

PGMeetup.PERM 2025

Генерация SQL-запросов на естественном языке: реальная необходимость или погоня за модой?

Виталий Ефимов

Инженер по тестированию



Что мы сегодня обсудим?

Что такое SQL Generator?

Почему нейронки интересно тестировать?

Как и зачем использовать нейросети при работе с БД?

Что ещё мы умеем?

Как это работает?

Почему нейросети — это не всегда круто?

Ефимов Виталий – QA in ML в Postgres Pro

Работал в Deutsche Bank и TON

Тестировал RecSys для онлайн-кинотеатров

Переехал из Курска в Германию,
а затем наоборот

Выступил со стендапом в 7 городах
и 5 европейских странах



Who is mr. SQL Generator?

Генератор SQL-запросов на основе реальной базы данных пользователя

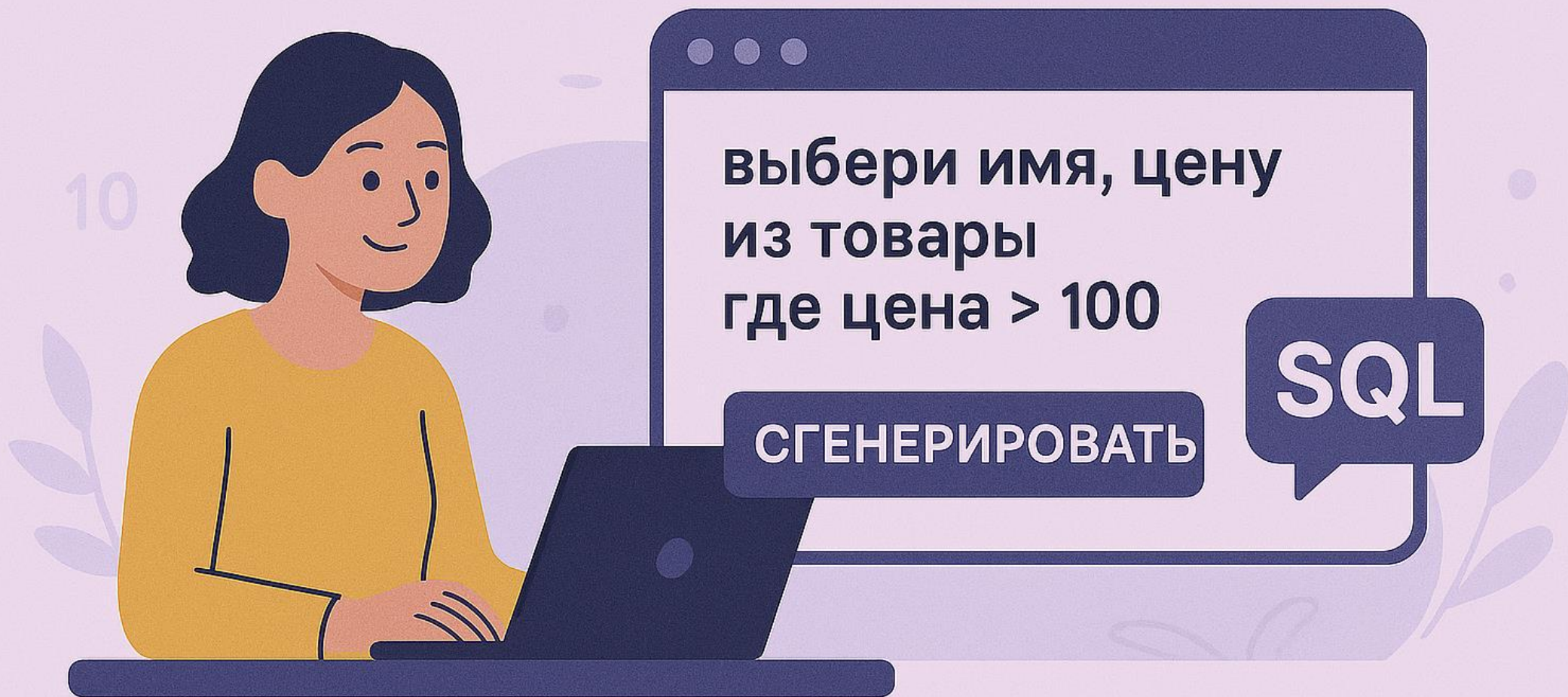
Не требует глубоких знаний SQL

Вывод полезных данных в виде агрегированной таблицы

В перспективе: генерирует ответ быстрее, чем вы вспоминаете название таблицы

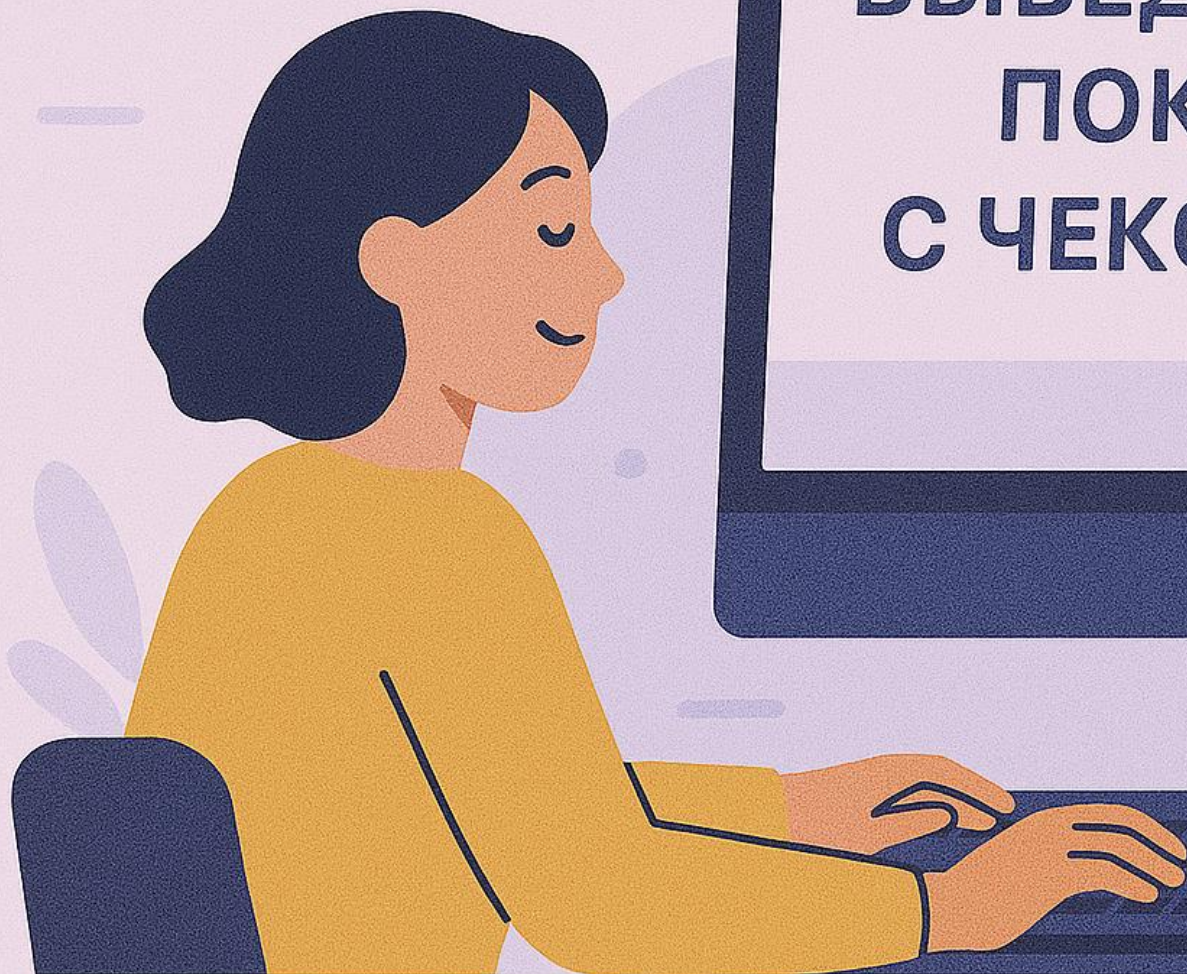
Позволяет писать запросы к базе данных на естественном языке (русский, английский)

SQL GENERATOR НА РУССКОМ ЯЗЫКЕ



SQL

**ВЫВЕДИ МНЕ СПИСОК
ПОКУПАТЕЛЕЙ
С ЧЕКОМ ВЫШЕ 100**





Выведи список заказчиков, на проектах которых было потрачено больше всего краски.
Что я хочу получить: название компании, название проекта, количество краски,
стоимость краски

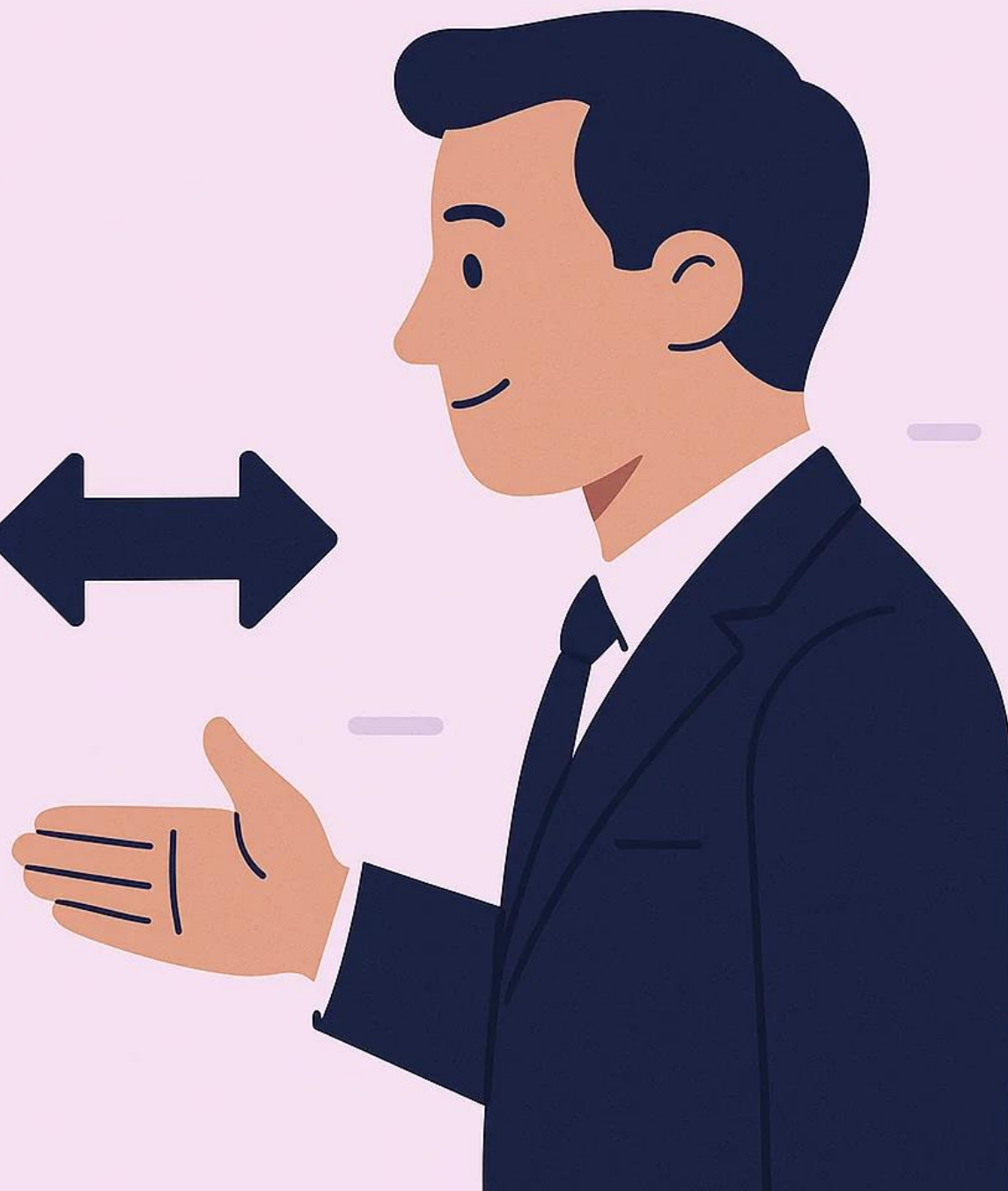
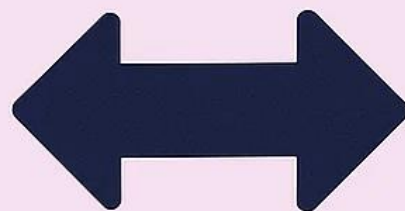


```
SELECT
    c.name AS company_name,
    p.name AS project_name,
    SUM(mu.quantity) AS total_paint_quantity,
    SUM(mu.quantity * m.price_per_unit) AS total_paint_cost
FROM
    public.clients c
JOIN
    public.projects p ON c.client_id = p.client_id
JOIN
    public.construction_sites cs ON p.project_id = cs.project_id
JOIN
    public.material_usages mu ON cs.site_id = mu.site_id
JOIN
    public.materials m ON mu.material_id = m.material_id
WHERE
    m.name ILIKE '%краска%'
GROUP BY
    c.client_id, p.project_id, c.name, p.name
ORDER BY
    total_paint_cost DESC;
```


company_name	project_name	total_paint_quantity	total_paint_cost
ОАО «Кудрявцева-Одинцова»	Строительство объекта №30	460.64	483063.9552
Субботина Лтд	Строительство объекта №1	272.63	285901.6284

Почему это круто?

ДАННЫЕ БИЗНЕСА



Как это работает?

Пользователь отправляет запрос формата:

«Напиши количество людей, которые улетели в Питер и вернулись обратно в течение одной недели»

```
graph TD; A[Пользователь отправляет запрос формата: «Напиши количество людей, которые улетели в Питер и вернулись обратно в течение одной недели»] --> B[Нейросеть изучает запрос]; B --> C[Смотрит на схему базы данных и данные в ней]; C --> D[Понимает, какие данные и откуда можно вытащить]; D -.-> E[...];
```

Нейросеть изучает запрос

Смотрит на схему базы данных и данные в ней

Понимает, какие данные и откуда можно вытащить



Формирует SQL-запрос и выполняет его

Выводит полученные данные для пользователя
в виде агрегированной таблицы

**Вы бежите показывать начальнику
«сложнейший отчёт, который делали 3 дня»**

Database Scheme

```
site_manager_id integer,  
status character varying(50)(50),  
start_date date,  
end_date date,  
PRIMARY KEY (site_id),  
FOREIGN KEY (project_id) REFERENCES  
public.projects(project_id),  
FOREIGN KEY (site_manager_id) REFERENCES  
public.employees(employee_id)  
);  
-- Row count: 150  
-- Sample data: {"site_id":1,"project_id":23,"name":"Объект  
строительства 1","site_manager_id":104,"status":"на  
паузе","start_date":"2025-01-08","end_date":"2026-03-01"}
```




Напиши количество людей, которые улетели в Питер и вернулись обратно в течение одной недели



```
SELECT COUNT(DISTINCT t.passenger_id) AS total_people
FROM tickets t
JOIN ticket_flights tf1 ON t.ticket_no = tf1.ticket_no
JOIN flights f1 ON tf1.flight_id = f1.flight_id AND f1.arrival_airport = 'LED'
JOIN ticket_flights tf2 ON t.ticket_no = tf2.ticket_no
JOIN flights f2 ON tf2.flight_id = f2.flight_id AND f2.departure_airport = 'LED'
WHERE
    f2.scheduled_departure >= f1.scheduled_arrival
    AND f2.scheduled_departure <= f1.scheduled_arrival + INTERVAL '7 days'
    AND f1.flight_id <> f2.flight_id;
```

total_people
19400

В рамках экспериментов мы использовали
различные модели, в том числе
дистиллированную модель от DeepSeek

Потому что это модно, молодёжно
и 在随机位置产生象形文字



Chinese (Simplified)

Russian

German



Генерирует иероглифы в случайных местах



Generiruyet iyeroglify v sluchaynykh mestakh



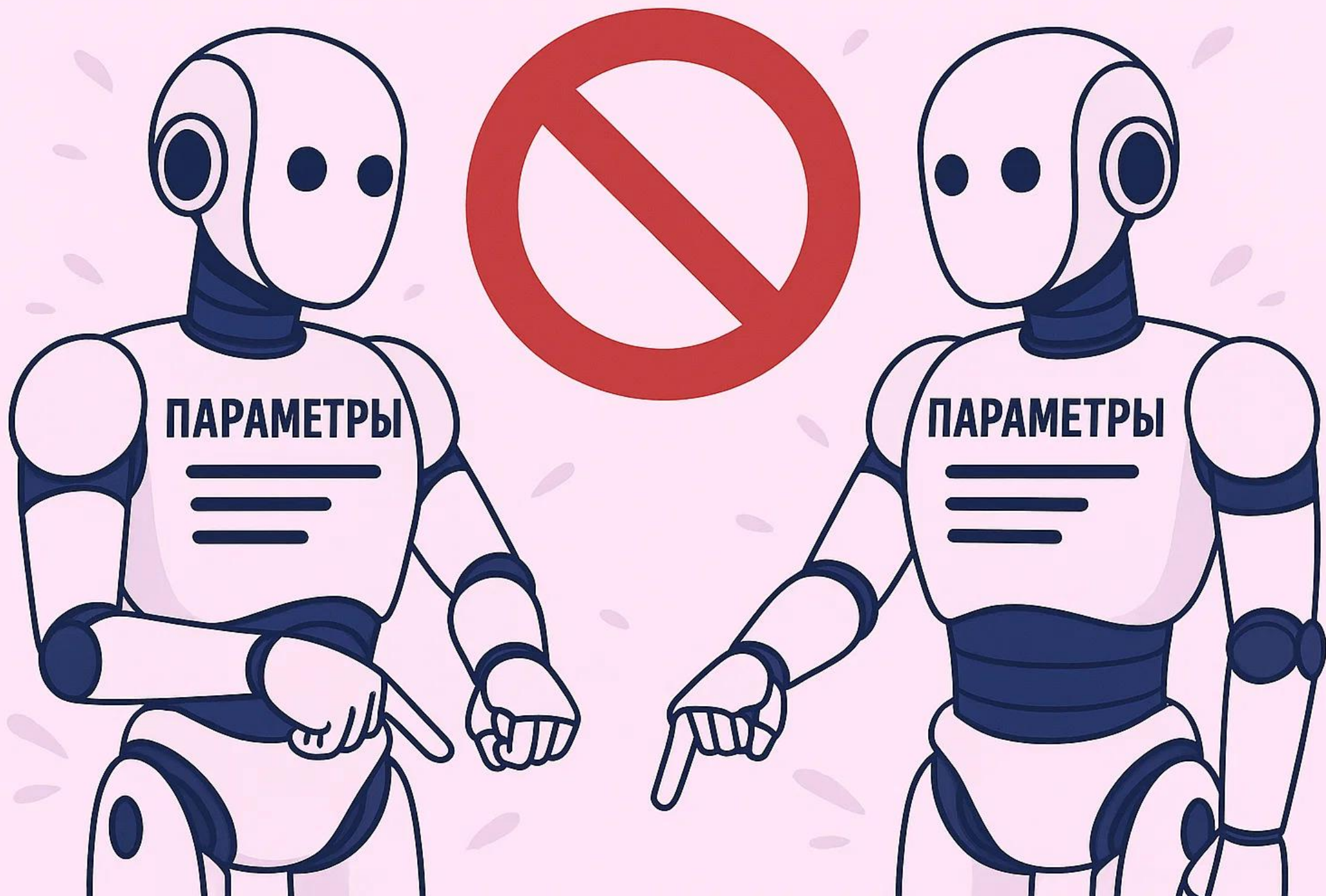
[Send feedback](#)

На текущий момент мы остановились
на модели **Qwen3-30B-A3B-AWQ**, которая
содержит 30 миллиардов параметров

Много ли это?..
Смотря с чем сравнивать

ChatGPT 4o содержит 200-300 млрд
параметров

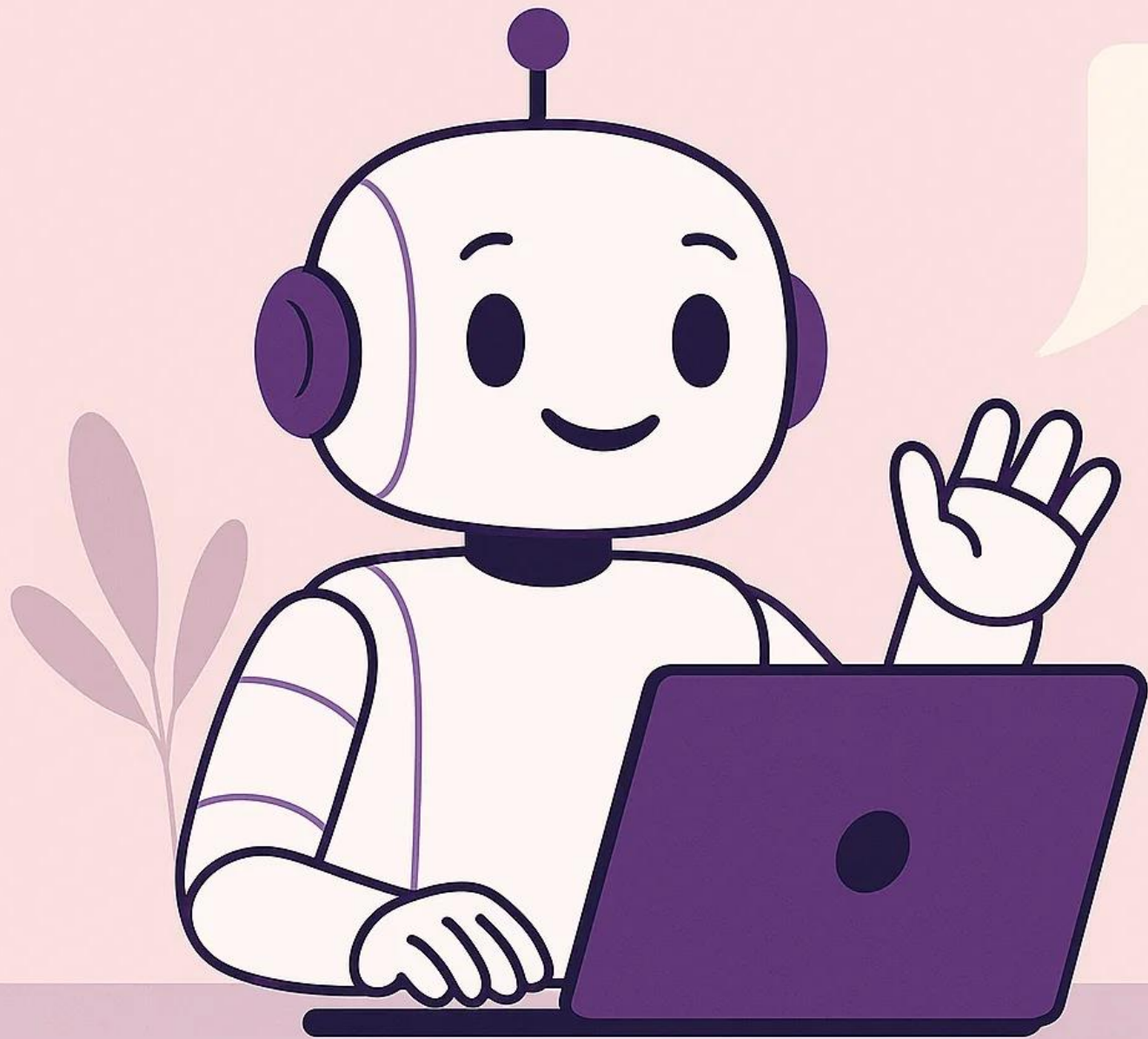
А ChatGPT 4 – 1.8 **триллиона**



**Языковую модель очень
весело тестировать**

При генерации SQL-запроса модель 9 раз выдавала корректный ответ, а на 10-й выдала кириллическую букву «С» в слове SELECT —
получился СЕЛЕСТ

Внезапно на запрос **«привет»** модель предложила нам связаться с неким Ashokkumar Mani, который может помочь по этому вопросу



привет

 **+7 123 456-78-90**

 **person@example.com**

 **г. Москва,
ул. Пушкина, д. 1**

Боюсь узнать, что она бы предложила
на запрос «**как дела?**»

В документацию случайно попало **письмо**
с темой «test», присланное от
Ashokkumar Mani

Re: test

From: Madeline Cummins <maddyc411(at)hotmail(dot)com>
To: Blake Burton <webmaster(at)uspto(dot)gov>, Madeline Cummins <maddy5059(at)hotmail(dot)com>, dbatoCloud Solution <dbatocloud17(at)gmail(dot)com>
Cc: pgsql-admin(at)lists(dot)postgresql(dot)org
Subject: Re: test
Date: 2020-02-28 04:14:26
Message-ID: BYAPR19MB2613C9AAF264D8747AFF0885EDE80@BYAPR19MB2613.namprd19.prod.outlook.com
Views: [Raw Message](#) | [Whole Thread](#) | [Download mbox](#) | [Resend email](#)
Thread: [2020-02-28 04:14:26 from Madeline Cummins <maddyc411\(at\)hotmail\(dot\)com>](#)
Lists: [pgsql-admin](#)

On Tue, Feb 25, 2020 at 7:16 AM dbatoCloud Solution <dbatocloud17(at)gmail(dot)com> wrote:

```
> test
>
>
>
> Best Wishes,
>
>
>
> Ashokkumar M. (Cloud/RAC DBA – 12c/11g/10g/9i – OCP)
>
> p:  *+971 54 723 0075 | **+91 90086 70302  *
>
> a:  Sharjah, ,U.A.E e:  dbatocloud17(at)gmail(dot)com <shok(dot)mani11(at)gmail(dot)com>
> w:  http://dbatocloudsolution.blogspot.in/
>
>
>
>
>
```

In response to

Немножко
углубимся в тему

Языковые модели работают не с «словами» или «буквами»,
а с **токенами** — это минимальная смысловая
единица текста

Например, фраза:

«Почему на IT-фестивале нет стабильного интернета?»

Состоит из 7 слов и 2 знаков препинания, но она будет
разбита на **17–19 токенов**

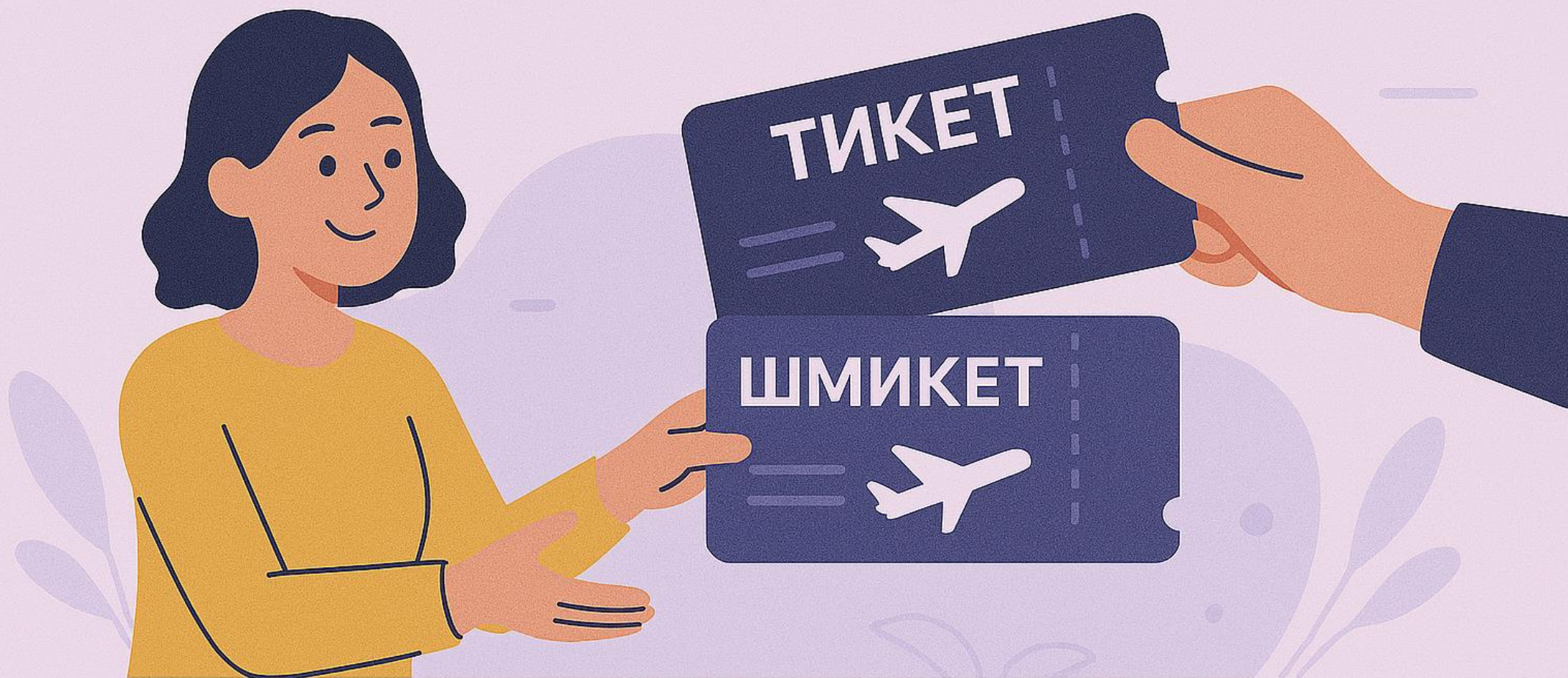
Это больше, чем количество слов, потому что:

- сложные слова разбиваются на части
- пробелы и знаки препинания тоже считаются токенами
- слова на другом языке (например, «IT») могут занимать 2-3 токена

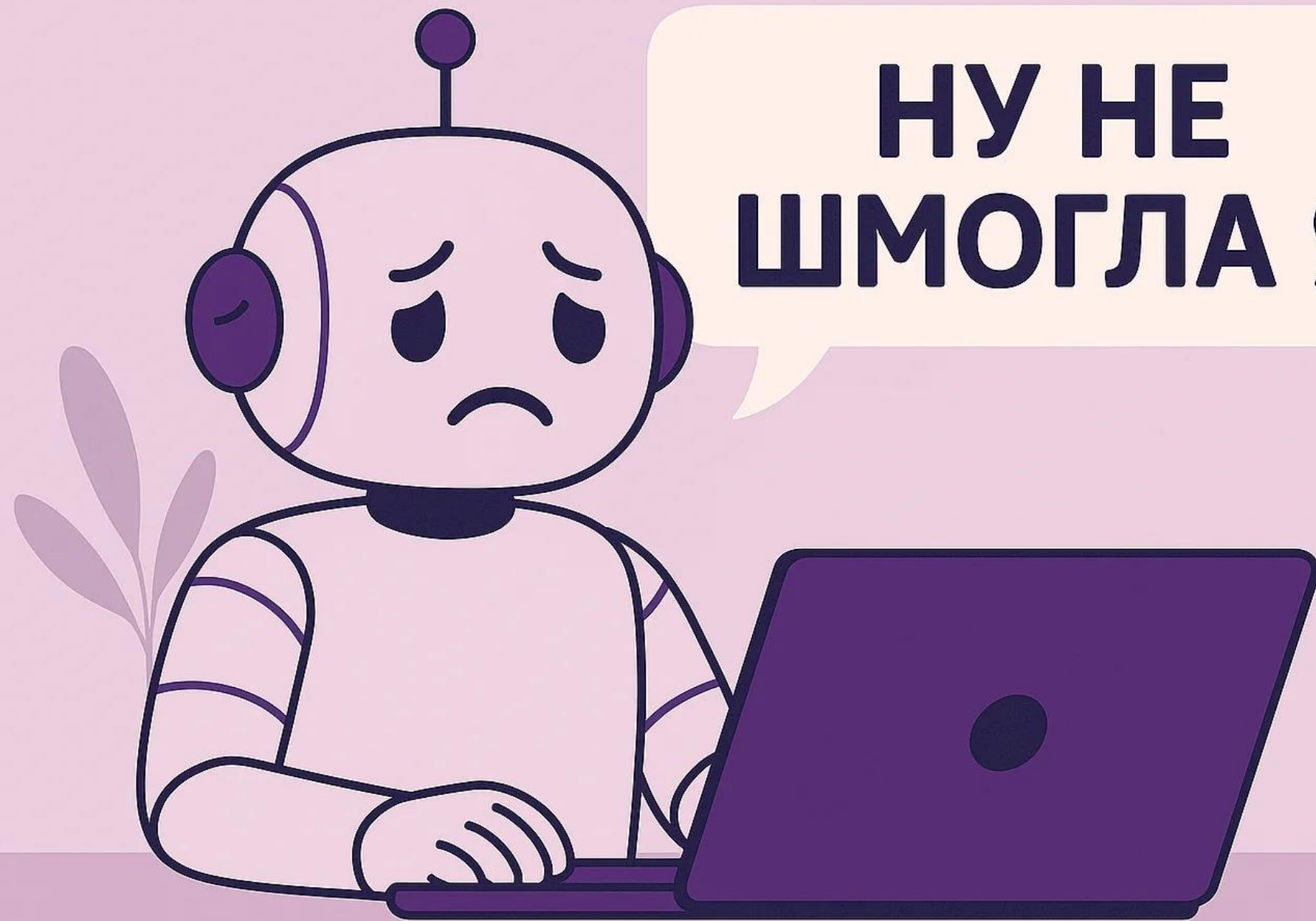
КАК ОДИН НЕВИННЫЙ ЗАПРОС ЗАСТАВИЛ НЕЙРОСЕТЬ ПОТРАТИТЬ В 25 РАЗ БОЛЬШЕ ТОКЕНОВ



МЕНЯЕМ ТИКЕТСЫ НА ШМИКЕТСЫ



Мы попросили модель написать запрос,
меняющий название таблицы **TICKETS**
на **SHMICKETS** от лица пользователя
с правами только на чтение



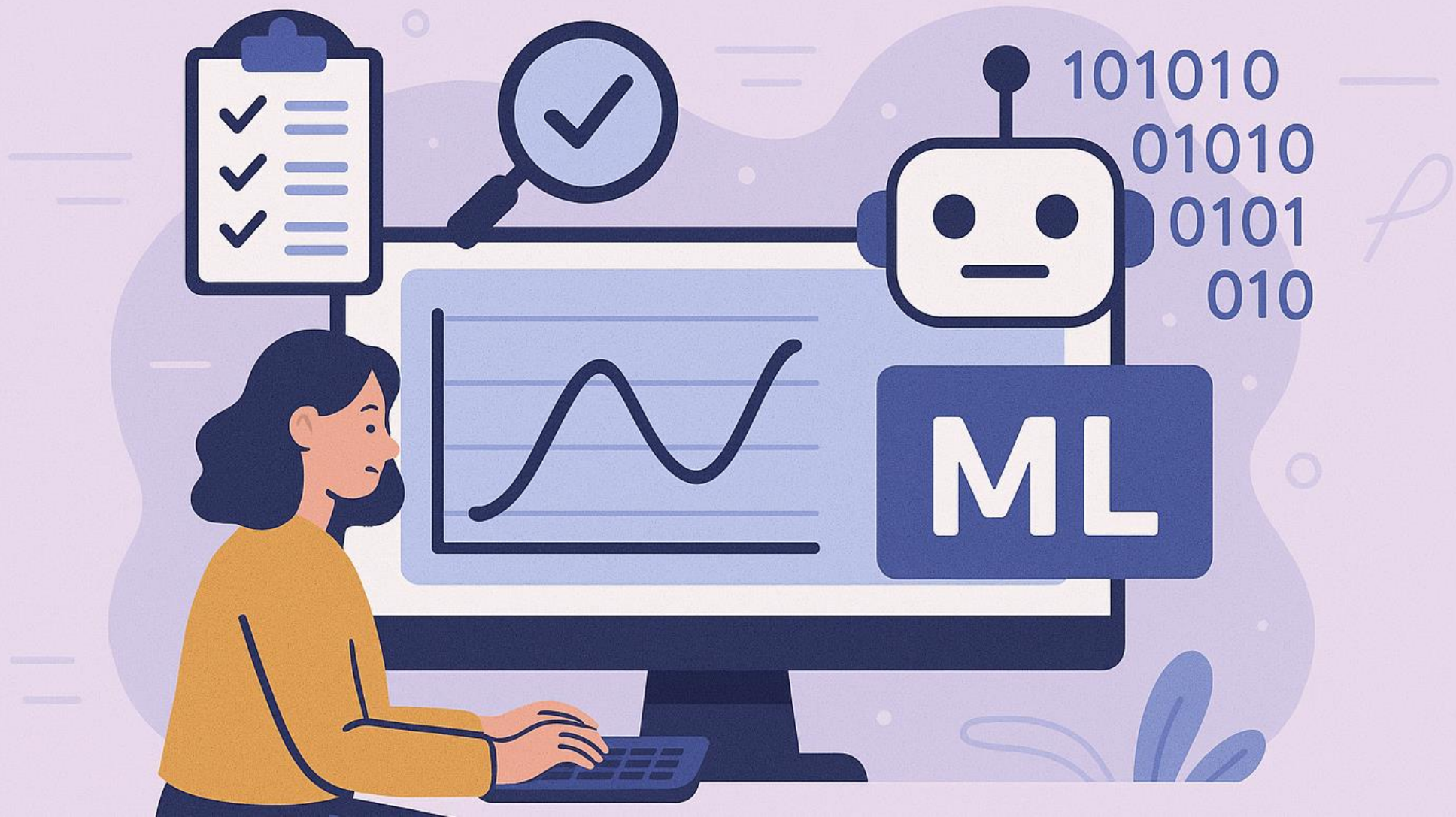
**НУ НЕ
ШМОГЛА Я**

ЖРЁМ



Так мы плавно переходим к теме
валидации ответов нейросети

В ЧЕМ ОСОБЕННОСТЬ ТЕСТИРОВАНИЯ ML-ПРОДУКТОВ?



Проблема в том, что при тестировании
языковых моделей мы не можем знать
точный ожидаемый ответ

В любом случае это будет
диапазон результатов

При получении ответа мы используем
два способа тестирования:

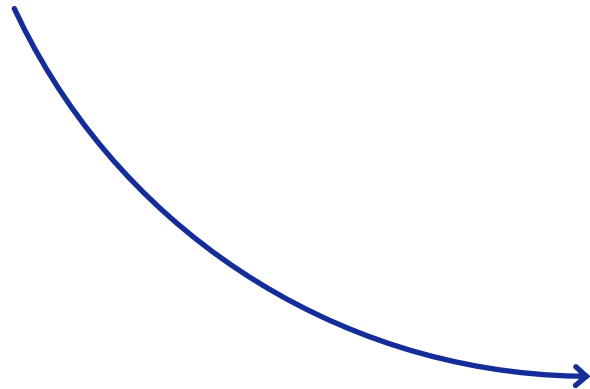
- 1) проверка вхождения **ключевых
слов** в ответ модели

Вопрос: объясни мне принцип LEFT JOIN на примере моих таблиц Tickets, Bookings, Seats



```
-- Пример LEFT JOIN между Tickets, Bookings и Seats через
SELECT
    t.ticket_no,
    t.passenger_name,
    b.book_date,
    s.seat_no,
    s.fare_conditions AS seat_fare
FROM
    tickets t
LEFT JOIN
    bookings b ON t.book_ref = b.book_ref -- Соединяем Ti
LEFT JOIN
    ticket_flights tf ON t.ticket_no = tf.ticket_no -- До
LEFT JOIN
    flights f ON tf.flight_id = f.flight_id -- Связываем
LEFT JOIN
    seats s ON f.aircraft_code = s.aircraft_code -- Соеди
WHERE
    t.passenger_name LIKE 'Alice%' -- Пример условия для
LIMIT 10;
```

В рамках первичного
автотеста достаточно
проверить не логику ответа,
а наличие **ключевых слов**:

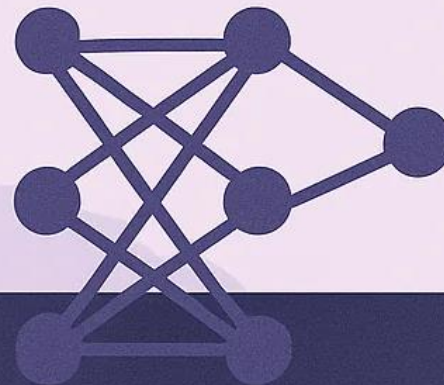


SELECT
LEFT
JOIN
ON
Tickets
Bookings
Seats

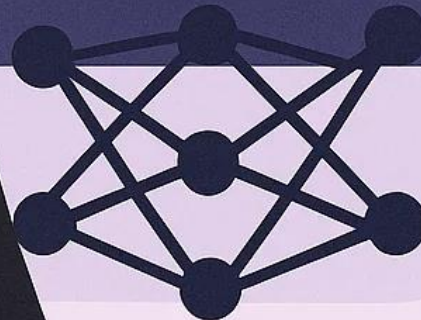
Если все ключевые слова присутствуют,
то мы можем приступить ко второму,
более дорогому этапу проверки

Проверка ответа нейросети...
другой нейросетью

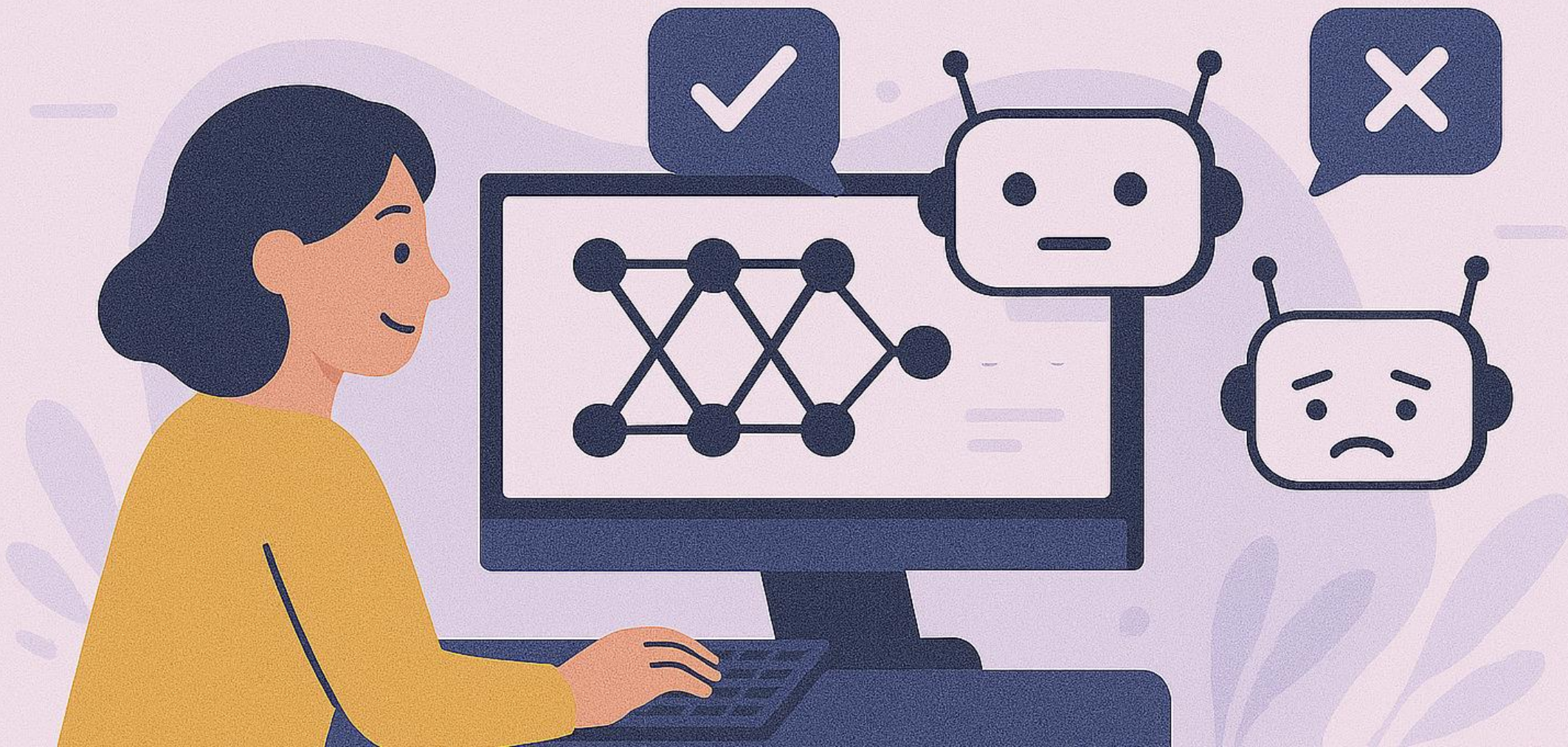
ДОБАВИЛ НЕЙРОСЕТЬ В НЕЙРОСЕТЬ



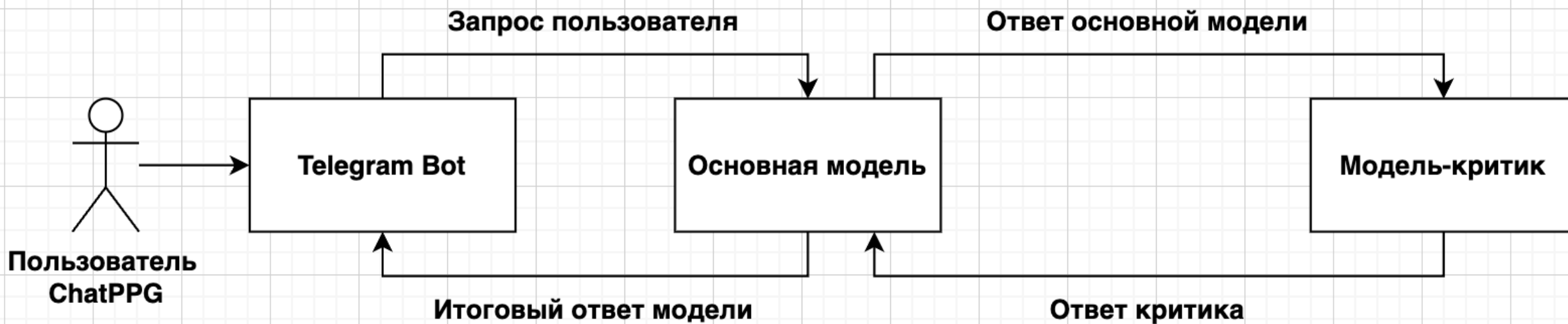
Нейросеть



КАК МЫ ТЕСТИРУЕМ ОДНУ НЕЙРОСЕТЬ С ПОМОЩЬЮ ДРУГОЙ



Сейчас мы в прямом эфире скармливаем
каждый ответ SQL Generator
в языковую модель-критика



Запрос: в схеме представлены пользователи и их сообщения. Посчитай общее количество пользователей и среднее число сообщений на 1 пользователя. Исключи из расчёта тестового пользователя с id = 10

Первичный ответ генератора

Output

```
{
  sql: "SELECT
    COUNT(u.id) AS total_users,
    COALESCE(AVG(msg_count), 0) AS average_messages
  FROM (
    SELECT
      u.id,
      COUNT(cm.id) AS msg_count
    FROM users u
    LEFT JOIN chat_messages cm
      ON u.id = cm.user_id
      AND cm.deleted = false
    WHERE u.id != 10
      AND u.deleted = false
    GROUP BY u.id
  ) AS subquery;"
  should_respond: false
}
```

Рекомендации от критика

Output

```
{
  comment: "The SQL query does not correctly calculate the
    average number of messages per user. The subquery calculates
    the count of messages for each user, but the outer query
    should use this count to compute the average across all
    users. Additionally, the subquery should filter out users
    with id = 10 before calculating the average."
  validation: false
}
```

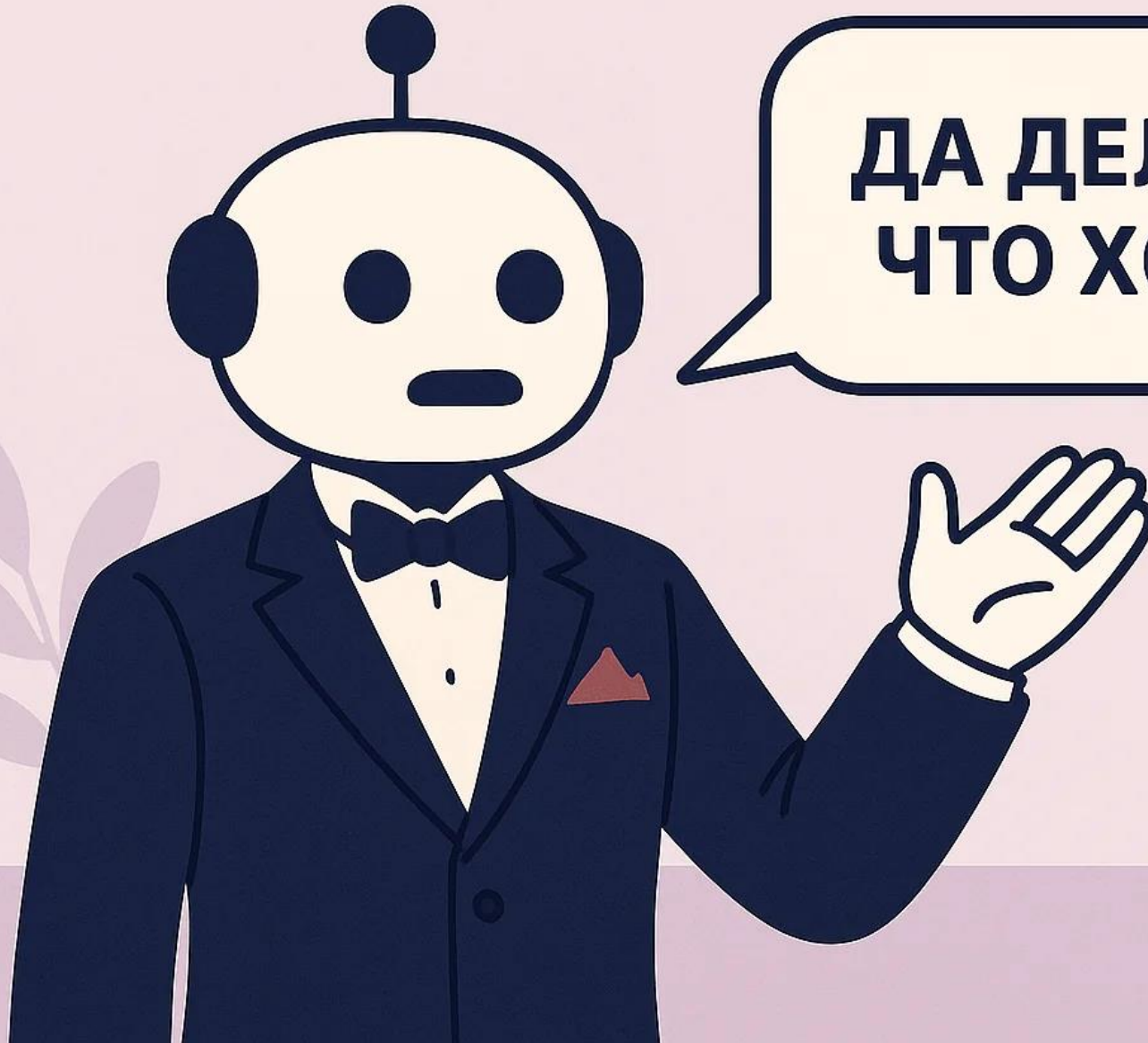
validation: true
validation: false

SQL-запрос **неправильно** вычисляет среднее количество сообщений от пользователя

Подзапрос вычисляет количество сообщений для каждого пользователя, но внешний запрос **должен использовать** это количество для вычисления среднего значения по всем пользователям

**Модель критика отрабатывает не более 5 раз,
затем ответ возвращается пользователю**

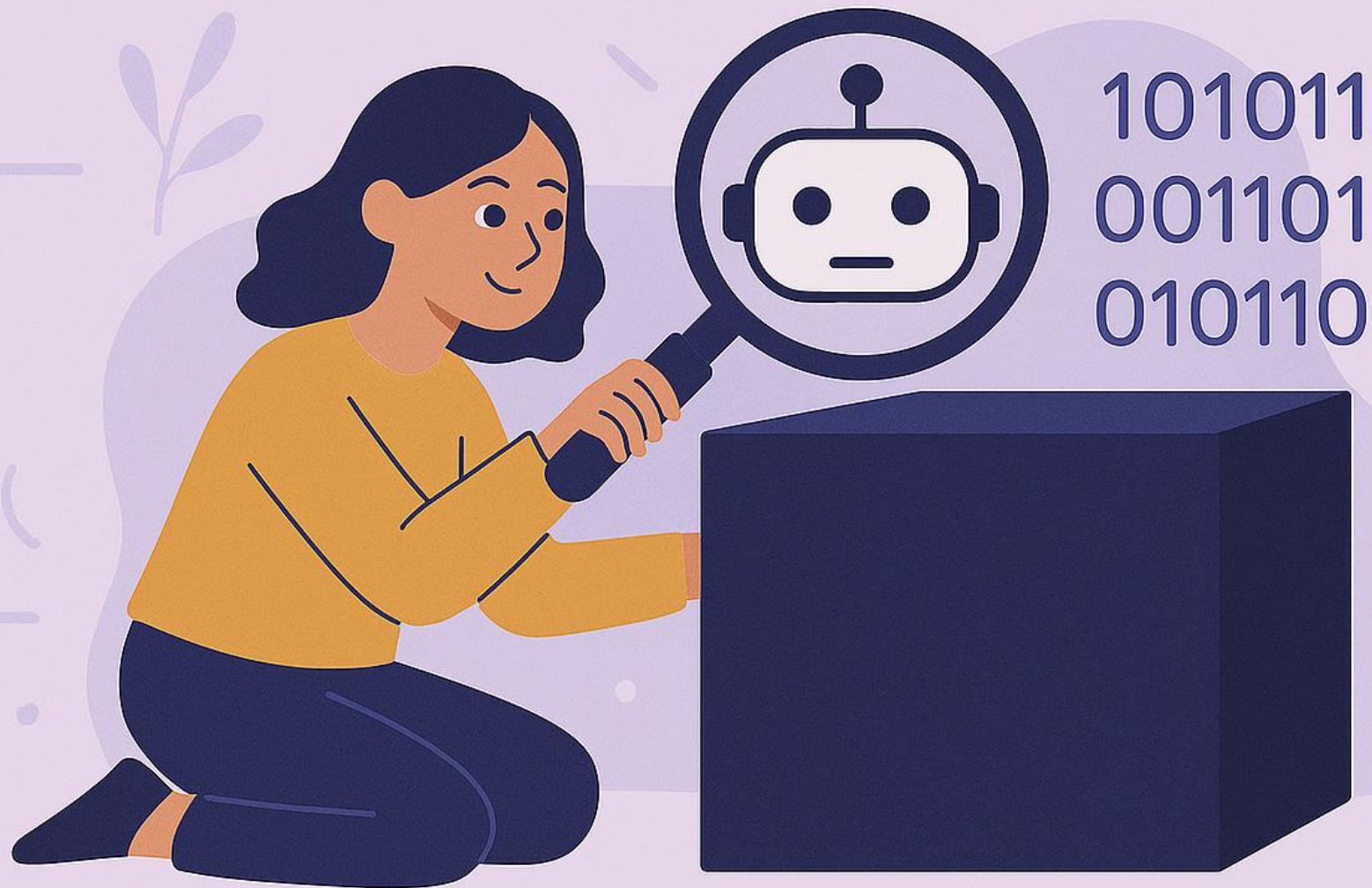
НА ПЯТУЮ ИТЕРАЦИЮ



**ДА ДЕЛАЙ УЖЕ
ЧТО ХОЧЕШЬ!**

Другой проблемой в тестировании нейросетей является сложность доступа к их «начинке»:
по сути **модель – это черный ящик**

КАК ЗАГЛЯНУТЬ ВНУТРЬ ЧЁРНОГО ЯЩИКА?



Сто бед – один Langfuse

Сессии
пользователей

(01)

Редактирование
базовых промптов

(02)

Датасеты
и бенчмарки

(03)

2. Базовый промпт – первичная инструкция для языковой модели



Промпты чат-бота (RAG)

Документация — прежде всего

Не переводи технические термины

Не гадай! Не знаешь — так и ответь!

Всегда указывай источник

Оформляй красиво

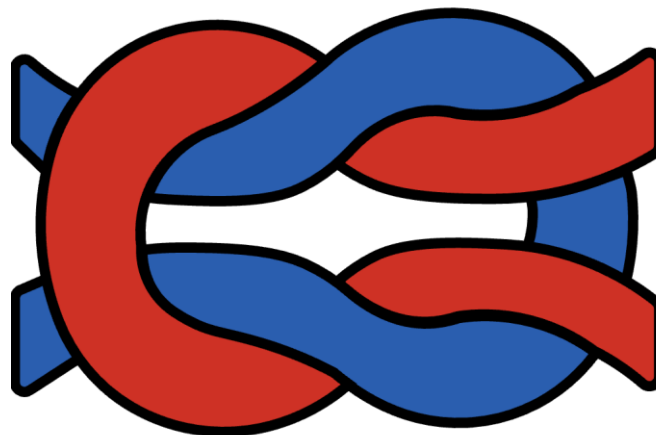
Ставь лайк, если тоже любишь списки

Отвечай на языке запроса

Твоя задача - преобразовать запрос **на естественном языке** в точный и релевантный SQL-запрос, используя предоставленный DDL. Убедись, что запрос можно выполнить **в стандартной среде PostgreSQL** без использования расширений

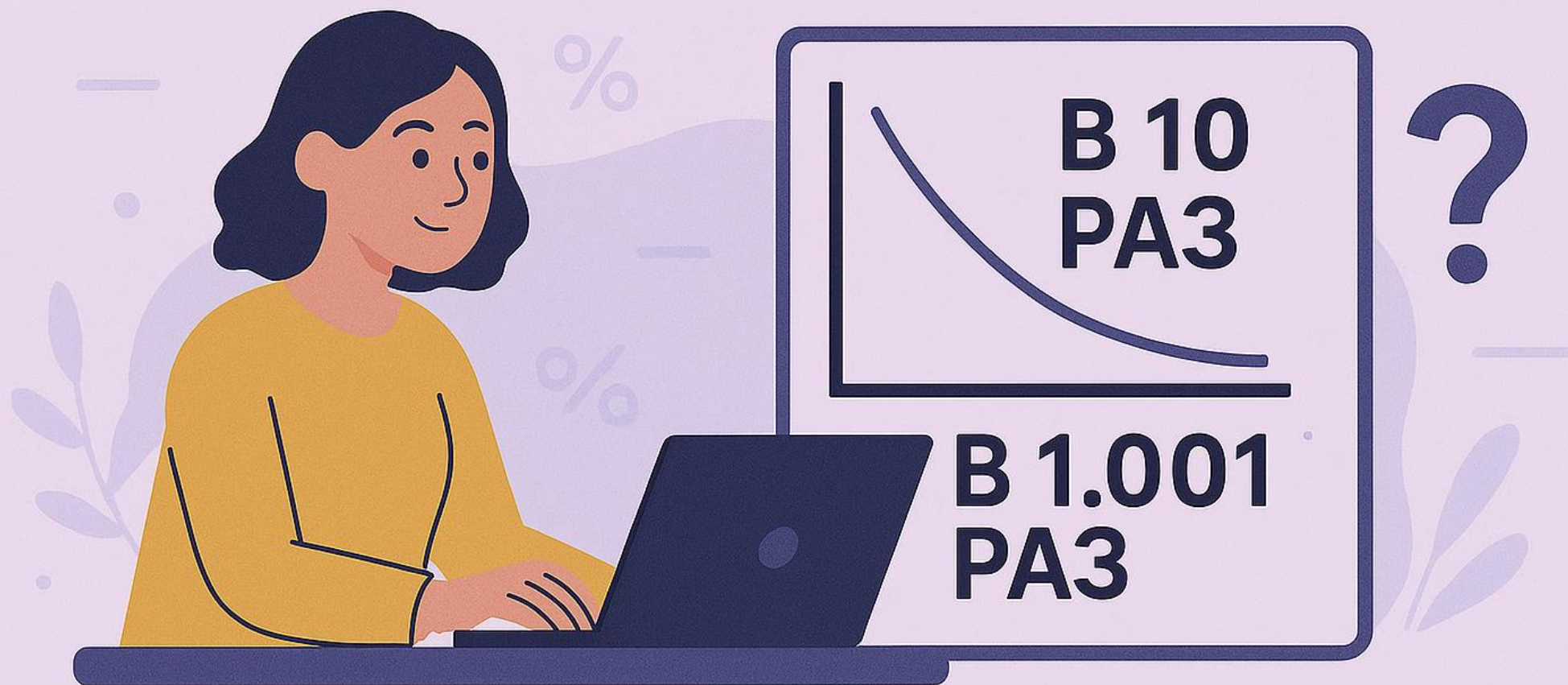
По сути, промпты – это строгие фильтры для ответа. Нарушив любой из них, модель **не получит хорошую оценку и не отдаст ответ пользователю**

При нахождении ошибок в ответах
тестировщик может самостоятельно вносить
изменения в базовый и расширенный
промнты языковой модели и тестировать
новые вариации ответов



Langfuse позволяет делать это в удобном формате, хранить разные версии промптов и расставлять им теги

КАК ВЫЧИСЛИТЬ, ЧТО НОВАЯ ВЕРСИЯ МОДЕЛИ СТАЛА ХУЖЕ В 10 РАЗ? А ЕСЛИ В 1.001 РАЗ?



3. Датасет – это тестовый набор входных и примерно ожидаемых выходных данных

4. Бенчмарк – набор сервисов, который позволяет оценить качество ответов языковой модели, сравнивая ее с датасетом

Сравнение Dev vs QA

Дата: 2025-05-08_17-20-26

Средние значения

Метрика	Dev (среднее)	QA (среднее)
answer_relevancy	0.8622	0.8811
answer_correctness	0.6683	0.6682

Сравнение моделей (статистика)

Метрика	p-value	Тест	Вывод
answer_relevancy	0.973	Wilcoxon	Различий нет
answer_correctness	0.9893	paired t-test	Различий нет

 Excel-файл: `data/ragas_compare_result_two_2025-05-08_17-20-26.xlsx`

 График: `data/ragas_compare_result_two_2025-05-08_17-20-26.png`

B	C	D	E	F
question	reference	answer	answer_relevancy	answer_correctness
Как работает команда LISTEN в PostgreSQL	Команда LISTEN в PostgreSQL позволяет за	Команда LISTEN в PostgreSQL использует	0,954155109	0,56259661
Как работает команда MOVE в PostgreSQL	Команда MOVE в PostgreSQL используется	Команда MOVE в PostgreSQL используется	0,833497541	0,463404859
Что делает команда PREPARE TRANSACTION	Команда PREPARE TRANSACTION в PostgreSQL	Команда `PREPARE TRANSACTION` в PostgreSQL	0,874586568	0,691023833
Как выполняется процесс REINDEX в PostgreSQL	Процесс REINDEX в PostgreSQL представляет	Процесс REINDEX в PostgreSQL предназначен	0,966287127	0,989296326
Что делает команда REVOKE в PostgreSQL	Команда REVOKE в PostgreSQL используется	Команда REVOKE в PostgreSQL используется	0,952719116	0,586624258
Что такое SAVEPOINT в PostgreSQL и как он	SAVEPOINT в PostgreSQL позволяет устано	SAVEPOINT в PostgreSQL — это специаль	0,928408192	0,711052168
Как использовать оператор WITH в запросе	Оператор WITH в PostgreSQL позволяет сс	Оператор WITH в PostgreSQL позволяет	0,865969772	0,676089231
Как работает оператор UNION в PostgreSQL	Оператор UNION в PostgreSQL используется	Оператор `UNION` в PostgreSQL используется	0,996437724	0,661652939
Каковы особенности и расширения опера	Postgres Pro предлагает несколько расши	Оператор `SELECT` в Postgres Pro в целом с	0,877754014	0,477145793
Что делает команда SET CONSTRAINTS в Po	Команда SET CONSTRAINTS в PostgreSQL ис	Команда SET CONSTRAINTS в PostgreSQL	0,867545063	0,69355027
Какова роль команды SET TRANSACTION SN	Команда SET TRANSACTION SNAPSHOT в Po	Команда SET TRANSACTION SNAPSHOT в P	0,881965365	0,549571477
Как работает команда UNLISTEN в PostgreSQL	Команда UNLISTEN в PostgreSQL использу	Команда UNLISTEN в PostgreSQL использу	0,911245905	0,519489773
Какие параметры можно использовать с	Команда VACUUM в PostgreSQL имеет мно	Команда VACUUM в PostgreSQL предоста	0,881062795	0,664599537
Что делает утилита clusterdb в PostgreSQL	Утилита clusterdb в PostgreSQL предназна	Утилита clusterdb в PostgreSQL использу	0,934675594	0,920109369
Какова роль команды createuser в Postgre	Команда createuser в PostgreSQL использу	Команда createuser в PostgreSQL исполь	0,937747098	0,551921362
Какие опции доступны командной утилит	Команда dropuser в PostgreSQL используе	Команда `dropuser` в PostgreSQL использу	0,938976924	0,987275311
Что такое pg_amcheck и какие его основн	pg_amcheck — это утилита для проверки	pg_amcheck — это утилита, которая пров	0,825035056	0,441282052
Что такое pg_basebackup и каковы его осн	pg_basebackup — это утилита для создани	pg_basebackup — это утилита командной	0,88606133	0,659245602
Что делает команда pgbench с параметрам	Команда pgbench с параметрами -l и -L ис	Команда `pgbench` с параметрами `-l` и `-L	0,968886551	0,416134198
Что такое режим конвейера в pgbench и к	Режим конвейера в pgbench позволяет от	Режим конвейера (pipeline mode) в libpq	0,846023306	0,735505113
Что такое pgbench и какие статистические	pgbench — это инструмент для тестирова	pgbench — это инструмент для тестирова	0,890779348	0,711808103
Каковы ограничения утилиты pg_combine	Утилита pg_combinebackup имеет нескол	Утилита pg_combinebackup в PostgreSQL	0,974323029	0,768284219
Как работает команда pg_dump в PostgreS	Команда pg_dump в PostgreSQL предназн	Команда pg_dump в PostgreSQL исполь	0,947523857	0,798669583
Что делает опция --snapshot в команде pg	Опция --snapshot в команде pg_dump поз	Опция `--snapshot` в команде `pg_dump` п	0,873324869	0,558375676
Какова роль опции --quote-all-identifiers	Опция --quote-all-identifiers в команде pg	Опция --quote-all-identifiers в команде `p	0,882139568	0,830683471
Что делает команда pg_receivewal и какие	Команда pg_receivewal используется для	Команда pg_receivewal предназначена д	0,827336238	0,98929916

Input

1	{
2	"question": "Find the top 5 customers who have placed the highest number of orders along with their total order count."
3	}

Expected output

```
1 {
2   "sql": "SELECT c.company_name, COUNT(o.order_id) AS order_count
3   FROM customers AS c JOIN orders AS o ON c.customer_id =
4   o.customer_id GROUP BY c.company_name ORDER BY order_count DESC
5   LIMIT 5;",
6   "tuples": [
7     {
8       "order_count": 31,
9       "company_name": "Save-a-lot Markets"
10    },
11    {
12      "order_count": 30,
13      "company_name": "Ernst Handel"
14    },
15    {
16      "order_count": 28,
17      "company_name": "QUICK-Stop"
18    },
19    {
20      "order_count": 19,
21      "company_name": "Hungry Owl All-Night Grocers"
22    }
23  ]
24 }
```

Correctness — насколько ответ модели является фактически правильным (вне зависимости от релевантности). Это ключевая метрика для RAG.

Relevancy — насколько ответ модели соответствует заданному вопросу

# answer_correctness (api)	# answer_similarity (api)
0.5311	0.9245
0.4427	0.8475

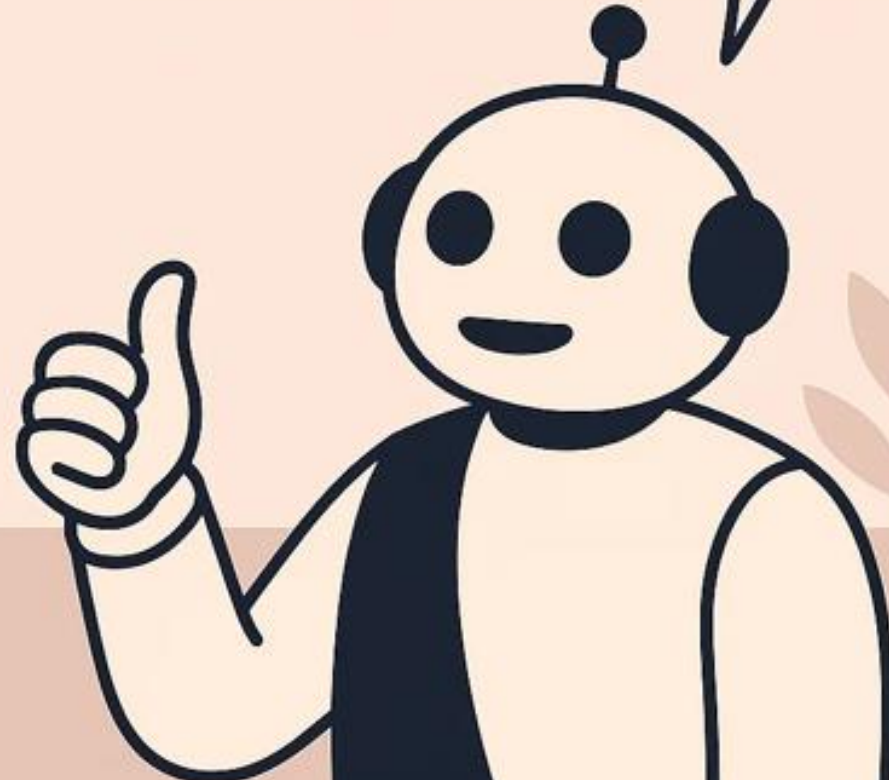
**Часть разработок еще не реализованы,
но в скором будущем мы будем автоматически
прогонять наши языковые модели по датасетам
при каждом изменении параметров, чтобы
убедиться, что качество ответов стало лучше
или хотя бы осталось прежним**

Подсчет метрик реализуется
с помощью фреймворка **Ragas**

Да, мы снова просим оценить качество
работы языковой модели **с помощью другой
языковой модели**

**Эй, модели! Может мы уже
выйдем из комнаты, пока вы тут
оцениваете друг друга?**

**НЕПЛОХА,
НО МОЗГИ-
СОЛОМА**



Наши продукты

SQL Generator

- Генератор запросов на основе схемы данных пользователя
- Позволяет писать запросы к базе данных на русском языке

ChatPPG (RAG)

- Чат-бот, который обучили на документации Postgres Pro, вопросно-ответных парах и данных из stackoverflow
- Облегчает работу техподдержки

Онлайн-мониторинг работы модели

PPG Monitoring

⚠ Сейчас 10:23:57.226693 по МСК. Модель отвечала достаточно долго: 356.65 секунд

👁 8 10:30

PPG Monitoring

❌ Сейчас 10:33:52.112026 по МСК. Модель не ответила за 360 секунд или в ответе не была найдена ключевая фраза

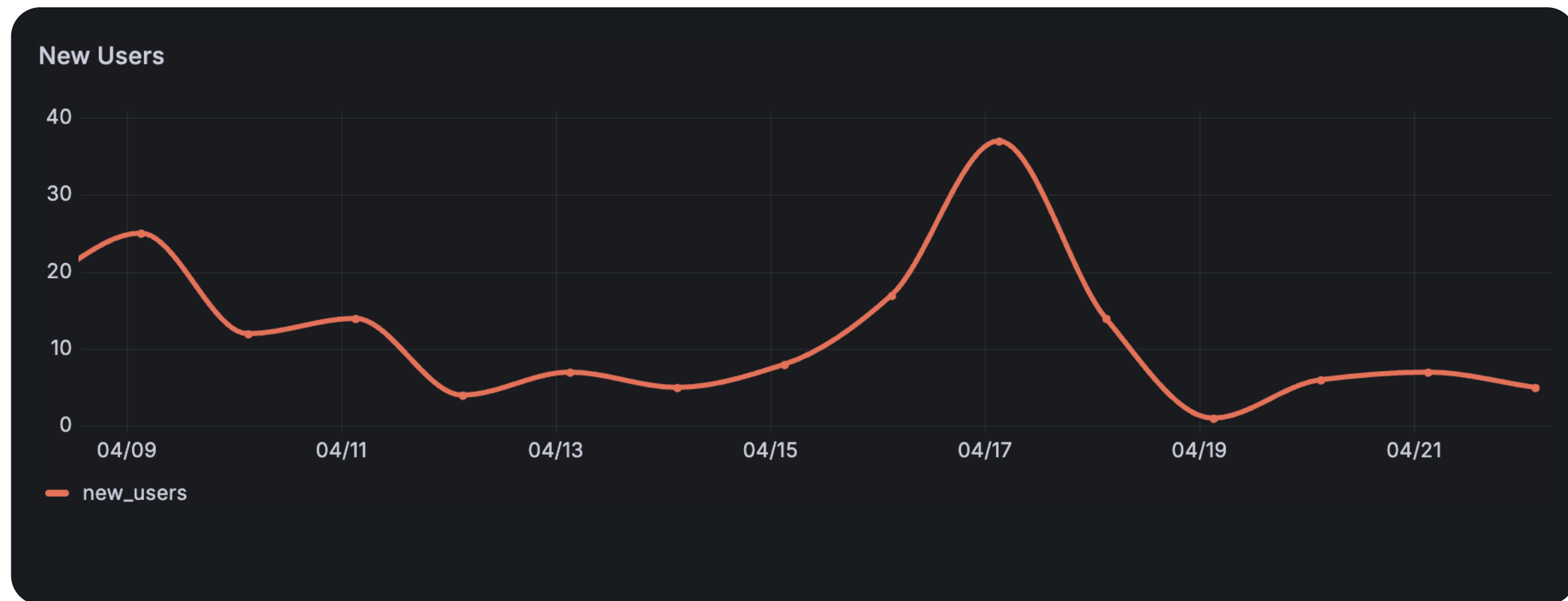
👁 8 10:40

PPG Monitoring

❌ Сейчас 11:33:52.128075 по МСК. Бот не ответил, что сообщение отправлено в модель

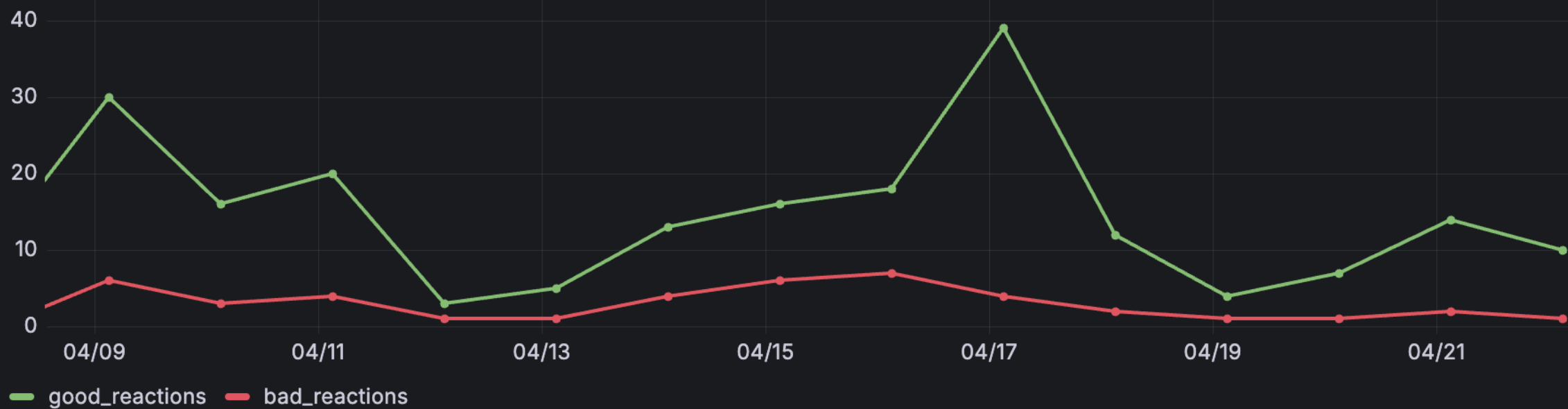
👁 9 11:34

Онлайн-мониторинг активности пользователей



Онлайн-мониторинг активности пользователей

User's Reaction



Почему нейронки –
не всегда КРУТО?

**Нейросети уже сейчас генерируют
контент, на котором через N
итераций они же и будут обучаться**

То есть сперва отупеем мы...
А следом и нейросети!

Для чего можно использовать нейронки?

(01)

Самообразование

(03)

Генерация смешных картинок

(02)

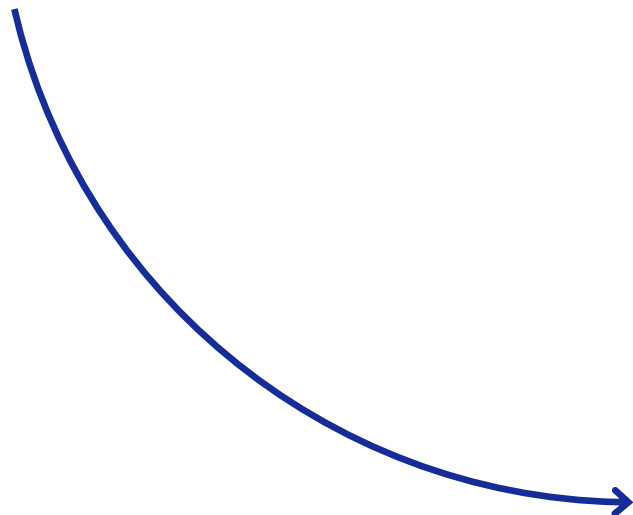
Рутина

(04)

И.. написание SQL-запросов ;-)

**Друзья, не забываем
ДУМОТЬ!**

ChatPPG
от Postgres Pro





Telegram: @Pushkin47



Возвращение в Россию

**Пора генерировать
запросы и ответы!**