# Projektaufgabe 3

## Effiziente Berechnung von XPath Achsen (12.5 Punkte)

In der dritten und letzten Projektaufgabe vergleichen wir die Performanz bei der Berechnung von XPath Ausdrücken des EDGE Models mit Varianten des XPath-Accelerators. Als Datensatz verwenden wir einen Ausschnitt der Daten der DBLP. Die DBLP ist die wohl umfangreichste und qualitativ hochwertigste Sammlung an Literaturreferenzen mit Informatikbezug. Kern der Aufgabe ist damit die (effiziente) Verarbeitung von XML-Daten mit unterschiedlichen Techniken und Tools.

**Hinweis.** Es ist hilfreich vor dem Beginn von Phase 1, sich den Benchmark in Phase 3 anzuschauen.

## Phase 1 – Das EDGE Modell (4 P)

In der ersten Phase legen Sie die Basis für das Gesamtprojekt. Dies umfasst:

- **EDGE Modell Import des Toy Beispiels (2 Punkte):** In dieser Aufgabe gebe ich das Toy-Beispiel vor. Sie finden es in der Datei `toy_example.txt`. Die Datei enthält eine Bibliografie aus 4 Publikationen. Sie beruht auf den BibTex-Einträgen der DBLP[1], die im weiteren Verlauf der Aufgabe verwendet werden. Ihre erste Teilaufgabe ist es, dieses Dokument (automatisch) in das Edge Modell zu überführen und die Daten in eine Datenbank zu importieren. Hierbei soll die Struktur der Daten wie folgt geändert werden:
  - Die Publikationen sollen erst nach Publikationsvenue (SIGMOD/VLDB) und dann nach Jahr gruppiert werden. D.h. Wurzelknoten bleibt `bib`. In der ersten Ebene unter der Wurzel hängen nur Konten der Publikationsvenues, direkt darunter Jahreszahlen, dann erst die Publikationen (siehe Abb. 1).
  - Sie können die Attribute `mdate` und `orcid` ignorieren.

  Das Ergebnis als Baum dargestellt, sieht dann aus wie in Abbildung 1.

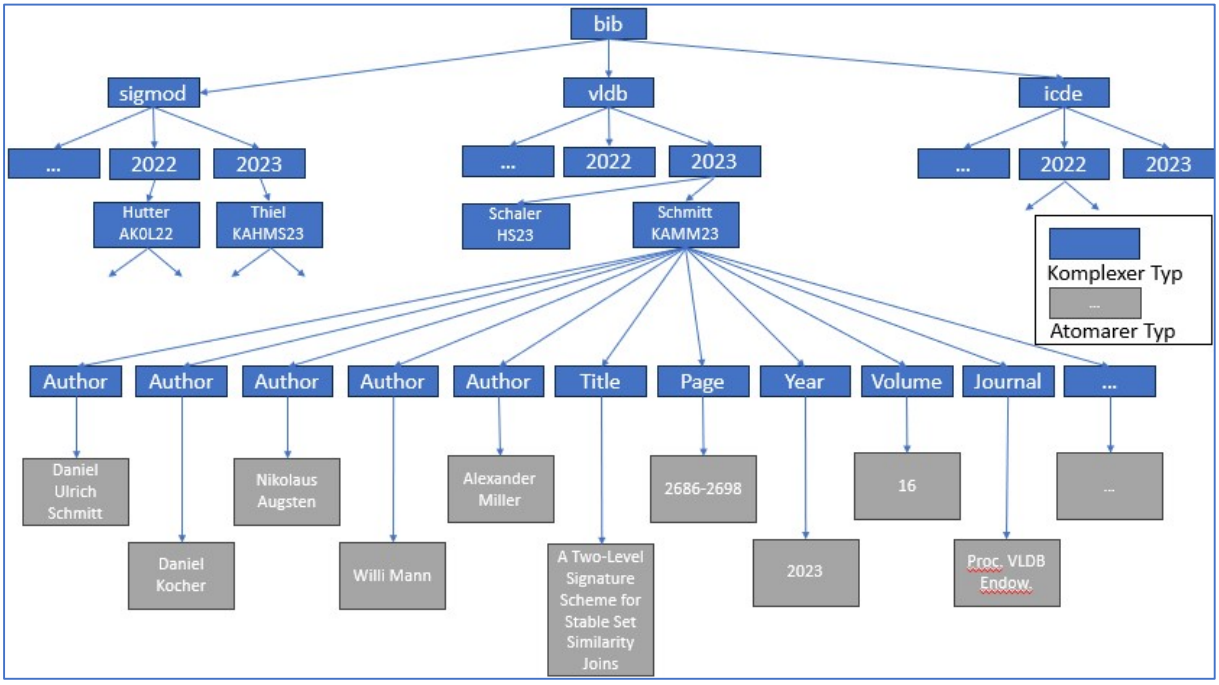


*Abbildung 1 Illustration des Dokuments nach Transformation. Der ICDE-Knoten wird im weiteren Verlauf relevant.*

[1] https://dblp.org/xml/

- Schreiben oder Verwenden Sie hierzu einen XML-Parser in der Programmiersprache ihrer Wahl. Zeigen Sie die Korrekte Funktionsweise, in dem Sie den Datenbestand des Toy- Beispiels durchgehen und auf der Konsole ausgeben. (0.5 P)

  **Hinweis**: Der Parser wird später benötigt um aus dem Gesamtdatenbestand der DBLP-XML (ca. 4GB) alle Publikationen der SIGMOD, VLDB und ICDE zu Filtern und in dieses Format zu überführen. Daher lohnt sich die Suche nach einem guten / schnellen XML-Parser[2]. Bei Problemen mit Sonderzeichen, hilft die mitgelieferte DTD.
- Legen Sie die notwendigen Tabellen des EDGE Modells im DBMS an und beschreiben Sie im Report kurz ihre Interpretation des Edge Models. (0.25 P)
- Befüllen Sie die Tabellen mit den Daten des Toy-Beispiels. Der Datenbestand kann dann so aussehen wie nachfolgend gezeigt (0.75 P).

  **Hinweis**: Unter Umständen lohnt es sich das geparste XML in eine eigene Baumstruktur/Liste zu überführen. Geben Sie der Klasse Node dann die notwendigen Member-Variablen und implementieren Sie eine Methode to_edge_model(), die den Knoten und alle seine ausgehenden Kanten in der DB speichert. Der Datenbankimport kann dann z.B. über eine Traversierung der Struktur erfolgen.

*Tabelle 1 Mögliche Umsetzung des Toy-Beispiels im Edge Modell*

**Node**

| id | s_id | type | content |
|---|---|---|---|
| 0 | bib | bib | null |
| 1 | vldb | venue | null |
| 2 | vldb _2023 | year | null |
| 3 | SchmittKAMM23 | article | null |
| 4 | null | author | Daniel Ulrich Schmitt |
| 5 | null | author | Daniel Kocher |
| 6 | null | author | Nikolaus Augsten |
| 7 | null | author | Willi Mann |
| 8 | null | author | Alexander Miller |
| 9 | null | title | A Two-Level Signature [...] |
| 10 | null | pages | 2686-2698 |
| 11 | null | year | 2023 |
| 12 | null | volume | 16 |
| 13 | null | journal | Proc. VLDB Endow. |
| 14 | null | number | 11 |
| 15 | null | ee | https://doi.org/10.14778/3611479.3611480 |
| 16 | null | url | db/journals/pvldb/pvldb16.html#SchmittKAMM23 |
| 17 | sigmod | venue | |
| 18 | sigmod_2022 | year | null |
| 19 | HutterAK0L22 | inproceedings | null |
| … | … | … | … |

**Edge**

| from | to |
|---|---|
| 0 | 1 |
| 1 | 2 |
| 2 | 3 |
| 3 | 4 |
| 3 | 5 |
| 3 | 6 |
| 3 | 7 |
| 3 | 8 |
| 3 | 9 |
| 3 | 10 |
| 3 | 11 |
| 3 | 12 |
| 3 | 13 |
| 3 | 14 |
| 3 | 15 |
| 3 | 16 |
| 3 | 17 |
| 0 | 17 |
| 17 | 18 |
| 18 | 19 |
| … | … |

[2] Sie können z.B. die SAX API verwenden. Ein Tutorial finden Sie unter Link mit Quellcode.

- **Berechnung der XPath Achsen im EDGE Modell (2 Punkte):** Schreiben Sie für jede der vier XPath-Achsen `ancestor`, `descendant`, `following-sibling`, und `preceding-sibling` eine Funktion, die die entsprechenden Knotenmengen bezogen auf einen Knoten `v` auf des Daten des EDGE Models berechnet.

  - Sie zeigen die Korrektheit am Toy Beispiel in dem Sie die
    i. Für den `Author` Knoten mit Inhalt „Daniel Ulrich Schmitt“ alle `ancestor` Knoten ausgeben (0.75 Punkte).
    ii. Für den Knoten, unter dem alle Publikationen der VLDB 2023 hängen (id=2 in Tabelle 1) alle `descendants` ausgeben (0.75 Punkte).
    iii. Für die Knoten aus Tabelle 1 (oder deren äquivalent in Ihrer EDGE Modell Tabell) mit `s_id` = „SchmittKAMM23“ und `s_id` = „SchalerHS23“ jeweils die `following` und `preceeding` siblings ausgeben (0.5 Punkte).

  Sie können die Funktionen als nutzerdefinierte Funktion (UDF) in der Datenbank umsetzten, oder in der Programmiersprache Ihrer Wahl. Stellen Sie bei letzterer Möglichkeit sicher, dass sämtliche Berechnungen in der Datenbankerfolgen, insbesondere die Rekursion mittels Joins. Verwenden Sie die Funktion Ihrer Programmiersprache im wesentlichen als Controller, der die Rekursion steuert, neue SQL-Anfragen generiert und am Ende das Ergebnis zurückgibt.

  **Hinweis**: Sie dürfen alle Vereinfachungen bzgl. des Datenbestandes annehmen (z.B. zur maximalen Baumhöhe), die sich aus der DTD[3] ableiten lassen.

## Phase 2 – Implementierung des XPath Accelerators (5 P)

In dieser Phase implementieren Sie den XPath accelerator und importieren die Daten für den Benchmark der letzten Phase.

- **Verwendung der Daten aus der DBLP (1 Punkt)**. Laden Sie die die (gepackten) Daten der DBLP als XML herunter[4] und entpacken Sie diese (ca. 4 GB!).
  - Extrahieren Sie alle Daten, die zu den Venues VLDB, SIGMOD und ICDE gehören mittels Ihres Parsers aus Phase 1. Erstellen Sie dann eine neue XML Datei `my_small_bib.xml`, die als Wurzelknoten `bib` enthält unter dem alle Publikationen wie in `toy_example.txt` hängen. Dieses Dokument enthält alle Publikationen des Toy Beispiels. Zeigen Sie die Korrektheit, indem die Publikationen des Toy Beispiels enthalten sind und zählen Sie pro Venue die Publikationen von ‚Nikolaus Augsten‘[5] (0.5 Punkte).

    Sie können dabei davon ausgehen, dass das `key` Attribut eines Artikels (`article`) folgende Semantik enthält.

    - Präfix "journals/pvldb/ " oder "conf/vldb/" spezifizieren eine Publikation auf der VLDB,
    - Präfix "journals/pacmmod/" oder "conf/sigmod/" spezifizieren eine Publikation auf der SIGMOD,
    - Präfix "conf/icde/" spezifiziert eine Publikation auf der ICDE.

[3] https://dblp.org/xml/ Datei dblp.dtd
[4] https://dblp.org/xml/dblp.xml.gz
[5] Zur Kontrolle: https://dblp.org/pid/76/3961.html

- Transformieren Sie die Daten in `my_small_bib.xml` in das Datenformat aus Abb 1. und importieren Sie sie in das EDGE Modell. Geben Sie die Anzahl der Tupel der Relationen `Node` und `Edge` an (0.5 Punkte).

- **Schemaerstellung (1 Punkt).** Erstellen Sie die Relation `accel` mit den Attributen `(pre,post,parent,kind,name)`, sowie die Relationen `content(pre, text)` und `attribute(pre, text)`. Erläutern Sie im Report kurz die Semantik der Tabellen und Attribute.
  **Hinweis**: Sie dürfen das Schema der Relationen Ihrer Namenskonvention und Interpretation des EDGE Modells anpassen. Für die Daten in Tabelle 1 bietet sich z.B. folgendes an: `accel (id,post,s_id,parent,type) und content(id, text)`.
- **Pre-/Post-Order Annotation (1 Punkt).** Schreiben Sie eine Funktion, welche alle Knoten eines Datenbestandes mit der zugehörigen pre und post order Werten annotiert. Es steht Ihnen frei, ob sie dies in der DB auf dem Edge Model, oder in einer Datenstruktur auf dem Client berechnen. Sie zeigen die Korrektheit an einem (Ausschnitt) des Toy Beispiels. Zeichnen Sie hierfür den Baum von Hand und annotieren Sie diesen. Dann zeigen Sie, dass ihre Lösung dasselbe Ergebnis erzeugt.
- **Achse als Fenster (2 Punkt).** Bilden Sie die vier XPath-Achsen `ancestor, descendant, following-sibling`, und `preceding-sibling` auf jeweils ein Fenster entsprechend LV-Folien (Kapitel 2.2 Folie 19) ab. Zeigen Sie die Abbildung für ihr Schema. (0.5 P). Erstellen Sie auf dieser Basis eine SQL-Anfrage, die die zugehörige Achse berechnet. Beachten Sie dabei folgende Vorgaben (0.5 P):
  - Input für die Funktion ist der Schlüssel des Kontextknoten v und die zu berechnende Achse. D.h. insbesondere, dass weitere relevante Information, wie `post(v)` oder `parent(v)` erst aus der DB angefragt werden müssen.
  - Output der Funktion ist eine Liste/Tabelle mit Knoten ids**.**
  - Nehmen Sie noch keine Optimierung zur Verkleinerung des Fensters vor.

  Sie zeigen die Korrektheit der Implementierung analog zu Phase 1 am Toy Beispiel (0.5 P).

## Phase 3 – Optimierungen und Benchmark (3.5 P)

- **Verkleinern der Fensteranfrage (0.5 P).** Implementieren Sie die Optimierungen zur Verkleinerung des Fensters der pre- und post Achse (Folie 25ff). Zeigen Sie, dass das Ergebnis auf dem Toy-Beispiel äquivalent zur Implementierung aus Phase 2 ist.
- **Zugriff mit nur einer Achse (1 Punkt).** Implementieren Sie eine Variante des XPath Accelerators mit nur einer Achse (Folie 30ff). Zeigen Sie die Korrektheit der Annotation am selben Ausschnitt des Toy-Beispiels, wie in Phase 2.
- **Benchmark (1 Punkt).** Nun führen Sie den Benchmark aus und visualisieren Sie die Ergebnisse
  - Verwenden Sie folgende Ansätze:
    - EDGE Model aus Phase 1 mit beliebigen Indexen/Clusterung
    - XPath Accelerator aus Phase 2 optional mit R-Baum index
    - XPath Accelerator aus Phase 3 mit kleinerem Fenster und R-Baum index
    - (nur descendants) XPath Accelerator aus Phase 3 mit einer Achse und clustered B+-Braum

    - Anfragen
        - `Ancestor Achse:` Selektieren Sie als Kontextknoten einen beliebigen `article`-Knoten und Berechnen Sie die Achse
        - `Descendants Achse:` Selektieren Sie als Kontextknoten einen beliebigen `year`-Knoten und Berechnen Sie die Achse
        - `Following` und `preceeding` siblings `Achse:` Selektieren Sie als Kontextknoten einen beliebigen `article`-Knoten, würfeln ob die preceeding oder following siblings bestimmt werden sollen, und Berechnen Sie die Achse.
    - Verwenden Sie folgende Datenbestand:
        - `my_small_bib.xml`
        - Erhöhen Sie den Datenbestand aus my_small_bib.xml um das doppelte, vierfache, 8-fache etc indem Sie mehr Venues beim parsen berücksichtigen (diese können frei gewählt werden). Als Basis gilt die tatsächliche Größe der Datei
- **Vorstellung Ihres Projektes (1 Punkt):** Stellen Sie ihr Projekt in Form eines 8–10-minütigen Vortrags inklusive Live-Demo der wesentlichen Bestandteile vor.
    - Generelles Vorgehen und Setting
    - Zeitplanung & deren Einhaltung
    - Was lief gut, was nicht?
    - Kernergebnis & Fazit

## Bewertung

Sie erhalten Sie oben angegebenen Punkte, wenn zum Ende der Phase vollständig erfüllt sind, sonst erhalten Sie Anteilig Punkte. Kleinere Nacharbeiten und Korrekturen nach Phase 1 werden individuell vereinbart, so dass die Gesamt-Punkte dennoch erreicht werden können. Nacharbeiten nach Phase 2 sind nicht zulässig.

## Zeitplanung

1. Abgabe Phase 1 04.06.
2. Abgabe Phase 2 11.06.
3. Abgabe Phase 3 18.06.