Super Graph

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, ...
Easy for Devs

Kinda looks like JSON if you squint.
complex nested GraphQL queries can be compiled into a single efficient SQL query.
Nested Selects, Inserts, Updates compile to a single SQL query
query {
  user: me {
    slug
    firstName: first_name
    lastName: last_name
    picture_url
    bio
  }
  thread(first: 20, after: $cursor, order_by: { cached_votes_total: desc }) {
    slug
    title
    published
    cached_posts_total
    vote: thread_vote(where: { user_id: { eq: $user_id } }) {
      created_at
    }
    topics {
      slug
      name
    }
    author: me {
      slug
    }
    posts(first: 20, order_by: { cached_votes_total: desc }) {
      slug
      body
      published
      created_at
      cached_posts_total
      vote: post_vote(where: { user_id: { eq: $user_id } }) {
        created_at
      }
      author: user {
        slug
        firstName: first_name
        lastName: last_name
      }
    }
  }
}

SELECT
  jsonb_build_object('thread', "_s1", "json") as "_root"
FROM
  (SELECT
    to_jsonb("_s1") AS "json"
  FROM
    (SELECT
      'threads' AS "type",
      'threads.slug' AS "slug",
      'threads.title' AS "title",
      'threads.published' AS "published",
      'threads.created_at' AS "created_at",
      'threads.cached_votes_total' AS "totalVotes",
      'threads.cached_posts_total' AS "totalPosts",
     ...
    )
    AS "_s1"
  )
  AS "_s1"
  )
  AS "_s2"
  )
  AS "_s3"
  )
  AS "_s4"
  )
  AS "_s5"
  )
FROM
  "threads"
WHERE
  ("threads"."published") = 'true' :: boolean
  OR ("threads"."user_id") = 5 :: bigint
LIMIT (1) :: integer
AS "threads_0"
APIs that took Weeks to build now take 5 MINUTES
Subscriptions

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

```graphql
subscription {
  customers(first: 5, after: $cursor) {
    id
    email
    products {
      name
      id
      price
    }
    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