HPI Hasso Plattner Institut
Digital Engineering · Universität Potsdam

Task 1 Presentation - ASE 2024/25

Robert Richter

26.11.2024

Reflection Task 1 - Implementing the Query

Own Implementation:

+ The performance was better compared to ChatGPT's variant (30s vs. 90s)
+ More correct variant (⇒ ChatGPT did not realize a person is *not* foreign to themself)
- It took quite a while to develop the code
- Bad readability

ChatGPT's Implementation:

+ Quick working solution (⇒ Copy-pasting task was nearly sufficient)
+ Perfectly Readable
- Worse Performance (not relevant for smaller one-time queries..)

Reflection Task 2 - Enhancing the Query

Own Implementation:

+ Still better performance (45s vs. 120s)
- High effort (≈ an hour of work)
- Code became messy

ChatGPT's Implementation:

+ No knowledge required (again simply copy-pasting of new task)
+ Highly readable
- Less performance

Reflection Task 3 - DBeaver Query Planning

Knotentyp	Entität	Kosten	Reihen	Zeit	Bedingung
Sort		46770350.11 - 46770350.1	154		
Nested Loop		4715204.77 - 46770350.10	154		
Hash Join		4715204.77 - 46770295.71	154		(foreignlikes.l_messageid = m_1.m_messageid)
Subquery Scan		4703936.19 - 46759027.12	4476		(foreignlikes.rn = 1)
WindowAgg		4703936.19 - 46759027.11	4476		((row_number() over (?)) <= 1)
WindowAgg		4703936.19 - 46759027.09	57521		
Nested Loop		4703936.19 - 46759027.08	57521		((k2.k_person1id is null)
Gather Merge		4703936.19 - 46746181.38	57521		
Incremental Sort		4702936.17 - 46745178.47	19174		
Nested Loop		31575.95 - 46745178.02	19174		
Merge Join		31575.95 - 33450.02	36480		
Sort		18370.49 - 18669.18	95553		
Parallel Seq Sca	message	0.00 - 6250.77	95581		
Materialize		13205.41 - 13752.61	101849		
Sort		13205.41 - 13479.01	101849		
Seq Scan	likes	0.00 - 1798.40	109440		
Seq Scan	knows	0.00 - 461.46	18188		
Materialize		0.00 - 602.19	28146		
Seq Scan	knows	0.00 - 461.46	28146		
Hash		11241.44 - 11241.44	345		
Aggregate		11160.04 - 11241.44	345		(count(l_1.l_personid) >= 20)
Hash Join		8853.56 - 11127.48	21623		(l_1.l_messageid = m_1.m_messageid)
Seq Scan	likes	0.00 - 1798.40	109440		
Hash		8640.30 - 8640.30	17445		
Seq Scan	message	0.00 - 8640.30	17445		(m_length > 100)
Seq Scan	person	0.00 - 35.28	1528		

Reflection Task 3 - ChatGPT's Query Planning

```
ID      Operation       Table   Index/Join Condition    Filter  Rows (Estimate)
1       Sort                    ORDER BY (foreign_likes / total_likes) DESC     1000
2       Nested Loop Join                                1000
3       ├── Nested Loop Join                            5000
4       │ ├── Filtered Aggregate         message m_length >= 100         2000
5       │ │ ├── Join     likes   m.m_messageid = l.l_messageid           50000
6       │ │ │ └── Filtered Join person  Exclude friends of m_creatorid  Based on knows subquery 10000
7       │ └── Hash Join total_likes     l.mid = r.mid   total_likes >= 20 and foreign_likes * 2 >= total_likes  5000
8       ├── Nested Loop Join    likes l2        r.mid = l2.l_messageid and r.min_date = l2.l_creationdate
9       └── Hash Join   person p2       r2.l_personid = p2.p_personid           1000
```

- Badly formatted
- Not clear what it means
- Precision? Less Operators, other results than in DBeaver ...
- We need knowledge to understand this ⇒ not simple anymore ...

Questions?