# MobilityDB: Hands on Tutorial on Managing and Visualizing Geospatial Trajectories in SQL

Esteban Zimányi, Mahmoud Sakr, Mohamed Bakli
Maxime Schomans, Dimitris Tsesmelis, Robin Choquet
Université libre de Bruxelles (ULB), Brussels, Belgium
esteban.zimanyi@ulb.be,mahmoud.sakr@ulb.be

## ABSTRACT

MobilityDB is an open source moving object database. It extends PostgreSQL and PostGIS with types and operations for managing continuous geospatial trajectories. This hand-on tutorial will introduce the attendees to: (1) trajectory data management in MobilityDB, (2) visualization of moving object data in QGIS, and (3) distributed spatiotemporal query processing using MobilityDB. All the tutorial queries will be in SQL.

## CCS CONCEPTS

• **Information systems** → **Database management system engines**.

## KEYWORDS

Moving Object Database, SQL, Citus, QGIS Visualization

**ACM Reference Format:**

Esteban Zimányi, Mahmoud Sakr, Mohamed Bakli and Maxime Schomans, Dimitris Tsesmelis, Robin Choquet. 2021. MobilityDB: Hands on Tutorial on Managing and Visualizing Geospatial Trajectories in SQL. In *3rd ACM SIGSPATIAL International Workshop on Geospatial Data Access and Processing APIs (SpatialAPI'21), November 2, 2021, Beijing, China.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3486189.3490016

## PROJECT HOME

https://mobilitydb.com/

**Mailing Lists:**

- mobilitydb-dev, mobilitydb-users@lists.osgeo.org

## 1 INTRODUCTION

The increased availability of geospatial trajectory data, fueled by the advances in embedded sensors, has opened opportunities and ambitions to build a new applications that make use of location intelligence. For instance, the European Strategy for Low-Emission Mobility takes a step forward towards linking the road taxes to the vehicle kilometrage and pollution [EC, 2016][1]. This and other applications in domains like maritime, social mobility, urban and logistics create needs for moving object data management tools.

The moving object database community has been active since the early 2000s, contributing methods and prototypes. ST-Hadoop [1] and HadoopTrajectory [3] both fork and extend Hadoop with trajectory types and operations, which can be invoked in the user map-reduce programs. Similar works based on Spark include TrajSpark [14], UITrajMan [8], and Apache Sedona [13]. These systems are more suited for compute intensive analysis tasks, where the gain of distributed processing is higher than the initial overhead of loading and distributing the data files. For a big class of applications, the requirements are to store and to do frequent queries that can make use of indexes. The aforementioned vehicle tax application, for instance, would involve dashboard style queries that select and/or aggregate by time, spatial regions, vehicle type, etc. For this kind of applications, a database platform will be more suitable.

MobilityDB is an SQL moving object database [15]. It uses the extensibility features of PostgreSQL to implement an abstract data type model of moving objects [10]. It defines the TEMPORAL type constructor, that extends the base types of PostgreSQL and the geometry types of PostGIS respectively into temporal and spatiotemporal types for representing temporal integers, temporal Booleans, temporal geometries, etc. Perhaps the most related work to MobilityDB is SECONDO [9]. Many of the types and functions of MobilityDB are inspired from SECONDO. The benefit of MobilityDB is that it builds on a widely used system. It is engineered as an extension to PostgreSQL, which can be dynamically loaded, without a need to restart the server. In contrast to a fork, an extension is fully compatible with PostgreSQL and PostGIS and their ecosystem, which enables users to seamlessly use MobilityDB in deployment environments. MobilityDB is an OSGeo community project[2].

The query API of MobilityDB, fully in SQL, covers (1) Input, output, and construction of (spatio)temporal types, (2) Coordinate system transformation, (3) Static and temporal properties of the moving object, (4) Projection and restriction in space and time, (5) Temporal topological predicates, (6) Intersection, distance, and knn operations, (7) Temporal speed and heading, (8) Temporal aggregations, (9) GiST and SP-GiST indexes. This API implements common functions in the literature, and the functions specified or suggested by the ISO and OGC standards on moving features [6]. In addition, PostGIS has a big number of spatial functions, which we lift, i.e., overload for spatiotemporal types, when relevant.

The limitations of MobilityDB are of two kinds. The first is platform-related, including limitations of the declarative nature of SQL. For complex analytic tasks, such as map making, one would clearly need to complement SQL with a general programming language. For this purpose MobilityDB implements a Python driver[3].

[1]https://eur-lex.europa.eu/resource.html?uri=cellar:e44d3c21-531e-11e6-89bd-01aa75ed71a1.0002.02/DOC$_1 format = PDF$

[2]https://www.osgeo.org/projects/mobilitydb/

[3]https://github.com/MobilityDB/MobilityDB-python



*SpatialAPI'21, November 2, 2021, Beijing, China*



ACM ISBN 978-1-4503-9103-0/21/11.
https://doi.org/10.1145/3486189.3490016

Another example is that PostGIS has limited support to geographies compared to geometries, limiting in turn the MobilityDB support for temporal geographies compared to temporal geometries. The second kind of limitations is related to the current state of implementation. The API of MobilityDB in the moment covers lifted function and spatiotemporal transformations. ML-based APIs such as data cleaning and trajectory segmentation are still missing.

The tutorial will introduce the attendees to use MobilityDB and some of its advanced features, with the aim of raising interest in the project and increasing its user and developer communities.

## 2 TUTORIAL OUTLINE

The opening of the tutorial will be a short 10 minutes presentation, where a general overview of MobilityDB will be explained. The rest of the tutorial will be hands-on.

### 2.1 Tutorial Material

In order to focus on the usability aspects of MobilityDB, we try to keep the installation requirements minimal. Most of the tutorial exercises can be executed in a browser, without special installation requirements. We prepare the following material. (1) Tutorial handouts: including the SQL queries, and snippets of the expected results (2) MobilityDB cluster on MS Azure: which will be made accessible to attendees. As such, they will not need to install MobilityDB on their machines. (3) Preloaded database: The MobilityDB cluster will be populated with the dataset that will be used in the tutorial. (4) pgAdmin Web Interface: which is an SQL client for issuing MobilityDB queries and visualizing the results in a web browser. Again the attendees will not need to install a client on their machines. (5) QGIS & MOVE Plugin: for the visualization part of the tutorial, attendees will need to install QGIS[2], and our MOVE plugin[12], which creates animated visualizations of moving objects.

### 2.2 Part I: Querying (35 minutes)

Attendees will experiment with constructing spatiotemporal trajectories from raw observation. They will then express and run several exploratory and analysis queries in SQL, covering different classes of operations: (1) Spatiotemporal select-project-join, (2) Optimization using spatiotemporal GiST and SP-GIST indexes, (3) Trajectory knn queries, (4) Temporal aggregations This part is based on a previous tutorial that was presented in pgConf.ru for the developers community of PostgreSQL [11]. The goal of this part is to introduce the attendees to the types and functions of MobilityDB, and to exercises expressing mobility-related queries.

### 2.3 Part II: Visualization (25 minutes)

QGIS is a common visualization tool for spatial databases. Efforts to create animated moving object visualization in QGIS include TimeManager [4], which has recently been re-implemented as a feature in QGIS called Temporal Controller. Our plugin MOVE [12] uses Temporal Controller for visualizing MobilityDB types: both moving points and moving regions. In addition it integrates with QGIS, allowing rich visualization of moving objects, their traces, and their interactions with the spatial context. The tutorial will cover: (1) Visually explore individual moving point and moving regions objects, (2) Visualize the spatial trajectories of moving points, and the traversed areas of moving regions, (3) Visually analyze the topological relationships of mixture of moving and spatial objects, (4) Create aggregate visualization such as flow maps.

[4]https://anitagraser.com/projects/time-manager/

### 2.4 Part III: DB Distribution (20 minutes)

MobilityDB integrates with Citus[7] for distributing moving object data and queries[4, 5]. Citus has capabilities to partition tables over a cluster of PostgreSQL nodes. Given a user SQL query, Citus break it into a plan consisting of a set of SQL queries to the cluster nodes and to the coordinator, which is equivalent to the user query.

In this part of the tutorial, the attendees will connect to the prepared coordinator node of the cluster, and experiment with the query distribution: (1) Understand the MobilityDB & Citus integration, (2) Understand data distribution, (3) Build local spatiotemporal indexes, (4) Distribute range queries, knn queries, and trajectory queries, (5) Explore the distributed query plans. (6) If time allows, we may discuss ongoing work on distributed spatiotemporal joins.

## REFERENCES


[1] Louai Alarabi, Mohamed F. Mokbel, and Mashaal Musleh. 2017. ST-Hadoop: A MapReduce Framework for Spatio-Temporal Data. In *Proceedings of the 15th International Symposium on Advances in Spatial and Temporal Databases, SSTD 2017*. Springer, Arlington, VA, USA, 84–104.

[2] QGIS.ORG association. 2002-2021. *The QGIS project.* https://qgis.org/

[3] Mohamed Bakli, Mahmoud Sakr, and Taysir Hassan A. Soliman. 2019. Hadoop-Trajectory: a Hadoop spatiotemporal data processing extension. *Journal of Geographical Systems* 21, 2 (2019), 211–235.

[4] Mohamed Bakli, Mahmoud Sakr, and Esteban Zimanyi. 2019. Distributed Moving Object Data Management in MobilityDB. In *Proceedings of the 8th ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data* (Chicago, Illinois) *(BigSpatial '19)*. ACM, New York, NY, USA, Article 1, 10 pages.

[5] Mohamed Bakli, Mahmoud Sakr, and Esteban Zimányi. 2020. Distributed Spatiotemporal Trajectory Query Processing in SQL. In *Proceedings of the 28th International Conference on Advances in Geographic Information Systems* (Seattle, WA, USA) *(SIGSPATIAL '20)*. ACM, New York, NY, USA, 87–98.

[6] OGC Open Geospatial Consortium. 2013. *OGC Moving Features.* https://www.opengeospatial.org/standards/movingfeatures

[7] Umur Cubukcu, Ozgun Erdogan, Sumedh Pathak, Sudhakar Sannakkayala, and Marco Slot. 2021. Citus: Distributed PostgreSQL for Data-Intensive Applications. In *Proc. of the 2021 International Conference on Management of Data, SIGMOD Conference 2021*. ACM. To appear.

[8] X. Ding, L. Chen, Y. Gao, C.S. Jensen, and H. Bao. 2018. UlTraMan: A Unified Platform for Big Trajectory Data Management and Analytics. *Proc. of the VLDB Endowment* 11, 7 (2018), 787–799.

[9] Ralf Hartmut Güting, Victor Almeida, Dirk Ansorge, Thomas Behr, Zhiming Ding, Thomas Höse, Frank Hoffmann, Markus Spiekermann, and Ulrich Telle. 2005. SECONDO: An Extensible DBMS Platform for Research Prototyping and Teaching. In *Proceedings of the 21st International Conference on Data Engineering, ICDE'05*. IEEE Computer Society, 1115–1116.

[10] 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 (2000), 1–42.

[11] Postgres Professional. 2015-2021. *PGConf.Online 2021 - Managing moving objects data with MobilityDB.* https://pgconf.ru/en/2021/291542

[12] Maxime SCHOEMANS. 2021. *MOVE: QGIS Plugin to display MobilityDB query results.* https://github.com/mschoema/move

[13] Jia Yu, Zongsi Zhang, and Mohamed Sarwat. 2019. Spatial Data Management in Apache Spark: The GeoSpark Perspective and Beyond. *Geoinformatica* 23, 1 (Jan. 2019), 37–78.

[14] Zhigang Zhang, Cheqing Jin, Jiali Mao, Xiaolin Yang, and Aoying Zhou. 2017. TrajSpark: A Scalable and Efficient In-Memory Management System for Big Trajectory Data. In *Proc. of the APWeb-WAIM Joint Conference on Web and Big Data* (Beijing, China). Springer, 11–26.

[15] Esteban Zimányi, Mahmoud Attia Sakr, and Arthur Lesuisse. 2020. MobilityDB: A Mobility Database Based on PostgreSQL and PostGIS. *ACM Transactions on Database Systems* 45, 4 (2020), 19:1–19:42.