|  |  |
| --- | --- |
|  | УТВЕРЖДАЮ  Заместитель главного инженера –  начальник технического управления  ОАО «Сургутнефтегаз»  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.Ф.Зенов  «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2010 г. |

**Проектное решение**

**по обмену данными ИС «ОКО ЦИТС» с информационными системами**

7.6500.ТП.027.КО.01.01

ПРОЕКТНОЕ РЕШЕНИЕ

на листах

2010

РАЗРАБОТАЛИ

|  |  |  |  |
| --- | --- | --- | --- |
| Должность | Ф.И.О. | Подпись | Дата |
| Ведущий инженер-программист  ОАТП ПУ «СургутАСУнефть» | И.Н.Балашов |  |  |

СОГЛАСОВАНО

|  |  |  |  |
| --- | --- | --- | --- |
| Должность | Ф.И.О. | Подпись | Дата |
| Заместитель начальника технического управления  ОАО «Сургутнефтегаз» | Э.П.Егоров |  |  |
| Начальник отдела внедрения СВТ, АСУ и ТК технического управления  ОАО «Сургутнефтегаз» | С.А.Снесарь |  |  |
| Заместитель начальника  ПУ «СургутАСУнефть» | С.Я.Коровин |  |  |
| Заместитель начальника  ПУ «СургутАСУнефть» | В.В.Кузнецов |  |  |
| Начальник комплекса КАСУ ПП  ПУ «СургутАСУнефть» | В.Ю.Назаров |  |  |
| Начальник отдела АТП  ПУ «СургутАСУнефть» | С.Н.Корнеев |  |  |

НОРМОКОНТРОЛЬ

|  |  |  |  |
| --- | --- | --- | --- |
| Должность | Ф.И.О. | Подпись | Дата |
| Инженер-технолог 3 категории  ПУ «СургутАСУнефть» | С.А.Евмененко |  |  |

СОДЕРЖАНИЕ

[1 ВВЕДЕНИЕ 4](#_Toc275787582)

[2 ОСНОВАНИЕ 5](#_Toc275787583)

[3 НАЗНАЧЕНИЕ 6](#_Toc275787584)

[4 ОБМЕН ДАННЫМИ 7](#_Toc275787585)

[4.1 Эффективность 7](#_Toc275787590)

[4.2 Механизм взаимодействия 7](#_Toc275787591)

[4.2.1 Выбор схемы 8](#_Toc275787592)

[4.2.2 Регистрация задания 8](#_Toc275787593)

[4.2.3 Поддержка списка реплицируемых параметров 8](#_Toc275787594)

[4.2.4 Получение данных 9](#_Toc275787595)

[4.2.5 Фиксация дат 10](#_Toc275787596)

[4.2.6 Разовые операции 11](#_Toc275787597)

[4.2.7 Циклический режим 11](#_Toc275787598)

[4.2.8 Замечания 11](#_Toc275787599)

[5 ЗАКЛЮЧЕНИЕ 12](#_Toc275787600)

[ПРИЛОЖЕНИЕ 13](#_Toc275787601)

[СПИСОК СОКРАЩЕНИЙ 14](#_Toc275787602)

# 

# ВВЕДЕНИЕ

Данное проектное решение описывает механизм репликации между различными базами данных. Проектное решение призвано ускорить репликацию данных, снизить нагрузку на БД «ОКО ЦИТС», а также обеспечить целостность репликаций данных из БД «ОКО ЦИТС» в другие информационные системы.

# ОСНОВАНИЕ

Основанием для разработки является календарный план по реализации «Проектного решения по обмену данными между НПК «Альфа» и ИС «ОКО ЦИТС», утвержденный 04.08.2010 начальником ПУ «СургутАСУнефть» С.В.Кононовым.

# НАЗНАЧЕНИЕ

Данное проектное решение предназначено для осуществления быстрой, стабильной и целостной репликации данных, удовлетворяющих заданным критериям отбора, между двумя базами данных.

# ОБМЕН ДАННЫМИ



Как правило, для производственной необходимости часто требуется информация из нескольких различающихся БД. По разным причинам (медленные каналы связи, длительное время обслуживания, сложность написания и сопровождения распределенного запроса и т.д.) не всегда удается оперативно получить нужную информацию из удаленных БД. Поэтому применяются различные механизмы обмена данными, обеспечивающие их предварительный запрос из удаленной БД, агрегирование и сохранение в локальной БД.

## Эффективность

Самым распространенным вариантом обмена данными между двумя различными БД является использование SQL-запросов. В развивающихся (часто изменяемых) системах такие запросы зачастую являются, либо могут стать, неэффективными в силу следующих причин:

* при написании не учитывалась специфика удаленной БД;
* в удаленной БД произведены какие-либо структурные изменения;
* произошли изменения в природе поступающих данных;
* человеческий фактор при написании запроса.

Более эффективным представляется метод, когда удаленная БД присылает только запрос на получение обновлений по некоему предопределенному набору параметров, после чего запрашиваемая БД локально готовит необходимые данные и предоставляет их уже в оптимальном для передачи и обработки виде.

При таком подходе появляется возможность применять весь мощный инструментарий используемой СУБД.

## Механизм взаимодействия

При проектировании шлюза один из основных упоров сделан на обеспечение максимальной производительности и необходимость разделения клиентской и серверной нагрузок с тем, чтобы обеспечить защиту серверной части от неэффективных клиентских обращений.

В связи с этим спроектирован следующий алгоритм взаимодействия.

### Выбор схемы

Предварительно должна быть создана пользовательская схема, через которую будет происходить обмен данными. Наименование схемы должно соответствовать, например, аббревиатуре программного комплекса (ESRF, GIS, ISAT, ZSOUR и т.п.). Этим будет исключено наложение одноименных задач по различным направлениям.

### Регистрация задания

После того как пользовательская схема определена, необходимо однозначно идентифицировать решаемую задачу посредством присвоения ей псевдонима (Task\_Name). Для создания последнего нужно *единожды* его зарегистрировать путем вызова процедуры OKO\_Data.Register\_Task(Task\_Name).

Вызов данной процедуры нужно производить только из той схемы, для которой регистрируется задача, поскольку последняя неявно привязывается к имени схемы.

В дальнейшем вся работа должна осуществляться именно под этим псевдонимом. С ним же будет связан перечень реплицируемых параметров. Псевдоним должен иметь длину не более 30 символов, быть уникальным в рамках схемы, а также быть осмысленным, т.е. отражать суть задачи.

Допускается неоднократная регистрация задания с одинаковым псевдонимом, однако при этом список параметров будет полностью очищаться (удобно использовать на этапе отладки).

### Поддержка списка реплицируемых параметров

Чтобы система-источник обеспечивала максимальную производительность передачи данных при минимальном сетевом трафике, необходимо, чтобы она имела информацию о реплицируемых параметрах.

Для этого, зная псевдоним (Task\_Name) задания, со стороны клиентского приложения, необходимо произвести заполнение таблицы запрашиваемых параметров ().

Формат таблицы запрашиваемых параметров (Par\_Request)

Таблица 1

| №п  п | Название поля | Значение по умолчанию | Описание |
| --- | --- | --- | --- |
| 1 | Task\_Name | Обязательное | Название задачи (псевдоним) |
| 2 | Obj\_Id | Обязательное | Идентификатор ОКО объекта |
| 3 | Par\_Id | Обязательное | Идентификатор ОКО параметра |
| 4 | DT\_Beg | От нуля часов текущих суток | Начальная дата интервала |
| 5 | DT\_End | Неограничена | Конечная дата интервала |
| 6 | Outbound | 0 – отключено | Признак поиска одного дополнительного значения с датой, предшествующей указанному началу интервала (DT\_Beg) |

Шлюз организован таким образом, что дает существенный выигрыш только в случае использования редко изменяемого набора параметров. Для этого нужно заполнить таблицу Par\_Request только один раз, а в дальнейшем лишь добавлять параметры.

Чтобы не загромождать клиентские приложения, а также ввиду того, что в них не всегда присутствует процедура удаления ставших уже ненужными параметров, в данном шлюзе реализован совмещенный механизм, позволяющий выполнять одновременно и добавление новых параметров, и удаление устаревших.

Для этого, при каких-либо изменениях, список всегда должен обновляться полностью. Тогда при работе шлюза, в случае обнаружения новых параметров, они будут автоматически добавлены к хранимому перечню. Причем данные по новым параметрам по умолчанию начнут возвращаться с календарного начала текущих суток (если явно не задан параметр DT\_Beg). Если же какой-либо из ранее реплицируемых параметров отсутствует в обновленном списке, то он будет помечен как скрытый и данные по нему перестанут передаваться через шлюз. Если скрытый параметр позже вновь появится в списке, то он будет восстановлен с той же датой, которую имел на момент сокрытия.

В процессе заполнения таблицы Par\_Request производится проверка существования затребованных параметров в БД ОКО. Параметры, для которых сопоставление пройдет успешно, будут зарегистрированы, остальные будут помещены в таблицу Par\_Trash, где будут доступны до ближайшего из следующих событий:

* очередное заполнение таблицы запрашиваемых параметров;
* операция получения данных (п.).

После любого изменения перечня запрашиваемых параметров необходимо выполнить фиксацию (commit) транзакции. В противном случае список не будет обработан.

В случае радикального изменения списка реплицируемых параметров, а также на этапе тестирования есть возможность выполнять очистку перечня путем перерегистрации задания (п.4.2.2) В этом случае будут потеряны все ранее зафиксированные репликационные даты.

### Получение данных

Ввиду особенностей СУБД Oracle информацию можно получать различными способами, которые зависят от способа соединения с БД.

Прямое соединение

Одной операцией одновременно производится и подготовка данных, и их возврат. Запрос должен выглядеть следующим образом:

select Obj\_Id, Par\_Id, DT, Quality, …

from Table(OKO\_Data.Response2Pipe(Task\_Name))

Соединение через dblink

Первоначально необходимо произвести выборку измененных данных посредством вызова процедуры OKO\_Data.Response2Temp(Task\_Name).

В процессе её выполнения производится сканирование исторических данных, начиная со времени последнего обращения (а точнее фиксации транзакции[[1]](#footnote-2)) клиентского приложения и до текущего момента (если границы интервала не были заданы явно); выявляются все произошедшие обновления, которые вносятся в специальную временную таблицу.

После того, как данные подготовлены, можно производить их считывание из таблицы возвращаемых данных Par\_Response.

Если предполагается производить соединение данных из нескольких различных таблиц, то в целях оптимизации межбазового взаимодействия и избегания излишнего сетевого трафика рекомендуется первоначально выбрать из таблицы Par\_Response все записи, удовлетворяющие условиям задачи, в промежуточную таблицу, а затем продолжить работу с ними уже локально.

Формат возвращаемых данных для обоих способов соединения с БД представлен в таблице 2.

Формат таблицы возвращаемых данных (Par\_Response)

Таблица 2

|  |  |  |
| --- | --- | --- |
| №п/п | Название поля | Описание |
| 1 | Obj\_Id | Идентификатор ОКО объекта |
| 2 | Par\_Id | Идентификатор ОКО параметра |
| 3 | DT | Дата измерения |
| 4 | Quality | Качество данных |
| 5 | Value\_Num | Значение типа «число» |
| 6 | Value\_DT | Значение типа «дата» |
| 7 | Value\_Char | Значение типа «строка» |

Если в таблице ничего не окажется или запрос ничего не вернет, значит, данные по запрашиваемым параметрам и удовлетворяющие заданным условиям отсутствуют.

### Фиксация дат

По окончании выбора данных (п.) нужно произвести фиксацию последних переданных дат по каждому из реплицируемых параметров путем вызова процедуры OKO\_Data.Fix\_Max\_DT(Task\_Name).

Зафиксированные таким образом даты будут использованы при последующих вызовах для того, чтобы возвращать только новые (поступившие после предыдущей фиксации) данные.

При вызове данной процедуры производится неявное подтверждение (commit) транзакции.

### Разовые операции

Есть возможность по любым из уже реплицируемых параметров единовременно получать данные за произвольные интервалы времени. В этом случае нужно заполнить таблицу 1 (п.4.2.3) с одновременным указанием хотя бы одной из границ интервала дат DT\_Beg и/или DT\_End.

Предоставив данные за указанный интервал (п.4.2.4), шлюз автоматически вернется в состояние передачи только обновлений.

Если при запросе была указана конечная дата (DT\_End) интервала и после получения данных выполнена фиксация дат (п.4.2.5), то последующие обновления по каждому из запрошенных параметров начнут передаваться с даты наиболее приближенной, но не большей DT\_End.

В связи с этим, при разовых операциях, фиксацию нужно выполнять очень внимательно и только в случае необходимости.

### Циклический режим

Если список запрашиваемых параметров не изменяется и по условию задачи требуется получать по ним только изменения, то достаточно повторять только действия, описанные только в пп.4.2.4–4.2.5.

Пример алгоритма для наиболее оптимального режима приведен в [приложении](#_ПРИЛОЖЕНИЕ).

### Замечания

Предложенный механизм по обмену не выполняет никаких расчетных задач, но лишь выбирает и предоставляет необходимые данные по прямым параметрам.

Также он не обеспечивает учет перехода на зимнее и летнее время. Поэтому работа с датами производится с тем временным смещением, с каким они хранятся в БД. Как правило, это время по Гринвичу.

# ЗАКЛЮЧЕНИЕ

Предложенный механизм обмена данными позволяет облегчить и ускорить обмен для некоторых задач, но не позволяет полностью подменить механизм SQL-запросов.

Он дает значительные преимущества только при периодических запросах между различными БД на получение обновлений по большому и достаточно статичному количеству параметров. В этом случае поведение шлюза хорошо предсказуемо, т.к. практически исключены ошибки разработчиков удаленной БД при написании запросов. К тому же он легко перенастраивается, а работа может подробно журналироваться. Его использование позволяет производить мониторинг и балансировку нагрузки, а также для некоторых задач снизить сетевой трафик.

Механизм же SQL-запросов остается максимально универсальным и гибким средством.

Рекомендуется применение данного проектного решения для случаев, когда БД «ОКО ЦИТС» выступает в роли периодического источника исторических данных по редко изменяемому набору параметров. В частности, для задач выборочной репликации данных из БД «ОКО ЦИТС» в БД НПК «Альфа».

# ПРИЛОЖЕНИЕ

Типовой алгоритм репликации данных (циклический режим)

Регистрация задания

Заполнение таблицы реплицируемых параметров

Получение данных

Фиксация дат

Локальная обработка данных

Успешно?

Зафиксировать даты?

да

да

Откат транзакции

нет

Устранение ошибок

нет

Успешно?

нет

да

# СПИСОК СОКРАЩЕНИЙ

|  |  |  |
| --- | --- | --- |
| БД | – | база данных |
| ИС | – | информационная система |
| НПК | – | нефтепромысловый комплекс |
| ОКО | – | оперативный контроль объектов |
| СУБД | – | система управления базой данных |
| ЦИТС | – | центральная инженерно-технологическая служба |

1. Транзакция – группа последовательных операций, которая представляет собой логическую единицу работы с данными [↑](#footnote-ref-2)