Интеграция с внешними системами

Раздаточный материал к 4-му уроку, посвященному интеграции с внешними системами.

|  |  |
| --- | --- |
| СОДЕРЖАНИЕ | |
| 1 | Концепция слоев в хранилище |
| 2 | Загрузка с помощью insert, copy |
| 3 | Внешние таблицы |
| 4 | Загрузка с помощью pxf |
| 5 | Загрузка с помощью gpfdist |
| 6 | Источники информации (ссылки) |

Концепция слоев в хранилище

**Корпоративное хранилище данных (EDW)** - представляет собой большой набор хранилищ данных, оно создано для целей аналитических отчетов предприятия и поддержки принятия решений. Оно фильтрует и интегрирует различные бизнес-данные и предоставляет предприятиям определенные возможности BI (бизнес-аналитики) для управления улучшением бизнес-процессов, контроля времени, затрат, качества и контроля.

Информация обычно поступает от различных систем, например, из ERP, CRM, физических записей и других неструктурированных файлов. Для подготовки данных к дальнейшему анализу их нужно поместить в единое хранилище. Благодаря этому разные подразделения бизнеса могут запрашивать их и анализировать информацию под различными углами. Однако, чтобы данные превратились в полезную информацию, они должны пройти долгий путь.

Существует определение основных функций EDW, которые составляют в целом концепцию хранения данных в нем:

**Служит в качестве всеобъемлющего накопителя данных.** Корпоративное хранилище данных — это единый репозиторий для всех корпоративных бизнес-данных, появляющихся в организации.

**Отражает исходные данные.** EDW предназначено для того, чтобы обеспечить схожесть изначальных исходных данных в одном репозитории. Так как и внутри, и снаружи компании всегда генерируются новые релевантные данные, прежде чем поток данных попадёт в хранилище, ему требуется специализированная инфраструктура для управления им.

**Хранение структурированных данных.** Хранящиеся в EDW данные всегда стандартизованы и структурированы. Благодаря этому конечные пользователи могут запрашивать их через интерфейсы бизнес-аналитики и формы отчётов.

**Данные ориентированы на предметную область.** Основная цель хранилища — это бизнес-данные, которые могут относиться к различным предметным областям. Чтобы понять, к чему относятся данные, они структурируются на основе конкретной предметной области, называемой моделью данных.

**Зависимость от времени.** Собираемые данные обычно являются историческими, потому что объясняют события прошлого. Чтобы понять, когда и в течение какого времени имела место определённая тенденция, подавляющая часть хранимой информации обычно разделена по временным периодам.

**Отсутствие изменений.** После помещения в хранилище данные больше никогда из него не удаляются. С данными могут выполняться манипуляции, их могут изменять или обновлять из-за изменений в источнике, однако они никогда не должны удаляться, по крайней мере, конечными пользователями.

**Слои хранилища данных**

**Источники данных.** Это все источники данных, откуда берутся «сырые» данные и/или где они хранятся. Они могут быть простыми электронными таблицами, неструктурированными файлами, реляционными базами данных SQL и так далее.

**Промежуточный этап (дополнительный).** В случае ETL, промежуточный этап — это место, где данные преобразуются перед загрузкой в EDW. Здесь они очищаются, избавляются от дубликатов, разделяются, объединяются и преобразуются в единый формат, соответствующий модели данных хранилища. Промежуточный этап может также включать инструментарий для управления качеством данных.

**Слой потребления.** Существует два основных подхода к получению данных из источников и передаче их в хранилище. Инструменты ETL (extract, transform, load) и ELT (extract, load, transform) подключаются ко всем источникам данных и выполняют их извлечение, преобразование и загрузку в централизованную систему хранения для удобного доступа и анализа. Различие между методиками ETL и ELT заключается в порядке событий.

В ETL преобразование происходит на промежуточном этапе, прежде чем данные попадут в EDW. ELT — это более современный подход, в нём все задачи по преобразованию выполняются внутри хранилища, а промежуточный этап отсутствует.

**Слой хранения** - преобразованные данные загружаются в пространство хранения. В методике ELT здесь они могут подвергаться преобразованиям. Но на этом этапе все основные изменения уже внесены, поэтому данные будут загружаться в свои окончательные модели.

**Модуль метаданных** — это данные о данных. Это объяснения, сообщающие пользователям или администраторам, с какой темой или предметной областью связана эта информация. Эти данные могут быть технической метой (например, указанием исходного источника) или бизнес-метой (например, регионом продаж). Все метаданные хранятся в отдельном модуле EDW и управляются менеджером метаданных.

**Витрины данных (дополнительные).** В некоторых случаях в EDW может существовать множество дополнительных подразделов, называемых витринами данных (data marts); они создаются специально под конкретную предметную область, бизнес-функцию или группу пользователей. Например, может существовать отдельная витрина данных для маркетинга и витрина данных для финансового отдела.

**Слой представления.** Последний блок EDW состоит из инструментов, дающих конечному пользователю доступ к данным. Этот слой, также называемый интерфейсом бизнес-аналитики (BI interface), служит в качестве дэшборда для визуализации данных, бизнес-отчётности и вывода отдельных элементов информации для задач наподобие машинного **обучения**.

Загрузка с помощью insert, copy

Самыми простыми способами загрузки данных в Greenplum являются следующие:

**- вставка (INSERT) со значениями столбцов** — оператор INSERT вставляет новые строки в таблицу. Можно вставить одну или несколько строк, через указанное пользователем выражение или ноль или более строк, полученных в результате запроса SELECT из таблицы.

Данные проходят через мастер-узел и распределяются по сегменту. Это самый медленный способ загрузки, который **не подходит для больших объемов данных**.

Имена целевых столбцов могут быть перечислены в любом порядке. Если список имен столбцов вообще не задан, по умолчанию используются столбцы таблицы в их объявленном порядке. Значения, предоставляемые предложением или запросом VALUES, связаны с явным или неявным списком столбцов слева направо.

Каждый столбец, отсутствующий в явном или неявном списке столбцов, будет заполнен значением по умолчанию, либо его объявленным значением по умолчанию, либо нулевым значением, если значение по умолчанию отсутствует.

Если выражение для любого столбца имеет неправильный тип данных, будет предпринята попытка автоматического преобразования типа.

**- копирование (COPY)** — этот оператор PostgreSQL копирует данные из внешнего файла в таблицу базы данных, делая это более эффективно, чем вышерассмотренный оператор INSERT. Но строки по-прежнему проходят через мастер-узел, а все данные копируются одной командой, т.е. это не параллельный процесс.

Поэтому этот способ загрузки из файла или стандартного ввода **подходит для добавления относительно небольших наборов данных**, например таблиц измерений до 10 000 строк или однократной загрузки данных.

Для повышения производительности можно запустить несколько команд COPY одновременно.

COPY TO копирует содержимое таблицы в файл (или несколько файлов на основе идентификатора сегмента при копировании ON SEGMENT), а COPY FROM копирует данные из файла в таблицу (добавляя данные к тому, что уже находится в таблице). COPY TO также может копировать результаты запроса SELECT.

Если указан список столбцов, COPY скопирует только данные из указанных столбцов в файл или из файла. Если в таблице есть какие-либо столбцы, которых нет в списке столбцов, COPY FROM вставит значения по умолчанию для этих столбцов.

COPY с именем файла указывает главному хосту базы данных Greenplum непосредственно читать или записывать в файл. Файл должен быть доступен для главного хоста, и имя должно быть указано с точки зрения главного хоста.

Внешние таблицы

Внешние таблицы обеспечивают доступ к данным в источниках за пределами Greenplum.

Доступ к ним можно получить с помощью SQL-операторов SELECT, и они обычно используются с ELT/ETL-шаблонами с быстрой параллельной загрузкой данных.

Внешние таблицы поддерживают различные SQL-операции: выбор, сортировка и объединение, поэтому данные можно загружать и преобразовывать одновременно или загружать в стейдж таблицу и преобразовывать в целевые таблицы СУБД.

В Greenplum внешняя таблица определяется с помощью оператора CREATE EXTERNAL TABLE с условиями LOCATION для задания местоположения данных и FORMAT для форматирования исходных данных, чтобы система могла анализировать входные данные.

**CREATE (READABLE) EXTERNAL TABLE** создает новое определение внешней таблицы с возможностью чтения из таблицы. Внешние таблицы с параметром readable обычно используются для быстрой параллельной загрузки данных. Как только внешняя таблица определена, вы можете запрашивать ее данные напрямую (и параллельно) с помощью команд SQL.

Например, вы можете выбирать, объединять или сортировать данные внешней таблицы. Вы также можете создавать вью для внешних таблиц. Операции UPDATE, INSERT, DELETE или TRUNCATE не разрешены для читаемых внешних таблиц, и вы не можете создавать индексы для читаемых внешних таблиц.

Основные ключевые фразы для определения ВТ:

**• Параметр WEB**

Создает доступное для чтения или записи определение внешней веб-таблицы в базе данных Greenplum. Существует две формы читаемых внешних веб-таблиц: те, которые обращаются к файлам по протоколу http://, и те, которые обращаются к данным, выполняя команды ОС. Доступные для записи внешние веб-таблицы выводят данные в исполняемую программу, которая может принимать входной поток данных. Внешние веб-таблицы нельзя повторно сканировать во время выполнения запроса.

Протокол s3 не поддерживает внешние веб-таблицы. Однако вы можете создать внешнюю веб-таблицу, которая запускает сторонний инструмент для чтения данных или записи данных в S3 напрямую.

**• TEMPORARY|TEMP**

Если указано, создает временное определение внешней таблицы с возможностью чтения или записи в базе данных Greenplum. Временные внешние таблицы существуют в специальной схеме; вы не можете указать имя схемы при создании таблицы. Временные внешние таблицы автоматически удаляются в конце сеанса.

Существующая постоянная таблица с тем же именем не видна для текущего сеанса, пока существует временная таблица, если только вы не сошлетесь на постоянную таблицу с ее именем, определенным схемой.

При определении внешней таблицы обязательно указывается имя внешней таблицы (с указанием) схемы.

Список столбцов для внешней таблицы определяется через простое перечисление (при этом нельзя задать ограничения столбцов или знаение по умолчанию) или через выражение LIKE

**• LIKE other\_table**

Предложение LIKE указывает таблицу, из которой новая внешняя таблица автоматически копирует все имена столбцов, типы данных и дистрибуцию. Если в исходной таблице указаны какие-либо ограничения столбцов или значения столбцов по умолчанию, они не будут скопированы в новое определение внешней таблицы.

**• LOCATION ('протокол://[хост[:порт]]/путь/файл' [, ...])**

Для читаемых внешних таблиц указывает URI внешних источников данных, которые будут использоваться для заполнения внешней таблицы или веб-таблицы.

Обычные внешние таблицы допускают использование gpfdist или файловых протоколов. Внешние таблицы c параметром WEB разрешают протокол http.

Если порт не указан, для протоколов http и gpfdist предполагается порт 8080.

При использовании протокола gpfdist путь указывается относительно каталога, указанного при запуске программы gpfdist).

Примеры:

'gpfdist://filehost:8081/\*'

'gpfdist://masterhost/my\_load\_file'

'файл://seghost1/dbfast1/external/myfile.txt'

'http://intranet.example.com/finance/expenses.csv'

**• ON MASTER**

Ограничивает все операции, связанные с таблицами, главным сегментом Greenplum. Разрешено только для внешних таблиц с возможностью чтения и записи, созданных с помощью s3 или пользовательских протоколов. Протоколы gpfdist, gpfdists, pxf и file не поддерживают ON MASTER.

Помните о потенциальном влиянии на ресурсы при чтении или записи во внешние таблицы, которые вы создаете с помощью команды ON MASTER. Вы можете столкнуться с проблемами производительности, если вы ограничиваете операции с таблицами исключительно главным сегментом Greenplum.

**• EXECUTE 'команду' [ON ...] (только для ВТ с параметром WEB)**

Разрешено только для внешних веб-таблиц, доступных для чтения, или внешних таблиц, доступных для записи. Для доступных для чтения внешних веб-таблиц указывает команду ОС, которая должна выполняться экземплярами сегмента. Команда может быть отдельной командой ОС или сценарием. Предложение ON используется для указания того, какие экземпляры сегмента будут выполнять данную команду.

ON ALL — значение по умолчанию. Команда будет выполняться каждым активным (первичным) экземпляром сегмента на всех хостах сегмента в системе базы данных Greenplum. Если команда выполняет сценарий, этот сценарий должен находиться в одном и том же месте на всех узлах сегмента и быть исполняемым суперпользователем Greenplum (gpadmin).

ON MASTER запускает команду только на главном хосте.

Число в команде ON означает, что команда будет выполняться указанным количеством сегментов. Конкретные сегменты выбираются случайным образом во время выполнения команды. Если команда выполняет сценарий, этот сценарий должен находиться в одном и том же месте на всех узлах сегмента и быть исполняемым суперпользователем Greenplum (gpadmin).

HOST означает, что команда будет выполняться одним сегментом на каждом узле сегмента (один раз на узле сегмента), независимо от количества активных экземпляров сегмента на узле.

HOST имя\_узла\_сегмента означает, что команда будет выполняться всеми активными (первичными) экземплярами сегмента на указанном узле сегмента.

SEGMENT segment\_id означает, что команда будет выполнена только один раз указанным сегментом. Вы можете определить идентификатор экземпляра сегмента, просмотрев номер содержимого в таблице системного каталога gp\_segment\_configuration. Идентификатор содержимого мастера базы данных Greenplum всегда равен -1.

Для внешних таблиц, доступных для записи, команда, указанная в предложении EXECUTE, должна быть подготовлена для передачи в нее данных. Поскольку все сегменты, имеющие данные для отправки, будут записывать свои выходные данные в указанную команду или программу, единственная доступная опция для предложения ON — ON ALL.

**• FORMAT 'TEXT | CSV (options)**

Когда предложение FORMAT определяет формат текста с разделителями (TEXT) или формат файла со значениями, разделенных запятыми (CSV), параметры форматирования аналогичны параметрам, доступным для команды PostgreSQL COPY. Их можно увидеть на экране

Если данные в файле не используют разделитель столбцов по умолчанию, escape-символ, нулевую строку и т. д., необходимо указать дополнительные параметры форматирования, чтобы данные во внешнем файле корректно считывались базой данных Greenplum.

**• FORMAT 'CUSTOM' (formatter=formatter\_specification)**

Задает пользовательский формат данных. Formatter\_specification указывает функцию, используемую для форматирования данных, за которой следуют параметры, разделенные запятыми, для функции форматирования. Длина спецификации средства форматирования, включая строку Formatter=, может составлять приблизительно до 50 КБ.

**• DELIMITER**

Задает один символ ASCII, разделяющий столбцы в каждой строке данных. По умолчанию используется символ табуляции для формата TEXT и запятая для CSV. В режиме TEXT для внешних таблиц с параметром readable разделитель может быть отключен для особых случаев использования, когда неструктурированные данные загружаются в таблицу с одним столбцом.

Для протокола s3 разделителем не может быть символ новой строки (\n) или символ переноса курсора (\r).

**• NULL**

Указывает набор символов, представляющих значение NULL. По умолчанию используется \N (обратная косая черта-N) в режиме TEXT и пустое значение без кавычек в режиме CSV. Вы можете предпочесть пустую строку даже в режиме TEXT для случаев, когда вы не хотите отличать значения NULL от пустых строк. При использовании внешних и веб-таблиц любой элемент данных, соответствующий этой строке, будет считаться значением NULL.

В качестве примера для текстового формата это предложение FORMAT можно использовать, чтобы указать, что строка из двух одинарных кавычек ('') является значением NULL.

FORMAT 'TEXT' (DELIMITER ',' null '\'\'\'\'' )

**• ESCAPE**

Указывает одиночный символ, который используется для escape-последовательностей (таких как \n,\t,\100 и т. д.) и для экранирующих символов, которые в противном случае могли бы восприниматься как разделители строк или столбцов. Обязательно выберите escape-символ, который нигде не используется в фактических данных столбца. Экранирующий символ по умолчанию — \ (обратная косая черта) для файлов в текстовом формате и " (двойная кавычка) для файлов в формате csv, однако можно указать другой символ для escape-символа.

Также можно отключить экранирование в тексте. форматированные файлы, указав значение 'OFF' в качестве escape-значения. Это очень полезно для таких данных, как данные веб-журнала в текстовом формате, которые содержат множество встроенных обратных косых черт, которые не предназначены для escape-последовательности.

**• NEWLINE**

Определяет новую строку, используемую в ваших файлах данных — LF (перевод строки, 0x0A), CR (возврат каретки, 0x0D) или CRLF (возврат каретки плюс перевод строки, 0x0D 0x0A). Если не указано иное, сегмент базы данных Greenplum будет определять тип новой строки, просматривая первую строку полученных данных и используя первый встреченный тип новой строки.

**• HEADER**

Для внешних таблиц типа readable указывает, что первая строка в файле(ах) данных является строкой заголовка (содержит имена столбцов таблицы) и не должна включаться в качестве данных для таблицы. При использовании нескольких файлов источников данных все файлы должны иметь строку заголовка.

Для протокола s3 имена столбцов в строке заголовка не могут содержать символ новой строки (\n) или возврат каретки (\r).

Протокол pxf не поддерживает параметр форматирования HEADER.

**• FORSE NOT NULL**

В режиме CSV каждый указанный столбец обрабатывается так, как если бы он был заключен в кавычки и, следовательно, не имеет значения NULL. Для null строки по умолчанию в режиме CSV (ничего между двумя разделителями) это приводит к тому, что отсутствующие значения оцениваются как строки нулевой длины.

**• FILL MISSING FIELDS**

Как в режиме TEXT, так и в режиме CSV для внешних таблиц c параметром readable указание FILL MISSING FIELDS установит значения отсутствующих полей в NULL (вместо сообщения об ошибке), когда в строке данных есть отсутствующие поля данных в конце строки или строки. Пустые строки, поля с ограничением NOT NULL и конечные разделители в строке по-прежнему будут сообщать об ошибке.

**• LOG ERRORS**

Это необязательное предложение, которое может предшествовать предложению SEGMENT REJECT LIMIT для регистрации информации о строках с ошибками форматирования. Информация журнала ошибок хранится внутри, и доступ к ней осуществляется с помощью, встроенной в базу данных Greenplum функции SQL gp\_read\_error\_log().

**• SEGMENT REJECT LIMIT count [ROWS | PERCENT]**

Запускает операцию COPY FROM в режиме изоляции ошибок одной строки. Если во входных строках есть ошибки формата, они будут отброшены при условии, что во время операции загрузки ни в одном из экземпляров сегмента Greenplum не будет достигнут лимит отклонения. Счетчик предела отклонения может быть указан как количество строк (по умолчанию) или процент от общего числа строк (1-100). Если используется PERCENT, каждый сегмент начинает вычислять процент плохих строк только после обработки количества строк, заданного параметром gp\_reject\_percent\_threshold. Значение по умолчанию для gp\_reject\_percent\_threshold — 300 строк. Ошибки ограничений, такие как нарушение ограничений NOT NULL, CHECK или UNIQUE, будут по-прежнему обрабатываться в режиме ввода «все или ничего». Если предел не достигнут, все хорошие строки будут загружены, а все строки с ошибками отброшены.

Примечание. При чтении внешней таблицы база данных Greenplum ограничивает начальное количество строк, которые могут содержать ошибки форматирования, если SEGMENT REJECT LIMIT не срабатывает первым или не указано. Если первые 1000 строк отклоняются, операция COPY останавливается и выполняется откат.

Ограничение на количество первоначальных отклоненных строк можно изменить с помощью параметра конфигурации сервера базы данных Greenplum gp\_initial\_bad\_row\_limit. Информацию об этом параметре см. в разделе Параметры конфигурации сервера.

**CREATE WRITABLE EXTERNAL TABLE** создает новое определение внешней таблицы с возможностью записи в базе данных Greenplum. Внешние таблицы с возможностью записи обычно используются для выгрузки данных из базы данных в файлы. После определения доступной для записи внешней таблицы данные можно выбирать из таблиц базы данных и вставлять в данную таблицу. Доступные для записи внешние таблицы допускают только операции INSERT — SELECT, UPDATE, DELETE или TRUNCATE не допускаются.

Одна внешняя таблица ADB не может одновременно обрабатывать запросы SELECT и INSERT. Для каждого типа запросов требуется отдельная внешняя таблица

Ключевые слова для данного типа внешних таблиц не отличаются от типа Readable, за исключением нескольких особенностей

**В команде LOCATION ('протокол://[хост[:порт]]/путь/файл' [, ...])**

Для внешних таблиц с возможностью записи указывает расположение URI процесса gpfdist или протокола S3, который будет собирать выходные данные из сегментов Greenplum и записывать их в один или несколько именованных файлов. Для gpfdist путь относится к каталогу, из которого gpfdist обслуживает файлы (каталог, указанный при запуске программы gpfdist). Если указано несколько местоположений gpfdist, сегменты, отправляющие данные, будут равномерно распределены по доступным выходным расположениям. Например:

Также для данного типа таблиц возможно определить дистрибуцию.

**DISTRIBUTED BY ({столбец [opclass]}, [ ... ] ) | DISTRIBUTED RANDOMLY**

Используется для объявления дистрибуции базы данных Greenplum для внешней таблицы с параметром Readable. По умолчанию доступные для записи внешние таблицы распределяются случайным образом. Если исходная таблица, из которой вы экспортируете данные, имеет политику распределения хэшей, определение одного и того же столбца (столбцов) ключа распределения и класса операторов, oplcass, для доступной для записи внешней таблицы улучшит производительность выгрузки за счет устранения необходимости перемещения строк через interconnect. Когда вы вводите команду выгрузки, такую как INSERT INTO wex\_table SELECT \* FROM source\_table, выгружаемые строки могут быть отправлены непосредственно из сегментов в выходное расположение, если две таблицы имеют одинаковую политику распределения хэшей.

**FORSE QUOTE**

В режиме CSV для доступных для записи внешних таблиц принудительно заключать в кавычки все значения, отличные от NULL, в каждом указанном столбце. Если указано \*, то значения, отличные от NULL, будут заключены в кавычки во всех столбцах. Выход NULL никогда не цитируется.

С помощью параметра #transform=trans\_name вы можете указать преобразование, которое будет применяться при загрузке или извлечении данных. Trans\_name — это имя преобразования в файле конфигурации YAML, которое вы указываете при запуске утилиты gpfdist.

Одна внешняя таблица ADB не может одновременно обрабатывать запросы SELECT и INSERT. Для каждого типа запросов требуется отдельная внешняя таблица

PXF

Когда во внешних системах существует несколько связанных наборов данных, зачастую более эффективно объединять эти наборы удаленно и возвращать только результаты, а не договариваться о времени и требованиях к хранилищу для выполнения довольно дорогостоящей операции полной загрузки данных. Эту возможность предоставляет Greenplum Platform Extension Framework (PXF), расширение Greenplum, которое обеспечивает параллельный высокопроизводительный доступ к данным и объединенную обработку запросов к разнородным источникам через встроенные коннекторы.

Они сопоставляют определение внешней таблицы Greenplum с внешним источником данных, не требуя загрузки больших наборов данных в эту MPP-СУБД. PXF также предоставляет возможность выполнять части оператора SELECT (через push down запроса) в других источниках данных: облачных хранилищах (AWS S3, Google Cloud Storage или Microsoft Azure), распределенных файловых системах типа Hadoop HDFS, NoSQL-хранилищах (Apache Hive, HBase, mongoDB) и реляционных базах (Oracle Database, PostgreSQL, MySQL и пр.).

**Особенности PXF:**

• Общается с сегментами ADB через REST

• Содержит подключаемые модули – коннекторы, плагины – для доступа к внешним системам

• Read & Write (зависит от плагина)

• Можно разрабатывать свои коннекторы

Таблица PXF и таблица во внешней базе данных должны иметь одинаковую схему – столбцы должны иметь имена и типы, совпадающие с именами столбцов и типами внешней таблицы в ADB.

**Плагин поддерживает следующие типы данных:**

• INTEGER, BIGINT, SMALLINT

• REAL, FLOAT8

• NUMERIC

• BOOLEAN

• VARCHAR, BPCHAR, TEXT

• DATE

• TIMESTAMP

• BYTEA

**С помощью PXF вы можете использовать SQL для запросов к этим разнородным системам источникам:**

• Hadoop, Hive, и HBase

• Azure Blob Storage и Azure Data Lake

• AWS S3

• Minio

• Google Cloud Storage

• Базы данных SQL включая Apache Ignite, Hive, MySQL, ORACLE, Microsoft SQL Server, DB2, and PostgreSQL (через JDBC)

**И к этим форматам данных:**

• Avro, AvroSequenceFile

• JSON

• ORC

• Parquet

• RC-файл

• SequenceFile

• Txt (обычный, с разделителями, встроенный перевод строки)

Используя PXF для получения данных из внешней системы в Greenplum возможно реализовать **следующие возможности:**

• Выполнять запросы к внешним данным, при этом не храня физически данные в Greenplum

• Загружать выбранное подмножество данных из внешней таблицы в Greenplum

• Возможность выполнять сложные запросы одновременно используя данные из внешней таблицы и данные из базы данные GP

• Возможность записи во внешние источники

**Примеры PXF таблиц:**

CREATE EXTERNAL TABLE adb.dl.bills2\_ext (plant varchar,calday date,calmonth int8,billnum int8

)

LOCATION ( 'pxf://dl.bills2?PROFILE=Jdbc&JDBC\_DRIVER=org.postgresql.Driver&DB\_URL=jdbc:postgresql://192.168.114.107:5432/postgres&USER=admin&PASS=1234'

) ON ALL

FORMAT 'CUSTOM' ( FORMATTER='pxfwritable\_import' )

ENCODING 'UTF8’;

CREATE EXTERNAL TABLE sales(id integer, cdate date, amt float8, grade text

)

LOCATION ('pxf://sales?PROFILE=JDBC&JDBC\_DRIVER=com.mysql.jdbc.Driver&DB\_URL=jdbc:mysql://192.168.200.6:3306/demodb&PARTITION\_BY=cdate:date&RANGE=2008-01-01:2010-01-01&INTERVAL=1:year')

FORMAT 'CUSTOM' (FORMATTER='pxfwritable\_import');

**Строка подключения:**

pxf://<full\_external\_table\_name>?<pxf\_parameters><jdbc\_required\_parameters><jdbc\_login\_parameters><plugin\_parameters>

**full\_external\_table\_name**

Имя таблицы, к которой вы хотите подключиться.

**pxf\_parameters**

В этом параметре можно указать PROFILE=JDBC, если база к которой вы подключаетесь поддерживает JDBC либо дополнительные параметры для баз не поддерживающих JDBC.

FRAGMENTER=org.apache.hawq.pxf.plugins.jdbc.JdbcPartitionFragmenter   
&ACCESSOR=org.apache.hawq.pxf.plugins.jdbc.JdbcAccessor   
&RESOLVER=org.apache.hawq.pxf.plugins.jdbc.JdbcResolver}

**jdbc\_required\_parameters**

&JDBC\_DRIVER=<external\_database\_jdbc\_driver> - Указание драйвера, установленного на сегменты на нодах кластера.

&DB\_URL=<external\_database\_url> - Путь к внешней базе

**jdbc\_login\_parameters**

Параметры jdbc\_login\_parameters опциональны, но если вы решили их указать, то необходимо использовать оба параметра.

&USER=<external\_database\_login> - Логин пользователя внешней базы данных

&PASS=<external\_database\_password> - Пароль пользователя внешней базы данных

**plugin\_parameters**

&BATCH\_SIZE=<batch\_size> - Пакетная обработка

* *integer > 1* – используется пакет заданного размера;
* *integer < 0* – используется безразмерный пакет (все записи отправляются одним огромным JDBC-запросом). Настройка может вызвать ошибки, так как каждая база данных имеет собственный лимит на размер запросов JDBC;
* *0* или *1* – пакетирование не используется.

&POOL\_SIZE=<pool\_size> - Пул потоков

Для включения пула потоков необходимо создать внешнюю таблицу с параметром POOL\_SIZE с одним из следующих значений:

* *integer > 1* – пул потоков состоит из заданного количества потоков;
* *integer < 1* – количество потоков в пуле равно количеству процессоров в системе;
* *integer = 1* – не использовать пул потоков.

&PARTITION\_BY=<column>:<column\_type> - Указание на столбец, который используется в качестве столбца партиционирования. Можно использовать только один столбец.

&RANGE=<start\_value>:<end\_value> - Диапазон данных для запроса. Если тип партиционирования enum, то параметр range, должен быть списком. В котором каждое значение формирует собственную партицию. Если тип данных столбца партиционирования int или date, то тогда параметр &RANGE должен быть конечным, замкнутым слева диапазоном

[&INTERVAL=<value>[:<unit>]] – Необходим для партиционирования типа int и date.

**Примеры plugin\_parameters:**

&PARTITION\_BY=id:int&RANGE=42:142&INTERVAL=2

&PARTITION\_BY=createdate:date&RANGE=2008-01-01:2010-01-01&INTERVAL=1:month

&PARTITION\_BY=grade:enum&RANGE=excellent:good:general:bad

GPFDIST

**GPFDIST** - утилита для чтения и записи данных из файлов, расположенных на удаленных серверах. Она установлена на всех хостах-сегментах кластера Greenplum и обеспечивает параллельную загрузку данных, распределяя их между сегментами равномерно или согласно заданному ключу дистрибуции. Это повышает производительность при работе с большими объемами внешних данных.

Когда вы запускаете экземпляр gpfdist, вы указываете порт и путь к каталогу, содержащему файлы для чтения или путь для записи файлов.

Условие LOCATION для команды CREATE EXTERNAL TABLE соединяет определение внешней таблицы с одним или несколькими экземплярами gpfdist.

Если внешняя таблица доступна для чтения, gpfdist сервер считывает записи данных из файлов из указанного каталога, упаковывает их в блок данных и отправляет блок в ответ на запрос сегмента Greenplum. Сегменты распаковывают полученные строки и распределяют их в соответствии с дистрибуцией внешней таблицы.

Если внешняя таблица доступна для записи, сегменты отправляют блоки строк в запросе к gpfdist, а gpfdist записывает их во внешний файл.

GPFDIST может работать с любыми текстовыми файлами, которые содержат разделители, а также со сжатыми файлами gzip и bzip2 или любыми текстовыми файлами с разделителями, поддерживаемыми командой CREATE EXTERNAL TABLE.

**Для чтения или записи файлов на внешнем сервере:**

1. Установите и запустите GPFDIST на удаленном сервере, где находятся нужные файлы.
2. Создайте внешнюю таблицу, которая будет ссылаться на эти файлы, в базе данных Greenplum.

Разберем подробно параметры, с которыми возможно развернуть утилиту GPFDIST.

**-d directory**

Каталог, из которого gpfdist будет обслуживать файлы для внешних таблиц, доступных для чтения, или создавать выходные файлы для внешних таблиц, доступных для записи. Если не указано, по умолчанию используется текущий каталог.

**-p http\_port**

Порт HTTP, на котором gpfdist будет обслуживать файлы. По умолчанию 8080.

**-P last\_http\_port**

Последний номер порта в диапазоне номеров портов HTTP (от http\_port до last\_http\_port включительно), на котором gpfdist будет пытаться обслуживать файлы. Gpfdist обслуживает файлы на первом номере порта в диапазоне, к которому он успешно подключится.

**-l log\_file**

Полный путь и имя log файла, в котором должны регистрироваться стандартные сообщения (по умолчанию - инфо сообщения, предупреждения и ошибки).

**-t timeout**

Устанавливает время, в течение которого база данных Greenplum устанавливает соединение с процессом gpfdist. По умолчанию 5 секунд. Допустимые значения: от 2 до 7200 секунд (2 часа). Может потребоваться увеличение в системах с большим сетевым трафиком.

**-m max\_length**

Устанавливает максимально допустимую длину строки данных в байтах. Значение по умолчанию — 32768. Следует использовать, когда данные содержат очень широкие строки. Допустимый диапазон: от 32 КБ до 256 МБ. (Верхний предел составляет 1 МБ в системах Windows.)

Примечание. Проблемы с памятью могут возникнуть, если вы укажете большую максимальную длину строки и запустите большое количество gpfdist одновременно. Например, установив это значение на максимум 256 МБ с 96 одновременными gpfdist процессам вам потребуется примерно 24 ГБ памяти ((96 + 1) х 246 МБ).

**- s**

Включает упрощенное ведение журнала. Если указан этот параметр, только сообщения уровня “предупреждение” и выше записываются в лог файл. Инфо сообщения записываются не будут. Если этот параметр не указан, в файл журнала будут записаны все сообщения

Вы можете указать этот параметр, чтобы уменьшить объем информации, записываемой в файл журнала.

**- S (use O\_SYNC)**

Открывает файл для синхронного ввода-вывода, без использования кеша

**- w time**

Устанавливает количество секунд, в течение которых база данных Greenplum задерживает закрытие целевого файла. Значение по умолчанию равно 0, без задержки. Максимальное значение — 7200 секунд (2 часа).

При работе нескольких сегментов при записи данных возможна задержка. Для того чтобы запись в файл прошла успешно необходимо указать данный параметр

**-- ssl certificate\_path**

Добавляет шифрование SSL к данным, передаваемым с помощью gpfdist. После выполнения gpfdistс- ssl certificate\_path вариант, единственный способ загрузить данные с этого файлового сервера — это gpfdist://протокол.

Расположение, указанное в certificate\_path, должно содержать следующие файлы:

Файл сертификата сервера, server.crt

Файл закрытого ключа сервера, server.key

Доверенные центры сертификации, root.crt

Корневой каталог (/) не может быть указан как certificate\_path .

**-- sslclean wait\_time**

При запуске утилиты с--ssl Параметр устанавливает количество секунд, в течение которых утилита откладывает закрытие сеанса SSL и очистку ресурсов SSL после завершения записи данных в сегмент базы данных Greenplum или из него. Значение по умолчанию равно 0, без задержки. Максимальное значение составляет 500 секунд. Если задержка увеличивается, скорость передачи уменьшается.

В некоторых случаях может возникать ошибка при копировании больших объемов данных: “сервер gpfdist закрыл соединение”. Чтобы избежать ошибки, вы можете добавить задержку, например--sslclean 5.

**-c config.yaml**

Указывает правила трансформации, которые gpfdist использует для выбора преобразования при загрузке или извлечении данных. Gpfdist файл конфигурации представляет собой YAML 1.1 документ.

Эта опция недоступна на платформах Windows.

**-v (verbose)**

Подробный режим показывает прогресс и сообщения о состоянии.

**-V (very verbose)**

Подробный режим показывает все выходные сообщения, созданные утилитой.

**-? (помощь)**

Отображает интерактивную справку.

**-- version**

Отображает версию утилиты.

Источники информации (ссылки)

1. [Разница между ODS и EDW](https://russianblogs.com/article/84351390331/)
2. [Enterprise Data Warehouse: компоненты, основные концепции и типы архитектур EDW](https://habr.com/ru/post/693360/)
3. [INSERT](https://gpdb.docs.pivotal.io/6-2/ref_guide/sql_commands/INSERT.html)
4. [INSERT (VMware)](https://docs.vmware.com/en/VMware-Tanzu-Greenplum/6/greenplum-database/GUID-ref_guide-sql_commands-INSERT.html)
5. [COPY](https://gpdb.docs.pivotal.io/6-3/ref_guide/sql_commands/COPY.html)
6. [Loading Data with COPY](https://gpdb.docs.pivotal.io/6-3/admin_guide/load/topics/g-loading-data-with-copy.html)
7. [CREATE EXTERNAL TABLE](https://docs.vmware.com/en/VMware-Tanzu-Greenplum/6/greenplum-database/GUID-ref_guide-sql_commands-CREATE_EXTERNAL_TABLE.html)
8. [Greenplum Platform Extension Framework (PXF)](https://greenplum.docs.pivotal.io/6-7/pxf/overview_pxf.html)
9. [Accessing External Data with PXF](https://docs.vmware.com/en/VMware-Tanzu-Greenplum/6/greenplum-database/GUID-admin_guide-external-pxf-overview.html)
10. [Как устроен PXF Greenplum: архитектура и принципы работы](https://www.bigdataschool.ru/blog/pxf-greenplum-architecture.html)
11. [Интеграция кластера ADB с JDBC-совместимыми источниками](https://docs.arenadata.io/adb/PXFJDBC/index.html)
12. [Использование внешних таблиц](https://cloud.yandex.ru/docs/managed-greenplum/operations/external-tables)
13. [GPFDIST](https://greenplum.docs.pivotal.io/6-3/utility_guide/ref/gpfdist.html)
14. [Using the Greenplum Parallel File Server (gpfdist)](https://docs.vmware.com/en/VMware-Tanzu-Greenplum/6/greenplum-database/GUID-admin_guide-external-g-using-the-greenplum-parallel-file-server--gpfdist-.html)