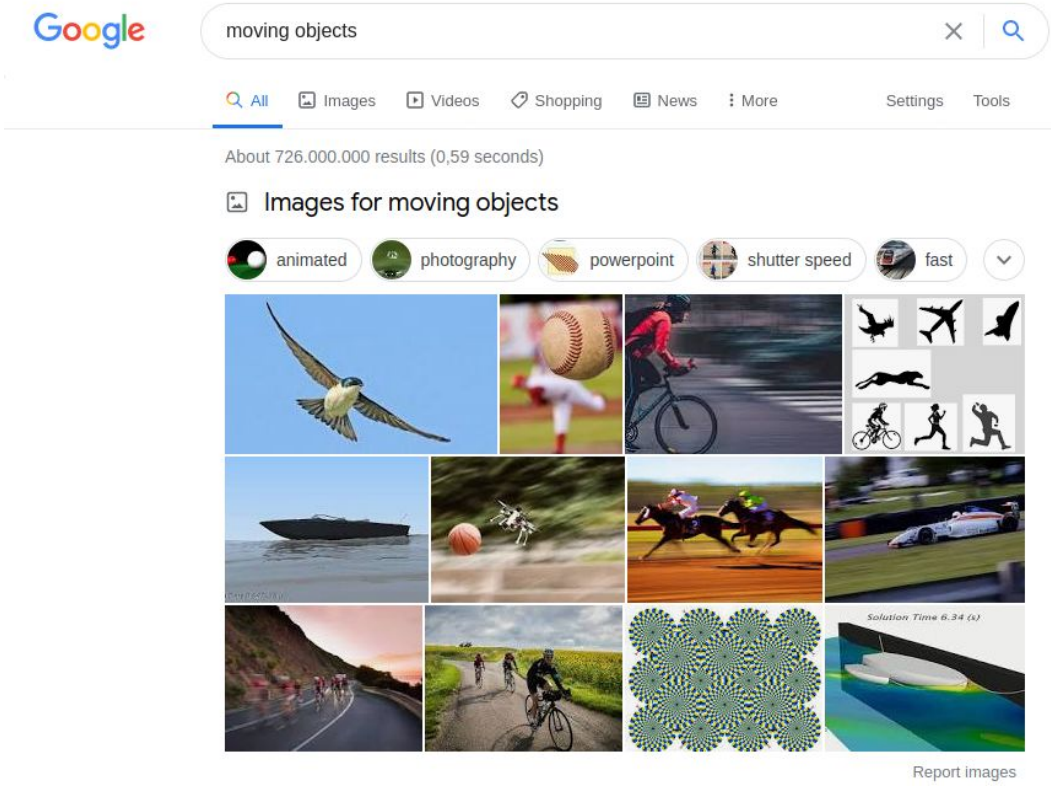
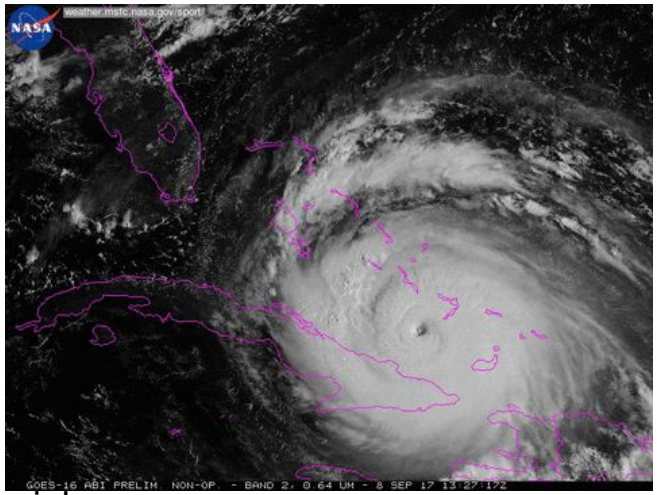
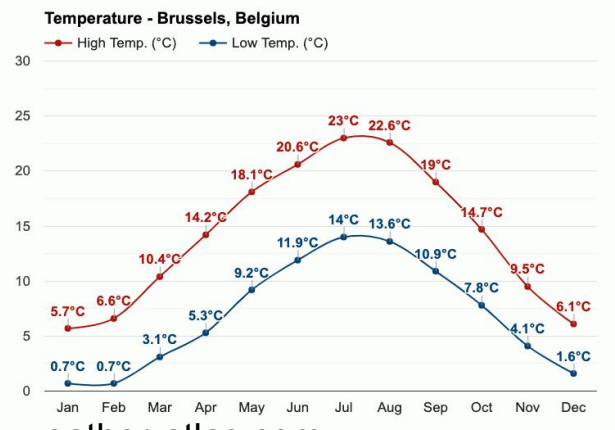
MobilityDB - A Moving Object Database

Moving Objects



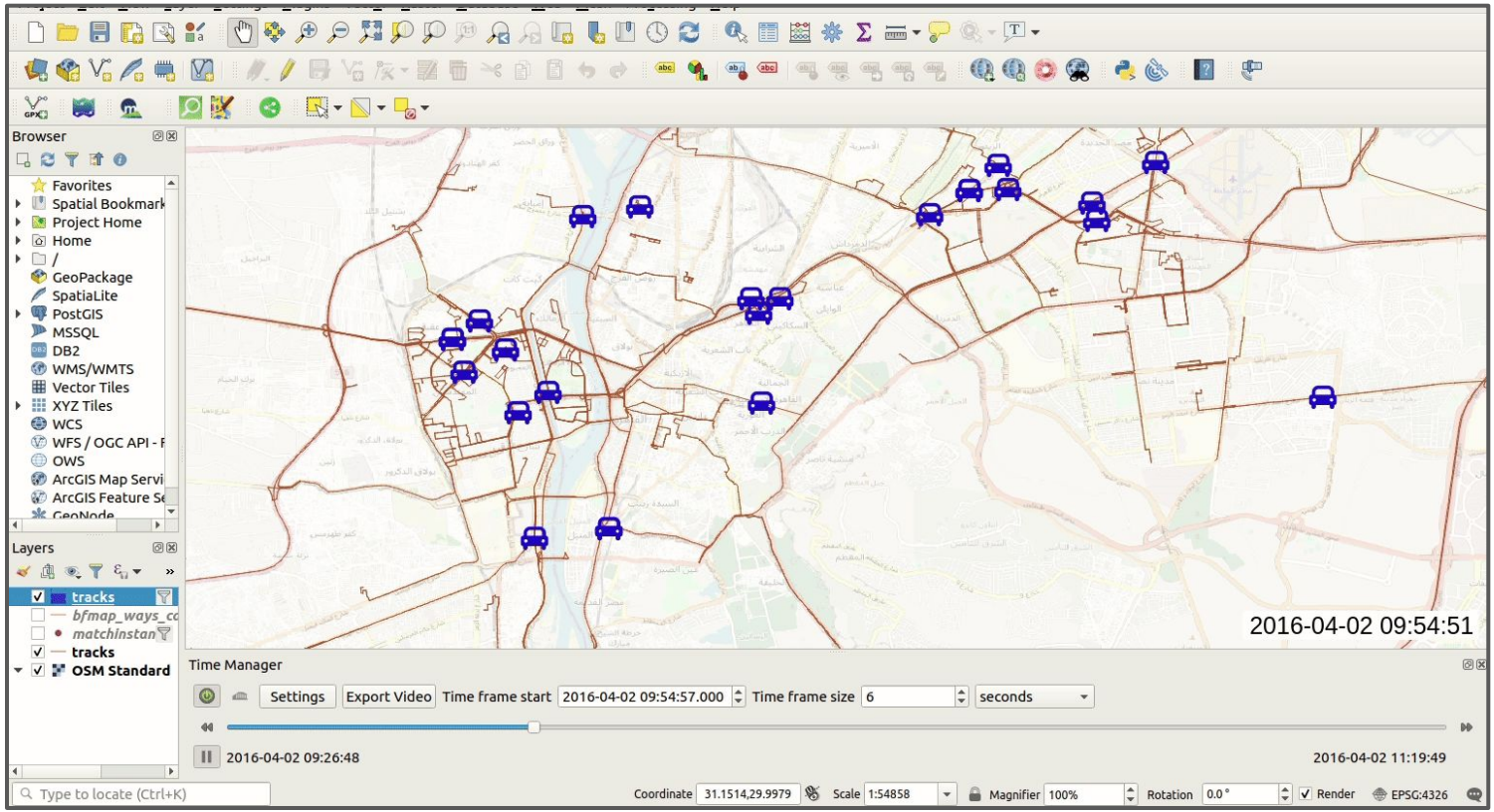


giphy.com

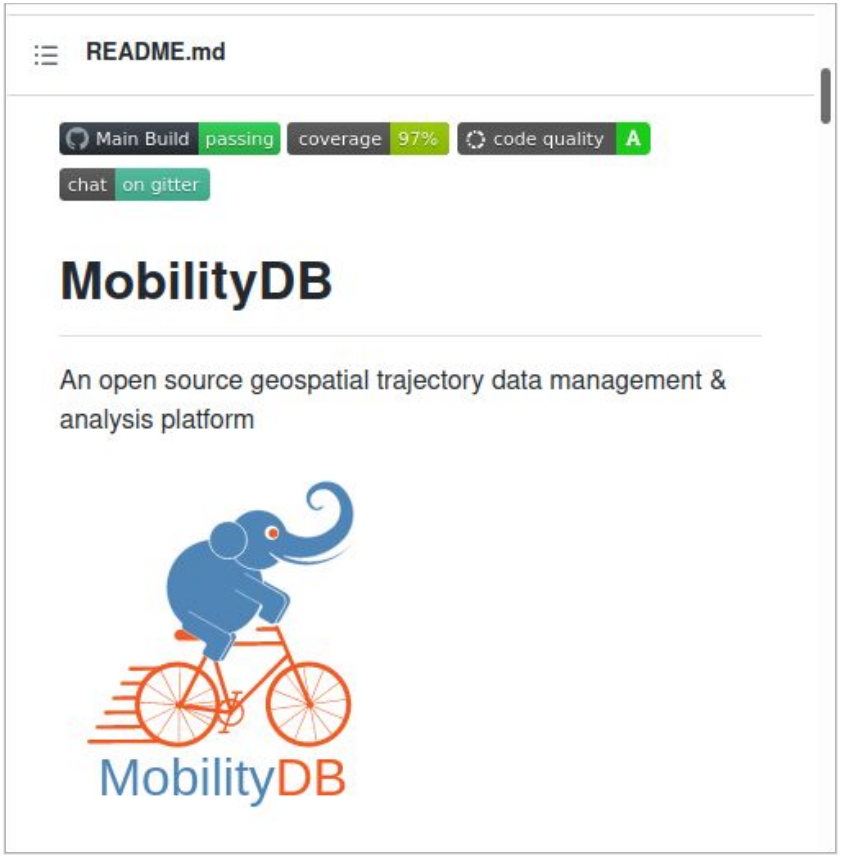


weather-atlas.com

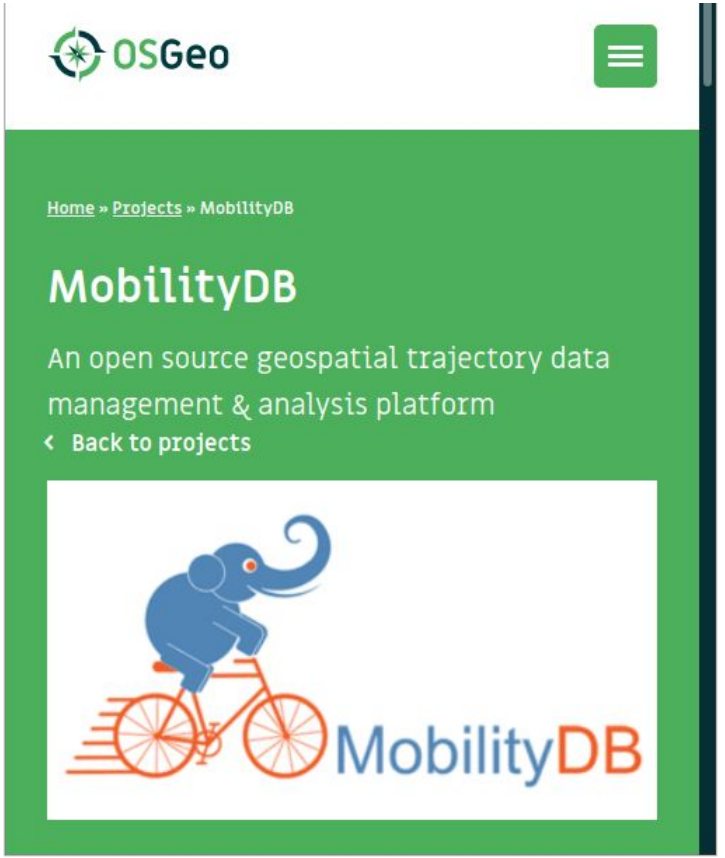
Moving Object Databases



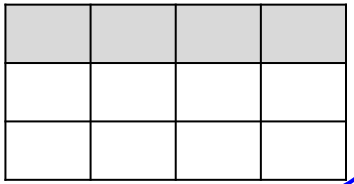
https://github.com/MobilityDB/MobilityDB



https://www.osgeo.org/projects/mobilitydb/



	PostgreSQL	PostGIS	MobilityDB
SQL Optimization	Scalar statistics & selectivity estimation	Grid-based statistics	Spatial grid + period bound histograms
Operations	Comparison, transformation, casting, ...etc	topological, CRS, properties, overlay, ...etc.	trajectory, temporal properties, lifted predicates, aggregations
Indexes	B-tree, hash, GiST, SP-GiST, GIN, BRIN	GiST, SP-GiST, BRIN	GiST, SP-GiST
Type System	numeric, character, date/time, bool, xml, json	Geometry, geography	tgeompoint, tgeogpoint, tint, tfloat, ttext, tbool



MobilityDB Ecosystem

MobilityDB Network	MobilityDB Geometry	MobilityDB MapMatch	MobilityDB Stream	MobilityDB View	MobilityDB Generator
pgRouting	AWS MobilityDB Cloud		kafka	MobilityDB Python	MobilityDB Java
docker	kubernetes	citusdata	psycopg	asyncpg	PostgreSQL JDBC
kepler.gl mapbox	QGIS	Grafana	plotly	python	Java
	MobilityDB	PostGIS	PostgreSQL	ubuntu	

Using MobilityDB

- How is MobilityDB packaged ? Which files constitute the package ?
- How to use MobilityDB ?

The scientific and engineering foundations

- The data model.
- The operations.
- Lifted operations
- Temporal aggregations.
- Indexing.
- Optimizer statistics, and selectivity estimation.
- Internal architecture.
- Eco system.
- OGC standards.

MobilityDB Data Model

TEMPORAL(timeType, baseType): time → value

timestamptz
timestampSet
period
periodSet

geometry(point)
geography(point)
bool
float
int
string
...

=

```
TEMPORAL(timestamptz,
                    geometry(point))
//database of car accidents
TEMPORAL(timestampSet,
                    geometry(point))
//foursquare check-ins
TEMPORAL(period,
                    geometry(point))
//car trajectory
...
```

Two Data Models

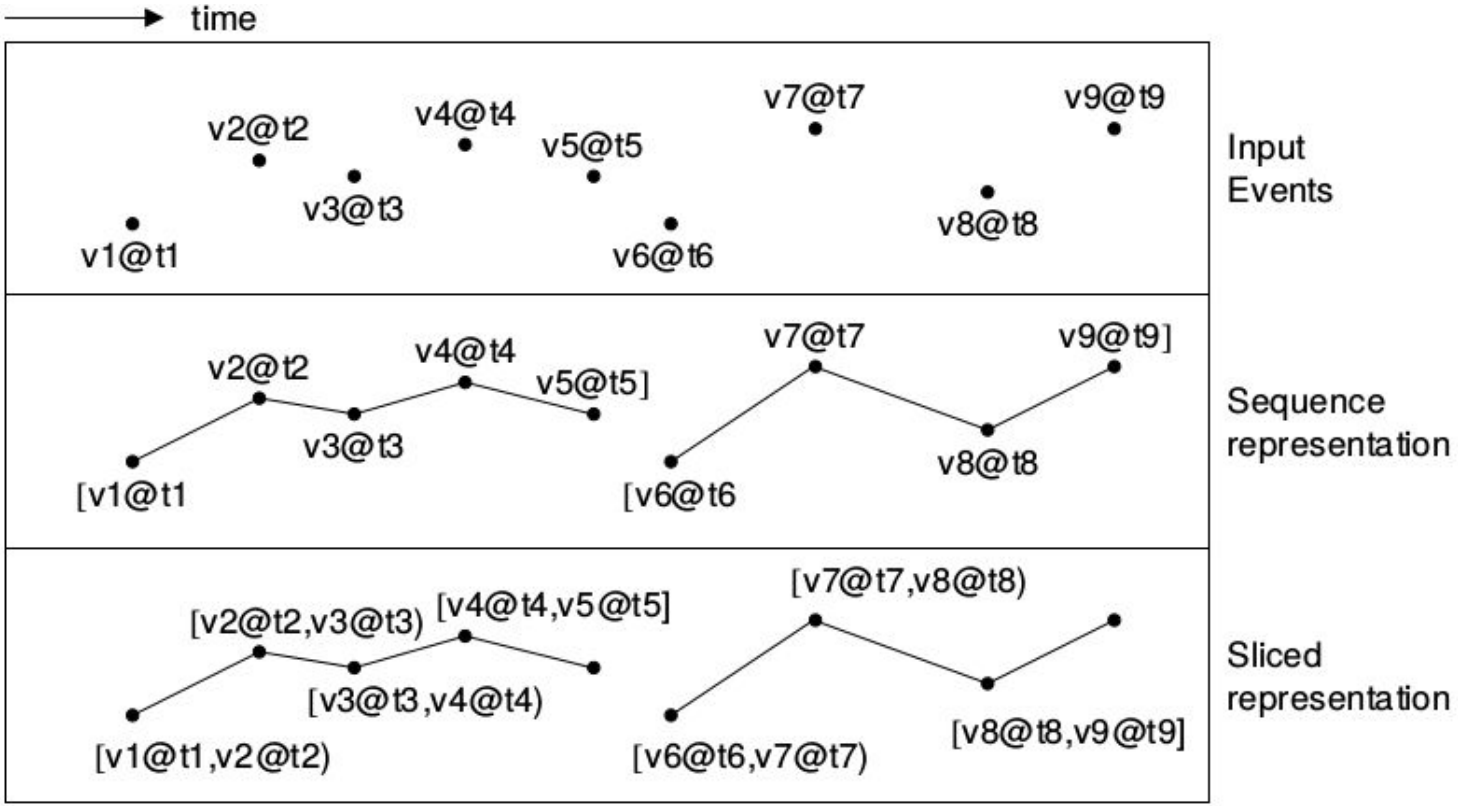
Sliced representation

Ralf Hartmut Güting, Michael H. Böhlen, Martin Erwig, Christian S. Jensen, Nikos A. Lorentzos, Markus Schneider, and Michalis Vazirgiannis. 2000. A foundation for representing and querying moving objects. *ACM Transactions on Database Systems*. 25, 1 (March 2000), 1–42. DOI:https://doi.org/10.1145/352958.352963

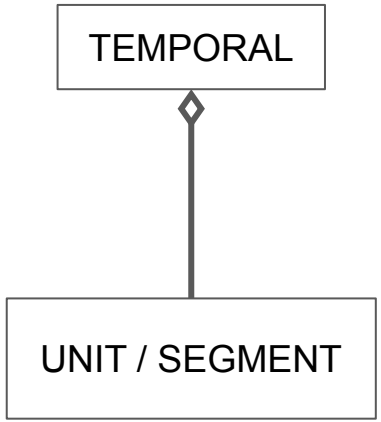
Sequence representation

[1] Esteban Zimányi, Mahmoud Sakr, Arthur Lesuisse, MobilityDB: A Mobility Database based on PostgreSQL and PostGIS. To appear in *ACM Transactions on Database Systems*, 2020. Preprint

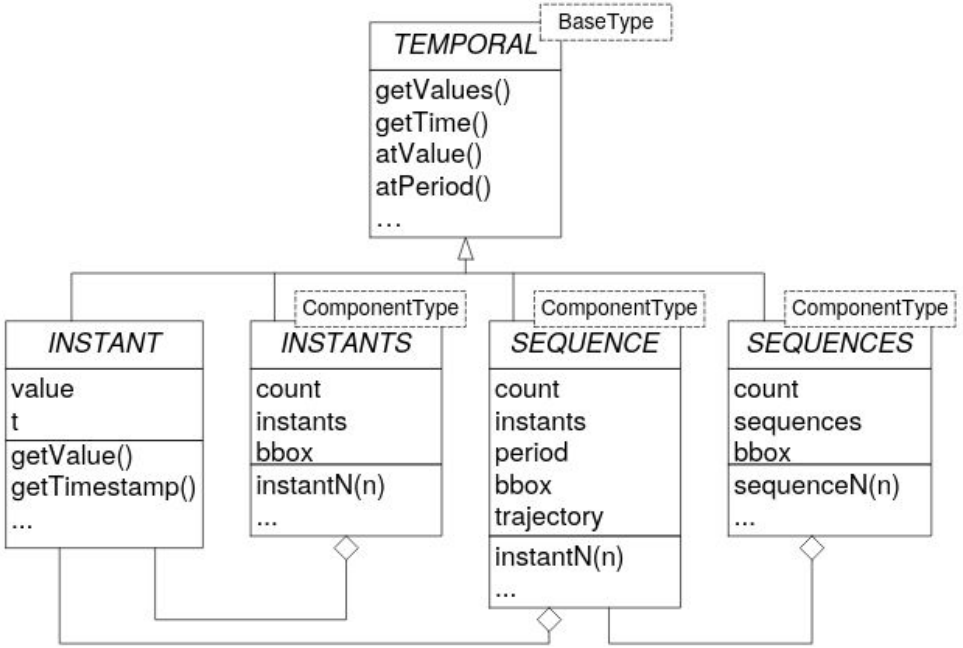
A not very accurate visualization of the two abstract models



Type hierarchies of the two models

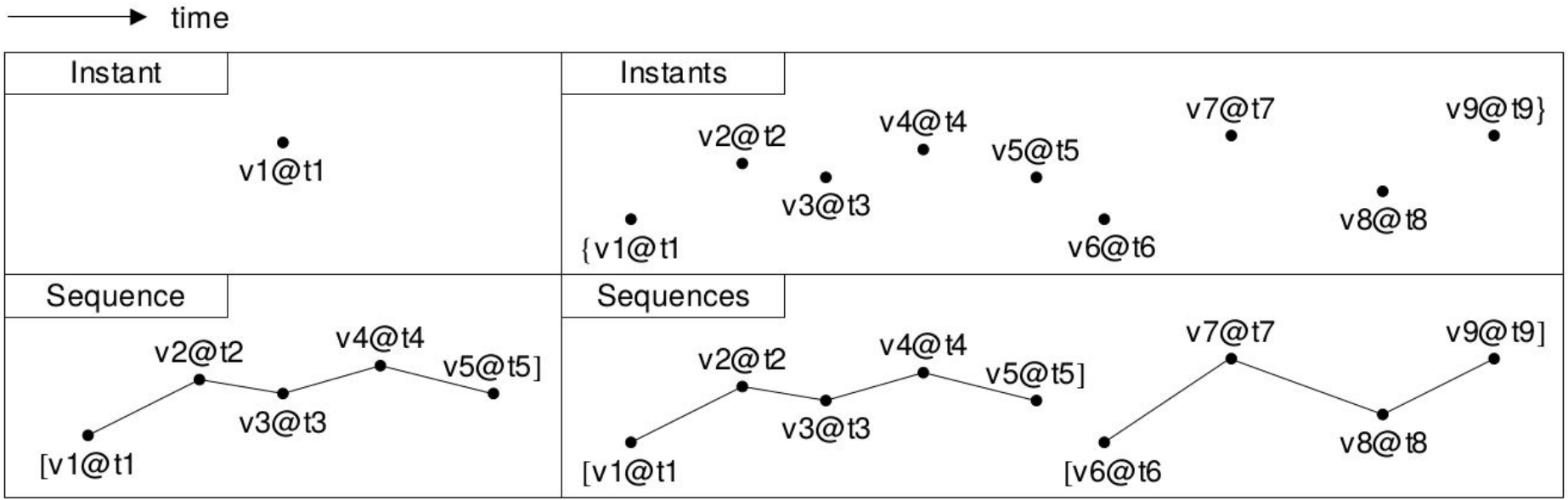


Sliced representation



Sequence representation

Comparison: storage, semantics



	Sliced representation	Sequence representation
INSTANT	[v1@t1, v1@t1]	v1@t1
INSTANTS	[v1@t1, v1@t1], [v2@t2, v2@t2], ...	{v1@t1, v2@t2, …}
SEQUENCE	[v1@t1, v2@t2), [v2@t2, v3@t3), ...	[v1@t1, v2@t2, v3@t3, …]
SEQUENCES	[v1@t2, v2@t2), ... [v4@t4, v5@t5], [v6@t6), ...	{[v1@t1, …, v5@t5], [v6@t6, …, v9@t9]}

Two abstract models: storage size

Algorithm 6: Generic strategy for a binary temporal operation on two temporal arguments α_1, α_2

begin
  $result \leftarrow \emptyset$;
  $syncPeriods \leftarrow$ synchronize(α_1, α_2);
  foreach *period* $p \in syncPeriods$ **do**
    if α_1 *is defined during* p *AND* α_2 *is defined during* p **then**
      $resultUnit \leftarrow$ apply the operation on α_1 and α_2 both restricted to the period p;
      $result \leftarrow result \cup resultUnit$;
  foreach *consecutive pair of units* $u_i, u_{i+1} \in result$ **do**
    if *end timestamp of* u_i = *start timestamp of* u_{i+1} *AND*
    the temporal functions of u_i *and* u_{i+1} *are equivalent* **then**
      merge u_i, u_{i+1} into one unit;
  return *result*;

MOD Functions

- Synchronization [1]
- Normalization [1]
- Lifted Operations
  - https://github.com/MobilityDB/MobilityDB/wiki/Lifting-a-Function-in-MobilityDB
- Lifted Topological Predicates
- Temporal aggregations.
- … [1]

Interfacing MobilityDB Types to Indexes

Access Methods

SELECT * FROM pg_am;

oid oid	amname name	amhandler regproc	amtype "char" (1)
2	heap	heap_tableam_...	t
403	btree	bthandler	i
405	hash	hashhandler	i
783	gist	gisthandler	i
2742	gin	ginhandler	i
4000	spgist	spghandler	i
3580	brin	brinhandler	i

Operator classes
Support functions

An operator class contains a set of index strategies and a set of support functions for an index to manipulate a certain data type.

User types

period
tgeompoint
tfloat
...

https://www.postgresql.org/docs/current/xindex.html

GiST (Generalized Search Tree)

Hellerstein Joseph, Naughton Jeffrey, Pfeffer Avi. Generalized search trees for database systems. In: Proceedings of the 21st international conferences on very large data bases. 1995. p. 562–73.

“we present the abstract data type (or “object”) Generalized Search Tree(GiST). We define its structure, its invariant properties, its extensible methods and its built-in algorithms.”

Extensible methods: consistent, union, compress, decompress, penalty, picksplit

http://gist.cs.berkeley.edu/ (last modified 1999)

SP-GiST (Space Partitioning Generalized Search Tree)

Eltabakh Mohamed, Eltarras Ramy and Aref Walid, "Space-Partitioning Trees in PostgreSQL: Realization and Performance," *22nd International Conference on Data Engineering (ICDE'06)*, Atlanta, GA, USA, 2006, pp. 100-100, doi: 10.1109/ICDE.2006.146.