KARPOV.COURSES >>>
КОНСПЕКТ

> Конспект > 2 урок > Объекты баз данных. Зачем и что используется

> Оглавление

> Таблица

Таблица - основной объект хранения данных в базе данных. Она состоит из:

- столбцов определенных типов;
- строк;
- данные в ячейках.

Столбцы в БД должны иметь уникальные имена.

Виды таблиц.

Постоянная таблица.

Хранится на диске, журналиуется, её свойства - атомарность, конститетность, изолированность и устойчивость. Особенности - в PostgreSQL это одна таблица, GreenPlum представляет собой набор небольших таблиц, и умеет хранить данные построчно и поколонночно. Каждый столбец и каждая колонка будет сохранена в свой отдельный файл, это позволяет получить выигрыш при чтении данных с диска.

Временная таблица - применяется в процессе вычислений, создается “на лету” и после окончания вычислений может быть уничтожена. Временная таблица может существовать во время одной транзакции. Особенность - операции по сбору статистики или сжатия должны выполняться разработчиками самостоятельно. В GreenPlum для удаления временных таблиц нужно выполнять отдельный процесс.

Нежурналируемая таблица - это таблица, изменения в которой не записываются в лог базы данных. Это ускоряет взаимодействие пользователя с данными, но при перезагрузке или сбое данные в этой таблице будут потеряны, останутся только атрибуты. Таблица существует в СУБД PostgreSQL.

Внешняя таблица - это специальный объект, для которого в базе данных хранится только описание или метаданные (метаинформация). Доступ к таким данным осуществляется с помощью специальных драйверов. Для пользователей внешняя таблица выглядит как обычная таблица, но для СУБД это отдельный объект. Поэтому доступ к такой таблице может занимать гораздо больше времени.

Представления - это виртуальная (логическая) таблица, представляющая собой поименованный запрос (синоним к запросу), который будет подставлен как подзапрос при использовании представления. Таблица позволяет получать результаты действий над другими таблицами и пользователям результат представляется “на лету”. Они (представления) обычно хранятся в системе управления базами данных в виде SQL-запроса.

В отличие от обычных таблиц реляционных баз данных, представление не является самостоятельной частью и хранятся в виде своего определения. Содержимое представления динамически вычисляется на основании данных, находящихся в реальных таблицах. Изменение данных в реальной таблице базы данных немедленно отражается в содержимом всех представлений, построенных на основании этой таблицы.

Отличие **GreenPlum** - это ключ дистрибуции. При создании таблиц или представлений необходимо его указывать, чтобы данные были разделены между сегментами.

Справка: ключ дистрибуции

Distribution Key, он же ключ распределения - важный элемент в обеспечении равномерности распределения данных по сегментам MPP СУБД и как следствие обеспечения эффективности ее работы.

Ключ дистрибуции указывается при создании таблицы и может быть изменен при необходимости.

GreenPlum (вы еще можете встретить сокращение GP) может использовать Hash distribution алгоритм при указании ключа (1 или несколько полей) или при автоматическом выборе ключа (первая колонка таблицы), если это явно не указано в DDL. Также может быть указан вариант распределения Randomly, когда

строки распрееляются по сегментам по кругу по порядку - обычно самое равномерное распределение.

В качестве ключа дистрибуции лучше выбирать поле, по которому по большим объектам будет происходить JOIN, желательно целочисленного типа, желательно без NULLS. Если у таблицы есть Primary Key - это лучший кандидат.

В случае возникновения сильного перекоса лучше сделать Distributed Randomly.

> Типы данных

Типы данных - доступные способы представления данных в БД. Каждый тип данных - это и набор операторов над этим типом данных, позволяющие осуществлять разные действия с данными.

Какие существуют типы данных:

- числовые
- символьные
- двоичные
- логические (истина или ложь)
- дата и время
- перечисляемый
- массивы
- составные
- диапазоны
- геометрические
- сложносоставные
- пользовательские

Массив - тип данных , который сохраняет несколько значений одного и того же подтипа.

Составной тип данных - в одном элементе сочетаются разнотипные данные.

Диапазоны - тип данных, при помощи которого можно хранить и обрабатывать диапазон дат, значений.

Сложносоставные типы данных, например xml, json - при помощи данных типов PostgreSQL и GreenPlum реализуют возможности noSQL БД, расширяя зону своего присутствия.

> Ограничения целостности

Ограничения целостности - это набор инструментов, позволяющих БД выполнять требования консистентности.

Какие бывают ограничения целостности:

Первичный ключ (Primary Key) - позволяет отслеживать отсутствие одинаковых значений в таблице по первичному ключу. Он может состоять из нескольких колонок таблицы, которые не должны повторяться.

Внешний ключ (Foreign Key) - он позволяет связывать таблицы между собой. Он не позволяет удалить запись из родительской таблица, если хоть одна запись из дочерней таблица ссылается на эту запись.

Уникальность (Unique) - это свойство требует уникальные значения в каком-либо столбце таблицы.

Обязательность (Not NULL) - в таблицу не может быть записана строка, в которой ограничение целостности Not NULL пустое.

Проверка (check) - в проверку могут быть заложены любые пользовательские условия, например ограничение по размеру информации или по типу данных.

Исключение (exclude) - позволяет проверять, что вставляемые в таблицу значения, уже не попадают в какой-либо диапазон. Это ограничения удобно применять со специфическими индексами GiST и SP-GiST

Справка.
Консистентность - это согласованность данных друг с другом, целостность данных, а также внутренняя непротиворечивость.

> Индексы

Индекс - это объект базы данных, создаваемые с целью повышения производительности поиска данных.

Они требуют: - дополнительного места- дополнительных ресурсов в словаре данных

- больше ресурсов на Insert/Update
- отдельное обслуживание

Какие бывают индексы.

BTree

BTree - это сбалансированное дерево поиска с несколькими ветвями, разработанное для дисков или других устройств хранения. (По сравнению с двумя вилками, каждый внутренний узел B-дерева имеет несколько ветвей, то есть несколько вилок). Подробнее.

Hash

Hash (PostgreSQL only) - позволяет построить Hash-таблицу по заданному полю и производить операции над таблицами, основываясь на этом Hash-значении. Недостатки - они не передаются в логи и не попадают в реплики базы данных

GiST

GiST (Generalized Search Tree) - позволяет осуществлять поиск по любым значениям.

SP-GiST

SP-GiST (Space-partitioned GiST) - SP-GiST представляет собой дерево поиска, ветви которого не пересекаются (в отличие от GiST).

GIN

GIN - индекс, который позволяет производить поиск по технологии “ключ-значение”, которые создаются пользователями.

BRIN

BRIN (Bitmap Range Index, работает только в PostgreSQL) - применим к полям с повторяющимся значениями. Занимает мало места, но поиск по нему занимает большое время, по сравнению с поиском бинарного дерева.

Bitmap

Bitmap индекс (GreenPlum only) - позволяет индексировать поля с повторяющимися значениями. Эти значения хорошо укладываются в битовую карту. Она позволяет получать строки или диапазон строк в поиске очень быстро.

> Последовательность и триггеры

Последовательности (Sequences)

Это специальные объекты, позволяющие получать уникальные и генерировать последовательность уникальных номеров в условиях многопользовательского асинхронного доступа. Обычно элементы последовательности используются для уникальной нумерации элементов таблиц (строк) в операциях модификации данных.

Это может быть отдельный объект или поле типа SERIAL.

Для последовательности можно задать начальное значение. Последовательности не гарантируют получение строго идущих друг за другом идентификаторов, они гарантируют уникальность полученных идентификаторов.

Триггеры

Это процедуры вызываемые при наступлении определенного события внутри базы данных (вставки, создания, удаления, обновления записей).

Триггер позволяет выполнить пользовательскую процедуру и сохранить данные в любую другую таблицу.

> Пользовательские функции

Пользовательские функции – программируемый объект базы данных, возвращающий значение предопределенного типа.Это могут быть обработчики для триггеров или программировать операторы базы данных. Пользователь может писать функции на:- PL\PGSQL

- PL\Python
- PL\Perl
- PL\TCL
- C

> Секции и правила

Секции

Секции - это разделение хранимых объектов баз данных (таких как таблиц, индексов, материализованных представлений) на отдельные части с раздельными параметрами физического хранения. Используется в целях повышения управляемости, производительности и доступности для больших баз данных.

Секции позволяют разбивать секции по:- датам- диапазонам дат/времени- список значений- диапазон значений

Правила

Правила - объект, который позволяет переписывать SQL-запросы “на лету”.

Набор правил, при получении запроса от парсера преобразовать его нужным образом и отправить планировщику переписанный запрос. Правила дают возможность гибко настраивать реакцию базы данных на SQL-запросы.