

Projektaufgabe 2

Phase 2 – Legen der Basis (2.5 P)

Datenmanagement jenseits von Relationen

Gruppen Nummer 8

Weilert Alexander, 12119653

Jovanovic Dragana, 11850325

May 21, 2024

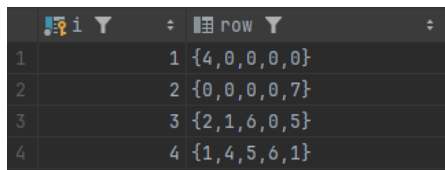
Dieses Reporting Template dient der Vorbereitung der Abgabe von Phase 2.

Alternativer Import für Ansatz 2 (0.5 Punkte)

Zeigen Sie das create table Statement für A oder B .

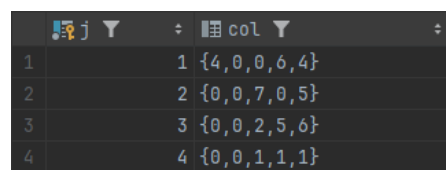
```
CREATE TABLE new_A (i INT PRIMARY KEY, row INT[])
```

Zeigen Sie, wie die Matrizen A und B des Toy Examples für Ansatz 2 in der DB aussehen.



| i | row |
|---|-------------|
| 1 | {4,0,0,0,0} |
| 2 | {0,0,0,0,7} |
| 3 | {2,1,6,0,5} |
| 4 | {1,4,5,6,1} |

Figure 1: Tabelle A



| j | col |
|---|-------------|
| 1 | {4,0,0,6,4} |
| 2 | {0,0,7,0,5} |
| 3 | {0,0,2,5,6} |
| 4 | {0,0,1,1,1} |

Figure 2: Tabelle B

Ansatz 2 (0.5 Punkte):

Geben Sie den Code der Funktion `dotproduct()` bzw. Ihrer Lösung für Ansatz 2 an.

```
public void ansatz2() {
    try (Statement statement = this.connection.createStatement()) {
        statement.execute("DROP TABLE IF EXISTS new_C");
        statement.execute("CREATE TABLE new_C ( " +
            + "i INT, j INT, val INT, PRIMARY KEY (i, j))");
        createFunction();
        statement.execute("INSERT INTO new_C (i, j, val) " +
            + "SELECT new_A.i, new_B.j, dotproduct(new_A.row, new_B.col) " +
            + "FROM new_A, new_B " +
            + "WHERE dotproduct(new_A.row, new_B.col) != 0");
    } catch (SQLException e) {
        throw new RuntimeException(e);
    }
}
```

```

public void createFunction() {
    try (Statement statement = this.connection.createStatement()) {
        statement.execute("DROP FUNCTION IF EXISTS dotproduct(int[], int[])");
        statement.execute("CREATE OR REPLACE FUNCTION " +
            "dotproduct(vector1 int[], vector2 int[]) RETURNS int AS $$\n" +
            "DECLARE\n" +
            "    result int := 0;\n" +
            "BEGIN\n" +
            "    FOR i IN 1..array_length(vector1, 1) LOOP\n" +
            "        result := result + vector1[i] * vector2[i];\n" +
            "    END LOOP;\n" +
            "    RETURN result;\n" +
            "END;\n" +
            "$$ LANGUAGE plpgsql;");
    } catch (SQLException e) {
        throw new RuntimeException(e);
    }
}

```

Zeigen Sie für das Toy Example, dass C korrekt berechnet wird (z.B. via Screenshot).

Wird korrekt berechnet, da die Ergebnisse übereinstimmen. Siehe Screenshots.

| i | j | val | |
|----|---|-----|----|
| 1 | 1 | 1 | 16 |
| 2 | 1 | 1 | 28 |
| 3 | 2 | 2 | 35 |
| 4 | 2 | 3 | 42 |
| 5 | 2 | 4 | 7 |
| 6 | 3 | 1 | 28 |
| 7 | 3 | 2 | 67 |
| 8 | 3 | 3 | 42 |
| 9 | 3 | 4 | 11 |
| 10 | 4 | 1 | 44 |
| 11 | 4 | 2 | 40 |
| 12 | 4 | 3 | 46 |
| 13 | 4 | 4 | 12 |

Figure 3: Tabelle newC nach dem Ansatz2

| i | j | sum | |
|----|---|-----|----|
| 1 | 1 | 1 | 16 |
| 2 | 1 | 1 | 28 |
| 3 | 2 | 2 | 35 |
| 4 | 2 | 3 | 42 |
| 5 | 2 | 4 | 7 |
| 6 | 3 | 1 | 28 |
| 7 | 3 | 2 | 67 |
| 8 | 3 | 3 | 42 |
| 9 | 3 | 4 | 11 |
| 10 | 4 | 1 | 44 |
| 11 | 4 | 2 | 40 |
| 12 | 4 | 3 | 46 |
| 13 | 4 | 4 | 12 |

Figure 4: View C aus Ansatz1 zum Vergleich

Benchmark Definition (0.5 P):

| Parameter | Interval | Kommentar |
|-----------|--------------------|----------------------|
| L | $2 \leq n \leq 10$ | $2^{(n+1)}$ |
| S | $0 \leq n \leq 8$ | $((n+1) \times 0.1)$ |

Table 1: Getestet Matrixgrößen L und sparsity Werte S

Speicherkapazitäten wurden berücksichtigt und Sparsity wurde mit niedrigen Werte getestet.

Auswertung (0.5 P)

Stellen Sie Ihre Messergebnisse grafisch dar.

| L | Sparsity | Matrix | Array | time in ms | Calc_0 | Calc_1 | Calc_2 |
|--------|----------|--------|-------|------------|--------|--------|--------|
| 2^3 | 0.5 | | 16 | 38 | 60001 | 11150 | 217409 |
| 2^4 | 0.5 | | 13 | 35 | 60005 | 7613 | 168847 |
| 2^5 | 0.5 | | 16 | 39 | 60001 | 4555 | 32822 |
| 2^6 | 0.5 | | 27 | 63 | 60012 | 2672 | 5320 |
| 2^7 | 0.5 | | 127 | 65 | 60017 | 1060 | 685 |
| 2^8 | 0.5 | | 206 | 331 | 60381 | 255 | 85 |
| 2^9 | 0.5 | | 916 | 1259 | 60377 | 57 | 7 |
| 2^10 | 0.5 | | 4204 | 5546 | 60063 | 12 | 1 |
| | | | | | | | |
| | | | | | | | |
| L | Sparsity | Matrix | Array | time in ms | Calc_0 | Calc_1 | Calc_2 |
| 10 0.1 | | | 19 | 45 | 60001 | 7770 | 86315 |
| 10 0.2 | | | 15 | 33 | 60001 | 7762 | 96711 |
| 10 0.3 | | | 13 | 32 | 60001 | 7858 | 114706 |
| 10 0.4 | | | 13 | 30 | 60001 | 7766 | 144902 |
| 10 0.5 | | | 13 | 31 | 60002 | 7812 | 167380 |
| 10 0.6 | | | 13 | 31 | 60002 | 7797 | 191930 |
| 10 0.7 | | | 13 | 31 | 60002 | 7858 | 238197 |
| 10 0.8 | | | 12 | 29 | 60002 | 7786 | 233777 |
| 10 0.9 | | | 13 | 30 | 60005 | 7822 | 274409 |

Figure 5: Messdaten

Benchmark Laufzeit:

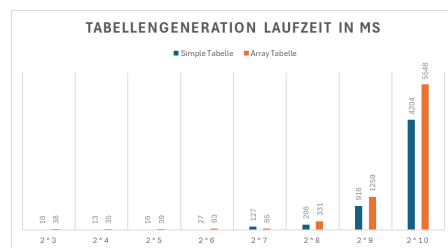


Figure 6: L wird erhöht

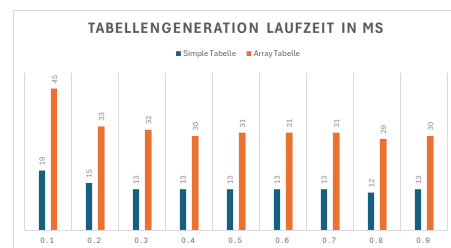


Figure 7: Sparsity wird erhöht

Benchmark Anzahl max. Iterationen innerhalb 1er Minute:

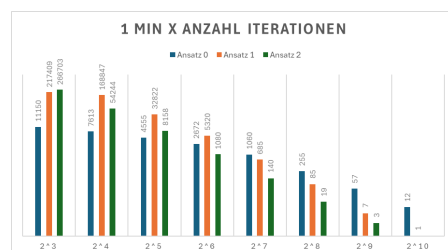


Figure 8: Erhöhung von L

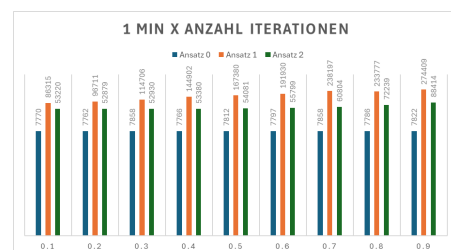


Figure 9: Erhöhung von Sparsity

Zeitmanagement

Benötigte Zeit pro Person (nur Phase 1): 6h

References

Important: Reference your information sources!
Remove this section if you use footnotes to reference your information sources.
