

Особенности хранения пространственных данных в современных СУБД

Васильева Полина Александровна

Приамурский государственный университет имени Шолом-Алейхема

Студент

Козич Виталий Геннадьевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Глаголев Владимир Александрович

Приамурский государственный университет имени Шолом-Алейхема

к.г.н., доцент кафедры информационных систем, математики и методик обучения

Аннотация

В данной статье описаны ключевые особенности пространственных типов данных и их хранение в различных СУБД.

Ключевые слова: пространственные данные, система управления базами данных, информационные технологии, типы данных, MySQL, SQL Server, PostgreSQL, SQL.

Features of storage of spatial data in modern DBMS

Vasilyeva Polina Alexandrovna

Sholom-Aleichem Priamursky State University

Student

Kozich Vitaliy Gennadievich

Sholom-Aleichem Priamursky State University

Student

Glagolev Vladimir Alexandrovich

Sholom-Aleichem Priamursky State University

candidate of geographical sciences, Associate Professor of the Department of Information System, Mathematics and teaching methods

Abstract

This article describes the key features of spatial data types and their storage in various DBMSs.

Keywords: spatial data, database management system, information technology, data types, MySQL, SQL Server, PostgreSQL, SQL.

В настоящее время достаточно большое количество пользователей тем или иным способом взаимодействуют с базами данных. Для этого разработали специальное программное обеспечение, которое называется система управления базами данных (СУБД). Но, если углубиться в вопрос хранения данных в БД, то люди, которые, непосредственно, работают с геоинформационными системами, сталкиваются с таким понятием как пространственные данные. Они представляют собой сведения о физическом расположении и форме геометрических объектов. Такими объектами могут быть точки или более сложные объекты: страны, дороги, озера и т.д. В данном исследовании рассмотрим работу с пространственными данными в современных и популярных системах управления баз данных (spatial database): SQL Server, MySQL и PostgreSQL.

Исследованием пространственных данных также занималось огромное количество людей. Тема на настоящее время особенно актуальна. Например, П.Бобков опубликовал статью на одном из интернет-ресурсов [1]. Студенты Северо-Кавказского федерального университета добавили информацию в один из файловых интернет-архивов студентов [2]. Ю.Н.Артеменко адаптировал на русский язык книгу «MySQL Справочник по языку», написанную американскими специалистами [3]. А.Коротков выпустил презентацию и видео, где рассказал о работе с пространственными данными в PostgreSQL [4]. О введении в пространственные данные в SQL Server подробно написано на одноименном сайте [5]. Было приведено сравнение существующих наиболее популярных СУБД, работающих с этими данными на ресурсе «GISLAB» [6].

Microsoft SQL Server — реляционная СУБД, которая, очевидно, из названия была разработана корпорацией Microsoft. В качестве основного языка запросов используется Transact-SQL, который был создан совместными усилиями Sybase и Microsoft. Реализован на стандарте ANSI/ISO по расширенному структурированному языку запросов SQL. Используется для работы размером от очень мелких до крупных баз данных. Сильный конкурент с другими СУБД на данном рынке.

В данном исследовании рассмотрим работу SQL Server с пространственными данными. Здесь есть географические (geography) и геометрические (geometry) типы пространственных данных.

Тип данных geography ориентирован на информацию для шарообразной земли. Такая модель использует кривизну земли при расчетах, т.е. когда данные хранятся в виде широты и долготы.

Тип данных geometry, наоборот, ориентирован на модель плоской земли. В данном случае это плоская проекция, которая не принимает в расчет кривизну поверхности земли. Удобно в использование, если описываются короткие расстояния.

Типы geography и geometry генерируются из векторных объектов, заданных в форматах Well-Known Text или Well-Known Binary. Они нужны для перенесения пространственных данных, описанных в функциях открытого геопространственного консорциума спецификаций SQL. На

рисунке 1 перечислены типы векторных объектов, которые поддерживаются данной СУБД.

Объект	Описание
Point	Расположение.
MultiPoint	Набор расположений.
LineString	Набор из нуля или более точек, соединенных линиями.
MultiLineString	Набор прямых (linestring).
Polygon	Непрерывная область, описываемая набором закрытых прямых (linestring).
MultiPolygon	Набор многоугольников (polygon).
GeometryCollection	Собрание типов geometry.

Рисунок 1. Типы векторных объектов

Чтобы создать тип geography с векторными объектами, в первую очередь надо объявить тип geography в сценарии T-SQL (см. рис. 2). Затем нужно вызвать один из методов, показанных на рисунке 3, и передать строку символов своего векторного объекта и идентификатор пространственного отношения (SRID). SRID был разработан для картографии, геодезии и хранения геодезических данных, задает для географических расчетов определенный тип эллипсоида. Это достаточно важный момент, т.к. наша планета не является идеальным шаром. SQL Server выполняет расчеты только на идентичных SRID.

```

DECLARE @geo1 geometry
SELECT @geo1 = geometry::STGeomFromText('POINT (3 4)', 0)
PRINT @geo1.ToString()

DECLARE @geo2 geometry
SELECT @geo2 = geometry::Parse('POINT(3 4 7 2.5)')
PRINT @geo2.STX;
PRINT @geo2.STY;
PRINT @geo2.Z;
PRINT @geo2.M;

DECLARE @geo3 geography;
SELECT @geo3 = geography::STGeomFromText(
    'LINESTRING(47.656 -122.360, 47.656 -122.343)', 4326);
SELECT @geo3.ToString();

--Results
--POINT (3 4)
--3
--4
--7
--2.5

DECLARE @gx geometry;
SET @gx = geometry::STPolyFromText(
    'POLYGON ((5 5, 10 5, 10 10, 5 5))', 0);
PRINT @gx.ToString();
--Results
--POLYGON ((5 5, 10 5, 10 10, 5 5))

```

Рисунок 2. Пример сценария T-SQL

Метод	Описание
STGeomFromText	Создает элементы geography из текста на вводе.
STPointFromText	Создает элемент geography Point из текста на вводе.
STMPointFromText	Создает элемент geography MultiPoint из текста на вводе.
STLineFromText	Создает элемент geography LineString из текста на вводе.
STMLineFromText	Создает элемент geography MultiLineString из текста на вводе.
STPolyFromText	Создает элемент geography Polygon из текста на вводе.
STMPolyFromText	Создает элемент geography MultiPolygon из текста на вводе.
STGeomCollFromText	Создает элемент geography GeometryCollection из текста на вводе.

Рисунок 3. Методы создания элементов

Очень важно знать о некоторых ключевых отличиях между типами geography и geometry. В типе данных geometry, расстояния и площади задаются в тех же единицах измерения, что и координаты объектов. В типе geography используются эллиптические координаты, заданные в градусах широты и долготы. Сценарий T-SQL вычисляет расстояние между точками (POINTS) (90 0) и (90 180) (см. рис. 4). Обе они находятся на северном полюсе. Значит расстояние в случае GEOGRAPHY должно равно 0, а в GEOMETRY расстояние оказывается 180.

```

DECLARE @g1 GEOMETRY, @g2 GEOMETRY, @g3 GEOGRAPHY, @g4 GEOGRAPHY
SELECT @g1 = GEOMETRY::STGeomFromText('POINT (90 0)', 0)
SELECT @g2 = GEOMETRY::STGeomFromText('POINT (90 180)', 0)

SELECT @g3 = GEOGRAPHY::STGeomFromText('POINT (90 0)', 4326)
SELECT @g4 = GEOGRAPHY::STGeomFromText('POINT (90 180)', 4326)
SELECT @g2.STDistance(@g1) AS 'GEOMETRY',
       @g4.STDistance(@g3) AS 'GEOGRAPHY';

--Results
--GEOMETRY          GEOGRAPHY
-----
--180                0

```

Рисунок 4. Пример сценария T-SQL

Ориентация пространственных данных тоже различается. В системе geometry, ориентация многоугольника не является важным фактором. А в типе данных geography используется модель, в которой необходимо определять ориентацию многоугольника. Нужно это для того, чтобы определить, например, к какому полушарию относится объект. Каждый элемент должен относиться только к одному полушарию. Большие объекты недопустимы и вызовут ошибку ArgumentException. Geography возвращает NULL, если результаты методов не относятся к одному полушарию.

MySQL — реляционная СУБД с открытым кодом, разработанная и поддерживаемая корпорацией Oracle. Один из плюсов является механизм репликации. Рассмотрим ключевые особенности данного продукта в работе с пространственными данными.

Здесь имеются типы данных, соответствующие классам OpenGIS. Некоторые из них хранят единственное геометрическое значение:

1. GEOMETRY
2. POINT
3. LINESTRING
4. POLYGON

GEOMETRY хранит геометрические значения абсолютно любого типа. Остальные типы имеют единственное значение, такие как POINT, LINESTRING и POLYGON ограничивают свои значения определенным геометрическим типом.

Некоторые типы данных хранят коллекции значений:

1. MULTIPOINT
2. MULTILINESTRING
3. MULTIPOLYGON
4. GEOMETRYCOLLECTION

GEOMETRYCOLLECTION дают возможность сохранения коллекции объектов любого типа. Остальные типы коллекций, такие как MULTIPOINT, MULTILINESTRING, MULTIPOLYGON и GEOMETRYCOLLECTION имеют ограничения коллекциями также с определенным типом геометрических объектов.

Данная СУБД имеет целый ряд функций, принимающих WKT-значения и SRID. Они дают возможность возвращать значение соответствующей геометрии. Функция GeomFromText() принимает WKT-значение любого геометрического типа. Также MySQL обеспечивает отдельные для каждого типа функции-конструкторы, которые необходимы для создания значений каждого типа геометрических объектов.

Система может предложить такой набор функций, который в качестве входных параметров принимает значения BLOB в WKB-формате и также SRID. Они могут вернуть значение соответствующей геометрии. Функция GeomFromWKT () имеет те же возможности что и функция, описанная выше.

MySQL позволяет очень эффективно управлять пространственными данными, но возможность создания пространственных индексов достигается лишь с помощью таблиц MyISAM, которые не поддерживают транзакции и ACID. Работать возможно через таблицы InnoDB, обладающие полной поддержкой ACID, но лишается возможность использования пространственных индексов. Функции пространственного отношения реализованы, в основном, для минимальных ограничивающихся прямоугольников.

PostgreSQL — свободная объектно-реляционная СУБД. Является сильным конкурентом другим продуктам на рынке своего сегмента. В данном исследовании рассмотрим расширение PostGIS, которое может хранить в базе географические данные или геометрию объектов. Плюсом является открытость и доступность, а также многопользовательская работа с одним и тем же слоем. Главным достоинством PostGIS является возможность пользоваться языком SQL совместно с различными пространственными функциями и операторами.

При генерировании пространственной БД в данном расширении автоматически создаются две таблицы: `SPATIAL_REF_SYS` и `GEOMETRY_COLUMNS`. Они создаются в соответствии со спецификацией, о которой было сказано при обзоре SQL Server.

Таблица `GEOMETRY_COLUMNS` сохраняет информацию о таблицах БД, содержащих пространственные данные. Её заполнение осуществляется собственноручно, либо как следствие выполнения особой процедуры `OGC AddGeometryColumn()`.

Таблица `SPATIAL_REF_SYS` включает в себя идентификаторы и описания систем координат, которые используются в пространственной БД. Одним из полей является поле `SRID`, о котором говорилось неоднократно в данной статье.

Ниже приведет пример запроса, в котором можно кратко охарактеризовать ключевые функции (см. рис. 5).

```
create table points ( pt geometry, name varchar );

insert into points values ( 'POINT(0 0)', 'Origin' );
insert into points values ( 'POINT(4 0)', 'X Axis' );
insert into points values ( 'POINT(0 3)', 'Y Axis' );

select name, ST_AsText(pt), ST_Distance(pt, 'POINT(4 3)') from points;
```

Рисунок 5. SQL-запрос

В результате создана таблицу `points`, включающая два поля: поле `pt` типа `geometry` и поле `name` типа `varchar`, после чего добавлены три записи, содержащих информацию о точках. Осуществлена выборка с использованием функций PostGIS: `ST_Distance()` и `ST_AsText()`. В качестве параметра обе функции используют объект типа `geometry`. Функция `ST_Distance()` вычисляет расстояние между двумя точками плоскости, а `ST_AsText()` дает возможность вернуть геометрию объекта в уже хорошо известном текстовом формате `Well-Known Text`.

В качестве заключения можно отметить, что были рассмотрены три системы управления базами данных, которые работают с пространственными данными, а также кратко описаны особенности хранения этих самых данных. Данные обзор и исследование помогут подробнее ознакомить студента с базами данных, в частности с работой различных географических и геометрических данных в них. Все системы, рассмотренные в данной статье, являются на текущее время самыми продвинутыми на рынке в работе с пространственной информацией.

Библиографический список

1. Пространственные типы данных // Лаборатория обработки изображений URL: http://loi.sccc.ru/gis/data_model/typs.htm (дата обращения: 20.01.2018).
2. Пространственные данные и пространственная информация // StudFiles URL: <https://studfiles.net/preview/4603447/> (дата обращения: 20.01.2018).
3. Артеменко Ю.Н. MySQL Справочник по языку. М.: Вильямс, 2005.
4. Работа с пространственными данными в PostgreSQL // PGConf.Russia URL: <https://pgconf.ru/en/2015/89356> (дата обращения: 20.01.2018).
5. Введение в пространственные данные // SQL Server URL: <http://serversql.ru/vvedenie-v-prostranstvennye-dannye.html> (дата обращения: 20.01.2018).
6. Место PostGIS/PostgreSQL среди СУБД с поддержкой пространственных данных // Географические информационные системы и дистанционное зондирование URL: <http://gis-lab.info/qa/postgis-compare.html> (дата обращения: 20.01.2018).