

# ORDBMS상에서 Trajectory 정보의 저장과 관리 기법

## Storing and Managing of Trajectory Information on ORDBMS<sup>†</sup>

양평우\* · 정용희 · 남광우 · 이연식

Pyoung Woo Yang\* · Yong Hi Jeong, Kwang Woo Nam · Yonsik Lee

군산대학교 컴퓨터정보공학과

{didvuddn, jyh2380, kwnam, yslee}@kunsan.ac.kr

### 요 약

최근 모바일 장비 및 시스템의 발달로 위치정보를 이용한 많은 서비스들이 나오고 있다. 이러한 위치정보 서비스와 맞물려 DB시스템들도 위치 정보의 저장을 지원하게 되었다. 하지만 현재의 DB시스템들은 위치 정보의 저장은 지원을 하지만, 이동 객체의 궤적을 저장을 지원 하지는 않고 있다. 본 논문에서는 공간 데이터베이스로 많이 활용되고 있는 PostGIS에서 이동 궤적을 저장 할 수 있도록 확장한 시스템을 설계하였다.

## 1. 서론

최근 스마트폰 및 모바일 장비의 발달로 위치 정보의 획득이 용이해지고, 이를 이용한 서비스에 대한 관심이 많이 늘어나고 있다. 하나의 이동 객체에 대한 정보를 저장하는데에는 스마트폰이나 모바일 장비만으로는 부족하기 때문에 대용량의 DBMS를 사용하여야 한다. 본 논문의 시스템은 이동 객체의 위치, 시간을 효율적으로 저장, 관리 할 수 있는 기법을 제시하였다.

## 2. Trajectory 시스템 설계

### 2.1. Trajectory DB 요소

PostGIS에서는 지원되는 기본데이터에는 trajectory 데이터를 저장할 수 있는 데이터 형태가 없기 때문에 본 시스템에서는 데이터 타입을 새로 tpoint라 정의하였다.(그림 1) 이 tpoint에는 객체의 움직임 정보를 저장할 point(GPS데이터)와 데이터의 시간을 저장하게 된다.

```
CREATE TYPE tpoint(  
    p geometry,  
    ptime timestamp with time zone  
);
```

그림 1. tpoint type

본 시스템에 사용되는 테이블은 총 3종류로 구성되어 있다. 첫 번째는 사용자가 객체에 대한 정보를 삽입할 user\_defined 테이블이다.(그림 2) 이 테이블에는 사용자가 입력한 객체에 대한 정보와 객체의 trajectory데이터를 저장하기 위한 trajectory\_column이 추가되어야 한다.

```
CREATE Table taxi  
(  
    taxi_id INTEGER,  
    taxi_number VARCHAR,  
    taxi_model VARCHAR,  
    taxi_driver VARCHAR,  
);
```

그림 2. user\_defined table의 예

<sup>†</sup> 본 연구는 국토해양부 첨단도시기술개발사업-지능형국토정보기술혁신사업과제의 연구비 지원(07국토정보C05) 및 산업자원부와 한국산업기술진흥원의 지역혁신인력양성사업으로 수행된 연구결과임

두 번째는 trajectory\_segtable로써 user\_defined 테이블을 생성하면 자동 생성되는 테이블이다.(그림 3) 이 테이블에는 객체가 이동한 궤적(trajectory)을 저장하게 된다. 각 객체는 개개의 mpid를 갖게 되고 각 mpid마다 수많은 데이터가 삽입이 되므로 segid를 만들어 순서를 쉽게 구별할 수 있어야 한다.

```
CREATE TABLE trajectory_segtable
(
    mpid integer,
    segid integer,
    next_segid integer,
    before_segid integer,
    mpcount integer,
    rect geometry,
    start_time timestamp with time zone,
    end_time timestamp with time zone,
    tpsseg tpoint[]
);
```

그림 3. trajectory\_segtable

세 번째로는 trajectory\_column에 대한 정보를 갖게 될 trajectory\_columns 테이블이다.(그림 4) 이 테이블에는 사용자가 만든 trajectory\_column에 대한 여러 가지 정보를 저장하게 된다.

```
CREATE TABLE trajectory_columns
(
    f_table_catalog VARCHAR(256) NOT NULL,
    f_table_schema VARCHAR(256) NOT NULL,
    f_table_name VARCHAR(256) NOT NULL,
    f_trajectory_column VARCHAR(256) NOT NULL,
    f_trajectory_segtable_name VARCHAR(256) NOT NULL,
    trajectory_compress VARCHAR(256),
    coord_dimension INTEGER,
    srid INTEGER,
    "type" character varying(30),
    f_segtableoid character varying(256) NOT NULL,
    f_sequence_name character varying(256) NOT NULL,
    tpsseg_size integer
);
```

그림 4. trajectory\_columns table의 예

## 2.2. Trajectory SQL의 설계

insert는 사용자가 데이터 입력을 하게 되면 user\_defined테이블의 trajectory\_columns 테이블에 있는 데이터를 이용하여 입력한 객체의 trajectory 데이터가 어느 trajectory\_segtable에 입력이 되어야 하는지 찾고 해당 입력한 데이터가 해당 trajectory\_segtable의 mpid와 segid를 계산하여 삽입을 해줘야 한다.

delete는 2가지의 경우로 나눌수 있다.

첫 번째 경우는 사용자가 객체의 정보를 지울때이다. 이 경우에는 객체의 정보를 삭제하면서 동시에 trajectory\_segtable에 있는 객체의 정보도 같이 지워줘야 한다. 두 번째 경우는 사용자가 기존에 있는 trajectory데이터에서 어느 일정 부분만을 삭제해줄 때 이다. 이 경우에는 해당 데이터를 삭제하면서 데이터가 삭제가 row에 더 이상의 trajectory데이터가 남아있지 않다면 해당 row를 삭제해주고 해당 row의 앞 row의 segid와 뒤 row의 segid를 맞게 수정해 줘야 한다.

modify의 경우에는 먼저 사용자가 수정하길 원하는 데이터의 시간과 사용자가 입력한 수정할 시간이 일치하는지 확인을 해야하고 일치할 경우 데이터 수정을 해야한다. 데이터 수정의 경우 한 row의 일정부분 데이터가 삭제가 되고 해당 row에 다 들어가지 않는 데이터가 삽입이 될 경우가 발생 할 수 도 있는데 이 경우에는 새로운 row를 생성하여 나머지 데이터들을 삽입해주고 segid를 알맞게 수정해줘야 한다.

select도 두가지의 경우가 있다. 첫 번째는 사용자가 객체의 id와 시간값을 가지고 검색을 할때와 객체의 id와 위치값을 가지고 검색을 경우이다. 첫 번째 경우일 때에는 각 row의 start, end time를 가지고 데이터를 먼저 필터링 한다음, 필터링 된 데이터의 tpoint값을 가져와서 일치하는 값을 가져오면 된다. 두 번째의 경우 각 row의 rect값을 가져와서 데이터를 필터링 한 다음, 필터링 된 데이터의 tpoint값을 가져와서 일치하는 값을 가져오면 된다.

## 3. 결론

이 논문은 PostGIS에서 trajectory 정보의 저장과 관리하는 기법을 개발 하였다. 이 시스템은 위치정보를 기반으로 이동 객체의 궤적을 활용한 서비스나 응용에 사용 될 수 있다.

## 참고문헌

- [1] Kostas Patoruuompas, Timos Sellis,  
“Managing Trajectories of Moving  
Objects as Data Stream“, Proceedings  
of the Second Workshop on  
Spatio-Temporal Database  
Management (STDBM’ 04), Toronto,  
Canada, August 30th, 2004
- [2] <http://www.postgresql.org/docs/8.4/static/release-8-4-3.html>