

История Postegres

Введение

PostgreSQL – это свободно распространяемая объектно-реляционная система управления базами данных (ORDBMS), наиболее развитая из открытых СУБД в мире и являющаяся реальной альтернативой коммерческим базам данных.

Краткая история PostgreSQL

Объектно-реляционная СУБД, в настоящее время известная как PostgreSQL ведёт свое происхождение от пакета POSTGRES, который был написан в департаменте Беркли, Калифорнийского Университета. Более чем двадцатилетняя разработка PostgreSQL сделала этот продукт одной из наиболее продвинутых СУБД с открытым исходным кодом.

Проект POSTGRES

Проект POSTGRES, под руководством профессора Майкла Стоунбрейкера (Michael Stonebraker), был спонсирован Агентством Расширенных Оборонных Исследовательских Проектов (Defense Advanced Research Projects Agency (DARPA)), Офисом Армейских Исследований (Army Research Office (ARO)), Национальным Научным Фондом (National Science Foundation (NSF)), а также ESL, Inc. Реализация реляционной СУБД POSTGRES началась в 1986. Затем вышло несколько версий Postgres. Версия 1 была выпущена в июне 1989 года и могла работать с несколькими внешними пользователями.

Модель данных

Фундаментальная характеристика объектно-реляционной базы данных — это поддержка пользовательских объектов и их поведения, включая типы данных, функции, операции, домены и индексы. Это делает Постгрес невероятно гибким и надежным. Среди прочего, он умеет создавать, хранить и извлекать сложные структуры данных.

Функции

Функции являются блоками кода, исполняемыми на сервере, а не на клиенте БД. Хотя они могут писаться на чистом SQL, реализация дополнительной логики, например, условных переходов и циклов, выходит за рамки собственно SQL и требует использования некоторых языковых расширений. Функции могут писаться с использованием одного из следующих языков:

Функции могут выполняться как с правами их создателя, так и с правами текущего пользователя.

Иногда функции отождествляются с хранимыми процедурами, однако между этими понятиями есть различие. С девятой версии возможно написание автономных блоков, которые позволяют выполнять код на процедурных языках без написания функций, непосредственно в клиенте.

Триггеры

Триггеры определяются как функции, инициируемые DML-операциями (язык управления (манипулирования) данными). Например, операция INSERT может запускать триггер, проверяющий добавленную запись на

соответствия определённым условиям. При написании функций для триггеров могут использоваться различные языки программирования (см. выше).

Правила и представления

Механизм правил (англ. *rules*) представляет собой механизм создания пользовательских обработчиков не только DML-операций, но и операции выборки. Правила позволяют переопределять поведение системы при выполнении SQL-операции к таблице. Хорошим примером является реализация механизма представлений (англ. *views*): при создании представления создается правило, которое определяет, что вместо выполнения операции выборки к представлению система должна выполнять операцию выборки к базовой таблице / таблицам с учетом условий выборки, лежащих в основе определения представления. Для создания представлений, поддерживающих операции обновления, правила для операций вставки, изменения и удаления строк должны быть определены пользователем.

Индексы

В PostgreSQL имеется поддержка индексов следующих типов: B-дерево, хэш, R-дерево, GiST, GIN. При необходимости можно создавать новые типы индексов, хотя это далеко не тривиальный процесс. Индексы в PostgreSQL обладают следующими свойствами:

- создание отдельного индекса для работы конструкции ORDER BY... DESC не нужно;
- возможно создание индекса над несколькими столбцами таблицы, в том числе над столбцами различных типов данных;

- то есть строиться не на базе набора значений некоего столбца / столбцов, а на базе набора значений функции от набора значений;
- то есть строиться только по части таблицы (по некоторой её проекции); в некоторых случаях это помогает создавать намного более компактные индексы или достигать улучшения производительности за счёт использования разных типов индексов для разных (например, с точки зрения частоты обновления) частей таблицы;
- планировщик запросов может использовать несколько индексов одновременно для выполнения сложных запросов.

Типы данных

PostgreSQL предоставляет пользователям богатый ассортимент встроенных типов данных. Кроме числовых, с плавающей точкой, текстовых, булевых и других ожидаемых типов данных (а также множества их вариаций), PostgreSQL может похвастаться поддержкой

Некоторые типы данных:

Денежный,

Перечисления,

Геометрические,

Бинарные,

Сетевых адресов,

Битовых строк,

Текстового поиска,

Массивы

Композитных типов и диапазонов,
и многие другие.

Справедливости ради стоит сказать, что MySQL, MariaDB и Firebird тоже имеют некоторые из этих типов данных, но только Постгрес поддерживает их все. Кроме того, пользователи могут создавать свои типы в PostgreSQL.

INSERT

Здесь *число* представляет количество добавленных или изменённых строк. Если *число* равняется одному, а целевая таблица содержит *oid*, то в качестве *oid* (Первичный ключ) выводится OID, назначенный добавленной строке. Эта одна строка должна быть добавлена, но не изменена. В противном случае в качестве *oid* выводится ноль.

Многомерные массивы

Поскольку Постгрес — это объектно-реляционная база данных, массивы значений могут храниться для большинства существующих типов данных. Сделать это можно путём добавления квадратных скобок к спецификации типа данных для столбца или с помощью выражения ARRAY. Размер массива может быть задан, но это необязательно. Давайте рассмотрим меню праздничного пикника для демонстрации использования массивов.

Чтобы хранить такие массивы значений в традиционных реляционных базах данных, придется использовать обходной путь и создавать отдельную таблицу со строками для каждого из значений массива.

Создание нового типа

Если вдруг так случится, что обширного списка типов данных Постгреса вам окажется недостаточно, вы можете использовать команду `CREATE TYPE`, чтобы создать новые типы данных, такие как составной, перечисляемый, диапазон и базовый. Рассмотрим пример создания и отправки запросов нового составного типа.

Где используется?

Если изначально `POSTGRES` использовался в основном в академических проектах для исследования алгоритмов баз данных, в университетах как отличная база для обучения, то сейчас PostgreSQL применяется практически повсеместно. Например, зоны `.org`, `.info` полностью обслуживаются PostgreSQL, известны многотерабайтные хранилища астрономических данных, Lycos, BASF. Из российских проектов, использующих PostgreSQL, наиболее известными является портал Рамблер.