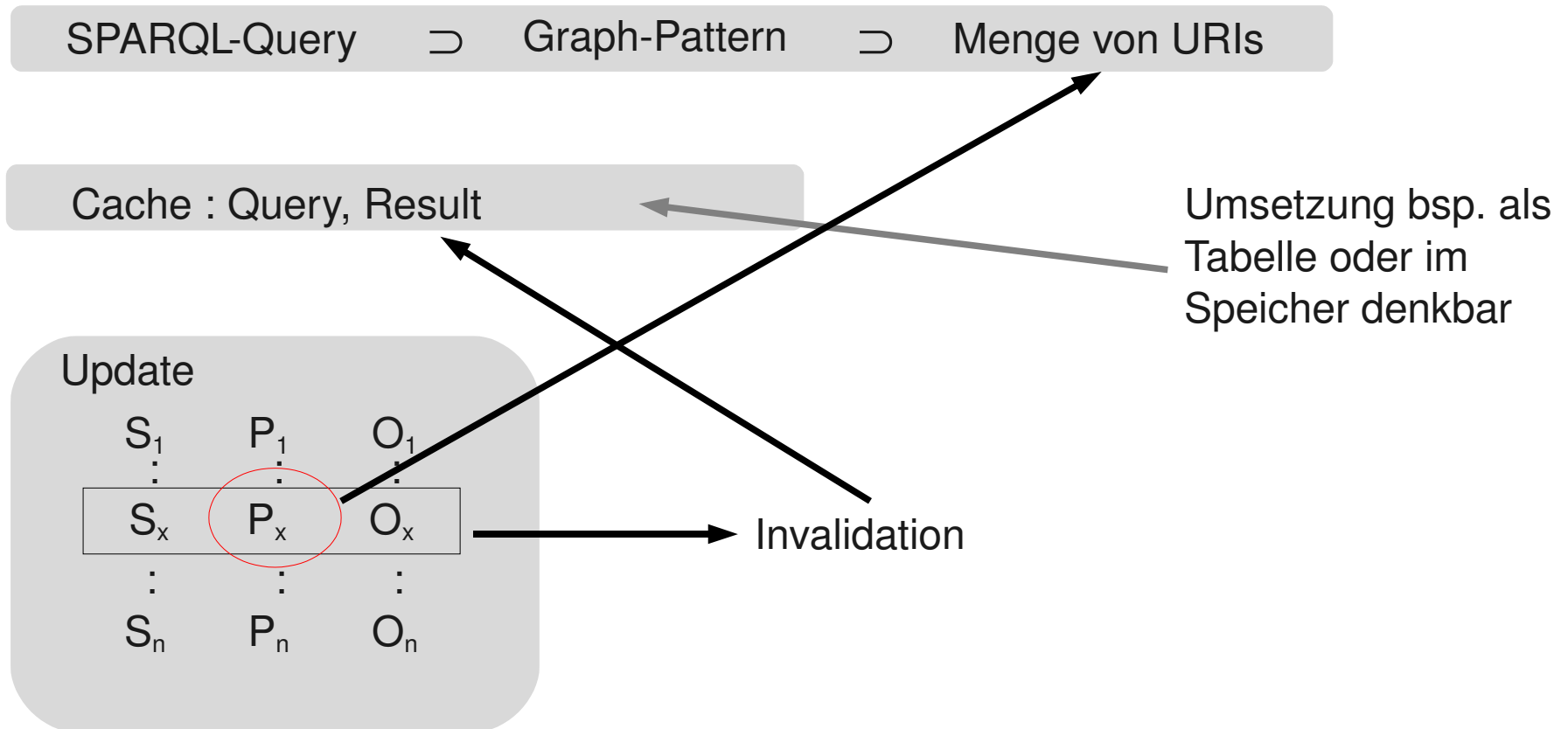
- Orientierung an View-Maintenance Algorithmen aus dem Datenbankbereich
- Abstrakte View-Maintenance-Model wird unter anderem Elemente Tripelpersistenz, Anfragen, Latenz sowie Ausführungs- und View-Materialisierungsgeschwindigkeit beschreiben.
- Heuristiken für das View/Cache-Management sowie regelbasierte Mechanismen zur Query-Subsumption und Cache-Object-Invalidation

## Ansätze zur Problemlösung

- Integration des Caches als Proxyschicht
- Für Interna der Proxyschicht verschiedene Ansätze denkbar



- Caching von Query-Results



## Naiver Ansatz zum Caching

- Umsetzung mit DB:

table graphPattern

qid	S	P	O
1	NULL	rdf:type	foaf:Person
1	NULL	foaf:name	NULL
...	...	...	...

**Query 1:**

```
SELECT ?x, ?y WHERE {
  ?x rdf:type foaf:Person .
  ?x foaf:name ?y }
```

table resultCache

qid	query hash	serial. result	count
1	<hash>	<result>	z
...	...	...	...

**Update : (Micha, rdf:type, foaf:person)**  
**Auswahl betroffener gecachter results:**  
 SELECT qid FROM graphPattern WHERE  
 (S='Micha' OR S=NULL) AND  
 (P="rdf:type" OR P=NULL) AND  
 (O=foaf:person OR O=NULL);

- Ansatz dient als Ausgangsbasis für weitere Evaluationen
- Benchmarking auf Referenzmaschine mit beispielsweise der Berlin SPARQL Benchmark
- Sukzessives Erweitern bis hinreichend annehmbare Performanz erreicht

- In Ontologien kodierte Objekte haben Abhängigkeiten zu anderen Objekten
- Web Applikationen bieten zumeist spezielle Sichten auf die kodierten Daten

table dependencies

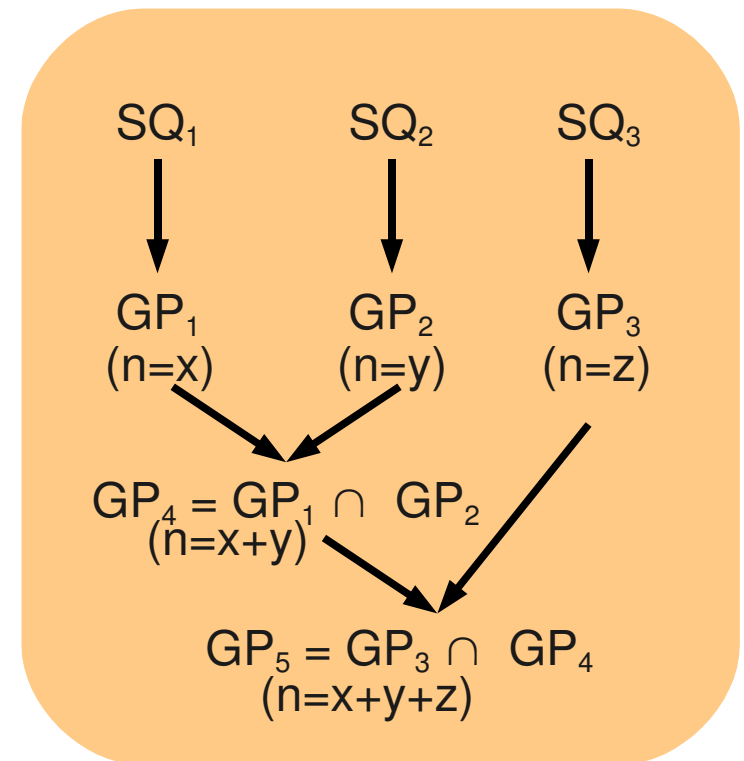
oid	dependsOn	SerializedObject
1	2,7,8	blob
2	4,9	blob
3	NULL	blob

- Ausschnitte aus Ressourcentabellen mittels views materialisieren
- Ähnlich Eigenschaftstabellen, allerdings abhängig von gegebenen GraphPattern
- Ansätze für Heuristiken zur Optimierung sind zu erruieren
- Erstellung/Manipulation/Reorganisation des Caches zu speziellen Zeitpunkten

## Caching mit View Materialization

- Möglicher Algorithmus:

1. Protokollierung aller Queries **MQ** mit Häufigkeit **n**
2. Berechnung aller Schnitte der Graphpattern in **MQ**
3. Sortierung der GP nach **n**
4. Erstellung der Views





- In RDF/S und OWL ist viel Wissen nur implizit kodiert: Inferenz zum Auflösen
- Spezielle Caching-Strategien sind ohne Weiteres nicht in einer unabhängigen Schicht realisierbar

- Erarbeitung des formalen Modells
- Implementierung der Proxyschicht
- Entwicklung der Heuristiken für View/Cache-Management
- Erarbeitung regelbasierter Mechanismen für Query-Subsumption / Cache-Object-Invalidation
- Evaluation / Validation / Benchmarking

- In den Applikationen/Bibliotheken Virtuoso (OpenLink), Jena, Sesame und weiteren sind schon performanzsteigernde Methoden integriert
- Andreas Harth und Stefan Decker. Optimized Index Structures for Querying RDF from the Web. In LA-WEB '05: Proc. of the Third Latin American Web Congress, Seite 71, Washington, DC, USA, 2005. IEEE Computer Society.
- Edward Hung, Yu Deng und V. S. Subrahmanian. RDF Aggregate Queries and Views. In ICDE, Seiten 717–728. IEEE Computer Society, 2005.
- Raphael Volz, Daniel Oberle und Rudi Studer. On Views in the Semantic Web. In 2nd International Workshop on Databases, Documents and Information Fusion (DB-FUSION02), Karlsruhe, Germany, 07 2002.
- Aimilia Magkanaraki, Val Tannen, Vassilis Christophides und Dimitris Plexousakis. Viewing the semantic web through RVL lenses. J. Web Sem., 1(4):359–375, 2004.

- Sören Auer, Sebastian Dietzold und Thomas Riechert. OntoWiki - A Tool for Social, Semantic Collaboration. In Proc. of 5th ISWC 2006, Jgg. 4273 of Lecture Notes in Computer Science, Seiten 736–749. Springer, 2006.
- Chris Bizer und Andreas Schultz. Berlin SPARQL Benchmark (BSBM) Specification- V2.0, September 2008.
- Jan Große. Speicherverfahren und Werkzeuge für RDF/S. XML Clearinghouse Report 6, Herausgeber: Prof. Dr. Ing. Robert Tolksdorf (Freie Universität Berlin) und Dr. Rainer Eckstein (Humboldt Universität Berlin), Dezember 2003.
- Yue Zhuge, Héctor García-Molina, Joachim Hammer und Jennifer Widom. ViewMaintenance in a warehousing environment. In Proc. of 1995 ACM SIGMOD, Seiten 316–327, New York, NY, USA, 1995. ACM.
- Michael Martin. Exploring the Netherlands on a Semantic Path. In CSSW, Jgg. 113 of LNI, Seiten 179–. GI, 2007.

# Danke für die Aufmerksamkeit

```
<foaf:name>Michael Martin</foaf:name>  
<foaf:mbox rdf:resource="mailto:martin@informatik.uni-leipzig.de" />
```

UNIVERSITÄT LEIPZIG