



Übung zur Vorlesung *Einsatz und Realisierung von Datenbanken im SoSe25*

Alice Rey, Maximilian Reif, Tobias Goetz (i3erdb@in.tum.de)

<http://db.in.tum.de/teaching/ss25/impldb/>

Blatt Nr. 10

Hinweise Die Aufgaben können auf <http://xquery.db.in.tum.de/> getestet werden. Die Daten für das Unischema können mit `doc('uni2')` geladen werden. Zur Lösung der Aufgaben können Sie die folgenden XQuery-Funktionen verwenden:

`max(NUM)`, `count(X)`, `tokenize(STR,SEP)`, `sum(NUM)`, `contains(HAY,NEEDLE)`

1. `max(NUMBERS)` - Returns largest number from list
2. `count(LIST)` - Return the number of elements in the list
3. `tokenize(STR,SEP)` - Splits up the string at the separator
4. `sum(NUMBERS)` - Returns sum of all numbers in list
5. `contains(HAY,NEEDLE)` - Checks if the search string (NEEDLE) is contained in the string (HAY)
6. `distinct-values(LIST)` - Returns the distinct values from the list

Hausaufgabe 1

Gegeben eine Tabelle *Produkte* mit folgendem Schema und 10000 Einträgen:

Id (8 Byte) | Name (32 Byte) | Preis (8 Byte) | Anzahl (8 Byte)

Wieviele Daten werden für folgende Queries in die CPU-Caches geladen? Unterscheiden sie jeweils zwischen Row und Column Store.

1. `select * from Produkte`
2. `select Anzahl from Produkte`

Hausaufgabe 2

Sie sollen für die Alexander-Maximilians-Universität (AMU) ein Hauptspeicherdatenbanksystem optimieren. In dem System sind die Daten aller Studenten gespeichert. Schätzen Sie für jede der untenstehenden Anfragen einzeln, ob ein Row- oder Column-Store besser geeignet ist.

Relationen

Studenten: MatrNr (8 Byte), Name (48 Byte), Studiengang (4 Byte), Semester (4 Byte)

MatrNr ist der Primärschlüssel der indiziert ist.

Anfragen:

1. `select * from Studenten;`
2. `select Semester, count(*) from Studenten group by Semester;`
3. `select Name, Studiengang, Semester from Studenten where MatrNr = 42;`

4. `select Studiengang from Studenten where MatrNr = 42;`
5. `select * from Studenten where Semester < 5;`
6. `select * from Studenten where Semester = 25;`
7. `insert into studenten values(4242, Max Meyer, Info, 7);`

Hausaufgabe 3

In (pseudo) C++ kann eine 'Row-Store-artige' Datenstruktur wie folgt angelegt werden:

```
struct Tuple {
    int MatrNr;
    RuntimeString Name;
    int Semester;
}
Tuple data[10000] = {};
```

Notieren Sie, wie die Daten in Form eines Column Stores gehalten werden können in (pseudo) C++.

Erklären Sie Ihrem Tutor, welche Vor- und Nachteile Row- und Column Stores jeweils haben. Was würden Sie für Amazons Webseite verwenden? Was verwenden Sie für die Controlling Datenbank?

Hausaufgabe 4

Formulieren Sie die zuvor in SQL bearbeiteten Anfragen zur Universitätsdatenbank in XQuery. Erstellen Sie insbesondere XQuery-Anfragen, um folgende Fragestellungen zu beantworten:

- a) Suchen Sie die Professoren, die Vorlesungen halten.
- b) Finden Sie die Studenten, die alle Vorlesungen gehört haben.
- c) Finden Sie die Studenten mit der größten Semesterzahl unter Verwendung von Aggregatfunktionen.
- d) Berechnen Sie die Gesamtzahl der Semesterwochenstunden, die die einzelnen Professoren erbringen. Dabei sollen auch die Professoren berücksichtigt werden, die keine Vorlesungen halten.
- e) Finden Sie die Studenten, die alle vierstündigen Vorlesungen gehört haben.
- f) Finden Sie die Namen der Studenten, die in keiner Prüfung eine bessere Note als 3.0 hatten.
- g) Berechnen Sie den Umfang des Prüfungsstoffes jedes Studenten. Es sollen der Name des Studenten und die Summe der Semesterwochenstunden der Prüfungsvorlesungen ausgegeben werden.
- h) Finden Sie Studenten, deren Namen den eines Professors enthalten.
- i) Ermitteln Sie den Bekanntheitsgrad der Professoren unter den Studenten, wobei wir annehmen, dass Studenten die Professoren nur durch Vorlesungen oder Prüfungen kennen lernen.

Hausaufgabe 5

Schreiben Sie eine Anfrage, die folgendes Ergebnis zurückgibt:

```

<Universitaet>
  <Fakultaet Name="Philosophie" AnzahlAssistenten="3">
    <Professor Name="Sokrates" AnzahlAssistenten="2"/>
    <Professor Name="Russel" AnzahlAssistenten="1"/>
  </Fakultaet>
  <Fakultaet Name="Physik" AnzahlAssistenten="2">
    <Professor Name="Kopernikus" AnzahlAssistenten="2"/>
  </Fakultaet>
  <Fakultaet Name="Theologie" AnzahlAssistenten="1">
    <Professor Name="Augustinus" AnzahlAssistenten="1"/>
  </Fakultaet>
</Universitaet>

```

Hausaufgabe 6

Beantworten Sie folgende Anfragen mithilfe von JSONpath über das nachfolgende JSON-Dokument. Zum Auswerten der Anfragen können Sie folgende Schnittstelle verwenden: <https://jsonpath.com/>

```

{
  "shop": {
    "name": "Tech_&_More",
    "categories": [
      {
        "name": "Laptops",
        "products": [
          { "id": 101, "name": "Laptop_A", "price": 899, "tags": ["ultrabook", "ssd"] },
          { "id": 102, "name": "Laptop_B", "price": 1299, "tags": ["gaming", "16GB"] }
        ]
      },
      {
        "name": "Phones",
        "products": [
          { "id": 201, "name": "Phone_X", "price": 699, "tags": ["5G", "oled"] },
          { "id": 202, "name": "Phone_Y", "price": 499, "tags": ["budget", "4G"] },
          { "id": 203, "name": "Phone_Z", "price": 999, "tags": ["flagship", "5G"] }
        ]
      }
    ]
  }
}

```

1. Geben Sie den Namen aller Produkte aus
2. Geben Sie die ersten zwei Produkte aus der Kategorie „Phones“ aus
3. Geben Sie alle Preise aus
4. Geben Sie alle Produkt-Tags aus von Produkten mit einem Preis < 1000€
5. Geben Sie alle Produkte aus mit einem 5G Tag