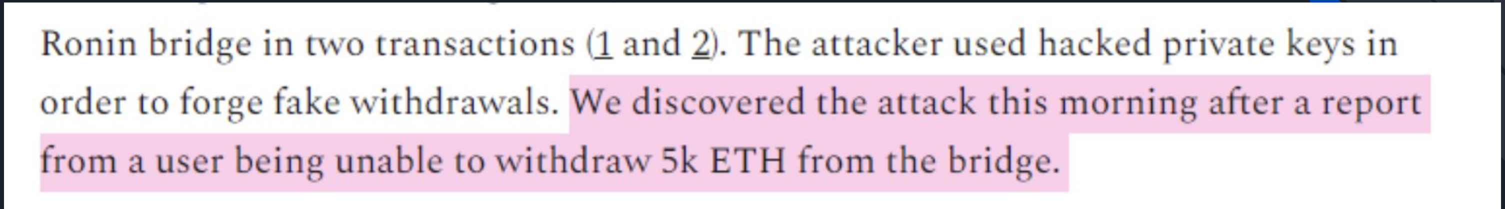
Good Solidity practices from a Data Analyst Perspective

by **wei3erHase**

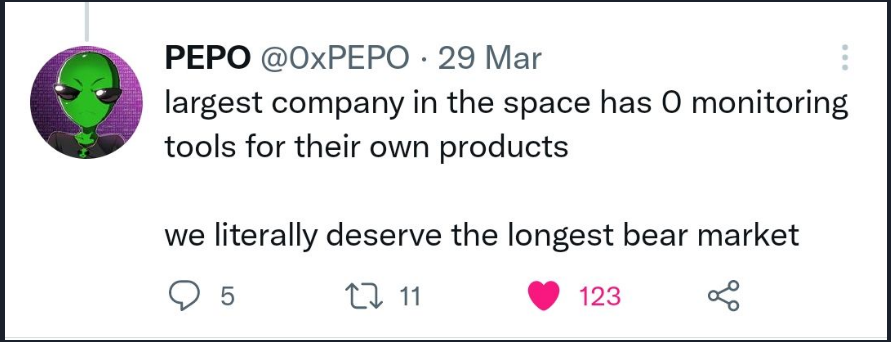
$600M

March 2022: Ronin was hacked for $600M in assets.

The hack was discovered after a user tried to withdraw ~$150k from the bridge (6 days later).



Ronin bridge in two transactions (1 and 2). The attacker used hacked private keys in order to forge fake withdrawals. We discovered the attack this morning after a report from a user being unable to withdraw 5k ETH from the bridge.

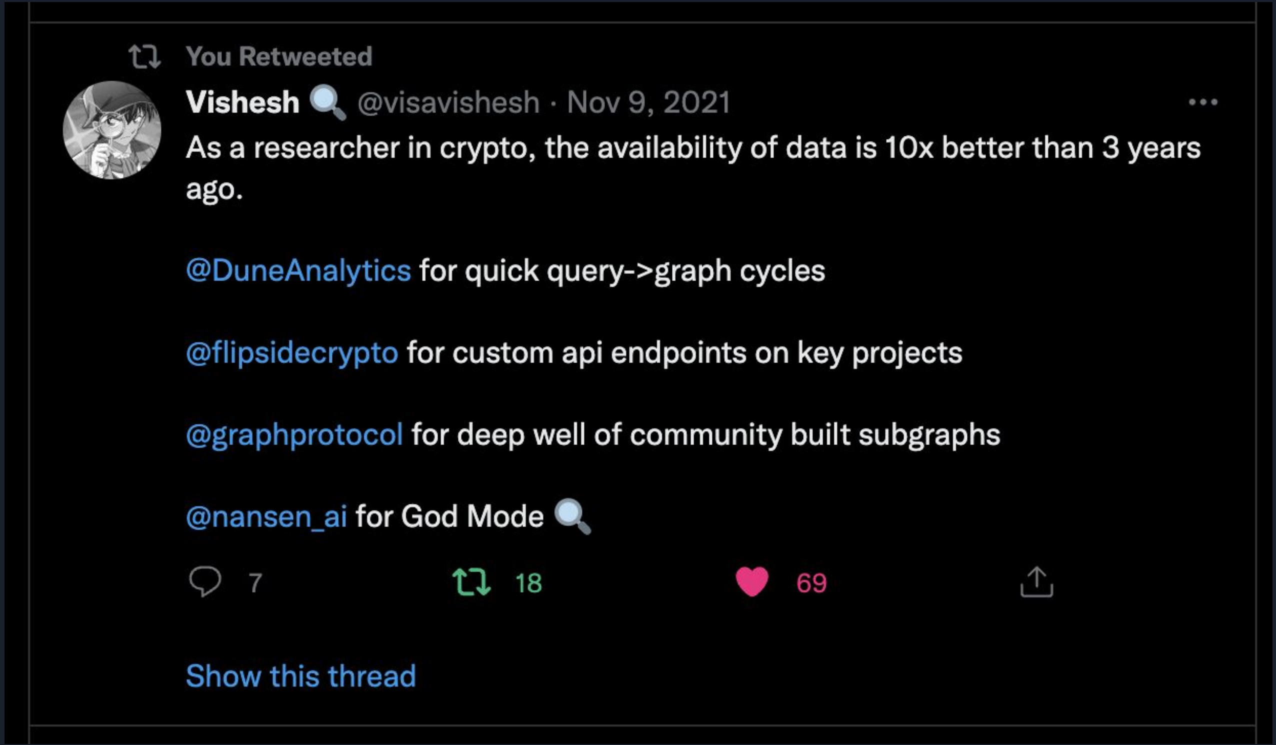


PEPO @0xPEPO · 29 Mar

largest company in the space has 0 monitoring tools for their own products

we literally deserve the longest bear market

Sad, but true.



You Retweeted

Vishesh 🔍 @visavishesh · Nov 9, 2021

As a researcher in crypto, the availability of data is 10x better than 3 years ago.

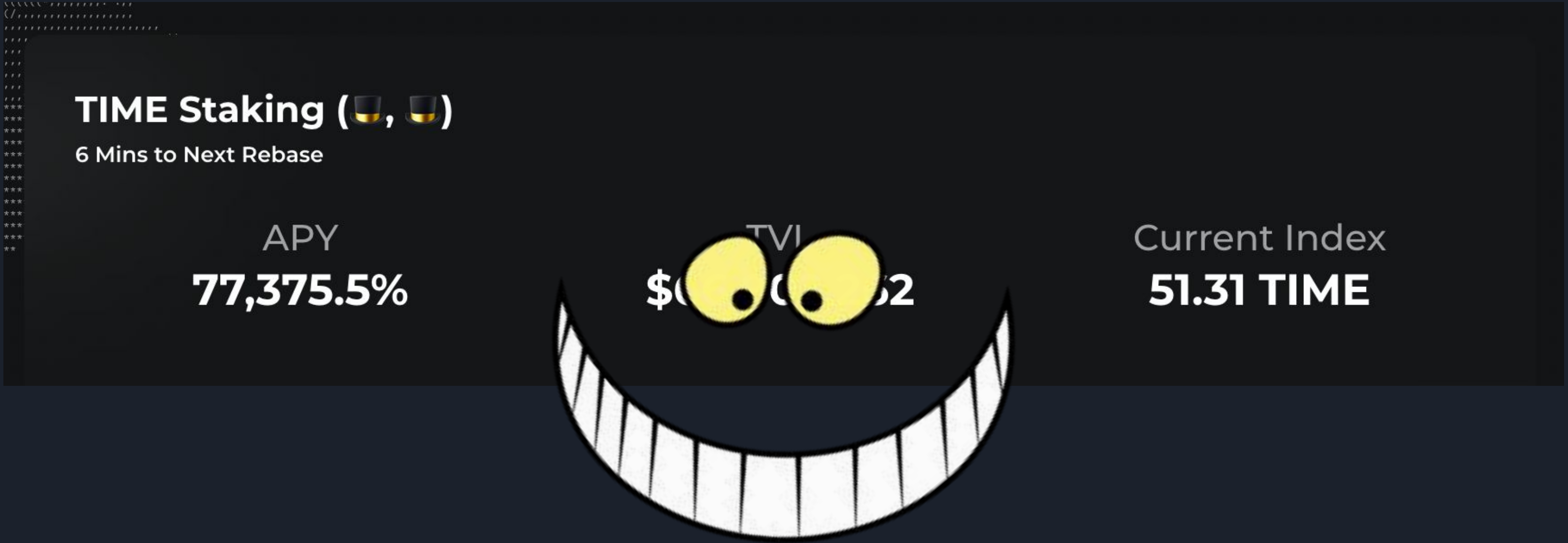
@DuneAnalytics for quick query->graph cycles

@flipsidecrypto for custom api endpoints on key projects

@graphprotocol for deep well of community built subgraphs

@nansen_ai for God Mode 🔍

Promising…



TIME Staking (🎩, 🎩)

6 Mins to Next Rebase

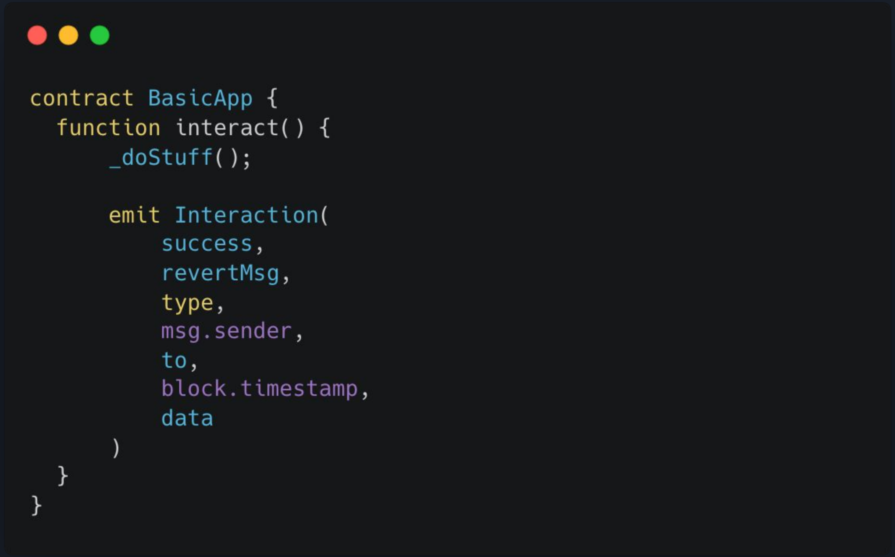
APY
77,375.5%

TVL
$[illegible]52

Current Index
51.31 TIME

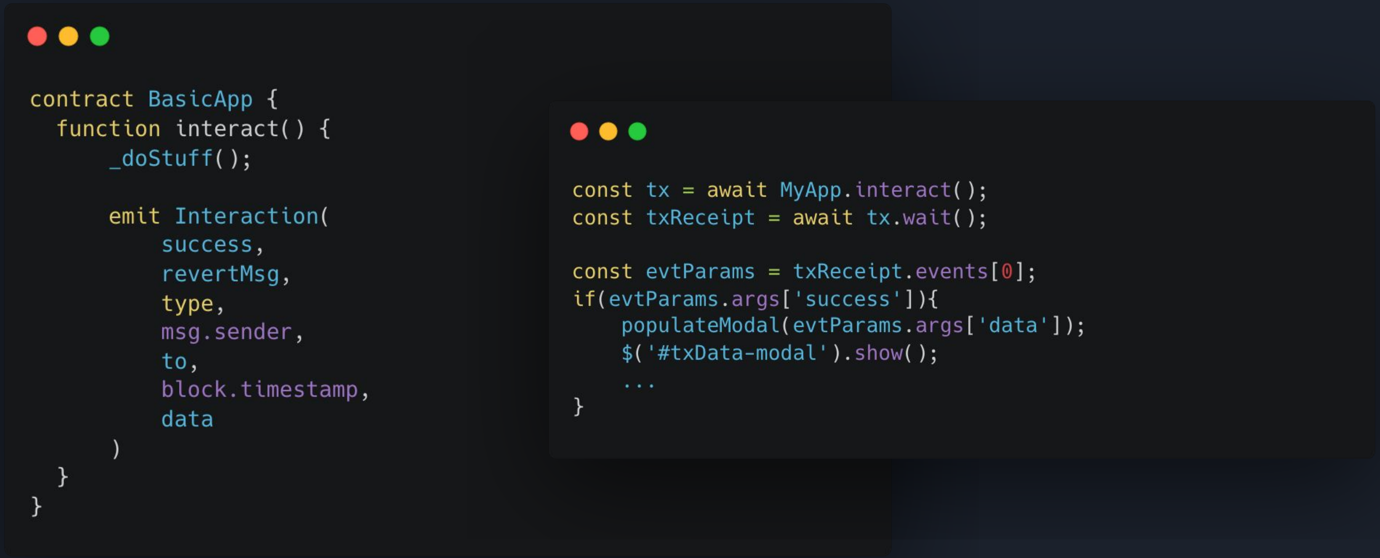
Happy, but false.

Early stages of events implementation



```
contract BasicApp {
  function interact() {
      _doStuff();

      emit Interaction(
          success,
          revertMsg,
          type,
          msg.sender,
          to,
          block.timestamp,
          data
      )
  }
}
```

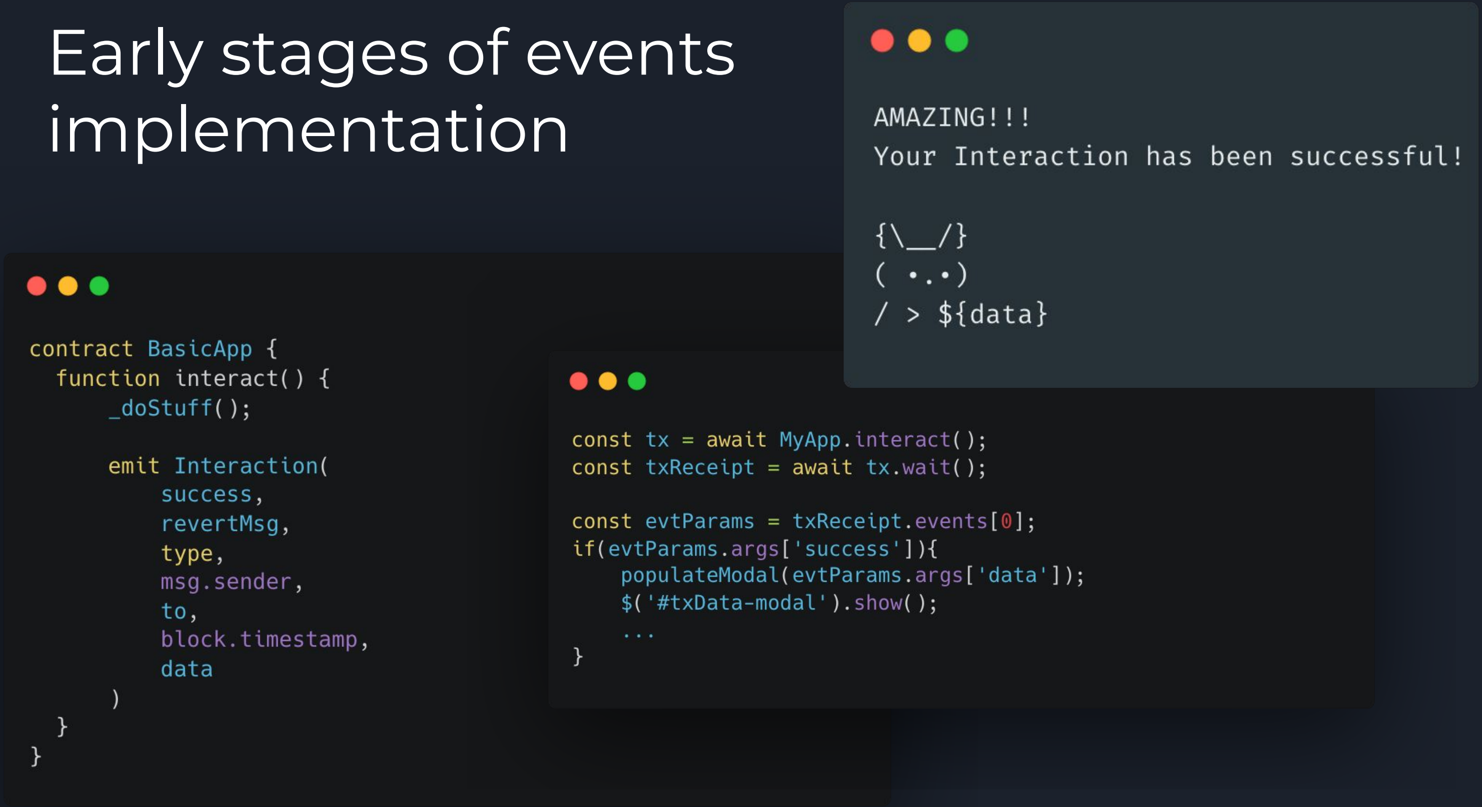
Early stages of events implementation



```
contract BasicApp {
  function interact() {
      _doStuff();

      emit Interaction(
          success,
          revertMsg,
          type,
          msg.sender,
          to,
          block.timestamp,
          data
      )
  }
}
```

```
const tx = await MyApp.interact();
const txReceipt = await tx.wait();

const evtParams = txReceipt.events[0];
if(evtParams.args['success']){
    populateModal(evtParams.args['data']);
    $('#txData-modal').show();
    ...
}
```

Early stages of events implementation



```
contract BasicApp {
  function interact() {
      _doStuff();

      emit Interaction(
          success,
          revertMsg,
          type,
          msg.sender,
          to,
          block.timestamp,
          data
      )
  }
}
```

```
const tx = await MyApp.interact();
const txReceipt = await tx.wait();

const evtParams = txReceipt.events[0];
if(evtParams.args['success']){
    populateModal(evtParams.args['data']);
    $('#txData-modal').show();
    ...
}
```

```
AMAZING!!!
Your Interaction has been successful!

{\__/}
( •.•)
/ > ${data}
```

Beyond the RPC

Using native RPC methods

- Relies on RPC for querying events
- Process each event parameter individually
- Archive nodes are expensive
- Hard time processing huge loads of data

- Requires less infrastructure
  (can be used on testnets)

Modern Data Sources

- Store filtered historical data
- Pre-processes events into tables or graphs
- Easy query between contracts
- Diverse pricing schemes
- Designed to process data

- Requires an ETL infrastructure

Current Data Sources

Centralized DBs
Community driven ETL
Embeddable [HTML]
Freemium service
[PostgreSQL]

Centralized DBs
Ethereum Airflow ETL
Open API and ML/AI tools
Pay as you Query
[BigQuery SQL]

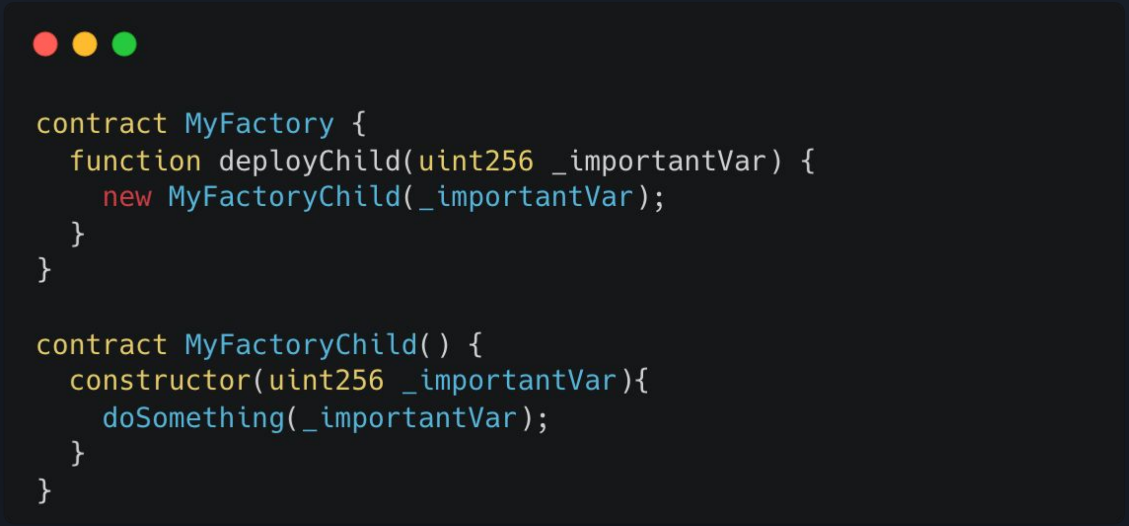
Distributed APIs
User designed schemas
Composable by design
Incentives mechanism
[GraphQL]

Demo

RPC event querying vs Graph querying

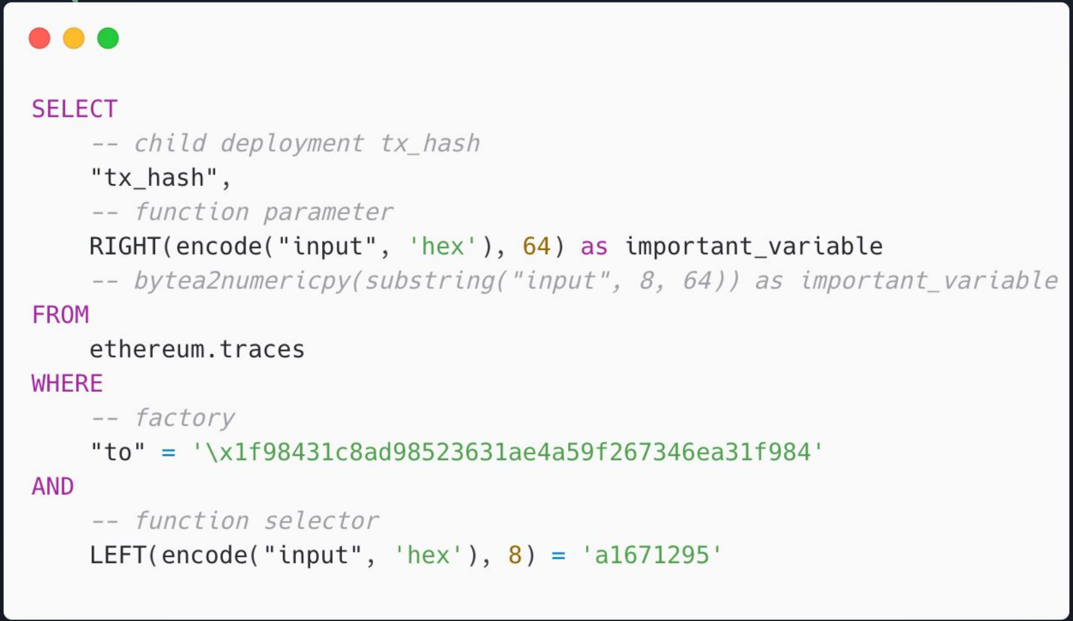
The Data Analyst Onboarding Week

The Return of the Prodigal Son



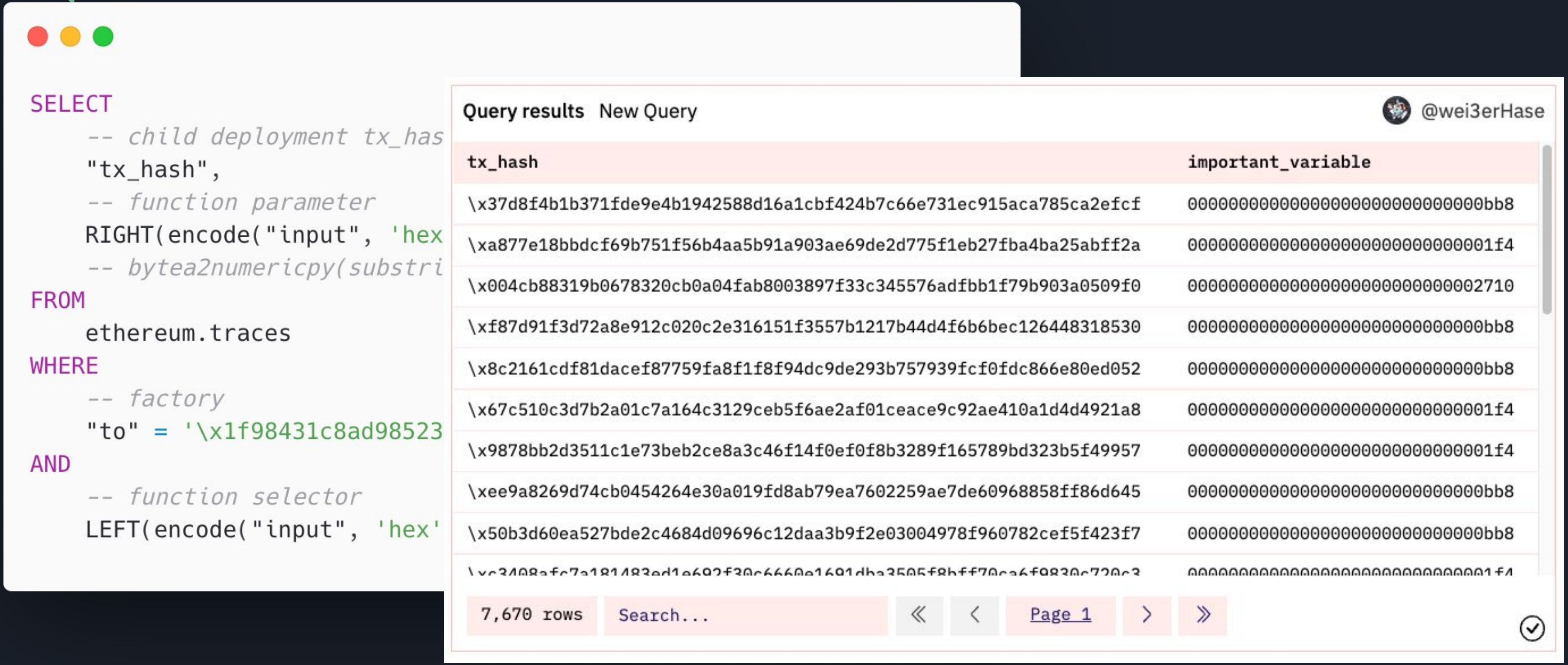
```
contract MyFactory {
  function deployChild(uint256 _importantVar) {
    new MyFactoryChild(_importantVar);
  }
}

contract MyFactoryChild() {
  constructor(uint256 _importantVar){
    doSomething(_importantVar);
  }
}
```

The Data Analyst Resources



```sql
SELECT
    -- child deployment tx_hash
    "tx_hash",
    -- function parameter
    RIGHT(encode("input", 'hex'), 64) as important_variable
    -- bytea2numericpy(substring("input", 8, 64)) as important_variable
FROM
    ethereum.traces
WHERE
    -- factory
    "to" = '\x1f98431c8ad98523631ae4a59f267346ea31f984'
AND
    -- function selector
    LEFT(encode("input", 'hex'), 8) = 'a1671295'
```

The Data Analyst Resources



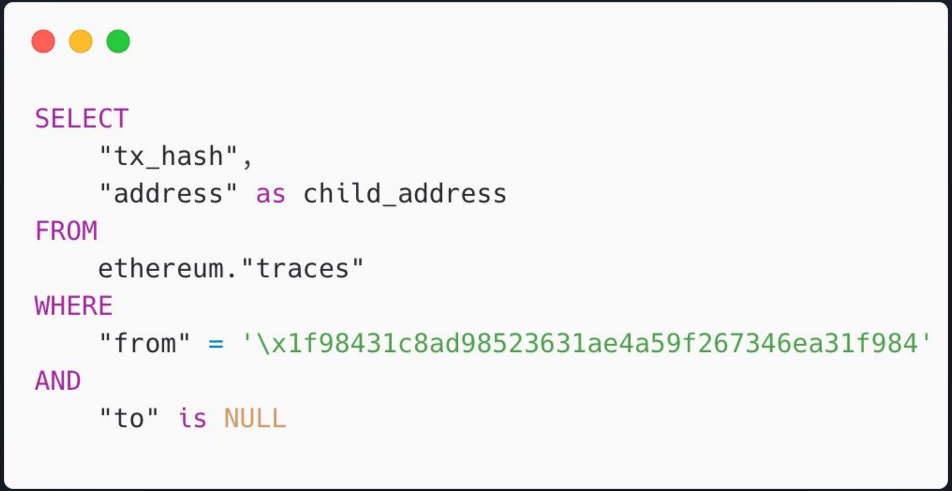
```
SELECT
    -- child deployment tx_has
    "tx_hash",
    -- function parameter
    RIGHT(encode("input", 'hex
    -- bytea2numericpy(substri
FROM
    ethereum.traces
WHERE
    -- factory
    "to" = '\x1f98431c8ad98523
AND
    -- function selector
    LEFT(encode("input", 'hex'
```

Query results New Query

@wei3erHase

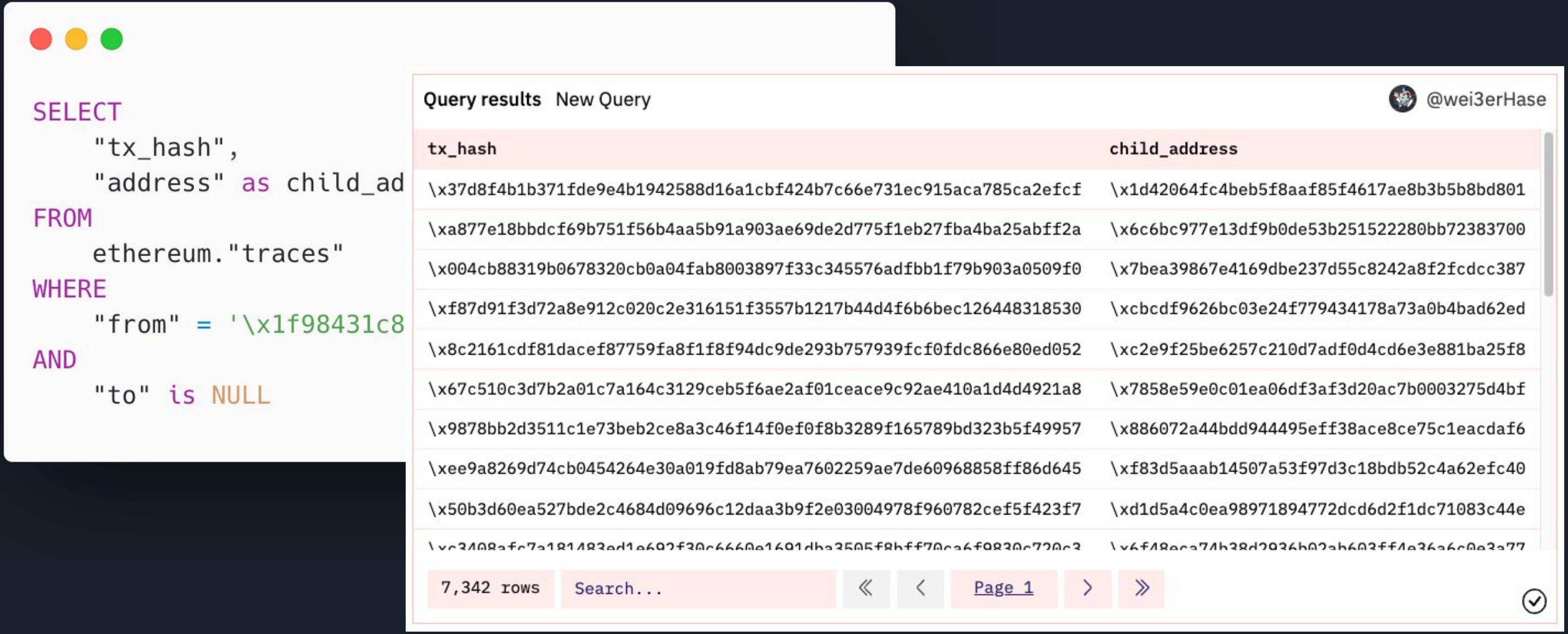
tx_hash	important_variable
\x37d8f4b1b371fde9e4b1942588d16a1cbf424b7c66e731ec915aca785ca2efcf	00000000000000000000000000000bb8
\xa877e18bbdcf69b751f56b4aa5b91a903ae69de2d775f1eb27fba4ba25abff2a	000000000000000000000000000001f4
\x004cb88319b0678320cb0a04fab8003897f33c345576adfbb1f79b903a0509f0	00000000000000000000000000002710
\xf87d91f3d72a8e912c020c2e316151f3557b1217b44d4f6b6bec126448318530	00000000000000000000000000000bb8
\x8c2161cdf81dacef87759fa8f1f8f94dc9de293b757939fcf0fdc866e80ed052	00000000000000000000000000000bb8
\x67c510c3d7b2a01c7a164c3129ceb5f6ae2af01ceace9c92ae410a1d4d4921a8	000000000000000000000000000001f4
\x9878bb2d3511c1e73beb2ce8a3c46f14f0ef0f8b3289f165789bd323b5f49957	000000000000000000000000000001f4
\xee9a8269d74cb0454264e30a019fd8ab79ea7602259ae7de60968858ff86d645	00000000000000000000000000000bb8
\x50b3d60ea527bde2c4684d09696c12daa3b9f2e03004978f960782cef5f423f7	00000000000000000000000000000bb8
\xc3408afc7a181483ed1e692f30c6660e1691dba3505f8bff70ca6f9830c720c3	000000000000000000000000000001f4

The Data Analyst Good Hunch



```
SELECT
    "tx_hash",
    "address" as child_address
FROM
    ethereum."traces"
WHERE
    "from" = '\x1f98431c8ad98523631ae4a59f267346ea31f984'
AND
    "to" is NULL
```

The Data Analyst Good Hunch

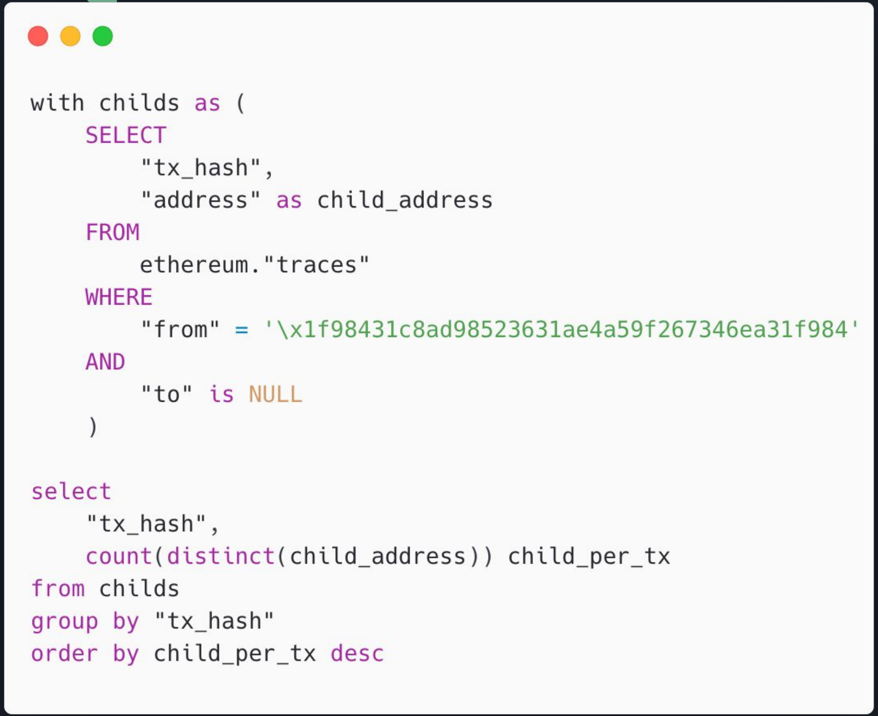


```sql
SELECT
    "tx_hash",
    "address" as child_ad
FROM
    ethereum."traces"
WHERE
    "from" = '\x1f98431c8
AND
    "to" is NULL
```

tx_hash	child_address
\x37d8f4b1b371fde9e4b1942588d16a1cbf424b7c66e731ec915aca785ca2efcf	\x1d42064fc4beb5f8aaf85f4617ae8b3b5b8bd801
\xa877e18bbdcf69b751f56b4aa5b91a903ae69de2d775f1eb27fba4ba25abff2a	\x6c6bc977e13df9b0de53b251522280bb72383700
\x004cb88319b0678320cb0a04fab8003897f33c345576adfbb1f79b903a0509f0	\x7bea39867e4169dbe237d55c8242a8f2fcdcc387
\xf87d91f3d72a8e912c020c2e316151f3557b1217b44d4f6b6bec126448318530	\xcbcdf9626bc03e24f779434178a73a0b4bad62ed
\x8c2161cdf81dacef87759fa8f1f8f94dc9de293b757939fcf0fdc866e80ed052	\xc2e9f25be6257c210d7adf0d4cd6e3e881ba25f8
\x67c510c3d7b2a01c7a164c3129ceb5f6ae2af01ceace9c92ae410a1d4d4921a8	\x7858e59e0c01ea06df3af3d20ac7b0003275d4bf
\x9878bb2d3511c1e73beb2ce8a3c46f14f0ef0f8b3289f165789bd323b5f49957	\x886072a44bdd944495eff38ace8ce75c1eacdaf6
\xee9a8269d74cb0454264e30a019fd8ab79ea7602259ae7de60968858ff86d645	\xf83d5aaab14507a53f97d3c18bdb52c4a62efc40
\x50b3d60ea527bde2c4684d09696c12daa3b9f2e03004978f960782cef5f423f7	\xd1d5a4c0ea98971894772dcd6d2f1dc71083c44e
\xc3408afc7a181483ed1e692f30c6660e1691dba3505f8bff70ca6f9830c720c3	\x6f48eca74b38d2936b02ab603ff4e36a6c0e3a77

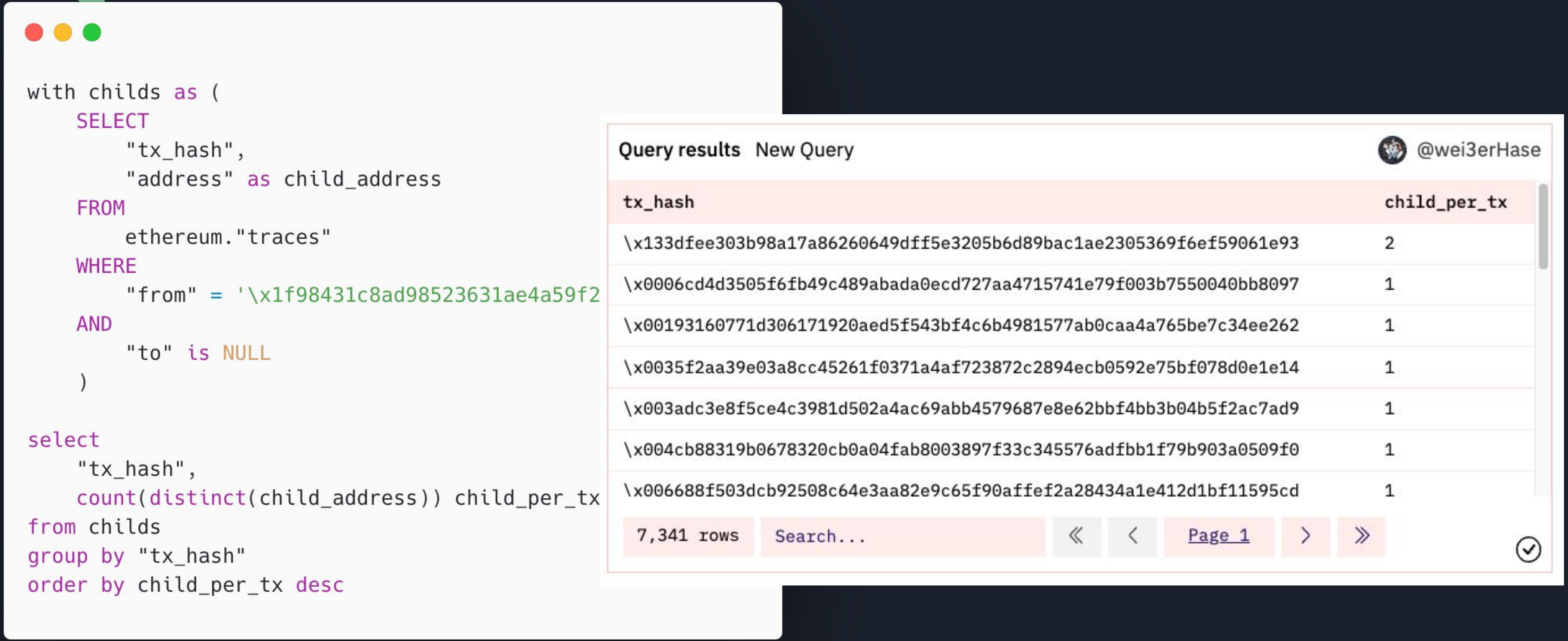
7,342 rows

The Data Analyst Frustration



```sql
with childs as (
    SELECT
        "tx_hash",
        "address" as child_address
    FROM
        ethereum."traces"
    WHERE
        "from" = '\x1f98431c8ad98523631ae4a59f267346ea31f984'
    AND
        "to" is NULL
    )

select
    "tx_hash",
    count(distinct(child_address)) child_per_tx
from childs
group by "tx_hash"
order by child_per_tx desc
```

The Data Analyst Frustration

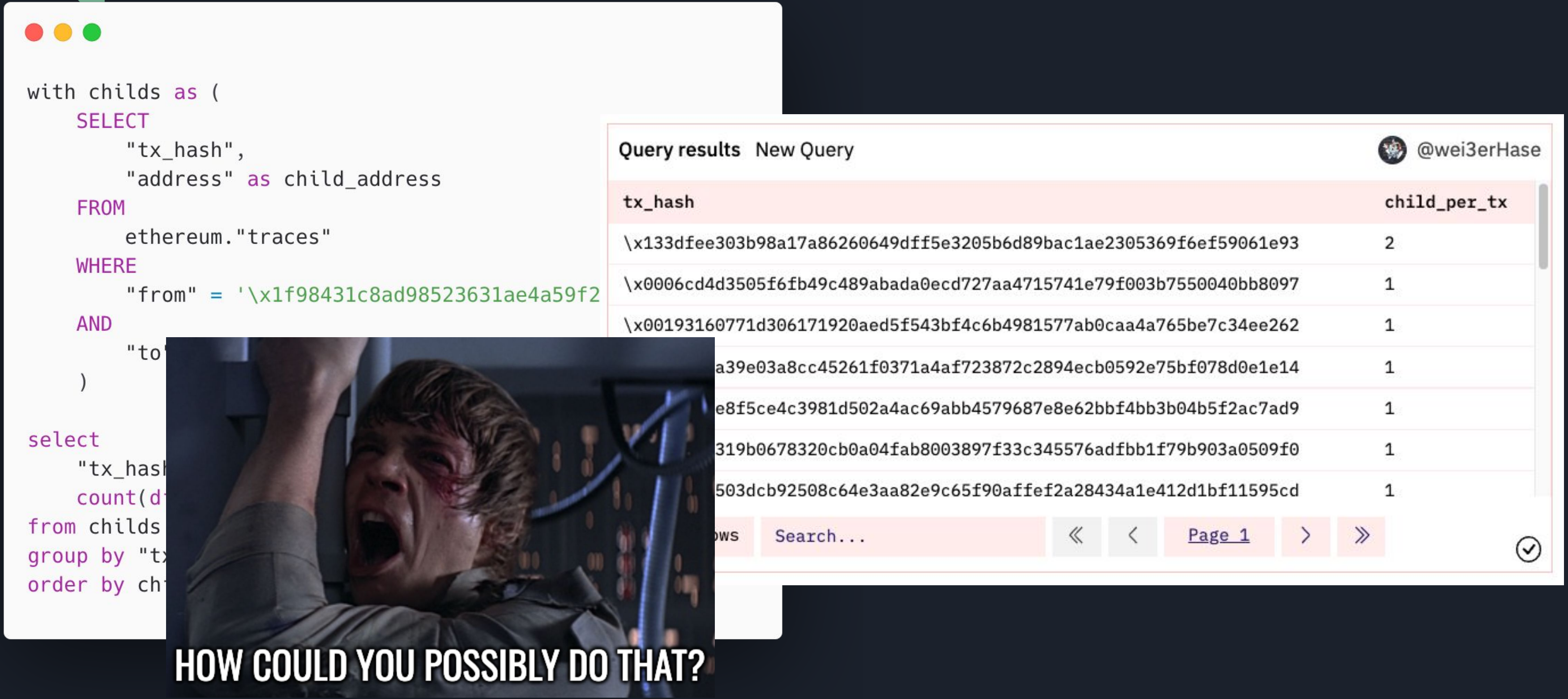


```
with childs as (
    SELECT
        "tx_hash",
        "address" as child_address
    FROM
        ethereum."traces"
    WHERE
        "from" = '\x1f98431c8ad98523631ae4a59f2
    AND
        "to" is NULL
    )

select
    "tx_hash",
    count(distinct(child_address)) child_per_tx
from childs
group by "tx_hash"
order by child_per_tx desc
```

tx_hash	child_per_tx
\x133dfee303b98a17a86260649dff5e3205b6d89bac1ae2305369f6ef59061e93	2
\x0006cd4d3505f6fb49c489abada0ecd727aa4715741e79f003b7550040bb8097	1
\x00193160771d306171920aed5f543bf4c6b4981577ab0caa4a765be7c34ee262	1
\x0035f2aa39e03a8cc45261f0371a4af723872c2894ecb0592e75bf078d0e1e14	1
\x003adc3e8f5ce4c3981d502a4ac69abb4579687e8e62bbf4bb3b04b5f2ac7ad9	1
\x004cb88319b0678320cb0a04fab8003897f33c345576adfbb1f79b903a0509f0	1
\x006688f503dcb92508c64e3aa82e9c65f90affef2a28434a1e412d1bf11595cd	1

7,341 rows

The Data Analyst Frustration

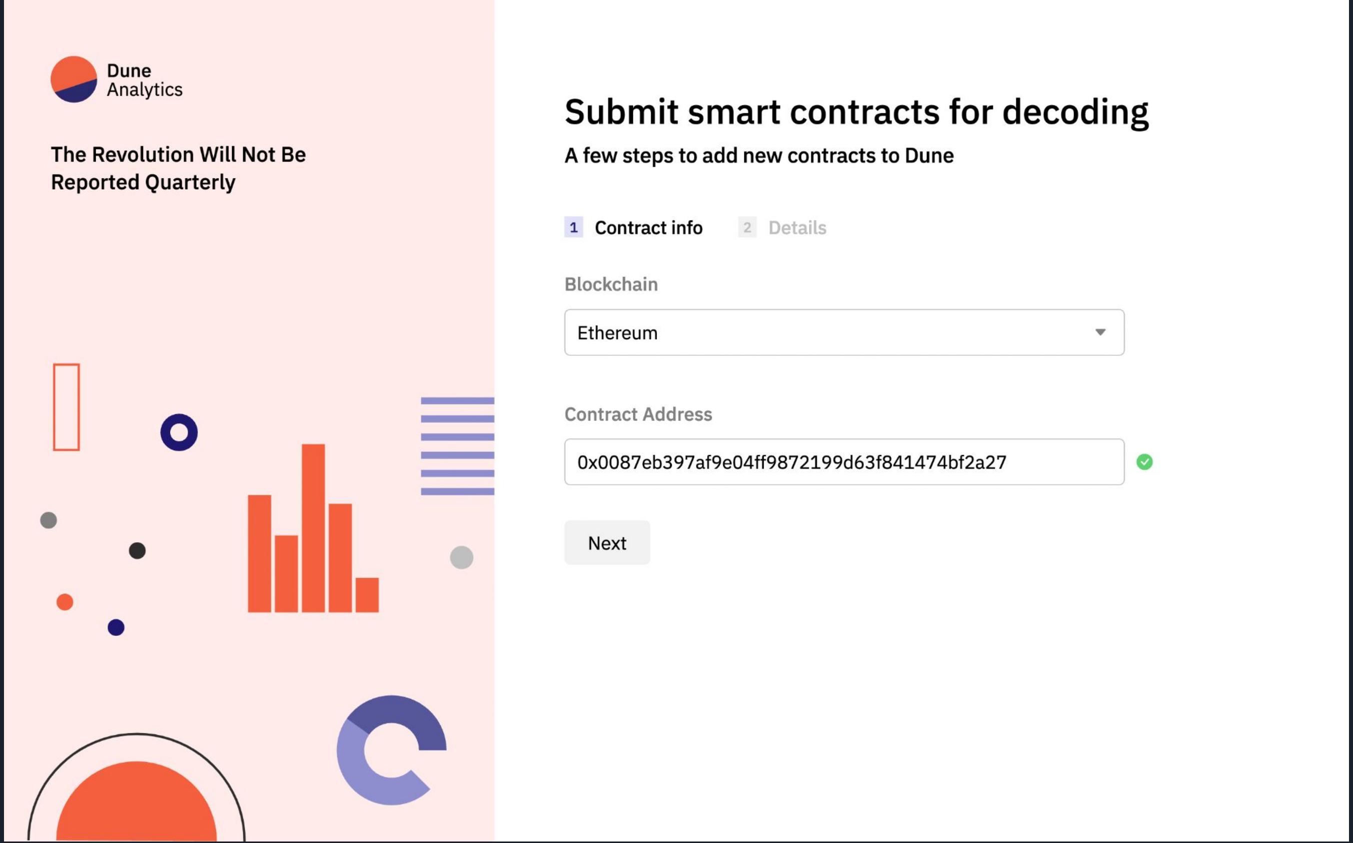


```
with childs as (
    SELECT
        "tx_hash",
        "address" as child_address
    FROM
        ethereum."traces"
    WHERE
        "from" = '\x1f98431c8ad98523631ae4a59f2
    AND
        "to
    )

select
    "tx_has
    count(d
from childs
group by "t
order by ch
```

Query results New Query @wei3erHase

tx_hash	child_per_tx
\x133dfee303b98a17a86260649dff5e3205b6d89bac1ae2305369f6ef59061e93	2
\x0006cd4d3505f6fb49c489abada0ecd727aa4715741e79f003b7550040bb8097	1
\x00193160771d306171920aed5f543bf4c6b4981577ab0caa4a765be7c34ee262	1
a39e03a8cc45261f0371a4af723872c2894ecb0592e75bf078d0e1e14	1
e8f5ce4c3981d502a4ac69abb4579687e8e62bbf4bb3b04b5f2ac7ad9	1
319b0678320cb0a04fab8003897f33c345576adfbb1f79b903a0509f0	1
503dcb92508c64e3aa82e9c65f90affef2a28434a1e412d1bf11595cd	1

What tools are available to make things easier?



The Revolution Will Not Be Reported Quarterly

Submit smart contracts for decoding

A few steps to add new contracts to Dune

1 Contract info   2 Details

Blockchain

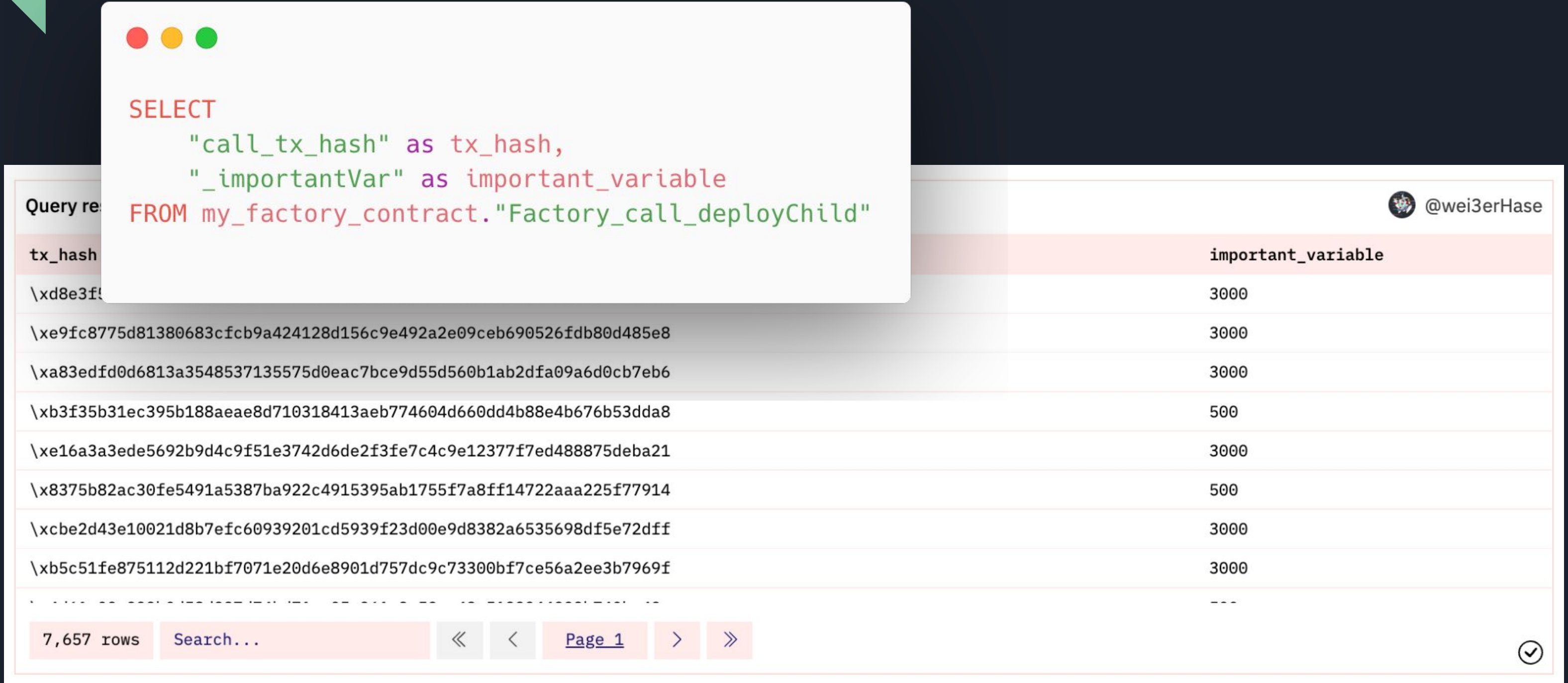
Ethereum

Contract Address

0x0087eb397af9e04ff9872199d63f841474bf2a27 ✓

Dune Analytics: The Data Analyst survival kit.

Preprocessed option - Querying Function calls



```sql
SELECT
    "call_tx_hash" as tx_hash,
    "_importantVar" as important_variable
FROM my_factory_contract."Factory_call_deployChild"
```

Query re

@wei3erHase

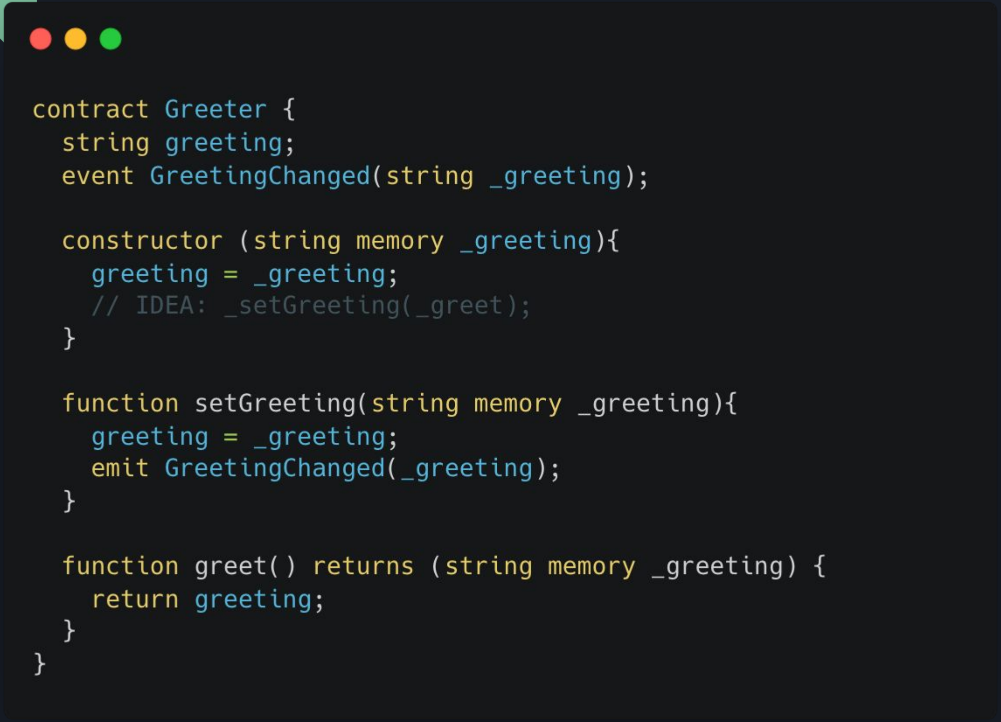
tx_hash	important_variable
\xd8e3f5	3000
\xe9fc8775d81380683cfcb9a424128d156c9e492a2e09ceb690526fdb80d485e8	3000
\xa83edfd0d6813a3548537135575d0eac7bce9d55d560b1ab2dfa09a6d0cb7eb6	3000
\xb3f35b31ec395b188aeae8d710318413aeb774604d660dd4b88e4b676b53dda8	500
\xe16a3a3ede5692b9d4c9f51e3742d6de2f3fe7c4c9e12377f7ed488875deba21	3000
\x8375b82ac30fe5491a5387ba922c4915395ab1755f7a8ff14722aaa225f77914	500
\xcbe2d43e10021d8b7efc60939201cd5939f23d00e9d8382a6535698df5e72dff	3000
\xb5c51fe875112d221bf7071e20d6e8901d757dc9c73300bf7ce56a2ee3b7969f	3000

7,657 rows

DOs & DON'Ts

So as a Solidity Developer, how can you contribute to Data Availability?

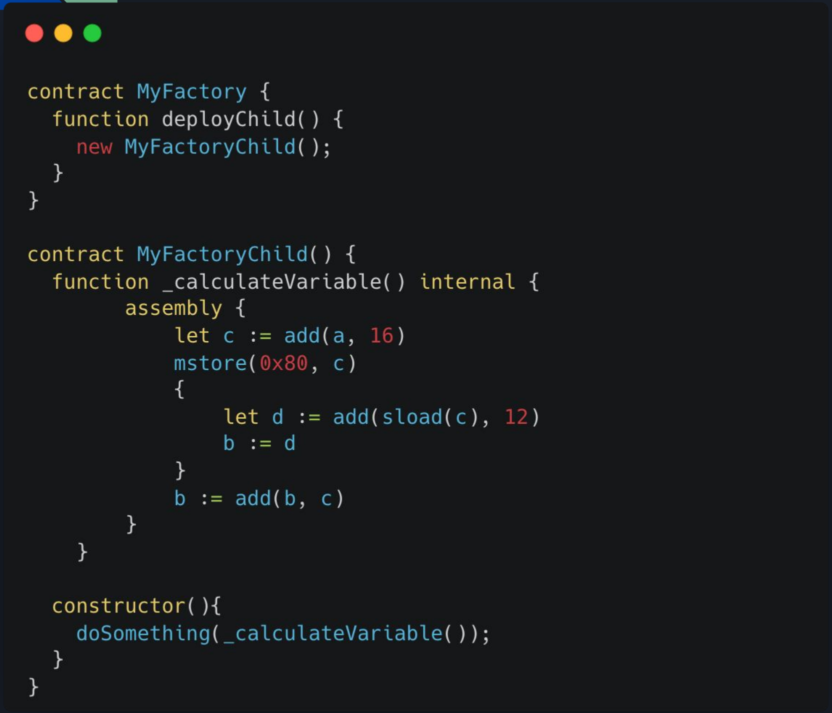
The naming OCD approach



```
contract Greeter {
  string greeting;
  event GreetingChanged(string _greeting);

  constructor (string memory _greeting){
    greeting = _greeting;
    // IDEA: _setGreeting(_greet);
  }

  function setGreeting(string memory _greeting){
    greeting = _greeting;
    emit GreetingChanged(_greeting);
  }

  function greet() returns (string memory _greeting) {
    return greeting;
  }
}
```

I should suggest using an internal function so that emits an event at genesis
Look it says GreetingChanged, has anything changed?
Just saved us 5k gas

The assembly optimizor approach



```
contract MyFactory {
  function deployChild() {
    new MyFactoryChild();
  }
}

contract MyFactoryChild() {
  function _calculateVariable() internal {
        assembly {
            let c := add(a, 16)
            mstore(0x80, c)
            {
                let d := add(sload(c), 12)
                b := d
            }
            b := add(b, c)
        }
    }

  constructor(){
    doSomething(_calculateVariable());
  }
}
```

I've saved 100k gas by using inline assembly and sending instructions straight to the EVM
But you used some of that gas saving to emit an event no?
f*ck it.

1%

danke.