СРАВНЕНИЕ POSTGRESQL И MONGODB

ЗВЕРКОВА Ксения

Технический Университет Молдовы

Аннотация*: Статья посвящена сравнению реляционной базы данных PostgreSQL и нереляционной базы данных MongoDB. Рассказано об организации и управлении данными, об отношениях между сущностями. Рассмотрено, когда подходят или не подходят в использовании PostgreSQL и MongoDB.*

Ключевые слова: *PostgreSQL, MongoDB, СУБД, таблицы, документы, управление данными, SQL, отношения.*

1. Введение

В течение десятилетий реляционная база данных служила основой для большинства приложений. Одной из широко используемых реляционных баз данных является PostgreSQL - наиболее развитая из открытых СУБД в мире. Однако в последнюю декаду гибкие методологии разработки требуют непрерывной интеграции новых функций и постоянных изменений в базовых моделях данных, объем которых катастрофически возрос. Разработчики все чаще работают с полуструктурированными и неструктурированными данными, а организации ищут распределенные, масштабируемые архитектуры. В результате появились нереляционные или NoSQL-базы данных, такие как MongoDB, для удовлетворения требований новых приложений.

2. Что представляют собой PostgreSQL и MongoDB

PostgreSQL

PostgreSQL - это свободно распространяемая объектно-реляционная система управления базами данных, являющаяся реальной альтернативой коммерческим базам данных. Современный проект PostgreSQL ведет происхождение из проекта Postgres, который разрабатывался в 1980-х под руководством Майкла Стоунбрейкера, профессора Калифорнийского университета в Беркли.

Эта система управления данными представлена в виде отношений. Отношение - это математически точное обозначение таблицы.

У PostgreSQL есть функции для хранения неструктурированных документов, но она обычно используется для хранения и запроса высокоструктурированных реляционных данных.

MongoDB

MongoDB - это открытая, нереляционная база данных, разработанная MongoDB, Inc.

В отличие от реляционных баз данных MongoDB предлагает документо-ориентированную модель данных, благодаря чему MongoDB работает быстрее, обладает лучшей масштабируемостью, ее легче использовать.

Для хранения данных в MongoDB применяется формат, который называется BSON или сокращение от binary JSON. BSON позволяет работать с данными быстрее: быстрее выполняется поиск и обработка, хоть данные в BSON-формате и занимают больше места, чем в формате JSON.

3. Организация данных

Данные в PostgreSQL организованы в таблицы и записи. Любая таблица представляет собой именованный набор строк. Все строки таблицы имеют одинаковый набор именованных колонок, при этом каждой колонке назначается определённый тип данных.Таблицы объединяются в базы данных, а набор баз данных, управляемый одним экземпляром сервера PostgreSQL, образует кластер баз данных. Организация данных в PostgreSQL представлена на рисунке 1.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Employee Count	Employee ID	Department	Job Role	Job Level	Gender	Age	Marital Status	Education	Education Field	Business Travel	Distance From Home in kms	Job Involvement	Job Satisfaction	Monthly Income in USD	Salary Hike pc
2	1	Emp1	Sales	Sales E	2_Experienc	Female	41	Single	College	Life Scienc	2_Travel I	1	3_High	1_Very High	5,993	11
3	1	Emp2	R&D	Resear	2_Experienc	Male	49	Married	Below Co	Life Scienc	3_Travel I	8	2_Medium	2_Medium	5,130	23
4	1	Emp3	R&D	Labora	1_Entry Leve	Male	37	Single	College	Other	2_Travel I	2	2_Medium	3_High	2,090	15
5	1	Emp4	R&D	Resear	1_Entry Leve	Female	33	Married	Master	Life Scienc	3_Travel I	3	3_High	3_High	2,909	11
6	1	Emp5	R&D	Labora	1_Entry Leve	Male	27	Married	Below Co	Medical	2_Travel I	2	3_High	2_Medium	3,468	12
7	1	Emp6	R&D	Labora	1_Entry Leve	Male	32	Single	College	Life Scienc	3_Travel I	2	3_High	1_Very High	3,068	13
8	1	Emp7	R&D	Labora	1_Entry Leve	Female	59	Married	Bachelor	Medical	2_Travel I	3	4_Very High	1_Low	2,670	20
9	1	Emp8	R&D	Labora	1_Entry Leve	Male	30	Divorced	Below Co	Life Scienc	2_Travel I	24	3_High	3_High	2,693	22
10	1	Emp9	R&D	Manufa	3_Manager	Male	38	Single	Bachelor	Life Scienc	3_Travel I	23	2_Medium	3_High	9,526	21
11	1	Emp10	R&D	Healthc	2_Experienc	Male	36	Married	Bachelor	Medical	2_Travel I	27	3_High	3_High	5,237	13
12	1	Emp11	R&D	Labora	1_Entry Leve	Male	35	Married	Bachelor	Medical	2_Travel I	16	4_Very High	2_Medium	2,426	13
13	1	Emp12	R&D	Labora	2_Experienc	Female	29	Single	College	Life Scienc	2_Travel I	15	2_Medium	3_High	4,193	12
14	1	Emp13	R&D	Resear	1_Entry Leve	Male	31	Divorced	Below Co	Life Scienc	2_Travel I	26	3_High	3_High	2,911	17
15	1	Emp14	R&D	Labora	1_Entry Leve	Male	34	Divorced	College	Medical	2_Travel I	19	3_High	1_Very High	2,661	11
16	1	Emp15	R&D	Labora	1_Entry Leve	Male	28	Single	Bachelor	Life Scienc	2_Travel I	24	2_Medium	3_High	2,028	14
17	1	Emp16	R&D	Manufa	3_Manager	Female	29	Divorced	Master	Life Scienc	2_Travel I	21	4_Very High	1_Low	9,980	11
18	1	Emp17	R&D	Resear	1_Entry Leve	Male	32	Divorced	College	Life Scienc	2_Travel I	5	4_Very High	2_Medium	3,298	12
19	1	Emp18	R&D	Labora	1_Entry Leve	Male	22	Divorced	College	Medical	1_Non Tr	16	4_Very High	1_Very High	2,935	13
20	1	Emp19	Sales	Manag	4_Director	Female	53	Married	Master	Life Scienc	2_Travel I	2	2_Medium	1_Very High	15,427	16
21	1	Emp20	R&D	Resear	1_Entry Leve	Male	38	Single	Bachelor	Life Scienc	2_Travel I	2	3_High	1_Very High	3,944	11
22	1	Emp21	R&D	Manufa	2_Experienc	Female	24	Divorced	College	Other	1_Non Tr	11	4_Very High	3_High	4,011	18

Рис. 1. Организация данных в PostgreSQL

MongoDB хранит данные в документах, которые затем организованы в коллекции. Документы хранятся в коллекциях аналогично тому, как файлы хранятся в папках в традиционной аналоговой файловой системе, и так же записи хранятся в таблицах в реляционной базе данных. Нет необходимости декларировать структуру документов в системе - документы самоописаны. Если необходимо добавить новое поле в документ, это поле можно создать, не затрагивая все остальные документы в коллекции, не обновляя центральный системный каталог и не отключая систему. Организация данных в MongoDB представлена на рисунке 2.

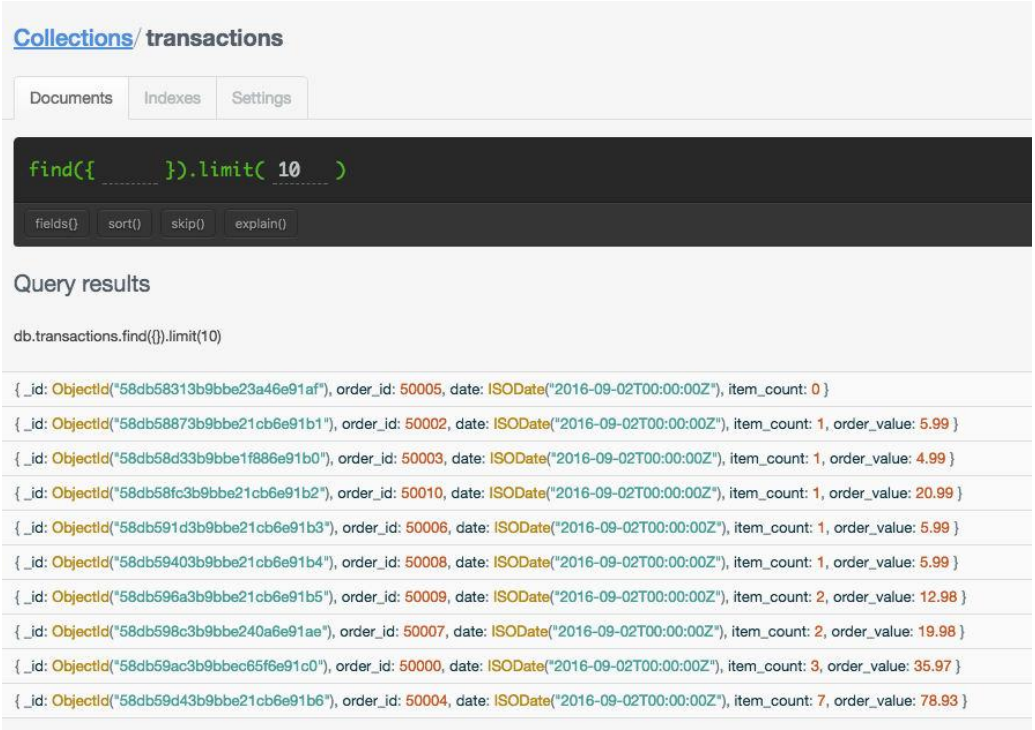

Рис. 2. Организация данных в MongoDB

4. Управление данными

PostgreSQL - это класс баз данных, неофициально называемый SQL, который означает, что он использует специфичный для базы данных вариант языка структурированных запросов (SQL) для извлечения или управления данными. SQL является основным способом взаимодействия с PostgreSQL и всеми функциями данных - от создания новых таблиц и записей до добавления ограничений и обновления индексов.

MongoDB - это класс баз данных, неофициально называемый NoSQL, который означает, что он не использует SQL для извлечения или управления данными. Вместо этого MongoDB использует вызовы функций, которые можно отправить на сервер базы данных через открытое соединение. Например, как только установлено соединение, можно создать новую базу данных, вызвав синтаксис использования: use dbname. В Mongo Shell это создает новую базу данных, если она не существовала раньше, и создает переменную db, которая ссылается на эту базу данных. С этого момента все функции выполняются по ссылке db.

5. Отношения между сущностями

PostgreSQL имеет надежный набор команд для обработки отношений между объектами, причем основной особенностью являются ограничения.

Отношения «один к одному» моделируются как поле в одной таблице, ссылающееся на другую, и наоборот. Поскольку записи в обеих таблицах имеют одно поле идентификатора, которое может ссылаться на другое, это гарантирует, что один объект связан только с этим другим объектом. Ограничение внешнего ключа может быть добавлено в поле для обеих таблиц, гарантируя, что поле внешнего ключа в одной записи относится к идентификатору другого.

Связывание «один ко многим», когда одна запись в одной таблице упоминается в нескольких записях другой, может быть смоделирована с использованием поля внешнего ключа в одной из двух связанных таблиц, причем многие стороны отношения содержат внешний ключ для одной стороны отношений.

Когда несколько записей в одной таблице должны ссылаться на несколько записей в другой таблице в отношениях «многие ко многим», таблица соединений - лучший для этого способ. Таблица соединений содержит два поля: по одной для каждой записи, к которой вы хотите присоединиться.

Документы в MongoDB не имеют таких отношений, как в реляционных базах данных. Документы, имеющие отношения «один к одному» с другими документами, обычно просто встроены в родительский документ напрямую. Например, если есть программа, содержащая объекты «ученик» и «первичное место жительства», основной объект проживания будет храниться в объекте-ученике с использованием стандартного синтаксиса встроенных объектов JSON.

Для отношений «один ко многим» можно также встраивать массивы объектов в документ.

Отношение «многие ко многим» моделируется с использованием ссылок ObjectID, но нет стандартного способа принудительного применения этих отношений, т. е. нет ограничений «внешнего ключа», которые могут автоматически применяться к данным MongoDB.

6. Когда подходят PostgreSQL и MongoDB

PostgreSQL отлично подходит для данных с высокой степенью структурированности и отношений, которые могут меняться со временем. В ней легко создавать и управлять новыми отношениями. PostgreSQL предоставляет гарантии того, как данные вставляются в базу данных. Она также имеет две функции программирования, которые называются триггерами и хранимыми процедурами, которые допускают некоторую степень программирования в самой базе данных. Триггеры - это события, которые могут быть сгенерированы при возникновении различных действий с базой данных, например, когда документ вставлен или проверен. Хранимые процедуры - это функции, написанные на процедурном языке, которые могут быть запущены в базе данных при возникновении различных триггеров.

MongoDB хорошо подходит, когда есть данные, которые часто меняются или данные, где схема неизвестна. Для документов в коллекции миграция данных не требуется, и коллекции могут содержать документы с различными полями. Это делает MongoDB идеальным для сред, где имеется большое количество разнородных данных с простыми и заранее определенными отношениями между объектами. Поскольку документы хранятся в формате JSON, MongoDB естественным образом подходит для команд, использующих Javascript, поскольку он по существу служит в качестве базы данных объектов для разработчиков Javascript

7. Когда не подходят PostgreSQL и MongoDB

Если структуры данных постоянно меняются или если происходят попытки найти дизайн данных, который хорошо работает в нескольких ситуациях, жесткость и структура PostgreSQL могут

стать разочаровывающими. Каждое изменение структуры данных требует, чтобы разработчик выполнил серию вызовов SQL, которые переносят старые данные, чтобы они соответствовали новой структуре. Это может быть очень утомительным, когда структуры данных часто меняются.

В средах, где данные сильно структурированы и имеют высокую реляционную структуру, MongoDB не может быть идеальным выбором. Поскольку данные в коллекции неоднородны, запросы по нескольким полям, которые имеют очень сложные отношения, могут занимать гораздо больше времени и требуют гораздо большего объема оперативной памяти, чем аналогичные запросы в реляционной базе данных. Как правило, если данные сильно структурированы и вряд ли будут изменены, или если есть планы запрашивать множество разных объектов со сложными отношениями, тогда лучше выбрать реляционную базу данных, такую как PostgreSQL.

8. Заключение

PostgreSQL является полнофункциональной объектно-реляционной СУБД, готовой для практического использования. Ее функциональность и надежность обусловлены богатой историей развития, профессионализмом разработчиков и технологией тестирования, а ее перспективы заложены в ее расширяемости и свободной лицензии.

MongoDB в большинстве случаев способна стать заменой реляционной базе данных. Она намного проще и понятнее; быстрее работает и имеет меньше ограничений для разработчиков приложений.

Когда начинается проект, выбор правильного варианта работы с данными имеет огромное значение. Если есть необходимсоть в высокой степени гибкости, MongoDB - отличный выбор. Если есть реляционные данные, которые предоставляют гарантии относительно структуры и последовательности, PostgreSQL - отличный вариант.

Литература

1. Comparing PostgreSQL and MongoDB. [Электронный ресурс]. -Режим доступа: https://www.mongodb.com/compare/mongodb-postgresql
2. Введение в MongoDB. [Электронный ресурс]. - Режим доступа: https://metanit.com/nosql/mongodb/1.1.php
3. Что такое PostgreSQL? [Электронный ресурс].-Режим доступа: http://www.sai.msu.su/~megera/postgres/talks/what_is_postgresql.html
4. Compose Compare: MongoDB vs. PostgreSQL. [Электронный ресурс].-Режим доступа: https://www.compose.com/articles/compose-compare-mongodb-vs-postgresql/