Автоматизированная система сбора, хранения и анализа статистических данных.

Предложение Синкретиса.

Содержание.

1. Предыдущие проекты компании.......................................................................................3

1.1. Ассимметричная платформа обработки данных............................................................4

1.2 Проект Almanac..................................................................................................................5

2. Описание предложения......................................................................................................6

2.1 Общая структура................................................................................................................7

2.2 Структура конвертера.......................................................................................................8

2.3 Структура ETL....................................................................................................................10

2.4 Структура DWH.................................................................................................................12

2.5 Структура BI......................................................................................................................13

3. Вопросы к требованиям....................................................................................................15

1. Предыдущие проекты компании.

У компании Синкретис большой опыт работы с большими данными. В рамках заказной разработки для Swiss Re делался проект под названием **Ассимметричная платформа обработки данных**. Этот проект делался на платформе StarGate/Palantir Foundry. Фактически в нём имплементировались все заявленные требования, но используя возможности платформы Palantir Foundry. К сожалению, этот проект является собственностью Swiss Re, поэтому компания Синкретис не может показать свои достижения.

Кроме того, у компании Синкретис есть проект Almanac. Это версионированное хранилище данных различных типов. Этот проект нацелен на Российский рынок, и компания Синкретис готова адаптировать его для нужды РНПК.

1.1. Ассимметричная платформа обработки данных.

Страховые компании знают своих клиентов, лучше чем крупная перестраховочная компания. Идея проекта состоит в том, чтобы устранить эту ассимметрию в информации за счёт изучения данных различного типа из различных источников.

Одно из приложений проекта состояло в определении подозрительных компаний из портфеля клиентов Swiss Re. Проект пользовался особой популярностью в период обновления договоров (renewals).

Проект представляет из себя законченную BI-систему, рассчитанную на андеррайтеров и реализованную на платформе StarGate/Palantir Foundry. В проекте используются данные различных типов --- дампы википедии, данные о катастрофах, данные о договорах, данные о контрактах.

Данные проекта приводятся к единому виду и валидируются в блоке аналогичному Конвертеру. Конвертер использовал свои преобразования, специфичные для каждого типа данных. Автоматическая валидация, сделанная на основе data expectations, позволяла сигнализировать о различных проблемах с данными.

Далее данные проекта преобразуются, добавляются результаты внешних ML моделей, добавляются новые аттрибуты. Всё это происходит в блоке похожем на ETL блок. Этот блок состоял из основного пайплайна проекта, который перезапускался раз в неделю или по нажатию кнопки. Именно в этом блоке использовались основные возможности кластера со Spark.

Далее данные присоединяются к существующей онтологии в data warehouse (первоначально использовалась схема звезда, но потом она стала снежинкой). Всё это происходит в блоке, похожем на DWH блок. Присутствует возможность коррекции данных, row level security, есть поддержка версионирования.

Далее эти данные визуализируются с помощью различных приложений Palantir Foundry --- всё это происходит в блоке, похожем на BI блок. За счёт row level security различным отделам предоставляются немного различные данные.

За счёт использования различных возможностей платформы Palantir Foundry достигалось использование таких возможностей как: ролевые модели, версионирование, low code решения, работа с историческими данными, использование внешних библиотек и др.

Таким образом, на проекте фактически была имплементирована вся логика требований, пусть и с применением всех возможностей Palantir Foundry. Мы рассчитываем, что опыт, полученный при работе на проекте поможет в имлементации платформы для РНПК.

1.2 Проект Almanac.

Проект является версионированным хранилищем данных разных типов. В нём поддерживаются различные онтологии (low code решение). При версионировании учитываются связи между различными объектами. Проект основан на опыте, полученном при работе над аналогичными проектами на Swiss Re. Проект является собственностью компании Синкретис и может быть использован в работе на РНПК.

1. Описание предложения.

В данном разделе коротко описывается предложение по имплементации требований по автоматизированной системе сбора, хранения и анализа статистических данных.

Сначала мы посмотрим на схему в целом, а потом разберём структуру каждого модуля.

2.1 Общая структура.

Общая структура в целом следует описанному в требованиях. Есть 4 модуля, у каждого модуля есть свой UI, с учётом ролевой модели, результаты работы каждого модуля доступны, как через UI, так и по REST. Модули называются Конвертер, ETL, DWH, BI, как на картинке ниже.

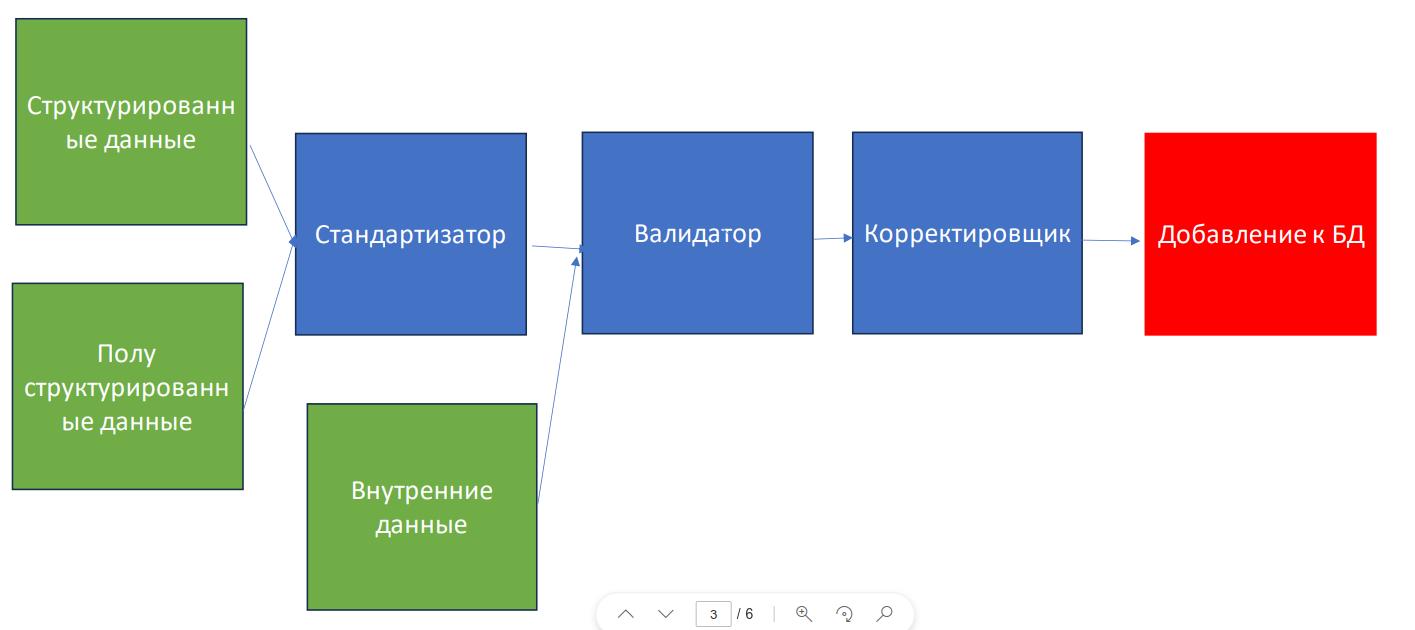
Эти блоки являются неотъемлимой частью любой BI-системы, как например, система, реализованная при работе на ассимметрической платформе обработке данных.

Для продвинутого учёта версионирования планируется использовать Almanac (все данные планируется хранить там).

2.2 Структура конвертера.

Задача конвертера состоит в обработке сырых данных, приведению их к единому типу (т.е. стандартизации), валидации данных. Кроме того, модуль предоставляет возможности ручной корректировки.

Структура модуля поясняется на картинке ниже:



Конвертер может работать как со структурированными данными (csv, xlsx), так и с полуструктурированными данными (json, xml, yaml). Для каждого типа данных (и для каждого шаблона данных) будет включена своя логика в стандартизатор. Задача стандартизатора состоит в приведении данных различных типов к единому формату.

После стандартизатора данные попадают в валидатор, в котором активно используются data expectations. Задача валидатора состоит в нахождении ошибок и несоответствий в данных. Валидатор смотрит на внешние данные и на внутренние данные, находит несоответствия между ними. В случае несоответствий внутренние данные правятся.

После валидатора данные попадают в корректировщик. В этом блоке пользователь может вручную поправить данные, устраняя, например, проблемы, найденные при валидации.

После корректировщика данные попадают в небольшой блок, задача которого состоит в добавлении к БД, т.е. к Almanac, который обеспечивает должную поддержку версионирования.

Все действия логгируются, случаи несоответствий логгируются отдельно, ручные корректировки логгируются отдельно.

В этом модуле не ожидается работа с большими данными. Этот модуль можно спокойно делать на локальной машине.

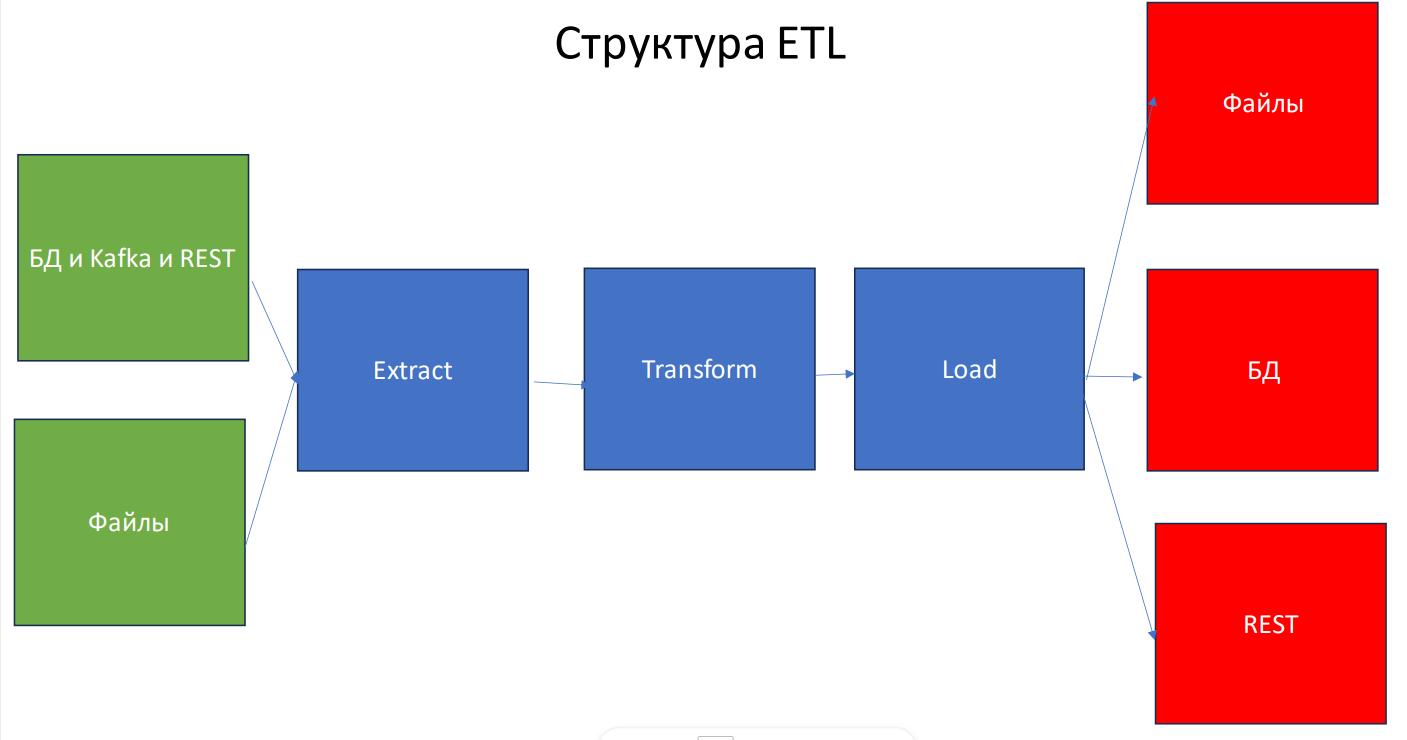
Все основные блоки в модуле придётся писать с нуля в зависимости от данных, которые нам предоставит РНПК. Преобразования будут слишком специфичными, чтобы использовать готовые решения.

2.3 Структура ETL.

Задача ETL блока состоит в запуске пайплайна по работе с данными. Запуск пайплайна проводится как по нажатию кнопки, так и по расписанию, которое можно настраивать с помощью планировщика расписания. В этом блоке будет проходить работа с достаточно большими данными. Есть 2 варианта работы: адаптация готового ETL-решения вроде Apach NiFi или написание с нуля логике по работе на локальной машине в batch моде. У каждого из этих путей есть и свои преимущества, и свои недостатки.

Более перспективным кажется пусть с наличием кластера на территории РНПК и использования Apach Nifi (или Spark, или Hadoop MapReduce), так как этот путь поддерживает масштабирование.

Структура ETL блока описана на картинке внизу:

Блок может брать данные из Almanac, из Kafka, принимать файлы.

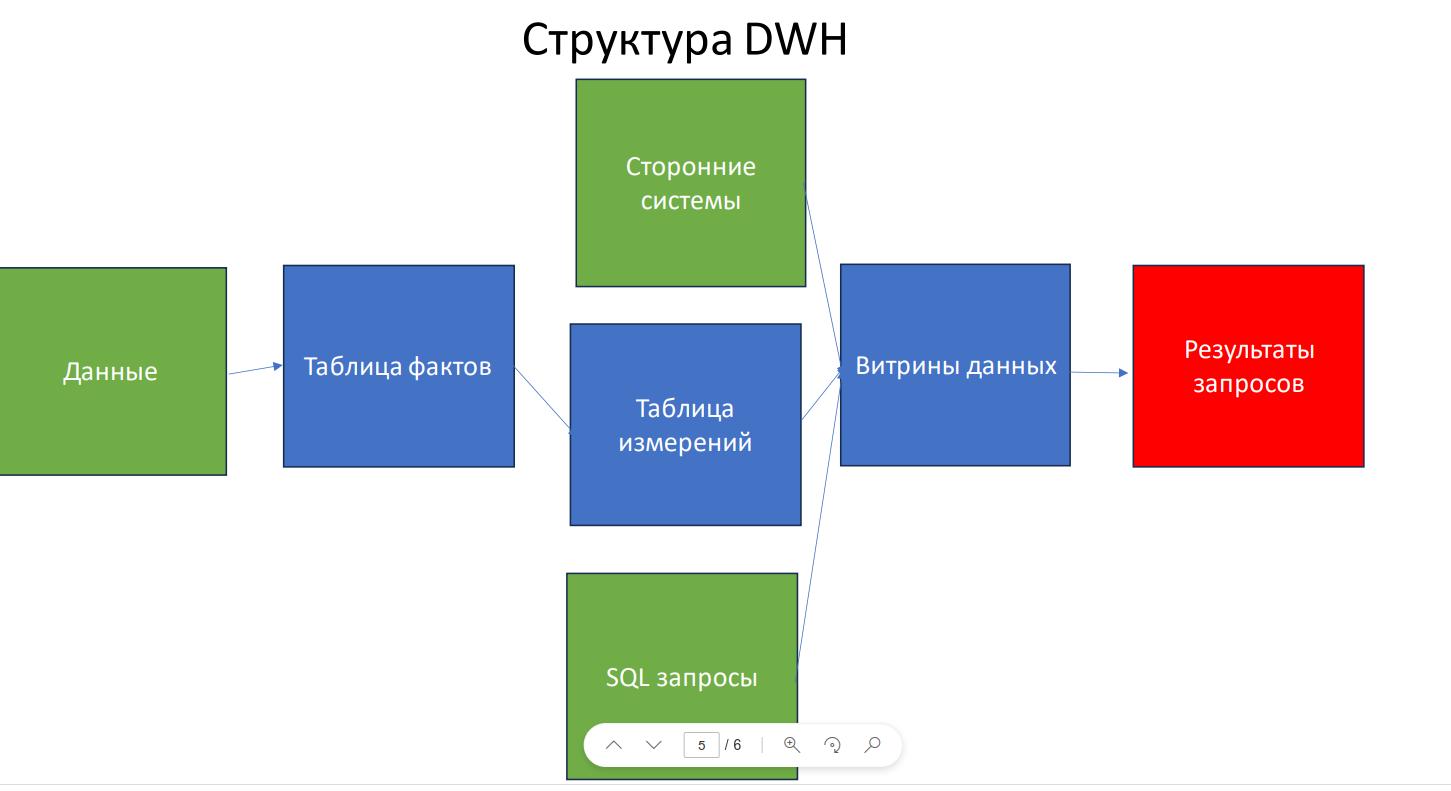
Модуль ETL естественным образом состоит из 3 блоков: extract, transform, load (как следует из названия). На этапе extract берутся данные из различных источников. Для каждого источника пишется своё преобразование. На этапе transform данные преобразуются. Поддерживаются все преобразования, описанные в требованиях. Этап transform --- по сути это то место, когда можно получить преимущество из-за использования big data решений (таких как Apache NiFi). После этапа transform начинается этап load, задача которого состоит в загрузке данных: в Almanac, в файлы, отправке их по REST. Для каждого типа выходной информации будет своя логика в блоке load.

Все действия логгируются, включая запуск пайплайна. За счёт Almanac предусматривается продвинутая логика версионирования (хранение данных в течение 10 лет).

2.4 Структура DWH.

Задача этого модуля состоит в хранении информации в таком виде, чтобы пользователю было бы удобно с ней работать. Этот модуль разумно поместить внутри Almanac, так как в нём происходят вещи, специфичные для работы в БД.

Структура DWH (data warehouse) описывается на картинке ниже.

Модуль берёт данные из Almanac. Данные предлагается хранить по схеме **звезда**, которая может со временем перерасти в схему **снежинка** (как было в **Ассимметричной платформе по обработке данных**). Согласно схеме **звезда** данные хранятся в **таблице фактов**, их аттрибуты --- в **таблицах измерений**.

Данные предлагается записывать в специальные выходные таблицы: витрины данных, выбранные, исходя из бизнес-логики.

При работе с данными и создании витрин используются SQL запросы пользователей, данные их сторонних систем, описанных в требованиях.

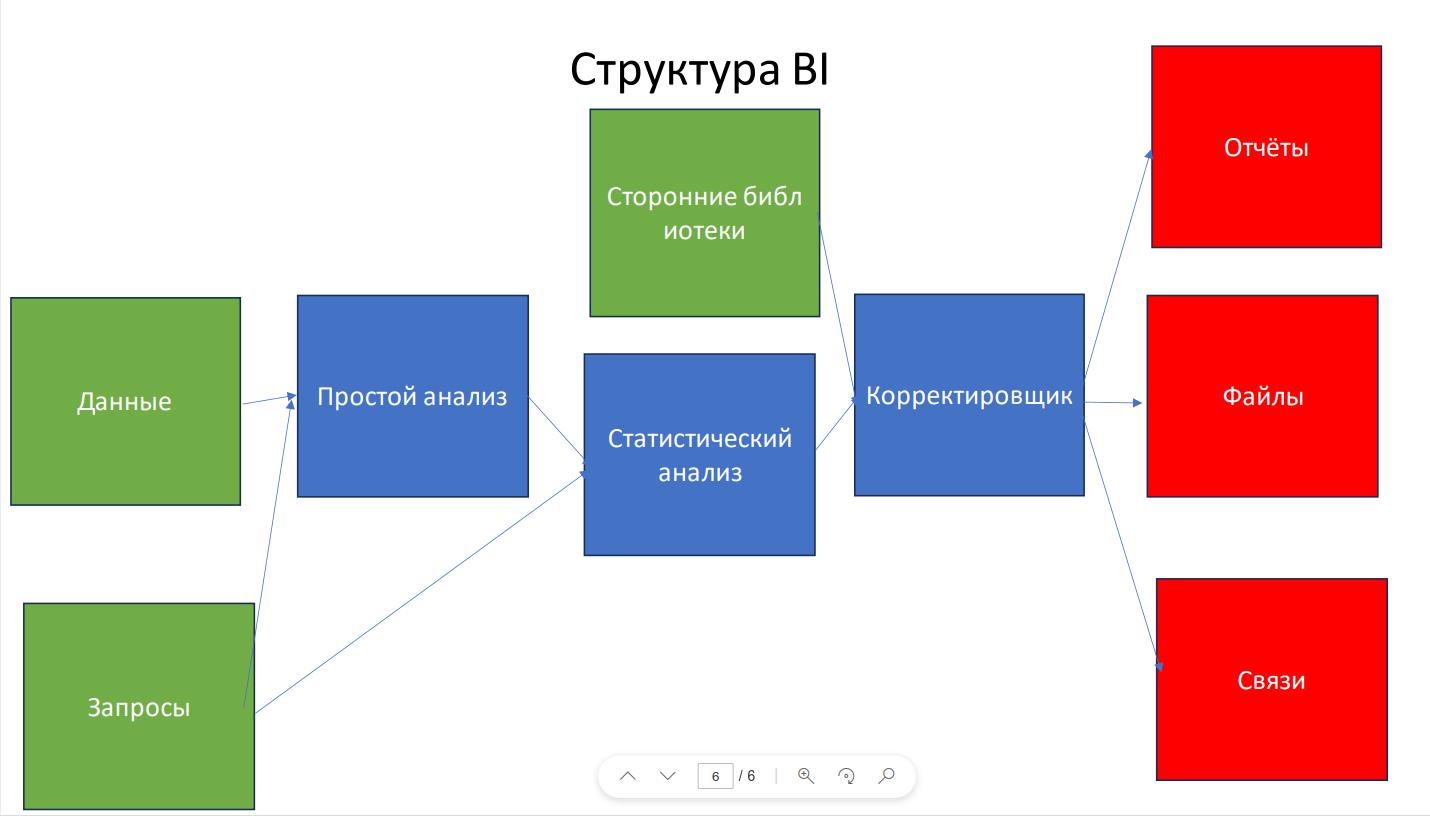
Результаты запросов формируются из витрин данных, они должны быть видны пользователю через UI.

Тут нет смысл использовать стороннее ПО, есть смысл расширять Almanac.

2.5 Структура BI.

Задача данного модуля состоит в визуализации данных, в предоставлении пользователю инструментария, подходящего для различных способов анализа данных. Ядро этого блока визуализация (т.е. FE) --- поэтому, нет смысл писать готовое решение, есть смысл писать свое решение с нуля и добавлять к нему UI.

Структура модуля отражена на картине ниже:



На вход модуль принимает данные из Almanac, запросы пользователей.

В блоке простого анализа проводятся простые преобразования данных (группировка, сортировка, pivot tables --- сводные таблицы и пр.).

В блоке статистического анализа вычисляются различные статистические характеристики, указанные в требованиях, также в нём можно решать задачу регрессии, анализировать структуру данных.

Есть возможность использования сторонних библиотек, в частности библиотек, связанных с ML. В корректирощике предоставляется возможность корректировки данных, все изменения в данных логгируются.

На выходе формируются отчёты согласно различным шаблонам, указанным заказчиками (у каждого отчёта свой шаблон). Кроме того, есть возможность записи выходных результатов в файлы. Также есть возможность визуализации связей между данными, например визуализация структуры портфеля или пересечения портфелей, рисование графиков, диаграмм.

1. Вопросы к требованиям.
2. В документе говорится о приведении неструктурированных данных к структурированных. Задача работа с неструктурированными данными (текстами в произвольной форме) кажется слишком сложной в общем случае. Предлагается ограничиться приведением полуструктурированных или малоструктурированных данных (json, xml, yaml, pdf, doc) к структурированному формату. Это возможно?
3. Мы бы предпочли, чтобы шаблоны загрузки определялись заказчиком, так как они сильно зависят от данных, а мы их не видим. Мы готовы поддерживать любые вменяемые. Это возможно?
4. Предлагается сделать расписание работы с ETL данными параметром, так как оно зависит от технических характеристик, от нужд компании, это расписание может измениться со временем. Это возможно?
5. Мы предпочли, чтобы перечень форматов загрузчиков (50) полностью определялся заказчиком при составлении ТЗ, так как заказчики лучше знают свои данные. Это возможно?
6. Есть ли необходимость поддерживать маппинг по текстовым полям (нечёткое соответствие, исправление опечаток, маппинг с помощью ML)?
7. Операция транспонирование подразумевает смену строк и столбцов местами, судя по требованиям речь идёт о переходах между широким (wide) и длинном (long) форматами данных. Это так?
8. Можно ли конкретизировать, что такое “низменная последовательность колонок”?
9. Как долго должны жить логи в системе логгирования? Тоже 10 лет? Или в каких-то случаях меньше, например, месяц?
10. Есть ли необходимость предусмотреть механизм профилирования пайплайна в блоке ETL (возможность увидеть время работы для разных частей пайплайна, возможность перезапуска пайплайна с какого-то места)?
11. Какие типы union и join предлагается поддерживать?