

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**  
**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**BỘ MÔN: HỆ CƠ SỞ DỮ LIỆU**



**BÁO CÁO**

**ĐỀ TÀI:**

**POSTGRESQL, POSTGIS VÀ MOBILITYDB**

**Giáo viên hướng dẫn: Nguyễn Thị Ái Thảo**

**Họ và tên:**

**MSSV**

Nguyễn Phúc Gia Khiêm

2211573

Nguyễn Trương Thái Bảo

2210251

**THÀNH PHỐ HỒ CHÍ MINH, THÁNG 12 NĂM 2024**

# Mục lục

<b>1. Giới thiệu .....</b>	<b>1</b>
<b>1.1. PostgreSQL.....</b>	<b>1</b>
<b>1.2. PostGIS .....</b>	<b>1</b>
<b>1.3. Động lực .....</b>	<b>2</b>
<b>2. Các kỹ thuật.....</b>	<b>3</b>
<b>2.1. Mở rộng kiểu dữ liệu .....</b>	<b>3</b>
Chi tiết các thành phần .....	4
<b>2.2. Nâng toán tử (Operator lifting) .....</b>	<b>5</b>
<b>2.3. Đồng bộ hóa và chuẩn hóa .....</b>	<b>5</b>
2.3.1. Synchronization .....	5
2.3.2. Normalization.....	6
<b>3. Cấu trúc của MobilityGB.....</b>	<b>7</b>
<b>3.1. Third-party tools and applications .....</b>	<b>8</b>
<b>3.2. PostgreSQL Extensions.....</b>	<b>8</b>
<b>3.3. Data Model.....</b>	<b>8</b>
<b>3.4. Access Methods.....</b>	<b>8</b>
<b>3.5. Query Optimizer .....</b>	<b>8</b>
<b>3.6. Operations.....</b>	<b>9</b>
<b>3.7. Ứng dụng và tích hợp.....</b>	<b>9</b>
<b>4. Đánh giá.....</b>	<b>9</b>

## 1. Giới thiệu

### 1.1. PostgreSQL

PostgreSQL (thường gọi tắt là Postgres) là một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở (open-source relational database management system - RDBMS). Nó nổi bật nhờ:

- Hỗ trợ các chuẩn SQL và ACID (Atomicity, Consistency, Isolation, Durability).
- Hỗ trợ cả mô hình cơ sở dữ liệu quan hệ (relational) và không quan hệ (NoSQL).
- Khả năng mở rộng, tích hợp tốt với các công cụ và công nghệ hiện đại.

PostgreSQL hỗ trợ nhiều kiểu dữ liệu cơ bản và nâng cao:

- **Cơ bản:** **integer, text, boolean, date**, v.v.
- **Nâng cao:**
  - **JSON/JSONB:** Lưu trữ dữ liệu dạng JSON với khả năng tìm kiếm và lọc dữ liệu hiệu quả.
  - **Array:** Hỗ trợ lưu trữ và thao tác với mảng dữ liệu.
  - **UUID:** Định danh duy nhất toàn cầu.
  - **HSTORE:** Lưu trữ dữ liệu dạng key-value.

Bên cạnh đó, PostgreSQL cung cấp các tính năng như:

- Index đa dạng: B-Tree, Hash, GIN, GiST, BRIN.
- Hỗ trợ truy vấn song song (parallel query).
- Caching và tối ưu hóa truy vấn hiệu quả.

### 1.2. PostGIS

PostGIS là một phần mở rộng (extension) của PostgreSQL, cho phép nó trở thành một hệ quản trị cơ sở dữ liệu không gian (spatial database). Với PostGIS, bạn có thể lưu trữ, truy vấn, và phân tích dữ liệu không gian như điểm, đường, vùng (polygons), v.v.

#### Các tính năng nổi bật

PostGIS mở rộng PostgreSQL với các kiểu dữ liệu không gian như:

- **Geometry:** Dữ liệu không gian 2D (điểm, đường, vùng).
- **Geography:** Dữ liệu không gian trên bề mặt trái đất (hệ tọa độ địa lý).
- **Raster:** Dữ liệu ảnh (ảnh vệ tinh, bản đồ độ cao).

PostGIS cung cấp hơn **300 hàm không gian** để thao tác và phân tích dữ liệu không gian, ví dụ:

- **ST\_Distance:** Tính khoảng cách giữa hai đối tượng không gian.
- **ST\_Intersects:** Kiểm tra xem hai đối tượng có giao nhau hay không.
- **ST\_Union:** Kết hợp hai vùng thành một.
- **ST\_Transform:** Chuyển đổi hệ tọa độ.

PostGIS tuân thủ các chuẩn OGC (Open Geospatial Consortium), đảm bảo tính tương thích với các công cụ GIS như QGIS, ArcGIS.

PostGIS sử dụng **GiST (Generalized Search Tree)** và **SP-GiST** để tăng tốc các truy vấn không gian.

### 1.3. Động lực

Những tiến bộ vượt bậc trong công nghệ theo dõi vị trí, với sự xuất hiện của chip GPS hàng loạt, định vị GSM, hệ thống nhận diện tự động AIS, định vị Wi-Fi và nhiều công nghệ khác, đã tạo ra một lượng dữ liệu khổng lồ. Tuy nhiên, việc tận dụng hiệu quả nguồn dữ liệu này vẫn còn nhiều hạn chế do thiếu các công cụ quản lý đa năng, có khả năng đáp ứng nhu cầu của các ứng dụng đa dạng như giao thông thông minh, an toàn hàng hải, nghiên cứu biến đổi khí hậu và phát triển phương tiện tự hành.

Hơn nữa, mặc dù đã có nhiều nghiên cứu về cơ sở dữ liệu đối tượng di chuyển từ đầu những năm 2000, vẫn chưa có một hệ thống MOD chính thống nhắm đến việc sử dụng trong công nghiệp. Các hệ thống hiện tại như **SECONDO** và **HERMES** chỉ là các nguyên mẫu nghiên cứu, không tập trung vào tích hợp với các hệ thống cơ sở dữ liệu chính thống.

MobilityDB được đề xuất để:

- **Tận dụng sức mạnh của PostgreSQL và PostGIS:** Sử dụng các tính năng quản lý dữ liệu mạnh mẽ, tối ưu hóa truy vấn, và hệ sinh thái phong phú của chúng.
- **Hỗ trợ SQL:** Cung cấp một giao diện truy vấn thân thiện để người dùng có thể dễ dàng thực hiện các phép toán trên dữ liệu di chuyển.
- **Tuân thủ tiêu chuẩn công nghiệp:** MobilityDB tuân theo các tiêu chuẩn OGC (Open Geospatial Consortium) để đảm bảo tính tương thích và khả năng ứng dụng thực tế.

MobilityDB hướng đến việc làm cho các nghiên cứu về MOD trở nên dễ tiếp cận hơn đối với người dùng trong công nghiệp thông qua tích hợp chặt chẽ vào các hệ thống cơ sở dữ liệu phổ biến.

Hệ thống được thiết kế để:

- **Sử dụng lại các cơ chế hiện có:** Tận dụng các cải tiến từ PostgreSQL và PostGIS (như xử lý song song, lập chỉ mục không gian, tối ưu hóa truy vấn).
- **Mở rộng linh hoạt:** Cho phép thêm các kiểu dữ liệu và chức năng mới trong tương lai.

### 1.4. Định nghĩa của MobilityDB

MobilityDB: Cơ sở dữ liệu di động (MOD) thế hệ mới

- **Mở rộng từ PostgreSQL và PostGIS:** MobilityDB được xây dựng dựa trên nền tảng vững chắc của PostgreSQL và PostGIS, hai trong số những hệ quản trị cơ sở dữ liệu quan hệ và không gian phổ biến nhất hiện nay.
- **Quản lý dữ liệu đối tượng di chuyển:** MobilityDB chuyên biệt trong việc quản lý dữ liệu về các đối tượng đang di chuyển, như phương tiện giao thông, động vật hoang dã, hay bất kỳ đối tượng nào có vị trí thay đổi theo thời gian.

- **Các kiểu dữ liệu trừu tượng (ADT) cho dữ liệu di động:** Hệ thống cung cấp các kiểu dữ liệu đặc biệt để biểu diễn chính xác và hiệu quả các thông tin về vị trí, thời gian, vận tốc và các thuộc tính khác của đối tượng di chuyển.
- **Tích hợp sâu với PostgreSQL và PostGIS:** MobilityDB tận dụng các cơ chế hiện có của PostgreSQL và PostGIS, bao gồm thao tác, lập chỉ mục, tổng hợp và tối ưu hóa, để cung cấp các tính năng mạnh mẽ cho việc quản lý và truy vấn dữ liệu di động.
- **Truy vấn dữ liệu di động bằng SQL:** Người dùng có thể sử dụng ngôn ngữ truy vấn SQL quen thuộc để thực hiện các truy vấn phức tạp trên dữ liệu di động, từ việc tìm kiếm các đối tượng trong một khu vực nhất định đến việc phân tích hành vi di chuyển của chúng.
- **Tuân thủ tiêu chuẩn OGC:** MobilityDB đáp ứng các tiêu chuẩn mở của Open Geospatial Consortium (OGC), đảm bảo khả năng tương tác và tích hợp với các hệ thống GIS khác.
- **Mã nguồn mở:** MobilityDB được phát hành dưới dạng mã nguồn mở, khuyến khích cộng đồng đóng góp và phát triển.
- **Mục tiêu ứng dụng thực tế:** MobilityDB hướng đến việc đưa các nghiên cứu về cơ sở dữ liệu đối tượng di chuyển vào ứng dụng thực tế, phục vụ cho các lĩnh vực như giao thông vận tải, quản lý đô thị, bảo tồn môi trường, v.v.

## 2. Các kỹ thuật

Công trình đã sử dụng nhiều kỹ thuật để phát triển MobilityDB, một cơ sở dữ liệu di động mở rộng từ PostgreSQL và PostGIS, nhằm giải quyết các thách thức trong việc quản lý dữ liệu đối tượng di chuyển. Các kỹ thuật chính được trình bày trong bài gồm:

### 2.1. Mở rộng kiểu dữ liệu

Hệ thống kiểu dữ liệu của MobilityDB được xây dựng dựa trên việc kết hợp kiểu dữ liệu cơ bản (base types) và kiểu dữ liệu thời gian (time types):

#### Base Types:

- int, float, bool, text: Biểu diễn các thuộc tính của đối tượng di chuyển.
- geometry(point) và geography(point): Biểu diễn tọa độ không gian của đối tượng di chuyển, sử dụng hệ tọa độ phẳng (planar) hoặc cầu (spherical).

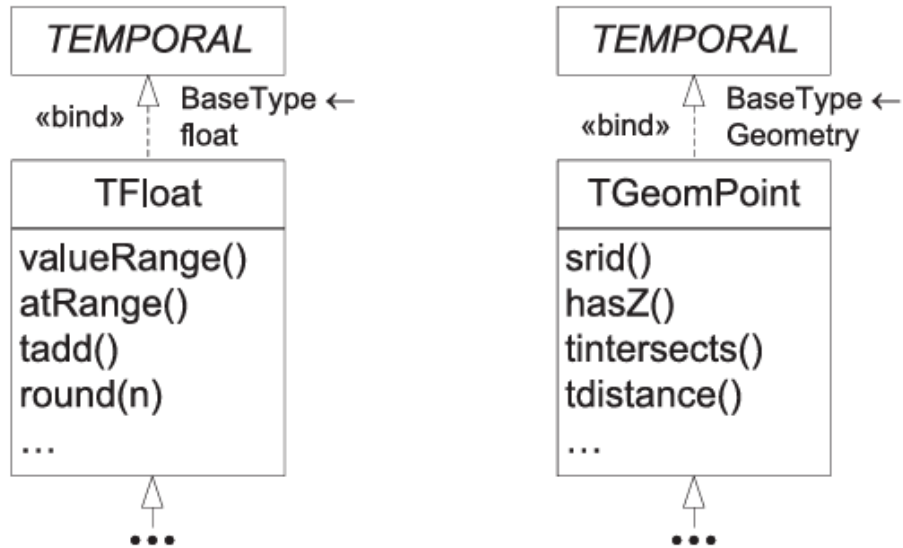
#### Time Types:

- timestamp: Biểu diễn một thời điểm cụ thể.
- timestampset: Tập hợp các thời điểm rời rạc.
- period: Một khoảng thời gian liên tục với giới hạn dưới và trên.
- periodset: Tập hợp các khoảng thời gian rời rạc và không liên kề.

**Temporal type:** được tạo thành từ việc kết hợp các base type với time type:

- tint: Giá trị số nguyên thay đổi theo thời gian.

- tgeompoint: Một điểm hình học di chuyển theo thời gian.
- tgeogpoint: Một điểm địa lý di chuyển theo thời gian.
- ttext: Chuỗi văn bản thay đổi theo thời gian.



Ví dụ về các Temporal Class

Hình minh họa này thể hiện **hệ thống phân cấp lớp (class hierarchy)** trong MobilityDB, mô tả cách các lớp dữ liệu **temporal** (thời gian) được tổ chức và liên kết với các kiểu dữ liệu cơ bản (base types). Hệ thống này cho thấy cách MobilityDB mở rộng các kiểu dữ liệu và cung cấp các phép toán đặc thù trên các đối tượng thay đổi theo thời gian.

### Chi tiết các thành phần

#### TEMPORAL

- Là lớp trừu tượng cha, cung cấp một khuôn mẫu chung cho tất cả các lớp temporal.
- TEMPORAL không hoạt động trực tiếp mà được kế thừa bởi các lớp cụ thể như **TFloat**, **TGeomPoint**.

#### BaseType

- Là kiểu dữ liệu cơ bản (ví dụ: **float**, **geometry**) mà các lớp temporal được liên kết (binding). Các kiểu temporal là sự mở rộng của những kiểu dữ liệu này để hoạt động trong miền thời gian.
- BaseType xác định kiểu giá trị cơ sở mà dữ liệu thời gian sẽ hoạt động trên đó:
  - Ví dụ: **TFloat** hoạt động trên kiểu **float**, còn **TGeomPoint** hoạt động trên kiểu **geometry**.

#### TFloat

- Là lớp con của TEMPORAL, biểu diễn các giá trị số thực thay đổi theo thời gian.
- Một số phương thức nổi bật:

- **valueRange()**: Trả về khoảng giá trị (min, max) của **TFloat** trong khoảng thời gian của nó.
- **atRange()**: Truy xuất giá trị tại một tập hợp các khoảng thời gian cụ thể.
- **tadd()**: Phép toán cộng thời gian (temporal addition), nghĩa là cộng các giá trị **TFloat** tại từng thời điểm.
- **round(n)**: Làm tròn giá trị **TFloat** đến **n** chữ số thập phân.

### **TGeomPoint**

- Là lớp con của **TEMPORAL**, biểu diễn các điểm hình học thay đổi theo thời gian.
- Một số phương thức nổi bật:
  - **srid()**: Trả về hệ tọa độ không gian (Spatial Reference System Identifier) của **TGeomPoint**.
  - **hasZ()**: Kiểm tra xem dữ liệu có chiều **Z** (3D) hay không.
  - **tintersects()**: Kiểm tra xem hai **TGeomPoint** có giao nhau tại bất kỳ thời điểm nào không.
  - **tdistance()**: Tính khoảng cách giữa hai **TGeomPoint** tại từng thời điểm.

Tóm lại, **TEMPORAL** cung cấp một cấu trúc chung, cho phép **MobilityDB** dễ dàng mở rộng thêm các kiểu temporal khác (như **TInt**, **TText**). Bên cạnh đó, các phương thức của **TEMPORAL** có thể được tái sử dụng hoặc ghi đè (override) trong các lớp con để phù hợp với kiểu dữ liệu cụ thể.

## **2.2. Nâng toán tử (Operator lifting)**

## **2.3. Đồng bộ hóa và chuẩn hóa**

### **2.3.1. Synchronization**

**Định nghĩa:** Synchronization trong **MobilityDB** đề cập đến quá trình đồng bộ hóa các sự kiện hoặc thông tin từ nhiều nguồn dữ liệu khác nhau về một điểm thời gian chung. Mục tiêu của quá trình này là đảm bảo rằng các dữ liệu được thu thập từ các hệ thống hoặc thiết bị khác nhau sẽ có thể được kết hợp và phân tích một cách chính xác và nhất quán.

#### **Mục đích:**

- **Đảm bảo tính chính xác:** Đồng bộ hóa giúp loại bỏ sự không nhất quán giữa các nguồn dữ liệu, đảm bảo rằng mọi sự kiện hoặc trạng thái đều phản ánh đúng thời điểm và tình huống xảy ra.
- **Phân tích đồng thời:** Đồng bộ hóa giúp việc phân tích dữ liệu trở nên mượt mà hơn khi các sự kiện từ các nguồn khác nhau có thể được xử lý và so sánh trực tiếp trên cùng một mốc thời gian.
- **Tăng cường tính khả dụng:** Đảm bảo rằng các hệ thống hoặc các thiết bị khác nhau có thể tương tác với nhau mà không gặp vấn đề về độ trễ hay mất mát dữ liệu.

#### **Giải thuật:**

- **Time Synchronization Algorithms:** Một trong các giải thuật phổ biến để đồng bộ hóa thời gian là sử dụng NTP (Network Time Protocol) để đồng bộ hóa đồng hồ hệ thống giữa các máy chủ. Ngoài ra, các phương pháp như clock skew adjustment cũng được sử dụng để điều chỉnh sự lệch giữa đồng hồ của các thiết bị.
- **Event-based Synchronization:** Sử dụng các sự kiện và dấu thời gian (timestamps) để đồng bộ hóa các sự kiện xảy ra tại các địa điểm khác nhau, qua đó giúp dữ liệu được tổ chức và phân tích trong một khung thời gian chung.

#### **Ứng dụng:**

- **Vận tải thông minh:** Trong các hệ thống giao thông thông minh, đồng bộ hóa dữ liệu từ các thiết bị cảm biến (camera, radar, GPS) là cần thiết để cung cấp thông tin chính xác về tình hình giao thông tại các điểm khác nhau trong cùng một thời điểm.
- **Theo dõi chuyển động:** Các hệ thống theo dõi chuyển động như trong lĩnh vực y tế hoặc thể thao cần đồng bộ hóa dữ liệu từ các cảm biến để theo dõi sự chuyển động của cơ thể qua thời gian.

**Ví dụ:** Một hệ thống giám sát giao thông sử dụng nhiều camera và cảm biến radar ở các điểm khác nhau. Để phân tích tình hình giao thông theo thời gian thực, tất cả dữ liệu từ các cảm biến này cần được đồng bộ hóa lại về cùng một mốc thời gian, đảm bảo rằng dữ liệu từ các nguồn khác nhau có thể được phân tích và so sánh chính xác.

### **2.3.2. Normalization**

**Định nghĩa:** Normalization trong MobilityDB là quá trình chuẩn hóa dữ liệu không gian và thời gian, nhằm giảm thiểu sự khác biệt giữa các đơn vị đo lường hoặc các đại lượng khác nhau. Trong bối cảnh MobilityDB, normalization thường liên quan đến việc chuẩn hóa dữ liệu không gian (vị trí địa lý) và thời gian để có thể so sánh và phân tích dữ liệu một cách nhất quán.

#### **Mục đích:**

- **Giảm thiểu sự khác biệt trong dữ liệu:** Bằng cách chuẩn hóa, các dữ liệu có thể được chuyển về cùng một phạm vi hoặc chuẩn để dễ dàng so sánh và phân tích.
- **Tăng cường tính đồng nhất:** Normalization giúp dữ liệu trở nên đồng nhất, giúp việc xử lý và phân tích không bị ảnh hưởng bởi các yếu tố ngoài ý muốn, chẳng hạn như đơn vị đo lường khác nhau.
- **Dễ dàng phân tích và trực quan hóa:** Dữ liệu đã được chuẩn hóa giúp việc trực quan hóa thông tin trở nên dễ dàng và hiệu quả hơn, đồng thời cũng giúp giảm thiểu khả năng sai sót trong phân tích.

#### **Giải thuật:**

- **Min-Max Normalization:** Quy trình chuẩn hóa dữ liệu bằng cách đưa tất cả các giá trị về một phạm vi nhất định (thường là  $[0, 1]$ ). Công thức



chuẩn hóa là:

$$X_{\text{norm}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}}$$

Đây là cách đơn giản để chuyển tất cả giá trị dữ liệu về phạm vi [0, 1], giúp tăng tính đồng nhất và dễ dàng so sánh.

- **Z-score Normalization (Standardization):** Sử dụng giá trị trung bình và độ lệch chuẩn của dữ liệu để chuẩn hóa. Công thức chuẩn hóa là:  
$$Z = \frac{X - \mu}{\sigma}$$
  
Trong đó,  $\mu$  là trung bình và  $\sigma$  là độ lệch chuẩn của tập dữ liệu. Phương pháp này giúp dữ liệu có phân phối chuẩn (mean = 0 và std = 1).
- **Normalization of Spatial Data:** Đối với dữ liệu không gian (ví dụ: tọa độ GPS), một phương pháp chuẩn hóa phổ biến là sử dụng global coordinate systems như UTM (Universal Transverse Mercator) hoặc chuyển đổi giữa các hệ tọa độ khác nhau để đảm bảo tính đồng nhất và khả năng so sánh.

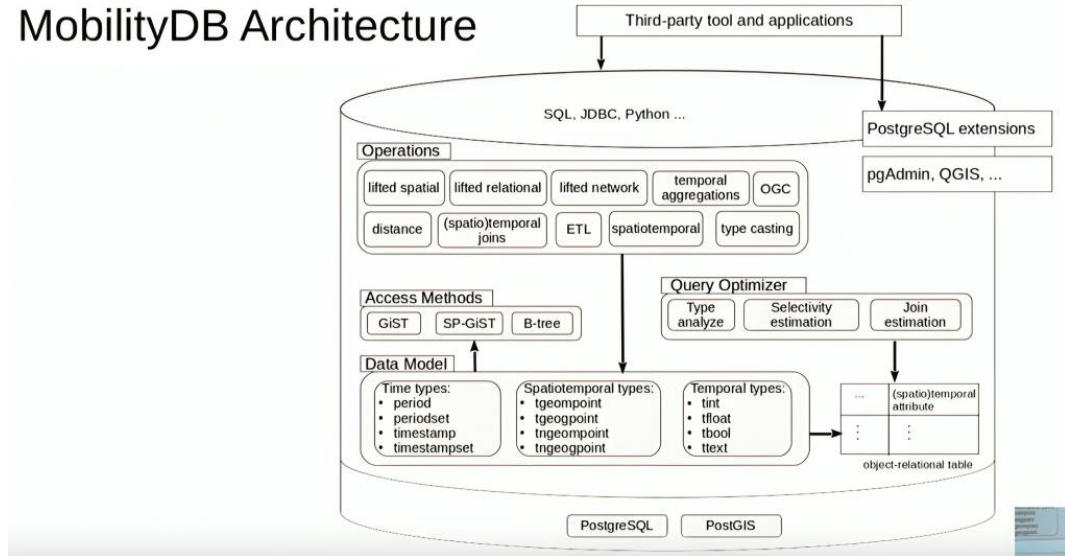
#### Ứng dụng:

- **Phân tích di chuyển:** Trong các hệ thống theo dõi di chuyển, normalization giúp chuẩn hóa dữ liệu vị trí (từ các hệ tọa độ khác nhau) và thời gian để các sự kiện hoặc hành vi có thể được so sánh và phân tích một cách chính xác.
- **Cơ sở dữ liệu không gian:** Các cơ sở dữ liệu như MobilityDB thường phải xử lý lượng lớn dữ liệu không gian và thời gian. Việc chuẩn hóa dữ liệu là bước quan trọng để đảm bảo rằng các truy vấn và phân tích được thực hiện chính xác.

**Ví dụ:** Trong một hệ thống theo dõi hành trình của các phương tiện giao thông, các tọa độ GPS từ các cảm biến có thể đến từ nhiều hệ tọa độ khác nhau. Để phân tích và so sánh các vị trí, dữ liệu cần được chuẩn hóa về cùng một hệ tọa độ và thời gian, đảm bảo việc theo dõi và phân tích được thực hiện chính xác.

### 3. Cấu trúc của MobilityGB

# MobilityDB Architecture



## 3.1. Third-party tools and applications

MobilityDB hỗ trợ tích hợp với nhiều công cụ và ứng dụng của bên thứ ba như SQL, JDBC, và Python, giúp người dùng dễ dàng truy vấn và quản lý dữ liệu. Đồng thời, hệ thống còn hỗ trợ các tiện ích phổ biến như pgAdmin và QGIS, cho phép trực quan hóa dữ liệu không gian và thời gian một cách hiệu quả.

## 3.2. PostgreSQL Extensions

MobilityDB được xây dựng dựa trên nền tảng PostgreSQL và mở rộng khả năng của PostGIS để hỗ trợ các loại dữ liệu spatio-temporal. Các tiện ích mở rộng này cung cấp cho người dùng các công cụ mạnh mẽ để quản lý dữ liệu không gian (spatial) và thời gian (temporal).

## 3.3. Data Model

Tầng mô hình dữ liệu của MobilityDB bao gồm:

- Time types: Các kiểu dữ liệu về thời gian như **period**, **periodset**, **timestamp**, và **timestampset**. Đây là các thành phần quan trọng giúp định nghĩa và tổ chức dữ liệu theo khía cạnh thời gian.
- Spatio-temporal types: Các kiểu dữ liệu như **tgeopoint**, **tfloat**, **tint**, và **ttext**, được sử dụng để lưu trữ dữ liệu có thuộc tính không gian và thời gian.
- Temporal types: Các kiểu dữ liệu thuần thời gian như **tint** (số nguyên theo thời gian), **tfloat** (số thực theo thời gian), **tbool** (boolean theo thời gian), và **ttext** (chuỗi văn bản theo thời gian).

## 3.4. Access Methods

MobilityDB hỗ trợ các phương pháp truy cập tiên tiến như GiST, SP-GiST, và B-tree, giúp tối ưu hóa việc tìm kiếm và xử lý dữ liệu lớn. Các phương pháp này rất hữu ích trong việc lập chỉ mục (indexing) và truy vấn dữ liệu không gian và thời gian.

## 3.5. Query Optimizer

MobilityDB sử dụng bộ tối ưu hóa truy vấn (Query Optimizer) để tăng hiệu quả trong việc xử lý và phân tích dữ liệu. Bộ tối ưu hóa này bao gồm:

- Type analysis: Phân tích kiểu dữ liệu để chọn phương pháp xử lý phù hợp.
- Selectivity estimation: Dự đoán tính chọn lọc của dữ liệu, giúp giảm tải hệ thống khi truy vấn.
- Join estimation: Tối ưu hóa các phép nối (join) giữa các bảng, đặc biệt là với dữ liệu không gian và thời gian.

### 3.6. Operations

MobilityDB hỗ trợ một loạt các phép toán mạnh mẽ, bao gồm:

- Lifted spatial và relational operations: Mở rộng các phép toán không gian và quan hệ để hỗ trợ dữ liệu spatio-temporal.
- Temporal aggregations: Thực hiện các phép tổng hợp dữ liệu theo thời gian.
- ETL (Extract, Transform, Load): Hỗ trợ xử lý và chuyển đổi dữ liệu từ các nguồn khác nhau.
- Spatio-temporal joins: Cho phép thực hiện các phép nối giữa các bảng dựa trên thuộc tính không gian và thời gian.

### 3.7. Ứng dụng và tích hợp

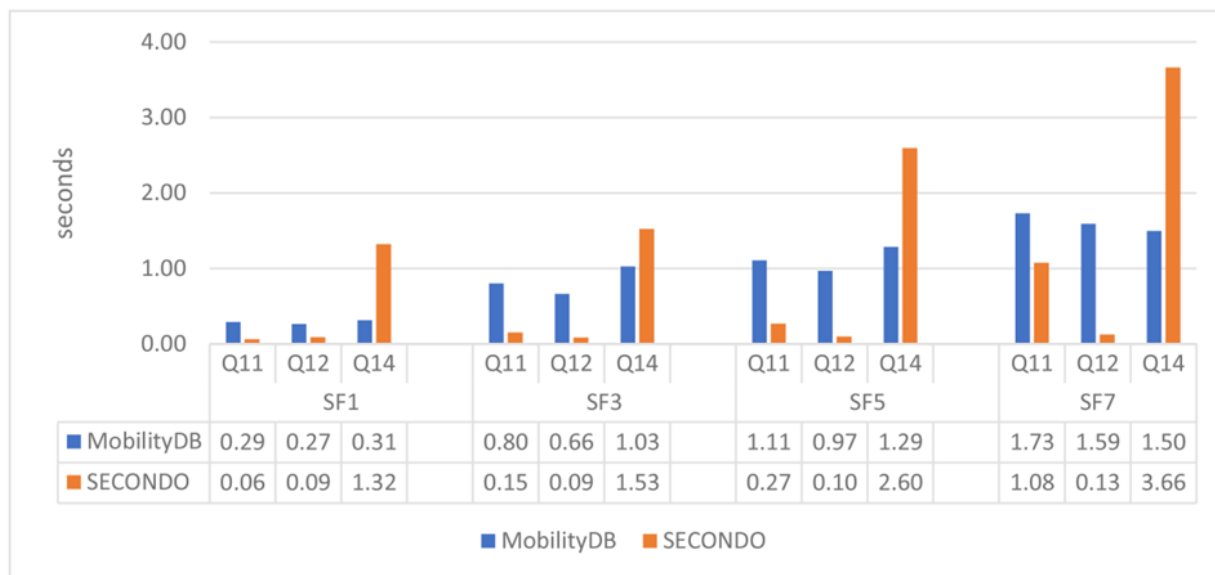
MobilityDB không chỉ hỗ trợ lưu trữ và truy vấn dữ liệu mà còn dễ dàng tích hợp với các ứng dụng GIS như QGIS, mang lại khả năng trực quan hóa dữ liệu trực tiếp trên bản đồ. Điều này rất hữu ích trong các lĩnh vực như quản lý giao thông, theo dõi vận chuyển, và phân tích hành vi di chuyển.

## 4. Đánh giá

Phần này tóm tắt lại việc sử dụng các số liệu hiệu suất để so sánh giữa MobilityDB và hệ thống, giải pháp cơ sở dữ liệu khác có cùng chức năng như SECONDO. Các số liệu này giúp đánh giá chính xác khả năng của PostgreSQL/PostGIS trong việc xử lý dữ liệu không gian (spatial data) và dữ liệu quan hệ (relational data). Dưới đây là các số liệu thường gặp và ý nghĩa của chúng trong phần Benchmark.

Thời gian thực thi là số liệu cơ bản nhất được sử dụng trong Benchmark. Nó đo lường khoảng thời gian cần thiết để thực hiện một truy vấn hoặc phép toán không gian. Thời gian thực thi ngắn hơn cho thấy hệ thống tối ưu hơn về xử lý dữ liệu. Và đây cũng là chỉ số quan trọng khi đánh giá hiệu suất tổng thể.

Bên cạnh đó, phần này còn xem xét tới khả năng mở rộng của dữ liệu, nó đo lường hiệu suất của hệ thống khi kích thước dữ liệu tăng lên.

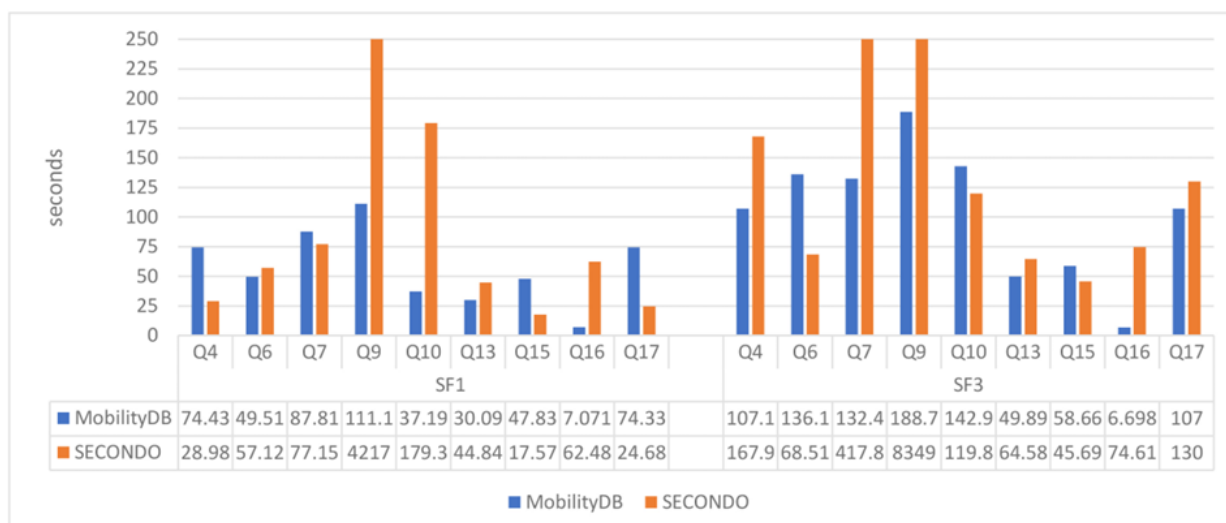


### Tốc độ truy vấn của MobitilyDB so với SECONDO

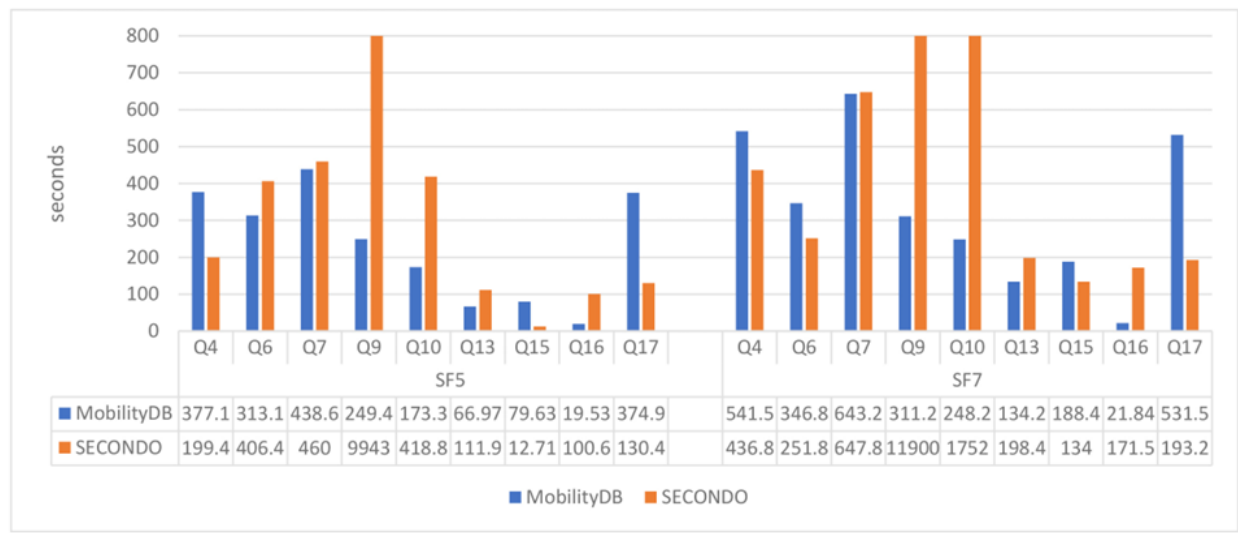
Khả năng mở rộng (Scalability)\*\*

- \*\*Ý nghĩa\*\*:

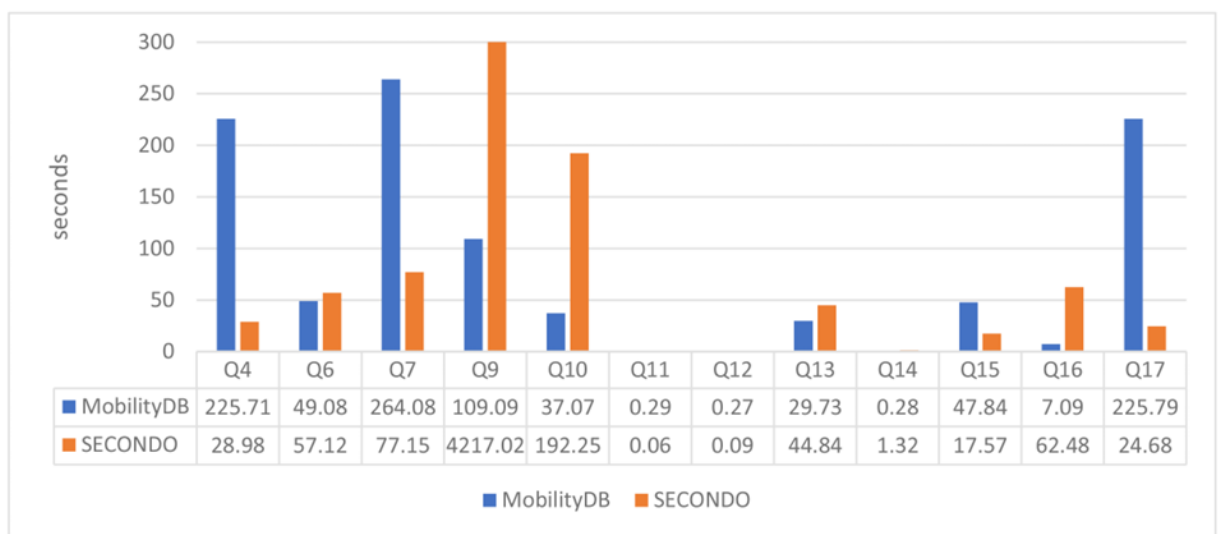
- Hệ thống có khả năng mở rộng tốt sẽ duy trì thời gian thực thi hợp lý khi số lượng đối tượng tăng từ hàng nghìn đến hàng triệu.
- Điều này rất quan trọng trong các ứng dụng GIS lớn (quy hoạch đô thị, phân tích giao thông).



Scale factors 1 (300K trips) and 3 (870K trips)



Scale factors 5 (1.5M trips) and 7 (2M trips)

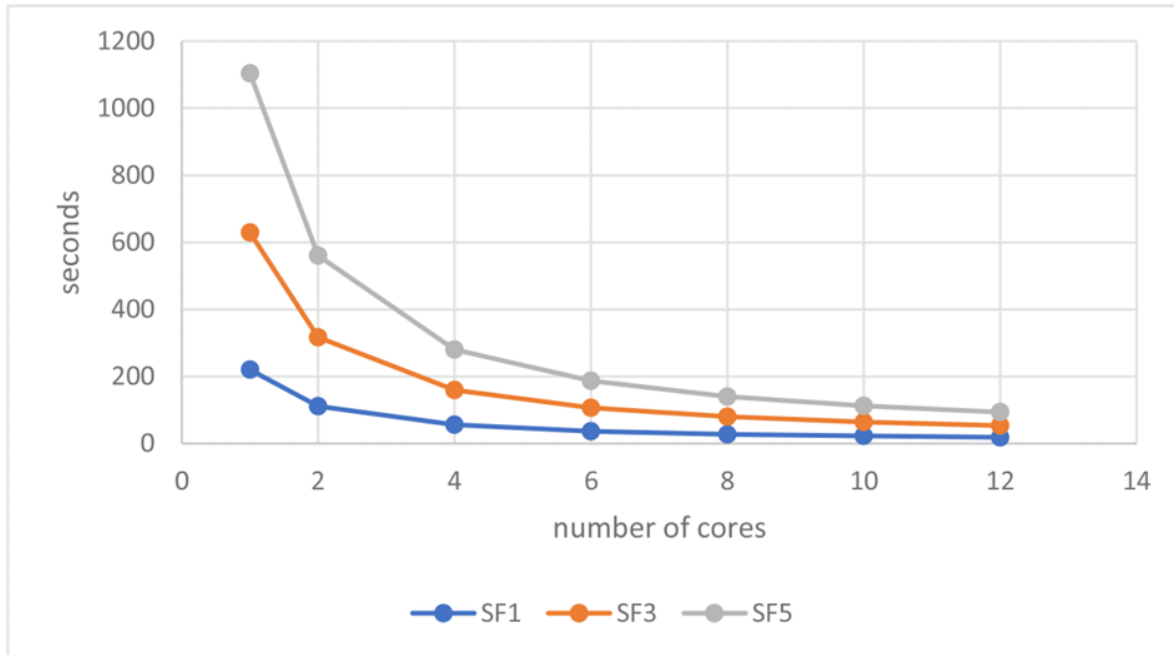


Benchmark results on scale factor 0.02 (62K trips).

### 1.3. Sử dụng tài nguyên (Resource Utilization)

- **\*\*Ý nghĩa\*\***:

- Hệ thống tối ưu sẽ sử dụng tài nguyên hiệu quả hơn, giảm tải cho phần cứng.
- Điều này đặc biệt quan trọng khi triển khai hệ thống trên các máy chủ với tài nguyên hạn chế.



Performance w.r.t. number of cores

### ### \*\*3. Kết luận từ Benchmark\*\*

Các số liệu so sánh trong phần Benchmark thường cho thấy:

- PostgreSQL + PostGIS vượt trội về hiệu suất xử lý dữ liệu không gian, đặc biệt khi sử dụng chỉ mục không gian.
- PostgreSQL/PostGIS là lựa chọn lý tưởng cho các dự án cần xử lý dữ liệu không gian quy mô lớn hoặc yêu cầu tích hợp với dữ liệu quan hệ truyền thống.
- Các hệ thống thương mại như Oracle Spatial có lợi thế về tính năng, nhưng chi phí cao và không luôn tối ưu cho dữ liệu lớn.
- MySQL Spatial phù hợp cho các ứng dụng nhỏ, nhưng không thể cạnh tranh về tính năng và hiệu suất.

Những số liệu này giúp người dùng lựa chọn hệ thống phù hợp với nhu cầu của họ, tùy thuộc vào quy mô dữ liệu, ngân sách, và yêu cầu kỹ thuật.