Unterstützung von datenbankorientierten Software-Produktlinien durch domänenspezifische Spracherweiterungen für SQL

„Eine Hauptaufgabe der Informatik ist systematische Abstraktion“ (H. Wedekind)

Christian Weber
christian@chrweber.de

29. März 2004

Motivation

- Wiederverwendung von Software
- Software-Produktlinien
- Generative Programmierung
- Domänenspezifische Sprachen
- Ziel der Arbeit:
  - Untersuchung der Konzepte zur Unterstützung datenbankorientierter Software-Produktlinien
  - Untersuchung von verschiedenen Datenbankschemata
  - Entwicklung von domänenspezifischen Datenbanktreibern
  - Durchführung von Leistungsuntersuchungen

Gliederung

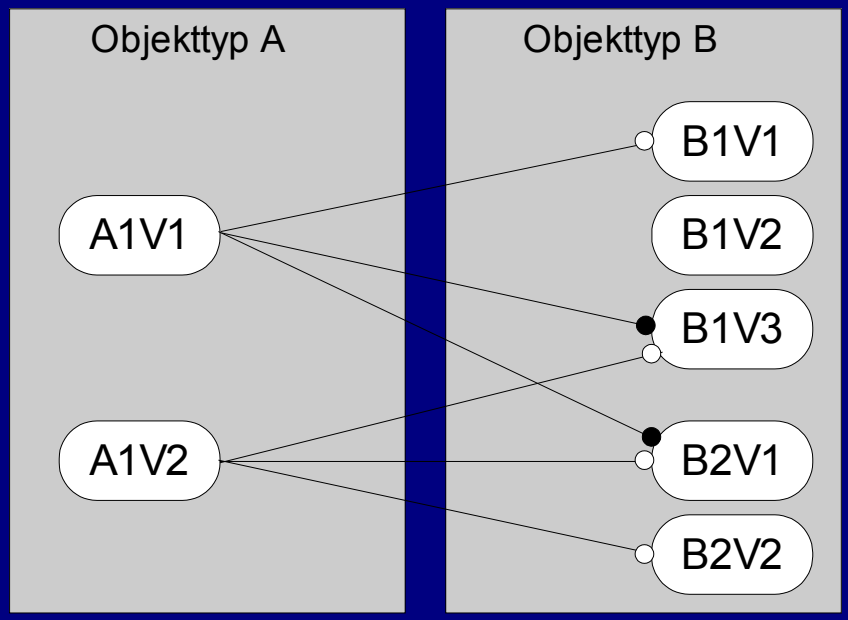
- **Anwendungsdomäne: Versionierungssysteme**
- Generative Programmierung
- Domänenspezifische Sprachen und Schemaentwurf
- Bewertung
- Zusammenfassung und Ausblick

Versionierungssysteme

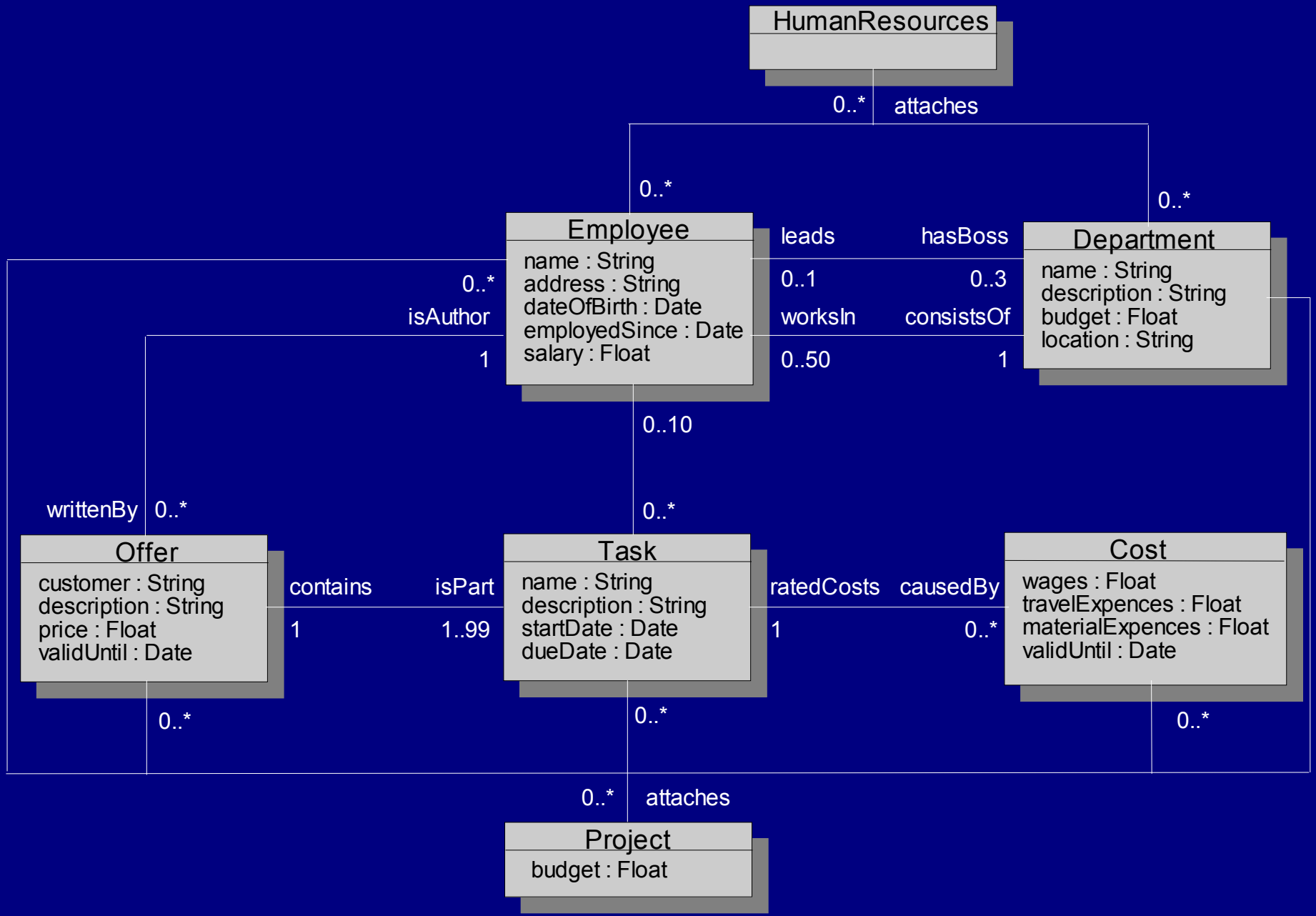
- Speicherung versionierter und nicht versionierter Objekte
- Operationen auf Objekten
- Beziehungen zwischen Objekten
  - Gleitende Beziehungsenden
  - Navigation
  - Propagierung von Operationen
- Informationsmodell

Gleitende Beziehungsenden

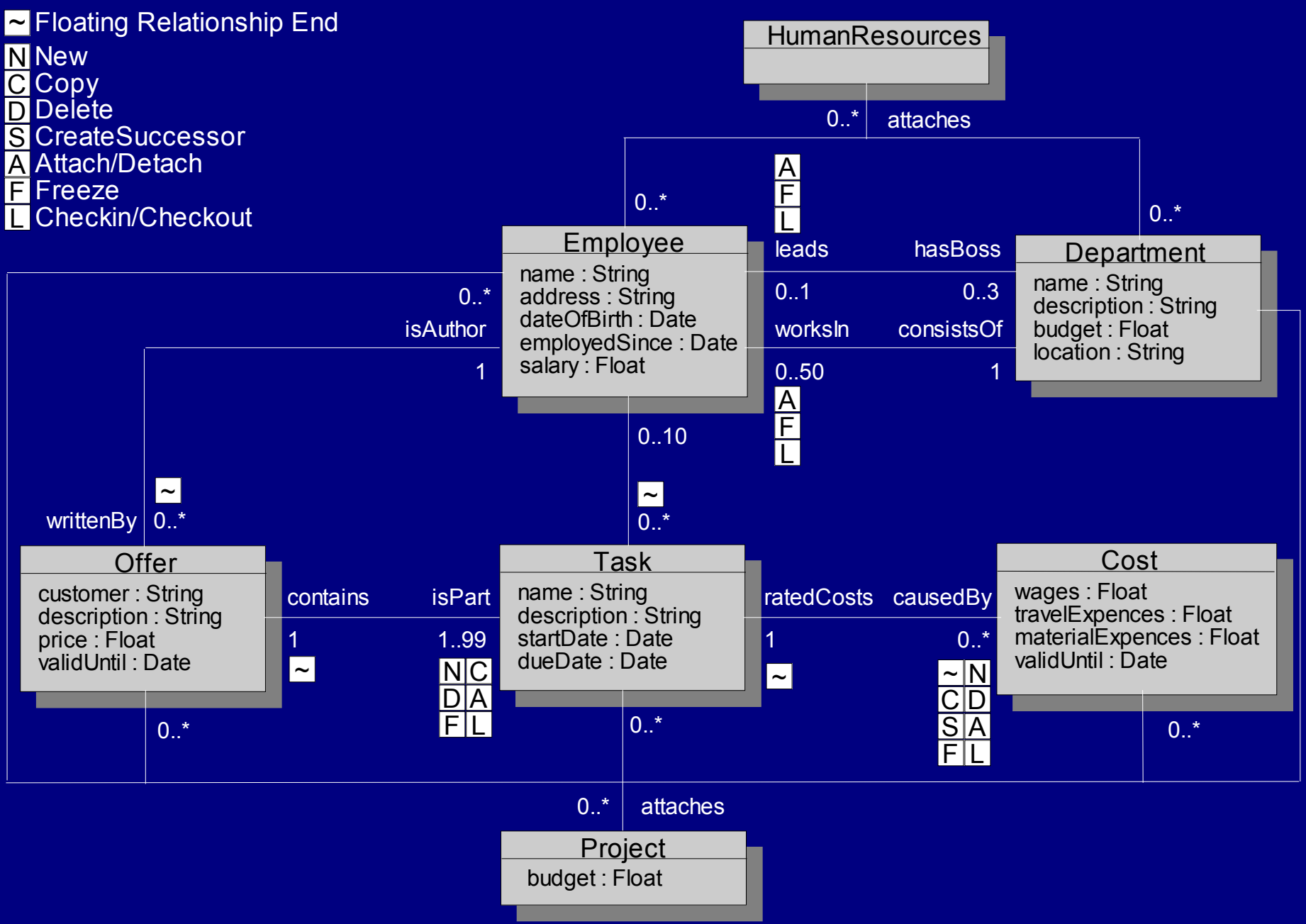
- *floating relationship ends*
- Assoziation nicht auf eine Objektversion festgelegt
- Kandidatenmenge (*candidate version collection*)
- Zugriffsmöglichkeiten:
  - Explizite Festlegung (*pinning*)
  - Regelbasierte Auswahl
  - Gesamte Kandidatenmenge
  - Arbeitskontexte



Informationsmodell



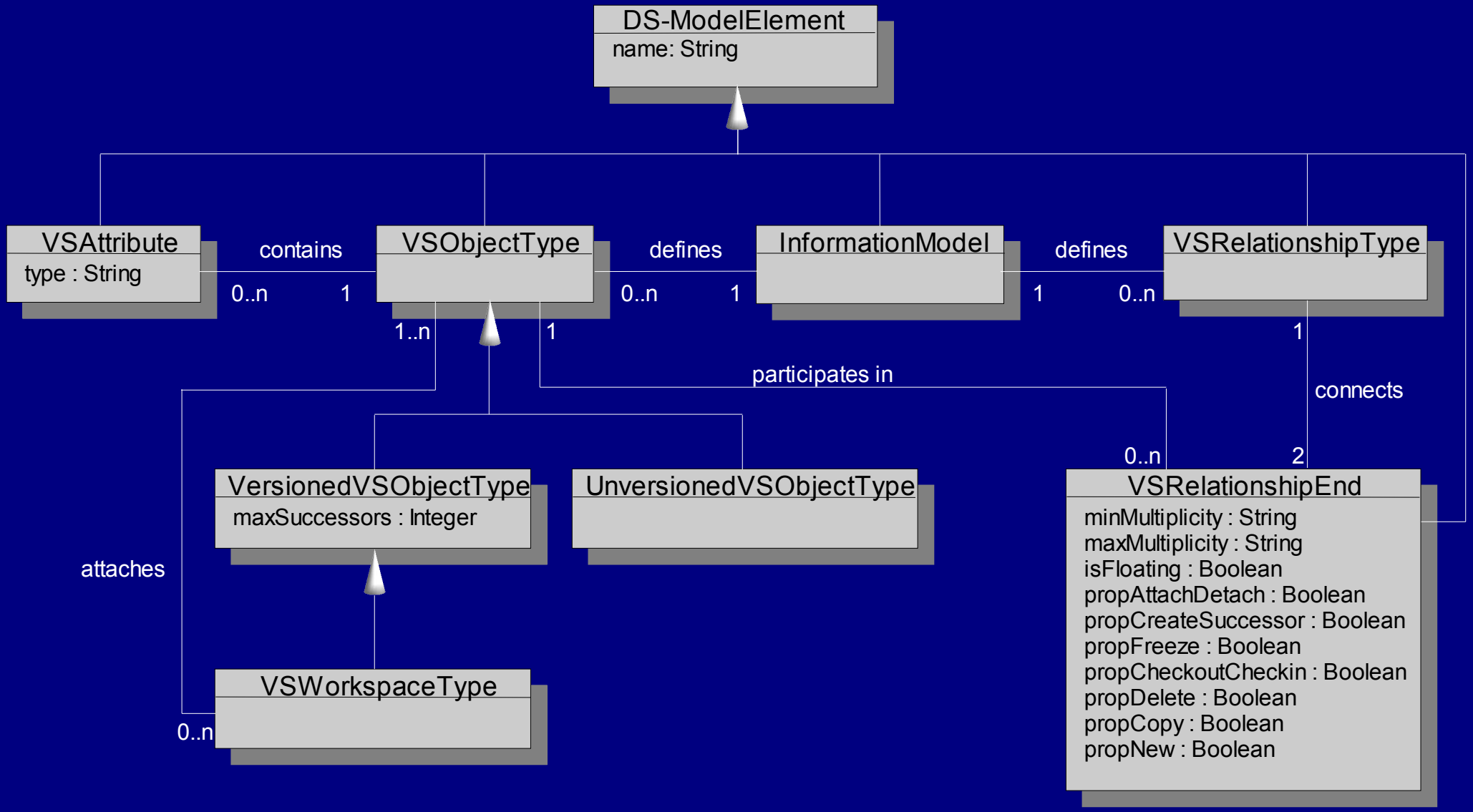
Informationsmodell mit DS-Erweiterungen



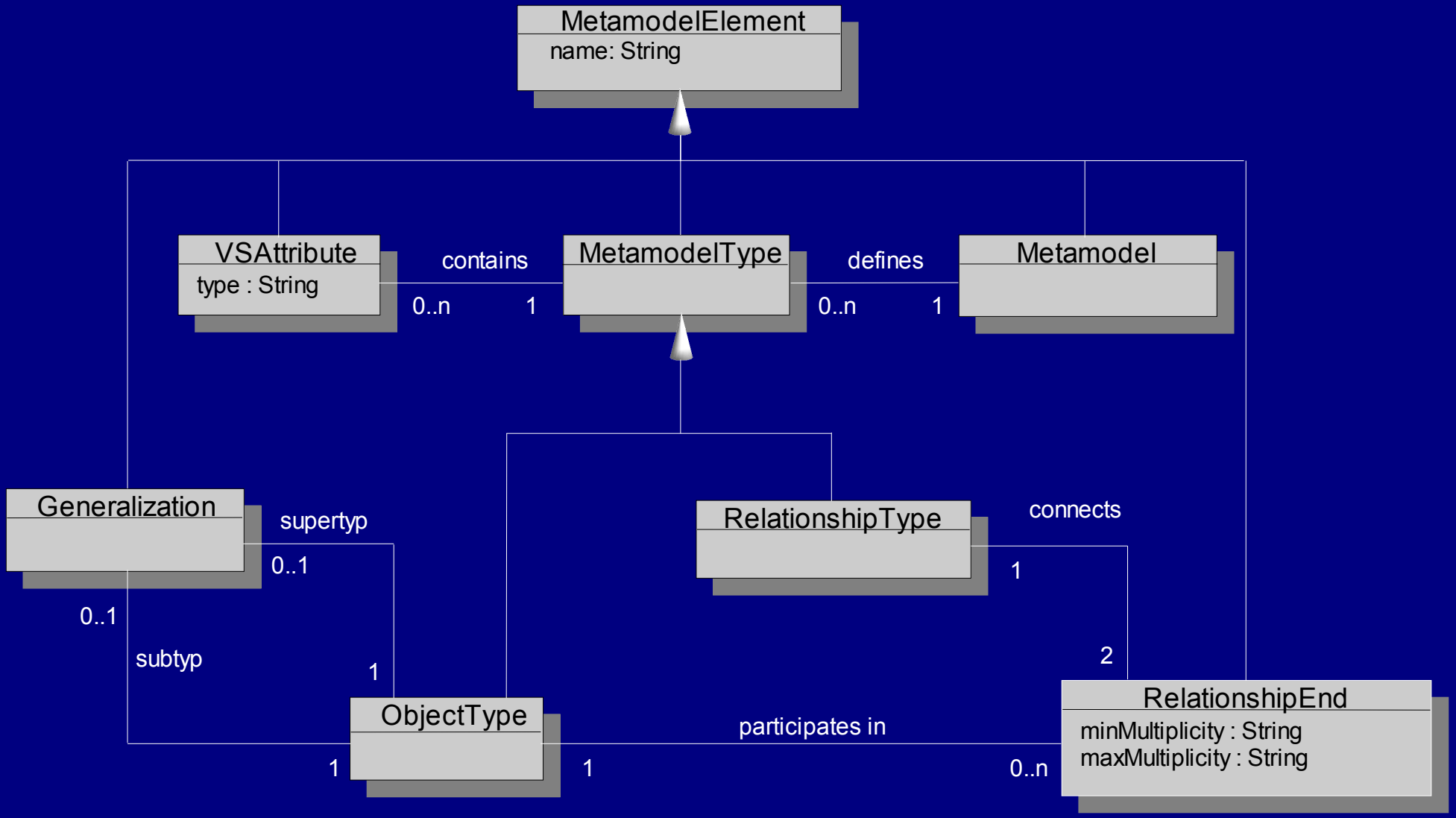
Gliederung

- Anwendungsdomäne: Versionierungssysteme
- **Generative Programmierung**
- Domänenspezifische Sprachen und Schemaentwurf
- Bewertung
- Zusammenfassung und Ausblick

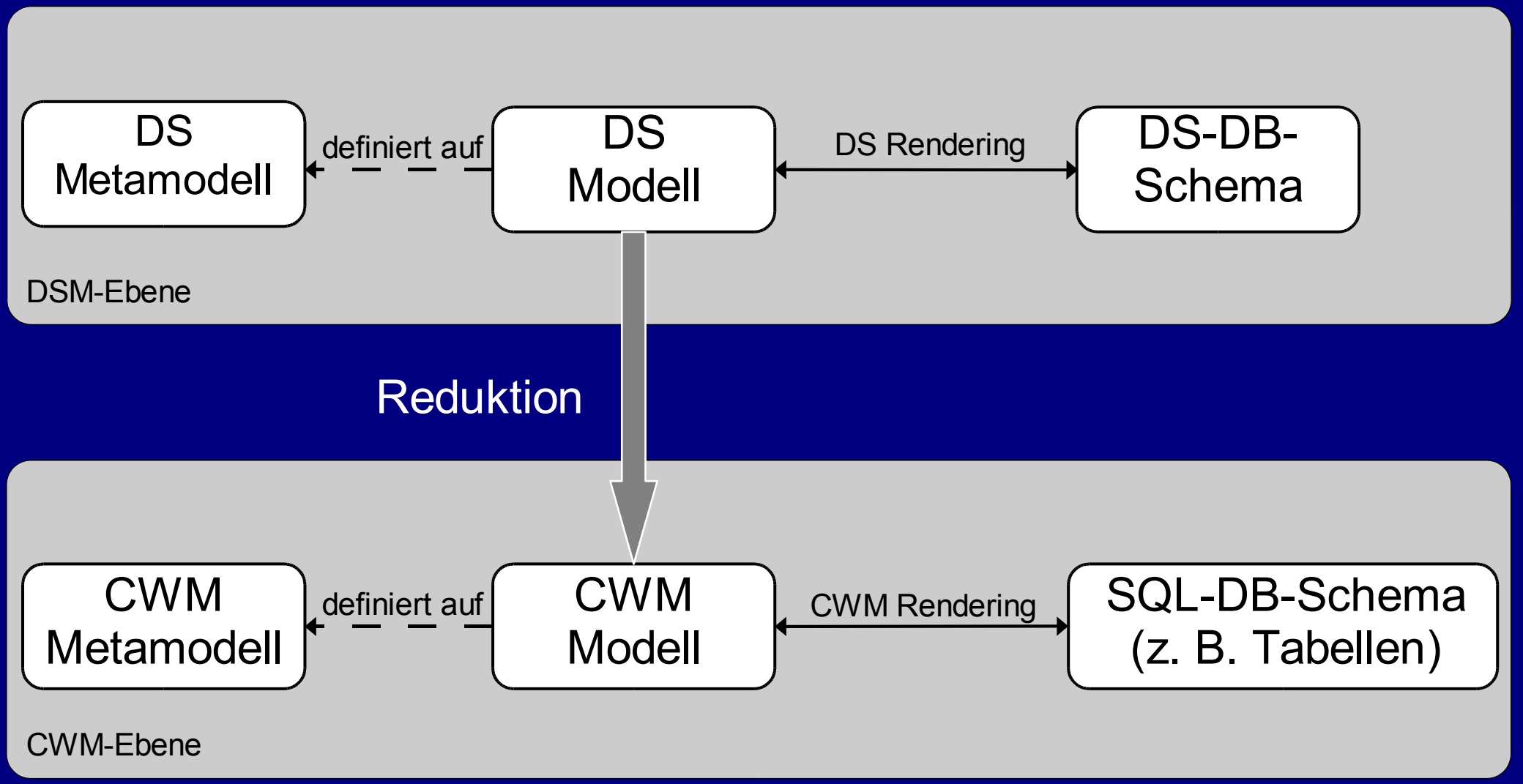
Metamodell für Versionierungssysteme



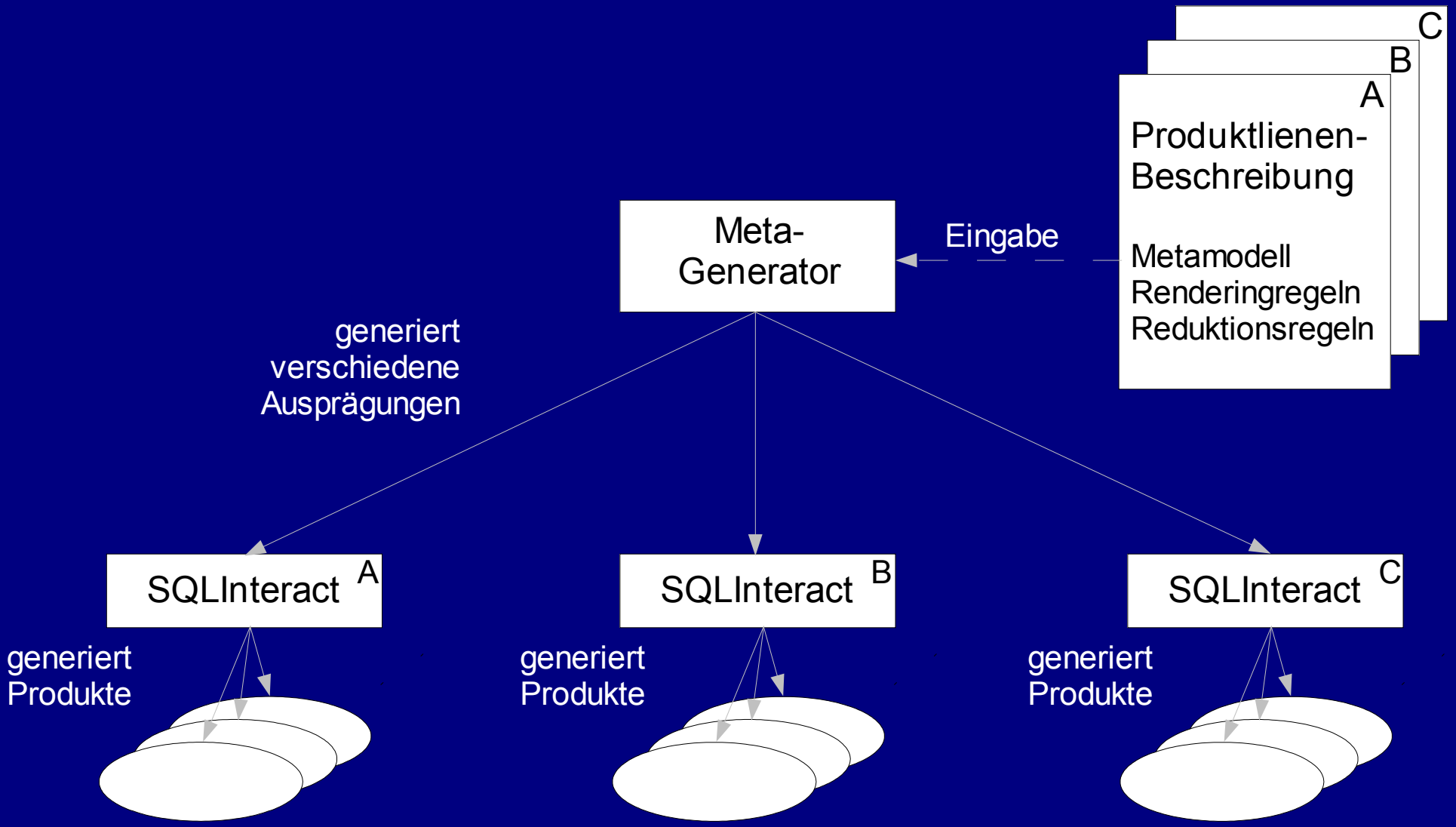
Meta-Metamodell



DSM- und CWM-Ebene

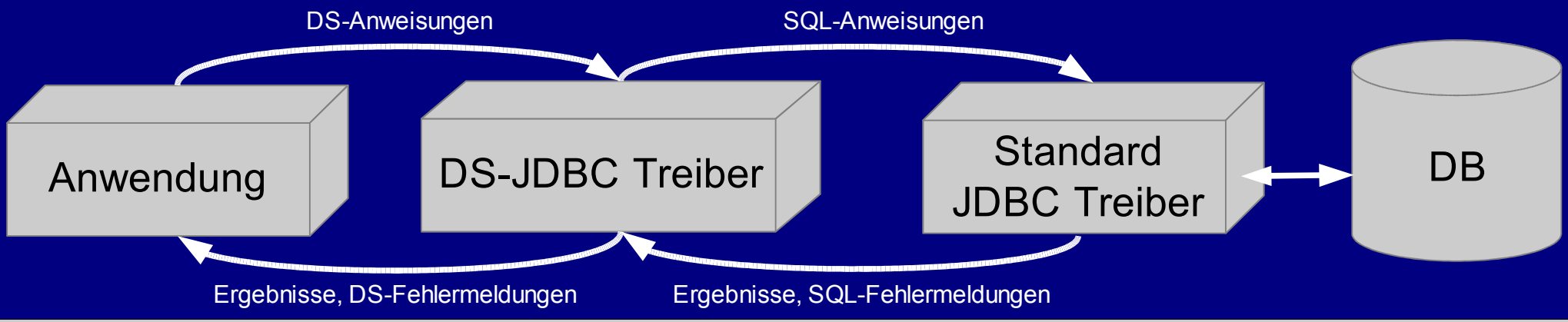


Metagenerator



Anwendungslogik

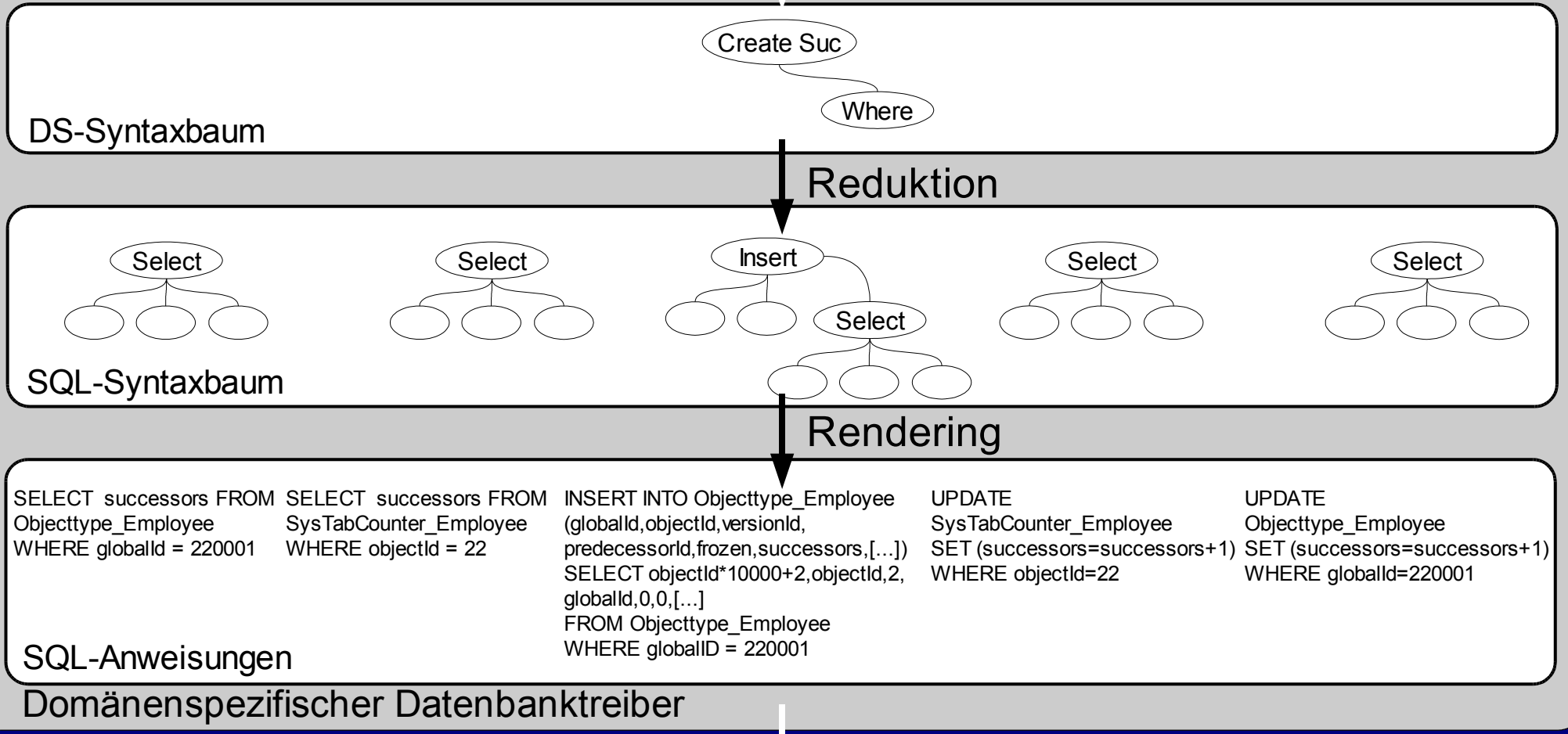
- Realisierung der Anwendungslogik:
  - innerhalb der Datenbank (z. B. gespeicherte Prozeduren)
  - außerhalb in einem domänenspezifschen Datenbanktreiber
- domänenspezifischer Datenbanktreiber:
  - komfortable Schnittstelle
  - Übersetzer der domänenspezifischen Anweisungen in SQL-Anweisungen
  - kein Performanz Gewinn



Abläufe im domänenspezifischen Datenbanktreiber

CREATE SUCCESSOR OF OBJECT Employee WHERE GID = 22001

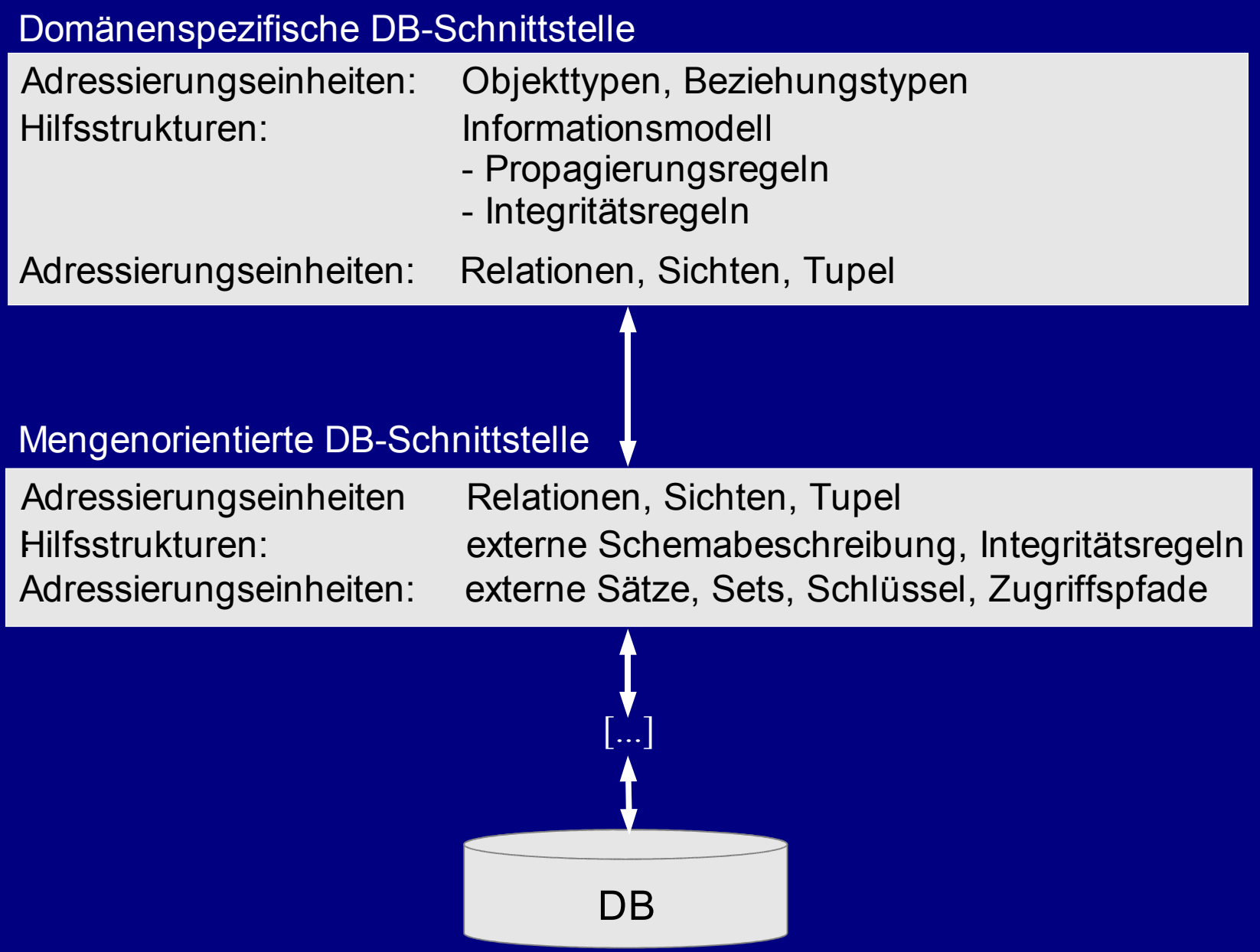
Parsen



SQL-Ausführung

DBMS-spezifischer Datenbanktreiber

Unabhänigkeit von logischen Datenstrukturen



Gliederung

- Anwendungsdomäne: Versionierungssysteme
- Generative Programmierung
- **Domänenspezifische Sprachen und Schemaentwurf**
- Bewertung
- Zusammenfassung und Ausblick

Domänenspezifische Sprachen

- Problemorientierte Sprache
  - Begriffe und Merkmale der Anwendungsdomäne
  - Höhere Abstraktionsebene
- Konfigurationssprache (DS-DDL)
- Anfrage- und Manipulationssprache (DS-DML)
  - Zusammenfassung meherer SQL-Anweisungen
  - Abstraktion vom Datenbankschema

Beispiel: Manipulationssprache für VS

CREATE SUCCESSOR OF OBJECT Offer WHERE GID = 7702

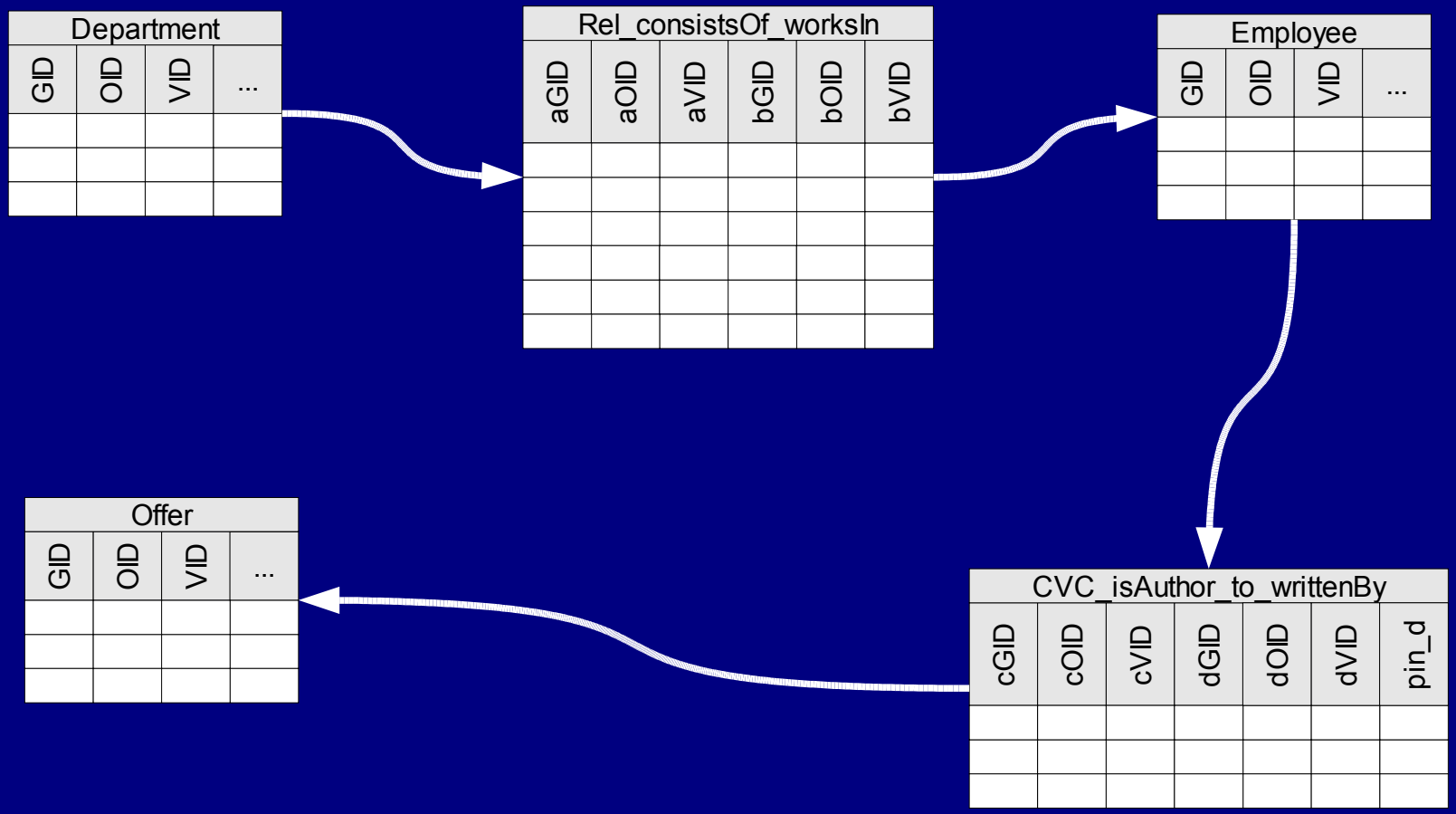
```
SELECT successors FROM Objecttype_Offer
WHERE globalId = 7702

SELECT successors FROM SysTabCounter_Offer
WHERE objectId = 77

INSERT INTO Objecttype_Offer
(globalId,objectId,versionId,predecessorId,frozen,successors,[...])
SELECT objectId*10000+2, objectId,2,globalId,0,0, [...]
FROM Objecttype_Offer
WHERE globalId = 7702

UPDATE SysTabCounter_Offer
SET (successors = successors+1)
WHERE objectId = 77

UPDATE Objecttype_Offer
SET (successors = successors+1)
WHERE globalId = 7702
```

Offer						
globalID	objectID	versionID	predecessorID	frozen	sucesseors	...
7701	77	1	null	1	1	
7702	77	2	7701	0	1	
7703	77	3	7702	0	0	

SysCounter_Offer	
objectID	successors
77	2

Beispiel: Anfragesprache für VS

```
SELECT Offer.*
FROM Department -- consistsOf --> Employee -- isAuthor --> Offer
WHERE Department.globalId = 4001
```



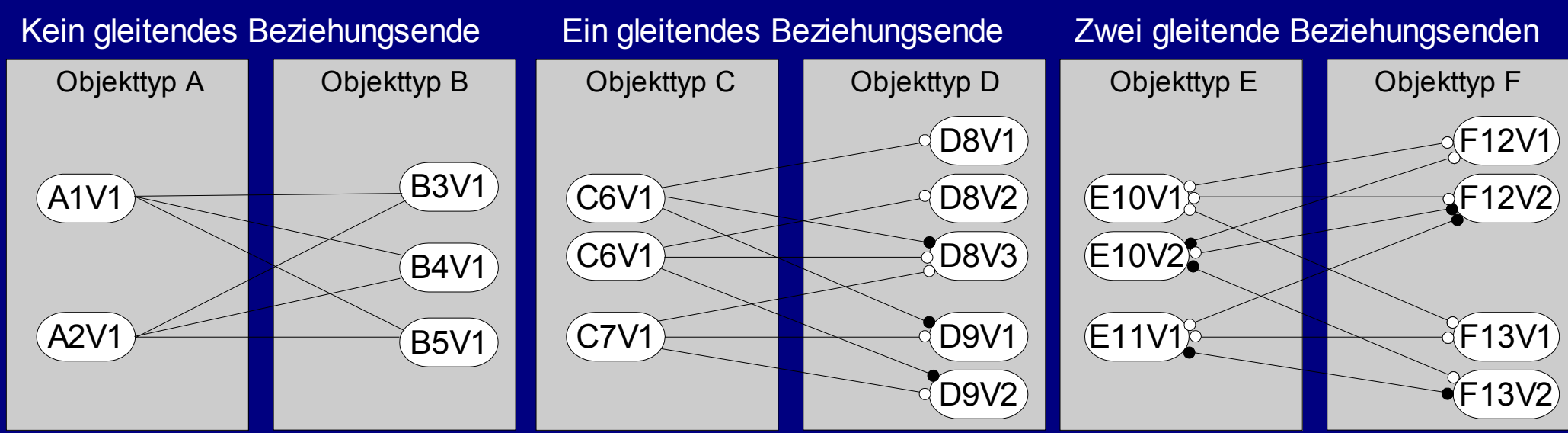
Beispiel: Anfragesprache für VS

```
SELECT Offer.*
FROM Department -- consistsOf --> Employee -- isAuthor --> Offer
WHERE Department.globalId = 4001
```

```
SELECT ObjectType_Offer.* FROM ObjectType_Offer
WHERE ObjectType_Offer.globalId in (
  SELECT writtenByGID AS globalID
  FROM Rel_CVC_isAuthor_to_writtenBy
  WHERE isAuthorGID  in (
    SELECT Rel_worksIn_consistsOf.worksInGID
    FROM ObjectType_Department , Rel_worksIn_consistsOf
    WHERE ObjectType_Department.globalId=4001
    AND Rel_worksIn_consistsOf.consistsOfGID = ObjectType_Department.globalId )
  AND pinned_writtenBy = 1 OR writtenByGID IN (
    SELECT max(writtenByGID) AS globalId
    FROM Rel_CVC_isAuthor_to_writtenBy
    WHERE isAuthorGID  in (
      SELECT Rel_worksIn_consistsOf.worksInGID
      FROM ObjectType_Department , Rel_worksIn_consistsOf
      WHERE ObjectType_Department.globalId=4001
      AND Rel_worksIn_consistsOf.consistsOfGID = ObjectType_Department.globalId )
  GROUP BY writtenByOID, isAuthorGID HAVING MAX(pinned_writtenBy) = 0 )
  )
```

Reduktion

- Objekttypen:
  - Unversionierte Objekttypen
    - GlobalId, Attribute für Nutzdaten
  - Versionierte Objekttypen
    - GlobalId, ObjectId, VersionId, PredecessorId, successors, checkout, frozen, Attribute für Nutzdaten
  - Arbeitskontexte
    - GlobalId, ObjectId, VersionId, PredecessorId, successors, checkout, frozen, Attribute für Nutzdaten
    - Tabellen für Attachment-Beziehungen
- Beziehungstypen



Reduktion der Beziehungstypen

- Variante 1:
  - eine Tabelle für alle Beziehungsinformationen
  - für jede Richtung der Beziehung ein Tupel
- Variante 2:
  - für jeden Beziehungstyp eigene Tabellen
  - Unterschiede bei der Reduktion der unterschiedlichen Beziehungstypen
- Variante 3:
  - Optimierungen falls die maximale Kardinalität 1 beträgt
- Variante 4:
  - Materialisierung der vorausgewählten bzw. durch eine Regel ausgewählten Objektversion

Gliederung

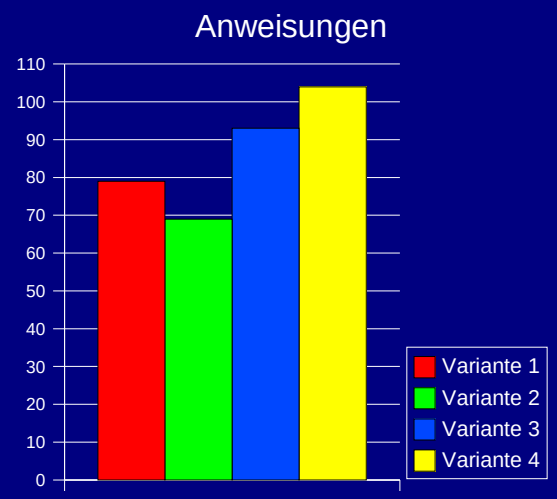
- Anwendungsdomäne: Versionierungssysteme
- Generative Programmierung
- Domänenspezifische Sprachen und Schemaentwurf
- **Bewertung**
- Zusammenfassung und Ausblick

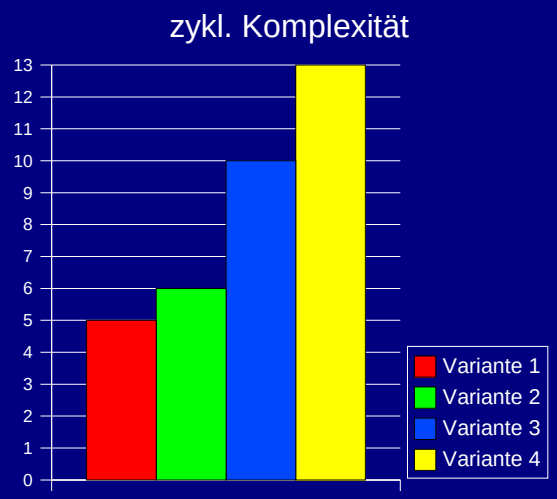
Hypothesen

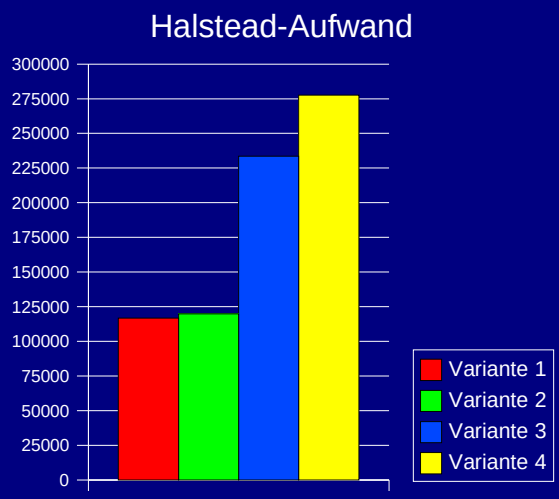
- Aufwand für die Erstellung und Wartung der Reduktionsmethoden steigt (akzeptabel)
- Die Zeiten für die Reduktion steigen
- SQL-Ausführungszeiten nehmen ab
- Die für die Reduktion benötigte Zeit ist gering im Vergleich zur SQL-Ausführung
- Operationen können mit Hilfe von gespeicherten Prozeduren beschleunigt werden

Analyse mit Softwaremetriken

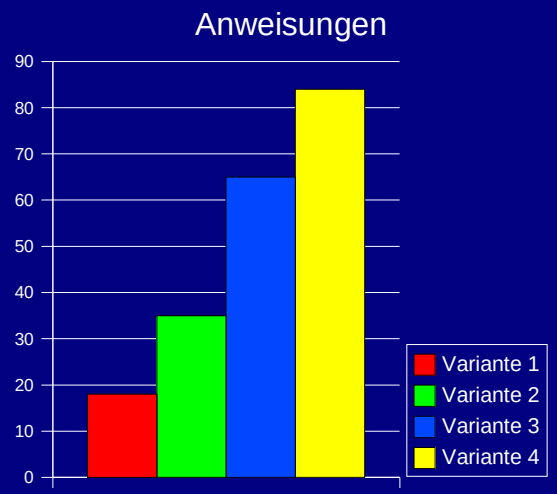
- Operation `NewRelationship`

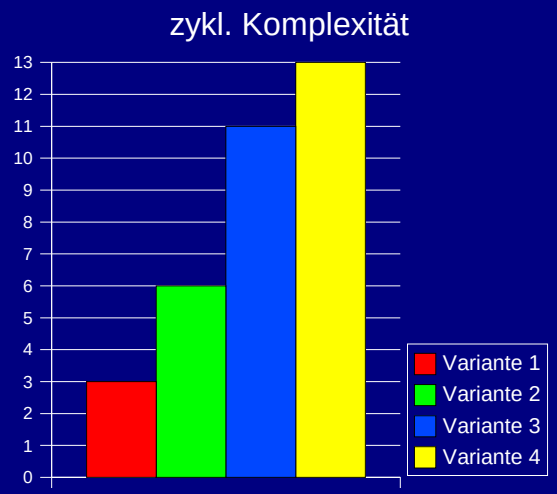


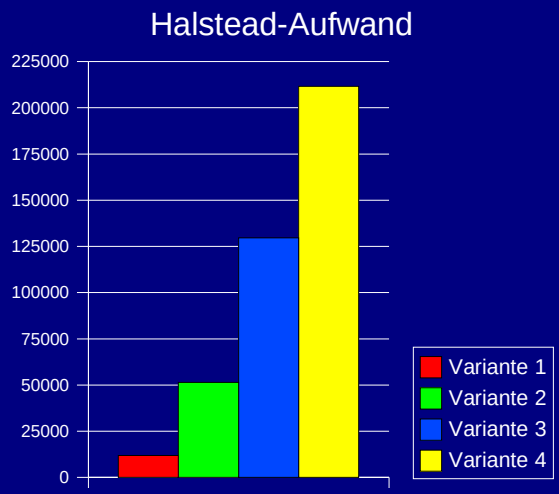




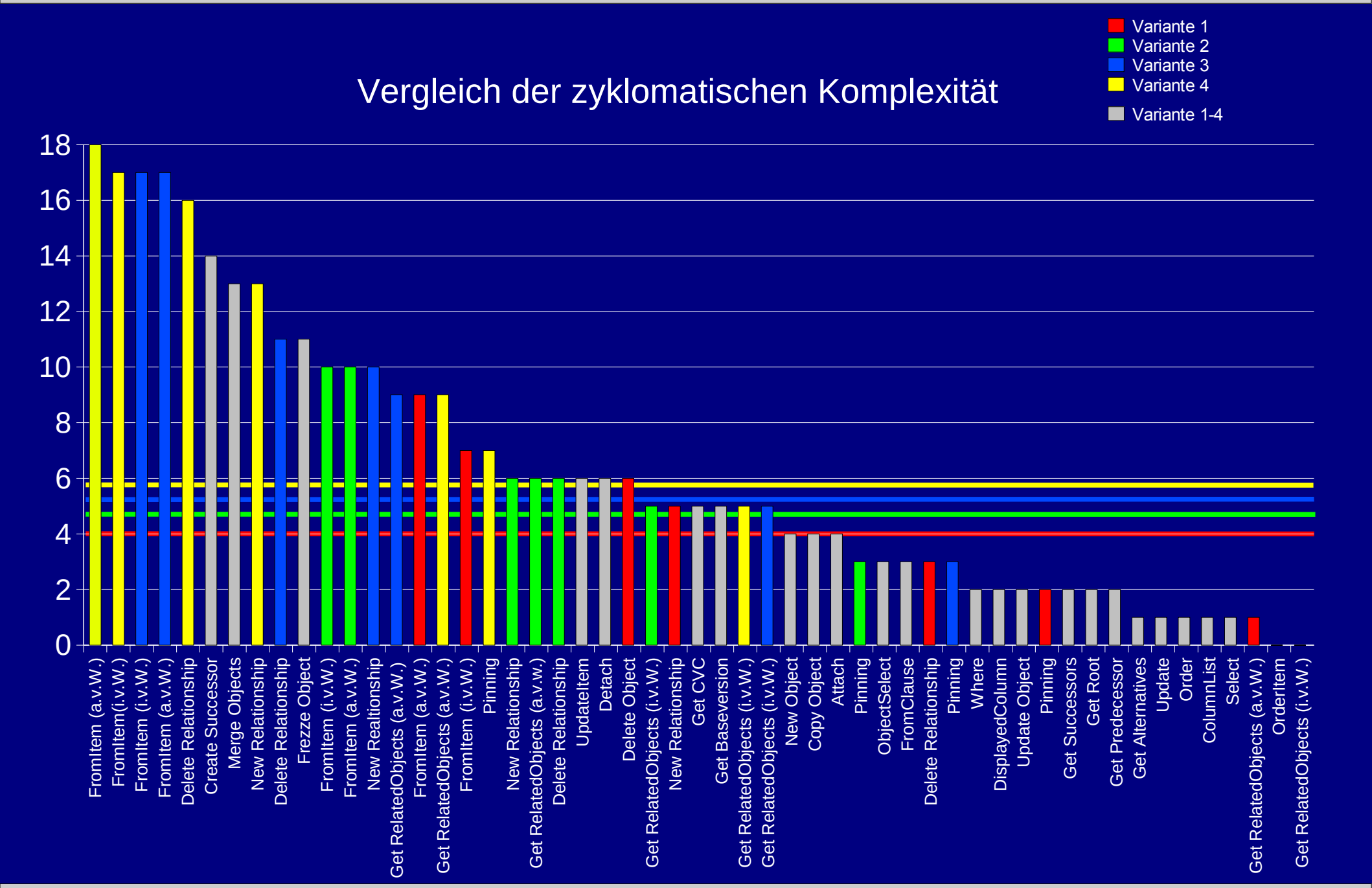
- Operation `DeleteRelationship`







Analyse mit Softwaremetriken

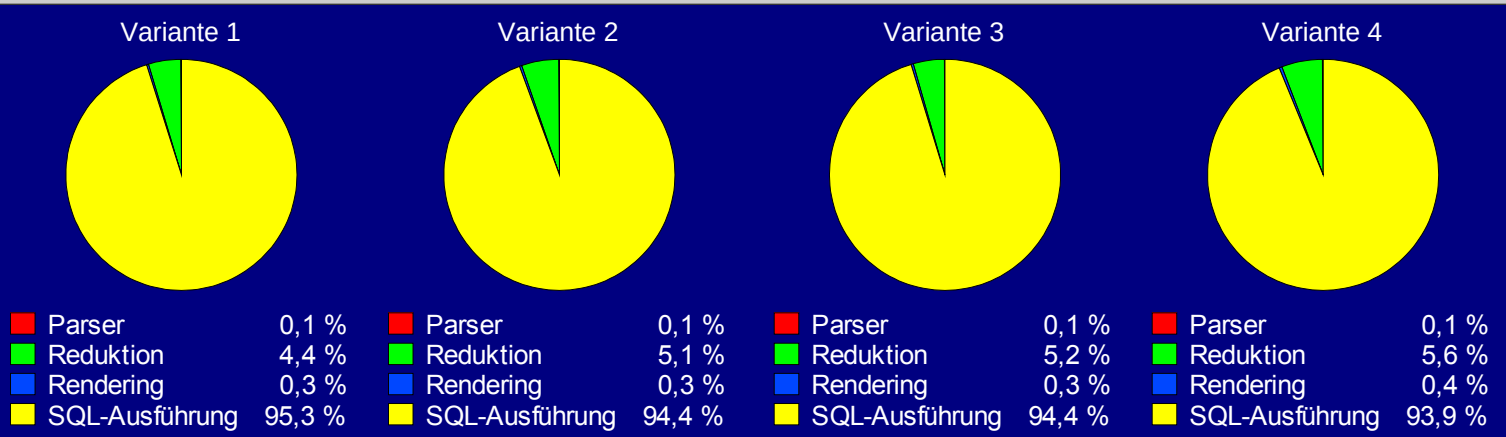


Durchführung der Leistungsuntersuchung

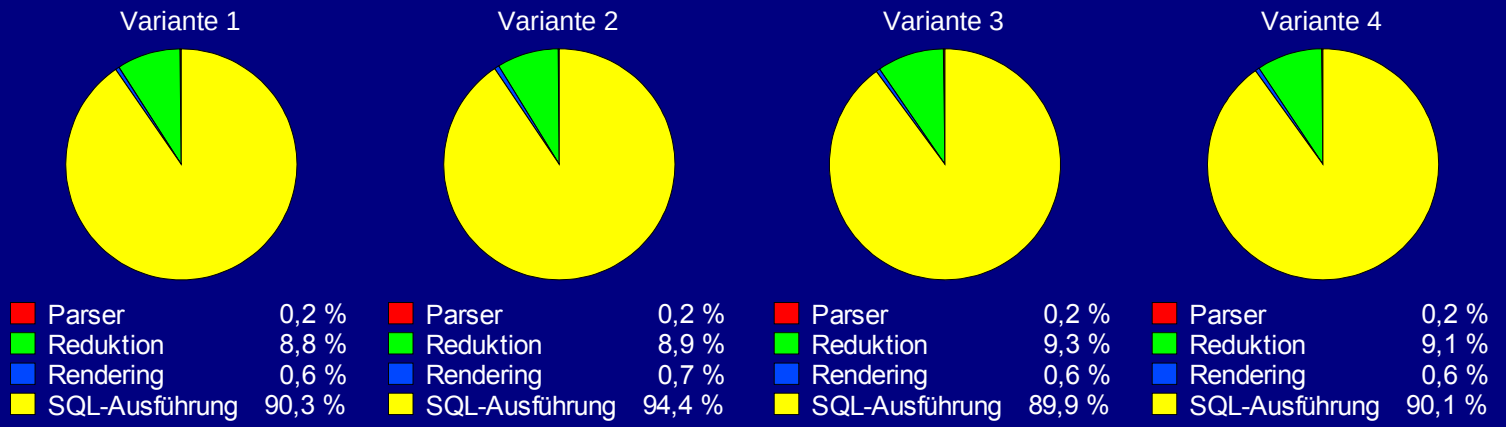
- DS-DML-Statementgenerator
  - sinnvolle Anweisungen
  - Zugriff auf aktuellen Datenbestand
- Messwerte:
  - *Parser-Zeit*
  - *Reduktionszeit*
  - *Rendering-Zeit*
  - *SQL-Ausführungszeit*
  - Anzahl der SQL-Anweisungen

Auswertung: Anteile der verschieden Zeiten

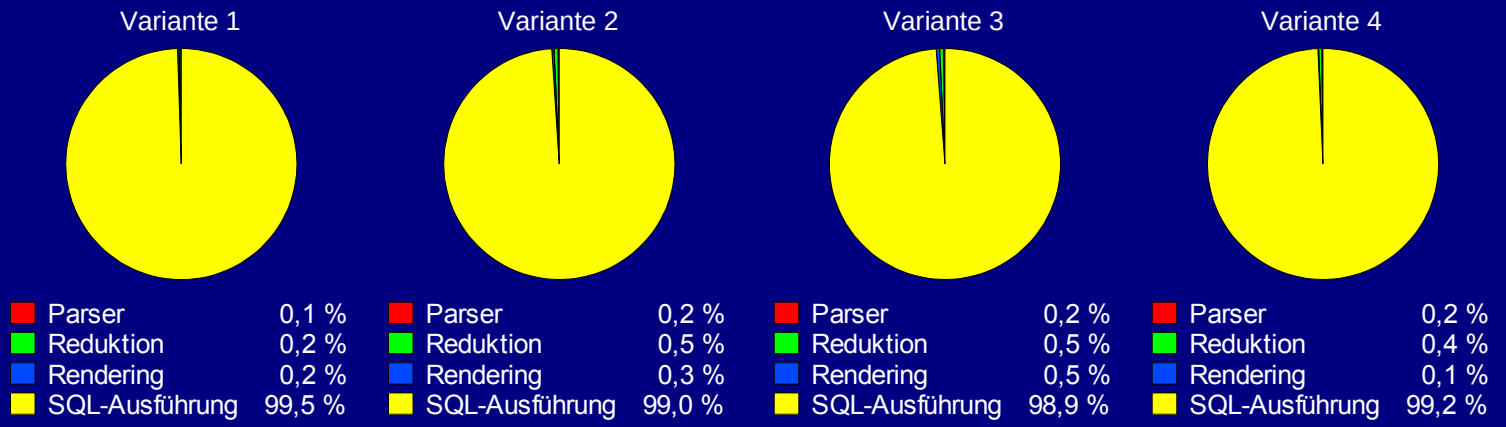
Create Successor



Merge Objects

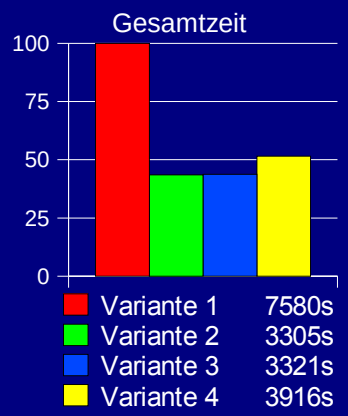


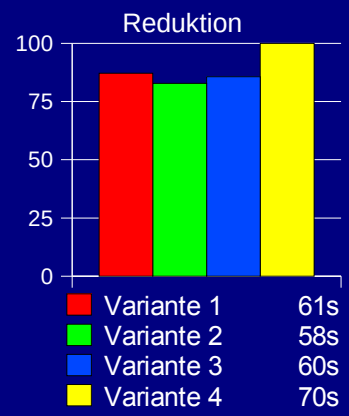
Select

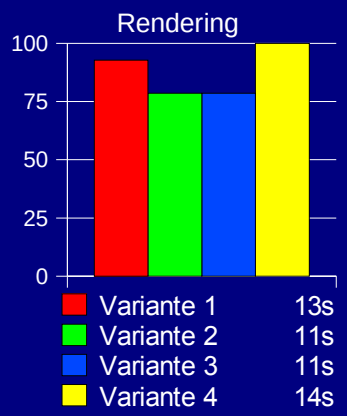


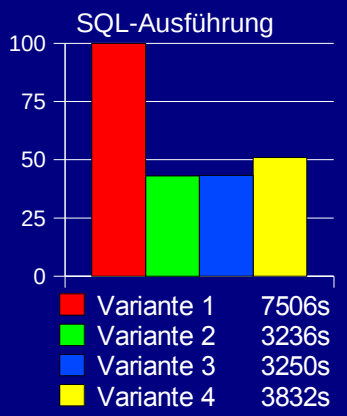
Auswertung: Zeitvergleich

Summe aller Befehle:

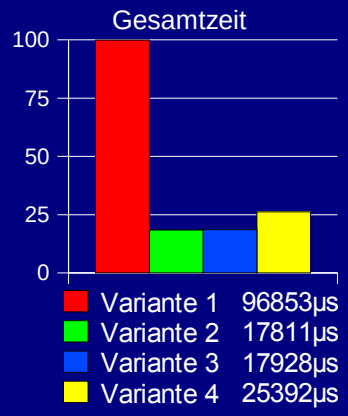


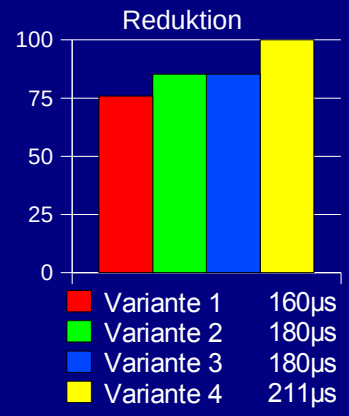


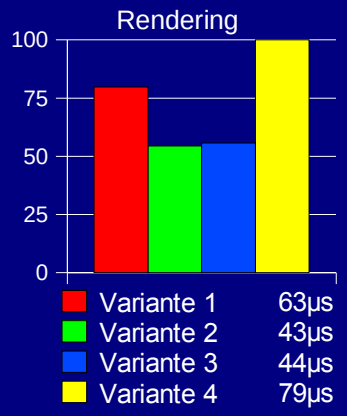


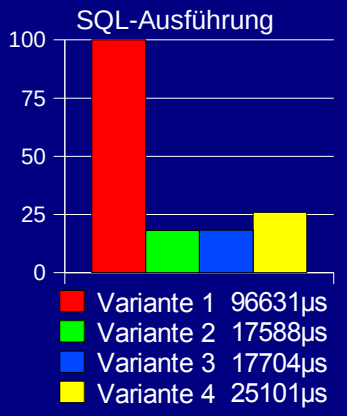


Create New Relationship:

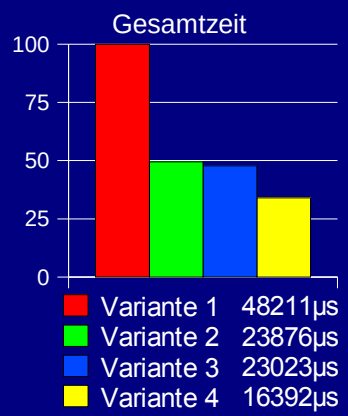


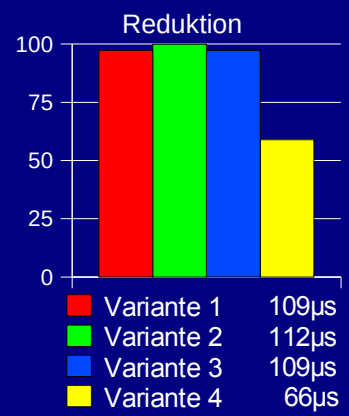


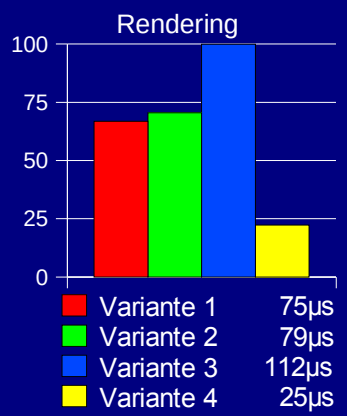


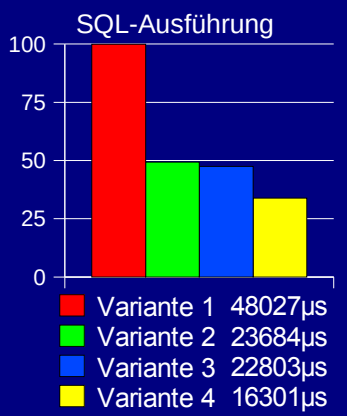


Select:

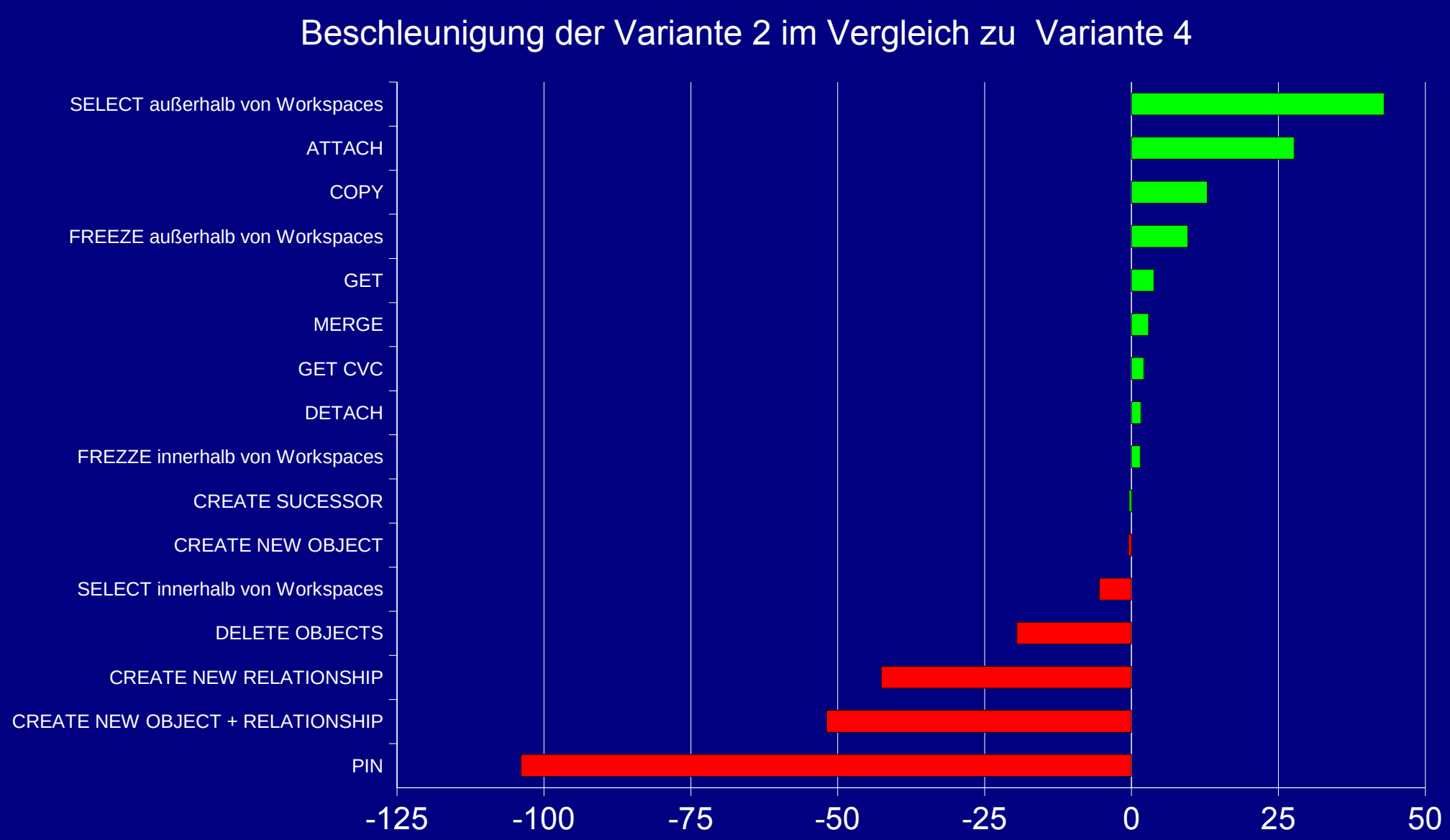






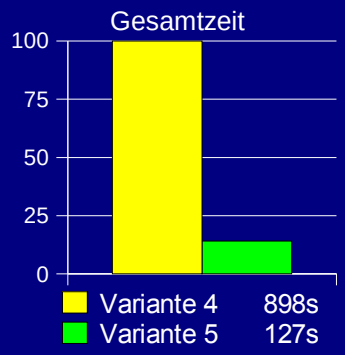


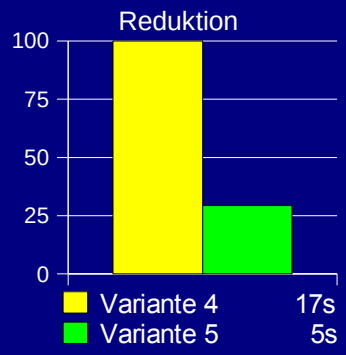
Auswertung: Beschleunigung

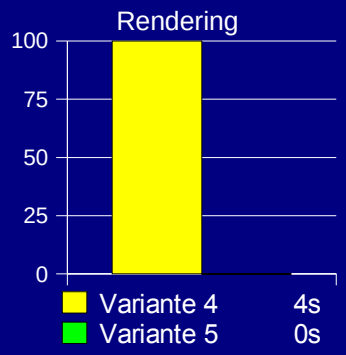


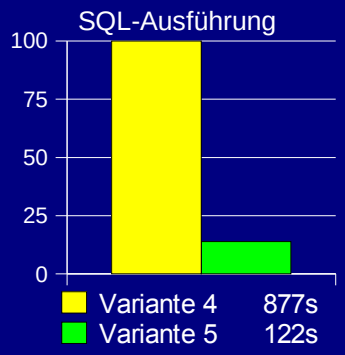
Auswertung: gespeicherte Prozeduren

Summe aller Befehle

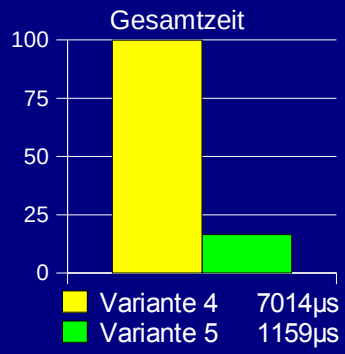


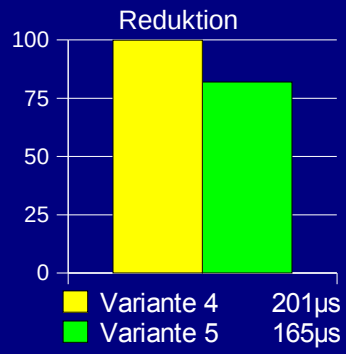


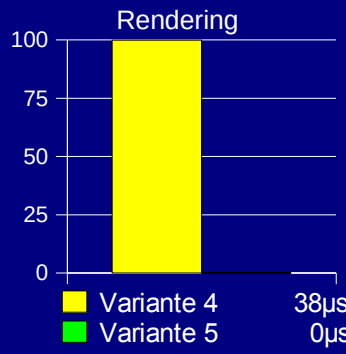


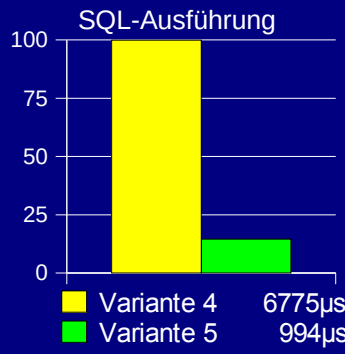


Get









Gliederung

- Anwendungsdomäne: Versionierungssysteme
- Generative Programmierung
- Domänenspezifische Sprachen und Schemaentwurf
- Bewertung
- **Zusammenfassung und Ausblick**

Zusammenfassung

- Erwartungen wurden bestätigt:
  - Aufwand für die Erstellung und Wartung der Reduktionsmethoden steigt (akzeptabel)
    Durchschnittliche zyklomatische Komplexität zwischen 4.0 und 5.8
  - Die Zeiten für die Reduktion steigen
    Durchschnittlich um 27 Prozent
  - SQL-Ausführungszeiten nehmen ab
    Durchschnittlich um 48 Prozent
  - Die für die Reduktion benötigte Zeit ist gering im Vergleich zur SQL-Ausführung
    Durchschnittlich zwischen 1.1 und 2.3 Prozent
  - Operationen können mit Hilfe von gespeicherten Prozeduren beschleunigt werden
    Nur 14 Prozent der Ausführungszeit mit DS-Datenbanktreiber
- Fazit:
  Der Einsatz von domänenspezifischen Spracherweiterungen lohnt sich

Ausblick

- Metasprachen
  - Sprachen zur Spezifikation der Logik der Reduktions- und Renderingmethoden
  - Kein Zugriff auf Informationsmodell durch Polymorphie
- Domänenspezifische Schemaevolution
  - Verwendung existierender Reduktionsmethoden
  - Anpassung der vorhandenen Daten
- Genetische Algorithmen
  - Suche nach einem optimalen Datenbankschema mit hilfe eines vorhandenen Benchmarks
  - Nachbildung der Evolution (Selektion, Paarung, Mutation)

Fragen ?