Arenadata™ Database

Версия - v6.7.0

Примечания к выпуску ADB 6

2020 Arenadata™

Оглавление

1 Новые функции **3**
- 1.1 Основные возможности PostgreSQL . . . 3
- 1.2 Алгоритм сжатия Zstandard . . . 4
- 1.3 Послабления для столбцов в распределенных таблицах . . . 4
- 1.4 Функции Ресурсных Групп . . . 5
- 1.5 Доработка процедурного языка PL/pgSQL . . . 5
- 1.6 Репликация данных таблицы . . . 5
- 1.7 Доработки параллельной обработки . . . 6
- 1.8 Дополнительные модули Contrib . . . 6
- 1.9 PXF версии 5.8.1 . . . 7
- 1.10 Дополнительные функции . . . 7

2 Beta-функции **8**

3 Измененные функции **9**

4 Удаленные функции **14**

5 Ключевые отличия ADB от open-source аналога **16**

Arenadata DB (ADB) – это сервер базы данных с массовой параллельной обработкой (MPP), который поддерживает хранилище данных следующего поколения и крупномасштабную аналитическую обработку. Благодаря автоматическому секционированию данных и выполнению параллельных запросов кластер может работать как один суперкомпьютер базы данных, работающий в десятки или сотни раз быстрее, чем традиционная база данных. ADB поддерживает SQL, параллельную обработку MapReduce и объемы данных от сотен гигабайт до сотен терабайт.

Important: Данная документация содержит информацию о выпуске Arenadata DB 6. Данный выпуск основан на версии Greenplum 6. Для выбора точной версии продукта необходимо перейти к соответствующей документации.

Important: Контактная информация службы поддержки – e-mail: info@arenadata.io

Глава 1

Новые функции

В **Arenadata DB 6** добавлено:

- *Основные возможности PostgreSQL*;
- *Алгоритм сжатия Zstandard*;
- *Послабления для столбцов в распределенных таблицах*;
- *Функции Ресурсных Групп*;
- *Доработка процедурного языка PL/pgSQL*;
- *Репликация данных таблицы*;
- *Доработки параллельной обработки*;
- *Дополнительные модули Contrib*;
- *PXF версии 5.8.1*;
- *Дополнительные функции*.

1.1 Основные возможности PostgreSQL

Arenadata DB 6 включает в себя несколько новых функций от **PostgreSQL** версий *8.4 – 9.4*.

Обработка типа данных *INTERVAL* улучшена в **PostgreSQL 8.4** в ключе синтаксического анализа литералов интервала в целях соответствия стандартам SQL. Это изменяет вывод для запросов, использующих литералы *INTERVAL* между версиями *5* и *6*. Например:

```
$ psql
psql (8.3.23)
Type "help" for help.

gpadmin=# select INTERVAL '1' YEAR;
 interval
----------
 00:00:00
(1 row)
```
```
``` sql
$ psql
psql (9.2beta2)
Type "help" for help.
```
```

```
gpadmin=# select INTERVAL '1' YEAR;
 interval
----------
 1 year
(1 row)
```

Так же в **Arenadata DB 6** включены следующие функции и изменения из **PostgreSQL**:

- Поддержка пользовательских преобразований ввода-вывода (PostgreSQL 8.4);
- Поддержка привилегий на уровне столбцов (PostgreSQL 8.4);
- Таблица каталога *pg_db_role_setting*, обеспечивающая поддержку настройки параметров конфигурации сервера для конкретной комбинации базы данных и роли (PostgreSQL 9.0);
- Значения в столбце *relkind* таблицы каталога *pg_class* изменены, чтобы соответствовать записям в PostgreSQL 9.3;
- Поддержка метода GIN-индекса (PostgreSQL 8.3);
- Поддержка Postgres Planner для метода доступа к индексам *SP-GiST* (PostgreSQL 9.2). GPORCA игнорирует индексы SP-GiST;
- Поддержка Postgres Planner для агрегатов упорядоченного набора (ordered-set aggregates) и агрегатов перемещения (moving-aggregates) (PostgreSQL 9.4).
- Поддержка типа данных *jsonb* (PostgreSQL 9.4);
- *DELETE*, *INSERT* и *UPDATE* поддерживают *CTE* (common table expression) условие *WITH* (PostgreSQL 9.1);
- Поддержка правил сортировки для задания ее порядка и классификации символов для данных на уровне столбцов (PostgreSQL 9.1). Но GPORCA поддерживает правила сортировки, только когда все столбцы в запросе используют одинаковые правила сортировки. Если столбцы в запросе используют разные правила сортировки, то ADB использует Postgres Planner.

1.2 Алгоритм сжатия Zstandard

В **Arenadata DB 6** для некоторых операций с базой данных добавлена поддержка сжатия *zstd* (Zstandard).

1.3 Послабления для столбцов в распределенных таблицах

В предыдущих выпусках при указании ограничения *UNIQUE* и условия *DISTRIBUTED BY* в операторе *CREATE TABLE* условие *DISTRIBUTED BY* должно было быть равно или являться левым подмножеством столбцов *UNIQUE*. **Arenadata DB 6** ослабляет это правило, так что любое подмножество столбцов *UNIQUE* принимается.

Это изменение также влияет на правила того, как **Arenadata DB 6** выбирает ключ распределения по умолчанию. Если параметр *gp_create_table_random_default_distribution* выключен (по умолчанию) и условие *DISTRIBUTED BY* не включено, то **ADB** выбирает ключ распределения таблицы, основываясь на следующем:

- Если указано условие *LIKE* или *INHERITS*, то ADB копирует ключ распределения из исходной или родительской таблицы;
- Если указаны ограничения *PRIMARY KEY* или *UNIQUE*, то ADB выбирает в качестве ключа распределения наибольшее подмножество из всех столбцов ключей;

- Если не указаны ни ограничения, ни условия *LIKE* или *INHERITS*, то ADB выбирает в качестве ключа распределения первый подходящий столбец (столбцы с геометрическими или пользовательскими типами данных не допускаются в качестве столбцов ключей распределения ADB).

1.4 Функции Ресурсных Групп

В **ADB** добавлены следующие новые функции для ресурсных групп:

- При настройке группы ресурсов больше не требуется указывать *MEMORY_LIMIT*. При указании `MEMORY_LIMIT=0` база данных использует глобальный пул общей памяти группы ресурсов для обслуживания запросов, выполняющихся в группе.
- При указании `MEMORY_SPILL_RATIO=0` для определения начального объема памяти оператора запроса база данных теперь использует параметр конфигурации сервера *statement_mem*.
- При совместном использовании настроек групп ресурсов `MEMORY_LIMIT=0` и `MEMORY_SPILL_RATIO=0` эти новые возможности обеспечивают схему управления памятью, аналогичную схеме, предоставляемой очередями ресурсов базы данных ADB.

Значения по умолчанию атрибутов *MEMORY_SHARED_QUOTA*, *MEMORY_SPILL_RATIO* и *MEMORY_LIMIT* для групп ресурсов *admin_group* и *default_group* установлены для использования схемы управления памятью, подобной очереди ресурсов, поэтому при первоначальном включении групп ресурсов запросы выполняются в аналогичной предыдущей среде памяти.

Таблица1.1.: Значения по умолчанию атрибутов ресурсных групп

Resource Group	admin_group	default_group
MEMORY_LIMIT	10	0
MEMORY_SHARED_QUOTA	80	80
MEMORY_SPILL_RATIO	0	0

1.5 Доработка процедурного языка PL/pgSQL

Процедурный язык PL/pgSQL в **Arenadata DB 6** включает поддержку следующих новых функций:

- Прикрепление текста *DETAIL* и *HINT* к выдаваемым пользователю сообщениям об ошибках. Также можно указать коды *SQLSTATE* и *SQLERRMSG* для возврата ошибки, вызванной пользователем (PostgreSQL 8.4);
- Оператор *RETURN QUERY EXECUTE*, задающий запрос для динамического выполнения (PostgreSQL 8.4);
- Условное выполнение с использованием оператора *CASE* (PostgreSQL 8.4). Подробнее в документации PostgreSQL.

1.6 Репликация данных таблицы

Команда *CREATE TABLE* поддерживает *DISTRIBUTED REPLICATED* в качестве политики распределения. Если указана эта политика распределения, база данных распределяет все строки таблицы по всем экземплярам сегмента в системе **ADB**.

Important: В пользовательских запросах к реплицированным таблицам нельзя ссылаться на скрытые системные столбцы (ctid, cmin, cmax, xmin, xmax и gp_segment_id), поскольку они не имеют единую,

однозначную величину. Для такого запроса ADB возвращает ошибку несуществующего столбца

1.7 Доработки параллельной обработки

Arenadata DB 6 включает следующие усовершенствования параллельной обработки:

- Global Deadlock Detector – предыдущие версии ADB препятствуют глобальной взаимоблокировке, удерживая исключительную блокировку таблиц для операций *UPDATE* и *DELETE*. Хотя эта стратегия действительно предотвращает deadlocks, это обеспечивается за счет низкой производительности при одновременных обновлениях. База данных Arenadata DB 6 включает в себя глобальный детектор взаимоблокировок – это внутренний процесс, который собирает и анализирует данные ожидания в кластере ADB. Если Global Deadlock Detector определяет, что deadlock существует, он ее прерывает, отменяя один или несколько внутренних процессов. По умолчанию детектор отключен, и для обновлений таблиц проводятся исключительные блокировки на уровне таблицы. А когда детектор включен, база данных ADB проводит исключительные блокировки на уровне строк, и в таком случае одновременные обновления допускаются;
- Transaction Lock Optimization – Arenadata DB 6 оптимизирует использование блокировки транзакций как при *BEGIN*, так и при *COMMIT* транзакции. Это особо выгодно при одновременных смешанных рабочих нагрузках;
- Upstream PostgreSQL Features – Arenadata DB 6 включает в себя основные функции PostgreSQL, в том числе fastpath lock, сокращающие конфликты блокировок. Это выгодно для одновременных коротких запросов и смешанных рабочих нагрузок;
- Команда *VACUUM* пропускает страницы, которые не может заблокировать. Это уменьшает частоту появления вакуума, который "застревает" в результате ожидания блокировки блока для очистки в то время, когда другая сессия удерживает блокировку этого блока. Теперь *VACUUM* пропускает блоки, который не может заблокировать сразу, и пытается повторить их блокировку позже;
- Команда *VACUUM* перепроверяет видимость блока после удаления "мертвых" кортежей. Если все оставшиеся кортежи в блоке видимы для текущих и будущих транзакций, блок помечается как полностью видимый;
- Таблицы, являющиеся частью иерархии партиционированных таблиц, но не содержащие данных, замораживаются, поэтому их не нужно очищать отдельно, и при этом они не влияют на вычисление количества оставшихся идентификаторов транзакций даже до того, как произойдет wraparound. Эти таблицы включают корневую и промежуточную таблицы в иерархии партиций и, если они являются append-optimized, связанные с ними таблицы метаданных. Это делает ненужной операцию *vacuum* для корневой партиции для сокращения ее возраста и устраняет вероятно излишнюю очистку всех дочерних таблиц.

1.8 Дополнительные модули Contrib

База данных **Arenadata DB 6** разворачивается с дополнительными модулями **PostgreSQL** и **Greenplum**:

- *auto_explain*;
- *diskquota*;
- *fuzzystrmatch*;
- *gp_sparse_vector*;
- *pageinspect*;
- *sslinfo*.

1.9 PXF версии 5.8.1

Arenadata DB 6 включает **PXF 5.8.1**, который вводит следующие новые и измененные функции:

- Коннектор PXF S3 теперь поддерживает доступ к данным *CSV* и *Parquet* на S3 с помощью сервиса Amazon S3 Select;
- PXF объединяет новые и обновленные библиотеки, чтобы обеспечить поддержку Java 11;
- В PXF добавлена поддержка типа *timestamptz* при записи данных *Parquet* во внешний источник данных;
- PXF теперь предоставляет команду сброса локального экземпляра PXF-сервера или всех экземпляров PXF-сервера в кластере в неинициализированное состояние;
- PXF больше не поддерживает уточнение *DELIMITER* в *CREATE EXTERNAL TABLE* команды *LOCATION URI*.

1.10 Дополнительные функции

Arenadata DB 6 также включает следующие функции, отличные от версии *5*:

- Recursive WITH Queries (Common Table Expressions) больше не считаются бета-функцией и теперь включены по умолчанию;
- *VACUUM* обновлен для упрощения пропуска страниц, которые нельзя заблокировать. Это изменение должно значительно снизить частоту "застревания" операции *VACUUM* во время ожидания завершения других сессий;
- Appendoptimized alias для опции хранения appendonly-таблицы;
- Новые gp_toolkit-представления *gp_resgroup_status_per_host* и *gp_resgroup_status_per_segment* отображают CPU группы ресурсов и использование памяти для каждого хоста и/или сегмента;
- Новое представление *gp_stat_replication* содержит статистику репликации, когда включено зеркалирование мастера или сегмента. Представление *pg_stat_replication* содержит только статистику репликации мастера;
- Программы *gpfdists* и *psql* в пакете Client and Loader Tools для Windows поддерживают шифрование OpenSSL;
- Arenadata DB 6 включает некоторые улучшения производительности PostgreSQL 9.6.
- Утилита *gpload* совместима с базой данных ADB 5.x.

Глава 2

Beta-функции

Поскольку база данных **ADB** основана на open-source аналоге, она включает в себя несколько экспериментальных функций, позволяющих заинтересованным разработчикам экспериментировать с их использованием в системах разработки. Обратная связь помогает стимулировать разработку данных функций, и они могут поддерживаться в будущих версиях продукта.

Important: Бета-функции не поддерживаются для развертывания в продуктивной среде

В **Arenadata DB 6** входят следующие экспериментальные функции:

- API-интерфейс плагина хранения для утилит **gpbackup** и **gprestore**. Партнеры, клиенты и разработчики OSS могут совместно использовать плагины при помощи **gpbackup** и **gprestore**;
- Коннекторы **Platform Extension** (**PXF**) для записи Parquet-данных.

Глава 3

Измененные функции

В **Arenadata DB 6** изменены следующие функции:

- Рабочие характеристики базы данных **ADB** при больших нагрузках изменились по сравнению с предыдущими версиями. В частности, можно заметить увеличение операций ввода-вывода на первичных сегментах из-за изменений, связанных с WAL-репликацией и другими функциями. Всем клиентам рекомендуется выполнять нагрузочное тестирование с использованием реальных данных, чтобы убедиться, что новая конфигурация кластера **ADB 6.0** отвечает их требованиям к производительности.
- Утилиты **gpbackup** и **gprestore** больше не устанавливаются с **Arenadata DB 6**, но они доступны отдельно и могут быть обновлены независимо от инсталляции базы данных.
- **ADB 6** использует новый алгоритм согласованного хеширования *jump consistent hash* для распределения данных по сегментам **ADB**. Алгоритм гарантирует, что после добавления новых сегментов в кластер **ADB 6** необходимо перемещать только те строки, которые хешируются в новый сегмент. Хэширование базы данных имеет характеристики производительности, схожие с предыдущими релизами системы, но должно обеспечивать более быстрое расширение базы данных. Важно обратить внимание, что новый алгоритм является более вычислительно интенсивным по сравнению с предыдущим, поэтому производительность *COPY* может несколько снизиться в системах, где производительность ограничена процессором.
- Хеш-функции предыдущей версии **ADB** представлены как классы нестандартных хеш-операторов с именем `cdbhash_*_ops`. Классы нестандартных операторов используются при обновлении базы данных **ADB** предыдущих версий. Унаследованные классы операторов совместимы друг с другом, но при смешении устаревших классов операторов с новыми, запросы требуют Redistribute Motions.

Параметр конфигурации сервера `gp_use_legacy_hashops` фиксирует – используются ли устаревшие хеш-функции или функции по умолчанию при создании таблиц, с указанным столбцом распределения

Системная таблица `gp_distribution_policy` теперь содержит больше информации о таблицах **ADB** и политике распределения табличных данных по сегментам, включая класс операторов хеш-функций распределения.

- Утилита **gpcheck** не включена в базу данных **Arenadata DB 6**.
- Изменился формат входного файла для утилит **gpmovemirrors**, **gpaddmirrors**, **gprecoverseg** и **gpexpand**. В качестве разделителя вместо использования символа двоеточия : новый формат файла использует символ вертикальной черты |. Например, в предыдущих выпусках строка во входном файле **gpexpand** выглядит:

```
sdw5:sdw5-1:50011:/gpdata/primary/gp9:11:9:p
```

Обновленный формат файла:

```
sdw5|sdw5-1|50011|/gpdata/primary/gp9|11|9|p
```

Кроме того, **gpaddmirrors** удаляет префикс зеркала из строк входного файла. Например, строка из предыдущей версии:

```
mirror0=0:sdw1:sdw1-1:52001:53001:54001:/gpdata/mir1/gp0
```

Обновленный формат:

```
0=0|sdw1|sdw1-1|52001|53001|54001|/gpdata/mir1/gp0
```

- **ADB** использует прямую диспетчеризацию для целевых запросов с *IS NULL* аналогично запросам, которые фильтруются по столбцу (столбцам) ключа распределения таблицы.
- Опция **gpinitsystem** для указания каталога данных резервного мастера изменена с `-F` на `-S`. Опция `-S` больше не определяет spread тип зеркалирования. Для указания конфигурации зеркалирования введена новая опция **gpinitsystem**: `--mirror-mode={group|spread}`.
- Значение по умолчанию параметра конфигурации сервера **log_rotation_size** изменилось с *0* до *1 ГБ*. Это меняет поведение по умолчанию ротации журнала, так что новый лог-файл открывается только тогда, когда в текущий файл записано более *1 ГБ* или когда он открыт в течение *24 часов*.
- Утилита **gpssh-exkeys** теперь требует настроенного доступа по SSH без пароля с мастер-хоста ко всем остальным хостам в кластере. После запуска утилиты доступ по SSH без пароля устанавливается с каждого хоста на любой другой хост.
- Изменено поведение smart shutdown в **gpstop**. Ранее при запуске `gpstop -M smart` (или просто `gpstop`) утилита осуществляла выход, выводя сообщение об активных клиентских подключениях. Теперь **gpstop** ожидает завершения текущих подключений, прежде чем завершить выключение. Если какие-либо соединения остаются открытыми после истечения времени ожидания или прерываются с помощью *CTRL-C* утилита выводит список открытых соединений и запрос о том, продолжать ли ожидание завершения соединения или выполнить незамедлительное отключение. Период ожидания по умолчанию составляет *120 секунд* и может быть изменен с помощью опции `-t timeout_seconds`.
- В системных представлениях **pg_stat_activity** и **pg_stat_replication** столбец *procpid* переименован в *pid*, чтобы соответствовать связанному изменению в **PostgreSQL 9.2**.
- В системной таблице **pg_proc** столбец *proiswin* переименован в *proiswindow* и перемещен в таблицу, чтобы соответствовать одноименной системной таблице в **PostgreSQL 8.4**.
- Запросы, использующие *SELECT DISTINCT* и *UNION/INTERSECT/EXCEPT*, больше не обязаны возвращать отсортированную информацию. Ранее эти запросы всегда удаляли повторяющиеся строки с помощью обработки Sort/Unique. Теперь они реализуют хеширование с целью соответствия поведению, введенному в **PostgreSQL 8.4**, и этот метод не осуществляет сортировку. В случае если приложение требует отсортированного вывода для таких запросов, необходимо явно использовать условие *ORDER BY* в самих запросах. Важно обратить внимание, что *SELECT DISTINCT ON* не использует хеширование, поэтому его поведение не отличается от предыдущих версий.
- В схеме **gp_toolkit** представление **gp_resgroup_config** больше не содержит столбцы *proposed_concurrency*, *proposed_memory_shared_quota* и *proposed_memory_spill_ratio*.
- В системной таблице **pg_resgroupcapability** столбец *proposed* удален.
- В системной таблице **pg_database** столбец *datconfig* удален. Теперь для отслеживания параметров конфигурации сервера для каждой базы данных и для каждой роли **ADB** использует системную таблицу **pg_db_role_setting** (**PostgreSQL 9.0**).
- В системной таблице **pg_aggregate** столбец *aggordered* удален и добавлено несколько новых столбцов для поддержки агрегатов ordered-set и moving-aggregates с помощью **Postgres Planner** (**PostgreSQL**

9.4). Сигнатуры команд *ALTER/CREATE/DROP AGGREGATE SQL* также обновлены, чтобы отразить изменения каталога **pg_aggregate**.

- В системной таблице **pg_authid** столбец *rolconfig* удален. Теперь для отслеживания параметров конфигурации сервера для каждой базы данных и для каждой роли **ADB** использует системную таблицу **pg_db_role_setting** (**PostgreSQL 9.0**).
- При создании и изменении таблицы, имеющей столбец распределения, теперь можно указать хеш-функцию, используемую для распределения данных между экземплярами сегментов.
- **Arenadata DB 6** удаляет опцию *RECHECK* из `ALTER OPERATOR FAMILY` и `CREATE OPERATOR CLASS DDL` (**PostgreSQL 8.4**). Теперь **ADB** на лету во время выполнения определяет, является ли оператор индекса "lossy".
- Таблицы operator-related системного каталога изменены для поддержки операторов семейств, совместимости и типов (сортировка или поиск).
- Записи таблиц системного каталога для функций HyperLogLog (HLL), агрегатов и типов переименованы с использованием префикса `gp_`. Переименование функций HLL предотвращает конфликты имен с внешними расширениями базы данных, использующими HLL. Любой пользовательский код, написанный с использованием встроенных функций HLL базы данных **ADB**, должен быть обновлен на имена с префиксом.
- "Устаревший оптимизатор" предыдущих выпусков **ADB** теперь упоминается как *Postgres planner* как в коде, так и в документации.
- Уровни изоляции транзакций в **Arenadata DB 6** изменены в соответствии с уровнями изоляции транзакций **PostgreSQL**, начиная с введения режима serializable snapshot isolation (SSI) в **PostgreSQL 9.1**. Новый режим SSI, который не реализован в **ADB**, обеспечивает истинную сериализуемость путем мониторинга одновременных транзакций и отката транзакций, которые могут привести к аномалии сериализации. Существующий режим изоляции моментальных снимков snapshot isolation (SI) гарантирует, что транзакции работают с одним согласованным снимком базы данных, но не гарантирует согласованного результата при выполнении набора одновременных транзакций в любой заданной последовательности.

Arenadata DB 6 теперь позволяет использовать ключевые слова *REPEATABLE READ* с SQL-выражениями, такими как *BEGIN* и *SET TRANSACTION*. Транзакция *SERIALIZABLE* в **PostgreSQL** версии *9.1* и выше использует новый режим SSI. В **ADB 6** транзакция *SERIALIZABLE* возвращается к *REPEATABLE READ*, используя режим SI. Далее в таблице показано соответствие стандарту SQL для каждого уровня изоляции транзакций в **Arenadata DB 6** и **PostgreSQL 9.1**.

Таблица3.1.: Соответствие уровня транзакций стандарту SQL

Уровень изоляции транзакции	Arenadata DB 6	PostgreSQL 9.1
READ UNCOMMITTED	READ COMMITTED	READ COMMITTED
READ COMMITTED	READ COMMITTED	READ COMMITTED
REPEATABLE READ	REPEATABLE READ (SI)	REPEATABLE READ (SI)
SERIALIZABLE	Возвращается к REPEATABLE READ (SI)	SERIALIZABLE (SSI)

- Команда `CREATE TABLESPACE` изменена:
  - Команде больше не требуется файловое пространство, создаваемое с помощью утилиты **gpfilespace**;
  - Условие *FILESPACE* удалено;
  - Условие *WITH* добавлено, чтобы позволить указывать расположение табличного пространства для конкретного экземпляра сегмента.
- Команда `ALTER SEQUENCE SQL` содержит новые условия *START [WITH] start* и *OWNER TO new_owner* (**PostgreSQL 8.4**). Условие *START* устанавливает начальное значение, которое будет использоваться

будущими командами `ALTER SEQUENCE RESTART`, но не изменяет текущее значение последовательности. Условие *OWNER TO* меняет владельца последовательности.

- Команда `ALTER TABLE SQL` содержит условие *SET WITH OIDS* для добавления системного столбца *oid* в таблицу (**PostgreSQL 8.4**). Важно обратить внимание, что использование *oids* с таблицами базы данных **ADB** настоятельно не рекомендуется.
- Команда `CREATE DATABASE SQL` содержит новые параметры *LC_ COLLATE* и *LC_ CTYPE* для указания порядка сортировки и классификации символов для новой базы данных.
- Команда `CREATE FUNCTION SQL` содержит новое ключевое слово *WINDOW*, которое указывает, что функция является не простой, а оконной (**PostgreSQL 8.4**).
- Указание имени индекса в команде `CREATE INDEX SQL` теперь необязательно. База данных **ADB** создает имя индекса по умолчанию из имени таблицы и индексированных столбцов.
- В команде `CREATE TABLE` парсер базы данных **ADB** позволяет размещать запятые между условиями *SUBPARTITION TEMPLATE* и *SUBPARTITION BY*, а также между последовательными условиями *SUBPARTITION BY*. Но эти недокументированные запятые являются устаревшими и генерируют предупреждение об устаревании.
- Для создания протокола теперь необходимы привилегии суперпользователя.
- Команда `CREATE TYPE SQL` содержит новое условие *LIKE=type*, которое копирует представление нового типа (*INTERNALLENGTH*, *PASSEDBYVALUE*, *ALIGNMENT* и *STORAGE*) из существующего (**PostgreSQL 8.4**).
- Команда `GRANT SQL` содержит новый синтаксис для предоставления прав на truncate, обертки сторонних данных и сторонние серверы (**PostgreSQL 8.4**).
- Команда `LOCK SQL` содержит опциональное ключевое слово *ONLY* (**PostgreSQL 8.4**). Если оно указано, таблица блокируется, но без блокировки наследуемых от нее таблиц.
- Использование выражения *LOCK table* вне транзакции вызывает ошибку. В более ранних выпусках **ADB** выражение выполняется, хотя оно полезно только при выполнении внутри транзакции.
- Команды `SELECT` и `VALUES SQL` поддерживают синтаксис *SQL 2008 OFFSET* и *FETCH* (**PostgreSQL 8.4**). Эти условия предоставляют альтернативный синтаксис для ограничения возвращаемых запросом результатов.
- Условие *FROM* может быть опущено в команде `SELECT`, но **ADB** больше не допускает такие запросы и ссылается на таблицы базы данных.
- Ключевые слова *ROWS* и *RANGE SQL* изменились с зарезервированных на незарезервированные и могут использоваться без кавычек в качестве имен таблиц или столбцов.
- В **Arenadata DB 6** запрос к внешней таблице с наследуемыми таблицами по умолчанию выполняется рекурсирсивно. Для того, чтобы ограничить запрос родительской таблицей, необходимо в запрос включить ключевое слово *ONLY*.
- Значение по умолчанию для параметра конфигурации сервера `optimizer_force_multistage_agg` изменилось с *true* на *false*. Теперь GPORCA по умолчанию выбирает между одноэтапным или двухэтапным планом агрегирования для скалярных агрегатов, содержащих оператор distinct, основываясь на стоимости.
- Команда `TRUNCATE SQL` имеет опциональное ключевое слово *ONLY* (**PostgreSQL 8.4**). Если оно указано, к таблице применяется операция truncate, не затрагивая наследованные ею таблицы.
- Утилита командной строки **createb** содержит новые опции `-l(--locale)`, `--lc-collate` и `--lc-ctype` для указания языкового стандарта и классификации символов для базы данных (**PostgreSQL 8.4**).
- Утилиты **pg_dump**, **pg_dumpall** и **pg_restore** содержат новую опцию `--role=rolename`, которая указывает утилите выполнять `SET ROLE rolename` после подключения к базе данных и перед запуском дампа или операции восстановления (**PostgreSQL 8.4**).

- Утилиты командной строки **pg_dump** и **pg_dumpall** содержат новую опцию `--lock-wait-timeout=timeout` (**PostgreSQL 8.4**). Если она задана, то в случае, когда утилита не может получить блокировку общей таблицы в течение указанного количества миллисекунд, дамп завершается ошибкой вместо бесконечного ожидания.
- Параметры командной строки `-d` и `-D` удалены из утилит **pg_dump** и **pg_dumpall**. Соответствующие им длинные версии `--inserts` и `--column-inserts` по-прежнему поддерживаются. Добавлена новая опция `--binary-upgrade` для in-place утилит обновления.
- Добавлена опция `-w (--no-password)` в утилиты **pg_dump**, **pg_dumpall** и **pg_restore**.
- Удалена опция `-D` из утилиты **gpexpand**. Схема расширения создается в базе данных **postgres**.
- Утилита **gpstate** содержит новую опцию `-x`, которая отображает детали текущего расширения системы. При этом `gpstate -s` и `gpstate` без указания параметров также сообщают, если выполняется расширение системы.
- Утилита **pg_restore** содержит новую опцию `-j (--number-of-jobs)`. Этот параметр позволяет сократить время восстановления большой базы данных за счет одновременного выполнения таких задач, как загрузка данных, создание индексов и создание ограничений.
- Утилита **vacuumdb** содержит новую опцию `-F (--freeze)` для фиксации информации о транзакциях строк.
- *ALTER DATABASE* включает условие `SET TABLESPACE` для изменения табличного пространства по умолчанию.
- *CREATE DATABASE* включает опции `COLLATE` и `CTYPE` для установки порядка сортировки и классификации символов новой базы данных.
- В схеме **gp_toolkit** представления `gp_workfile_*` изменились благодаря улучшениям рабочего файла **Arenadata DB 6**.
- Параметр конфигурации сервера **gp_workfile_compress_algorithm** изменен на **gp_workfile_compression**. При включенном сжатии рабочего файла **ADB** использует компрессию **Zstandard**.
- Функции совместимости с **Oracle** теперь доступны в **ADB** в виде расширения, основанного на проекте **PostgreSQL orafce** по ссылке https://github.com/orafce/orafce. Теперь для установки функций совместимости в базе данных вместо выполнения SQL-скрипта выполняется команда SQL `CREATE EXTENSION orafce`. Расширение *orafce* базы данных **Arenadata DB 6** основано на выпуске **orafce 3.7**.
- В качестве ключа распределения **ADB 6** поддерживает указание столбца таблицы типа данных *citext*.
- **Arenadata DB 6** предоставляет единый пакет *Client and Loader Tools*, который можно загрузить и установить в клиентской системе. Предыдущие выпуски **ADB** предоставляют отдельные пакеты клиента и загрузчика.
- **Arenadata DB 6** включает в себя модули *contrib* как для PostgreSQL-sourced, так и для Greenplum-sourced. Большинство из этих модулей теперь упакованы как расширения, которые регистрируются с помощью команды `CREATE EXTENSION name`.
- В режиме высокой доступности (High Availability, HA) базы данных **ADB** основной сегмент находится в согласованном состоянии со своим зеркалом с помощью синхронной потоковой репликации на основе журнала предзаписи (Write-Ahead Logging, WAL). Представление **gp_stat_replication** содержит статистику репликации при включенном зеркалировании мастера или сегмента. В предыдущих выпусках зеркалирование сегментов использовало физическую схему репликации файлов.
- В таблице **gp_segment_configuration** удален порт `replication_port`. При этом для отображения каталога данных экземпляра сегмента добавлен столбец `datadir`. Значения столбца `mode` теперь `s` (синхронизированы) или `n` (не синхронизированы). Для определения состояния синхронизации необходимо использовать представление **gp_stat_replication**.

Глава 4

Удаленные функции

В **Arenadata DB 6** больше не используется:

- Утилита **gpseginstall**. Необходимо установить RPM программного обеспечения **Arenadata DB** на каждом хосте сегмента.
- Утилита **gptransfer**. Необходимо использовать **gpcopy** для всех функций, которые предоставляла **gptransfer**.
- Системная таблица **gp_fault_strategy**. Теперь для определения включенного зеркалирования база данных **ADB** использует системную таблицу **gp_segment_configuration**.
- Утилиты управления **gpcrondump**, **gpdbrestore** и **gpmfr**. Теперь для резервного копирования и восстановления базы данных используется **gpbackup** и **gprestore**.
- **Veritas NetBackup**.
- Прямой ввод-вывод для обхода буферизации памяти в кэше файловой системы для резервного копирования.
- Протокол внешних таблиц **gphdfs external table protocol** для доступа к системе **Hadoop**. Теперь для доступа к **Hadoop** используется платформа расширения **Platform Extension Framework** (**PXF**).
- **SSLv3**.
- Следующие параметры конфигурации сервера:
  - gp_analyze_relative_error
  - gp_backup_directIO
  - gp_backup_directIO_read_chunk_mb
  - gp_connections_per_thread
  - gp_enable_sequential_window_plans
  - gp_idf_deduplicate
  - gp_snmp_community
  - gp_snmp_monitor_address
  - gp_snmp_use_inform_or_trap
  - gp_workfile_checksumming
- Недокументированная функция **gp_cancel_query()** и параметры конфигурации **gp_cancel_query_print_log** и **gp_cancel_query_delay_time**.

- Функция **string_agg(expression)**, объединяющая текстовые значения в строку. Функция **string_agg(expression, delimiter)** по-прежнему поддерживается.
- Больше не поддерживается возможность настройки системы базы данных для запуска триггеров **SNMP** (Simple Network Management Protocol) или отправки уведомлений по электронной почте системным администраторам при возникновении определенных событий в базе данных. Теперь для обнаружения и реагирования на происходящие в **ADB** события используются оповещения Command Center.
- Утилита **gpfilespace**. Команде `CREATE TABLESPACE` больше не требуется файловое пространство, созданное утилитой.
- В **Arenadata DB 6** больше не выполняется автоматическое преобразование текста из устаревшего формата временных меток `YYYYMMDDHH24MISS`. Формат не может быть однозначно проанализирован в предыдущих выпусках базы данных. Формат не поддерживается в **PostgreSQL 9.4**. Например, эта команда возвращает ошибку в базе данных **ADB 6** (в предыдущих версиях возвращается метка времени):

```
# select to_timestamp('20190905140000');
```

В версии **ADB 6** метка времени возвращается по такой команде:

```
# select to_timestamp('20190905140000','YYYYMMDDHH24MISS');
```

В **Arenadata DB 6** больше не поддерживаются следующие функции:

- Параметр конфигурации сервера **gp_ignore_error_table** устарел и будет удален в следующем мажорном выпуске.

При удалении этого параметра из **ADB 6** база данных возвращает ошибку, если команда `CREATE EXTERNAL TABLE` или `COPY` содержит условие из **ADB 4** `INTO ERROR TABLE`. Поэтому рекомендуется устанавливать значение данного параметра `true` во избежание ошибки базы данных при запуске приложений, выполняющих указанные команды.

- *Specifying* в качестве имени оператора в команде `CREATE OPERATOR`.
- API таблицы *external table C API*. Вместо данного API рекомендуется использовать новый – *Foreign Data Wrapper API*.

Глава 5

Ключевые отличия ADB от open-source аналога

ADB 6 имеет ряд ключевых отличий от open-source аналога:

- Установка продукта с помощью **ADCM**;
- Интеграция с бандлом мониторинга **ADCM**;
- Поддержка расширенного двустороннего обращения к источникам и приемникам данных через **JDBC**;
- Поддержка коннектора для Clickhouse (в enterprise версии).