Программа краткого курса

## Контроль и повышение производительности в решениях аналитики и отчётности в условиях высоких нагрузок

Редакция 2021-08-24

Легенда:

Вы устроились работать разработчиком программного обеспечения. Вашрабочийстек: **C#, MS SQL Server, Net Core, Net framework**.

Вам передают на обслуживание старую систему (**legacy**) для подготовки специализированной отчетности. Система (**legacy**) реализована на следующем стеке**: MSSQLServer, ASP.net, Netframework 4.7.**

Данные отчеты направляются по защищенному каналу разным потребителям: ЦБ РФ, Министерство финансов и на анализ независимым аудиторам. Для каждого потребителясистема формирует один и те же данные, но в разных форматах. Каждый выгруженный отчет фиксируется статусом. Если отчет не принят, то формируется повторный отчет с обновленными данными. Если отчет принят условно, то формируется дополнение к ранее отправленному отчету, в котором хранятся данные, которые только дополняют ранее отправленные данные.

Перед Вами непростая задача. Текущее (**legacy**)приложение ввиду того, что создавалось на «ходу» имеет много недостатков, но основной недостаток – это производительность. Данные долго готовятся. Проверяются, а затем выгружаются. В случае, если коллектив не успел в срок, организация автоматически получает штраф.

Ваша задача: найти и устранить все «слабые» места в приложении. При этом, не возможно провести полную замену приложения ввиду большой и постоянной нагрузки, высоких требований и жестких регламентов.

|  |
| --- |
| **Оглавление**   1. Анализ текущего проекта и замеры производительности. 2. Проведение оптимизации базы данных без изменения структуры. 3. Проведение оптимизации приложения с использованием библиотеки **Dapper**. 4. Проведение оптимизации приложения с использованием многопоточного программирования. 5. Итоги. |

Этап 1: Анализ текущего проекта.

Проект представляет с собой высоконагруженную базу данных, которая разделена на три части: учет фактов, учет обработанных данных, учет выгруженной информации. Учет фактов состоит из таблицы **tblTransactionFacts** в которой содержатся все исходные банковские транзакции, которые подлежат контролю. Учет обработанных данных состоит из нескольких таблиц:

|  |  |
| --- | --- |
| refAccounts | Счета клиентов |
| refContracts | Контракты клиентов |
| refCustomers | Клиенты |
| tblTransactions | Подготовленные транзакции |
| refReportPeriods | Справочник периодов |

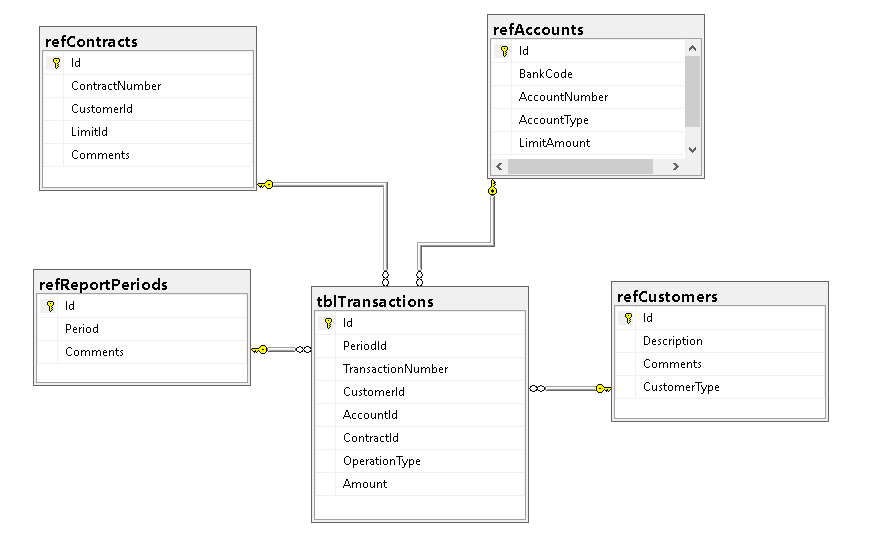


Рис. 1. Схема связей таблиц для учета обработанных данных

Учет выгруженной информации состоит из следующих таблиц:

|  |  |
| --- | --- |
| refReportPeriods | Таблица с периодами |
| repCustomerHistory | Таблица с информацией о выгрузке в разрезе каждого клиента. Для аудита |
| refRecipients | Справочник получателей информации |
| repCustomerTotals | Итоги выгрузки в разрезе получателя и клиента |
| refCustomers | Клиенты |

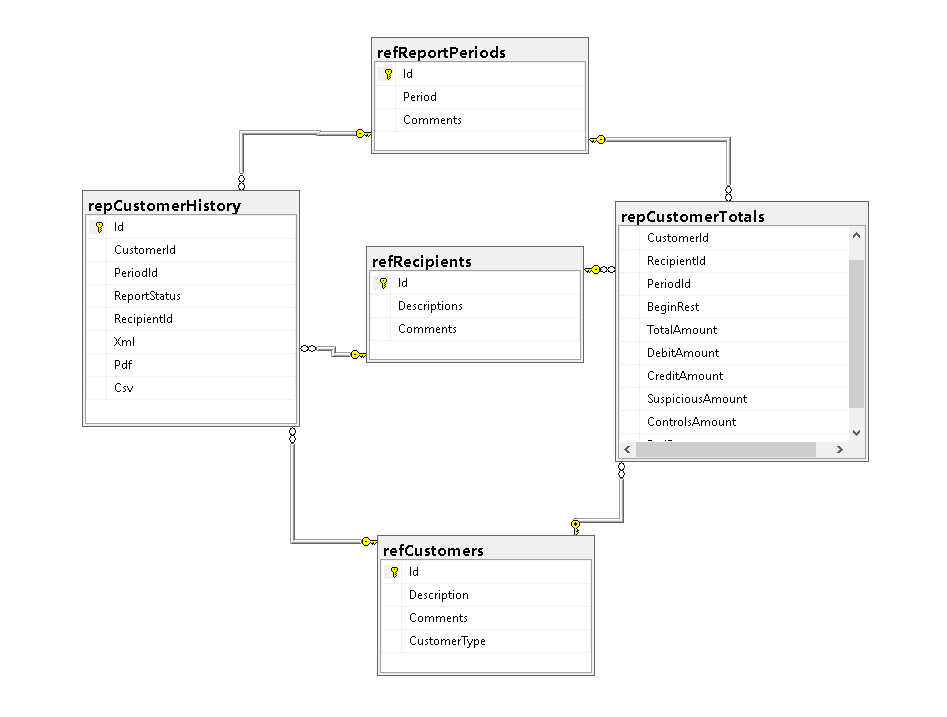


Рис. 2 Схема связей таблиц для учета выгруженной информации.

Исходные метрики:

|  |  |
| --- | --- |
| Количество записей в исходной таблице фактов | 1 00000 |
| Разброс по периодам | 2018-08-19 00:00:00.000 2021-05-15 00:00:00.000 |
| Объем базы данных | ~15 ГГб |
| Выборка и обработка данных | |
| Получить список периодов | 2,3 сек |
| Получить список записей для обработки за 1 месяц | 4 сек |
| Подготовить данных за 10 дней |  |

Этап 2. Оптимизация базы данных без изменения структуры

В ходе анализа установлены несколько ключевых запросов. Данные операции выполняются часто и приводят к большим затратам ресурсов сервера. Пример алгоритма подготовки данных.

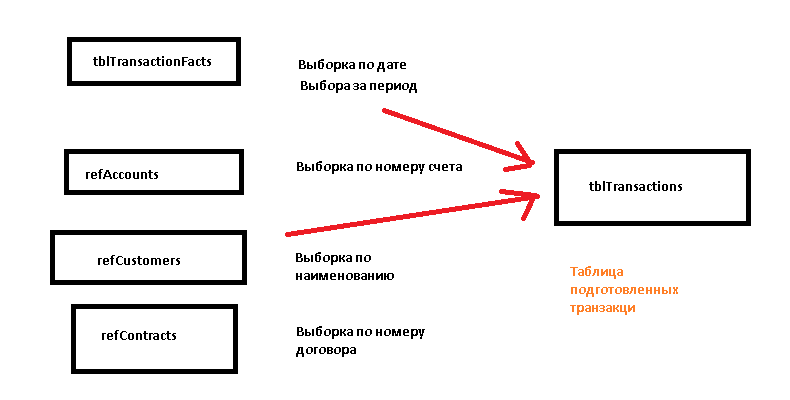


Рис. 3. Схема работы алгоритма для подготовки данных

Описание:

* Определить блок новых данных

Для каждой записи таблицы **tblTransactionFacts** формируется 4 операции:

* подобрать (добавить) счет
* подобрать (добавить) клиента,
* подобрать (добавить) контракт

Для определения ключевых запросов сформируем набор модульных тестов с определенными критериям и произведем запуск тестов. Далее, проведем анализ наиболее длительных тестов.

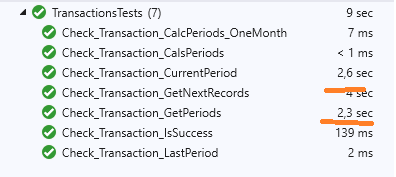


Рис. 4. Изображение замера работы модульных тестов.