Query Time Scaling of Attribute Values in Temporal Databases

University of Zurich[UZH]

Anton Dignös, Michael Böhlen, and Johann Gamper

{adignoes, boehlen}@ifi.uzh.ch, gamper@inf.unibz.it

Freie Universität Bozen
Libera Università di Bolzano
Free University of Bozen · Bolzano

Goal and Requirements

Goal: Scaling of attribute values when temporal operators modify attached timestamps.

For instance, attribute values that record total quantities over time often must be scaled if the associated timestamp changes.

Example: A budget of 181K during [Feb, Aug) corresponds to a budget of 89K during [Feb, May).

Application Requirements: Applications have the choice to specify **whether** and **how** to scale attribute values at query time.

Solution

Solution:

- An algebra that performs **adjustment** of timestamps automatically and allows to **scale** attribute values.
- User-defined functions that scale attribute values based on:
  - the value x to be scaled;
  - the new and old timestamp

Challenge:

- Old values are required for scaling
- Old timestamps are required for scaling
- New timestamps are required for scaling

Example

Input: Project N with budget B in department D during time T. **Query:** How does the available budget per department change?

p

N	B	D	T
X	181K	DB	[Feb, Aug)
Y	184K	DB	[May, Aug)
Z	153K	AI	[Apr, Sep)

$_D\vartheta^T_{SUM(scale(B))}(\mathbf{p})$

Result

D	SUM	T
DB	89K	[Feb, May)
DB	276K	[May, Aug)
AI	153K	[Apr, Sep)

$scale(181\text{K}, [\text{Feb}, \text{May}), [\text{Feb}, \text{Aug})) = 89\text{K}$

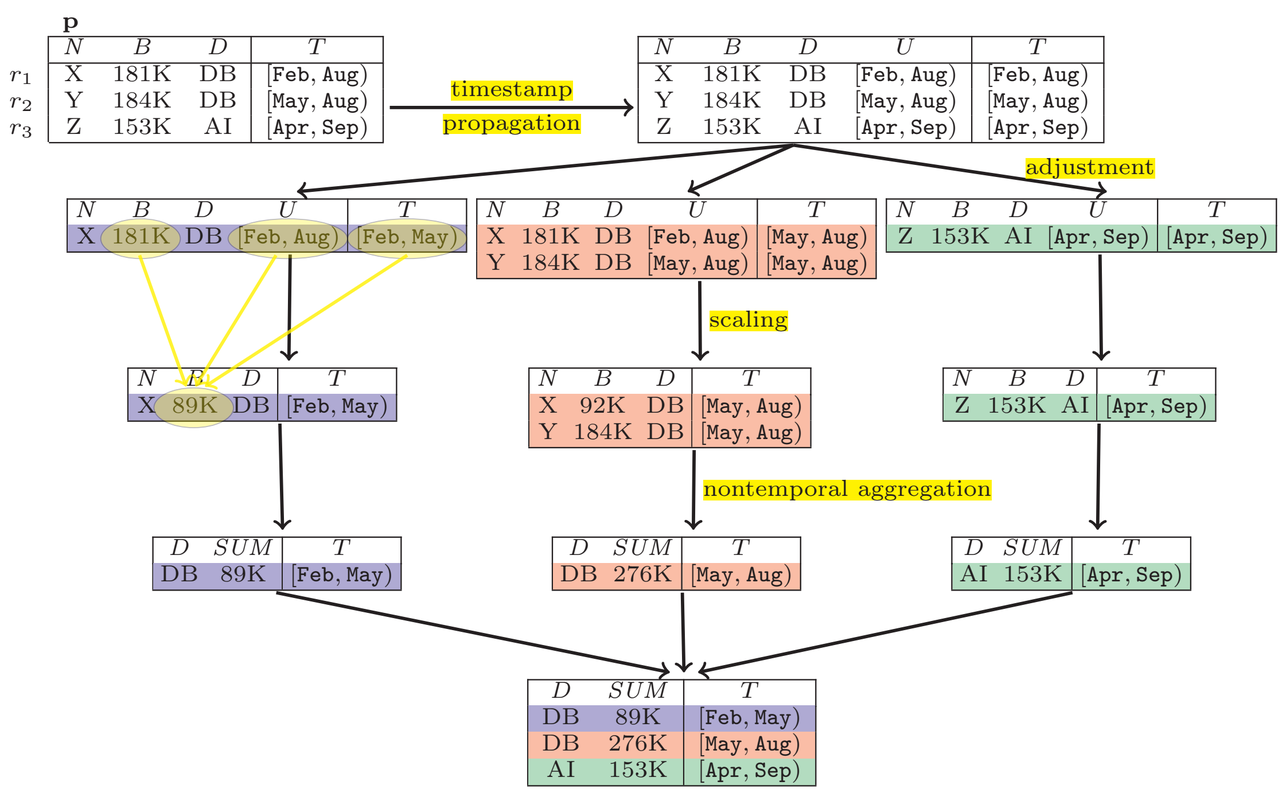
Query: $_D\vartheta^T_{SUM(scale(B))}(\mathbf{p})$

$_D\vartheta^T_{SUM(scale(B))}$
|
p

is transformed to:

1. Timestamp propagation: $\mathbf{p_1} \leftarrow \epsilon_U(\mathbf{p})$
2. Temporal adjustment: $\mathbf{p_2} \leftarrow \mathcal{N}_{\mathbf{x}.D=\mathbf{y}.D}(\mathbf{p_1}/\mathbf{x}, \mathbf{p_1}/\mathbf{y})$
3. Scaling: $\mathbf{p_3} \leftarrow \pi_{D,N,scale(B,T,U)/B,T}(\mathbf{p_2})$
4. Nontemporal aggregation: $\mathbf{q} \leftarrow {}_{D,T}\vartheta_{SUM(B)}(\mathbf{p3})$

$_{D,T}\vartheta_{SUM(B)}$
|
$\pi_{D,N,scale(B,T,U)/B,T}$
|
$\mathcal{N}_{\mathbf{x}.D=\mathbf{y}.D}$
|
ϵ_U | ϵ_U
|
$\mathbf{p/x}$ | $\mathbf{p/y}$



Background

Temporal Algebra:

Operator		Reduction
$\sigma^T_\theta(\mathbf{r})$	=	$\sigma_\theta(\mathbf{r})$
$\pi^T_{\mathbf{B}}(\mathbf{r})$	=	$\pi_{\mathbf{B},T}(\mathcal{N}_{\mathbf{r.B}=\mathbf{s.B}}(\mathbf{r}, \mathbf{r/s}))$
${}_{\mathbf{B}}\vartheta^T_F(\mathbf{r})$	=	${}_{\mathbf{B},T}\vartheta_F(\mathcal{N}_{\mathbf{r.B}=\mathbf{s.B}}(\mathbf{r}, \mathbf{r/s}))$
$\mathbf{r} -^T \mathbf{s}$	=	$\mathcal{N}_{\mathbf{r.A}=\mathbf{s.A}}(\mathbf{r}, \mathbf{s}) - \mathcal{N}_{\mathbf{r.A}=\mathbf{s.A}}(\mathbf{s}, \mathbf{r})$
$\mathbf{r} \cup^T \mathbf{s}$	=	$\mathcal{N}_{\mathbf{r.A}=\mathbf{s.A}}(\mathbf{r}, \mathbf{s}) \cup \mathcal{N}_{\mathbf{r.A}=\mathbf{s.A}}(\mathbf{s}, \mathbf{r})$
$\mathbf{r} \cap^T \mathbf{s}$	=	$\mathcal{N}_{\mathbf{r.A}=\mathbf{s.A}}(\mathbf{r}, \mathbf{s}) \cap \mathcal{N}_{\mathbf{r.A}=\mathbf{s.A}}(\mathbf{s}, \mathbf{r})$
$\mathbf{r} \times^T \mathbf{s}$	=	$\alpha(\phi_\top(\mathbf{r}, \mathbf{s}) \bowtie_{\mathbf{r}.T=\mathbf{s}.T} \phi_\top(\mathbf{s}, \mathbf{r}))$
$\mathbf{r} \bowtie^T_\theta \mathbf{s}$	=	$\alpha(\phi_\theta(\mathbf{r}, \mathbf{s}) \bowtie_{\theta \wedge \mathbf{r}.T=\mathbf{s}.T} \phi_\theta(\mathbf{s}, \mathbf{r}))$
$\mathbf{r} ⟕^T_\theta \mathbf{s}$	=	$\alpha(\phi_\theta(\mathbf{r}, \mathbf{s}) ⟕_{\theta \wedge \mathbf{r}.T=\mathbf{s}.T} \phi_\theta(\mathbf{s}, \mathbf{r}))$
$\mathbf{r} ⟖^T_\theta \mathbf{s}$	=	$\alpha(\phi_\theta(\mathbf{r}, \mathbf{s}) ⟖_{\theta \wedge \mathbf{r}.T=\mathbf{s}.T} \phi_\theta(\mathbf{s}, \mathbf{r}))$
$\mathbf{r} ⟗^T_\theta \mathbf{s}$	=	$\alpha(\phi_\theta(\mathbf{r}, \mathbf{s}) ⟗_{\theta \wedge \mathbf{r}.T=\mathbf{s}.T} \phi_\theta(\mathbf{s}, \mathbf{r}))$
$\mathbf{r} \triangleright^T_\theta \mathbf{s}$	=	$\phi_\theta(\mathbf{r}, \mathbf{s}) \triangleright_{\theta \wedge \mathbf{r}.T=\mathbf{s}.T} \phi_\theta(\mathbf{s}, \mathbf{r})$

SQL mapping:

Operator		SQL
$\epsilon_U(\mathbf{r})$	:	`SELECT Ts Us, Te Ue, * FROM r`
$\mathcal{N}_\theta(\mathbf{r}, \mathbf{s})$	:	`FROM (r NORMALIZE s ON` θ`) r`
$\phi_\theta(\mathbf{r}, \mathbf{s})$	:	`FROM (r ALIGN s ON` θ`) r`
$\alpha(\mathbf{r})$	:	`SELECT ABSORB * FROM r`

Properties:

- Access to old timestamps
- Old and new timestamp available together
- Control over new timestamps

Example RA mapping:

$$_D\vartheta^T_{SUM(B)}(\mathbf{p}) \equiv {}_{D,T}\vartheta_{SUM(B)}(\mathcal{N}_{\mathbf{p1}.D=\mathbf{p2}.D}(\epsilon_U(\mathbf{p})/\mathbf{p1}, \epsilon_U(\mathbf{p})/\mathbf{p2}))$$

Example SQL mapping:

```
WITH
 px AS (SELECT Ts Us, Te Ue, * FROM p)
SELECT D, SUM(B), Ts, Te
FROM (px p1 NORMALIZE px p2 ON p1.D=p2.D) p
GROUP BY D, Ts, Te
```

http://www.ifi.uzh.ch/dbtg/research/align.html

A. Dignös, M. H. Böhlen, and J. Gamper, "Temporal alignment". In *Proc. of SIGMOD*, 2012, pp. 433–444.

Scaling Functions

Definition: Scales an attribute value x from an old timestamp T_{old} to a new timestamp T_{new}:

$$scale(x, T_{new}, T_{old}) \rightarrow x'$$

Example: UDF-function scaling uniformly in PostgreSQL's procedural language PL/pgSQL:

```
CREATE FUNCTION
scaleU(x FLOAT, ts_new DATE, te_new DATE,
                ts_old DATE, te_old DATE)
RETURNS FLOAT AS $$
BEGIN
RETURN x * (te_new - ts_new) / (te_old - ts_old);
END; $$ LANGUAGE PLPGSQL;
```

Other scaling functions:

- Considering working days only
- Atomic attribute values
- Trends

Published in: *Proceedings of the 29th IEEE International Conference on Data Engineering (Demo)*, Brisbane, Australia, April 8-11, 2013.