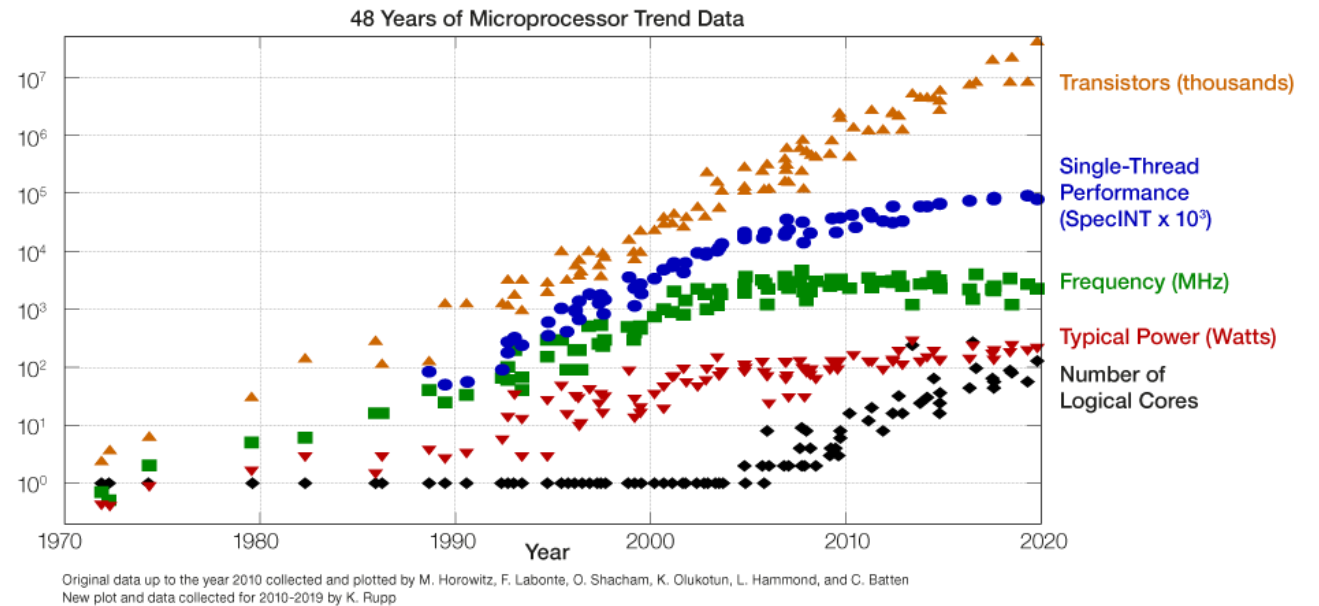
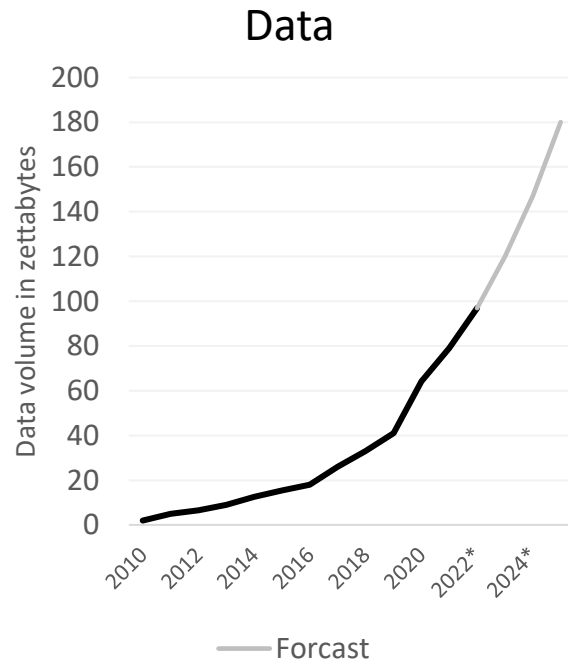


Query Processing on Tensor Computation Runtimes

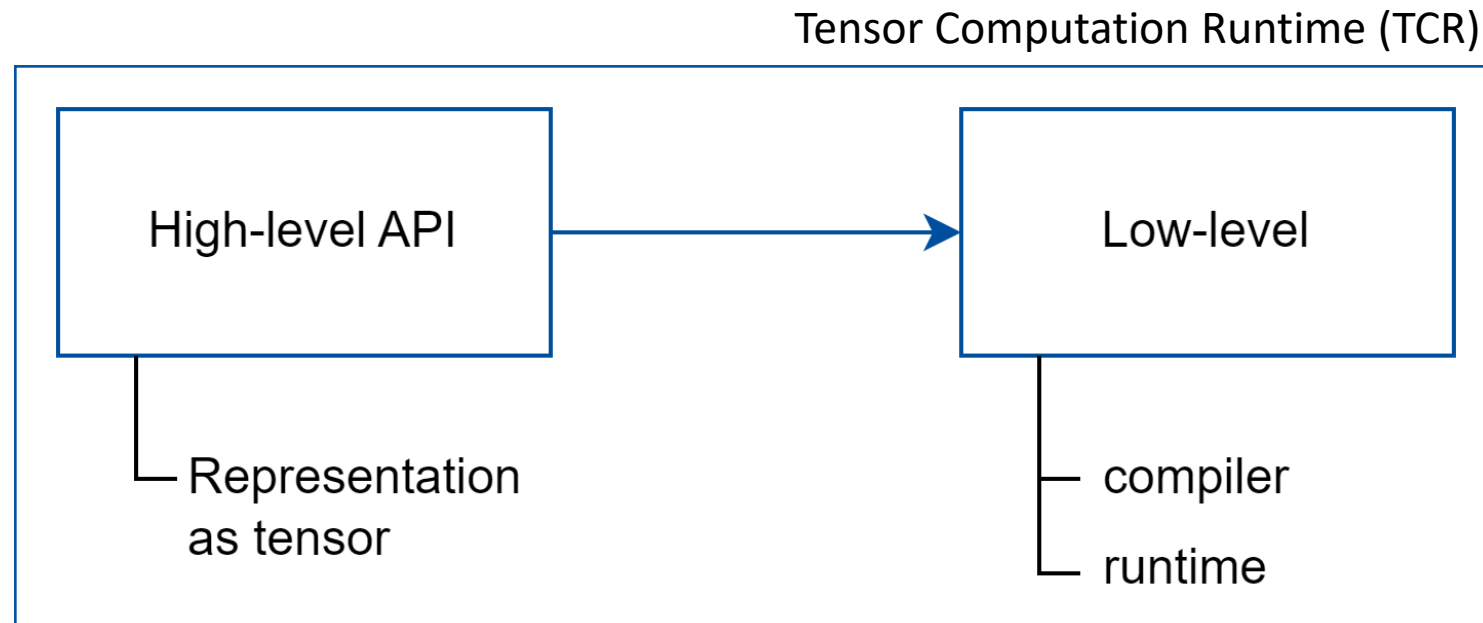
Outline

- More data – more analysis
- Challenges
- Tensor query processor
- Operator implementation in TQP
- Evaluation

More data - more analysis



Calculating on GPU



Calculating on GPU



1d-tensor



2d-tensor



3d-tensor



4d-tensor



5d-tensor



6d-tensor

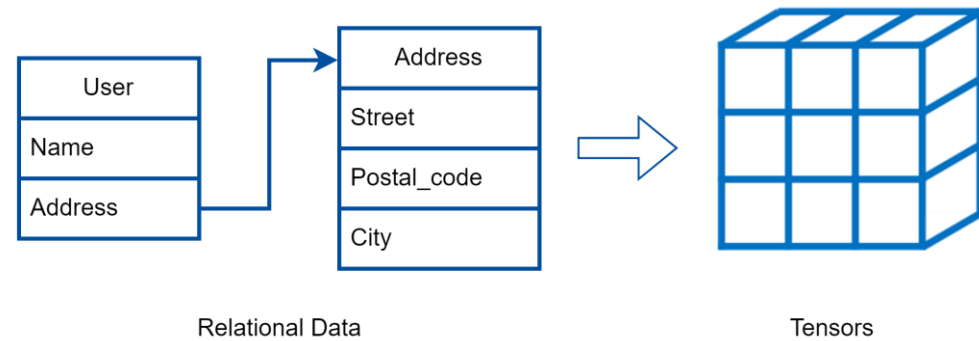
Challenges

- Expressivity

```
WITH data_series AS (  
  SELECT  
    RANK() OVER (ORDER BY day) AS row_number,  
    day,  
    day - RANK() OVER (ORDER BY day) AS series_id  
  FROM user_registration )  
SELECT  
  MIN(day) AS series_start_day,  
  MAX(day) AS series_end_day,  
  MAX(day) - MIN (day) + 1 AS series_length  
FROM data_series  
GROUP BY series_id  
ORDER BY series_start_date
```

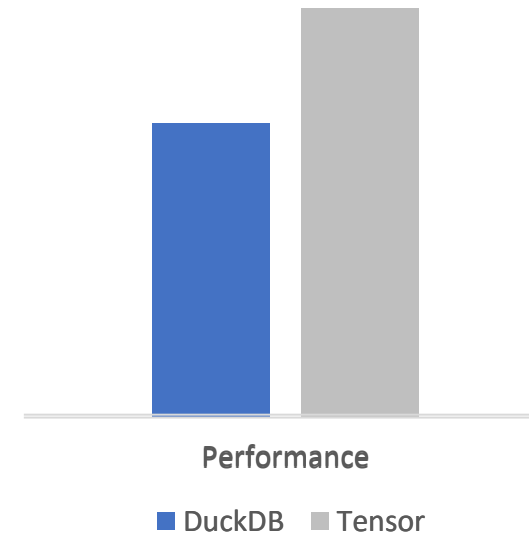
Challenges

- Expressivity
- Data representation



Challenges

- Expressivity
- Data representation
- Performance

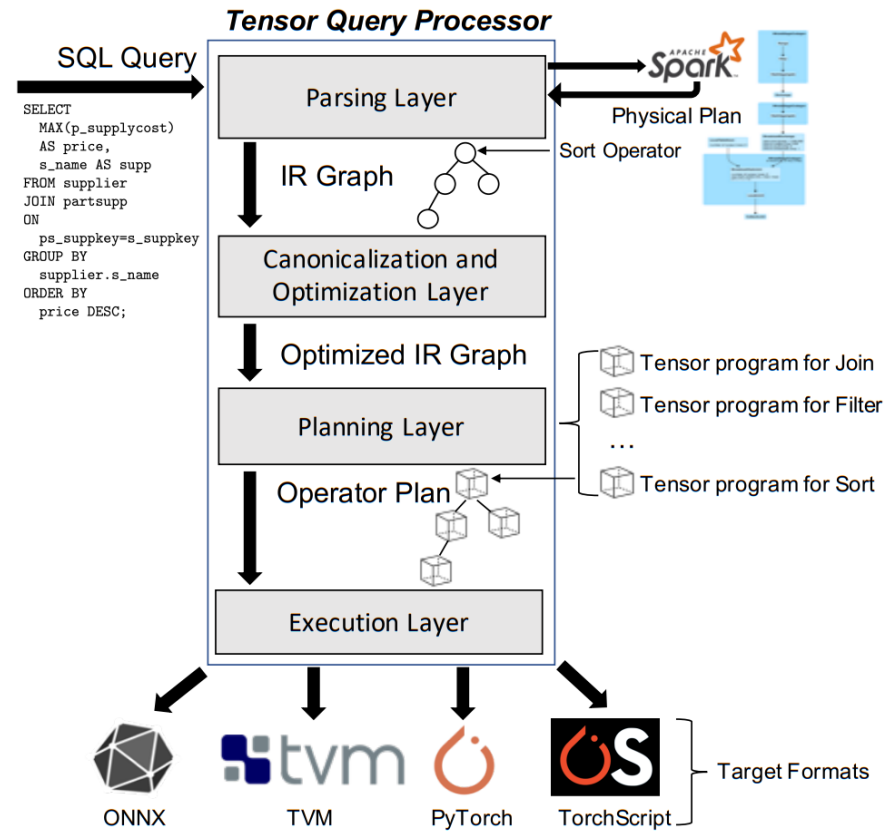


Challenges

- Expressivity
- Data representation
- Performance
- Extensibility



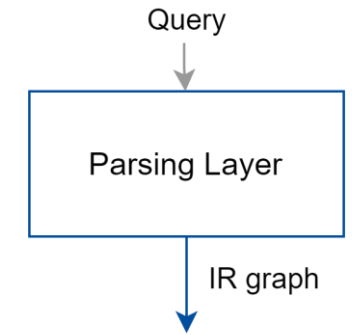
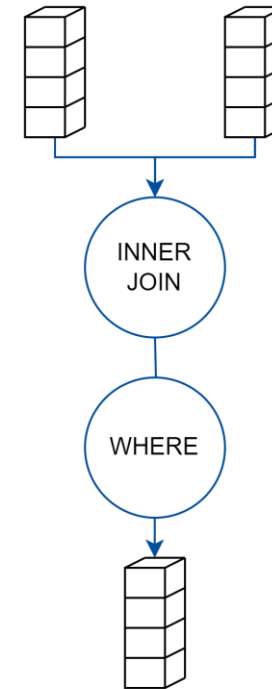
Tensor Query Processor



Tensor Query Processor

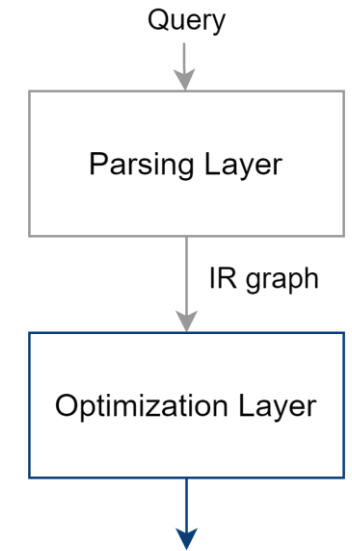
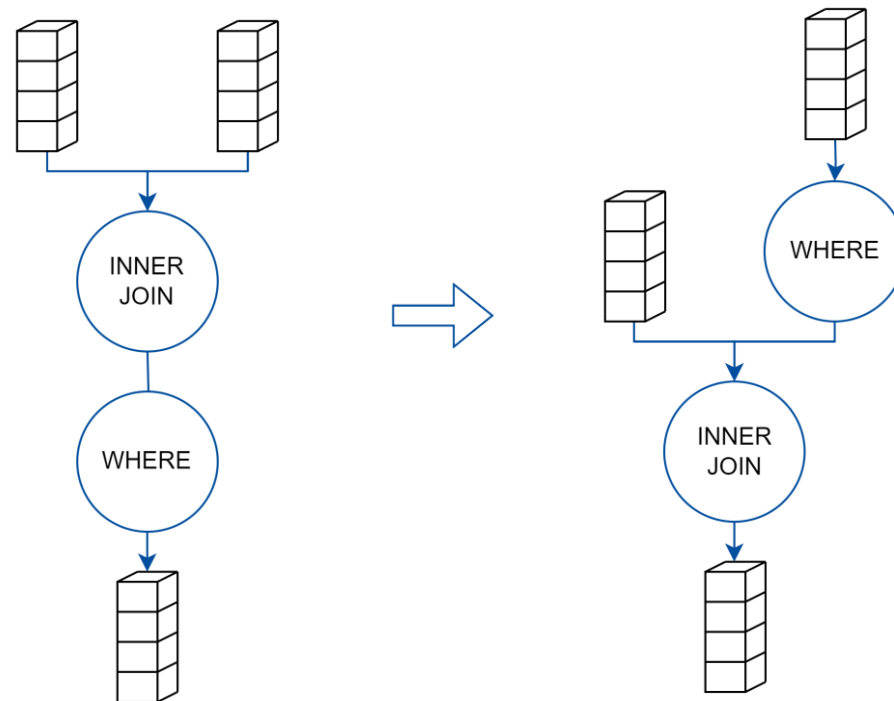
Parsing layer

```
SELECT o.OrderID, c.CustomerName, o.OrderDate  
FROM Orders AS o  
INNER JOIN Customers AS c  
  ON o.CustomerID=c.CustomerID  
WHERE c.Budget > 500;
```



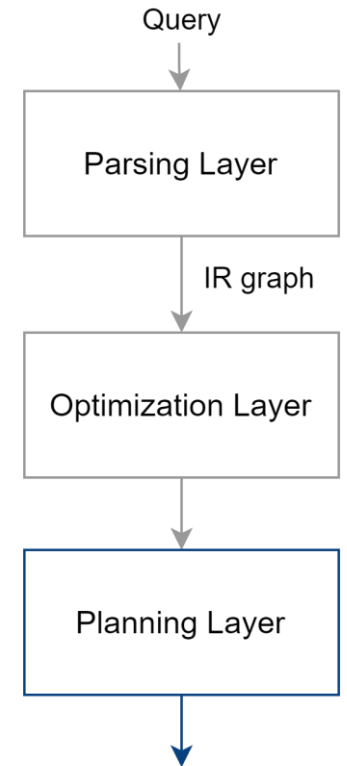
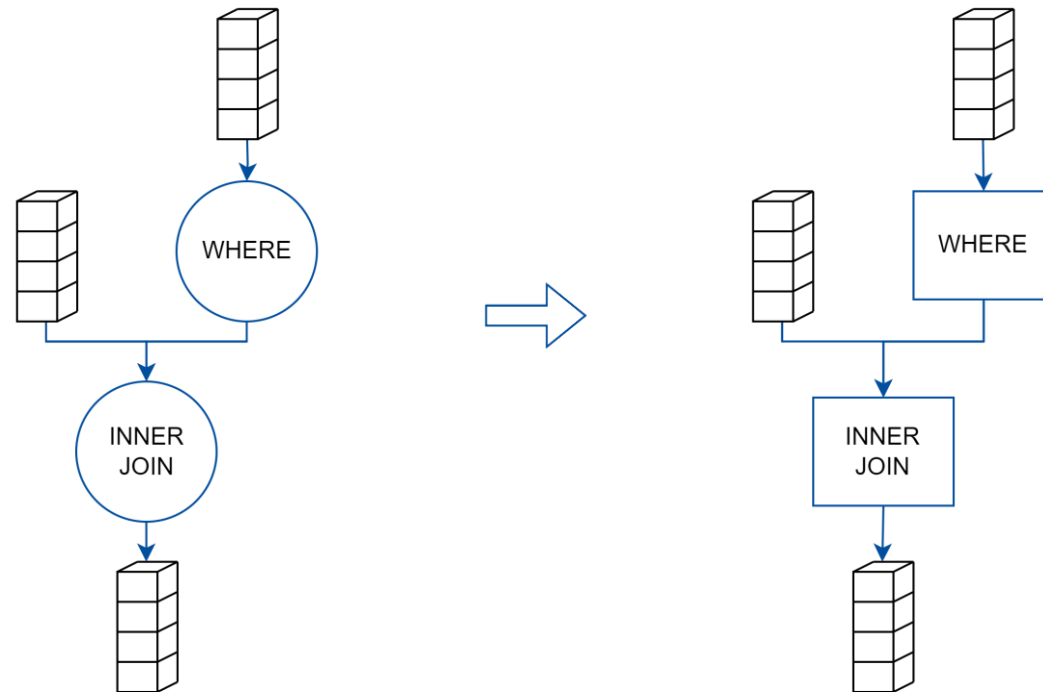
Tensor Query Processor

Optimization layer



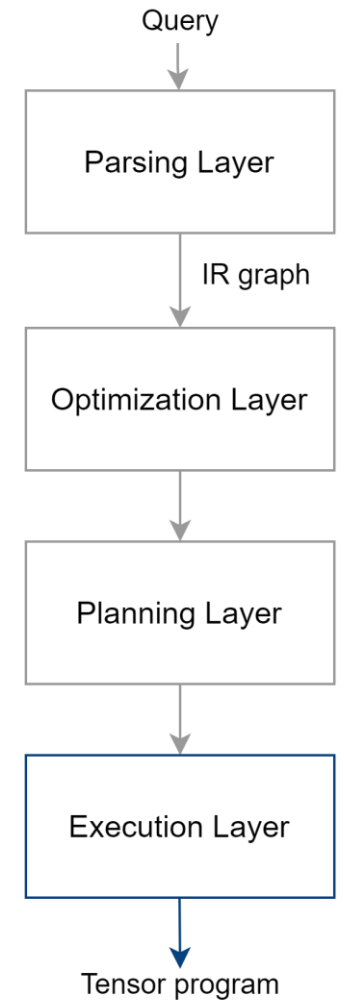
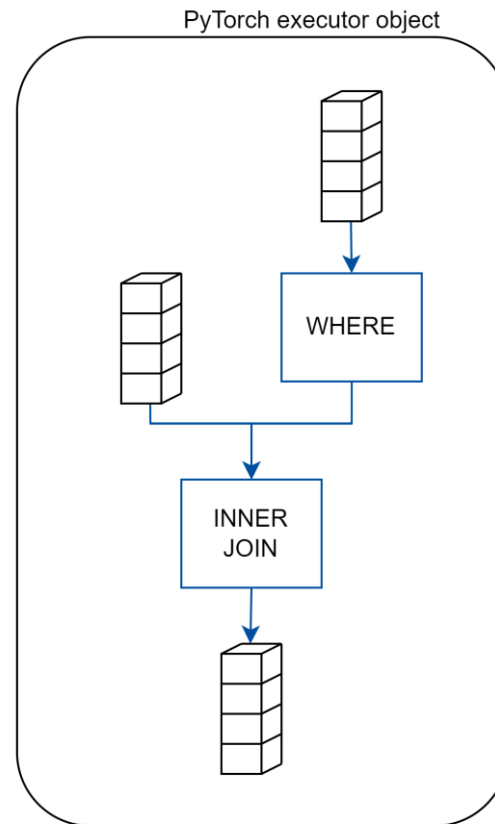
Tensor Query Processor

Planning layer



Tensor Query Processor

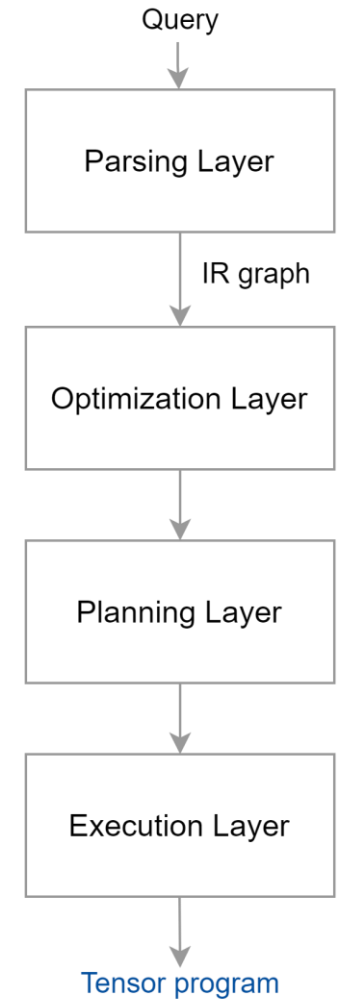
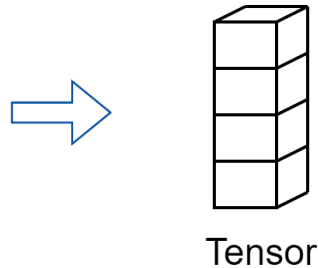
Execution layer



Tensor Query Processor

Query execution

- Converting data into tensors
- Data movement
- Scheduling of operators



Operator implementation in TQP

Filter with expressions

```
SELECT EmpName FROM Employee  
WHERE EmpId < 20 AND EmpSalary > 2000
```

```
torch.logical_and(  
    torch.lt(c_empid < 20),  
    torch.gt(c_empsalary, 2000))
```



```
torch.masked_select(...)
```

EmpId	EmpSalary	Bitmap
2	4000	✓
4	2500	✓
3	50	✗
4	2500	✓
5	600	✗

Evaluation

Query	CPU (1 core)			GPU		
	DuckDB	TQP	TQPJ	Omnisci	TQP	TQPJ
Q1	0.664	7.535	7.301	0.095	0.027	0.026
Q2	0.101	0.629	0.577	0.351	0.039	0.028
Q3	0.273	1.154	1.165	0.293	0.027	0.024
Q4	0.216	1.050	1.087	0.292	0.020	0.018
Q5	0.302	2.459	2.963	0.064	0.048	0.042
Q6	0.156	0.143	0.073	0.047	0.003	0.002
Q7	0.430	2.236	1.931	0.067	0.042	0.035
Q8	0.278	2.460	2.503	0.079	0.050	0.039
Q9	2.533	4.518	4.616	0.072	0.105	0.092
Q10	0.430	1.168	1.184	0.740	0.057	0.052
Q11	0.034	0.476	0.324	0.084	0.016	0.009
Q12	0.309	0.976	0.966	0.062	0.025	0.021
Q13	0.181	9.379	9.197	0.069	0.153	0.136
Q14	0.171	0.124	0.096	N/A	0.007	0.005
Q15	0.291	0.133	N/A	0.086	0.129	N/A
Q16	0.093	3.664	3.699	3.689	0.320	0.301
Q17	0.381	2.303	2.466	0.132	0.061	0.051
Q18	0.765	2.245	2.406	0.593	0.053	0.048
Q19	0.419	1.577	1.316	0.058	0.042	0.036
Q20	0.276	2.032	1.975	N/A	0.048	0.041
Q21	0.932	25.49	24.25	N/A	0.158	0.151
Q22	0.069	0.315	0.296	N/A	0.011	0.010

Thanks for watching

References:

Paper:

Dong He, Supun C Nakandala, Dalitso Banda, Rathijit Sen, Karla Saur, Kwanghyun Park, Carlo Curino, Jesús Camacho-Rodríguez, Konstantinos Karanasos, and Matteo Interlandi. 2022. Query processing on tensor computation runtimes. Proceedings of the VLDB Endowment 15, 11 (jul 2022), 2811–2825. <https://doi.org/10.14778/3551793.3551833>

Additional information's:

- <https://www.statista.com/statistics/871513/worldwide-data-created/>

Images:

- Henrik Bilar (2021) MAN <https://www.man.com/single-core-stagnation-and-the-cloud>
- Maja Zalac (2021) <https://visagetechologies.com/tensors-in-computer-vision/>
- Francois Chollet (2017) Twitter <https://twitter.com/fchollet/status/871089784898310144>
- https://kr.freepik.com/free-vector/illustration-of-jigsaw-icon_2606569.htm
- Some images are created by the above-mentioned paper „Query processing on tensor computation runtimes“