Snapshot Semantics for Temporal Multiset Relations

Anton Dignös[1], Boris Glavic[2], Xing Niu[2], Michael Böhlen[3], Johann Gamper[1]

[1]Free University of Bozen-Bolzano, [2]Illinois Institute of Technology, [3]University of Zurich

Freie Universität Bozen
Libera Università di Bolzano
Università Liedia de Bulsan

ILLINOIS INSTITUTE OF TECHNOLOGY

University of Zurich UZH

Goal

Goal: Provably correct framework for relational algebra operators with snapshot semantics

Desiderata:

- Support multiset relations
- Snapshot reducibility
- Uniqueness of encoding
- Implementation on SQL:2011 period relations

Problems: Aggregate Gap and Bag Difference Bug

name	skill	period
Ann	SP	[03, 10)
Joe	NS	[08, 16)
Sam	SP	[08, 16)
Ann	SP	[18, 20)

Aggregation: *How many workers are in the factory at any given time?*

```sql
SELECT count(*)
FROM workers;
```

cnt	period
...	...
2	[10, 16)
0	[16, 18)
1	[18, 20)
...	...

AG bug: No results for gaps!

mach	skill	period
M1	SP	[03, 12)
M2	SP	[06, 14)
M3	NS	[03, 16)

Difference: *How many workers are missing to operate all machines?*

```sql
SELECT skill FROM assign
EXCEPT ALL
SELECT skill FROM works;
```

skill	period
SP	[06, 08)
SP	[10, 12)
NS	[03, 08)

BD bug: No result for bag difference if tuple exists on right-hand side!

State-of-the-art

Approach	Bag	AG bug free	BD bug free	Unique encoding
Interval preservation [2] (ATSQL)	✓	×	×	×
Teradata [10]	✓	×	N/A	×[a]
Change preservation [3, 4]	×	×	N/A	×
TSQL2 [5, 6, 8]	×	N/A	N/A	✓
ATSQL2 [1]	✓	N/A	×	×
TimeDB [9] (ATSQL2)	✓	N/A	×	×
SQL/Temporal [7]	✓	×	×	×
SQL/TP [11][b]	✓	✓	✓	×
Our approach	✓	✓	✓	✓

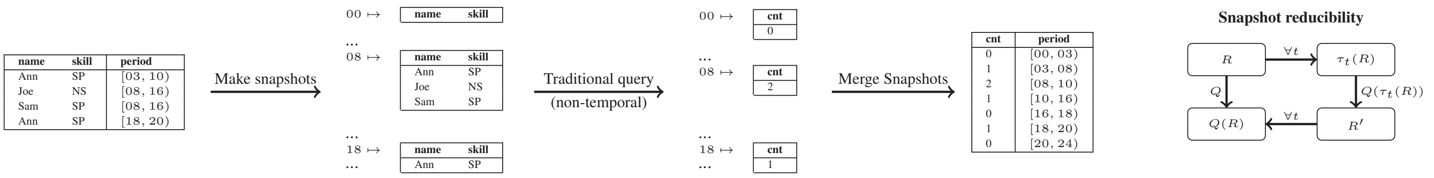
[a]Optionally, coalescing (NORMALIZE ON in Teradata) can be applied to get a unique encoding at the cost of loosing multiplicities.
[b]Sequenced semantics can be expressed, but this is inefficient

Background on Temporal Snapshot Semantics

- The result of a **temporal operator** is defined by the result of its **corresponding nontemporal operator** applied on each snapshot

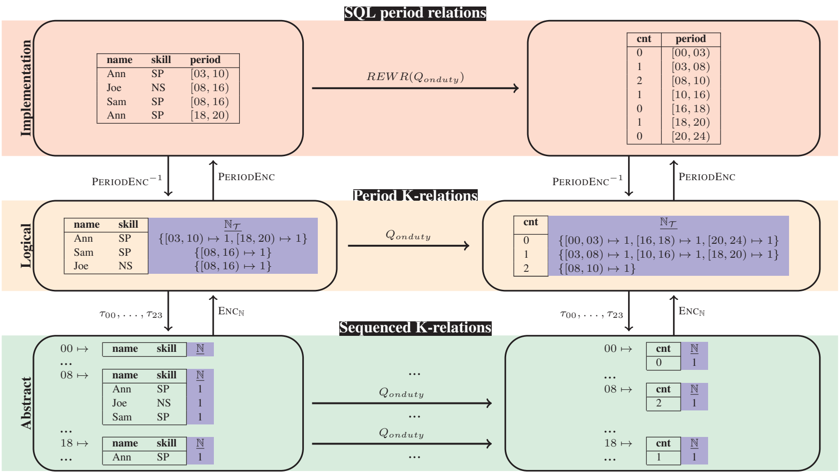
Q_{onduty}: *How many specialized workers (skill=SP) are in the company at each given time?*

```sql
SELECT count(*) FROM works WHERE skill = 'SP';
```



- **Most approaches** assume duplicate free snapshot relations (**set semantics**).

Approach Overview



- Standard SQL period relations
- Query rewriting to standard SQL
- Period semirings (**multisets**)
- Multiset coalescing (**uniqueness**)
- Snapshot semirings for set and multisets (**snapshot reducibility**)

References

[1] M. H. Böhlen, C. S. Jensen, and R. T. Snodgrass. Evaluating and enhancing the completeness of tsql2. Technical Report TR 95-5, Computer Science Department, University of Arizona, 1995.

[2] M. H. Böhlen, C. S. Jensen, and R. T. Snodgrass. Temporal statement modifiers. *ACM Trans. Database Syst.*, 25(4):407–456, 2000.

[3] A. Dignös, M. H. Böhlen, and J. Gamper. Temporal alignment. In *SIGMOD*, pages 433–444, 2012.

[4] A. Dignös, M. H. Böhlen, J. Gamper, and C. S. Jensen. Extending the kernel of a relational DBMS with comprehensive support for sequenced temporal queries. *ACM Trans. Database Syst.*, 41(4):26:1–26:46, 2016.

[5] R. T. Snodgrass, editor. *The TSQL2 Temporal Query Language*. 1995.

[6] R. T. Snodgrass, I. Ahn, G. Ariav, D. S. Batory, J. Clifford, C. E. Dyreson, R. Elmasri, F. Grandi, C. S. Jensen, W. Käfer, N. Kline, K. G. Kulkarni, T. Y. C. Leung, N. A. Lorentzos, J. F. Roddick, A. Segev, M. D. Soo, and S. M. Sripada. TSQL2 language specification. *SIGMOD Record*, 23(1):65–86, 1994.

[7] R. T. Snodgrass, M. H. Böhlen, C. S. Jensen, and A. Steiner. Adding valid time to sql/temporal. *ANSI X3H2-96-501r2, ISO/IEC JTC*, 1, 1996.

[8] M. D. Soo, C. S. Jensen, and R. T. Snodgrass. An algebra for tsql2. In *The TSQL2 temporal query language*, pages 505–546. 1995.

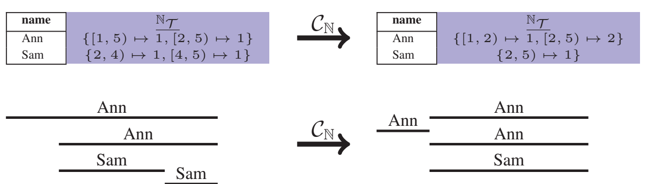
[9] A. Steiner. *A generalisation approach to temporal data models and their implementations*. PhD thesis, ETH Zurich, 1998.

[10] Teradata. Teradata database - temporal table support. http://www.info.teradata.com/download.cfm?ItemID=1006923, Jun 2015.

[11] D. Toman. Point-based temporal extensions of SQL and their efficient implementation. In *Temporal databases: research and practice*, pages 211–237. 1998.

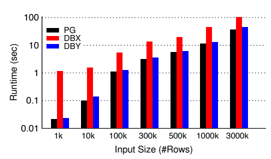
Generalized Coalescing

- Ensures uniqueness of encoding
- Merge tuples over snapshots with identical values and annotations into periods
- **Multiset coalescing**: Merge tuples over snapshots with identical values and cardinalities



Implementation and Experimental Results

- Middleware implementation based on query rewriting into standard SQL
- Optimizations: pre-aggregation and elimination of coalescing steps



Multiset coalescing

TPC-BiH (SF10 = ∼ 10GB)				
Query	PG-Seq	PG-Nat	DBY-Seq	Bug
Q1	63.85	TO (2h)	82.61	
Q5	5.85	1794.10	14.89	
Q6	7.70	126.91	7.28	AG
Q7	28.70	1642.20	21.75	
Q8	21.78	1484.61	17.33	
Q9	129.01	TO (2h)	71.37	
Q12	10.49	264.57	13.30	
Q14	26.55	3436.30	23.79	AG
Q19	9.60	2873.13	22.35	AG

Comparison with native implementation

Presented at: *45th International Conference on Very Large Data Bases*, Los Angeles, California, USA, August 26-30, 2019.