

An automagical GraphQL to SQL compiler

## WRITING APIS TO TALK TO DATABASES IS 90% OF WEB APP BACKENDS

MOST DEVS.

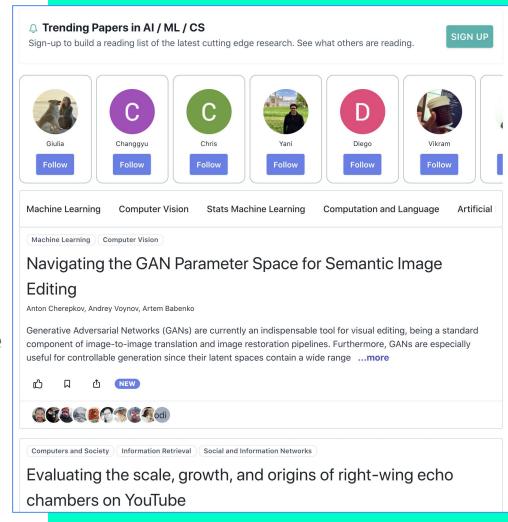
DON'T KNOW SQL

DON'T PLAN TO LEARN IT!

### **Always Evolving UIs**

UIs today are complex and fast evolving which require APIs to keep up.

https://42papers.com



### GRAPHQL GIVES FRONTEND DEVS. MORE POWER

### GRAPHQL GIVES FRONTEND DEVS. MORE POWER

#### **Build a Blog**

Fetch Posts, Votes on Posts, Authors, Comments, Comment Authors, ...

```
posts(first: 5, after: $cursor) {
 title
 body
 votes {
   User
 author {
   User
   body
   author {
     User
```

#### **Easy for Devs**

Kinda looks like JSON if you squint.

```
posts(first: 5, after: $cursor) {
       title
       body
       votes {
         User
         User
11 ▼
         body
           ...User
```

## COMPLEX NESTED GRAPHQL QUERIES COMPILE INTO A SINGLE EFFICIENT SQL QUERY.

NESTED
SELECTS, INSERTS, UPDATES
COMPILE TO
A SINGLE SQL QUERY

```
query {
 user : me {
  first name
   last name
   picture url
  thread(first: 20, after: $cursor, order by: { cached votes total: desc }) {
    cached posts total
    vote : thread vote(where: { user id: { eq: $user id } }) {
     created at
    topics {
    author : me {
    posts(first: 20, order_by: { cached_votes_total: desc }) {
      created at
      cached votes total
      vote : post_vote(where: { user_id: { eq: $user_id } }) {
       created at
      author : user {
        firstName : first name
        lastName : last name
```

```
...
SELECT
 jsonb_build_object('thread', "__sj_0".'json") as "__root"
FROM
    SELECT
      to_jsonb("__sr_0") A5 "json"
    FROM
        SELECT
          "threads_0"."slug" AS "slug",
          "threads B". "title" A5 "title".
          "threads_0"."published" A5 "published",
          'threads_0'."created_at' AS "createdAt",
          'threads 0'."cached votes total" AS 'totalVotes'.
          "threads_0"."cached_posts_total" AS "totalPosts",
          "__sj_1"."json" AS "posts",
          "_sj_1"."cursor" A5 "posts_cursor",
          " sj 4"."json" AS "author",
          '_sj_5"."json" AS "topics",
          "_sj_6". "json" AS "vote"
        FROM
            SELECT
              "threads"."slug".
              "threads", "title",
              "threads". "published",
              "threads"."created_at",
              "threads". "cached votes total",
              "threads", "cached posts total",
              "threads"."id",
              "threads". "user id".
              "threads"."topic_ids"
            FROM
              "threads"
            WHERE
                    ("threads"."published") = 'true' :: boolean
                    ("threads"."user_id") = '5' :: bigint
            LINIT
              ('1') :: integer
```

# APIS THAT TOOK WEEKS TO BUILD NOW TAKE 5 MINUTES

#### **Subscriptions**

Horizontally scaling, 100K subscribers with less than 5 queries per second.

```
customers(first: 5, after: $cursor) {
  email
  my_vote :vote(where: { user_id: $user_id }) {
    created_at
```

# ROLE BASED ACCESS CONTROL FULL TEXT SEARCH EFFICIENT CURSOR PAGINATION POLYMORPHIC RELATIONSHIPS

## STANDALONE SERVICE & GO LIBRARY

1K+ GITHUB STARS 25+ CONTRIBUTORS GROWING DISCORD USED IN PROD. GREAT DOCS.

THANKS

TWITTER.COM/DOSCO