

Руководство пользователя SDP AnalyticDB

Оглавление

1	Введение. О сервисе	3
2	Доступ к базе данных	4
2.1	Подключение к базе данных	4
2.2	Поддерживаемые клиентские приложения	4
2.3	Клиентские приложения SDP AnalyticDB	5
2.4	Подключение с помощью psql	5
3	Хранение информации и основные операции в SDP AnalyticDB.....	7
3.1	Сравнение хранилищ типа “heap” и “append-optimized”	7
3.2	Хранение данных по столбцам и строкам	7
3.3	Сжатие	7
3.4	Распределение	7
3.5	Управление памятью	8
3.6	Партиционирование	8
3.7	Индексы	9
3.8	Очереди ресурсов.....	9
4	Команды SQL.....	10

1 Введение. О сервисе

SDP AnalyticDB – сервис СУБД аналитических витрин хранилища данных, позволяющий эффективно решать задачи построения аналитических хранилищ данных больших объемов, предоставляя к ним полноценный SQL-доступ. SDP AnalyticDB позволяет решать следующие задачи:

- Обработка больших объемов данных
- Настройка ограничения доступа к данным в распределенной файловой системе
- Репликация данных на уровне узлов кластеров для повышения отказоустойчивости файловой системы

Функционал SDP AnalyticDB:

- Возможность создавать новые базы данных с идентичной логической структурой данных (деление по темам);
- Запись, хранение, поиск, редактирование данных;
- Модификация базы данных;
- Ввод основной массы данных посредством считывания (записи) массива данных;
- Контроль достоверности вводимых данных в каждом поле таблиц базы данных (обязательное поле, умолчание, шаблон и т.д.);
- Нахождение слов, выделение, просмотр и редактирование таблиц базы данных с удобным графическим интерфейсом;
- Резервное копирование всех данных кластера базы данных автоматически и с заданной регулярностью.

2 Доступ к базе данных

В этом разделе описываются различные клиентские инструменты, которые вы можете использовать для подключения к SDP AnalyticDB.

2.1 Подключение к базе данных

Пользователи могут подключаться к SDP AnalyticDB с помощью клиентской программы, совместимой с PostgreSQL, например, `psql`. Пользователи и администраторы всегда подключаются к SDP AnalyticDB через машину мастера, сегменты не могут принимать клиентские соединения.

Чтобы установить соединение с сервером мастера SDP AnalyticDB, вам необходимо знать следующую информацию о подключении и соответствующим образом настроить клиентскую программу.

Таблица 1 – Параметры подключения

Параметр подключения	Описание	Переменная окружения
Имя приложения	Имя приложения, которое подключается к базе данных. Значение по умолчанию, содержащееся в параметре подключения <code>application_name</code> , <code>-psql</code> .	<code>\$PGAPPNAME</code>
Имя базы данных	Имя базы данных, к которой вы хотите подключиться. Для вновь инициализированной системы используйте базу данных <code>postgres</code> для первого подключения.	<code>\$PGDATABASE</code>
Имя хоста	Имя хоста сервера мастера SDP AnalyticDB. Хост по умолчанию – это локальный хост.	<code>\$PGHOST</code>
Порт	Номер порта, на котором работает инстанс мастера SDP AnalyticDB. По умолчанию 5432.	<code>\$PGPORT</code>
Имя пользователя	Имя пользователя (роли) базы данных для подключения. Это не обязательно имя пользователя вашей ОС. Если вы не уверены, какое имя пользователя базы данных у вас, обратитесь к администратору SDP AnalyticDB. Обратите внимание, что каждая система SDP AnalyticDB имеет одну учётную запись суперпользователя, которая создается автоматически во время инициализации. Эта учётная запись имеет то же имя, что и имя пользователя ОС, инициализировавшего систему SDP AnalyticDB (обычно <code>gadmin</code>).	<code>\$PGUSER</code>

2.2 Поддерживаемые клиентские приложения

Пользователи могут подключаться к SDP AnalyticDB с помощью различных клиентских приложений:

- 1) Ряд клиентских приложений SDP AnalyticDB входит в комплект поставки SDP AnalyticDB. Клиентское приложение `psql` предоставляет интерактивный интерфейс командной строки для SDP AnalyticDB.

- 2) Используя стандартные интерфейсы приложений баз данных, такие как ODBC и JDBC, пользователи могут создавать свои собственные клиентские приложения, которые взаимодействуют с SDP AnalyticDB.
- 3) Большинство клиентских инструментов, использующих стандартные интерфейсы базы данных, такие как ODBC и JDBC, можно настроить для подключения к SDP AnalyticDB.

2.3 Клиентские приложения SDP AnalyticDB

SDP AnalyticDB поставляется с установленным рядом клиентских служебных приложений, расположенных в каталоге \$GPHOME/bin вашей установки хоста мастера SDP AnalyticDB. Ниже приведены наиболее часто используемые клиентские служебные приложения:

Таблица 2 – Часто используемые клиентские приложения

Имя	Использование
createdb	Создание новой базы данных.
createlang	Определение нового процедурного языка.
createuser	Определение новой роли базы данных.
dropdb	Удаление базы данных.
droplang	Удаление процедурного языка.
dropuser	Удаление роли.
psql	Интерактивный терминал PostgreSQL.
reindexdb	Переиндексация базы данных.
vacuumdb	Сбор мусора и анализ базы данных.

Примечание

`createlang` и `droplang` устарели и могут быть удалены в будущем выпуске.

При использовании этих клиентских приложений вы должны подключиться к базе данных через инстанс мастера SDP AnalyticDB. Вам нужно будет знать имя вашей целевой базы данных, имя хоста и номер порта мастера, а также имя пользователя базы данных для подключения. Эта информация может быть предоставлена в командной строке с помощью параметров `-d`, `-h`, `-p` и `-U` соответственно. Если найден аргумент, не принадлежащий ни одному параметру, он будет сначала интерпретирован как имя базы данных.

Все эти параметры имеют значения по умолчанию, которые будут использоваться, если параметр не указан. Хост по умолчанию – это локальный хост. Номер порта по умолчанию – 5432. Имя пользователя по умолчанию – это имя пользователя вашей ОС, как и имя базы данных по умолчанию. Обратите внимание, что имена пользователей ОС и имена пользователей SDP AnalyticDB не обязательно совпадают. Если значения по умолчанию неверны, вы можете установить для переменных окружения `PGDATABASE`, `PGHOST`, `PGPORT` и `PGUSER` соответствующие значения или использовать файл `psql ~/.pgpass` для хранения часто используемых паролей.

2.4 Подключение с помощью psql

В зависимости от используемых значений по умолчанию или установленных вами переменных окружения следующие примеры показывают, как получить доступ к базе данных через `psql`:

```
$ psql -d gpdatabase -h master_host -p 5432 -U gpadmin
```

```
$ psql gpdatabase
```

```
$ psql
```

Если пользовательская база данных ещё не создана, вы можете получить доступ к системе, подключившись к базе данных postgres. Например:

```
$ psql postgres
```

После подключения к базе данных psql отображает подсказку с именем базы данных, к которой в настоящее время подключён psql, за которым следует строка => (или =#, если вы являетесь суперпользователем базы данных). Например:

```
gpdatabase =>
```

В командной строке вы можете ввести команды SQL. Команда SQL должна заканчиваться на ; (точка с запятой) для отправки на сервер и выполнения. Например:

```
=> SELECT * FROM mytable;
```

3 Хранение информации и основные операции в SDP AnalyticDB

3.1 Сравнение хранилищ типа “heap” и “append-optimized”

Существуют два типа хранения данных в таблицах – heap и append-optimized. Хранение типа heap следует использовать для таблиц и партиций, которые получают итеративные пакетные, однострочные, а также синхронные операции UPDATE, DELETE и INSERT.

Хранение append-optimized следует использовать для таблиц и партиций, которые редко обновляются после начальной загрузки и в которых последующие вставки выполняются только в больших пакетных операциях.

Внимание.

Никогда не выполняйте однострочные операции INSERT, UPDATE или DELETE в таблицах типа append-optimized.

Внимание.

Никогда не выполняйте параллельные пакетные операции UPDATE или DELETE в append-optimized (это не касается одновременных операций вставки INSERT).

3.2 Хранение данных по столбцам и строкам

Построчное хранение необходимо использовать для рабочих нагрузок с итеративными транзакциями, в которых требуются обновления и частые вставки. Также его следует использовать для общих целей или в случае, если рабочие нагрузки носят смешанный характер.

Хранить данные по столбцам стоит в том случае, когда запросы необходимы для поиска узкого спектра данных. Также хранение по столбцам подходит при наличии отдельных столбцов, которые регулярно обновляются без изменения других столбцов в строке.

3.3 Сжатие

Для улучшения ввода-вывода в системе следует использовать сжатие для append-optimized таблиц. Параметры сжатия столбцов следует установить на уровне, на котором находятся данные. Рекомендуется балансировать более высокий уровень сжатия с временем и циклами ЦП, необходимыми для сжатия и распаковки данных.

3.4 Распределение

Для корректного распределения данных необходимо следовать рекомендациям:

- 1) Задать значения столбца или выбрать случайное распределение для всех таблиц. Не использовать значение по умолчанию.
- 2) Использовать один столбец, который будет равномерно распределять данные по всем сегментам.
- 3) Не распределять на столбцы, которые будут использоваться при запросе WHERE.

- 4) Не распределять на даты или временные метки.
- 5) Никогда не распределять и не разделять таблицы в одном столбце.
- 6) Достичь локальных объединений для значительного повышения производительности, распределив их в одном столбце для больших таблиц, которые обычно объединяются вместе.
- 7) Убедиться, что данные равномерно распределены после начальной загрузки и после инкрементных нагрузок.
- 8) В конечном итоге убедиться, что нет искажений данных.

3.5 Управление памятью

Для управления памятью необходимо следовать рекомендациям:

- 1) Установить значение параметра `vm.overcommit_memory` на 2.
- 2) Не настраивать ОС для использования огромных страниц.
- 3) Использовать `gp_vmem_protect_limit`, чтобы установить максимальный размер памяти, которую инстанс может выделить для работы, выполняющейся в каждом сегментном хосте.
- 4) Не допускать, чтобы значение `gp_vmem_protect_limit` превышало значения физического RAM в системе.
- 5) Установить правильное значение для `gp_vmem_protect_limit` следующим образом: $(\text{SWAP} + (\text{RAM} \times \text{vm.overcommit_ratio})) \times 0.9 / \text{number_segments_per_server}$
- 6) Использовать `statement_mem` для выделения памяти, используемой для запроса на сегмент db.
- 7) Использовать очереди ресурсов для установки количества активных запросов (`ACTIVE_STATEMENTS`) и суммы памяти (`MEMORY_LIMIT`), которые могут использоваться запросами в очереди.
- 8) Связать всех пользователей с очередью ресурсов. Не использовать значения, установленные по умолчанию.
- 9) Установить `PRIORITY` в соответствии с реальными потребностями очереди для рабочей нагрузки.
- 10) Убедиться, что распределение памяти очереди ресурсов не превышает `gp_vmem_protect_limit`.
- 11) Обновить параметры очереди ресурсов в соответствии с ежедневным потоком операций.

3.6 Партиционирование

При партиционировании необходимо следовать рекомендациям:

- 1) Можно партиционировать только больших таблиц. Не партиционируйте маленькие таблицы.
- 2) Используйте партиционирование только в том случае, если удаление партиций (сокращение партиций) может быть достигнуто на основе критериев запроса.
- 3) Выбирайте партиционирование по диапазону вместо партиционирования по списку.
- 4) Разбивайте таблицу на партиции на основе часто используемого столбца, например, столбца даты.
- 5) Никогда не разбивайте и не распределяйте таблицы по одному столбцу.
- 6) Не используйте партиционирование по умолчанию.
- 7) Не используйте многоуровневое партиционирование; создавайте меньше партиций с большим количеством данных в каждой партиции.

- 8) Убедитесь, что запросы выборочно сканируют партиционированные таблицы путём изучения плана запроса EXPLAIN.
- 9) Не создавайте слишком много партиций с хранилищем, ориентированным на столбцы, из-за общего количества физических файлов в каждом сегменте: физические файлы = сегменты x столбцы x партиции.

3.7 Индексы

При работе с индексами необходимо следовать рекомендациям:

- 1) Обычно индексы в SDP AnalyticDB не нужны.
- 2) Создавайте индекс в одном столбце таблицы (хранение данных по столбцам) для сквозного доступа для таблиц высокой кардинальности, требующих запросов с высокой избирательностью.
- 3) Не индексируйте часто обновляемые столбцы.
- 4) Рассмотрите возможность удаления индексов перед загрузкой данных в таблицу. После загрузки заново создайте индексы для таблицы.
- 5) Создавайте выборочные индексы B-дерева.
- 6) Не создавайте bitmap-индексы в столбцах, которые находятся в процессе обновления.
- 7) Избегайте использования bitmap-индексов для уникальных столбцов, данных с очень большой или маленькой кардинальностью. Bitmap-индексы работают лучше всего, когда столбец имеет маленькую кардинальность – от 100 до 100 000 различных значений.
- 8) Не используйте bitmap-индексы для транзакционных нагрузок.
- 9) Как правило, не индексируйте партиционированные таблицы. Если индексы необходимы, столбцы индекса должны отличаться от столбцов партиций.

3.8 Очереди ресурсов

Для управления рабочей нагрузкой в кластере необходимо использовать очереди ресурсов:

- 1) Свяжите все роли с определяемой пользователем очередью ресурсов.
- 2) Используйте параметр `ACTIVE_STATEMENTS`, чтобы ограничить количество активных запросов, которые члены определённой очереди могут выполнять одновременно.
- 3) Используйте параметр `MEMORY_LIMIT` для управления общим объёмом памяти, который могут использовать запросы, выполняемые через очередь.
- 4) Динамически изменяйте очереди ресурсов в соответствии с рабочей нагрузкой и временем суток.

4 Команды SQL

Внимание.

С описанием, синтаксисом, используемыми параметрами и примерами SQL-команд вы можете ознакомиться на [официальном сайте PostgreSQL](#).

В SDP AnalyticDB доступны следующие команды SQL:

- `ABORT` – прервать текущую транзакцию;
- `ALTER AGGREGATE` – изменить определение агрегатной функции;
- `ALTER COLLATION` – изменить определение правила сортировки;
- `ALTER CONVERSION` – изменить определение перекодировки;
- `ALTER DATABASE` – изменить атрибуты базы данных;
- `ALTER DEFAULT PRIVILEGES` – определить права доступа по умолчанию;
- `ALTER DOMAIN` – изменить определение домена;
- `ALTER EXTENSION` – изменить определение расширения;
- `ALTER EXTERNAL TABLE` – изменить определение внешней таблицы;
- `ALTER FOREIGN DATA WRAPPER` – изменить определение обёртки сторонних данных;
- `ALTER FOREIGN TABLE` – изменить определение сторонней таблицы;
- `ALTER FUNCTION` – изменить определение функции;
- `ALTER GROUP` – изменить имя роли или членство;
- `ALTER INDEX` – изменить определение индекса;
- `ALTER LANGUAGE` – изменить определение процедурного языка;
- `ALTER OPERATOR` – изменить определение оператора;
- `ALTER OPERATOR CLASS` – изменить определение класса операторов;
- `ALTER OPERATOR FAMILY` – изменить определение семейства операторов;
- `ALTER PROTOCOL` – изменить определение протокола;
- `ALTER RESOURCE GROUP` – изменить пределы группы ресурсов;
- `ALTER RESOURCE QUEUE` – изменить пределы очереди ресурсов;
- `ALTER ROLE` – изменить роль в базе данных;
- `ALTER SCHEMA` – изменить определение схемы;
- `ALTER SEQUENCE` – изменить определение генератора последовательности;
- `ALTER SERVER` – изменить определение стороннего сервера;
- `ALTER TABLE` – изменить определение таблицы;
- `ALTER TABLESPACE` – изменить определение табличного пространства;
- `ALTER TEXT SEARCH CONFIGURATION` – изменить определение конфигурации текстового поиска;
- `ALTER TEXT SEARCH DICTIONARY` – изменить определение словаря текстового поиска;
- `ALTER TEXT SEARCH PARSER` – изменить определение анализатора текстового поиска;
- `ALTER TEXT SEARCH TEMPLATE` – изменить определение шаблона текстового поиска;
- `ALTER TYPE` – изменить определение типа;
- `ALTER USER` – изменить роль в базе данных;
- `ALTER USER MAPPING` – изменить определение сопоставления пользователей;

- ALTER VIEW – изменить определение представления;
- ANALYZE – собрать статистику по базе данных;
- BEGIN – начать блок транзакции;
- CHECKPOINT – произвести контрольную точку в журнале предзаписи;
- CLOSE – закрыть курсор;
- CLUSTER – кластеризовать таблицу согласно индексу;
- COMMENT – задать или изменить комментарий объекта;
- COMMIT – зафиксировать текущую транзакцию;
- COPY – копировать данные между файлом и таблицей;
- CREATE AGGREGATE – создать агрегатную функцию;
- CREATE CAST – создать приведение;
- CREATE COLLATION – создать правило сортировки;
- CREATE CONVERSION – создать перекодировку;
- CREATE DATABASE – создать базу данных;
- CREATE DOMAIN – создать домен;
- CREATE EXTENSION – установить расширение;
- CREATE EXTERNAL TABLE – создать внешнюю таблицу;
- CREATE FOREIGN DATA WRAPPER – создать новую обёртку сторонних данных;
- CREATE FOREIGN TABLE – создать стороннюю таблицу;
- CREATE FUNCTION – создать функцию;
- CREATE GROUP – создать роль в базе данных;
- CREATE INDEX – создать индекс;
- CREATE LANGUAGE – создать процедурный язык;
- CREATE OPERATOR – создать оператор;
- CREATE OPERATOR CLASS – создать класс операторов;
- CREATE OPERATOR FAMILY – создать семейство операторов;
- CREATE PROTOCOL – создать протокол;
- CREATE RESOURCE GROUP – создать группу ресурсов;
- CREATE RESOURCE QUEUE – создать очередь ресурсов;
- CREATE ROLE – создать роль в базе данных;
- CREATE RULE – создать правило перезаписи;
- CREATE SCHEMA – создать схему;
- CREATE SEQUENCE – создать генератор последовательности;
- CREATE SERVER – создать сторонний сервер;
- CREATE TABLE – создать таблицу;
- CREATE TABLE AS – создать таблицу из результатов запроса;
- CREATE TABLESPACE – создать табличное пространство;
- CREATE TEXT SEARCH CONFIGURATION – создать конфигурацию текстового поиска;
- CREATE TEXT SEARCH DICTIONARY – создать словарь текстового поиска;
- CREATE TEXT SEARCH PARSER – создать анализатор текстового поиска;
- CREATE TEXT SEARCH TEMPLATE – создать шаблон текстового поиска;
- CREATE TYPE – создать новый тип данных;
- CREATE USER – создать роль в базе данных;
- CREATE USER MAPPING – создать сопоставление пользователя для стороннего сервера;
- CREATE VIEW – создать представление;

- DEALLOCATE – освободить подготовленный оператор;
- DECLARE – определить курсор;
- DELETE – удалить записи таблицы;
- DISCARD – очистить состояние сеанса;
- DO – выполнить анонимный блок кода;
- DROP AGGREGATE – удалить агрегатную функцию;
- DROP CAST – удалить приведение типа;
- DROP COLLATION – удалить правило сортировки;
- DROP CONVERSION – удалить преобразование;
- DROP DATABASE – удалить базу данных;
- DROP DOMAIN – удалить домен;
- DROP EXTENSION – удалить расширение;
- DROP EXTERNAL TABLE – удалить внешнюю таблицу;
- DROP FOREIGN DATA WRAPPER – удалить обёртку сторонних данных;
- DROP FOREIGN TABLE – удалить стороннюю таблицу;
- DROP FUNCTION – удалить функцию;
- DROP GROUP – удалить роль в базе данных;
- DROP INDEX – удалить индекс;
- DROP LANGUAGE – удалить процедурный язык;
- DROP OPERATOR – удалить оператор;
- DROP OPERATOR CLASS – удалить класс операторов;
- DROP OPERATOR FAMILY – удалить семейство операторов;
- DROP OWNED – удалить объекты базы данных, принадлежащие роли;
- DROP PROTOCOL – удалить протокол;
- DROP RESOURCE GROUP – удалить группу ресурсов;
- DROP RESOURCE QUEUE – удалить очередь ресурсов;
- DROP ROLE – удалить роль в базе данных;
- DROP RULE – удалить правило перезаписи;
- DROP SCHEMA – удалить схему;
- DROP SEQUENCE – удалить последовательность;
- DROP SERVER – удалить описание стороннего сервера;
- DROP TABLE – удалить таблицу;
- DROP TABLESPACE – удалить табличное пространство;
- DROP TEXT SEARCH CONFIGURATION – удалить конфигурацию текстового поиска;
- DROP TEXT SEARCH DICTIONARY – удалить словарь текстового поиска;
- DROP TEXT SEARCH PARSER – удалить анализатор текстового поиска;
- DROP TEXT SEARCH TEMPLATE – удалить шаблон текстового поиска;
- DROP TYPE – удалить тип данных;
- DROP USER – удалить роль в базе данных;
- DROP USER MAPPING – удалить сопоставление пользователя для стороннего сервера;
- DROP VIEW – удалить представление;
- END – зафиксировать текущую транзакцию;
- EXECUTE – выполнить подготовленный оператор;
- EXPLAIN – показать план выполнения оператора;
- FETCH – получить результат запроса через курсор;
- GRANT – определить права доступа;

- INSERT – добавить строки в таблицу;
- LOAD – загрузить файл разделяемой библиотеки;
- LOCK – заблокировать таблицу;
- MOVE – переместить курсор;
- PREPARE – подготовить оператор к выполнению;
- REASSIGN OWNED – сменить владельца объектов базы данных, принадлежащих заданной роли;
- REINDEX – перестроить индексы;
- RELEASE SAVEPOINT – высвободить ранее определённую точку сохранения;
- RESET – восстановить значение по умолчанию заданного параметра времени выполнения;
- REVOKE – отозвать права доступа;
- ROLLBACK – прервать текущую транзакцию;
- ROLLBACK TO SAVEPOINT – откатиться к точке сохранения;
- SAVEPOINT – определить новую точку сохранения в текущей транзакции;
- SELECT – получить строки из таблицы или представления;
- SELECT INTO – создать таблицу из результатов запроса;
- SET – изменить параметр времени выполнения;
- SET CONSTRAINTS – установить время проверки ограничений для текущей транзакции;
- SET ROLE – установить идентификатор текущего пользователя в рамках сеанса;
- SET SESSION AUTHORIZATION – установить идентификатор пользователя сеанса и идентификатор текущего пользователя в рамках сеанса;
- SET TRANSACTION – установить характеристики текущей транзакции;
- SHOW – показать значение параметра времени выполнения;
- START TRANSACTION – начать блок транзакции;
- TRUNCATE – опустошить таблицу или набор таблиц;
- UPDATE – изменить строки таблицы;
- VACUUM – провести сборку мусора и, возможно, проанализировать базу данных;
- VALUES – вычислить набор строк.