ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

***«*САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**ПЕТРА ВЕЛИКОГО»**

**Институт компьютерных наук и технологий**

|  |
| --- |
|  |

**Отчет о прохождении производственной (научно-исследовательская работа) практики**

|  |
| --- |
| Ульянова Алексея Александровича |

*(Ф.И.О. обучающегося)*

|  |
| --- |
| 2 курс, 3540203/00101 |

*(номер курса обучения и учебной группы)*

|  |
| --- |
| 02.04.03 Математическое обеспечение и администрирование информационных систем |

*(направление подготовки (код и наименование))*

|  |
| --- |
| **Место прохождения практики:** ФГАОУ ВО «СПбПУ», ИКНиТ, ВШИИ, |

*(указывается наименование профильной организации или наименование структурного подразделения*

|  |
| --- |
| г. Санкт-Петербург, ул. Обручевых, д. 1, лит. В |

*ФГАОУ ВО «СПбПУ», фактический адрес)*

|  |  |
| --- | --- |
| **Сроки практики:** с02.09.2021 по 23.12.2021 | |
| **Руководитель практической подготовки от ФГАОУ ВО «СПбПУ»:** Белых Игорь Николаевич, к.ф.-м.н., доцент ВШИИ | | | |
| **(***Ф.И.О., уч. степень, должность)* | | | |
| **Консультант практической подготовки от ФГАОУ ВО «СПбПУ»:** Сабинин Олег Юрьевич, к.т.н., доцент ВШИИ | | | |
| **(***Ф.И.О., уч. степень, должность)* | | | |
| **Руководитель практической подготовки от профильной организации**: нет | | | |
| **Оценка:** |
| Руководитель практической подготовки  от ФГАОУ ВО «СПбПУ»: Белых И.Н. | | |
| Консультант практической подготовки  от ФГАОУ ВО «СПбПУ»: Сабинин О.Ю. | | |
| Руководитель практической подготовки  от профильной организации: | | |
| Обучающийся: Ульянов А.А. | | |
| Дата: |

Содержание

[Содержание 2](#_Toc95013450)

[ВВЕДЕНИЕ 3](#_Toc95013451)

[1. Описание принципа работы по oracle goldengate 6](#_Toc95013452)

[2. Подготовительный этап перед разработкой 7](#_Toc95013453)

[3. Описание разработки прототипа системы репликации данных в реальном времени при помощи ПО ORACLE GOLDENGATE 8](#_Toc95013454)

[3.1. Включение на БД-источнике журналирования и дополнительного логирования 8](#_Toc95013458)

[3.2. Создание пользователя для работы с Oracle GoldenGate 10](#_Toc95013459)

[3.3. Процесс настраивания непрерывной репликации данных 11](#_Toc95013460)

[3.4. Тестирование разработанного прототипа 16](#_Toc95013461)

[ЗАКЛЮЧЕНИЕ 18](#_Toc95013462)

[СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ 19](#_Toc95013463)

[СПИСОК ИСТОЧНИКОВ 20](#_Toc95013464)

[Приложение 1 21](#_Toc95013465)

[Приложение 2 21](#_Toc95013466)

[Приложение 3 22](#_Toc95013467)

[Приложение 4 22](#_Toc95013468)

[Приложение 5 23](#_Toc95013469)

[Приложение 6 23](#_Toc95013470)

ВВЕДЕНИЕ

Сложно оспорить тот факт, что мир управляется данными. И чаще всего эти данные хранятся в базах данных. Они предназначены для структурированного хранения и быстрого доступа к различным сведениям. Любая база данных должна иметь определенную модель работы, по которой будет выполняться обработка хранимой информации. Но зачастую, в больших компаниях используются сразу несколько баз данных, которые могут годами накапливать информацию разного вида. И в условиях 21-го века, где конкуренция обострена до предела, компании, которые могут извлечь из этих данных максимум полезных сведений, обладают ключевым преимуществом. Как раз для решения такого вида проблем бизнес-аналитики и создаются хранилища данных.

Хранилище данных – это система, в которой собираются данные из различных источников и их анализ используется для принятия наилучших управленческих решений. Структура хранилища данных зависит от текущей ситуации и потребностей конкретной организации. Чаще всего данные в хранилище поступают из OLTP-систем – это системы со специальным способом организации, при котором обработка транзакций происходит в реальном времени; чаще всего они используются для ввода, структурированного хранения и обработки информации в режиме онлайн. В то время как OLTP базы данных подвержены частым изменениям, хранилища используются исключительно для выборки данных и анализа, так как в них накапливаются исторические данные, которые могут поступать из различных источников (например, журналы приложений и приложения транзакций, транзакционные системы, реляционные базы данных и др.). Наиболее распространенные сферы применения – это онлайн обработка транзакций, онлайновая аналитическая обработка и прогнозная аналитика. При большой разрозненности источников данных хранилище можно разделять на витрины данных для грамотного распределения ресурсов при использовании определенных бизнес-функций. В качестве системы, обеспечивающей выгрузку данных из OLTP-источников в хранилище в реальном времени, будет использована Oracle GoldenGate,

Опираясь на анализ, проделанный в прошлой научной исследовательской работе, можно выделить следующую модель разработки системы сбора, преобразования и доставки данных в хранилище в реальном времени из OLTP-источников при помощи системы Oracle GoldenGate, которая состоит из следующих этапов:

* Проанализировать суть данных, содержащихся в OLTP-источниках, и выделить бизнес-требования, по которым будет проводиться их анализ;
* Проанализировать формат данных, который содержится в   
  OLTP-источниках, и определить какие требуются преобразования над ними перед переносом;
* Спроектировать хранилище данных в соответствии с выбранными   
  бизнес-требованиями и исходя из выявленных преобразований над данными;
* При помощи технологии Oracle GoldenGate настроить маршрутизацию данных из источников в хранилище в режиме онлайн.

В данной работе будет подробно разобран последний пункт из вышеперечисленного списка. Но перед тем, как переходить на работу с реальными данными, нужно более детально разобраться с используемом ПО Oracle GoldenGate и получить практический опыт в настройке маршрутизации данных между двумя базами данных.

Таким образом, цель данной научной исследовательской работы – разработка прототипа системы, которая реплицирует данные из одной БД в другую в режиме онлайн при помощи технологии Oracle GoldenGate, и написание отчета по проделанной работе. Для её достижения необходимо решить следующие задачи:

* Провести анализ принципа работы ПО Oracle GoldenGate;
* Создать две базы данных на разных серверах: target и source БД;
* Настроить маршрутизацию между ними при помощи ПО Oracle GoldenGate;
* Включить и проверить репликацию данных в режиме онлайн.

1. Описание ПРИНЦИПА РАБОТЫ ПО ORACLE GOLDENGATE

Oracle GoldenGate – это высокопроизводительное программное обеспечение, которое обеспечивает двустороннюю репликацию на основе журналов СУБД, а также использующееся для сбора данных из источников, их трансформации и доставки уже в структурированном виде в хранилище в реальном времени. Данная система может быть использована не только с СУБД Oracle, она интегрируема и с другими.

Ее главной задачей является обеспечение потока данных о событиях, происходящих с данными, из OLTP-источников в единое хранилище в реальном времени. Ключевая особенность - использование redo-log- или archive redo-log-файлов для сбора изменений и пересылка их по средствам передаточных trail-файлов на целевой сервер. Данный подход является весьма эффективным, так как используются исключительно журналы повторного выполнения; не происходит никаких лишних запросов и действий с источником. Помимо этого, появляется возможность получить все изменения данных в режиме онлайн и привязать их к конкретной дате и времени, когда они произошли в базе данных.

Теперь, описав как именно работает ПО Oracle GoldenGate, перейдем к описанию процесса разработки прототипа.

1. Подготовительный этап перед разработкой

Первым делом нужно сымитировать две стороны, участвующие в передаче данных. Для этого нужно использовать два отдельных сервера, с разными ip-адресами. Это можно достичь при использовании 2-х персональных компьютеров или при помощи использования виртуальной машины со специальной сетевой конфигурацией, при которой она считается в сети отдельной от хоста единицей. В моем случае был выбран для использования первый вариант. После этого нужно установить СУБД на обе машины. Как уже было сказано ранее, это не обязательно должны быть СУБД одной конфигурации, типа и производства. Однако, в связи со сразу перечнем факторов, связанных в первую очередь с наибольшим практическим опытом, было выбрано для прототипа использовать на обеих сторонах СУБД Oracle версии 19c. Во время установки, было принято решение создать подключаемую базу данных (PDB), поскольку в реальной ситуации вполне возможно, что к одному серверу будут подключены сразу несколько PDB и реплицировать данные нужно будет с каждой из них.

Таким образом, на каждой стороне было создано по своей PDB со следующими именами: unik – это БД-источник и orclpdb – целевая БД.

Далее можно было перейти к установке корректного соединения между этими БД при помощи ПО Oracle GoldenGate.

1. ОПисание разработки прототипа системы репликации данных в реальном временИ ПРи помощи  
   ПО ORACLE GOLDENGATE

Итак, первым делом нужно установить соответствующее программное обеспечения на оба компьютера. Линейка поддерживаемых СУБД и ОС крайне разнообразна, но в данном случае была скачена версия для Oracle и Windows x64 на обе машины. К моменту установки данной системы на компьютере уже должна быть развернута соответствующая БД и определена глобальная переменная ORACLE\_HOME. Именно это значение используется в качестве соответствующего пути к БД, с которой будет связан GoldenGate. Также в процессе установки можно выбрать расположение для переменной GG\_HOME, с которой будет работать система, порт для задач, связанных с передачей данных, и другие менее важные параметры.

После установки, для связи с агентом Oracle GoldenGate можно использовать утилиту командной строки ggsci.exe. Далее будет поэтапно расписаны все действия, которые необходимо осуществить для установки соединения между двумя базами данных.

3. 1. Включение на БД-источнике журналирования и дополнительного логирования

Исходя из принципа работы ПО Oracle GoldenGate, описанного ранее, ему необходимо, чтобы все изменения, связанные с данными, были отражены в журналах redolog. Для включения и выключения этой функции в БД Oracle есть специальный параметр log\_mode, который может принимать значение либо «ARCHIEVELOG», либо «NOARCHIVELOG». При этом для того, чтобы его изменить, БД должна находиться на этапе mount для изменения файлов контроля. Также стоит отметить, что данный параметр меняется на уровне корневой БД, а не на уровне PDB.

Однако журналы Oracle в первую очередь оптимизированы для быстрого восстановления данных, а не репликации. В стандартной конфигурации они не хранят информацию по всем атрибутам изменяющихся таблиц. Но есть специальный процесс SUPPLEMENTAL LOGGING, который отвечает за запись дополнительной информации в журнал во время операций изменения. У него, в свою очередь, есть разные типы, позволяющие выбрать либо логировать дополнительную информацию при изменении любого столбца таблички, либо определить набор столбцов, изменения которых нужно отслеживать, и разные уровни:

* Minimal - в этом режиме БД логирует дополнительный объем данных, который необходим для идентификации, группировки и соединения операций Redo, связанных с DML изменениями; не добавляет значительную нагрузку на БД;
* ALL - безусловно заставляет БД записывать в журналы состояние-до-изменения для всех столбцов в изменяемой строке (за исключением LOBs, LONGS, и ADTs);
* PRIMARY KEY - безусловно (даже если первичный ключ не меняется) заставляет базу записывать в журналы состояние-до-изменения для первичного ключа в изменяемой строке;
* UNIQUE - при условии, что изменяется столбец, входящий в уникальный или bitmap индекс, логируется состояние-до-изменения для всех столбцов, принадлежащие этому индексу;
* FOREIGN KEY — при условии, что изменяет столбец, входящий в FK, логируется состояние-до-изменения для всех столбцов FK.

В рамках тестирования на небольших данных мной был выбран уровень ALL.

Резюмируя все вышесказанное, был написан скрипт, состоящий из SQL-команд, для включения журналирования и дополнительного логирования информации для всех столбцов (Приложение 1). Его нужно выполнять только на стороне БД-источника.

Теперь, когда мы подключили необходимый функционал для полноценного и необходимого логирования в журналы, нужно изменить еще один параметр типа Boolean в БД, который указывает СУБД использовать или нет сервисы, поставляемые Oracle GolgenGate. Это параметр enable\_goldengate\_replication. По своей сути он точно также указывает включить дополнительное логировние в журналы, но без его установления в TRUE, на моменте репликации будет возникать ошибка. Для этого используем следующую команду (ее нужно также выполнять на уровне корневой БД источника):

* alter system set enable\_goldengate\_replication = true scope = both.
  1. Создание пользователя для работы с Oracle GoldenGate

Теперь для корректной работы с ПО Oracle GolgenGate нужно создать специального пользователя и выделить ему все необходимые привилегии. Данный этап нужно выполнить на обеих базах данных и это один из самых тонких моментов, по следующим причинам:

* пользователя необходимо создавать на уровне корневой БД, так как для репликации данных сервис Oracle GoldenGate обязательно должен подключаться к корневой БД, несмотря на то, что данные будут выбираться из конкретных PDB;
* глобальный пользователь обладает куда большими возможностями, поэтому привилегии ему нужно выделять вдвойне аккуратно;
* для использования GoldenGate пользователю необходимы некоторые специфические привилегии, о которых будет рассказано далее.

Одна из особенностей глобальных пользователей в системе Oracle, которые не относятся к группе системных, это то, что они должны начинаться с префиксом «c##». Таким образом, на обеих БД были созданы пользователь c##ggowner и табличное пространство goldengate для него (скрипт их создания представлен в приложении 2). Дальше начался процесс определения нужных привилегий для него.

Помимо стандартных привилегий на создание сессий, таблиц, выделение ресурсов и тд, этому пользователю необходимо выделить некоторые специфические. Полный список всех выделенных привилегий доступен в приложении 2, далее же будут разобраны наиболее неочевидные:

* в папке, куда был установлен goldengate, нужно запустить sqlplus под пользователем SYSDBA и скрипт role\_setup.sql. Это встроенный стандартный скрипт, который создает специальную роль GGS\_GGSUSER\_ROLE с минимальным набором привилегий, необходимых пользователю для работы с Goldengate, и включает переданного пользователя в эту роль;
* выполнить команду вида: exec dbms\_goldengate\_auth.grant\_admin\_privilege('c##ggowner', 'apply',  
  container=>'orclpdb'). Эта команда выделяет все необходимые привилегии для взаимодействия GoldenGate и PDB, которая указывается через параметр container (можно указать all для всех PDB, относящихся к корневой системе);
* выполнить в интересующей вас PDB команду вида: GRANT SELECT ANY DICTIONARY TO c##ggowner. Она также оказалась необходимой, иначе в дальнейшем будут ошибки при репликации.
  1. Процесс настраивания непрерывной репликации данных

Теперь все подготовительные действия сделаны, можно переходить к описанию команд, необходимых для настраивания непрерывной репликации через утилиту ggsci.

Для начала нужно на обеих базах данных запустить эту утилиту и выполнить команду «Edit params ./GLOBALS». Она создает глобальный файл конфига для работы агента Oracle GoldenGate, в котором можно указать глобальные параметры. В моем случае я остановился на указании схемы, которую агент будет использоваться для поддержания корректной репликации. В этом качестве будет использована схема пользователя c##ggowner, который был создан на этапе ранее.

Далее при помощи команды, представленной на рисунке 1, мы подключимся к корневой базе данных на стороне-источнике. Теперь агент GoldenGate связан непосредственно с БД и можно использовать весь функционал.

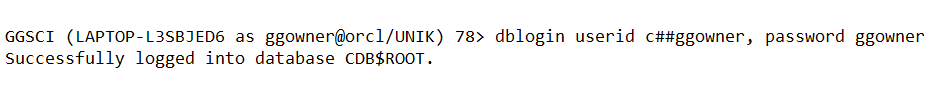


Рис. 1. Подключение к БД на стороне источнике

Сам алгоритм репликации можно разбить на 3 процесса: 2 процесса извлечения EXTRACT на стороне источника и один процесс репликации REPLICAT на стороне целевой БД.

Для начала разберемся с процессами EXTRACT:

* первый процесс отвечает за выборку данных из таблиц, которые выбраны для репликации. Для этого нужно создать файл с параметрами, который обязательно должен содержать логин и пароль пользователя для подключения и названия табличек. В моем случае еще был указан путь для хранения файлов трассировки и указана конкретная PDB база данных, в которой содержится интересующая меня табличка. Данный файл представлен в приложении 3. Стоит отметить, что можно использовать таблички из разных контейнеров, но с прямым их указанием перед названием схемы. При необходимости можно указать не конкретную табличку, а только схему, тогда все ее таблички будут включены в репликацию;
* второй процесс отвечает за настройку подключения к целевой базе данных и к ее конкретным табличкам. Для это процесса также нужно создать файл с параметрами (приложение 4), который содержит:
  + ip-адрес и порт, на котором работает агент GoldenGate на целевой стороне;
  + путь для файлов трассировки;
  + время между чекпоинтами процессов EXTRACT и REPLICAT;
  + таблички на стороне целевой БД, в которые будут реплицироваться данные, с прямым указанием нужной PDB и схемы;

Теперь разберем процесс REPLICAT, который устанавливается для целевой базы данных. Для него также создается файл с параметрами (приложение 5), в котором указывается:

* логин и пароль для подключения к целевой базе данных, причем в случае использования PDB, нужно указать именно ее для подключения;
* соотнести таблички для репликации: через MAP указать табличку на стороне источнике с указанием PDB и схемы, через TARGET указать табличку на стороне целевой БД с указанием PDB и схемы.

Создав нужные файлы для процессов, осталось их зарегистрировать и добавить на исполнение для агента Oracle GoldenGate. Команды, необходимые для этого на стороне-источнике, приведены в приложении 6, на целевой стороне достаточно выполнить команду вида: add replicat repora, EXTTRAIL ./dirdat/rt, checkpointtable с##ggowner.checkpointtable.

По причине использования на источнике PDB и ссылок на нее в первом EXTRACT файле, нужно произвести еще одну команду (рис. 2) по их связыванию между собой. Иначе агент GoldenGate при подключении к корневой БД не будет видеть PDB, к которой идет обращение, и будет возникать ошибка во время репликации.

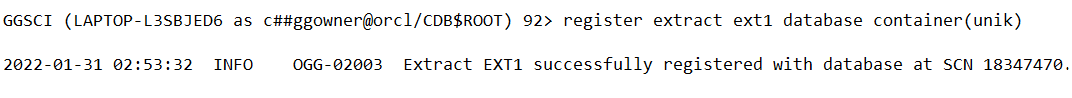


Рис. 2. Команда на регистрацию базы PDB из-под корневой для отслеживания агента GoldenGate

Теперь еще осталось напрямую указать агенту отслеживание транзакций для реплицируемых табличек и можно запускать созданные сервисы. Это делается на стороне источнике при подключении к коренвой БД с явным указанием PDB, таблички или схемы. В нашем случае используется репликация только одной таблички, поэтому использовался первый вариант. Данная команда приведена на рисунке 3.

Изображение выглядит как текст

Автоматически созданное описание

Рис. 3. Добавление отслеживания транзакций к табличке

Теперь все готово для запуска трех созданных процессов. На стороне-источнике используем подключение к корневой системе и команду start extract; на целевой – подключение к PDB и команду start replicat. Текущее состояние процессов можно вывести при помощи команды info all. Данные команды представлены на рисунках 4 и 5. Стоит отметить, что на этапе отладки было выявлено не мало ошибок при запусках той или иной службы. Для вывода в консоль трассировочных файлов использовалась команда view report.

Изображение выглядит как стол

Автоматически созданное описание

Рис. 4. Запуск на стороне источника

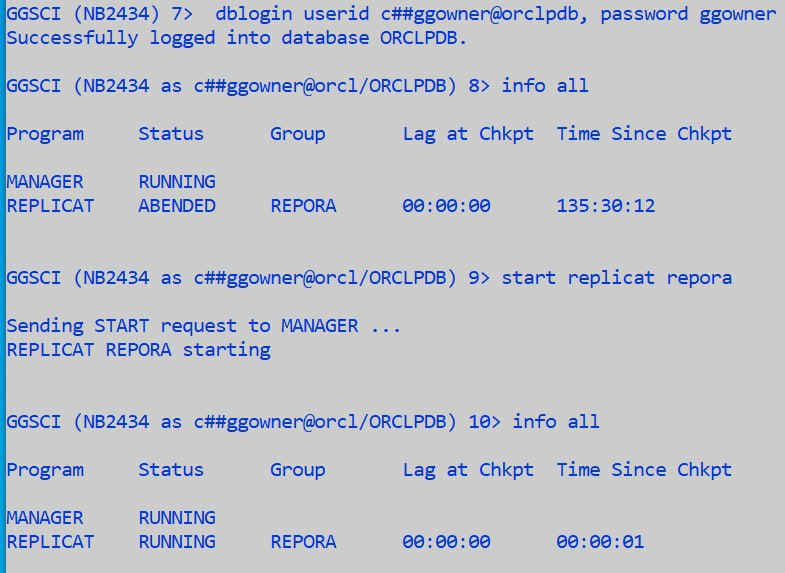


Рис. 5. Запуск на стороне целевой БД

* 1. Тестирование разработанного прототипа

Теперь, когда все сервисы запущены и при их запуске не было выведено никаких ошибок, можно перейти к тестированию на данных таблички, указанной в созданных файлах процессов EXTRACT REPLICAT. В нашем случае это табличка STUDENT2 схемы GGTEST на обеих сторонах. Выполним скрипт по вставке данных на стороне источника (приложение 7), закоммитим наши изменения таблички и проверим, что данные реплицировались. Промежуточно, можно также отследить отправлялись ли изменения через команду stats утилиты ggsci.exe. Пример этой команду продемонстрирован на рис. 6, по статистике видно, что были вставлены 8 записей, что соответствует выполненному скрипту.

Изображение выглядит как текст

Автоматически созданное описание

Рис. 6. Пример вывода команды stats

Осталось лишь визуально убедиться в том, что все строки вставились корректно. Это сравнение продемонстрировано на рис. 7.

Изображение выглядит как текст

Автоматически созданное описание Изображение выглядит как текст

Автоматически созданное описание

Рис. 7. Сравнение табличек в разных БД: слева – источник, справа - целевая

ЗАКЛЮЧЕНИЕ

Данная работа была в первую очередь направлена на изучение системы Oracle GoldenGate и получение практического опыта во взаимодействии с ней. Сперва был рассмотрен принцип работы данного ПО и разобраны его особенности. После этого был подробно описан ход разработки прототипа по репликации данных между двумя удаленными серверами. В ходе разработки был получен колоссальный опыт по установлению правильной конфигурации для непрерывной передачи данных и устранению ошибок, которые возникали по той или иной причине. Финальным этапом было проведение тестирования настроенной репликации, которое завершилось успешно.

Таким образом, описанную в прошлой научно-исследовательской работе модель системы по репликации данных в реальном времени, удалось реализовать на практике по средствам разработанного прототипа для тестовых данных. В дальнейшем при разработке полноценной системы, заявленной в дипломной работе, именно эти знания и практический опыт, полученные в результате проделанной работы, будут использованы в качестве точки отсчета.

СПИСОК сокращений и условных обозначений

БД – база данных.

ПО – программное обеспечение.

СУБД – системы управления базами данных.

PDB – PLUGGABLE DATABASE.

СПИСОК ИСТОЧНИКОВ

1. Золотые ворота: как реплицировать терабайт в час, или Опыт использования CDC на Oracle GoldenGate в ВТБ: [Электронный ресурс] // habr.com

URL: https://habr.com/ru/company/vtb/blog/479080/

(Дата обращения: 31.01.2022).

1. Опыт использования решения Oracle GoldenGate:

[Электронный ресурс] // jetinfo.ru

URL: https://www.jetinfo.ru/opyt-ispolzovaniya-resheniya-oracle-goldengate-dlya-business-intelligence/

(Дата обращения: 31.01.2022).

1. [GoldenGate Most likely errors](https://oralink.wordpress.com/2020/02/20/goldengate-most-likely-errors/) [Электронный ресурс] // oralink.wordpress.com

URL: https://oralink.wordpress.com/tag/ora-01031/

(Дата обращения: 31.01.2022).

1. Oracle GoldenGate: [Электронный ресурс] // fors.ru

URL: https://www.fors.ru/business-solutions/analytical-systems-and-data-warehouse/oracle-goldengate/

(Дата обращения: 31.01.2022).

1. REPLICATING PDBS WITH GOLDENGATE 12C [Электронный ресурс]   
   // oracledb101.wordpress.com

URL: https://oracledb101.wordpress.com/2014/03/26/replicating-pdbs-with-goldengate-12c/

(Дата обращения: 31.01.2022).

1. Setting up Oracle GoldenGate Replication: A Comprehensive Guide [Электронный ресурс] // hevodata.com

URL: https://hevodata.com/learn/oracle-goldengate-replication/

(Дата обращения: 31.01.2022).

# **Приложение 1**

**Скрипт для включения режима ARCHIVELOG и дополнительного логирования**

shutdown immediate;

startup nomount;

alter database mount;

alter database archivelog;

alter database add supplemental log data(all) columns;

--alter database add supplemental log data;

alter database force logging;

alter system switch logfile;

# **Приложение 2**

**Скрипт создания пользователя для работы с Oracle GoldenGate**

Create user c##ggowner identified by ggowner;

Grant resource, dba, connect to c##ggowner;

Create tablespace goldengate

datafile 'goldengate.dbf'

size 100m

autoextend on;

alter user c##ggowner default tablespace goldengate;

Grant connect, resource to c##ggowner;

Grant alter session to c##ggowner;

Grant create table to c##ggowner;

Grant flashback any table to c##ggowner;

Grant select any dictionary to c##ggowner;

Grant select any table to c##ggowner;

Grant execute on dbms\_flashback to c##ggowner;

Grant execute on utl\_file to c##ggowner;

exec dbms\_goldengate\_auth.grant\_admin\_privilege  
('c##ggowner','capture',container=>'all');

-- в PDB

GRANT SELECT ANY DICTIONARY to c##ggowner;

# **Приложение 3**

**Файл c параметрами процесса EXTRACT-EXTRACT**

EXTRACT ext1

USERID c##ggowner, PASSWORD ggowner

EXTTRAIL ./dirdat/ex1

SOURCECATALOG unik

TABLE ggtest.student;

# **Приложение 4**

**Файл c параметрами процесса EXTRACT-PUMP**

EXTRACT pumpora

PASSTHRU

RMTHOST \*\*\*.\*\*\*.\*.\*\*\*, MGRPORT 7809

RMTTRAIL ./dirdat/RT1

CHECKPOINTSECS 1

TABLE unik.ggtest.\*;

# **Приложение 5**

**Файл c параметрами процесса REPLICAT**

REPLICAT repora

USERID c##ggowner@orclpdb, PASSWORD ggowner

ASSUMETARGETDEFS

MAP unik.ggtest.\*, TARGET orclpdb.ggtest.\*;

# **Приложение 6**

**Команды для добавления процессов extract на исполнение агентом   
Oracle GoldenGate**

ADD EXTRACT ext1, INTEGRATED TRANLOG, BEGIN now

ADD EXTTRAIL ./dirdat/e1, EXTRACT ext1

ADD EXTRACT pumpora, EXTTRAILSOURCE ./dirdat/e1

Add RMTTRAIL ./dirdat/r1, EXTRACT pumpora