Journal of Information and Data Management, 2026, 17:1, doi: 10.5753/jidm.2026.5890
This work is licensed under a Creative Commons Attribution 4.0 International License.

MasterMobilityDB: A Storage and Processing Layer for Multiple Aspect Trajectory Data at Scale

Flaris Roland Feller [**Universidade Federal de Santa Catarina** | ***flaris.feller@gmail.com***]
Ronaldo dos Santos Mello [**Universidade Federal de Santa Catarina** | ***r.mello@ufsc.br***]

Departamento de Informática e Estatística, Universidade Federal de Santa Catarina, Bairro Trindade, Florianópolis, SC, 88040-900, Brasil.

Received: 12 April 2025 • **Published:** 13 March 2026

Abstract Spatiotemporal data capturing the movement of real-world entities, gathered from sensors and GPS devices, along with its integration with other contextual georeferenced data, has led to the generation of large and complex trajectory datasets. These datasets, referred to as multiple aspect trajectories (MATs), present new challenges for moving object databases. This work introduces MasterMobilityDB, a persistence layer for MATs based on the Master representation model, built on top of the MobilityDB database through a dedicated API. A comparison with the state-of-the-art SecondoDB shows that MasterMobilityDB enables more natural query expressions and delivers better performance for MATs. This is a pioneer solution to deal with MAT persistence and management.

Keywords: Moving Object Databases, Trajectory, Multiple Aspect, Spatio-temporal.

1 Introduction

With the widespread adoption of GPS enabled devices, the capture of position data has become effortless, leading to the daily accumulation of large amounts of such data. As a result, trajectory data management and moving object databases (MODs) have become prominent areas of research.

According to [Güting *et al.*, 2015], a trajectory represents the movement of an entity, such as a person, a vehicle, or an animal. At a lower level of abstraction, it consists of a time-ordered sequence of spatial positions, commonly referred to as a raw trajectory. When this raw trajectory is further enriched with multiple contextual semantic dimensions that vary in nature, it becomes what is known as a multiple aspect trajectory (MAT) [Mello *et al.*, 2019]. Figure 1 illustrates an example of a MAT showing an individual's daily movement, starting from home, passing through their office, and ending at a restaurant. This movement is enhanced with various semantic attributes, including weather conditions, mean of transportation, and points of interest (POI) characteristics.

This work adopts the Master data model [Mello *et al.*, 2019], which offers a more comprehensive approach to data variety. Unlike previous models such as symbolic trajectories [Güting *et al.*, 2015], which represent aspects as simple semantic labels, or CONSTANT [Bogorny *et al.*, 2014], which is restricted to predefined aspects, Master provides greater flexibility in handling diverse contextual dimensions.

For the persistence of trajectory data, the database research community has developed several implementations of moving object database (MOD), including Secondo [Güting *et al.*, 2015], Hermes [Pelekis *et al.*, 2015], and MobilityDB [Zimányi *et al.*, 2019]. MobilityDB, built on top of PostgreSQL and the PostGIS extension, benefits from active support by a large community of companies and individuals. It offers numerous features for handling large-scale trajectory data, including abstract data types, specialized indexes, an extensive set of operators and functions, and compatibility with various environments, tools, and programming languages. These capabilities make MobilityDB a strong choice for storing and processing trajectory data. However, while MobilityDB primarily focuses on raw trajectories, Secondo and Hermes are limited to a single or a few fixed semantic properties, making them incapable of representing multiple aspect trajectories (MATs).

This paper revisits MasterMobilityDB, a persistence layer built on top of MobilityDB MOD to manage MATs using the Master model. We designed and implemented abstract data types for Master model entities and efficient methods for their manipulation in MasterMobilityDB. All are extending the MobilityDB data model through a PLSQL API.

This paper is an extended version of [Feller and Mello, 2024], presented at XXXIX Brazilian Symposium on Databases ($SBBD2024$). This SBBD paper introduces MasterMobilityDB. We have significantly improved Section 3 updating related work with more recent proposals for semantic trajectories persistence. Moreover, we have extended Section 5 by including new experiments with a greater volume of data to emphasize the robustness and performance of the solution.

The rest of this paper is organized as follows. Section 2 presents basic concepts regarding MAT and the Master Data Model. Section 3 presents a state-of-the-art review of proposals for semantic trajectory persistence. Section 4 details the MasterMobilityDB. Section 5 discusses experiments on the known semantic Geolife trajectory datasets. Section 6 concludes the paper.

2 Multiple Aspect Trajectory and the Master Data Model

A MAT [Mello *et al.*, 2019] is defined as a sequence of points $(p_1, p_2, ..., p_n)$ recorded for a moving object, where each point $p_i = (x, y, t, A)$ represents the object's position at coordinates *(x, y)* and time *t*, along with a set $A = a_1 : v_1, a_2 : v_2, ..., a_r : v_r$ of r aspect-value pairs. These aspects provide semantic context to the movement, capturing various real-world factors such as social media activity, weather conditions, or transportation modes. Each aspect includes attributes that offer detailed information, as illustrated in Figure 1 for an individual's MAT.

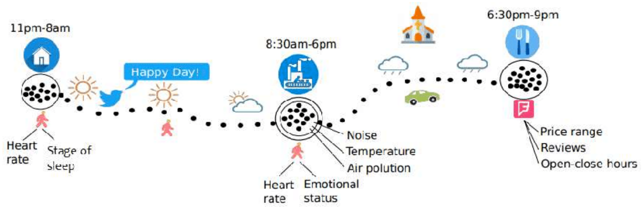


Figure 1. A MAT example.

Building on the definition of a MAT, the Master data model [Mello *et al.*, 2019, 2021] was proposed to overcome the limitations of earlier representation models. Its relational logical schema is shown in Figure 2[1]. A key feature of Master is the introduction of the concept of an aspect—a real-world factor that is significant for the analysis of trajectory data.

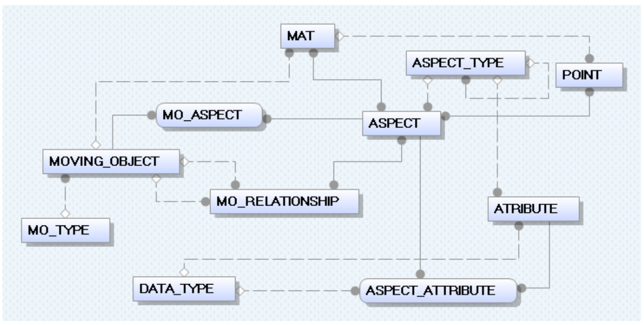


Figure 2. The Master data model

An aspect is defined by an *aspect type*, which specifies its metadata and attributes. Aspects can support various data types, including numbers, value ranges, text, geometries, or complex objects. For instance, an aspect type *hotel* may include attributes such as *geographical coordinates, address, stars, room types*, and *facilities*. An example of an aspect of this type could be: *"Il Mare Resort"*, with the following attribute-value pairs: *geographical coordinates, −27.439771, −48.500802; address, "Main Ave., 45, Atlantis"; stars, 5; room types, "suite", "junior suite"; facilities, "gym", "swimming pool", "restaurant", "bar", "beach service"*.

An aspect can be associated with a MAT, a point within a MAT, a moving object (MO), or even a relationship between MOs. A MAT is linked to an MO, which may represent entities such as a person, drone, animal, vehicle, or even a natural phenomenon like a hurricane.

[1]Table attributes were omitted to simplify the understanding of the proposed model.

3 Related Work

An initial step toward achieving the goals of this work involved conducting a systematic review of the state-of-the-art in the persistence of semantic trajectories in databases, as well as the implementation of representation models [Feller and Mello, 2022]. We have also revisited the research repositories to find new proposals.

A total of twenty-four works were selected (see Table 1) and categorized according to the main database technologies they utilize: *object-relational (OR), NoSQL, triplestore*, and *multimodel*. We also include MasterMobilityDB.

Table 1. Solutions for persistence of semantic trajectories

MOD DB	Base DB	Extension	Tech
Hermes@Oracle	Oracle	Spatial	OR
Hermes@Postgres	PostgreSQL	PostGIS	OR
Secondo	BerkeleyDB	Algebra	OR
Weka-STPM	PostgreSQL	PostGIS	OR
[Brandoli *et al.*, 2022]	MobilityDB	PostGIS	OR
STKST	PostgreSQL	PostGIS	OR
MODB	MySQL	Spatial	OR
[Xu *et al.*, 2023]	Secondo	Symbolic	OR
NALMO	Secondo	Symbolic	OR
DeepVQL	PostgreSQL	PostGIS	OR
[Gómez *et al.*, 2024]	MobilityDB	PostGIS	OR
MasterMobilityDB	**MobilityDB**	**PostGIS**	**OR**
[Gómez *et al.*, 2019]	Neo4j	Spatial	Graph
[Noureddine *et al.*, 2021]	Neo4j	Spatial	Graph
[Karim *et al.*, 2021]	Neo4j	Spatial	Graph
[He *et al.*, 2023]	Neo4j	Spatial	Graph
GSM	Neo4j	Spatial	Graph
Hermes@Neo4j	Neo4j	Spatial	Graph
[Huang and Li, 2022]	Cassandra	GeoMesa	W-column
[Zhao *et al.*, 2024]	LevelDB	S2geometry	Key-value
FrameSTEP	Virtuoso	-	Triplestore
STriDE	Stardog	-	Triplestore
[Torres *et al.*, 2020]	Apache Jena	-	Triplestore
[Tamilmani *et al.*, 2019]	PgSQL/Neo4j	PostGIS	Multimodel
[Wannous *et al.*, 2013]	Oracle	Spatial/RDF	Multimodel

When implementing a MOD using an object-relational (OR) database, a trajectory is typically represented as an abstract data type (ADT), which encapsulates spatiotemporal and semantic characteristics along with operations on these properties. Persistence is managed through relations with attributes derived from ADTs and is manipulated using SQL.

MOD implementations based on triplestores adopt ontology-driven approaches to model semantic trajectories, relying on three core ontologies: domain, time, and spatial. These ontologies provide context for movement, space, and time. In addition to the ontologies, rule sets are defined, with proposals emphasizing the use of SPARQL and GeoSPARQL for querying.

NoSQL database persistence strategies differ based on the underlying technology. In property graph-based approaches, semantic trajectories are represented through vertices—denoting points or episodes—and edges that connect them. Both vertices and edges can store semantic and spatiotemporal attributes as properties. In wide-column database implementations, the MO identifier is the partition key, while latitude, longitude, and timestamp are clustering keys. Additional columns, including trajectory ID, store the semantic attributes associated with the trajectory. Lastly, key-value DB-based implementations encode keys as a combination of trajectory ID and spatiotemporal attributes, and semantic attributes are serialized in the data value.

Finally, multimodel approaches encapsulate both the geometry and semantics of mobility data using different data models. For instance, the relational model stores raw GPS trajectories, while the graph model captures various semantic properties.

By generalizing the aspect concept, *MATs* represent a new view of trajectories and a new paradigm for mobility data. The MasterMobilityDB differs from the state-of-the-art by being the first proposal for persistence and manipulation of MATs. We adopt a MOD based on the OR technology as the basis for MasterMobilityDB (*i.e.*, MobilityDB) to take the advantage of the reusability principle for extending the MobilityDB data model as well as its data access methods.

4 MasterMobilityDB

MasterMobilityDB is a MAT data management layer on top of MobilityDB DB Management System (DBMS). Figure 3 gives an overview of the stack of DB technologies on which MasterMobilityDB is based. MobilityDB, in turn, was built over PostGIS DBMS, which is an extension of the PostgreSQL relational DB for managing georeferenced data. Each component on this stack of DB technologies processes a part of a MAT according to the data types it supports, and communicates with the others through available APIs. PostgreSQL handles alphanumeric data, relationships, referential integrity, and transactions. PostGIS treats geometries and spatial operations. MobilityDB handles raw trajectories and, finally, MasterMobilityDB is based on the Master data model.

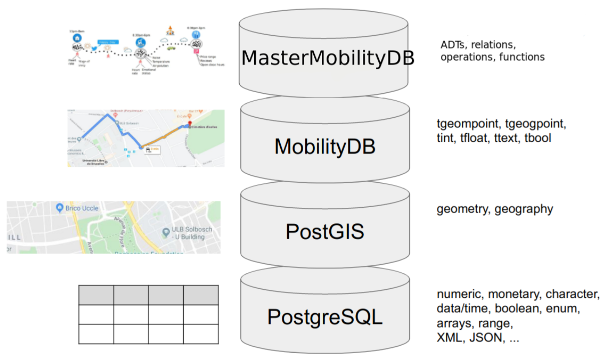

MasterMobilityDB
ADTs, relations, operations, functions
MobilityDB
tgeompoint, tgeogpoint, tint, tfloat, ttext, tbool
PostGIS
geometry, geography
PostgreSQL
numeric, monetary, character, data/time, boolean, enum, arrays, range, XML, JSON, ...
Brico Uccle
ULB Solbosch - U Building

Figure 3. MasterMobilityDB overview. [Feller, 2023]

4.1 Design issues

MasterMobilityDB implements several issues to efficiently support the persistence and manipulation of MATs, such as ADTs, tables and methods. Such objects are organized into logical groups using *DB schemes*, providing a way to separate different parts of the development according to their purpose. The schemes are as follows: *(i) Master*: the objects that support the Master model and extensions Master_DR and Master_SG; *(ii) partitions*: physical segmentation of high-volume tables; *(iii) util*: methods used in ETL procedures; and *(iv) staging*: user-developed ETL procedures; *v Master_dr* [Mello *et al.*, 2021] introduced the modeling of dependency rules (DRs) as patterns discovered on data present in MATs datasets; *mat_sg* [Machado *et al.*, 2022] for MAT summarizing, which determines representative points generated from a set of MATs with representative values for each aspect and whose sequencing generates representative trajectories. The source code for MasterMobilityDB, as well as its API, are available at *https://github.com/ffeller/MasterMobilityDB*.

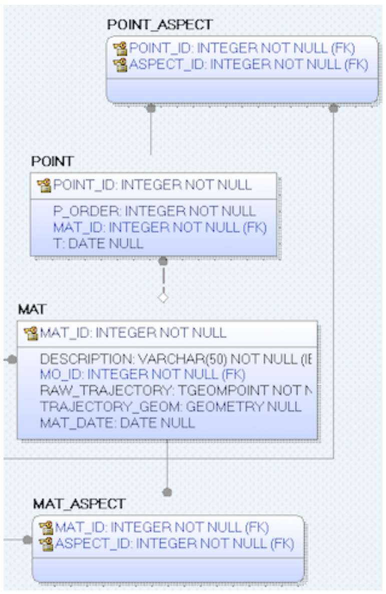

POINT_ASPECT
POINT_ID: INTEGER NOT NULL (FK)
ASPECT_ID: INTEGER NOT NULL (FK)
POINT
POINT_ID: INTEGER NOT NULL
P_ORDER: INTEGER NOT NULL
MAT_ID: INTEGER NOT NULL (FK)
T: DATE NULL
MAT
MAT_ID: INTEGER NOT NULL
DESCRIPTION: VARCHAR(50) NOT NULL (IE
MO_ID: INTEGER NOT NULL (FK)
RAW_TRAJECTORY: TGEOMPOINT NOT N
TRAJECTORY_GEOM: GEOMETRY NULL
MAT_DATE: DATE NULL
MAT_ASPECT
MAT_ID: INTEGER NOT NULL (FK)
ASPECT_ID: INTEGER NOT NULL (FK)

Figure 4. Part of the MasterMobilityDB logical data model.

In order to design MasterMobilityDB, we consider spatial data types offered by PostGIS and the *TGeomPoint* type introduced in MobilityDB for the persistence of raw trajectories. Based on these spatiotemporal data types, MasterMobilityDB defines a series of ADTs, *i.e.*, composite data types that mirror the Master model. An example of an ADT is *aspect*, which holds six attributes: *(i) aspect_id*, an integer that identifies the aspect; *(ii)* a variable character (*description*); *(iii)* two float values (x and y representing spatial coordinates); *(iv)* a *timestamp (t)*; *(v)* the *space_time flag* (integer), which indicates the aspect's nature (*spatial*, *temporal*, *spatiotemporal*) or *semantic*, and; *(vi) aspect_type_id*, an integer value that is a reference to a key in the corresponding *aspect_type* relation.

A key-design issue we consider in the definition of the MasterMobilityDB logical data model is the double representation of trajectory points, as shown in Figure 4. For the MAT representation, we consider the *TGeomPoint* type, as stated before (the *RAW_TRAJECTORY* attribute in the MAT table). This specific MobilityDB data type is efficient for filtering and querying conventional raw trajectories. Nevertheless, we also define the *POINT* table in order to allow the association of aspects that are specific to a trajectory point, like a point that holds a POI. This is in consonance with the Master model, giving flexibility to bind aspects to parts of a trajectory besides the whole trajectory.

In order to simplify the manipulation of MATs, we also design several methods for MasterMobilityDB. These methods were implemented within the *Master* and *util* schemes as procedures and functions in MobilityDB's PL/SQL language. Since the volume of data tends to increase, something considered in the layer design is the manipulation of several tuples in one operation, like *create_many()*, which inserts several tuples in one method call. As an example, Table 2 shows the main methods for the *aspect* ADT.

Table 2. Examples of methods for the *aspect* ADT

Method	Parameters	Description
aspect_count()		counts tuples
aspect_create()	inout aspect_typ	creates one tuple
aspect_create_many()	inout aspect_typ[]	creates tuples
aspect_delete_many()	in aspect_typ[]	deletes tuples
aspect_delete_all()		delete all tuples
aspect_delete_by_id()	in integer	deletes a tuple by PK id
aspect_delete_by_name()	in varchar	deletes a tuple by name
aspect_find_all()		finds all tuples
aspect_find_by_id()	in integer	finds a tuple by PK id
aspect_find_by_name()	in varchar	finds a tuple by name
aspect_update()	inout aspect_typ[]	updates a tuple

4.2 Functionalities for improving data access performance

The integration of motion data and alphanumeric data, as well as the transformation of this large volume of data into useful information, is usually complex and expensive. In this sense, MasteMobilityDB adopts valuable features inherited from MobilityDB for loading, querying, and building MATs: (i) object partitioning, (ii) temporary tables, (iii) external tables, and (iv) specialized indexes. They are detailed as follows.

For datasets that generate large volumes of MATs, tables such as *MAT* or *Point* from the Master model can be divided into smaller objects called partitions. It performs better when accessing these tables and their indexes, as it is propagated to the partitions using parallelism. For this purpose, the *create_partitions_by_date()* and *drop_partitions_by_date()* methods are in the *util* schema described in Table 3. With this, partitions are created in the *partitions* schema according to the period and interval (i.e., day, week, month, year). Table 3 also shows other methods of the *util* schema.

A temporary table in MobilityDB is a table that exists for a session in the DB and is automatically deleted when the session is closed. They help to store intermediate results within a specific context. In MasteMobilityDB, temporary tables are widely used in complex queries to manipulate a large volume of tuples. The strategy here is to divide a complex query into more straightforward queries that have their results stored in indexed temporary tables. It provides superior performance and lower resource consumption.

External tables (or *Foreign Data Wrappers (FDW)*) is a practical feature of MobilityDB that allows working with remote data or data in other formats, such as CSV files, spreadsheets, and JSON documents. FDWs enable the definition of external tables that access these remote data sources and integrate them into a single DB. For MasteMobilityDB, external tables are essential when mapping data from CSV files.

Table 3. Utility methods for ETL procedures

Method	Parameters	Description
create_partitions_by_date	p_schema, p_table, p_start, p_end, p_column, p_interval, p_partschema	Creates partitions for relation p_schema.p_table for the period.
drop_partitions_by_date	p_schema, p_table, p_start, p_end, p_interval, p_partschema	Drops partitions of relation p_schema.p_table for the period.
disable_fks	p_schema text, p_table text	Disables FKs for the relation.
enable_fks	p_schema text, p_table text	Enables FKs for the relation.
disable_indexes	p_schema text, p_table text	Disables indexes for the relation.
enable_indexes	p_schema text, p_table text	Enables indexes for the relation.
reset_sequence	p_schema text, p_table text	Resets sequence's value.
reset_sequences	p_schema text	Resets all sequences's values.

When implementing ETL processes, the following settings must be configured: *(i)* the directory of input files; *(ii)* a user mapping and permissions; and *(iii)* an external table for each file. This strategy makes easier to work with input data without managing it in regular tables, and can be combined with temporary tables.

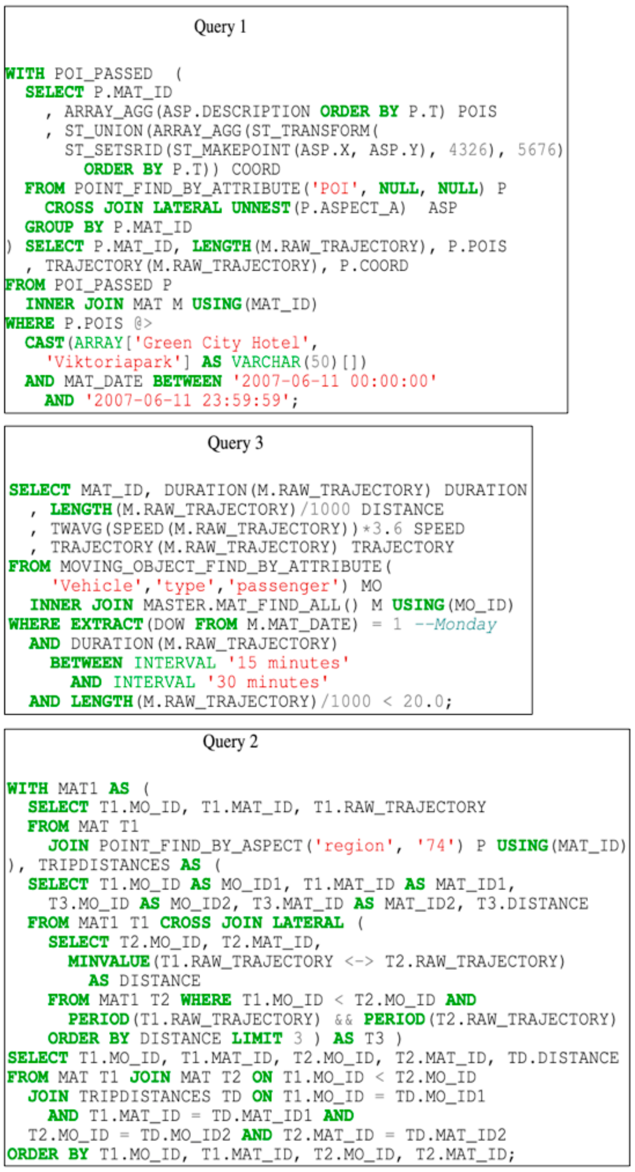
Finally, *GiST (Generalized Search Tree)* and *SP-GiST (Space-Partitioned Generalized Search Tree)* offer specialized trajectory indexing Capabilities. The query optimizer can automatically choose GiST or SP-GiST indexes without programming intervention. In MasteMobilityDB, MAT queries can be resolved efficiently using alphanumeric indexes for aspects and attributes as a primary filter, and GiST or SP-GiST indexes are considered to accelerate operations and searches over spatiotemporal data, providing the system with speed and agility.

4.3 Examples of queries

MasteMobilityDB extends the query capabilities of MobilityDB by allowing queries involving the semantic dimension besides queries over raw trajectories, which are limited to spatiotemporal dimensions. Some examples of queries are presented in Figure 5. They show the usage of new proper MasteMobilityDB methods that allow filters over semantic aspects and their relationships with the main spatiotemporal Master data model entities (moving objects, MATs, and points). A complete list of MasteMobilityDB query operations is available at Feller [2023].

Query 1 selects MATs whose sequence of points contains the *'Green City Hotel'* and *'Viktoriapark'* POIs, which are aspects, together with the sequence of visited locations, the distance traveled being returned by the MobilityDB *LENGTH()* method and the trajectory geometry by the MobilityDB *TRAJECTORY()* method. The use of semantic annotations for POIs and the usage of the MasteMobilityDB *POINT_FIND_BY_ATTRIBUTE()* method allows a more efficient implementation of the query because it avoids POI and trajectory geometry comparison.

Query 2 presents a *nearest-neighbor* query where both the reference and the candidate objects are moving objects that



Query 1

```sql
WITH POI_PASSED  (
  SELECT P.MAT_ID
    , ARRAY_AGG(ASP.DESCRIPTION ORDER BY P.T) POIS
    , ST_UNION(ARRAY_AGG(ST_TRANSFORM(
      ST_SETSRID(ST_MAKEPOINT(ASP.X, ASP.Y), 4326), 5676)
        ORDER BY P.T)) COORD
  FROM POINT_FIND_BY_ATTRIBUTE('POI', NULL, NULL) P
    CROSS JOIN LATERAL UNNEST(P.ASPECT_A)  ASP
  GROUP BY P.MAT_ID
) SELECT P.MAT_ID, LENGTH(M.RAW_TRAJECTORY), P.POIS
  , TRAJECTORY(M.RAW_TRAJECTORY), P.COORD
FROM POI_PASSED P
  INNER JOIN MAT M USING(MAT_ID)
WHERE P.POIS @>
  CAST(ARRAY['Green City Hotel',
    'Viktoriapark'] AS VARCHAR(50)[])
  AND MAT_DATE BETWEEN '2007-06-11 00:00:00'
    AND '2007-06-11 23:59:59';
```

Query 3

```sql
SELECT MAT_ID, DURATION(M.RAW_TRAJECTORY) DURATION
  , LENGTH(M.RAW_TRAJECTORY)/1000 DISTANCE
  , TWAVG(SPEED(M.RAW_TRAJECTORY))*3.6 SPEED
  , TRAJECTORY(M.RAW_TRAJECTORY) TRAJECTORY
FROM MOVING_OBJECT_FIND_BY_ATTRIBUTE(
    'Vehicle','type','passenger') MO
  INNER JOIN MASTER.MAT_FIND_ALL() M USING(MO_ID)
WHERE EXTRACT(DOW FROM M.MAT_DATE) = 1 --Monday
  AND DURATION(M.RAW_TRAJECTORY)
    BETWEEN INTERVAL '15 minutes'
      AND INTERVAL '30 minutes'
  AND LENGTH(M.RAW_TRAJECTORY)/1000 < 20.0;
```

Query 2

```sql
WITH MAT1 AS (
  SELECT T1.MO_ID, T1.MAT_ID, T1.RAW_TRAJECTORY
  FROM MAT T1
    JOIN POINT_FIND_BY_ASPECT('region', '74') P USING(MAT_ID)
), TRIPDISTANCES AS (
  SELECT T1.MO_ID AS MO_ID1, T1.MAT_ID AS MAT_ID1,
    T3.MO_ID AS MO_ID2, T3.MAT_ID AS MAT_ID2, T3.DISTANCE
  FROM MAT1 T1 CROSS JOIN LATERAL (
    SELECT T2.MO_ID, T2.MAT_ID,
      MINVALUE(T1.RAW_TRAJECTORY <-> T2.RAW_TRAJECTORY)
        AS DISTANCE
    FROM MAT1 T2 WHERE T1.MO_ID < T2.MO_ID AND
      PERIOD(T1.RAW_TRAJECTORY) && PERIOD(T2.RAW_TRAJECTORY)
    ORDER BY DISTANCE LIMIT 3 ) AS T3 )
SELECT T1.MO_ID, T1.MAT_ID, T2.MO_ID, T2.MAT_ID, TD.DISTANCE
FROM MAT T1 JOIN MAT T2 ON T1.MO_ID < T2.MO_ID
  JOIN TRIPDISTANCES TD ON T1.MO_ID = TD.MO_ID1
    AND T1.MAT_ID = TD.MAT_ID1 AND
  T2.MO_ID = TD.MO_ID2 AND T2.MAT_ID = TD.MAT_ID2
ORDER BY T1.MO_ID, T1.MAT_ID, T2.MO_ID, T2.MAT_ID;
```

Figure 5. Examples of queries executed by MasteMobilityDB.

have passed by *region 74* (an aspect), which can be accessed by the the MasteMobilityDB *POINT_FIND_BY_ASPECT()* method. The query starts by computing the nearest-neighbors in a MobilityDB temporary table *TRIPDISTANCES*. Then, the main query verifies for each pair of trips *T1* and *T2* where both belong to *TRIPDISTANCES*.

Finally, in Query 3, trajectories of *passenger vehicles* (a moving object aspect) are returned with the aid of the MasteMobilityDB *MOVING_OBJECT_FIND_BY_ATTRIBUTE()* method. Their related trips are filtered using the *EXTRACT()* method, which returns the day of the week, and the MobilityDB *DURATION(), SPEED(), LENGTH(),* and *TRAJECTORY()* methods, which return the trajectory duration, speed, distance, and geometry, respectively.

5 Experimental Evaluation

We provide an experimental evaluation of MasterMobilityDB by comparing the time performance of data operations against the symbolic trajectories model offered by Secondo DB, the state-of-the-art in terms semantic trajectory DBMS. The well-known Gowalla [Cho *et al.*, 2011] trajectory dataset was considered: Gowalla is a location-based social networking website where users share their locations by checking in. The friendship network comprises 196591 nodes or users and 950327 edges or relations between them. This dataset includes long-term check-in data worldwide collected from Gowalla API. It has collected 6,442,890 check-ins from these users between February 2009 and October 2010. It contains files in CSV format with eight columns, as shown in Table 4. The dataset was imported into Secondo DB and MasterMobilityDB (MMDB), generating 174,531 trajectories calculated weekly from real user check-ins with at least ten points. The average record length of the input data was 2,786 bytes, with a total size of 424,108,032 bytes.

Table 4. Data from the Gowalla dataset

#	Description	MMDB	Secondo
1	User ID (anonymized)	Moving Object	Attribute
2	UTC time	MAT/Point	RT Att.
3	Latitude	MAT/Point	RT Att.
4	Longitude	MAT/Point	RT Att.
5	Location ID (Gowalla)	Aspect	Sem. Att.
6	Location cat. ID (Gowalla)	Attribute	Sem. Att.
7	Location cat. name (Gowalla)	Attribute	Sem. Att.
8	User ID (related)	MO Relationship	Sem. Att.

For Secondo DB, a *CheckinSymTraj* table was created to hold user trajectories. It contains an *User ID* that identifies the moving object, and a *RT* attribute to hold the weekly raw trajectory built by the aggregation of *Latitude, Longitude*, and *UTC time*. Besides *RT*, some *semantic* attributes were defined to label some *Location* aspects, as stated in Table 4.

Table 5. Queries for Gowalla dataset

#	Description
1	How many trajectories exist for each user with 400 or more connections?
2	Where have the users connected to User_1802 been at each instant from Instants?
3	Which user IDs are related to Users who passed the POIs from Locations?
4	What is the minimum distance between places in the perimeter of 50km of point with coordinates 18.0580794811 and 59.3301577, where ten users, with at least 400 connections, were and another ten?
5	What are the pairs of user IDs within 10m or less of each other?
6	What are the IDs of the users who reached the POIs before all users during the entire observation period?
7	What do users travel the total distances with User IDs with at least 400 connections during each interval from Periods?
8	What does a user travel the greatest distance during each interval from Periods?
9	When and where did users with User IDs meet other users (distance < 3m), and what are these last IDs?
10	Which users passed by a POI point from Locations in one of the Instants?
11	Which users found themselves at a point from Locations in an instant of Instants?
12	Which users passed by one of the Regions during each interval from Periods?
13	Which users passed by one of the Regions in an instant of Instants?
14	Which users passed a point from Points during a period from Periods?
15	List the pairs of users that were both located within a region from Regions during a period from Periods.
16	List the first time at which a user visited a point in Points?
17	How long does building MATs from the Gowalla dataset take?

For MasterMobilityDB, a *Location* was mapped to an aspect with a *Location Category ID*, and *Location Category Name* added to the *Attribute* table. In turn, MATs were created in a same way of Secondo DB, i.e., generated by the aggregation of the same CSV colunms by *User ID* and *week*. Additionally, the intersection of the Locations' geometries with MAT and points generated the relationships from the Location aspect with *MAT* and *Points* table.

For this experiment, we collect the execution mean times for 16 queries, three times each, as well as the load time, as shown in Table 5. Tables 6 and 7, in turn, shows the query execution times for MasterMobilityDB, Secondo, as well as the difference. To avoid been exhaustive, we do not show the implementation of each query in MasterMobilityDB and Secondo DB. The queries comprise combinations of alphanumeric, temporal, spatial, and spatiotemporal predicates, operations, and aggregations over the data, and the specified query filters allows to test a wide range of index structures, access methods, and spatiotemporal operators.

Computing times were achieved on an Intel Core I7 2.8 GHz computer with 16 GB RAM and 512 GB HD, running Ubuntu 22.04. Measurements are carried out by varying the considered percentage of the input data sample: 10%, 40%, 70%, and 100%.

Tables 6 and 7 presents the query execution times in seconds for MasterMobilityDB (MM), Secondo DB (Sec), and the spent time difference (Dif). The 10% sample has shown close values, but for higher samples, the difference was more significant in favor of MasterMobilityDB. It can be explained by the way Secondo DB persists semantic data in terms of labels, dealing with one semantic aspect at a time. So, all semantic complex attributes that hold these labels must be searched for queries that involve multiple aspects to check if the aspects occur together. On the other hand, MasterMobilityDB models the semantic dimensions as aspects and attributes, which permit access to any information for the moving objects, as well as their MATs and points. Instead, load time (Query 17) was slightly lower for Secondo DB due to the few number of tables to insert data. Furthermore, all experiments were conducted in the same manner in both databases, including the creation of indexes to optimize queries and the use of temporary tables to pre-process parts of the results.

Table 6. Query execution times over MATs for Gowalla dataset

Qry		**10%**			**40%**	
#	**MM**	**Sec**	**Dif**	**MM**	**Sec**	**Dif**
1	0.02	0.05	-0.03	0.02	0.07	-0.05
2	0.02	0.07	-0.05	0.03	0.10	-0.07
3	0.07	5.68	-5.61	0.05	25.87	-25.82
4	0.05	0.13	-0.08	0.05	0.21	-0.17
5	0.05	0.12	-0.07	0.05	0.17	-0.12
6	0.13	7.28	-7.14	0.20	41.68	-41.48
7	0.02	18.61	-18.60	0.03	26.86	-26.83
8	0.08	0.20	-0.12	0.23	0.68	-0.45
9	0.03	0.41	-0.38	0.03	5.68	-5.64
10	0.12	0.40	-0.28	0.23	1.96	-1.73
11	2.57	1.02	1.55	2.10	6.55	-4.46
12	1.06	0.30	0.76	3.56	1.19	2.38
13	2.64	0.83	1.82	2.71	3.12	-0.41
14	2.33	0.97	1.35	2.36	3.38	-1.02
15	1.04	25.26	-24.22	7.71	99.66	-91.95
16	0.17	0.31	-0.15	0.45	1.09	-0.64
17	9.83	2.41	7.43	42.60	11.47	31.14

Table 7. Query execution times over MATs for Gowalla dataset

Qry		**70%**			**100%**	
#	**MM**	**Sec**	**Dif**	**MM**	**Sec**	**Dif**
1	0.02	0.07	-0.05	0.03	0.08	-0.05
2	0.03	0.10	-0.07	0.07	0.13	-0.07
3	0.05	25.87	-25.82	0.07	77.63	-77.57
4	0.05	0.21	-0.17	0.07	0.31	-0.25
5	0.05	0.17	-0.12	0.07	0.41	-0.35
6	0.20	41.68	-41.48	0.23	137.64	-137.41
7	0.03	26.86	-26.83	0.07	28.79	-28.73
8	0.23	0.68	-0.45	0.59	1.91	-1.32
9	0.03	5.68	-5.64	0.07	32.46	-32.39
10	0.23	1.96	-1.73	0.41	7.97	-7.56
11	2.10	6.55	-4.46	2.95	34.91	-31.96
12	3.56	1.19	2.38	9.01	4.52	4.49
13	2.71	3.12	-0.41	2.76	9.60	-6.85
14	2.36	3.38	-1.02	22.23	10.71	11.52
15	7.71	99.66	-91.95	37.39	254.40	-217.01
16	0.45	1.09	-0.64	1.42	2.76	-1.34
17	75.37	20.53	54.85	109.23	30.20	79.04

6 Conclusion

Mobility data management and analysis have emerged in the last decade as a very active research domain addressing diverse issues such as clustering, integration, mining, indexing, persistence, and privacy. While previous research focused on processing raw trajectories collected from sensors and GPS devices, for example, recent research focuses on methods for enriching a trajectory with more contextualized and application-oriented information. Adding semantics to movement data brings enormous potential for the analysis of mobility-related phenomena. For example, understanding why and how people and animals move, what places they visit and for what purposes, what their activities are, and what resources they use is extremely important for decision-making, particularly for public authorities responsible for managing society.

Within this context, this work addresses an important issue that has not tackled by the literature: the persistence of MATs based on the pioneer data model in the literature called Master. The purpose here is to store and manipulate unlimited semantic data about mobility, where the representation and query of MATs are considered from a DB perspective. We materialize our solution, called *MasterMobilityDB* as an extension of a comprehensive raw trajectory DBMS called MobilityDB. New ADTs and an API dedicated to the representation and manipulation of MATs, respectively, were developed. An experimental evaluation comparing MasterMobilityDB against the state-of-the-art in terms of DBMS for management of semantic trajectory data demonstrates the feasibility and superior time performance for querying MAT data.

Mobility data has multiple applications, with *bigdata analytics* and data mining standing out. We consider MasterMobilityDB as a basis for several future projects related to MAT data management. One ongoing work in our DB research group is the support for MAT data integrity constraints, *i.e.*, to allow the definition and guarantee of some classes of integrity constraints over MAT data without introducing a significant overhead in data management. Another study in progress is the support for MAT data analytics through the modeling of specific machine learning methods that could be applied over MAT entities, as well as the persistence of analysis results. Future work on creating new utility routines in MasterMobilityDB that assist in applying methods, such as clustering and classification, related to such applications is of great value. Among these utility routines, we can mention the creation of ETLs from the Master model generating *dataframes* and *datasets*.

References

Bogorny, V., Renso, C., Aquino, A., Siqueira, F., and Alvares, L. (2014). CONSTAnT - A Conceptual Data Model for Semantic Trajectories of MOs. *Trans. GIS*, 18(1):66–88. DOI: 10.1111/tgis.12011.

Brandoli, B. *et al*. (2022). From Multiple Aspect Trajectories to Predictive Analysis: A Case Study on Fishing Vessels in the Northern Adriatic Sea. *GeoInformatica*, 26:551–579. DOI: 10.1007/s10707-022-00463-4.

Cho, E., Myers, S. A., and Leskovec, J. (2011). Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1082–1090. DOI: 10.1145/2020408.2020579.

Feller, F. (2023). MasterMobilityDB - Uma Camada de Persistência e Manipulação para Trajetórias de Múltiplos Aspectos. Master's thesis, Universidade Federal de Santa Catarina.

Feller, F. and Mello, R. (2022). A Survey on Persistence Strategies for Semantically Enriched Trajectories. In *GEOINFO 2022*, pages 50–62. MCTIC/INPE.

Feller, F. and Mello, R. (2024). Mastermobilitydb: A persistence and manipulation layer for trajectories of multiple aspects. In *Anais do XXXIX Simpósio Brasileiro de Bancos de Dados*, pages 574–586, Porto Alegre, RS, Brasil. SBC. DOI: 10.5753/sbbd.2024.240245.

Güting, R., Valdés, F., and Damiani, M. (2015). Symbolic Trajectories. *ACM Trans. Spatial Algorithms Syst.*, 1(2):7:1–7:51. DOI: 10.1145/2786756.

Gómez, L., Vaisman, A., and Zimányi, E. (2024). Querying mobile pollution data using mobilitydb. In *2024 25th IEEE International Conference on Mobile Data Management (MDM)*, pages 227–234. DOI: 10.1109/MDM61037.2024.00047.

Gómez, L. *et al*. (2019). Analytical Queries on Semantic Trajectories using Graph Databases. *Transactions in GIS*, 23:1078–1101. DOI: 10.1111/tgis.12556.

He, Y., Hofer, B., Sheng, Y., Yin, Y., and Lin, H. (2023). Processes and events in the center: a taxi trajectory-based approach to detecting traffic congestion and analyzing its causes. *International Journal of Digital Earth*, 16:509–531. DOI: 10.1080/17538947.2023.2182374.

Huang, B. and Li, Z. (2022). Spatiotemporal indexing and query application on cassandra for large-scale trajectory data. In *2022 5th International Conference on Data Science and Information Technology (DSIT)*, pages 1–6. DOI: 10.1109/DSIT55514.2022.9943905.

Karim, L., Boulmakoul, A., *et al*. (2021). Trajectory-based modeling for fraud detection and analytics: Foundation and design. In *2021 IEEE/ACS 18th International Conference on Computer Systems and Applications (AICCSA)*, pages 1–7. DOI: 10.1109/AICCSA53542.2021.9686920.

Machado, V., Mello, R., and Bogorny, V. (2022). A method for summarizing trajectories with multiple aspects. In *DEXA 2022, Part I*, volume 13426, pages 433–446. Springer. DOI: 10.1007/978-3-031-12423-5_33.

Mello, R., Bogorny, V., Alvares, L., Santana, L., Ferrero, C., Frozza, A., Schreiner, G., and Renso, C. (2019). MASTER: A Multiple Aspect View on Trajectories. *Transactions in GIS*, 23(4):805–822. DOI: 10.1111/tgis.12526.

Mello, R., Schreiner, G. A., Alchini, C. A., dos Santos, G. G., Bogorny, V., and Renso, C. (2021). Dependency Rule Modeling for Multiple Aspects Trajectories. In *40th International Conference on Conceptual Modeling, ER*, volume 13011 of *Lecture Notes in Computer Science*, pages 123–132. Springer. DOI: 10.1007/978-3-030-89022-3_11.

Noureddine, H. *et al*. (2021). A Hierarchical Indoor and Outdoor Model for Semantic Trajectories. *Transactions in GIS*, 26:214–235. DOI: 10.1111/tgis.12841.

Pelekis, N., Frentzos, E., Giatrakos, N., and Theodoridis, Y. (2015). HERMES: A Trajectory DB Engine for Mobility-centric Applications. *Int. J. Knowl. Based Organ*, 5:19–41. DOI: 10.4018/ijkbo.2015040102.

Tamilmani, R., Stefanakis, E., *et al*. (2019). Modelling and Analysis of Semantically Enriched Simplified Trajectories Using Graph Databases. *Advances in GIScience of the ICA*, 1:1–8. DOI: 10.5194/ica-adv-1-20-2019.

Torres, Y. *et al*. (2020). Stop-and-Move Sequence Expressions over Semantic Trajectories. *International Journal of Geographical Information Science*, 35:1–26. DOI: doi.org/10.1080/13658816.2020.1793157.

Wannous, R., Malki, J., Bouju, A., and Vincent, C. (2013). *Time Integration in Semantic Trajectories Using an Ontological Modelling Approach*, volume 185, pages 187–198. Springer Berlin Heidelberg. DOI: 10.1007/978-3-642-32518-2_18.

Xu, J., Lu, H., and Bao, Z. (2023). A Query Optimizer for Range Queries over Multi-Attribute Trajectories. *ACM Trans. Intell. Syst. Technol.*, 14(1). DOI: 10.1145/3555811.

Zhao, X., Lam, K.-Y., and Kuo, T.-W. (2024). Indexing spatiotemporal trajectory data streams on key-value storage. *Computing*, 106:1–29. DOI: 10.1007/s00607-024-01304-y.

Zimányi, E., Sakr, M. A., Lesuisse, A., and Bakli, M. S. (2019). Mobilitydb: A mainstream moving object database system. In Aref, W. G., Bertolotto, M., Bouros, P., Jensen, C. S., Mahmood, A. R., Nørvåg, K., Sacharidis, D., and Sarwat, M., editors, *Proceedings of the 16th International Symposium on Spatial and Temporal Databases, SSTD 2019, Vienna, Austria, August 19-21, 2019*, pages 206–209. ACM. DOI: 10.1145/3340964.3340991.