



Использование языковых моделей для
поиска по Confluence и сокращения
рутины аналитика



Цель

Исследовать весь потенциал использования передовых языковых моделей (семейства GPT, Claude, Gemini) для повышения эффективности работы компании:

- Исследуем **не только Open Source** решения, а самые мощные на открытых данных (Семейства GPT, Claude, Gemini)
- Делаем чат-бота за минимум времени **без написания кода**, проверяем опцию - “Работа с документами”
- Автоматизация рутины аналитика и изменение его роли (сводим роль аналитика к правильной постановке вопросов, описанию ожидаемого результата и валидации ответов с небольшими правками)



Поиск по базе знаний компании с помощью LLM. Семейство Claude 3

Claude — чат-бот с искусственным интеллектом для генерации текста.

Создатели нейросети — [стартап Anthropic AI](#), который в 2021 году основали выходцы из OpenAI — компании-разработчика ChatGPT. В конце марта 2024 года языковая модель Claude 3 [превзошла GPT-4 на платформе Chatbot Arena](#), где пользователи сравнивают возможности разных чат-ботов на базе ИИ и выбирают «победителя».

Claude умеет отвечать на вопросы, писать тексты, генерировать идеи, решать математические задачи, помогать с программированием. Одна из главных особенностей — **умение работать с длинным контекстом**



Примеры использования и потенциальные выгоды

Что на входе?

На вход загружена небольшая база данных, которая содержит постановки на новые унифицированные агрегаты

Что дальше?

Задаем ряд вопросов, получаем ответы, оцениваем результат

Что на выходе?

Результат превзошел ожидания: ответы получены качественные, развернутые и без ошибок в логике



Vika

Какие агрегаты содержатся в базе знаний? Выдай перечень



ChatGPT | Midjourney | Claude | Suno AI — GPT4Telegrambot Inc.

В базе знаний содержатся следующие агрегаты:

1. agg_gismt.writeoffs
2. agg_gismt.cis
3. agg_gismt.codes_history
4. agg_gismt.daily_sp_sales
5. agg_gismt.daily_sp_sales_kkt
6. agg_gismt.funnel
7. agg_common.accruals
8. agg_gismt.billing_documents
9. agg_gismt.billing_documents_stat
10. agg_gismt.in_circulation
11. agg_gismt.codes_turnover
12. agg_gismt.billing

Это основной перечень агрегатов, которые содержатся в базе знаний. Некоторые из них находятся в стадии разработки или проектирования, о чем указано в соответствующих разделах.



Vika

Расскажи об агрегате `agg_gismt.writeoffs`



ChatGPT | Midjourney | Claude | Suno AI — GPT4Telegrambot Inc.

Агрегат `agg_gismt.writeoffs` предназначен для хранения информации о списаниях марок табачной продукции в ГИС МТ.

Основные характеристики агрегата:

1. Назначение:

- Учет списаний марок табачной продукции в ГИС МТ.
- Хранение подробной информации о каждом факте списания.

2. Источники данных:

- Данные о списаниях из ГИС МТ.

3. Ключевые поля:

- `id` - уникальный идентификатор записи
- `writeoff_id` - уникальный идентификатор списания
- `writeoff_date` - дата списания
- `writeoff_type` - тип списания (например, брак, утрата, возврат и т.д.)
- `writeoff_reason` - причина списания
- `writeoff_quantity` - количество списанных марок
- `product_id` - идентификатор продукта
- `product_name` - наименование продукта
- `product_code` - код продукта
- `product_type` - тип продукта (например, сигареты, сигары и т.д.)
- `product_brand` - бренд продукта
- `product_category` - категория продукта
- `product_subcategory` - подкатегория продукта
- `organization_id` - идентификатор организации
- `organization_name` - наименование организации
- `organization_inn` - ИНН организации



Vika

А в чем отличие агрегатов: `agg_gismt.writeoffs` и `agg_gismt.daily_sp_sales`?



ChatGPT | Midjourney | Claude | Suno AI — GPT4Telegrambot Inc.

Агрегаты `agg_gismt.writeoffs` и `agg_gismt.daily_sp_sales` имеют следующие отличия:

1. Источники данных:

- `agg_gismt.writeoffs` использует данные из промежуточного слоя `cis_state`, в частности, данные по операциям списания (`t=9, 103, 104, 21, 124`).
- `agg_gismt.daily_sp_sales` использует данные из промежуточного слоя `cis_state`, в частности, данные по операциям выбытия (`t=6, 118, 13, 108`).

2. Назначение:

- `agg_gismt.writeoffs` содержит информацию по списаниям, аннулированиям и их причинам.
- `agg_gismt.daily_sp_sales` содержит информацию по всем выбытиям КМ, за исключением списаний, которые учтены в `agg_gismt.writeoffs`.

3. Алгоритм формирования:

- `agg_gismt.writeoffs` отбирает операции списания (`t=9, 103, 104, 21, 124`) и выбытия (`t=6` с определенными `elr`) при выполнении определенных условий.
- `agg_gismt.daily_sp_sales` отбирает операции выбытия (`t=6, 118, 13, 108`) и рассчитывает показатели первичной продажи от владельца и невладелецца.

4. Состав полей:

- `agg_gismt.writeoffs` содержит поля, связанные со списаниями и аннулированиями, такие как причина списания (`elr`), тип операции (`t`), количество списанных КМ (`cnt`).
- `agg_gismt.daily_sp_sales` содержит поля, связанные с выбытиями, такие как ИНН продавца (`inn`), количество проданного товара (`cnt, cnt_attempts`), цена (`price, vat`).

В целом, `agg_gismt.writeoffs` фокусируется на учете списаний и аннулирований, а `agg_gismt.daily_sp_sales` - на учете всех выбытий, за исключением списаний.



Vika

Из каких источников строится агрегат?



ChatGPT | Midjourney | Claude | Suno AI — GPT4Telegrambot Inc.

Агрегат `agg_milk.in_circulation` строится из следующих источников:

1. HDFS `/data/milk/silver/work/data_circulation`

- Это стейджинговая таблица, содержащая данные о вводе в оборот молочной продукции по экземплярному учету.

2. HDFS `data/milk/staging/data_imp`

- Стейджинговая таблица, содержащая данные об импорте молочной продукции.

3. ClickHouse `dict.milk_gtins`

- Справочник, содержащий информацию о GTIN-кодах молочной продукции, в том числе признаки подконтрольности и переменного веса.

Таким образом, агрегат объединяет данные о вводе в оборот из стейджинговых таблиц, обогащая их справочной информацией из `dict.milk_gtins`.

Сокращение рутины аналитика с помощью языковых моделей



Выявим рутину и делегируем LLM:

1. Проверка постановки и ее возможная оптимизация
2. Тестирование
3. Проведение Демо заказчику
4. Оптимизация SQL запросов
5. Дайджест



Vika

Есть ли в постановке из документа ошибки? найди их



ChatGPT | Midjourney | Claude | Suno AI — GPT4Telegrambot Inc.

Я внимательно проанализировал постановку из документа и не нашел явных ошибок. Документ выглядит проработанным и содержит подробное описание алгоритма расчета агрегата `agg_milk.in_circulation`. Однако, есть несколько моментов, на которые стоит обратить внимание:

1. Поле `country`:

- В описании указано, что это поле будет заполняться значением `sender_country_code` при его наличии, иначе будет использоваться значение `country`.
- Однако, в алгоритме для сортового учета это поле не заполняется, там стоит пустая строка `' '`.
- Возможно, стоит проверить, нужно ли заполнять это поле и для сортового учета.

2. Поле `inn_return`:

- В описании указано, что это поле будет заполняться значением `sinn` для повторных вводов в оборот (`operation_type = 106`) и это не первый ввод в оборот.
- Однако, в алгоритме для сортового учета это поле не заполняется.
- Возможно, стоит проверить, нужно ли заполнять это поле и для сортового учета.

3. Поле `release_date` и `decl_date`:

- В описании указано, что эти поля не актуальны для сортового учета и будут заполняться значением `'1970-01-01'`.
- Однако, в алгоритме для сортового учета эти поля не заполняются.
- Возможно, стоит проверить, нужно ли заполнять эти поля и для сортового учета.

Кроме того, в описании есть несколько комментариев, помеченных как "будущие доработки", которые стоит учесть при дальнейшей работе над агрегатом.

В целом, документ выглядит хорошо проработанным, и я не вижу в нем явных ошибок. Однако, стоит внимательно проверить согласованность алгоритмов для экземплярного и сортового учета.



Vika

А можно ли сократить и оптимизировать постановку?



ChatGPT | Midjourney | Claude | Suno AI — GPT4Telegrambot Inc.

Да, можно оптимизировать и сократить постановку агрегата `agg_milk.in_circulation`. Вот некоторые предложения:

1. Упростить алгоритм определения значений полей:

- Для поля `is_imp` можно использовать более простую логику: `is_imp = CASE WHEN data_in_circulation.j_decr IS NOT NULL THEN 1 ELSE 0 END.`

- Для поля `pvad` можно использовать более простую нормализацию: `pvad = REPLACE(data_in_circulation.pvad, '-', '')`.

- Для поля `gtin_volume_measure` можно использовать более простую логику: `gtin_volume_measure = CASE WHEN is_var_quantity THEN 'кг' ELSE milk_gtins.weight_measure_unit END.`

2. Упростить логику определения значений полей `oinn` и `actual_prid`:

- Для поля `oinn` можно использовать более простую логику: `oinn = COALESCE(dict.oinn_osu_exceptions.oinn, data_in_circulation.j_contract0, data_in_circulation.inn).`

- Для поля `actual_prid` можно использовать более простую логику: `actual_prid = COALESCE(oinn, data_in_circulation.prid).`

3. Упростить логику определения значения поля `is_correct`:

- Можно использовать более простую логику, например: `is_correct = CASE WHEN data_in_circulation.cor = '1' THEN 0 WHEN data_in_circulation.document_type IN (133, 134, 135, 141) AND data_in_circulation.operation_type = 113 THEN 1 WHEN data_in_circulation.document_type IN (...) AND data_in_circulation.operation_type = 12 THEN 1 ELSE -1 END.`

4. Упростить логику определения значения поля `inn`:

- Можно использовать более простую логику, например: `inn = CASE WHEN data_in_circulation.document_type IN (188, 29) AND data_in_circulation.operation_type = 108 THEN data_in_circulation.rinn WHEN`



Vika

Предложи основные сценарии тестирования агрегата `agg_gismt.writeoffs`



ChatGPT | Midjourney | Claude | Suno AI — GPT4Telegrambot Inc.

Для тестирования агрегата `agg_gismt.writeoffs` можно предложить следующие основные сценарии:

1. Проверка корректности расчета количества списанных КМ (cnt):
 - Проверить, что количество списанных КМ (cnt) соответствует количеству уникальных КМ в исходных данных с операциями списания ($t=9, 103, 104, 124$).
 - Проверить, что количество списанных КМ (cnt) учитывает только те КМ, у которых статус (st) равен 3 (списан).
2. Проверка корректности заполнения полей, связанных с датами:
 - Проверить, что дата списания (dt) соответствует дате операции списания (`write_off_opd`).
 - Проверить, что дата эмиссии (`emission_date`) соответствует дате эмиссии (`to_date(emission_opd)`).
 - Проверить, что дата нанесения (`application_date`) соответствует дате нанесения (`to_date(applied_opd)`).
 - Проверить, что дата ввода в оборот (`circulation_dt`) соответствует дате ввода в оборот (`to_date(circulation_opd)`).
 - Проверить, что дата оплаты (`pay_dt`) соответствует дате первой платной оплаты (`payment_set[0].opday`).
3. Проверка корректности заполнения полей, связанных с идентификаторами:
 - Проверить, что ИНН производителя (prid) корректно заполнен.
 - Проверить, что ИНН списывающей организации (sinn) корректно заполнен.
 - Проверить, что место осуществления деятельности (omod) корректно заполнено.
4. Проверка корректности заполнения полей, связанных с характеристиками товара:
 - Проверить, что GTIN товара (gtin) корректно заполнен.
 - Проверить, что тип упаковки (pt) и расширенный тип упаковки (pte) корректно заполнены.
 - Проверить, что товарная группа (pg) корректно заполнена.
5. Проверка корректности заполнения поля причины списания (elr):
 - Проверить, что причина списания (elr) корректно заполнена для каждой записи.



Сверь загруженные данные из файла data_writeoffs и постановку на агрегат agg_gismt.writeoffs. Найди ошибки



ChatGPT | Midjourney | Claude | Suno AI — GPT4Telegrambot Inc.

Хорошо, давайте сверим данные из файла data_writeoffs и постановку на агрегат agg_gismt.writeoffs.

1. Проверим соответствие полей:

Поле в агрегате | Поле в файле data_writeoffs | Комментарий

— | — | —

dt | dt | OK

emday | emday | OK

apday | apday | OK

circulation_dt | circulation_opd | Разные названия, но соответствуют

pay_dt | payment_set[0].opday | Разные названия, но соответствуют

gtin | gt | OK

pt | pt | OK

pte | pte | OK

sinn | coalesce(write_off.sinn, sales.sinn) | Разные названия, но соответствуют

omod | coalesce(write_off.omod, sales.omod) | Разные названия, но соответствуют

elr | coalesce(write_off.elr, sales.elr) | Разные названия, но соответствуют

t | coalesce(write_off.t, sales.t) | Разные названия, но соответствуют

pg | pg | OK

et | et | OK

prid | prid | OK

oinn | oinn | OK

actual_prid | actual_prid | OK

plid | plid | OK

cnt | count(c) | Разные названия, но соответствуют

c_example | max(c) | Разные названия, но соответствуют

ts_insert | - | Нет в файле

arch | - | Нет в файле

domain | - | Нет в файле

Все поля соответствуют, кроме ts_insert, arch и domain, которых нет в файле data_writeoffs.



Агрегат `agg_gismt.writeoffs` формируется из следующих источников:

1. Экземплярный учет из промежуточного слоя `cis_state`:
 - Операции списания ($t=9, 103, 104, 21, 124$) где $st=3$ или операции вывода из оборота ($t=6$) с `elr` из группы "Списание/вывод из оборота без получателя".
2. Сортвой учет из золотого слоя `/data/unrelated/gold/osu_sales`:
 - Операции вывода из оборота ($t=6$) с `elr` из группы "Списание/вывод из оборота без получателя".

Для сверки источника можно использовать следующие запросы:

1. Проверка данных из `cis_state`:

```
sql
SELECT
  c.c,
  c.dt,
  c.sinn,
  c.omod,
  c.elr,
  c.t
FROM agg_gismt.writeoffs a
JOIN cis_state c ON a.c = c.c
WHERE c.t IN (9, 103, 104, 21, 124) AND c.st = 3
      OR (c.t = 6 AND c.elr IN (SELECT elr FROM dict.lp_elr WHERE group =
'Списание/вывод из оборота без получателя'))
```

2. Проверка данных из `osu_sales`:

```
sql
SELECT
  a.c,
  a.dt,
  a.sinn,
  a.omod,
```




Vika

Напиши ТОП-5 запросов к agg_gismt.daily_sp_sales для Демо заказчика?



ChatGPT | Midjourney | Claude | Suno AI — GPT4Telegrambot Inc.

Вот пример ТОП-5 запросов к агрегату agg_gismt.daily_sp_sales для Демо заказчика:

1. Общий объем продаж по дням:

```
sql
SELECT dt, sum(cnt) as total_sales
FROM agg_gismt.daily_sp_sales
WHERE pg = 8 -- Молоко
GROUP BY dt
ORDER BY dt DESC
LIMIT 10;
```

2. Топ-10 GTIN по объему продаж:

```
sql
SELECT gt, sum(cnt) as total_sales
FROM agg_gismt.daily_sp_sales
WHERE pg = 8 -- Молоко
GROUP BY gt
ORDER BY total_sales DESC
LIMIT 10;
```

3. Топ-10 производителей по объему продаж:

```
sql
SELECT prid, sum(cnt) as total_sales
FROM agg_gismt.daily_sp_sales
WHERE pg = 8 -- Молоко
GROUP BY prid
ORDER BY total_sales DESC
```



Vika

Напиши SQL запрос, который выводит отчет о количестве кодов маркировки по годам и типам эмиссии



ChatGPT | Midjourney | Claude | Suno AI — GPT4Telegrambot Inc.

Вот SQL-запрос, который выводит отчет о количестве кодов маркировки по годам и типам эмиссии:

```
sql
SELECT
    date_trunc('year', dt) AS year,
    emission_type,
    COUNT(*) AS count
FROM agg_milk.agg_in_circulation
GROUP BY date_trunc('year', dt), emission_type
ORDER BY year, emission_type;
```

Этот запрос выполняет следующие действия:

1. Группирует данные по году (используя `date_trunc('year', dt)`) и типу эмиссии (`emission_type`).
2. Для каждой группы подсчитывает количество записей (`COUNT(*)`).
3. Сортирует результаты по году и типу эмиссии.

Результат запроса будет выглядеть примерно так:

year	emission_type	count
2020	0	1000
2020	4	500
2021	0	1500
2021	4	800
2022	0	2000



Vika

Ты опытный копирайтер. А теперь придумай 10 ярких кратких заголовков с юмором и без (5 слов максимум) для сделанной задачи. Предложи 10 вариантов



ChatGPT | Midjourney | Claude | Suno AI — GPT4Telegrambot Inc.

Вот 10 вариантов ярких кратких заголовков для задачи BFR-13988:

1. Закрытие периодов: Фиксируй и контролируй!
2. Акцизный срез: Ежемесячный срез данных
3. Табак под контролем: Закрытие периодов в деле
4. Бухгалтерский детектив: Расследуем акцизные данные
5. Акцизный агрегат: Твой надежный помощник
6. Закрытие периодов: Точность и прозрачность
7. Табачный учет: Автоматизируем закрытие периодов
8. Акцизный срез: Ключ к финансовой стабильности
9. Табачный контроль: Фиксируй, анализируй, управляй
10. Акцизные данные: Под замком, но доступны



Vika



BFR-13988.doc

18.7 KB - [Show in Finder](#)



ChatGPT | Midjourney | Claude | Suno AI — GPT4Telegrambot Inc. pinr



Vika



330607677_3b1f80f1113f4374b9f2a3de801df932-040624-1529-560.pdf

150.1 KB - [Show in Finder](#)

Напиши результат по проделанной работе в 2 абзацах и в 2-х вариантах: емко и с юмором.
Информация будет использована для рассылки для дайджеста



ChatGPT | Midjourney | Claude | Suno AI — GPT4Telegrambot Inc.

Вариант 1 (емко):

Команда разработчиков успешно реализовала агрегат для закрытия периодов акцизных данных. Теперь ФНС может получать регулярные срезы данных в заданные даты, что позволит им эффективно отслеживать изменения и доначислять акциз при необходимости. Проект завершен в срок, и пользователи уже оценили новый функционал.

Вариант 2 (с юмором):

Разработчики провернули настоящий "акцизный переворот", создав агрегат для закрытия периодов данных. Теперь ФНС может ежемесячно устраивать "акцизные ревизии", сверяя актуальные цифры с зафиксированными срезами. Можно представить, как налоговики потирают руки в предвкушении доначислений. Проект завершен вовремя, и пользователи уже готовы танцевать "акцизный танец победы".



Технические аспекты внедрения. Ограничения модели

Размер файлов.

За раз можно прикрепить 5 файлов размером не больше 10 Мб каждый, включая изображения и документы. Лимиты касаются и платной, и бесплатной версий. То есть, если вы работаете с крупным файлом, нужно разделить его на более мелкие. Это неудобно. Для сравнения: в платный ChatGPT можно загрузить файл объемом 512 Мб, то есть в 50 раз больше.

Пользоваться в России сложно.

Для регистрации на сайте обязательно нужен зарубежный номер и иностранный IP-адрес. Существуют сторонние сайты с более свободным доступом к нейросети, но у них свои ограничения (использовала телеграмм)



Технические аспекты внедрения

1. Выделение разных конкретных задач с понятными сценариями использования и ролями (например, сокращение времени на онбординг новых дата-аналитиков)
2. Данные для обучения под конкретную задачу (сейчас есть ограничение на длину контекста и нужно делить информацию под разные задачи). Продумать механизмы для сбора, фильтрации и разметки данных
3. Интеграция и API (с учетом ограничений и блокировок в РФ):
 - а. Разработка API и интерфейсов для интеграции модели Claude 3 с существующими системами и приложениями.
 - б. Документация и поддержка для разработчиков, использующих модель.
4. Серверная и сетевая инфраструктура - нужно исследовать
5. Финансовые затраты на запросы через API



Открытые вопросы/Выводы

Чат-бот по собственной базе знаний:

1. Конкурировать с передовыми решениями сложно и дорого, а также они всегда будут на шаг впереди. Самый быстрый путь - использовать модель Claude Opus, исследовать интеграцию/делать расчет стоимости и следить за новинками (есть вариант, что через несколько месяцев она уже станет бесплатной, но вероятно с лимитом запросов или Яндекс внедрит нечто похожее)
2. Вопросы безопасности. Использование нашей базы знаний и данных в облаке (один из путей - обезличивать данные)
3. Сбор и подготовка качественных обучающих данных на русском языке являются критически важными для успешного обучения модели
4. Успешное внедрение потребует команды экспертов включая devops, специалистов по системной интеграции, расчету вариантов развертывания инфраструктуры

Автоматизация рутины аналитика (сокращаем прежнюю работу в 2 раза)

1. Локально уже использовать можно и нужно
2. Выявлять рутину локально и далее делать более системное решение через интеграции с Claude-3 (идти от конкретной задачи, “боли” и масштабировать, пример - автоматизация дайджеста с 2-х дней до 2-х часов)

Итого: приоритезация цели, ограничения и ресурсы